5-2016

Cognitive Big Data Analytics and Persuasive Social Influence Diffusion

Eman Ahmed Ghanim Abukhousa

Follow this and additional works at: https://scholarworks.uaeu.ac.ae/all_dissertations
Part of the Computer Sciences Commons

Recommended Citation

This Dissertation is brought to you for free and open access by the Electronic Theses and Dissertations at Scholarworks@UAEU. It has been accepted for inclusion in Dissertations by an authorized administrator of Scholarworks@UAEU. For more information, please contact fadl.musa@uaeu.ac.ae.
United Arab Emirates University
College of Information Technology

COGNITIVE BIG DATA ANALYTICS AND PERSUASIVE SOCIAL INFLUENCE DIFFUSION

Eman Ahmed Ghanim Abukhousa

This dissertation is submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

Under the Supervision of Professor Yacine Atif

May 2016
Declaration of Original Work

I, Eman Ahmed Ghanim Abukhousa, the undersigned, a graduate student at the United Arab Emirates University (UAEU), and the author of this dissertation entitled "Cognitive Big Data Analytics and Persuasive Social Influence Diffusion", hereby, solemnly declare that this dissertation is my own original research work that has been done and prepared by me under the supervision of Professor Yacine Atif, in the College of Information Technology at UAEU. This work has not previously been presented or published, or formed the basis for the award of any academic degree, diploma or a similar title at this or any other university. Any materials borrowed from other sources (whether published or unpublished) and relied upon or included in my dissertation have been properly cited and acknowledged in accordance with appropriate academic conventions. I further declare that there is no potential conflict of interest with respect to the research, data collection, authorship, presentation and/or publication of this dissertation.

Student’s Signature: [Signature] Date: 23/06/2016
Copyright © 2016 Eman Ghanem Abukhousa
All Rights Reserved
Advisory Committee

1) Advisor: Yacine Atif
Title: Professor
Enterprise Systems Track
College of Information Technology

2) Co-Advisor: Mohammad Mehedy Masud
Title: Assistant Professor
Enterprise Systems Track
College of Information Technology

3) Member: Nader Mohamed
Title: Associate Professor
Networking Track
College of Information Technology
Approval of the Doctorate Dissertation

This Doctorate Dissertation is approved by the following Examining Committee Members:

1) Advisor (Committee Chair): Yacine Atif
   Title: Associate Professor
   Department of Enterprise Systems
   College of Information Technology
   Signature [Signature]
   Date [Signature]

2) Member: Khaled Shuaib
   Title: Professor
   Department of Information Security
   College of Information Technology
   Signature [Signature]
   Date [Signature]

3) Member: Mohamed Adel Serhani
   Title: Associate Professor
   Department of E-Commerce
   College of Information Technology
   Signature [Signature]
   Date [Signature]

4) Member (External Examiner): Morad Benyoucef
   Title: Professor
   Department of Management Information Systems and Analytics
   Institution: University of Ottawa, Canada
   Signature [Signature]
   Date [Signature]
This Doctorate Dissertation is accepted by:

Dean of the College of Information Technology: Professor Omar El-Gayar

Signature [Signature Image] Date June 14, 2016

Dean of the College of Graduate Studies: Professor Nagi T. Wakim

Signature [Signature Image] Date 29 June 2016

Copy 5 of 7
Abstract

Current demands in local and global economies and the pursuit of competitiveness are calling for data-driven strategies. Data-driven solutions analyze trends, make predictions about future events, and prescribe what to do next in an actionable manner. However, cognitive and behavioral data are distinguished by their multiplicity and rapid changes to meet evolving and dynamic goals of individuals. This research work is concerned with the utility of analytical solutions to synthesize and influence cognitive and behavioral adoption. We propose a multidimensional data model to identify and extract cognitive indicators for analysis and persuasive interventions. The process starts by discovering behavioral features to create a cognitive profile and diagnose individual deficiencies. Then, we develop a fuzzy clustering algorithm that predicts similar patterns with controlled constraint-violations to construct a social structure for collaborative cognitive attainment. This social framework facilitates the deployment of novel influence diffusion approaches, whereby we propose a reciprocal-weighted similarity function and a triadic closure approach. In doing so, we investigate contemporary social network analytics to maximize influence diffusion across our synthesized social network. The outcome of this social computing approach leads to a persuasive model to support behavioral changes and developments. The performance results obtained from both analytical and experimental evaluations validate the proposed our data-driven strategy for persuasive behavioral change.

In order to illustrate our solutions, we selected higher education application domain, where our data analytics techniques have shown a potential for maximizing cognitive retention in streamlining career development paths. This is in response to 21st Century professional competencies expected from higher education to deliver career-ready graduates who are immediately ready to meet current industry needs. We deploy our data-driven solutions across a spectrum of career development stages to bridge formal higher education program and industry. Towards closing this digital gap, we propose career development framework that utilizes our cognitive analytics models across three sequential phases: (1) career readiness to measure the general cognitive dispositions required for a successful career in the
21st Century, (2) career prediction to persuade future graduates into a desired career path; and (3) career development to drive growth within a social network structure where social influence mining and persuasive techniques are applied to propagate the adoption of desired career behaviors.

**Keywords:** Learning analytics, big data, social networks, data mining computational science, clustering algorithms, fuzzy logic, career readiness, community of practice.
تحليل البيانات المرفية وتطويرها باستخدام التأثير الاجتماعي في شبكات التواصل الإلكترونية

المتخصّص

تستدعي متطلبات الاقتصاد الحالي والعالمي الحالى والسعى لتحقيق القدرة التنافسية الحصول على استراتيجيات وحلول بيانية تساعدها على التعامل مع التغييرات المستقبلية بطريقة مبتكرة وقابلة للتنفيذ. إلا أن البيانات المرفية والسلوكية للأفراد تميز نمطها ونوعها السريعه تبعاً لتغيير وتطور الأهداف الشخصية للأفراد. يختص هذا البحث بدراسة فائدة الطرق التحليلية للبيانات من أجل التأثير على الإفراز لتحسين التطور المرفي والسلوكي. تفترض في هذا البحث نموذج لجمع وتحليل البيانات المرفية متعددة الأبعاد، بدف تحديد واستخرج المؤثرات التي تضمن أوج القصور الفردية وتسويق بتقديم التغذية الراجعة للفوضي. كما يقدم هذا البحث خواص وراثية تكنولوجية لتوقع الأنماط المرفية المتشابهة وجمعها لبناء هيكل اجتماعي تفاعلي لتحديد الأهداف المرفية. يمثل هذا إطار المجتمع ووضع مناهج مودة للتأثير الاجتماعي، بحث توجيه الأفراد لتنمية معرفيات وسلوكيات معينة. بداية تفترض تقوية النسج الاجتماعي عن طريق التشابه البديلي بين أفراده وإلغاء التكامل الثلاثي، وعند القيام بذلك، تتحقق من تحليل الشبكات الاجتماعية الركبة والكشف عن المجتمعات فيها ومن ثم بناء نموذج إقلاع لدعم التغيرات والتطورات السلوكية. تؤكد التقييمات التحليلية والتجريبية التي حصلنا عليها من الحلول البيانية المفتاحة في هذا البحث الأداء الاجتماعي لهذا الحلول لتحليل البيانات المرفية وتحقيق التأثير الاجتماعي والإقناع لتطويرها.

ومن أجل توضيح حلولنا، اخترنا نطاقاً تطبيقياً في التعليم الجامعي، حيث أظهرت تقنيات تحليلات بياناتنا إمكانية تحقيق أقصى قدر من التطور المرفي في تحسين طرق التحص
المهني، وجاء ذلك استجابة لتحديات القرن الواحد والعشرين للكفاءات المهنية والتأهيل النوعي والخصم الذي يحتاج إلى سوق العمل. لهذا قمنا بتقديم حلولاً بسيطة كآليات لربط خريجين التعليم العالي مع احتياجات سوق العمل عبر مجموعة متنوعة من مراحل التطوير المهني في إطار من ثلاث مراحل متتابعة: (1) الاستعداد المهني لقياس المبولة الحرفية العامة اللازمة لحياة مهنية ناجحة في القرن الواحد والعشرين (2) التوقع المهني لقناعات خريجي المستقبل على اتباع المسار الوظيفي المطلوب في سوق العمل (3) التطور الوظيفي لدفع عينة النمو ضمن هيئة الشبكة الاجتماعية حيث يتم تطبيق التأثير الاجتماعي والتقييمات المختلفة للجهة السلوك المهني المرغوب.

مبادئ البحث الرئيسية: البيانات المرفقة، حلول بسيطة كهندسة، خوارزمية تنظيمية، شبكات التواصل الإلكترونية، نموذج للإقناع والتآثر الاجتماعي.
Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Professor Yacine Atif for the continuous support of my Ph.D study and related research, for his patience and motivation. His guidance helped me in all the time of research and writing of this dissertation. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisor, I would like to thank the members of my advisory and examination committees for their insightful comments and encouragement.

Thanks must also go to faculty and staff members in the College of Information Technology at UAEU who supported this research and whose cooperation and input made this research possible.

My sincere thanks also goes to Mr. Ahmed Taha - Main Library- who participated in this research by finding and advising research materials.

I want to acknowledge a special debt to Dr. Mohammed Abukhousa for his help on MatLab and for the simulating discussions.

Last but not the least, I would like to thank my family: my parents and to my brothers and sisters for supporting me spiritually throughout writing this thesis and my life in general.
Dedication

To my mother who helped me become the person I am
and who valued education above all.
# Table of Contents

Title .................................................................................. i
Declaration of Original Work ................................................. ii
Copyright ........................................................................... iii
Advisory Committee ............................................................. iv
Approval of the Doctorate Dissertation ................................. v
Abstract ............................................................................... vii
Title and Abstract (in Arabic) ................................................ ix
Acknowledgments ................................................................. xi
Dedication ........................................................................... xii
Table of Contents ................................................................... xiii
List of Tables ......................................................................... xvi
List of Figures ......................................................................... xvii

Chapter 1: Introduction ........................................................ 1
1.1 Motivational Scenario ..................................................... 1
1.2 Community of Practice Apprenticeship Model ................. 3
1.3 Research Design and Methodologies ............................... 5
1.3.1 Theoretical and Empirical Developments ................. 5
1.3.2 Our Research Developments ..................................... 7
1.4 Sample Case Study ....................................................... 11

Chapter 2: Research Problem and Contributions ................. 14
2.1 Problem Statement ....................................................... 14
2.2 Scope and Assumptions ............................................... 15
2.3 Research Questions ..................................................... 15
2.4 Research Contributions ............................................... 16
2.5 Thesis Organization ..................................................... 19

Chapter 3: Multidimensional Data Warehousing for Computational Analytics .................................................. 23
3.1 Introduction ............................................................... 23
3.2 Background and Related Work ..................................... 25
3.2.1 Data warehousing in the age of Big Data ................. 25
3.2.2 Data Analytics and Visualization ............................. 31
3.2.3 Data in Higher Education Environments ................. 36
3.3 Cognitive Data Metrics ............................................... 44
3.3.1 Multidimensional Data Model ................................. 44
3.3.2 Data Measurement Instrument ............................... 46
3.3.3 Profiling Structure ................................................. 55
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.3.4 Data Warehouse Architecture</td>
<td>56</td>
</tr>
<tr>
<td>3.4 Data Analytics and Visualization</td>
<td>57</td>
</tr>
<tr>
<td>3.4.1 Data Design</td>
<td>58</td>
</tr>
<tr>
<td>3.4.2 Dashboard Views</td>
<td>63</td>
</tr>
<tr>
<td>3.5 Implementation and Application</td>
<td>68</td>
</tr>
<tr>
<td>3.5.1 Portal Design</td>
<td>68</td>
</tr>
<tr>
<td>3.5.2 Deployment and Evaluation</td>
<td>70</td>
</tr>
<tr>
<td>3.6 Summary</td>
<td>72</td>
</tr>
<tr>
<td>Chapter 4: Predictive Analytics Approach to Career Prevention</td>
<td>74</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>74</td>
</tr>
<tr>
<td>4.2 Background and Related Work</td>
<td>77</td>
</tr>
<tr>
<td>4.2.1 Data Clustering</td>
<td>77</td>
</tr>
<tr>
<td>4.2.2 Semi-Supervised Clustering</td>
<td>83</td>
</tr>
<tr>
<td>4.2.3 Communities of Practice (CoPs)</td>
<td>98</td>
</tr>
<tr>
<td>4.3 Fuzzy Pairwise-constraints K-Means (FPKM)</td>
<td>101</td>
</tr>
<tr>
<td>4.4 Experiment and Performance Analysis</td>
<td>106</td>
</tr>
<tr>
<td>4.4.1 Experiment Environment and Metrics</td>
<td>106</td>
</tr>
<tr>
<td>4.4.2 Artificial Data</td>
<td>107</td>
</tr>
<tr>
<td>4.4.3 Benchmarked Data</td>
<td>111</td>
</tr>
<tr>
<td>4.4.4 Data Simulated from Real World Scenario</td>
<td>119</td>
</tr>
<tr>
<td>4.5 Summary</td>
<td>124</td>
</tr>
<tr>
<td>Chapter 5: Social Learning Analytics for Career Development</td>
<td>125</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>125</td>
</tr>
<tr>
<td>5.2 Background and Related Work</td>
<td>127</td>
</tr>
<tr>
<td>5.2.1 Social Network</td>
<td>127</td>
</tr>
<tr>
<td>5.2.2 Community Detection</td>
<td>130</td>
</tr>
<tr>
<td>5.2.3 Social Learning Analytics (SLA)</td>
<td>138</td>
</tr>
<tr>
<td>5.3 CoP-Network</td>
<td>147</td>
</tr>
<tr>
<td>5.3.1 Distance and Similarity</td>
<td>147</td>
</tr>
<tr>
<td>5.3.2 Reciprocal-Weighted Euclidean Similarity</td>
<td>151</td>
</tr>
<tr>
<td>5.4 Triadic Closure to Enhance Community Detection</td>
<td>153</td>
</tr>
<tr>
<td>5.5 Performance Evaluation</td>
<td>159</td>
</tr>
<tr>
<td>5.5.1 Experiment Setup</td>
<td>159</td>
</tr>
<tr>
<td>5.5.2 Experiment Parameters</td>
<td>159</td>
</tr>
<tr>
<td>5.5.3 Candidate Algorithms</td>
<td>160</td>
</tr>
<tr>
<td>5.5.4 Performance Metrics</td>
<td>163</td>
</tr>
<tr>
<td>5.5.5 Experiment Results</td>
<td>164</td>
</tr>
<tr>
<td>5.6 Examining CoP-Network: A SNA Example</td>
<td>170</td>
</tr>
<tr>
<td>5.7 Summary</td>
<td>177</td>
</tr>
<tr>
<td>Chapter 6: Social Influence</td>
<td>179</td>
</tr>
<tr>
<td>6.1 Introduction</td>
<td>179</td>
</tr>
<tr>
<td>6.2 Background</td>
<td>180</td>
</tr>
<tr>
<td>6.2.1 Social Influence</td>
<td>180</td>
</tr>
<tr>
<td>6.2.2 Structural Influence Diffusion Models</td>
<td>182</td>
</tr>
<tr>
<td>6.2.3 Persuasive Technology</td>
<td>186</td>
</tr>
</tbody>
</table>
List of Tables

<table>
<thead>
<tr>
<th>Table Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Requirements Criteria of Conceptual Multidimensional Data Models</td>
<td>31</td>
</tr>
<tr>
<td>3.2</td>
<td>Emerging Technologies in Education</td>
<td>40</td>
</tr>
<tr>
<td>3.3</td>
<td>Career Dispositions Scores</td>
<td>47</td>
</tr>
<tr>
<td>3.4</td>
<td>Total Variance Explained by Identified Factors</td>
<td>51</td>
</tr>
<tr>
<td>3.5</td>
<td>Users’ Feedback Form</td>
<td>71</td>
</tr>
<tr>
<td>4.1</td>
<td>Popular K-mean based semi-supervised methods</td>
<td>97</td>
</tr>
<tr>
<td>4.2</td>
<td>Notions and Symbols</td>
<td>106</td>
</tr>
<tr>
<td>4.3</td>
<td>Overlap degree Vs Accuracy of FPKM on 2-dimensional Artificial Data</td>
<td>110</td>
</tr>
<tr>
<td>4.4</td>
<td>Precision of FPKM for different number of seeds (IRIS)</td>
<td>113</td>
</tr>
<tr>
<td>4.5</td>
<td>Recall of FPKM for different number of seeds (IRIS)</td>
<td>113</td>
</tr>
<tr>
<td>4.6</td>
<td>F-Score of FPKM for different number of seeds (IRIS)</td>
<td>114</td>
</tr>
<tr>
<td>4.7</td>
<td>Precision of all methods for different overlap region-size (IRIS)</td>
<td>115</td>
</tr>
<tr>
<td>4.8</td>
<td>Recall of all methods for different overlap region-size (IRIS)</td>
<td>116</td>
</tr>
<tr>
<td>4.9</td>
<td>F-Score of all methods for different overlap region-size (IRIS)</td>
<td>117</td>
</tr>
<tr>
<td>4.10</td>
<td>Career Profile %-based probability of attributes</td>
<td>120</td>
</tr>
<tr>
<td>4.11</td>
<td>Performance of all Methods (real scenario, $s^2 = 15$)</td>
<td>121</td>
</tr>
<tr>
<td>4.12</td>
<td>Performance of all Methods (real scenario, $s^2 = 20$)</td>
<td>122</td>
</tr>
<tr>
<td>4.13</td>
<td>Performance of all Methods (real scenario, $s^2 = 25$)</td>
<td>122</td>
</tr>
<tr>
<td>4.14</td>
<td>Performance of all Methods (real data, $s^2 = 30$)</td>
<td>122</td>
</tr>
<tr>
<td>5.1</td>
<td>Features of community detection algorithms</td>
<td>138</td>
</tr>
<tr>
<td>5.2</td>
<td>Five Approaches of Social Learning Analytics A (SLA)</td>
<td>140</td>
</tr>
<tr>
<td>5.3</td>
<td>Five Approaches of Social Learning Analytics B (SLA)</td>
<td>141</td>
</tr>
<tr>
<td>5.4</td>
<td>Vector Representation of Learners Interests in Fig.5.13</td>
<td>153</td>
</tr>
<tr>
<td>5.5</td>
<td>Metrics obtained using Absolute Similarity Threshold filtering</td>
<td>165</td>
</tr>
<tr>
<td>5.6</td>
<td>Metrics obtained using k-Nearest Neighbors</td>
<td>165</td>
</tr>
<tr>
<td>5.7</td>
<td>Average results of different network sizes for TC_CNM</td>
<td>167</td>
</tr>
<tr>
<td>5.8</td>
<td>Average, maximum and minimum $Q$ value of N-size Networks</td>
<td>167</td>
</tr>
<tr>
<td>5.9</td>
<td>Selected key learners using Degree Centrality measure</td>
<td>173</td>
</tr>
<tr>
<td>5.10</td>
<td>Selected key learners using Betweenness Centrality measure</td>
<td>173</td>
</tr>
<tr>
<td>5.11</td>
<td>Selected key learners using Degree Centrality measure A</td>
<td>176</td>
</tr>
<tr>
<td>5.12</td>
<td>Selected key learners using Degree Centrality measure $\mathfrak{B}$</td>
<td>176</td>
</tr>
<tr>
<td>6.1</td>
<td>Influential nodes in CoP Network</td>
<td>199</td>
</tr>
</tbody>
</table>
### List of Figures

| Figure 1.1:   | Career Readiness Framework                                      | 4  |
| Figure 1.2:   | Research Design and Methodologies                               | 6  |
| Figure 1.3:   | A High Level Implementation Model of Our Proposed Solutions     | 10 |
| Figure 1.4:   | The CoP-Apprenticeship Model: IT Scenario                       | 12 |
| Figure 3.1:   | Learning Analytics Reference Model                              | 24 |
| Figure 3.2:   | Career Readiness as a Function of Learner, Career Dispositions and Seamster | 27 |
| Figure 3.3:   | Scatter Plot Matrix of the IRIS Data                            | 33 |
| Figure 3.4:   | IRIS Data Set Visualized by Star Glyphs                         | 34 |
| Figure 3.5:   | Iris Data Visualized by Dimensional Stacking                     | 35 |
| Figure 3.6:   | Key Big Data Opportunities for Three End-Users in Higher Education | 38 |
| Figure 3.7:   | Blended Learning                                                | 42 |
| Figure 3.8:   | Use Case for an Open Learning Environment                       | 44 |
| Figure 3.9:   | Career Dispositions Dimensions                                   | 45 |
| Figure 3.10:  | EFA Steps                                                        | 49 |
| Figure 3.11:  | Find Factorability of the Correlation matrix                    | 50 |
| Figure 3.12:  | Approaches to Decide the Number of Factors                      | 52 |
| Figure 3.13:  | Example of Oblique (Promax) Rotation of SRCDS Sample Data        | 54 |
| Figure 3.14:  | Factors Labeling and Item Selection                             | 55 |
| Figure 3.15:  | Career Profile Structure                                         | 56 |
| Figure 3.16:  | Multidimensional Data Warehouse Architecture                    | 57 |
| Figure 3.17:  | Conceptual Schema for the Base Data for Learners                | 59 |
| Figure 3.18:  | The Conceptual Schema for the Basic Multidimensional Information for the Base Data Considered in Fig. 3.17 | 60 |
| Figure 3.19:  | The Multidimensional Conceptual Schema for the Part of Base Data in Fig. 3.17. | 62 |
| Figure 3.20:  | The Schema of Aggregated Cube View Presenting Learner’s Career Dispositions-by-Domain-and-Semester | 64 |
| Figure 3.21:  | Circle Diagram Visualization Results of Career Dispositions Report of Individual Learner | 65 |
| Figure 3.22:  | Example of Different Visualizations of Career Dispositions       | 65 |
| Figure 3.23:  | Career Dispositions Across Group of Learners                     | 67 |
| Figure 3.24:  | Analytics Dashboard for an Individual Learner                   | 68 |
| Figure 3.25:  | Learner Use Cases                                                | 69 |
| Figure 3.26:  | Mentor Use Case                                                  | 69 |
| Figure 3.27:  | User’s Feedback Results                                          | 72 |
| Figure 4.1:   | Predictive analytics iterative process                           | 75 |
| Figure 4.2:   | Spectrum between supervised and unsupervised learning: dots correspond to points without any labels. Points with labels are denoted by circles, asterisks and crosses. In (c), the must-link and must-not-link constraints are denoted by solid and dashed lines, respectively. | 79 |
| Figure 4.3:   | A dendrogram that Represents Divisive vs Agglomerative Community Detection Algorithms | 80 |
Figure 4.4: Illustration of K-mean algorithm. (a) Two-dimensional input data with three clusters; (b) three seed points selected as cluster centers and initial assignment of the data points to clusters; (c) and (d) intermediate iterations updating cluster labels and their centers; (e) final clustering obtained by K-means algorithm at convergence [1].

Figure 4.5: Example of the fuzzy Semi-supervised learning

Figure 4.6: Different levels of overlap degree

Figure 4.7: Accuracy of three clustering methods (2 - d Artificial Data)

Figure 4.8: F Score of three clustering methods (2 - d Artificial Data)

Figure 4.9: The performance of FPKM method for different number of seeds in the overlapped region size (IRIS)

Figure 4.10: Performance of three methods on IRIS

Figure 4.11: Performance of all Methods on real scenario

Figure 5.1: Social Network Structure

Figure 5.2: Visualization of the steps of Guillaume algorithm

Figure 5.3: Sketches of two graphs (N=20). The left one is below the the 3-clique (triangle) percolation threshold and so only two small 3-clique percolation clusters (distinguished by black and dark gray edges) can be observed. In the right graph, most 3-cliques accumulate in a "giant" 3-clique percolation cluster (black edges).

Figure 5.4: Comparison of the CPM and the CPMw

Figure 5.5: Schematic example of natural community for a node (sky-blue point in the figure). The blue nodes are the other members of the group and have positive fitness within the group, while the red nodes have all negative fitness with respect to the group.

Figure 5.6: Ego networks and “whole” networks

Figure 5.7: Network Density

Figure 5.8: K-core decomposition

Figure 5.9: Centrality Measures

Figure 5.10: Identifying set of key players

Figure 5.11: Tie Strength and Social Network Formation

Figure 5.12: Probability distribution functions for object representation

Figure 5.13: Example of interest tree structure representation in Career Profile

Figure 5.14: Overview of the proposed process to study network enhancements

Figure 5.15: A new edge (shown in red) is created between nodes 3 and 4 as they have a high similarity. Similarity is calculated using Jaccard Index.

Figure 5.16: Two new edges (shown in red) are created between nodes 3 - 4 and 1-2 as the four nodes belong to a quasi-clique, missing only two edges.

Figure 5.17: A new edge (shown in red) is created between nodes 3 - 4 as they have similar interests. The color encoding represents the interests of an individual.

Figure 5.18: A new node is connected (shown in red) to all other nodes pursuing similar career goals and becomes part of their community.

Figure 5.19: A visualization of the community structure at maximum modularity

Figure 5.20: Frequency distribution of the similarity weights (S1)

Figure 5.21: Network built using K-nn method (n= 100, k=5)
Figure 5.22: Differences in graph structures using similarity-based Triadic Closure

Figure 5.23: Q Vs Network Size

Figure 5.24: Q Vs Similarity Threshold (S1) (N = 100, S2 = 85)

Figure 5.25: Q by TC_CNM Vs Similarity Threshold S2 of different network size

Figure 5.26: Q Vs Number of added edges (N = 100)

Figure 5.27: Q Vs Number of added edges (N = 50)

Figure 5.28: CoP network of learners

Figure 5.29: Visualize CoP Network as connections between local and expatriate

Figure 5.30: Visualize connections between learners and mentors in CoP Network

Figure 5.31: Visualization of the Degree Centrality scores in CoP Network

Figure 5.32: Visualization of the Betweenness Centrality scores in CoP Network

Figure 5.33: Visualization of the H&A scores in CoP Network

Figure 6.1: Bernoulli distribution: node $u$ will have fixed probability to influence its inactive neighbor $v_1$, $v_2$ and $v_3$. If $u$ attempt is “successful” node $v$ will be activated otherwise node $v$ will remain inactive.

Figure 6.2: Example of Independent Cascade Model process

Figure 6.3: Example of Linear Threshold Model process

Figure 6.4: Analyzing the Persuasion Context of PDS

Figure 6.5: The proposed Behavioral Change Support System for Career-Adoption (BCSS-CA)

Figure 6.6: FBM factors in our proposed career adoption persuasive model

Figure 6.7: Percentage of influenced vertices when $k = 5\%$
Chapter 1: Introduction

1.1 Motivational Scenario

Worldwide, 38 percent of employers are having difficulties filling available positions in 2015 not because there aren't enough workers, but because of “a talent mismatch between workers' qualifications and the specific skill sets and combinations of skills employers want”[2]. This shortage percentage has been increasing since 2010\(^1\), and it is expected to result in a huge reduction in innovation, creativity and competitiveness [2, 3]. With engineering and IT among the top employers having this difficulty, lack of technical and professional competencies; and lack of experience are among the top reasons for this difficulty with percentages of 34%, 17% and 22% respectively. The lack of suitable talent available in the market is also an issue in UAE [4]. There is a growing skills deficit that endangers the UAE future economic growth and competitiveness despite the high investment in education [5, 6]. Thus, demand is increasing throughout the UAE economy for “the 21st-century skills” that are needed in the job market. A recent survey released in August 2015 indicates that transferable skills – mainly in the form of soft skills – are gaining an increasing importance for employers in UAE [7]. The evidences are thus far provided that new educations approaches are urgent need to prepare graduates to enter the workforce by instilling higher skills and developing a career readiness capacity which are fundamental to success in a knowledge economy [8, 9].

The fact is that higher education has traditionally been notorious for the inefficient use of data to improve the quality and the value of graduates in meeting market needs [10]. Failure to utilize readily evident data and feedback on learning practices to meet market needs, increases further the gap between education and industry and reduces intervention opportunities to prepare graduates for a successful career path and a superior professional performance. On the other hand, the fluid demands of local and global economies and the pursuit of competitiveness prompt a “War for talent” by employers to actively seek out

\(^{1}\)www.manpowergroup.com
graduates who are resourceful and adaptable and who thrive in a changing environment [11]. The pressure induced by education reform and market needs require the integration of a new learning model in higher education to bridge diverse viewpoints and develop a common assertion of what it means to be career ready. Yet, the amount of data available to higher education sector surpasses their utilization opportunities, due to sustained streams of data and the inadequacy of traditional databases to deal with such continuous data flows. This explosion of data led to the emerging Big Data field in order to harvest the wealth of information out of such continuous flows of data streams. In educational contexts, the application of computational techniques for managing, processing and analyzing these large volumes of data prompted the establishment of Learning Analytics (LA) field to improve learning processes [12]. While they are still in their early stages of implementation and experimentation, the interplay between LA and Big Data is expected to revolutionize future higher education [13, 14].

Yet, the world of work is changing at an ever-faster pace, and graduates as professionals will have to acquire new knowledge and skills on an almost continuous basis; and they will have to learn how to apply them effectively in a dynamic context. This new career model has shifted the responsibility for career development from the organizations to the individuals who have to manage their own careers in order to realize their processional growth goals [15, 16, 17]. This situation of responsibility and autonomy challenges graduates who at the beginning stage of their professional career; and also it impacts their career success opportunities [18, 19, 20]. They are now required to develop a set of qualities or career dispositions necessary for building up their competency in learning how to learn, in embracing change, and in making informed career development decisions. This explains why education and applied psychology research associate career success with adaptability and a commitment to lifelong learning, along with mastery of fundamental knowledge, key skills, personal traits and professional dispositions [21, 15, 10, 22, 23].

Developing such a career-readiness capacity requires a sustained and progressive growth of professional habits. These habits or dispositions mature over time through a par-
allel path of professional development through university experience, alongside the formal academic path. However, current methods of teaching and learning in higher education programs are not sufficient to facilitate the development of career-readiness capacity. This has brought the attention to Community of Practice (CoP) as one of the most informal ways to achieve this aim [24]. Actually, the movement toward including CoP as a dominant component in different educational systems has gained momentum since the 1990s [25, 26, 27]. For higher education in particular, many studies addressed the need to move towards a model based on the notion of CoP to better serve the needs of 21st century students [28].

1.2 Community of Practice Apprenticeship Model

In traditional higher education programs, students may spend years learning about a subject (learning about); only after amassing sufficient explicit knowledge, they are expected to start acquiring the (tacit) knowledge or exercise of how to be an active practitioner/professional in a targeted field (learning to be). But viewing learning as the process of joining a CoP fosters a new form of apprenticeship as students observe and emulate mentors, while engaging in a “learning to be” cycle to master the skill of a field. The proposed approach also encourages “productive inquiry” that is, the process of seeking the knowledge when it is needed in order to carry out a particular situated task [25]. This involves acquiring the practices and the norms of established practitioners in the field through early and continuous cognitive and practical apprenticeship experiences. Under the guidance of established practitioners, students work together in a common (virtual) social space and participate in each other’s learning process, while benefiting from mentor’s feedback ([10, 24]. The contemporary Social Networks (SNs) can be employed to build online CoPs within higher education context [29, 30]. Recent research indicates a substantial value of SNs in strengthening student-to-student interactions, enhancing student social engagements, and building campus communities toward improving student learning [31]. Facebook, one of the most powerful SNs, enhances the connectedness as well as social learning in higher education settings [32, 33, 34]; and information-sharing for knowledge development and
innovation[35]. However, seldom has research tapped into the emergence and cultivation of a social structure that emphasizes on the learner and tries to maximize the network in which this learner navigates in support of his or her learning towards new professional practices [36].

In this research work, we adopt the notions of CoP, SNs, LA and underlying concepts that draw from a variety of related research fields including career success, social theories, social influence and persuasion science, education and learning technologies, and data management to introduce a framework to improve career readiness and to enhance the career's success opportunities of learners in higher education institution. Our framework bridges the gap between higher education and industry by constructing an online social structure made up of interconnected CoPs. This structure extends the perspective of educational institutions and develop a joint effort with the industry to leverage education and workforce development. The proposed approach also provides indicators and means for institutions to intervene in order to positively affect career readiness. It also incorporates social influence and persuasion to guide learners to adopt career paths that are currently demanded by the industry. To enable this novel structure, we advocate three major modules as illustrated in Fig.1.1:(1) career readiness, to assert professional dispositions, (2) career prediction to identify-or to persuade- a domain of employment, and (3) career development that evolve into motivation and skills relevant to the predicted and/or persuaded career domain.

![Career Readiness Framework](image_url)

**Figure 1.1: Career Readiness Framework**

Toward implementing our framework, we implement novel Career Dispositions standards and create related mentoring workflows to support a portfolio of assessments that gauge learners' progress along their academic learning as well as career development process. No learner can optimize the developmental realization of this magnitude individually; however, uniting around a common goal is a powerful catalyst for progress. This
is why we propose clustering learners into online (CoP) to bridge diverse viewpoints and develop a joint effort to leverage education and workforce development, and become career ready. A formal mentoring mechanism for salient practices is associated with each CoP to drive its members' academic and professional development. With the emergence of data-intensive universities, the power of LA is unleashed to bring further awareness about career dispositions while producing and consuming learning. This trend will allow universities to see what students see in terms of personal career dispositions to advocate guided-coaching following a critical path analysis which promote goal setting, skill management and performance review. This data-aware analysis supports career preparedness and sustains life-long learning. The proposed methodology to achieve these outcomes consists in first defining and validating standard career disposition dimensions. These intangible disposition indicators are converted into numerical "raw scores" which are then stored in a data warehouse for novel aggregation and analysis methods. This process creates the opportunity to systematically identify similar career patterns to cluster individuals whose career prospects are deemed similar, into networked CoPs. This new online social structure extends the perspective of educational institutions to provide a platform to build up learners' career readiness capacity and evaluate their professional development during the course of their academic study. Next, we present an overview of theoretical frameworks and computing techniques we relay on to develop our research solutions.

1.3 Research Design and Methodologies

1.3.1 Theoretical and Empirical Developments

The theoretical concepts and empirical methods of our work are drawn from a variety of related research in order to understand the problem in its field and apply contemporary computing solutions. As illustrated in Fig.1.2, we survey related literature from multiple disciplines in order to identify personal traits and behaviors, which influence the ability to develop a successful career; and which can be influenced to trigger desired reactions and behaviors. We review the pedagogical approaches that combine learning models and online
activities; and their potential to provide an optimal environment for skills development. We next investigate virtual communities of practices (CoPs) as a social mechanism to collaborative learning. We also explore online social networks practices to extend our knowledge of the role of modern social processes in instilling professional development and career success; and as a possible mechanism to build CoPs. This led us to study social influence across social networks; and how it can be employed to promote behavior diffusion among users in social networks. We accordingly explore the emerging trend of computer-based persuasive technology to influence and change human behavior.

In parallel literature review, we survey the literature of data science to capture current methods to collect and analyze data to support learners in higher education context. This emerging learning analytics field is propelled by the current evolution of multidimen-
sional data and Big Data techniques which we employ for the design of a career readiness warehouse using data mining approaches (i.e. data cube aggregation, data visualization, clustering) to drive individual career success. We also investigate distance and similarity metrics to construct social networks; and algorithms to detect communities on those networks. In this area, we review the most popular information diffusion models to spread influence in social networks; and examined how social network analysis metrics could be used to select initial seeds for these models so as to maximize influence propagation. In addition, we studied the relation between social influence and persuasion with respect to the size of social interactions; and reviewed persuasion profiling and persuasive system design frameworks and models. We present our interdisciplinary literature review in details at each chapter to highlight the gaps, and to propose our solutions to build on existing approaches. Our proposed solutions are briefly described next as within this research developments.

1.3.2 Our Research Developments

Data-Driven Solutions

In our research work, we focus on developing techniques to get more rich and useful information out of learning data. Accordingly, we first design and model a multidimensional data warehouse (Career Readiness Data warehouse) to store various types of data about learners and learning settings from different sources. In this part, we devise a new data structure (Career Profile) to store career-related data; and a new scale to measure and collect career readiness metrics (Self-Reflective Career Dispositions Scale). We then develop multidimensional aggregation functions to deliver analytics results and feedback through a portal design.

We also propose, develop and test a semi-supervised clustering algorithm (Fuzzy Pairwise-Constrained K-Mean) as a predictive model to analyze data from Career Profile in order to predict a hypothetical career practice and bring learners with similar career patterns together into a common cluster. Our method initially form a CoP with a seed
set of learners who can drive the CoP activities and sustain its effectiveness. In doing so, we emphasize the natural overlap nature of industrial needs and career paths by allowing cluster overlap, i.e. each learner to be in more than one cluster. In order to support learning in more social environment, we develop our Reciprocal-Weighted Interest Accumulation Networking (RWIAN) algorithm to construct CoP-Networks by establishing (weighted) links between learners with highest similarity. We also devise a Triadic Closure-based method to enhance community detection in CoP-Network. Finally, we implement social influence mining method with different seeding strategies using social network analytics (SNA) metrics to identity key users to most maximum social influence across CoP-Network. Identified key users are then employed as persuasive agents in our proposed Behavioral Change Support System (BCSS-CA).

Application Domain

Our proposed solutions support the formal curriculum instruction and physical classroom environments in higher education settings with a virtual "cognitive apprenticeship" environment synthesized by CoP model. This social structure influences 21st Century education to narrow the industry gap by guiding learners towards a desired career path. Ancient apprenticeship methods helped earlier learners seeing parents or mentors plant or harvest corps with other partners, and piece together garments under the supervision of a more experienced tailor. We use this inspiration to augment formal schooling with the process of becoming a member of a mentored CoP that supports a successful career, immediately upon graduation. This process involves developing an identity as a member of a community. The process starts by joining the most suitable CoP based on initial career dispositions data and advertised career profile interests. CoP provides an apprenticeship model to promote learning environments which render key aspects of a discipline and make domain-specific practices visible to learners, while still enrolled in academia. Implemented CoP as online social network acts as a virtual classroom where social interactions and collective intelligence contribute to the development of individual career interests under a supervision of
academic mentor and/or career-domain expert professionals.

The learning experience with CoP apprenticeship model starts as learners fill a self-report tool in order to measure initial career skills; and capture career interests in career profile. Career profile data are then used to cluster learners of similar career patterns to participate on CoPs. Learners of similar CoPs are connected by their common social interest to build more focused Cop-Network. Learners interactions aim potentially to support them developing the required career skills, where the development of these skills is super-intended by a mentor and related education and career professionals to manage the learning process within CoP-Networks; and to track and analyze progress and difficulties to propose appropriate interventions promptly.

The high level implementation model of proposed solutions in relate to the three major modules is in presented in Fig.1.3. Career readiness phase is composed of portal module used to measure the general professional dispositions required for a successful career; convert them into analytical results; and to valise findings to provide feedback in understandable and actionable manner.

Career prediction phase target to cluster learners who share similar career interests to initiate formation of CoPs. Learners may actually be assigned to several CoPs according to their interests, which result in potential overlap between CoPs as learners interests may initially span multiple specialty prospects. At the hub of each CoP, there is a group of learners who displayed a high level of career dispositions (inferred from the portal analytics in the previous step). These seed learners support the elaboration of relationships with other learners within selected disciplines labelling the CoP. Accordingly, we propose a semi-supervised clustering method within which side information based on a prior knowledge of the problem domain is utilized to guide the clustering process. Our model also suggests to survey the current industry needs as part of CoP metadata. Each CoP is assigned an expert mentor to operate the community synergistic relationships. This includes sharing experiences and learning resources to sustain the development of interest and skills of community members in a collaborative effort. Our model suggests a new role provided
by the industry that is incorporating representative professional with a pedagogical profile to mentor the community. CoP admits automatically all learners who pass the disposition threshold and meet the advertised discipline by the CoP.

After the initial formation of CoPs, we move into a more precise representation of the CoPs by building a classification model, which considers each CoP as a class and each learner belonging to a CoP as a member of the corresponding class. The learners' profiles and other attributes, along with the CoP membership information would be used as a training data for a classifier. A similar but independent classification model could also be produced, to serve a recommendation system for the mentors assigned to a particular CoP based on their own expertise and the generic profiles within each CoP. Classification model is highlighted in gray in Fig.1.3 as it is not handled in this research work. However, it is a
straightforward deployment that can be considered as an output from this research work.

Career development phase aims at constructing CoP-Networks of dense community structure to connect learners who belong to the same CoP based on their social similarity. Learners in such networks can benefit from the power of collaborative knowledge towards achieving career goals and objectives. This module also proposes using social network analytics (SNA) engine to investigate networking process, roles, properties of ties, relationships and how learners develop and maintain these relationships to support their career development. SNA metrics are also used to explore social influence propagation in the CoP-Network; and to identify most influential learners based on their structural position in the network. Those learners are then employed as "persuasive agents" in our proposed behavioral change support system (BCSS) to perused learners to adopt career path on demands by industry; and to accordingly develop required skills.

1.4 Sample Case Study

Nowadays, Computer Science and Information Technology (IT) graduation and post-graduation qualifications come at the top of educational and academic qualifications that are emphasized by UAE market when looking for suitable candidates [7]. In UAE and worldwide, the most in demand IT careers are mainly in the areas of data science (including data modeler, data analyst and business intelligence developer), network architecture, computer system engineering/analysis, software engineering/analysis and security analysis [37, 38]. It is very interesting how the title "analyst" appears almost in every job in demand. One more interesting observation is the emerge of new IT job titles: social medial director and chief data officer [38]. Using this input from industry market, we analyze a typical higher educational program as a scenario-based mapping to our CoP-Apprenticeship model. In IT college, learners are expected to complete general education, college requirements, major requirements, and the apprenticeship program. The major requirements in the Bachelor of Science in Computer Science for example include a variety range of courses such as programming, security principles, artificial intelligence, database systems and software en-
engineering fundamentals. Our model fits in this scenario to drive learners into professions in demands for next years (upon their graduation) at early stage of their journey as illustrated in Fig 1.4.

At the first stage, learners of IT college fill out the Career Profile where they provide information about their current competencies, qualifications, skills; and also list their future career interests. Learners also are required to complete the Career Readiness instrument in order to measure their Career Dispositions. The provided instrument presents a storehouse view of career dispositions relevant to the general IT practitioner through an integrated portal which captures self-stated learning experiences and converts them into analytical results to diagnose deficiencies and prescribe improvement recommendations.

![Diagram of CoP Apprenticeship Model: IT Scenario](image)

**Figure 1.4: The CoP-Apprenticeship Model: IT Scenario**

In the next step, the Career Predication module allocates and connects learners who
share similar career interests to initiate the CoP experience. These interests are analyzed against the market needs in order to map the CoP construct to a certain career path (or overlapped paths) that is highly demanded (i.e. Data Science). Each CoP will be assigned an academic supervisor and/or industry professional to expert mentor this community. The mentor and the initial members create the structure and processes for how their community will to operate. This includes sharing experiences and learning materials to sustain the Career Development of community members in a collaborative effort. To support learning relationships, we reinforce the learners’ ties by their social similarity by developing CoP-Network. The members’ constant interactions within CoP and developed CoP network create a live or a dynamic knowledge container and a repertoire of shared practices and experiences. Our dynamic CoP structure evolves to be more focused domains by forming a sub-CoP of all learners who are “influenced” to tranche their initial major into sub-disciplines (i.e. Data Modeler, Data Analyst, Business Intelligent Analysts, Data chief officer), forming new CoPs as illustrated further in Fig. 1.4. As the community thrives, learners develop their practice domain, and may recognize and then reach out for other potential members (outside their CoP) who should belong to their community (to migrate to other CoPs i.e. statistics, semantics web, governance).
Chapter 2: Research Problem and Contributions

2.1 Problem Statement

Today's problems call for multidimensional data structures to infer patterns based on data analytic models in order to simplify and guide decision-making processes, and actions adoption. This process is further exacerbated when dealing with cognitive and behavioral issues that are notoriously difficult to grasp and analyze. Collecting indicative data to diagnose these issues is the first challenge along this process to treat and visualize results out of a continuous stream of data, shaping contemporary data-intensive applications. The next challenge is to predict clusters, which organize data based on soft assignments that allow controlled constraint-violations. This challenge contrasts with traditional and rather rigid clustering methods. Dealing with clusters in Big Data age falls in social analytics context where multi-attribute community detection schemes are expected to optimize the density and information flow transmission across social structures. However, there are currently limitations perceived in social influence propagation domains when further argumentation, rather than simple information, is required to trigger adoption of certain desired behaviors.

Various application domains face the above challenges. For example, traditional higher education institutions struggle to utilize data-driven solution to address learning and behavioral actions of students towards their future career. This has evolved into the notorious gap between education and industry as illustrated earlier in Chapter 1. In this application context, there is a need to support current academic methods of teaching and learning to deliver career-ready graduates who are immediately ready to meet the industry needs. However, developing career-readiness capacity requires a sustained and progressive growth of professional habits. Synthesizing these habits into well-defined data dimensions creates a social platform that nurtures their development through an educational environment that cluster learners into specific domain-related skills used to claim career-readiness upon graduation.
2.2 Scope and Assumptions

The scope of our thesis is driven by novel data analytics schemes that span multiple grain of data patterns starting from individual attributes to community structures. This theoretical scope is tightly related to our motivational scenario to justify the advocated solutions in a specific application context. Hence, our domain of research is aligned with the emerging social learning analytics (SLA) area. SLA highlights the social perspective of learning that new skills and ideas are developed and passed on through interactions and collaboration; and that learning cannot be understood without reference to context [39]. Accordingly, SLA employs types of analytics to make use of data generated by learning activity in social settings [40]. In particular, the focus of SLA is on processes in which learners are not solitary, and are not necessarily doing work to be marked, but are engaged in social activity, either interacting directly with others (i.e., messaging, friending or following), or using platforms in which their activity traces will be experienced by others (i.e., publishing, searching, rating or tagging) [41].

Towards constructing the platform for learning social activities and analytics, we collect multidimensional data to profile learners and to build social communities based on similar attributes. functions. In our work, we assume the reliability of self-reported data based on individuals' awareness of the research objective. This assumption is alleviated by the expected motivation of learners to share their behavioral data as it is poised to improve their professional development and so to enhance their future career advancement. We also assume the involvement of academic mentorship and industrial partnership to support changes towards desired behaviors.

2.3 Research Questions

To target the challenges discussed in the problem statement, our thesis addresses the following research questions:

1. How to quantify and measure cognitive dispositions?
2. How to aggregate cognitive dispositions for visual analytics?

3. How to allow a data instance to be assigned to multiple clusters for predictive analytics?

4. How to decide if a violation of the link constraint in the semi-supervised constrained K-Means clustering should be penalized?

5. How to allow self-input while matching users profiles to establish links in social network?

6. How to incorporate profile attributes to enhance community density in social network?

7. How to expand the impact of social influence to achieve behavioral changes?

2.4 Research Contributions

In this research work, we adopt the notion of CoP-based apprenticeship model, SNs concepts and emerging LA to introduce a framework to improve career readiness and enhance the career's success opportunities of learners in higher education institution. As illustrated in the framework consists of three major processes: 1) career readiness, 2) career prediction, and 3) career development. The purpose of career readiness process is to assess current career readiness by measuring the general professional dispositions required for a success career in the 21st Century. Career prediction uses career readiness measurements along with learners' information (e.g. career interest, domain-related qualifications) to predict their future career path and assign them to CoPs within which they can work collaboratively to advance their professional development as well as their academic performance. The social ties within and among online apprenticeship CoPs realize our proposed (virtual) career development social structure, that unite like-minded learners with common career prospects and expert mentors. Career development utilizes the constructed social structure to maintain a potential influence from peers across CoPs to keep learners' horizons open in adjusting their career plan. Thus, the main contributions of this work are presented are as
follows:

Contribution 1: Cognitive Apprenticeship  Introduce a cognitive apprenticeship process in formal education to elevate career-readiness habits, and help individual learners taking a greater responsibility of assessing deficiencies and constructing their own professional profile.

Contribution 2: Cognitive Dispositions  We develop the conceptual basis of career dispositions as a six-dimensional model that describes attitudes, skills and preparations of each individual towards a professional practice. The 6-dimensional construct comprises: Openness to challenge (OC), Critical Thinking (CT), Resilience (R), Learning Relationships (LR), Responsibility for Learning (RL), and Creativity (C). This model can be used to diagnose and improve career success related skills and behaviors of current learners in higher education institutions.

Contribution 3: Cognitive Attainment Metrics  We introduce a reliable and valid metrics to assess cognitive dispositions attainment, such as a Self-Reflective Career Dispositions Scale (SRCDS). This particular scale is presented as a 35-items questionnaire against which respondents are asked to rate themselves using a five-point Likert scale from 1 (Not at all like me) to 5 (very like me). This scale reveals early indicators of learner’s generic professional behavior.

Contribution 4: Profile Data Structure and Multidimensional Data Warehouse  We introduce a Career Profile construct that is designed as a standard mean to collect and access information about learners while they are moving towards a predestined career path. Career profile augments an existing IEEE Learner Information Package (LIP) standard to capture learning data as well as career indicators. Our propose construct of career Profile is structured into three main categories aimed at predicting and assisting learners with their career development throughout their formal education. We use LIP-defined Competency
and Goal categories to specify domain-related qualifications, and long term career objectives of individual learners. We also introduce a new category labeled Professional as a slot for career dispositions ratings and other generic attributes pertaining to career readiness. We also propose a model for multidimensional data warehouse to store a large collection of career-oriented, integrated, time-varying, and non-volatile data in support of career development. Our research results in multidimensional logical views about learners and their career readiness evolution. This facilitates the development of LA methods and clustering process of learners into CoPs.

**Contribution 5: Aggregated Analytic Visual Views of Cognitive Dispositions**  We implement a diagnostic LA tool to analyze and visualize information on collected career data stored in Career Readiness Warehouse; and deliver it to users through a portal design. Our LA tool allows learners to view their current data and feedback as well career-guidance information via a friendly Analytics Dashboard. It also enables mentors and academic supervisors to track and view performance of learners in simple and aggregated analytic visual presentations.

**Contribution 7: Fuzzy Semi-Supervised Clustering**  We devise a fuzzy semi-supervised clustering method to utilize data from career profiles and market outlooks, to cluster learners into desired communities with similar patterns, under the mentorship of an expert coach. We emphasized the natural overlap nature of industrial needs and career paths by allowing each learners to be in more than one cluster. This predictive approach provides learners and higher education institutions with a platform to streamline career identification and cognitive apprenticeship matchmaking processes, to embrace prospective careers that are aligned with industry needs.

**Contribution 9: Reciprocal-Weighted Similarity Function**  We propose a reciprocal-weighted euclidean similarity function \( RWD \) to compare profiles and establish links between participants in forum-based CoP to construct CoP-Network. \( RWD \) function considers
self-assigned weights and position orders to profile attributes, when determining similarity measure across profiles.

**Contribution 10: Triadic Closure to Enhance Community Detection**  We propose an algorithm to build strong communities over the CoP-Network using a Triadic Closure as an attribute based enhancement that captures socio-demographic features and interests of learners (represented by their nodes in the network) and use these properties to augment the network. Our algorithm then applies a quality-optimized version of CNM algorithm to detect communities to demonstrate that our technique derive better quality community structures.

**Contribution 11: Social Network Analytics to Maximize Influence Diffusion**  We employ social mining techniques in order to select initial seed nodes effectively so as to maximize influence diffusion in CoP-Network. We use different social network analytics metrics for seeds selection then apply two influence diffusion models to compare the effect of differing seeding strategies.

**Contribution 12: Persuasive Approach for Behavioral Change**  We propose a persuasive approach for Behavioral Change Support System for Career-Adoption (BCSS-CA) by identifying the most influential individuals in the CoP-Network to be hired as persuasive agents. Persuasive agents be will treated by different persuasive strategies according to their "persuasive profile" to get them to adopt the desired career behaviors. Pervasive agents are then expected to leverage their powerful structure position in the CoP-Network to influence other learners to adopt same behaviors as required by the persuasion source (i.e. human resource planning authority, recruitment agency).

2.5 Thesis Organization

The rest of this thesis is organized as follows. In Chapter 3, we introduce the basic concepts related to multidimensionality and multidimensional data warehouses. We review-in
particular— the extended entity-relationship (EER) models as conceptual data models that are capable of describing the data requirements or conceptual schema for a multidimensional data warehouse in a direct and easy to understand graphical notation. We discuss how to analyze and represent multidimensional data to allow making visual insights into the data set analyzed. We also review data sources in higher education environment; and current gaps and trends in using this data for improving learning processes. We then present our proposed career readiness data metrics; and the process to develop and validate the instrument to collect this data. We discuss our model for career readiness multidimensional data warehouse a centralized repository for different analytics and visualizations queries. We discuss as example the multidimensional aggregation functions on career readiness metrics; and present the resulting data visualization views. Finally, we present the portal design to collect and display data for both learners and mentors; and the evaluation feedback received on portal design and functionality.

In Chapter 4, we introduce predictive analytics as they are increasingly used in higher education to gain more value out of large amounts of raw data for prediction and intervention purposes. We discuss data clustering as one method to analyze multivariate data in order to classify students and to predict their behavior. We focus our discussion on partitional semi-supervised clustering and review major semi-supervised methods that based on K-means. We also discuss CoP concepts, features, development, benefits and sustainability. We then reveal our Fuzzy Pairwise-constraints K-Means (FPKM) algorithm to cluster learners into overlapped CoPs as a prediction of their future career practice. Finally, we present and discuss our experimental results of applying our proposed algorithm on different data sets. Our findings show the effectiveness of our proposal when overlap degree increases.

In Chapter 5, we highlight the importance of SNs as a platform for social learning; and discuss underlying networking structures and mechanisms. In particular, we illustrate and discuss closure and brokerage mechanisms and their roles in identifying communities and structural positions of vertices in network; and in facilitating information spread. We
also define community structure within network and describe most important community detection algorithms. In parallel, we discuss social network analysis (SNA) for the visualization and analysis of social networks structures and mechanisms. We then reveal our method to construct CoP-Network a CoP-Network of a dense community structures. We introduce a Reciprocal-Weighted Euclidean (RWE) similarity function that considers order and weight assigned by learners to their attributes (career and social interests) and which are employed to match learners of high similarity. We explain the hierarchical representation of Career Profile to capture the order and weight of each single attribute. Learners of a match score that is above certain threshold are linked by a weighted edge \( \text{weight} = \text{matchscore} \) in the CoP-Network. We also discuss our a Triadic Closure approach (TC-CNFM) for community detection enhancement in the generated CoP-Network. We present and discuss our experimental results. The findings show that our method resulting in forming densely connected community structures within CoP-Network. Finally, we apply SNA as a case study to examine the structure of CoP-Network in a way that allows visual mapping and quantitative analysis. We also provide example of several SNA metrics that can be used to identify key users and key ties in the network depending on the purpose of the analysis.

In Chapter 6, we build in our work from previous chapter to introduce social influence across network and discuss reveal means to maximize social influence by identifying and using the most influential users in a social network. We describe two-widely recognized social influence models: Independent Cascade (IC) Models and Linear Threshold (LT) and discuss the problem of influence maximization under the two models. We review and discuss the emerging filed of persuasive technology; and how it uses the means and insights of social theories to change attitudes and behaviors of groups of individuals in an intended direction (i.e. adopting certain career paths). We then present our persuasive approach to support career adoption in higher education in coordination with human resource planning authorities and recruitment agencies. Our model propose employing set of key users as a persuasive agents to propagate the desired behavior among other users. For purpose of identifying most influential key users in social network, we introduce a social
mining technique to analyze social influence in CoP-Network using five different seeding strategies under the two popular influence maximization models IC and LT. Our technique allow us to learn about important structural metric(s) (i.e., coreness); and to identify most influential individuals by each seeding strategy. We also introduce a model for stimulating selected key users to act as persuasive agents according to their specific Persuasive Profile.

In Chapter 7, we provide concluding remarks of this dissertation and discuss directions for future research.
Chapter 3: Multidimensional Data Warehousing for Computational Analytics

3.1 Introduction

"Data: the prerequisite for everything analytical. You can't be analytical without data and you can't be really good at analytics without really good data" [42]. New computer-supported interactive learning methods and tools, intelligent tutoring systems, simulations, and games have opened up novel opportunities to collect massive quantities, ranges and scales of data about learners and their activities throughout their education. This explosion of data is aligned with the emerging Big Data field in order to harvest the wealth of information out of such continuous steams of data. In educational contexts, the application of computational techniques for managing, processing and analyzing these large volumes of data prompted the establishment of new computational analytics techniques to improve learning processes. Learning Analytics (LA) is an emerging research line of data analytics that focuses on educational data and which is presented as a rising new trend in higher education as discussed further in Section 3.2.3. LA is defined on LAK11 website 1 as "the measurement, collection, analysis and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and the environments in which it occurs"; while Siemens [43] describes LA as “the use of intelligent data, learner-produced data, and analysis models to discover information and social connections for predicting and advising people's learning”. According to the 2010 Horizon Report for Digital Education [44], LA refers to the "interpretation of a wide range of data produced by and gathered on behalf of students in order to assess academic progress, predict future performance, and spot potential issues". As a concept, LA is drawn from contemporary computational fields such as data mining research with applications to education [45]. A typical LA model (shown in Fig. 3.1) has four key components: data and environment (what kind of data to collect and analyze), stakeholders (who is targeted by the analysis), objectives (why analyzing the collected data) and methods (how to perform the analysis) [46, 47].

1https://tekri.athabascau.ca/analytics/
LA employs different techniques depending on the objectives of the analytics task. However, the challenge is to design and develop effective, usable and useful analytical tools that may help education stakeholders such as learners and teachers to achieve their analytics objectives without the need for having an extensive knowledge about the techniques underlying these tools. Effective tools minimize the time frame between the analysis and the actions leading to prompt corrective interventions [48, 49]. These tools are expected to provide an appropriate visualization framework that significantly contribute to reveal patterns and trends across large amounts of data [50, 51]. The goal of information visualization is to transform data into perceived knowledge to gain insights into processes and relationships and guide improvement practices [48].

In this chapter, we present a multidimensional data model to design, analyze and visualize cognitive skills developmental data (to derive career professional attainment). We propose the structure of a Career Readiness Data Warehouse to collect historical data of various types and from different sources toward supporting career readiness within higher education contexts. We also present a measurement Scale to capture personal traits and
other soft skills, which we label as “Career Dispositions”. These attributes define attitudes and behaviors of learners in terms of their growth towards career practices. This interdisciplinary part of our work reflects research studies in areas related to the relationship between personality traits and career success. The prominent Big Five model of personality dimensions as well as attributes supporting long-life learning [52, 53] contributed to the definition of our career disposition model. We propose also a computational tool to track the progress of learners in developing their career dispositions across our proposed data analytics model.

The rest of this chapter is organized as follows: Section 2 surveys the background and related work in the following relevant research areas: data warehousing, multidimensional data modeling, as well as data analytics and visualization. This section also provides an overview about data-intensive environments in higher education to identify gaps and trends. In Section 3, we present the design and validation of our proposed analytical instrument to collect and measure cognitive data metrics. Section 4 reveals our LA tool to analyze and visualize collected dispositions data. Section 5 describes the implementation and evaluation of the analytic model via a portal application. Finally, Section 6 summarizes this chapter with concluding remarks.

3.2 Background and Related Work

3.2.1 Data warehousing in the age of Big Data

Multidimensionality

Data warehousing systems (DWS) homogenize and integrate massive historical and multidimensional data of organizations in order to extract relevant knowledge and support decision making [54]. The data distinguishing the objects are called multidimensional data if the objects \( X_i = (x_{i1}, x_{i2}, ..., x_{im}), i = 1, ..., m \) are described by more than one feature. If the number of features is \( n \), then \( X_1, X_2, ..., X_m \) are the \( n \)-dimensional data items that are interpreted as points in the multidimensional space \( R^n \), where \( n \) defines the dimensionality of the
space, and and \( m \) is the number of points in the data set. The coordinate values of point \( X_i \) are values of the features \( x_{i1}, x_{i2}, \ldots, x_{im} \). If the data set consists of a large number of objects (i.e., large \( m \)) then the data set is called a large data set. If the number \( n \) is large, then the data set is called a high-dimensional data set [55]. The recent developments in database technologies made it possible to collect and maintain large and complex amounts of data in many forms and from multiple sources referred to as Big Data. Big Data describes data that is fundamentally too big and moves too fast, thus exceeding the processing capacity of conventional database systems [56]. It also covers the computational techniques and technologies to turn this complex data into meaningful patterns and visualize large-sized data sets with diverse structures [49, 48, 56].

Thus, the multidimensionality in data is a technique to represent data as it is placed in an n-dimensional space using fact/dimension dichotomy [57]. The multidimensionality conceptual view allows better understanding and analysis of data in terms of the subjects (facts) and the different and range of views where a subject can be analyzed from (dimensions). Each dimension is associated with hierarchal levels which contain consolidated data or descriptors. On the other hand, a fact contains measures (also known as variables or metrics). One fact and several dimensions to analyze it define the data cube or a simple aggregation function [58, 57]. For example, Fig. 3.2 shows our Career Readiness aggregation as a function of learner, career dispositions and semester. The measure of career readiness of the learner Ghanim, in OC (openness to challenge), in Spring 2014, is 80. Thus, dimensions offer concise and intuitive way of organizing and selecting data for retrieval, exploration and analysis.

Multidimensionality is close to the way humans think and so it provides a friendly, easy to understand and intuitive visualization of data for analysts as well as non-expert end-users. Developing a DWS requires a conceptual modeling of multidimensional data in a way that reflects real-world situations [59]. This in turn requires placing a special attention on defining relationships between facts and dimensions; and between various levels of aggregation in a dimension hierarchy [60]. The importance of conceptual modeling of data as
A major step of DWS development is widely recognized to be the necessity foundation for building well-documented and user requirements-driven database [61, 62, 63]. Conceptual multidimensional modeling provides a high level of abstraction to describe the warehousing process and architecture, independently from implementation issues. It uses graphical notions to facilitate writing, understanding and managing the conceptual representation of data or data schema. The existing approaches of multidimensional modeling in literature can be framed into three categories [59, 57, 61]: (1) extensions to Unified Modeling Language (e.g., [64, 65]); (2) extension to Entity-Relationship (e.g., [66, 67]); and (3) ad-hoc models (e.g., [68, 69]). The difference between these categories lie in the possibility of representing more advanced concepts such as many-to-many associations and irregular hierarchies. As a good DWS should support user-definable multiple hierarchies among arbitrary dimension [70], extended ER (EER) models emerge as the most cost effective design that supports both multidimensional aggregations and multiply hierarchies. In next section, we present a review of the EER approaches for conceptual multidimensional data modeling.

Extended Entity-Relationship (EER) Models

The entity-relationship (ER) model has been introduced by P. Chen in 1976 [71] to conceptualize and graphically represents multilevel views of data for description and manipulation purposes. A large number of extensions to this model has been proposed to support the
conceptualization of multidimensionality and higher order of hierarchical relationships as in the group of Extended Entity-Relationship (EER) models. ERR model is a conceptual data model that is capable of describing the data requirements or conceptual schema for a DWS in a direct and easy way to understand graphical notations. A typical EER model [72] uses inductive developments of data structuring. Basic attributes are assigned to base data types. Complex attributes can be constructed by applying constructors (i.e. tuple, list or set of constructors) to basic attributes. Entity types conceptualize structuring of objects through a set of attributes. Relationship types link entity types that have already been constructed into an association type. Cluster types allow to generalize or to combine types into singleton types. The types may be restricted by integrity constraints and by specification of identification of objects defined on the corresponding type. Typical integrity constraint of the EER model are participation, lookup and general cardinality constraints. Entity, cluster and relationship classes contain a finite set of objects defined on these types. The types of an EER schema are typically represented by an ER diagram.

Several works in the literature extended the standard ER model with constructs that allow the modeling of multidimensional data and the structure of aggregation. For example, Data Warehouse Conceptual Data Model (DWCDM) [73] introduces complex descriptions of the structure of aggregated entities and multiple hierarchically organized dimensions based on Description Logics DL. DL is a class of formalisms for which it is possible to study the expressivity in relation with reasoning problems. The model translates ER schema into a corresponding DL by extending the semantics of DL to describe the components of aggregations along the relationships between properties of the components and that of the aggregation itself. Multidimensional Entity Relationship (ME/R) model [66] is also a specialization of the ER model that allows the separation of qualifying and quantifying data; and supports complex structure of dimensions. The authors in [66] describe two special relationship sets connecting dimension levels: fact and rolls up. The fact relationship set is a specialization of a general n-array relationship set. It connects n different dimension level entities to represent a fact of dimensionality n. A description of the fact is used as the name
for the set. The directly connected dimension levels are called atomic dimension levels. The fact relationship set models the inherent separation of qualifying and quantifying data. The attributes of the fact relationship set model the measures of the fact (quantifying data) while dimension levels model the qualifying data. The rolls-up relationship is a binary relationship set to model the structure of qualifying data. It relates a dimension level A to a dimension level B representing concepts of a higher level of abstraction (e.g., city rolls-up to country).

StarER model [67] combines the constructs of ER model with the star schema, which is dominant in data warehouses. StarER model has the following constructs: (1) fact set that represents a set of real-world facts sharing the same properties. A fact set is represented as a circle that is always associated to time (i.e., data is generated in terms of facts, each time an event related to the fact takes place); (2) entity set: represents a set of real-world objects with similar properties; it has the same meaning as in traditional application modeling; and (3) relationship set: represents a set of associations among entity sets or among entity sets and fact sets. The model then represents facts and their properties, connects the temporal dimensions to the fact, represent objects and their properties and associations, records associations between objects and facts, and distinguishes dimensions and categorizes them into hierarchies.

MultiDimER [74] is another extension of ER model that comprises a finite set of dimensions and fact relationships to allow several kinds of hierarchies (i.e., simple, multiple, parallel, independent, dependent, strict or non-strict). MultiDimER defines a dimension as an abstract concept for grouping data that shares a common semantic meaning within the domain being modeled. Each dimension is either a level, or one or more hierarchies. Every instance of a level is called a member. A hierarchy contains several related levels (the last level being a leaf) that are used for roll-up and drill-down operations. The relationships joining levels in hierarchy are characterized by the cardinalities and the analysis criterion. Cardinalities indicate the minimum and the maximum numbers of members in one level that can be related to a member in another level. The analysis criterion expresses different
structures used for analysis. The binary relationship linking the levels of a hierarchy is only used for traversing from one level to the next one; while a fact relationship represents an n-array relationship between leaf levels. The MultiDimER model allows to represent shared levels as well as shared dimensions. Sharing levels allows to reuse of existing data; while sharing dimensions opens the possibility to analyze measures presented in different fact relationships.

CGMD [75] combines the semantics of ER model with the GMD logic-based formalism proposed in [76]; and extends the idea of (DWCDM) in [73]. The definition of a GMD schema is given as follows: consider the signature < F,D,L,M,V,A >, where F is a finite set of fact names, D is a finite set of dimension names, L is a finite set of level names - each one associated to a finite set of level element names, M is a finite set of measure names, V is a finite set of domain names - each one associated to a finite set of values, A is a finite set of level attributes. CGMD utilizes GMD to represent database schema in an ER diagram and describe multidimensional structure including dimensions with their hierarchically organized levels and the structure of aggregations. It extends standard ER schema with constructs of aggregated entities together with their interrelationships and other parts of the schema. It presents simple aggregated entities and multidimensional aggregated entities. An n-dimensional aggregation consists of n dimensions and their associated levels along with a fact involved in the aggregation. The model also allows the computation of aggregations from pre-computed aggregations.

For evaluating EER models discussed above, we considered some requirements criteria from the literature that are relevant to modeling multidimensional data warehouse [77, 78]. These requirements are: (1) implementation independent; (2) explicit separation of structure and contents; (3) explicit hierarchy in the dimension; (4) multiple hierarchies in dimension; (5) dimension/level attributes (i.e. specify the attributes that do not define hierarchies); (6) support for aggregation; (7) handling complex measures; (8) handling different levels of granularity; (9) support for non-strict hierarchies (i.e. allows mapping between different levels of hierarchies); (10) support for non-onto hierarchies (i.e. for each
### Table 3.1: Requirements Criteria of Conceptual Multidimensional Data Models

<table>
<thead>
<tr>
<th>Model</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>DWCDM</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>ME/R</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>p</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>StarER</td>
<td>x</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>p</td>
<td>√</td>
<td>x</td>
<td>x</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>p</td>
<td>x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MultiDimER</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
</tr>
<tr>
<td>CGMD</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
</tr>
</tbody>
</table>

Based on these criteria, a comparison of conceptual multidimensional models reviewed in this chapter is presented in Table 3.1. The rows correspond to each model, and columns correspond to the requirements criteria introduced above; while cells indicate if the model supports given criteria (denoted by “√”) or partially supports (denoted by “p”) or doesn’t support it at all (denoted by “x”). It is shown in Table 3.1 that the CGMD model fulfills all key requirements for multidimensional data warehouse construction and aggregations; and appears syntactically and semantically richer than the other models. Accordingly, we adopt CGMD to design and model our career multidimensional data as will be presented later in this chapter.

### 3.2.2 Data Analytics and Visualization

Data analytics refers to the “use of data, statistical analysis, and explanatory and predictive models to gain insights and act on complex issues” [79]. By this definition, data analytics itself is not the goal but a tool used to address strategic questions and to support decision making. Thus, data analytics process is initiated by a strategic question, followed by steps to:

1. Find and collect the most appropriate data to answer that question. Collected data must be filtered for relevance and stored in a form that is useful.
2. Analyze the data. Data analysis includes connecting, linking, clustering, classifying, associating, and correlating different data sets to be able to grasp the information that is supposed to be conveyed by these data.

3. Visualize data where findings are represented in understandable and actionable manners; to guide decision-making processes.

4. Provide feedback and integrate findings into the existing processes of addressing the strategic problems.

As data is collected in the data warehouse and analyzed by different functional modules, the data visualization step that comes next is a graphical presentation of the data to provide a qualitative understanding of the information contents and trends in a natural and direct way [63]. Having multidimensional data, data visualization aims at representing each of the features $x_1, x_2, ..., x_n$ characterizing the object $X_i = (x_{i1}, x_{i2}, ..., x_{in}), i \in 1, ..., m$, in a visual form acceptable for a human being to make visual insights into the data set [55]. Effective graphical representations of the data help users to detect and explore the expected patterns, as well as discovering the unexpected ones. The direct visualization techniques of multidimensional data could be classified into three broad categories according to the overall approaches taken to generate resulting visualizations [55, 80]: geometric methods; iconography displays; and hierarchical display.

a) Geometric Methods

Geometric visualization methods display multidimensional points using the axes of the selected geometric shape [81]. Scatter plots are one of the most commonly used techniques for data representation on a plane $R^2$ or space $R^3$. Points are displayed in the classic $(x, y)$ or $(x, y, z)$ format. Usually, the two-dimensional ($n = 2$) or three-dimensional ($n = 3$) points are represented by this technique. The scatter plots can be also applied to visualize more higher dimensionality data using a matrix of scatter plots. The matrix of scatter plots is an array of scatter plots displaying all possible pairwise combinations of features. It is then
useful for observing all possible pairwise interactions between features [82]. The matrix of scatter plots of the IRIS data set \(^2\) is presented in Fig. 3.3. IRIS data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant (Setosa, Versicolour, Virginica). The latter two classes are somewhat overlapped. It has 4 features (sepal length, sepal width, petal length, and petal width). We can see in Fig. 3.3 that Setosa flowers (blue) are significantly different from Versicolor (red) and Virginica (green).

![Figure 3.3: Scatter Plot Matrix of the IRIS Data](image)

**b) Iconography displays**

Iconography displays, also called glyph methods, aim to help understanding multidimensional data by displaying each object of \(n\) features as a glyph [83]. Color, shape, and location of the glyph depend on the values of features. The most common glyphs used for data visualization are Chernoff faces (data features are mapped to facial features) [84]; and stars [85]. Each object is displayed by a one stylized star. In the star plot, the features are

\(^2\)https://archive.ics.uci.edu/ml/datasets/Iris
represented as equal angular axes radiating from the center of a circle, with an outer line connecting the data value points on each axis. The IRIS data plotted by star glyphs are presented in Fig.3.4 where the stars corresponding to Setosa irises (a) are smaller than the two other species while the larger stars correspond to Virginica irises (c). A variety of examples of more glyphs have been proposed and used in the literature [83].

![Figure 3.4: IRIS Data Set Visualized by Star Glyphs](image)

c) **Hierarchical displays**

Hierarchical displays create a structure of an image such that some features are embedded in displays of other features [80, 55]. Visualization of some features is displayed in the structure depending on the values of other features. The most famous methods are: dimensional stacking [86] and trellis display [87]. Dimensional stacking, also called general logic
diagrams, partitions the data space into 2-dimensional subspaces which are stacked into each other. The IRIS data, visualized by the dimensional stacking method, are presented in Fig.3.5. Notice that Setosa irises (black cells) are displayed separately from the other two species which overlap. A major advantage of dimensional stacking method is that no aggregation function is needed to plot the data. The trellis display method is similar to the dimensional stacking within which two features are selected at first to be axis variables. Other features are called conditioning variables. The ranges of the values of these features are divided into non-overlapping subranges; then the panel plots are drawn for each pair of subranges. The panel plots can be a bar, a scatter plot, etc.

Figure 3.5: Iris Data Visualized by Dimensional Stacking
3.2.3 Data in Higher Education Environments

Data used now in higher education are becoming much more extensive, intensive and automatic [88, 89]. Extensive data is a wide but shallow data that is collected from large number of participants on limited dimensions. Intensive data, on the other hand, is a detailed multidimensional observations from relatively a small number of participants. Both types of these data in educational contexts can be extracted automatically from different resources and transformed to support informed instructional practices and to improve learning processes. In higher education in particular, the adoption rate for data warehousing has been rising since 1990s [54, 90, 91, 92]. With the evolving world of Big Data and analytics, warehouse systems become more valuable for higher education institutions. In the following subsections, we present current data sources fueling higher education environments; and discuss gaps and future trends in leveraging data use to respond to some of the growing challenges related to education quality, behavioral and attitudinal outcomes, and relevance to the market.

Data Sources

Data environments in higher education can be classified into seven main categories: student information systems (SIS), traditional learning management systems (LMS) and Grade Performance System (GPS), web-based courses, adaptive intelligent educational systems (including intelligent tutoring systems (ITS) and adaptive hypermedia systems (AHS), personal learning environments (PLE), Progress and Course Engagement (PACE), social media and open data sets.

While learners and instructors are using these different educational platforms, they generate various datasets that include but are not limited to: (1) personal data (i.e. demographics, disposition and preferences indicators), (2) academic data (i.e. grades, certifications); (3) interaction data (i.e., number of discussion posts, produced or read); and (4) traces (i.e. number of times and types of educational resources accessed, social network
activities, etc.). In different classification, the type of generated data can be categorized as:

(1) student performance data (i.e. grades and achievement levels); (2) learning outcomes data (i.e., generic skills, critical areas of low performance); (3) institution level data (i.e. program level reports, strategic planning), and (4) context based data (i.e. demographic and socio-economic factor linked to academic performance).

Emergence of Big Data in Higher Education

New computer-supported interactive learning methods and tools—such as intelligent tutoring systems, simulations, games—have opened up opportunities to collect a massive quantity, range and scale of data about learners and their activities. This explosion of data brings new opportunities and challenges for higher education institutions to utilize the vast array of data that are poised to ultimately shape the future of higher education [93]. Wagner and Ice [94], noted that applications of Big Data can serve as catalysts for the move towards advanced growth of analytics in higher education. Thus, it has the potential to influence the practice of higher education environments, from enhancing students learning experience and improved academic outcomes, to more effective evidence-based decision making and strategic response to the changing global trends. Fig. 3.6 depicts the key opportunities Big Data brings to administrators, students and lecturers as key players in higher education.

As information technology continues to penetrate all facets of higher education, there is a growing number of systems and tools that leverage the generation of Big Data. This trend is favored by the combinations of online and virtual resources such as social networking tools, blogs, podcasts, online videos, and instant messaging, and more traces of online activities.
Gaps and New Trends

Despite the expanded presence of information technology within higher education, there was no significant difference between using computing solutions to generate information and using them to construct knowledge in higher education [79, 48]. In contrast, there is a deep disconnect between adopting new technologies and truly leveraging data to enhance quality, especially in terms of teaching and learning [93]. The Information Age arguably requires higher education institutions to focus less on the basic disciplines and offer more on professional programs to deliver graduates with generalist knowledge and advanced social skills who are adaptable, responsible, life-long learners and creative; and who are also able to engage in effective and complex communication with others [10]. This has placed
an increasing emphasis on the outcomes of higher education and the evaluation to these outcomes using new data sets and indicators to demonstrate that learners have actually mastered specific knowledge and skills objectives as a result of their learning. Example of the mostly required outcome indicators focus on interactions between learners and professors, behavioral skills, career expectations, graduation and success in finding a job [95, 96, 97].

Further, the areas of concerns in higher education nowadays lie in three main domains:

1. Shaping the knowledge society by delivering graduates who can make consequential decisions in complex and risk-fraught environments, and who should be able to boil down complexity in a way for which simple technical “know-how” alone would not be sufficient

2. Employability or market relevance to ensure a stronger link between higher education and practice, since higher education programs which are merely based on tacit or technical contents are no longer considered adequate to meet the needs of professional practice

3. Lifelong learning capacity to allow learners to acquire further qualifications or skills independently throughout their career path

Several educational research studies and reports (e.g., NMC³, EDUCAUSE⁴) identify the emerging educational data driven information technology trends and solutions to address the above concerns and enhance higher education outcomes. Table 3.2 depicts the trends of using information technology in higher education, as a result from the reports published over the last five years, between 2011-2015 The emerging technologies are classified according to the adoption time in three categories: one year or less, two to three years and four to five years. Next we discuss the top three trends (underlined in Table 3.2); and which are directly related to our research scope. A comprehensive discussion of new technologies for higher education summarized in Table 3.2 can be found in [98]

³http://www.nmc.org
⁴http://www.educause.edu
<table>
<thead>
<tr>
<th>Year</th>
<th>One Year or Less</th>
<th>Two to Three Years</th>
<th>Four to Five Years</th>
</tr>
</thead>
<tbody>
<tr>
<td>2011</td>
<td>Mobile Computing: Open Content</td>
<td>Electronic Books; Simple Augmented Reality</td>
<td>Gesture-Based Computing; Visual Data Analysis</td>
</tr>
<tr>
<td>2012</td>
<td>Mobile Applications; Tablet Computing</td>
<td>Gesture-Based Computing; Learning Analytics</td>
<td>Gesture-Based Computing; Internet of Things</td>
</tr>
<tr>
<td>2013</td>
<td>Massively Open Online Courses; Tablet Computing</td>
<td>Games and Gamification; Learning Analytics</td>
<td>3D Printing; Wearable Technology</td>
</tr>
<tr>
<td>2014</td>
<td>Flipped Classroom; Learning Analytics</td>
<td>3D Printing; Games and Gamification</td>
<td>Quantified Self; Virtual Assistants</td>
</tr>
<tr>
<td>2015</td>
<td>Bring Your Own Device; Blending learning; Learning Analytics; Flipped Classroom; Mobile Applications</td>
<td>Collaborative Environments; Makerspaces; Wearable Technology; Games and Gamification</td>
<td>Adaptive Learning Technologies; The Internet of Things; Wireless Power; Flexible Displays</td>
</tr>
</tbody>
</table>

Table 3.2: Emerging Technologies in Education

a) Learning Analytics and Smart Universities

As higher education institutions adopt blended technology-based teaching approaches, learning occurs more frequently within online environments. The educational data mining methods have already explored ways to track learners' activities and behaviors and to add these data to the ever-growing warehouses of learner-related information. Learning Analytics (LA) aims to access and understand these data and adds a new dimension to the learning process. In general, LA synthesizes several existing techniques such as information retrieval, machine learning and statistical algorithms to achieve [99, 48]:

1. Descriptive Analytics aims at exploring and describing data and discovering hidden patterns and trends
2. Predictive Analytics aims at evaluating the likelihood of future events and identifying any risk or opportunity by looking into current trends and recognizing associations about related future issues in academic programs, teaching and learning settings.

3. Prescriptive analytics aims at assessing current situations and enabling informed decision-making for future actions by combining analytical outcomes from descriptive and predictive models.

LA has increased in importance over the past two years and is expected to grow further over the next years as evidenced by the growing volume of research work as well as projects carried out in this area [100, 101]. Predictive analytic tools, particularly become of more significance as majority of LA studies investigate issues related to learners’ behavior modeling and prediction of performance in formal and informal learning environments. This is driven by the need to support higher education institutions to improve their learning outcomes in terms of “smart competencies” required for a successful career path in 21st Century [97, 102]. Accordingly, it is critical for higher education institutions to evolve into “smart universities” delivering graduates with a bouquet of skills required by the market [103]. LA has the potential to support education institutions to provide a seamless integration between education and industry through constructing and enriching professional profiles and, consequently, curricula and courses which contents are fresh and relevant that adhere to the skills sets demanded by the industry and professional communities.

b) Blending formal and informal learning

Blended learning, also referred to as hybrid learning, (see Fig.3.7) is a pedagogical approach that combines learning models of face-to-face instructions and online activities leveraging the strength of each to provide an optimal environment for effective learning processes [104, 105]. The fact is that virtual learning environments are perceived to offer more learning affordances than physical classrooms, including opportunities for extended collaborative learning; and thus “education paradigms are shifting to include more online learning,
blended and hybrid learning, and collaborative models"[106]. The introduction of more online learning platforms through courses make dynamic, flexible and accessible contents. When blended learning programs are implemented effectively, they can make better use of facilities and instructional resources, and increase instruction offerings and speeding up the graduation pathway for students in higher education [107]. A study at George Mason University (GMU) [108] showed that students who collaborated with others outside of the classroom for online components of a management course reported better outcomes. In practice, there is a number of innovative systems of online learning programs used by higher education institutions, some of which specialize in helping learners acquiring in-demand skill sets. Yammer, for example, is a platform used in the educational industry -mostly by graduating learners- as it provides flexible environment for supporting blended learning in light of social presence and organizational cultures [109, 110].

![Blended Learning](image)

**Figure 3.7: Blended Learning**

c) Collaborative models and enabled technologies

A strong trend is the devolvement of learning models that supports collaboration and interaction among learners encourage creative and risk-taking activities that more accurately reflect the contemporary workplace. This drive the arise of more innovative learning platforms that emphasize human interactions and multidimensional learning through cultivating 21st century skills such as intercultural communication and social entrepreneurship. It also results in the emergence of new competency-based education, which tracks student skills
instead of credit hours. Collaborative learning is “based on the idea that learning is a naturally social act” [111]. The social based learning processes are enabled in higher education by contemporary social media platforms brought by the evolution of Web 2.0 tools that have been increasingly adopted for creating and sharing contents, as well as for providing means for communication and collaboration. Collaborative open learning environments (see Fig.3.8) fulfill certain requirements in terms of pedagogical, social and technological aspects. In social dimension, the learners’ interactions with external learners and experts in different social media platforms are highly emphasized as they bring new insights into tacit knowledge content, and optimize the understanding of its applications. This will require learners to build a public profile for better personalization of the learning process; and for assuring a continuity of the learning community through establishing and maintaining collaborative learning ties. There are several artificial intelligence based techniques and tools that have been developed to support and guide collaborative learning [112, 113]. However, Facebook at Work⁵ is one of the very new projects in this area exhibiting a potential solution for creating learning communities on a popular social network.

In our research application domain, we propose computing solutions within these trends to implement a comprehensive career readiness system into higher education programs. As presented in Fig. , the proposed solutions aim to implement three main processes: career readiness, career prediction, and career development. Career readiness process involves defining career dispositions dimensions and creating means to measure and analyze them. We introduce novel career dispositions scale within higher education environments; and develop diagnostic and visualization tool to track their maturation across mentoring workflows. Next , we present our work to implement our proposed Career Readiness module.

⁵https://www.facebook.com/help/work
3.3 Cognitive Data Metrics

3.3.1 Multidimensional Data Model

Our multidimensional data model captures cognitive dispositions from the Big Five model and lifelong learning dimensions. Our proposed model identifies five categories of personal traits that influence career success; while lifelong learning dispositions make up the individuals' capacity for developing lifelong learning attributes. Career dispositions emerge as the joint set of attitudes and generic skills that dispose individuals to engage profitably with learning from new professional environment in order to be able to adapt to career changes and to manage their career growth. In this essence, we define career dispositions as: "A scheme of attitudes, assumptions and skills that engenders professional behaviors; and influences the ability to adapt and respond to changing work situations and environments."

Further, we combined lifelong learning [53] and Big Five [114] frameworks to model career disposition as a 6-dimensional construct that comprises: Openness to challenge (OC),
Critical Thinking (CT), Resilience (R), Learning Relationships (LR), Responsibility for Learning (RL), and Creativity (C) (Fig.3.9). The six dimensions of career dispositions describe the natural tendencies, mind state and preparations of each individual towards a professional practice. We later prove statistically the validity of this model in terms of its validity to measure what is intended for. A brief definition of each career dispositions dimension is discussed next, while Table 3.3 summarizes the individual characteristics corresponding to high and low scores along each dimension.

**Openness to Challenge (OC)** refers to the degree to which an individual is intellectual, creative, curious and open to new ideas and experiences. Individuals with high openness degree thrive in situations that require flexibility and learning new skills, which make them highly adaptable to change. They also tend to seek feedback on their performances; and to build new relationships.

**Critical Thinking (CT)** refers to the degree to which an individual is investigative, attentive reader/listener, inquisitive, analytical and an evidence-based decision-maker. Critical thinkers strive for understanding, keep curiosity alive, remain patient with complexity, and are ready to invest time to overcome confusion. They tend to develop their own ideas about any topic, however, they show interest in other people ideas even if they disagree on the principle, in order to practice fair-mindedness and avoid extreme views.

**Resilience (R)** refers to the degree to which an individual is conscientious, determined, assertive and achievement-oriented. Individuals with high resilience degree have
good communication and social skills, thrive in social contexts and generally make a positive impression of themselves. They also show high levels of drive and energy that make them able to work under conditions of uncertainty, deal with the unexpected, and solve problems as they arise.

**Learning Relationships (LR)** refers to the degree to which an individual is cooperative, expressive, agreeable and social oriented. Individuals with high LR degree tend to establish social networks and use them effectively to collect information and seek feedback. They enjoy learning with and from others.

**Responsibility for Learning (RL)** refers to the degree to which an individual is dependable, autonomous, motivated, organized and punctual. Responsible learners tend to take part on deciding what will be learned and how. They are able to identify their strengths and weaknesses, develop strategies for learning, manage time and available resources, monitor their progress and make changes when existing learning strategies are not working or when they are challenged. They always try to test their new learning approaches in real-life applications.

**Creativity (C)** refers to the degree, to which an individual is intellectual, imaginative, adventurous, curious and original. Creative individuals tend to regard problems and controversial issues as exciting challenges. They accept new ideas, focus on details, and ask questions rather than only accepting what they are told. They also tend to take risks and try new things regardless of rules and regulations.

### 3.3.2 Data Measurement Instrument

In order to measure career disposition data values, we developed a scale that is conceptually underpinned by constructs from career success and lifelong learning literature following our blended approach to career disposition. Based on this scale, data is routinely captured through the proposed instrument to elicit quantitative reflections of career disposition via an enterprise-wide electronic platform as we will present later in this chapter. We validate this instrument using known statistical methods to ensure that it measures what it is intended
<table>
<thead>
<tr>
<th>Dimension</th>
<th>High scores are</th>
<th>Low scores are</th>
</tr>
</thead>
<tbody>
<tr>
<td>Openness to Challenge (OC)</td>
<td>Intellectual, creative, curious, open to new ideas and experiences</td>
<td>Skeptical, conventional, practical</td>
</tr>
<tr>
<td>Critical Thinking (CT)</td>
<td>Investigative, attentive reader/listeners, inquisitive, analytical, experimenting, evidence-based decision makers</td>
<td>Inpatient with complexity, confused, impression-based decision makers, impulsive</td>
</tr>
<tr>
<td>Resilience (R)</td>
<td>Determined, assertive, energetic, social, competitive, achievement oriented</td>
<td>Passive, inconsistent, droopy</td>
</tr>
<tr>
<td>Learning Relationships (LR)</td>
<td>Cooperative, expressive, agreeably, social oriented</td>
<td>Quite, uncooperative, distant, introvert</td>
</tr>
<tr>
<td>Responsibility for Learning (RL)</td>
<td>Dependable, autonomous, motivated, organized, dedicate, punctual</td>
<td>Dependent, unreliable, careless, feeble</td>
</tr>
<tr>
<td>Creativity (C)</td>
<td>Intellectual, imaginative, adventuresome, curious and original</td>
<td>Mechanical, unoriginal, spontaneous</td>
</tr>
</tbody>
</table>

Table 3.3: Career Dispositions Scores

for.

**Development of the instrument**

Self-Reflective Career Dispositions Scale (SRCDS) metric is a self-report instrument that asks respondents to describe their attitudes and behaviors in terms of the identified six dimensions of career dispositions. The scale is presented as 35 items in a questionnaire against which respondents are asked to rate each item, as it relates to them using a five point Likert scale from 1 (Not at all like me) to 5 (very like me). Some items were negatively worded - marked as “reversed”- in order to help preventing biased responses. Those items have to be reversed before an individual’s score can be computed. If an item has to be reversed, a respondent who has circled 1 for that item now receives a score of 5 for the positively worded version of that item. A higher total dimension score indicates a higher level of readiness for a career path along the given dimension.
In developing the questionnaire items for our instrument, we considered research results from educational psychology domains to derive the most widely used and validated scales [115, 116], based on which we drew relevant items that measure the psychological traits encompassing the six proposed dimensions earlier. We augmented few other items based on the combination of pre-established learning dispositions and personality factors criteria. Next, we discuss the validation of this proposed model.

**Instrument validation**

The SRCDS was validated using Exploratory Factor Analysis (EFA) to identify the underlying relationships between the measured items of the instrument and their contribution to the measured goal [117]. EFA is a well-known multivariate statistical technique used to identify common clusters (called factors) of inter-correlated items. This is to confirm that the questionnaire items do measure the intended dimensions of the survey. This validation methodology is commonly used in psychometrics instrument development, in order to simplify the structure used for data collection by exploring and summarizing underlying correlations between factors. This approach has three basic decision points: (1) decide the number of factors, (2) choose an extraction method, and (3) choose a rotation method (to check whether items relate to more than one factor). Thus, EFA is a four steps approach as illustrated in Fig. 3.10:

1. Check data set to decide if it is suitable for EFA through examining the sample size, outliers and factorability of the correlation matrix.

2. Decide the factor extraction method. This involves deciding number of factors.

3. Simplify the data structure to learn to decide variables-factor loadings. This involves selecting the rotation method maximize the high item loadings and minimize low item loadings.

4. Interpret and label
1) Sample adequacy

This step checks whether our data set is suitable for EFA through examining:

- **Sample size**: the literature suggests that sample sizes should be 100 or greater. Our survey was administered in hard copy to groups of University students (individually and during class time). The participants were senior students in an Information Technology undergraduate program (58 female and 42 male). The completed survey responses \((Q = 100)\) for set of 35 variables \((V_1, \ldots, V_k; k = 35)\) were then used to demonstrate the factorial validity of SRCRS.

- **Outliers**: few outliers were identified (3 values). However, EFA is sensitive for outlying cases and so the mean was substituted for the extreme value.

- **Factorability of the correlation matrix**: the correlation matrix that is used in the EFA process displays the relationships between individual variables \((v_1, \ldots, v_k)\). Factorability of the correlation matrix is checked by inspecting the matrix for correlation coefficients over 0.3 to produce factor matrix \((F \ast V)\) (see Fig.3.11). If no correla-
tions exceed 0.3 then factor analysis is most properly not the appropriate statistical method to utilize. In other words, if the factors account for approximately 30% relationship within the data sample, it would indicate that a third of the variables have too much common variance, and hence it is not practical to determine the level of correlation between variables. The factorability test of our data sample indicates that the correlation matrix between the variables of SRCDS data has correlations that are over 0.3.

![Data Matrix](image)

Figure 3.11: Find Factorability of the Correlation matrix

2) Factors Extraction

- **Extraction method**: the most common used method to extract factors is the Principal Components Analysis (PCA). This method produces loading matrix \(L\) and Eigen Values \(EVs\). Loading matrix is a matrix of coefficients or weights (Factor loadings FLs) for a set of linear equations relating observed variables (or items) to factors (or dimensions). The rows of the matrix correspond to the observed variables and the columns correspond to the factors while FLs indicate relative importance of each item to each factor. \(EVs\) represent the variance accounted for by each underlying factor. \(EVs\) are represented by percentages which cumulated value is the number of items. Each factor has \(EV\) that indicates the amount of variance each other factor accounts for. A good factor solution is the one that explains the most variance with fewest factors.
<table>
<thead>
<tr>
<th>Eigen Values</th>
<th>% of Variance</th>
<th>Cumulative %</th>
</tr>
</thead>
<tbody>
<tr>
<td>20.71</td>
<td>54.5</td>
<td>54.5</td>
</tr>
<tr>
<td>2.86</td>
<td>7.5</td>
<td>62.0</td>
</tr>
<tr>
<td>2.67</td>
<td>7.2</td>
<td>69.0</td>
</tr>
<tr>
<td>1.72</td>
<td>5.7</td>
<td>74.8</td>
</tr>
<tr>
<td>1.119</td>
<td>4.5</td>
<td>82.2</td>
</tr>
</tbody>
</table>

Table 3.4: Total Variance Explained by Identified Factors

- **Number of factors**: the most common approach to decide the number of factors is the Kaiser-Guttman rule which simply states that the number of factors are equal to the number of factors with $EVs > 1.0$. Table 3.4 and Fig 3.12a. demonstrate the results obtained by applying PCA on SRCDS data. As shown, a cumulative percentage of variance of 82.2% is explained by a total of 6 components (factors) which have an $EV > 1$. The first factor explains the most variance and the last factor explains the least variance. Scree test or generating a scree plot is another approach for deciding the number of factors. The scree plot is a two dimensional graph with factors on the x-axis and $EVs$ on the y-axis. $EVs$ are typically arranged in a scree plot in descending order (shown in Fig.3.12b ). As Fig. 3.12b indicates, the first six factors account for most of the variance, while the remaining factors all have small $EVs$ ranging from 0.9 to 0.0. However, until FLs are rotated (in Step 3 of the validation process), it is difficult to interpret and extract factors. A rotation of the $L$ matrix helps to find more interpretable FLs as well as the factors they represent.
3) Rotation method

Rotation produces a simple data structure by maximizing the high item loadings and minimizing low item loadings. This consideration decides how many factors to analyze based on whether a variable might relate to more than one factor, to simplify the data structure used by the instrument. A simple data structure is defined as "a condition in which variables
near 1 are clearly important in the interpretation of the factor, and variables that load near 0 are clearly unimportant. A simple structure is derived to simplify the task of interpreting the factors. There are five criteria that need to be met to achieve a simple structure:

- Each variable should produce at least one zero loading on some factor. Zero loadings include any that fall between -0.10 and +0.10.
- Each factor should have at least as many zero loadings as there are factors.
- Each pair of factors should have variables with significant loadings (loadings of +.30 or higher) on one and zero loadings on the other.
- Each pair of factors should have a large proportion of zero loadings on both factors (if there are say four or more factors total).
- Each pair of factors should have only a few complex variables, which are with loadings of +.30 or higher on more than one factor.

There are two common rotation techniques used in the literature: orthogonal rotation (varimax/quartimax) and oblique rotation (oblimin/promax) [118]. Orthogonal rotation methods assume that the factors in the analysis are uncorrelated while oblique rotation methods assume that they are correlated. The literature suggests that whichever rotated method produces the best fit and factorial suitability, both intuitively and conceptually, should be used. EFA results for our proposed SRCDs' data samples were rotated using two orthogonal methods and one oblique method. The best fit was produced by the Promax, which is a common oblique method (see Fig. 3.13). Promax operates basically on the output of Varimax, the orthogonal rotation method. First, Varimax takes the original loading matrix, $L$, and multiplies it by a transformation matrix $\Delta$, to obtain the Varimax transformed matrix $\Delta^T V L$. The transformation matrix takes the form: The angle, $\alpha$, is determined so as to maximize the Varimax measure. Next, Promax transforms $V L$ by raising loadings to powers, generally 2, 4, or 6. This drives down the values of all LFs, with the smallest values from Varimax becoming much smaller, while larger LFs are not reduced as much. Consequently, the result in Promax solution is a set of LFs that generally reflects a simple structure better.
than the set from the Varimax solution, particularly when the underlined factors are highly correlated. The results revealed in Table 2 show that factors 1, 2, 3 and 4 are all interrelated as well as factors 2 and 5.

![Table 2: Factor Correlation Matrix](image)

Figure 3.13: Example of Oblique (Promax) Rotation of SRCDS Sample Data

4) Interpretation and labeling

After rotation, lines of best fit (vectors) were re-arranged as seen in Fig.3.13. To optimally go through clusters of variables in order to make it easier to interpret LF1s and factors they represent. However, due to the low commonalities, some items or variables didn’t load highly on any factor. Low communality of a variable indicates that a considerable proportion of this variable’s variance is unexplained by the extracted factor. In the case of low commonalities, more factors have to be extracted in order to explain the variance or these variables should be removed from the EFA. We removed unexplained items and ended up with 6 factors that explain the variance of 22 items/variables. Each of the extracted factors shows 3 or more strong loadings. The analysis also indicates 8 complex variables. This result is expected as the underlying factors are theoretically correlated. Factor 1 was labeled OC as it reflects elements of openness to new learning experiences. Factor 2 relates more to practices that reflect questioning things and developing new ideas. Consequently, we have labeled it CT. Factor 3 highlights elements that define determination, flexibility, resource management and attitude to hard working and so it was labeled R. Factor 4 consists of items
related to the ability of establishing learning relationships and work with other learners and so it was labeled LR. Factor 5 reflects all core components of developing decision-making ability, responsibility of learning and autonomous learning. Thus, Factor 5 was labeled RL. Finally, Factor 6 relates to creativity reflecting practices of accepting new ideas, focusing on details, and asking questions. Consequently, Factor 6 was labeled C. Fig. 3.14 demonstrates the factors after labeling and the items selection.

<table>
<thead>
<tr>
<th>Item</th>
<th>Factor 1</th>
<th>Factor 2</th>
<th>Factor 3</th>
<th>Factor 4</th>
<th>Factor 5</th>
<th>Factor 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>I try to understand contents as thoroughly as possible</td>
<td>-0.2037</td>
<td>-0.0613</td>
<td>0.0063</td>
<td>-0.0064</td>
<td>0.4935</td>
<td>0.2725</td>
</tr>
<tr>
<td>I know what I want to learn even if I cannot guarantee a good grade</td>
<td>0.2323</td>
<td>-0.0267</td>
<td>0.2199</td>
<td>-0.009</td>
<td>0.3054</td>
<td>0.0094</td>
</tr>
<tr>
<td>I take responsibility for my learning experiences</td>
<td>0.2514</td>
<td>0.0349</td>
<td>0.0063</td>
<td>0.0155</td>
<td>0.2495</td>
<td>-0.0161</td>
</tr>
<tr>
<td>I am good at meeting deadlines</td>
<td>-0.2238</td>
<td>-0.5457</td>
<td>0.8785</td>
<td>0.1396</td>
<td>0.4080</td>
<td>-0.2854</td>
</tr>
<tr>
<td>In a learning experience, I prefer to take part in deciding what will be learned and how</td>
<td>0.2968</td>
<td>0.0018</td>
<td>-0.0549</td>
<td>-0.0577</td>
<td>-0.5437</td>
<td>0.2781</td>
</tr>
<tr>
<td>I rarely think about my own learning and how to improve it. REVERSED</td>
<td>-0.0114</td>
<td>0.3443</td>
<td>0.1099</td>
<td>0.3271</td>
<td>0.1153</td>
<td>0.1518</td>
</tr>
<tr>
<td>I like new learning experiences</td>
<td>-0.4228</td>
<td>-0.3656</td>
<td>0.1348</td>
<td>0.4544</td>
<td>0.1466</td>
<td>0.1262</td>
</tr>
<tr>
<td>I prefer course material that stimulates my curiosity, even if it is difficult to learn.</td>
<td>0.7794</td>
<td>-0.7906</td>
<td>-0.4012</td>
<td>0.3143</td>
<td>-0.2219</td>
<td>0.1061</td>
</tr>
</tbody>
</table>

Figure 3.14: Factors Labeling and Item Selection

The data generated by the proposed instrument is captured in a Career Profile structure; and then used to cluster learners who are developing their career readiness into CoPs. CoP act as a virtual parallel educational path that focuses on career disciplines rather than career behavior. This new phase of professional traits development moves learners from generic professional attributes to specific domain-oriented readiness. Next, we further discuss our profiling structure of multidimensional data.

3.3.3 Profiling Structure

Out of collected data, we extract a structure to determine a career readiness construct, labelled Career Profile (Fig.3.15). This construct is designed as a standard mean to collect and
access information about learners while they are moving towards a predestined career path. Career profile augments an existing IEEE Learner Information Package (LIP) standard\(^6\) to capture learning data as well as career indicators. Our proposed construct of career profile is structured into three main categories aimed at predicting and assisting learners with their career development throughout their formal education. We use LIP-defined Interests, Competency, and Goal, categories to specify career interests, domain-related qualifications, and long term career objectives of individual learners. We differentiate two types of Interests: Career Interests and Social Interests. We also introduce a new category labeled Professional as a slot for career dispositions ratings and other generic attributes pertaining to career readiness. As shown in Fig.3.15, the multidimensional data attributes reflecting the professional aptitude, career prospects and dispositions of a learner are used to detect a CoP, where members share knowledge, experience and passion for a predicted practice to build capabilities and maintain momentum.

![Figure 3.15: Career Profile Structure](image)

### 3.3.4 Data Warehouse Architecture

We propose to build and maintain a multidimensional data warehouse which stores a large collection of learner-oriented, time-varying, integrated and non-volatile data to support cognitive apprenticeship development. The proposed career readiness warehouse presents multidimensional logical views about apprentices. The accumulated data across a range of assessment tool deployment instances provide a rich career-related data warehouse to facilitate the analysis based on historical data. To further assist mentors in deriving learning path

---

\(^6\)[http://www.imsglobal.org/profiles/index.html]
construction so that their comments are systematic and more tailored to their apprentices' profile, we could augment the data set with three additional sources, besides the assessment-tool generated data about learning dispositions. The first is retrieved from learner profiles, which are standard attributes stereotyping learners. The second is derived from domain ontologies to customize learning recommendations according to domain knowledge. Finally, the last data set is collected from social interactions of traces left by learners within communities (such as clickstream data logs). The role of the mentor itself could be based on some recommendation engines to automatically derive a recommended learning path. Thus, career readiness warehouse presents a centralized repository that stores learners' data from multiple information sources, and transforms them into a common, multidimensional data model for efficient querying and analysis.

![Multidimensional Data Warehouse Architecture](image)

**Figure 3.16: Multidimensional Data Warehouse Architecture**

### 3.4 Data Analytics and Visualization

Following the design of a data warehouse repository aligned across our multidimensional structure, we present analytical tools and view models to derive valuable information, as discussed further next.
3.4.1 Data Design

Multidimensional Conceptual Schema

In order to design the model of our multidimensional data reflecting career attributes, we use the CGMD model introduced earlier in the background section, as it meets most of the criteria for multidimensional data modeling. The syntax and semantics of the CGMD data model are discussed further in [75]. The conceptual schema for our base data used to model learners is presented in Fig.3.17; while Fig.3.18 illustrates the conceptual schema for the basic multidimensional information of the base data considered in Fig.3.17. Entities are Learner, Career Specialization, Self Evaluation Report, Courses, Mentor, Community of Practice and Social Network; whereas relationships are for instance Enrollment, Evaluation, Participation, Building and Monitoring. The cardinality constraints such as (1, n) on the "Evaluation" relationship between "Learner" & "Self Evaluation Report" entities express that learners fill the self-evaluation report at some dates during their academic year; while the cardinality constraint (1, n) on the "Participation" relationship between "Learner" & "Community of Practice" indicates that learners can join more than one CoP according to their career interests. The basic multidimensional entities are utilized to analyze career readiness of learners by considering, among others, the dimensions related to their career interests, career dispositions, courses they enroll and CoPs they join. This is with respect to the Career Specialization (associated to the industry needs) and Evaluation Date collected by the Self Evaluation Report. So, the entity Learner may represent a basic cube whose dimensions are Career Profile, Enrolls and Participation (identifying relationships) which are restricted to the basic levels Self Evaluation Report, Courses, and CoP (associated entities) respectively. This part of the diagram still uses the standard constructs. A cube that is defined on the specified dimensions with their basic levels (e.g. Learners-by-Career Specialization-and-Mentor) is called a basic cube, otherwise, it is referred to as an aggregated cube.
Figure 3.17: Conceptual Schema for the Base Data for Learners
Figure 3.18: The Conceptual Schema for the Basic Multidimensional Information for the Base Data Considered in Fig. 3.17
Multidimensional aggregation schema

For building the aggregation level hierarchies of each dimension, we consider the followings: (1) discriminator of an entity such as creating ICTLearner or EngineeringLearner entities that are inherited from Learner properties; (2) generalization/specialization hierarchy such as creating a single entity of all subclasses (e.g. academic year of semesters) or a disjoint of a superclass (e.g. divide mentors into academic professors and industry experts; or domains to set of interests); (3) partial relationship; and (4) One-to-many relationships and many-to-many relationship which can be converted into levels of one-to-many relationships. Fig. 3.19 presents example of multidimensional conceptual schema including level hierarchies for considered dimensions. Outer boxes indicate levels; and inner boxes are their elements. The bold arrows (from lower level to higher level) denote a hierarchy. The disjoint-total constraint is drawn with a circle having “d” inside.

The levels “Domain” and “Interest” are created from the partitions of “Career Specialization” entity. An entity Career Specialization is partitioned according to an attribute such as “area” - which is a discriminator- into four basic career specializations and two higher level career specializations. Interest-area aggregates four basic career specialization (partitions) namely, Data modeling, Web design, Electrical, and Electronic; and Domain-area aggregates higher-level two (partitions) of career specialization via ICT and Engineering. Metric aggregates set of the associated basic items into their higher level of skills which aggregates six elements to present Career Dispositions. Thus, the first three constraints that are discriminator, generalization/specialization, and One-to-many relationships hold. Similarly, multiple hierarchies are created for the evaluation date dimension derived by the Self Evaluation Report entity to include year, semester and month. Study-time and Study-break is another optional level as the relation between evaluation date and Study-time and Study-break is partial. When including activities about learners’ tracking in the virtual space, we may have more partial relationships along date dimension to capture for example days, log times and duration of learners’ activities online.
Figure 3.19: The Multidimensional Conceptual Schema for the Part of Base Data in Fig. 3.17.
Next, if we need to track the progress of learners in developing their career dispositions by semester we perform the analysis of learners along multiple dimensions. For this example, the query “visualize learner’s career dispositions by domain and semester” is an aggregated cube along the Evaluation, Profiling, and Career Specializations dimensions involving levels of Self Evaluation Report, Career dispositions and Domain, respectively. Fig. 3.20 presents the conceptual schema for this aggregated cube or query. This view includes the basic cube (i.e. learner’s career dispositions by career specializations and evaluation date) and the definition of aggregation as a new aggregation entity Learner’s Career Dispositions-by-Domain-and-Semester. We may also compute aggregation view for mentors to visualize all or certain levels of career dispositions (i.e. OC) of a group of learners in a given class or group of learners with specific career interests during one semester (i.e. mid-term, final-term) or over a couple of years, and so on. In next section, we provide examples of visualization views for career dispositions for both learners and mentors.

3.4.2 Dashboard Views

In order to visualize generated career disposition reports of a learner, we adopt a dashboard design as a circle segment [83] model (discussed earlier in the Background section), which assigns attributes on the segments of a circle. Data items are arranged within a segment so that a single data item appears in the same position at different segments. On completion of the self-report survey (SRCDS), the career warehouse generates a circle diagram (Fig. 3.21), that provides a visualization for the learner to reflect on (their own perception of) their career readiness in terms of general skills. The circle segment graphically depicts the pattern and relative strength of individual scores. It also provides a brief report on side about the scores as well as the updated feedback from mentors while observing the development of these dimensions. The scores produced are a percentage of the total possible score for that dimension. Fig. 3.22 presents an example of different visualizations for career dispositions across a group of learners (that may be a study group or a community on a class).
Figure 3.20: The Schema of Aggregated Cube View Presenting Learner's Career Dispositions-by-Domain-and-Semester.
According to feedback from your mentor, you will start to build more learning relationships with your colleagues, and taking more of learning challenges that may improve your resilience.

Your Career Dispositions report show that you are intellectual and creative individual who likes new experience. You need to develop your ability to make evidence-based decisions, and to develop more effective social relationships to support your learning. These social relationships will be part of your learning resources, which you are required to be able to manage them along with other learning resources including time. Your weakened in maintaining high communication and social skills leads also affects your resilience degree.

Figure 3.21: Circle Diagram Visualization Results of Career Dispositions Report of Individual Learner

Figure 3.22: Example of Different Visualizations of Career Dispositions
Career dispositions data can be also aggregated across groups of learners in order to provide a mentor with a view of the collective profile on all or specific career dispositions as illustrated in Fig 3.23a. Fig. 3.23a indicates direct to the mentor that almost 35% - 45% of learners score good (yellow) on all dimensions; while almost 30% of learners score poorly on all dimensions but reliance that comes with least poor percentage (17%). Most importantly, it indicates that a same percentage of learners (25%) score high (green) on all dimensions, and those learners make up the seed set we are selecting to help their colleagues develop career capacity while interacting in CoP networks. Finally, Fig.3.23b presents an aggregate visual analytics on a specific dimension that is openness to challenge.
Career Disposition Aggregated Scores for Learners

<table>
<thead>
<tr>
<th>Responsibility for Learning</th>
<th>Learning Relationships</th>
<th>Resilience</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good 37%</td>
<td>Good 43%</td>
<td>Poor 33%</td>
</tr>
<tr>
<td>Excellent 26%</td>
<td>Excellent 24%</td>
<td>Excellent 24%</td>
</tr>
<tr>
<td>Critical Thinking</td>
<td>Openness to Challenge</td>
<td>Creativity</td>
</tr>
<tr>
<td>Good 47%</td>
<td>Excellent 26%</td>
<td>Good 45%</td>
</tr>
<tr>
<td>Excellent 26%</td>
<td>Good 45%</td>
<td>Poor 33%</td>
</tr>
<tr>
<td>Poor 27%</td>
<td>Good 45%</td>
<td>Poor 29%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) Visual Analytics on Aggregate Career Disposition Data for all Dimensions

(b) Visual analytics on aggregate Career Disposition Data for Specific Dimension

Figure 3.23: Career Dispositions Across Group of Learners

Fig. 3.24 shows an example of an individual learner’s dashboard – providing ‘Ghanim’ with an overview of different analytics (career dispositions, social networks analytics) and recommendations from mentors. The dashboard may also have links to up-to-date in-
dustry trends and job market needs for future planning. For mentors, they will be provided with an overview of their learners' analytics and recommendations; and they would have agreed access to elements of their dashboards in order to view the system generated recommendations and add their own feedback. Thus, learners will be able to work on their career capacity with clear visualizations and prompt feedback and recommendations.

Figure 3.24: Analytics Dashboard for an Individual Learner

3.5 Implementation and Application

3.5.1 Portal Design

A portal architecture is suggested to collect the career-related data of learners in order to be stored in the career readiness warehouse shown in Fig.3.16. The portal presents an analytical platform for reporting individual performances, visualizations and recommendations towards a predicted career path. Each learner is assessed by different instruments such as the career dispositions survey and career profile forms. The proposed data design keeps
track of previous instruments and forms in case of subsequent updates to link assessment results to the corresponding instrument. In the example of career dispositions, answers to the survey questionnaires are related to corresponding categories to analyze the improvement across a given dimension. Finally, the mentor provided comments are related to the corresponding coaches or advisors who made specific recommendations to their apprentices as presented in previous section. To further clarify the design of our portal: 3.25 and 3.26 show the use cases advertised by the portal for both the learner and the mentor.

Figure 3.25: Learner Use Cases

Figure 3.26: Mentor Use Case
3.5.2 Deployment and Evaluation

An empirical study on the usability of the portal was conducted to evaluate users' perception, as a step prior to its deployment and exploitation over the upcoming year to assess its effectiveness in improving education. The usability study provides inputs on how real users perceive the proposed portal in terms of its utility, its promises to improve their cognitive learning ability and its ease of use. This is an essential step before moving to the data analysis step of the portal. It will ensure the sustainability of a genuine self-reflective declaration and a pedagogically-grounded cognitive learning apprenticeship experience (away from technical aspects of the portal), in order to formulate conclusions that are based solely on users' belief in the portal facility for the intended career skills development purpose. Without a usability testing, the data collected later upon full deployment of the portal may not be related solely to advancing an individual career readiness power but to the functionality related issues of the portal. Therefore, a preliminary test sample of 23 participants from various educational backgrounds and computer literacy was selected for this usability testing experiment. The test sample is formed by 12 undergraduate students and 11 graduate students. An initial demonstration of the portal system was conducted to introduce the participants to the use of the cognitive learning apprenticeship instrument and the portal processes. The system was deployed on a university-wide accessible server where network bandwidth and availability were not a concern. The participants were then asked to use the portal and report back their experience by answering a separate questionnaire, which was designed to better understand users' motivation and perceptions.

The feedback from the participants was very encouraging. Sixteen participants reported that they understood the principles behind the tool and the portal system, while seven indicated partial comprehension. Almost all (22) participants reported that the system was easy to use and eager to embrace the cognitive apprenticeship experience. In an explicit statement, many (18) of them indicated that they were willing to continue using the portal framework to be part of the second phase of the portal deployment.
We also evaluated the learnability of the portal to assess how well the portal design enables users to learn how to use it. An average time taken to complete the instrument in the first attempt was 24 minutes, while the subsequent attempts were completed in an average time of 11 minutes. This indicates that it took 13 minutes in average to initially learn about using portal. We also collected feedback from participants about the portal design. The results of this feedback are shown in Table 3.5 and Fig. 3.27 where we observe that most participants are able to use the portal processes with minimal efforts. The most enjoyable experience, as suggested by most of them, is the presentation of learning disposition results in a graphical format which gives them a nutshell view of their cognitive learning apprenticeship experience. The feedback from the mentor were deemed useful to diagnose deficiencies and transform some learning habits towards maximizing learning dispositions.

<table>
<thead>
<tr>
<th>Questions</th>
<th>Possible Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. What is your opinion on</td>
<td>A. Easy to Use</td>
</tr>
<tr>
<td>System Design?</td>
<td>B. Some challenges but easy to use</td>
</tr>
<tr>
<td></td>
<td>C. Needs redesign</td>
</tr>
<tr>
<td>2. Describe your experience in</td>
<td>A. Easy to use</td>
</tr>
<tr>
<td>using the tool.</td>
<td>B. Spent some time getting familiar to use</td>
</tr>
<tr>
<td></td>
<td>C. Spent some time getting familiar to use</td>
</tr>
<tr>
<td></td>
<td>D. Spent too much time and was difficult to comprehend</td>
</tr>
</tbody>
</table>

Table 3.5: Users' Feedback Form
Most participants were able to successfully use the portal framework, but a few suggested that the navigation could be improved as shown in 3.27. We also found that people who entered a low ease-of-use rating were not as comfortable as the others with English language (as the portal was used within a non-English native community of learners).

3.6 Summary

We presented our data-analytics solution to support career preparedness for learners in higher education; and sustain lifelong learning across our multidimensional data model. We proposed a design structure of a data warehouse to collect and store career-related data of various types and from multiple sources (career profile, domain ontologies and social logs). We also introduced the concept of “Career Dispositions” as a six-dimensional model to describe cognitive attributes such as attitudes, skills and preparation levels of each learner towards professional practices. We developed and validated a new scale to measure these dispositions metrics; and a portal application to collect data and visualize derived results while monitoring feedbacks. Our proposed module of career readiness allows learners to view their own profile data and feedback as well guiding information via a friendly inter-
face for analytics via dashboard views. It also enables mentors and academic supervisors to track and view performance of learners in simple and aggregated analytic visual presentations. The evaluation of this module of our overall framework shows that users appreciate its objective, simplicity and usability. Thus, the module provides a descriptive and prescriptive analytical tool to reinforce career readiness experience of learners in higher education contexts. The proposed multidimensional data structure can also be used to develop more learning analytics required in higher education as we discussed earlier in this chapter. Further, we build on this work to propose our predictive analytic approach for career prevision, described in the next chapter.
Chapter 4: Predictive Analytics Approach to Career Prevision

4.1 Introduction

Data mining and predictive analytics are increasingly used in higher education to classify students and to predict their behavior. Data mining techniques are used to find patterns in data; and it is typically defined as a "process of selection, exploration and modeling of large quantities of data to discover regularities or relations that are at first unknown with the aim of obtaining clear and useful results" [119]. It is an integral step that is efficient for searching through large amounts of data to discover novel and potentially useful information [120, 63]. Predictive analytics is a branch of data mining that is concerned with building and assessing empirical models to analyze current and historical data in order to predict future behavior. Building a predictive model requires careful definition of the prediction goal as it impacts the type of predictive model and methods to be used. One common goal of predictive models is to predict the output value \( y \) for observations given their input values \( x \) such that \( y = f(x) \), given that measurable values \( x \) are available to time of model deployment[121]. This goal is known as prediction for numerical outcomes or classification for categorical outcomes. In this essence, prediction is based on classifying a set of inputs into many classes through constructing a classification model to approximate the true mapping from input data to the appropriate outputs with the intent of generating predictions of outputs for new, previously unseen data[63]. The output accuracy and usability will depend greatly on the level of data analysis and the quality of assumptions when building and validating the prediction model; and this is why the predictive analytics is usually an iterative process (see Fig. 4.1).
In recent years, predictive data mining has introduced in higher education research as a solution to gain more value out of large amounts of raw data for prediction and intervention purposes [122, 123, 124]. This involves estimating learners' future knowledge or performance in terms of finding early indicators for learning success, failure, and potential dropouts, to be able to offer proactive interventions and support for learners in need of assistance. Actually, there is a wide range of potential applications of predictive data analytics in higher education including predicting alumni contributions, predict registration, predicting standardized test scores, predicting outcomes and intervention success, predicting student performance and identifying at-risk students, and identifying appropriate academic programs for each learner to improve future offerings of courses and provide timely feedback to students during the semester [125, 126]. One in-practice example of a distinctive use of predictive analytics is higher education is presented at the University of Wollongong, where Library Cube uses library usage data to predict student grades [127]. Macfadyen and Dawson [128] examined the effect of set of variables (total time online, number of web links visited, etc.) on predicting students' final grade. Zimmermann et al. [129] constructed a predictive model to predict graduate level of performance from undergraduate achievements in order to improve future graduate study admission procedures. Koprinska [130] showed how predictive analytics can be used to gain a better understanding of the

Figure 4.1: Predictive analytics iterative process
assessment results toward predicting final marks. Minaei-Bidgoli and colleagues presented an approach to classify students in order to predict their final grade based on features extracted from logged data in an education web-based system [131]. Kovačić [132] employed a classification tree model to predict student success by mining their enrolment data; while Dekker et al. [133] tried to predict students' dropout and identify factors of success based on the use of different classification algorithms. Vandamme, Meškens & Superby [134] used decision trees, neural networks and linear discriminant analysis for the early identification of three categories of students: low, medium and high-risk students. Kizlice et al. [135] classified learners according to their interactions with course content in learning activities in MOOCs\(^1\); and according to their engagement patterns, and then compared clusters based on learners' characteristics and behaviors.

A recent overview of empirical evidence behind key objectives and methods of the potential adoption of data mining in education research from 2008 up to 2014 [136] reveals that the majority of studies investigate issues related to student behavior modeling and prediction of performance; while at the top of popular methods adopted for data analysis is clustering. Clustering is an instance based method that uses attributes of already stored set of similar related instances/objects to recognize the class or cluster for a set of unclassified instances (unlike classification that maps objects to a set of predefined classes). The instance are clustered that the intra-class similarities are maximized and the inter-class similarities are minimized based on some criteria. Once the clusters are identified, the objects are labeled with their matching clusters. Clustering methods are evaluated with a set of algorithmic criteria include precision, accuracy, sensitivity, coherence, and similarity weights.

In this chapter, we propose a semi-supervised clustering algorithm as a data mining predictive method to assign learners into common virtual CoPs according to their career interests. The proposed method uses a fuzzy-logic objective function to address issues pertaining to overlapping domains of career interests. Career interests data are drawn from

\(^1\)http://mooc.org
career profile; while careers labels that best fit our prediction is obtained from researching up-to-date industrial needs (i.e survey of in-demand top IT careers). For experiment purposes, we stimulate data from a real world scenario to apply our clustering method in addition to the typically used simulated and benchmarked data sets.

This chapter is organized as follows: Section 2 provides a background and related work literature review in data clustering with a focus on semi-supervised and fuzzy semi-supervised clustering methods. Background section also discusses CoP concepts, features, development, benefits and sustainability. In Section 3, we introduce and discuss our fuzzy semi-supervised clustering algorithm. We also explain our underlying assumptions and techniques to handle cluster overlap. Section 4 presents an experimental of our proposed algorithm in comparison with baseline methods for three different data sets of different overlap nature. Finally, Section 5 concludes this chapter with a summary of results.

4.2 Background and Related Work

4.2.1 Data Clustering

Cluster analysis is a discipline that involves set of methods and algorithms to analyze multivariate data in order to discover the natural groupings of set of points, objects, or patterns according to perceived or measured intrinsic features or similarity. Merriam-Webster Online Dictionary (2015) defines cluster analysis as: "a statistical classification technique for discovering whether the individuals of a population fall into different groups by making quantitative comparisons of multiple characteristics."\(^2\). Operationally, clustering can be defined as follows: given a representation of \( n \) objects, find the \( K \) groups based on a measure of similarity such that the similarities between objects in the same group are high, while the similarities between objects in different groups are low\(^1\). Accordingly, a cluster is a set of objects that is compact and isolated. Clustering analysis aims to find underlying structure in data and identify salient features. Data clustering has also been used to identify

\(^2\)http://www.merriam-webster.com/dictionary/cluster%20analysis)
the degree of similarity among organisms and as a method for organizing and summarizing the data through cluster prototypes [137].

As one of the most important topics in data mining, clustering has been studied for many years. Numerous methods have been developed for various clustering problems (i.e., documents categorization, analyzing time course gene expression, identifying customer profiles from shopping cart) [138, 139, 140, 141]. According to the characteristics of the data, clustering algorithms can be categorized into connectivity based models (connect objects to form clusters based on their distance), centroid based models (assign the objects to the nearest cluster center), distribution based models (objects belong most likely to the same distribution), and density based models (clusters are defined as areas of higher density than the remainder of the data set). According to the relationships of the clusters, clusterings can be categorized into hard clustering where each object belongs to at most one cluster, and soft clustering where each object belongs to one or more clusters. Clustering algorithms can be further categorized into unsupervised and semi-supervised based on if we have certain prior knowledge about the clusters. either small portion of labeled objects or a set of pair-wise constraints. Unlike supervised classification, unsupervised clustering assumes we do not have any knowledge about the clusters. Semi-supervised clustering, on the contrary, assumes that we know the labels of certain objects. These objects are usually used as “seeds” and the clustering then utilizes these seeds to improve the clustering performance. Notice certain noise can exist for the seeds, such as the labels for some seeds might be wrong, not all the classes have seeds, etc. Semi-supervised clustering is essentially similar to constrained clustering. In constrained clustering, the final clusters need to satisfy certain constrains. The most often used constrains are must-link and cannot-link. If two objects are connected by a must-link, they must be in the same cluster. If two objects are connected by a cannot-link, they must be in different clusters. Fig. 4.2 illustrates the spectrum of different learning problems across supervised and unsupervised methods. Notice if we know the labels of the seeds, we can infer must-links and cannot-links. However, if we know only the must-links and cannot-links, we can not infer the label of the relevant ob-
jects. Again, there could be noise on the constrains where the must-links and cannot-links might be wrong.

Figure 4.2: Spectrum between supervised and unsupervised learning: dots correspond to points without any labels. Points with labels are denoted by circles, asterisks and crosses. In (c), the must-link and must-not-link constraints are denoted by solid and dashed lines, respectively.

Clustering algorithms can be also broadly divided -based on the cluster structure- into two groups: hierarchical and partitional. Hierarchical clustering algorithms recursively find nested clusters either in agglomerative mode (starting with each data point in its own cluster and merging the most similar pair of clusters successively to form a cluster hierarchy) or in divisive (top-down) mode (starting with all the data points in one cluster and recursively dividing each cluster into smaller clusters) (see Fig.4.3). Input to a hierarchical algorithm is an $n \times n$ similarity matrix (SM), where $n$ is the number of objects to be clustered. Partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure using either an $n \times d$ pattern matrix (PM), where $n$ objects are embedded in a d-dimensional feature space, or an $n \times n$ (SM) (that can be also easily derived from a pattern matrix). The hierarchical and partitional algorithms partition the data into different non-overlapping subsets (hard assignment). A partition of a dataset $X = \{x_1, x_2, ..., x_N\}$, where $x_j = (x_{j1}, x_{j2}, ..., x_{jd}) \in \mathbb{R}^d$ with each measure $x_{ji}$ called a feature (attribute, dimension or variable) and $d$ is the input space dimensionality [142], is a collection $C = \{C_1, C_2, ..., C_k\}$ of $k$ non-overlapping data subsets. $C_i \neq \emptyset$ (non- null clusters) such that $C_1 \cup C_2 \cup ... \cup C_k = X$, where $X$ is the super cluster and $C_i \cap C_j = \emptyset$ for $i \neq j$. The data partition is overlapping if the condition ($C_i \cap C_j = \emptyset$ for $i \neq j$) is ignored and in that case the cluster will have sub clusters of different levels inside it [143].
However, since partitional algorithms use a number of relocation schemes that provide optimization to the clusters, clusters can be refined at each revisiting step; and thus gives them an advantage over hierarchical clustering. Partitional algorithms are also preferred in pattern recognition due to the nature of available data. We focus our work on partitional algorithms, and namely on $K$-means algorithm as the most popular partitional algorithm. Its wide popularity is attributed to its ease of implementation, simplicity, efficiency, and empirical success [1].

$K$-Means Algorithm  

$K$-means algorithm finds a partition such that the squared error between the empirical mean of a cluster and the points in the cluster is minimized. It generates a new partition by assigning each point to its closest cluster center then computes the new cluster centers. Let $X = \{x_i\}, i = 1, ..., n$ be the set of $n$ $d$-dimensional points to be clustered into a set of $K$ clusters $C = \{c_k, k = 1, ..., K\}$. Let $\mu_k$ be the mean of cluster $c_k$, the squared error or the vector quantization error ($\text{VQE}$) between $\mu_k$ and the points in cluster $c_k$ is defined:

\[
\text{VQE}(c_k) = \sum_{x_i \in c_k} \|x_i - \mu_k\|^2
\]

The objective of $K$-means is to find the set of partitions that (locally) minimizes the $\text{VQE}$ for all $K$ clusters:
\[ VQE(c_k) = \sum_{k=1}^{K} \sum_{\mu \in c_k} ||x_i - \mu||^2 \]

Selecting an initial \( K \), the algorithm keeps assigning points and updating the cluster centers until no more points to add or the centroids remain unchanged which means cluster membership is stabilized. The \( K \)-means algorithm can be represented as shown in Algorithm 1. Fig.4.4 shows an illustration of the \( K \)-means steps performing on a 2-dimensional data set with three clusters.

**Algorithm 1 \( K \)-means Algorithm**

1. Select \( K \) data points as the initial cluster centers \((\mu_1, \mu_2, \ldots, \mu_K)\)
2. Assign each point \( x_i \) to its closest cluster centers
3. Update cluster center as the mean of its constituent data points
4. Repeat 2 and 3 until \( K \)-means objective function is optimized

Minimizing objective function of the the sum of the squared error over all \( K \) clusters is known to be an NP-hard problem (even for \( K = 2 \))[144]. Thus \( K \)-means can only converge to a local minimum despite being a greedy algorithm. Since the squared error always decreases with an increase in the number of clusters \( K \) (with \( VQE(C) = 0 \) when \( K = n \)), it can be minimized only for a fixed number of clusters. The worst-case running time of \( k \)-means is superpolynomial. In particular, with set of \( n \) data points and a set of adversarially chosen cluster centers, the algorithm requires \( 2^{\Omega(\sqrt{n})} \) iterations [145]. Even if the initial cluster centers are chosen uniformly at random from the data points, the running time is still superpolynomial with high probability. While the upper bound is polynomial in the number of points, number of clusters, and the spread of the point set [146]. Let \( t_{dis} \) be the time to calculate the distance between two objects, then each iteration time complexity is \( O(Knt_{dis}) \); bound number of iterations \( I \) giving \( O(IKnt_{dis}) \). The space complexity for \( K \)-means is to store points and centroids in a \( d \) dimensional vector model is: \( O((n + K)d) \).
Figure 4.4: Illustration of K-means algorithm. (a) Two-dimensional input data with three clusters; (b) three seed points selected as cluster centers and initial assignment of the data points to clusters; (c) and (d) intermediate iterations updating cluster labels and their centers; (e) final clustering obtained by K-means algorithm at convergence [1].

K-means algorithm requires three user-specified parameters that are: (1) number of clusters $K$, (2) cluster initialization, and (3) distance metric. K-means run independently for different values of $K$ and the partition that appears the most meaningful to the domain expert is selected. On the other hand, different runs of the K-means will produce different results due to the different (random) initialization of the centroids. Hence, selecting the right initial centroids is very critical to improve the quality of resulting clusters. $K$-mean++ method proposes to select centroids in regions with high concentration of data points [147]. As for distance metric, K-means is typically used with the Euclidean metric for computing the distance between points and cluster centers. As a result, K-means finds spherical or ball-shaped clusters in data [1]. However, in the work of [148], Mahalanobis distance, which is a measure of the distance between a point and a distribution is used. The authors combined the Newton Raphson method and iterative projection together to learn a Mahalanobis distance for K-means clustering. In the work of [149], Jensen-Shannon divergence, which is a measure of the distance between two distributions is used. The authors used gradient descent for weighted Jensen-Shannon divergence in the context of
Further, the basic \( K \)-means algorithm has been extended in many different ways. In one extension of the hard assignment \( K \)-means, \textit{Fuzzy C-means}, a data point can be a member of multiple clusters with a membership value (soft assignment)[150]. Semi-supervised \( K \)-means algorithms are also introduced that use constraints or seeds to improve clustering quality. Next, we present and discuss the semi-supervised clustering research line and semi-supervised partitional and \( K \)-means models.

### 4.2.2 Semi-Supervised Clustering

The nature of clustering problems where is the aim is to define unknown number of partitions based only on intrinsic information, makes it very difficult to design clustering algorithms to correctly find clusters in the given data. Any external or side information available to support the \((n \times d)\) PM or the \((n \times n)\) SM can be extremely useful in finding a good partition of data. Clustering methods that utilize any side information are said to be operating in a semi-supervised mode [151]. One of the most common methods to specify the side information are in the forms of: (a) pairwise constraints where set of must-link and cannot-link specifies weather point pair connected by the constraint belong or not belong to the same cluster; and (b) seeding, where some labeled data is used along with large amount of unlabeled data for better clustering [1].

Must-link denoted by \( c = (x, y) \) and cannot-link denoted by \( c \neq (x, y) \), meaning that two instance \( x \) and \( y \) must be in the same cluster or cannot be in the same cluster respectively. Must-link constraints are an example of an equivalence relation and hence are symmetrical, reflexive and transitive; this means that \( c = (x, y) \) and \( c = (y, z) \) \( \Rightarrow c = (x, z) \) such that \( x, y, z \) form a connected component, i.e., each is connected to the other by a explicit or implied must-link constraint. Besides the must-link and cannot-link constraints, there are other versions of constraints: minimum separation constraint, or \( \sigma \)-constraint [152], which requires that any pair of points which are in two different clusters must have a distance greater than or equal to \( \sigma \); \( \rho \)-constraint [153], for each point \( x \) in a cluster, there must exist a neighbor
y of x, such that the distance between x and y is at most \( \rho \). Constraints can be used in clustering algorithms in two ways: (1) they can be used to modify the cluster assignment step in order to enforce satisfaction of the constraints as much as possible; or (2) to train the distance of the clustering algorithm either before or during the actual clustering [152]. In both cases, constraints can also be used to form initial clusters such that must-linked instances are in the same clusters and cannot-linked instances are in different clusters. Accordingly, the performance of constrained clustering methods is heavily dependent on the constraints and distance metrics.

The hierarchical clustering algorithm proposed in [154] illustrated that how the must-link and cannot-link constrains can be built in the hierarchies and how they can be satisfied during the hierarchical clustering process. In [155], the authors proposed a framework to cast clustering into a matrix completion problem to satisfy the must-link and cannot-link constraints. The goal of the proposed algorithm is to sample only a small portion of the constraints to recover the true data partition that satisfies all the constraints, thus the efficiency of the clustering can be significantly improved. The constraints are encoded as 1 or 0 binary values in a binary similarity matrix, indicating whether the instance is in a cluster or not. The clustering problem is then converted to a matrix completion problem where the missing entries in the matrix need to be filled out from the partial constraints and the input data matrix. The contribution of the work is to develop a matrix completion algorithm that explicitly incorporates the data matrix into the matrix completion process. The crucial assumption here is the cluster membership vectors can be well approximated by the top few singular vectors of the data matrix. Based on this assumption, a new objective function that fits the observed constraints is proposed and the FSGD (fast stochastic sub gradient descent) technique is applied to optimize the objective function. In [156], the authors propose methods for integrating the constraints into the Gaussian Mixture Models (GMMs) under the Expectation Maximization (EM) framework. They assume that the data distribution in each cluster is a Gaussian distribution. They applied a generalized EM algorithm to handle cannot-link constraints whereas a closed form EM algorithm is successfully
obtained for must-link constraints. The EM algorithm considers log-likelihood over only cluster assignments complying strictly the given constraints. Thus it is not able to handle noisy constraints. In [157], an extended EM algorithm Penalized Probabilistic Clustering (PPC), is proposed based on the work of [156]. PPC integrates pairwise constraints into the Gaussian Mixture Models under the EM algorithm to adjust the prior distributions through a weighting function. The weighting function has large values when the label assignment is consistent with the given pairwise constraints and low values when the assignment conflicts with the constraints. The genetic algorithm proposed in [158] considers a penalty for the cluster impurity where the constraints are violated. The penalty cost function is based on Gini-index, which is a measure to determine the impurity of a certain split in decision trees.

As the performance of clustering is heavily dependent on the distance metric, there are lots of research focusing on learning an optimal distance metric for semi-supervised clustering. The distance metric learning methods can be categorized mainly into two categories: global distance metric learning [159, 160, 161, 162], where all the constraints are satisfied simultaneously and local distance metric learning [163, 164], where the constraints are satisfied locally. The distance metric learning methods have a tendency to over-fit, especially for high dimensional data. In [165], the authors proposed an evolutionary distance metric learning (EDML) framework that utilizes an evolutionary algorithm to search a sub-optimal metric transformation. They showed that method is more resistant to over-fitting. On the other hand, existing constraints of the semi-supervised clustering methods only consider the given constraints and do not consider the neighbors around the data points constrained by the constraints. The must-link and cannot-link constraints often are not allocated to different clusters. The data points of these constraints are often close to each other in the transformed space or with the learned similarity measure. Thus, in order to ensure the performance of semi-supervised clustering, generally, a large number of constraints are needed. In [166], the authors aim to use fewer labeled data points for semi-supervised clustering, based on constraint projection. The goal of constraint projection [167, 168] is to use the constraints to determine the low-dimensional space. In [166], a CNP (Constrained
Neighborhood Projection) approach is proposed, where both the given constraints and the neighbors of the constraint points are used to find the transformation. The objective function is to find a transformation matrix that encourages data points involved in must-link set and their neighborhood be close while data points involved in cannot-link set and their neighborhood be far away from each other in the transformed lower-dimensional space. The transformation matrix is then applied to convert the original space into a lower-dimensional space and K-means is applied on the new space for the clustering. In different work, [169], the authors argue that the constraints should be actively selected as maximally informative ones rather than chosen at random. This would imply that fewer constraints will be required to significantly improve the clustering accuracy. In their work, pairwise constraints are selected among points lying at the clusters boundaries. The work also assigns weights to the constraints, which are used to measure the cost of violation of these constraints. The distance metric is learned based on the Gaussian function with a cluster dependent scaling parameter so that each cluster can be scaled differently. In addition, boosting algorithms have been applied for semi-supervised clustering so that the clustering results can be gradually improved during the boosting process. In [170], DisBoost algorithm builds a set of weak hypotheses in each iteration from the Gaussian Mixture Models, trained on the observed data, the weights, the hidden labels of the data and the pairwise constraints. The weights of all data are initialized the same and the weights of the misclassified pairs will be increased so that in the next iteration the weak hypothesis will focus on satisfying these constraints. The distance function will be the weighted sum of all weak hypotheses at each iteration. In [171], BoostCluster is proposed to improve the performance of arbitrary clustering algorithms by using pairwise constraints as side-information. The objective function consists of two components, one is the disagreement between the kernel similarity matrix and the must-link constraints, the other is the inconsistency between the kernel similarity matrix and the cannot-link constraints. This objective function is minimized iteratively in a boosting style by updating the kernel similarity matrix, according to the clustering results of the given algorithm on the transformed input data from the previous step.
There are semi-supervised clustering methods that combine both the constraint-based and distance-based approaches. For example, in [172], a probabilistic framework based on Hidden Markov Random Fields (HMRFs) is proposed. The work assumes the prior probability of a label configuration follows the Gibbs distribution. The EM algorithm aims to maximize a joint probability of the observed data, the label configuration and the model parameters conditioned on the constraints. The distance metric learning is integrated into the EM process by applying penalty scaling functions which are monotonically increasing functions of the distance between the data points based on the distance function. The optimization problem is challenging so many approximation techniques are applied for local optimal solutions, which may lead to poor performance. The MKM (metric learning KMEANS) and MPKM (metric pairwise constraint KMEANS) algorithms proposed in [173] are special cases of the above framework.

More improvements on constrained semi-supervised clustering are proposed by maximizing constraint margin as illustrated in detail by Wang [174] and Zeng et al. [175, 176]. In [174], the method seeks a linear subspace in which the constraint margin of the data set is maximized. The constraint margin considers both the scattersness of different classes and the compactness of same class. Moreover, a constraint margin maximization criterion is given to identify an optimal projection matrix such that in the projected lower-dimensional space, the constraint margin of the data points is maximized. The criterion can be also used to estimate the optimal dimensionality of such a linear space automatically. A kernelized version of the algorithm is proposed to handle nonlinear data sets. Spectral algorithms-based pairwise constraints to learn distance metrics are stated by the authors in [138, 177]. In [178], a new graph-based constrained clustering algorithm SCRAWL is proposed with the aim to utilize the limited but informative pairwise constraints. The basic idea is to expand the constraint influence to the unconstrained edges. A component membership degree of each vertex is defined as the degree of influence it receives from the constrained vertex. Then the affecting scope of each pairwise constraint is derived as the edge set connecting the components around the two constrained vertices. The fraction of a fractional edge is
the product the fractions of the connected vertices. To expand the constraint influence, the components of the fractional edges are grouped into clusters. The cluster assignments of the vertices are determined by combining the cluster membership of the fractional vertices.

A semi-supervised algorithm Constrained-DBSCAN [179] based on DBSCAN [180] is developed. DBSCAN is a density-based clustering algorithm. It requires two parameters: the radius of neighborhoods, the minimum number points threshold needed to form a cluster. DBSCAN first picks randomly a point which has not been visited yet. If the number of points in its neighborhood is less than the minimum number points threshold, that point will be labelled as noise. Otherwise a cluster is formed with the initial points are the points in that neighborhood. The Constrained-DBSCAN follows the same idea of DBSCAN with the extension that adding points to a cluster must guarantee the constraints. The algorithm compute the transitive closures of must-link constraints by applying the transitivity property of must-link constraints and update the set of cannot-link constraints by the entailment property of cannot-link constraints and must-link constraints. Then, the algorithm goes through unclassified points and extends (if possible) the cluster of each point but still maintains the constraints. Active-Selecting-DBSCAN [179][43] is developed to obtain the set of informative constraints and then passes them to Constrained-DBSCAN. Active-Selecting-DBSCAN is able to determine if a data point is likely to be a border point by checking if the size of the neighborhood is smaller than the minimum number points threshold. Active-Selecting-DBSCAN is shown to achieve good performance due to the fact that the constraints obtained contains good information about the cluster boundaries.

In[181], an ensemble-based selection procedure for identifying the most informative constraints is proposed. The ensemble-based procedure consists of two phases: imputing constraints from pairwise co-associations and selecting informative constraints. A base clustering algorithm on sampled data points is applied to construct the co-association matrix. The entry in the matrix indicates the fraction of base clustering in which the two points are assigned to the same cluster. Given minimum and maximum confidence thresholds that two points are in the same cluster, the sets of imputed must-link constraints and
cannot-link constraints are obtained. Then $K$ neighborhoods and their representatives are computed from the must-link constraints. The initial set of clusters are then constructed. In the selection phase, the set of constraints will be expanded by selecting the most uncertain point to form the queries. Cluster size based weights are added to compute a certainty value. The queries between the most uncertain points and the $K$ representatives are posed.

There is another line of research on interactive semi-supervised clustering, where the user iteratively provides feedback to a clustering algorithm. The feedback is collected in the form of cannot-link/must-link constraints which the clustering algorithm tries to satisfy in the next iterations by adjusting the distance metric. In [182], Explore-Consolidate is proposed to actively ask the user to provide feedbacks to queries. The active learning scheme is divided into two phases: Explore and Consolidate. The Explore phase tries to get $K$ pairwise disjoint non-null neighborhoods as fast as possible where $K$ is the number of clusters. A neighborhood is defined as a set of data objects belonging to the same cluster. The Consolidate phase is used to identify the neighborhood (cluster) for the remaining data objects. A drawback of the Explore-Consolidate scheme is that the Consolidate phase only selects random points different from the points selected in the Explore phase (denoted as the skeletal points) to form queries. The work in [183] suggested that selecting the most uncertain points in the Consolidate phase to form the queries could improve the performance. The certainty of a data point is defined as the maximal similarity between the point and all other points. In [184], a new constraint is added to force the selection process favors the centroids of dense region. Also instead of selecting pairwise constraints, the queries are now on selecting the seeds. In [185], the work IG-KMEANS addressed another drawback of Explore-Consolidate [182]: the queries do not take into account the intermediate clustering results (which can be very useful in determining the informative constraints). In each iteration, the algorithm selects the best data points pairs to form queries based on a gain function measuring how much information obtained when revealing the judgements of the selected pairs. A gain function is proposed for this purpose. The selection of the optimal data points pairs is very expensive. So a greedy strategy is proposed, where a datapoint-gain
function is proposed to measure the average information of a data point. The data points are then ranked according to their gain function. The data point with the highest value is paired with the data point in the same cluster to ask for the judgment. The datapoint-gain function has different form for cases where the data points pairs are selected independently or dependently. The drawback of the method is the time complexity to compute the gain function.

The “curse-of-dimensionality” is always a challenging problem for clustering. In [167], a semi-supervised clustering method SCREEn (spheRical K-means via fEature projectioN) was proposed to address the clustering problem where the dimensionality is high and the constraints are must-link and cannot-link. The algorithm first reduces instances by replacing the transitive closure of must-link instances as an average instance. Then a constraint-guided feature projection approach is applied to further improve the efficiency of the semi-supervised clustering. The objective function of the projection is to learn a projection matrix to project the original datasets into a low-dimensional space such that the distance between any pair of instances involved in the cannot-link constraints are maximized while the distance between any pair of instances involved in the must-link constraints are minimized. Spherical-KMeans (SPKMeans) algorithm [139] first normalize the data vectors to have unit $L_2$ norm and then apply the standard $K$-means algorithm. The method is shown to have good performance for sparse high dimensional data, which is common for domains like text clustering. Once the optimal projection matrix is identified, Spherical $K$-means algorithm is applied on the lower-dimensional space to cluster the data points. The work assumes all the constraints are correct during the initialization step, which is not realistic. Thus a possible extension is to include constraint violation cost into the objective function.

Major semi-supervised methods based on K-means

A) Constraints-Based Methods There are many constraints-based semi-supervised clustering method based on k-means algorithm. One of the first is the Constraint K-means
(COP-K Means) algorithm [186], that uses the two types of constraints -must-link and cannot-link- in the clustering process to generate a partition that satisfies all the given constraints. The algorithms takes in a data set, a set of \( c = (x, y) \) constraints, and a set of \( c \neq (x, y) \). It returns a partition of the instances in the data set that satisfies all specified constraints such that when updating cluster assignment, it ensures that none of specified constraints are violated. However, as discussed earlier in this review, when constraints are generated from labeled data there is the possibility of class label noise result in generating constraints between two instances that should not be. Even if the constraints are generated by domain experts, some constraints may be ill-specified or even contradictory. This leads to the introduction of the algorithms with distance penalties that ignore noisy or inappropriate constraints by allowing constraints to be left unsatisfied but with a penalty. This involves a trade-off between finding the best clustering and satisfying as many constraints as possible. To achieved this, the penalty of ignoring a constraint must be in the same units as the measure for how good the clustering of the data is. The PCK-Means (Pairwise-constrained k-Means) algorithm and the CVQE (constrained vector quantization error) algorithm are the most popular partition algorithms which penalize constraint violations using distance.

PCK-Means algorithm [173] considers the must-link and cannot-link constraints and it allows violation of constraints if it leads to a more cohesive clustering, and uses them to seed the initial cluster centroids and to influence the clustering. The objective function of PCK-Means is different from the standard K-means objective function, as it includes the cost of violating any pairwise constraints. MPCK Means algorithm [173] is similar to PCK Means and it performs distance-metric training in each clustering iteration, making use of both unlabeled data and pairwise constraints. The algorithm assumes the distance metric for each cluster is different and it learns the individual metrics for each cluster. Then it uses a symmetric positive-definite weight matrix to adjust distances of points in different clusters, allowing clusters of different shapes.

Two other K-means related methods that allow violations of constraints are CVQE [153] and LCVQE [187]. The constraint violation costs are added into the objective func-
tion \((VQE)\) of \(K\)-means. When a must-link constraint is violated, the penalty of the distance between two nearest cluster centers of these two points is added to the objective function. The cost of violating a cannot-link constraint is computed as the distance between the cluster center that these two points are assigned to and the cluster center nearest to this cluster center. For each pair of points in the constraints, the \(CVQE\) objective value is calculated for each combination of cluster assignments, and the cluster assignments which minimally increase the \(CVQE\) objective value is selected. The cluster centers are updated when the violation happens. The steps for algorithms employ \(CVQE\) objective function is described in Algorithm 2. The \(LCVQE\) [187] modifies the \(CVQE\) objective function as the following: the penalty of violating a must-link constraint is the distance from second point of the constraint (assume the second point is the violated point) to the center of the cluster where the first point is assigned to; the penalty of violating a cannot-link constraint to be the distance from the farthest point (with respect to the center of the cluster where the two points belong to) to another nearest center. \(LCVQE\) improves \(CVQE\) by not computing all possible \(K^2\) combination assignments but only at most three reasonable assignments.

**Algorithm 2 CVQE Algorithm to penalize constraint violation using distance**

1. For instances that are not part of constraints, perform a nearest cluster assignment as in Algorithm 1
2. For pairs of instances in a constraint, assign instances to the cluster that minimally increase the \(CVQE\) such that:
   - If \(c = (x, y)\) is violated
     - Then the penalty is the distance between the two centroids of the clusters containing the two instances that should be together.
   - If \(c \neq (x, y)\) is violated
     - Then the penalty is the distance between the cluster centroid the two instances are assigned to and distance to the nearest cluster centroid.
3. Update the cluster centroids so as to minimize \(CVQE\)

**B) Seeding-Based Methods**  Seeded clustering algorithms use a small amount of labeled data to aid and bias the clustering of unlabeled data. Given a dataset \(X\), the goal is to split this dataset into \(K\) disjoint clusters \(\{X_h\}_{h=1}^{K}\) such that the local objectives function is minimized. Let \(S \subseteq X\) be the subset of data objects, called the seed set. For each \(x_i \in S\), the label \(y_i = h\) of \(x_i\) denotes the cluster \(X_h\) which \(x_i\) belongs to. The seed set \(S\) is partitioned into
L disjoint sets: \( \{S_h\}_{h=1}^L \), where \( L \leq K \). If \( L = K \), the seed set is called complete. Otherwise, it is the case of an incomplete seeding. In CSK-means [188], a complete seed set is used to initialize the k-means algorithm. Thus, rather than initializing k-means from \( k \) random data points, the initial center of the \( lth \) cluster is initialized with the mean of the \( lth \) partition \( S_l \) of the seed set. In ISK-Means [188], an incomplete seed set is used to initialize the \( K \)-means algorithm.

In the SKM algorithm proposed in [189], the mean of the clusters are initialized by the mean of the seeds of each class. The seeds are then reassigned with labels computed using \( K \)-means; while in the C-KMeans algorithm proposed by same authors, the mean of the clusters are initialized by the mean of the seeds of each class. However, the label of the seeds remain unchanged. Algorithm 3 illustrates the basic steps of seeding-based \( K \)-means (SKM) methods 3. In [190], a semi-supervised subtractive clustering algorithm by seeding is proposed. The algorithm applies the similar idea of the SKM from [189] to initialize the clusters from the seeds. However, instead of using the seeds to compute the cluster center directly, it computes a mountain function for each data point and select the data point with the highest value as a cluster center. The label of the cluster center is assigned as the label of its nearest seed. The mountain function has an advantage that once a cluster center is picked, the values of all the data points close to the center drop dramatically compared with the values of the data points far from the center, which encourages the identification of new cluster centers from the data points far away.

**Algorithm 3 Seeded K-Means (SKM)**

1. Initial center of the \( lth \) cluster with the mean of the \( lth \) partition \( S_l \) of the seed set.
2. Assign each data point \( x_i \in S_l \) is to the cluster center \( \mu_l \)
3. Assign each data point \( x_i \in X-S_l \) to the closet cluster center
4. Update cluster center as the mean of its constituent data points
5. Repeat 3 and 4 until \( K \)-means objective function is optimized
Fuzzy Semi-Supervised Clustering

Fuzzy clustering in general is used for categorization problems which require a realistic overlapping clusters representation, as the problem we address in this research. In [191], Bora et al. compared the two types of clustering algorithms: Soft Clustering (Overlapping Clustering) and Hard Clustering (or Exclusive Clustering). They showed that in many situations, fuzzy clustering is more natural than hard clustering, especially for objects on the boundaries between several classes as they are not forced to fully belong to one of the classes, but rather are assigned membership degrees between 0 and 1 indicating their partial membership.

In [192], Girar et al. studied the semi-supervised fuzzy clustering problem. They considered cannot-link and must-link constraints and they proposed a Fuzzy C-means based algorithm: pairwise-constrained competitive agglomeration (PCCA). Their algorithm assigns a membership score to an instance belonging to different clusters and the algorithm considered the costs of violations on the constraints. The advantage of PCCA over fuzzy C-means is that it does not require pre-specification of the number of clusters. Instead, it conducts an iterative reallocation that partitions a data set into an optimal number of clusters by locally minimizing the sum of intra-cluster distances and the costs of violations on the constraints. The link constraints they used here do not have associated labels, namely two instances can be either cannot-link or must-link, but not both. This is indeed not natural as the instances in the overlapped region can be in both different clusters and in the same cluster. The membership score for the instances on the borders is highly likely to be inaccurate. And as the two instances of a link constraint might have different membership scores, it is inappropriate to consider simply cannot-link and must-link without considering membership score. The constraints in the overlapped region can be very useful to reconstruct such region accurately. However, the membership scores making constraint definition in the overlapped region less accurate. Thus in general the membership scores making the semi-supervised fuzzy clustering problem less natural and inaccurate.
Cominetti et al. [193] proposed a novel fuzzy clustering algorithm which can better handle clusters that are non-linear, curved, elongated etc. They showed that the traditional fuzzy clustering methods such as Fuzzy C-means have limitations that they all rely on the centroids of the clusters and thus these methods tend to have poor performance on clusters with non-circle shapes. The focus of the work is to apply concepts from diffusion processes in graphs to fuzzy clustering. It divides the instances into hard instances (belonging to only one cluster) and soft instances (belonging to multiple clusters). It identifies all the hard instances first using their neighborhoods. Then it assigns a membership value to the soft instances based on their closest hard instances. They compare the performance of their method against Fuzzy C-mean on Iris data set. Although they achieved better performance, the improvement is subtle, indicating Iris data set is more “convex-shaped”, namely circle shaped.

Endo and Yukihiro [194] considered semi-supervised fuzzy clustering with mutual relation constraints (FCMMR), which differs from must-link or cannot-link in that it is not determined whether a group of data in one mutual relation are placed in the same cluster or not in advance. The instances in the same mutual relation share certain amount resources which are the constraints here. The objective is to find a set of cluster partition so that the total resource allocated to all the instances in each mutual relation satisfy the constraints and in the meanwhile the intra cluster distances is minimized. Thus the instances in the same mutual relation can be in different clusters. Here the mutual relation constraints are not relevant to the clusters and thus the membership score would not affect the constraints.

Fu and Enzo [195] proposed a fuzzy clustering method FLAME, which aims to address the clustering problems with non-linear properties. Similar to [193], it uses the memberships of an instance’s nearest neighbors to infer the membership of the instance. Therefore, the memberships of instances do not depend on the centroids of the clusters, but rather the memberships of their neighbors. They showed that a high membership score indicates that the instance belongs to certain cluster and a low membership score indicates that the instance does not belong to the cluster. An instance with all low membership scores
is likely to be an outlier. However, the threshold of high and low membership score is not straight-forward to determine.

Zhang and Lu [196] proposed a kernel-based semi-supervised learning algorithm SSKFCM to enhance the fuzzy partition quality. The method extends semi-supervised clustering to a kernel space in order to partition the clusters into groups with nonlinear boundaries in the input space. The objective function of SSKFCM is defined by adding classification errors of both the labeled and the unlabeled data, and its global optimum has been obtained through repeatedly updating the fuzzy memberships and the optimized kernel parameter. They showed that using more constraints, the performance of the semi-supervised learning algorithm can be improved.

As a summary, it is not straight-forward to determine a threshold score for high and low membership scores by Fuzzy C-means. The constraints are not straight-forward to define in the overlapped regions where the instances can have different membership scores to the same cluster. On the other hand, it is not clear how the constraints helped to improve the accuracy on the overlapped region. It might be more appropriate to let the user specify the constraints in the overlapped region that whether a link can be both must-link and cannot-link, namely both instances are in the overlapped region. Then based on such constraints we capture the true overlapped region with high confidence. In Table 4.1, we summarize some of the most popular $K$-mean based hard and soft semi-supervised methods.
<table>
<thead>
<tr>
<th>Method</th>
<th>Criteria</th>
<th>Main Idea</th>
</tr>
</thead>
<tbody>
<tr>
<td>COP-KMeans</td>
<td>Hard Clustering; Hard constraint satisfaction</td>
<td>Allows only membership assignments that don’t violate the constraints</td>
</tr>
<tr>
<td>MPC-KMeans</td>
<td>Hard Clustering; Hard constraint satisfaction</td>
<td>A probabilistic generative model where constraints are accompanied by associated violation costs $W$</td>
</tr>
<tr>
<td>Pairwise Constrained-Means (PC-KM)</td>
<td>Hard Clustering; Hard constraint satisfaction</td>
<td>A seed set is used to initialize cluster centers. Seeds memberships could be changed</td>
</tr>
<tr>
<td>Seeded-KMeans (SKM)</td>
<td>Hard Clustering; Hard constraint satisfaction</td>
<td>A seed set is used to initialize cluster centers. Seeds memberships kept unchanged</td>
</tr>
<tr>
<td>P-PKM</td>
<td>Hard Clustering; Soft constraint satisfaction</td>
<td>Finds maximum number of instances per neighborhood and enforces a probabilistic penalty of constraint violation</td>
</tr>
<tr>
<td>M-KMeans (MKM)</td>
<td>Hard Clustering; Soft constraints satisfaction</td>
<td>Performs metric learning from constraints, but does not require that the constraints be satisfied</td>
</tr>
<tr>
<td>CVQE</td>
<td>Hard Clustering; Soft constraint satisfaction</td>
<td>Penalizes constraint violations using distance</td>
</tr>
<tr>
<td>AFCC</td>
<td>Soft Clustering</td>
<td>A data element could belong to some clusters depending on the membership degree. Integrating membership degree in objective function</td>
</tr>
<tr>
<td>FLAME</td>
<td>Soft Clustering</td>
<td>Uses the memberships of an instance’s nearest neighbors to infer the membership of the instance</td>
</tr>
<tr>
<td>SSKFCM</td>
<td>Soft Clustering, Hard constraints satisfaction</td>
<td>Adds classification errors of both the labeled and the unlabeled data; and obtain the global optimum through repeatedly updating the fuzzy memberships and the optimized kernel parameter</td>
</tr>
<tr>
<td>FCMMR</td>
<td>Soft Clustering, Hard constraints satisfaction</td>
<td>Find a set of clusters such that the total resource allocated to all the instances in each mutual relation satisfy the constraints and in the meanwhile the intra cluster distances is minimized</td>
</tr>
</tbody>
</table>

Table 4.1: Popular $K$-mean based semi-supervised methods
4.2.3 Communities of Practice (CoPs)

CoP Definition

Virtual Community of Practice (CoPs) is defined as "computer-mediated discussion form focused on problems of practice that enable individuals to exchange advice and ideas with other based on common interest" [197]. Simply, it is an online CoP; and this why we refer to our computerized method as a CoP in general. CoPs play central role in knowledge management (KM) strategies and collaborative learning as perceived as an effective mechanism for knowledge creation and exchange. Knowledge in its turn is central to formal education and professional practice. KM literature differentiates between knowledge sharing and knowledge exchange that knowledge sharing can occur in one-way broadcast form, whereas knowledge exchange occurs at dyadic level and indicates a reciprocal relationship [198]. Knowledge is also often divided into two distinct entities: explicit knowledge (knowing that) refers to possession of information and facts; and tacit knowledge (knowing how) refers to procedural and application form of knowledge [199]. CoPs are particularly celebrated for the creation and exchange of tacit knowledge.

The theory of CoPs lies at the intersection of knowledge transfer and learning process; and it has become more widespread in higher education due to the benefits derived from collaborative generation of knowledge and cooperative learning activities within and outside the classroom [28, 200, 201, 202]. Johnson noted "the learning evolved form these communities is collaborative, in which the collaborative knowledge of the community is greater than any individual knowledge"[203]. It is also argued that individual learning is enhanced through engagement with others enabling the extension of the individuals’ capability to a new and higher level[204]. The Learning in CoPs is recognized as “situated learning” that is defined as knowing how to be in practice rather than knowing about the practice [25, 26, 205]. This involves the process of formation of the individual identity as becoming a member of the community and participate in knowledge development. Supported with the
sense of connectedness, knowledge development within CoPs can be continuous, and fluid in a cyclic pattern [112]. Learning in CoPs also occurs within the context of the cyclical process of DDAE: dialogue, decision-making, action and evaluation [206]. Examining DDEA dynamics in CoPs draw a clear link between the group's capacity to evaluate its current practices and earn through this evaluation by talking about it (dialogue), make decisions based on this discussion, and subsequently implement this decision into action [112, 207].

Learners participate in CoPs are engaged to interact and gain knowledge and skills from community members, partly who positioned as masters or experts. This view reiterates the apprenticeship model and suited learning in higher education [208]. The demands to shift the emphasis from the abstract bodies of knowledge thought in formal higher education towards the CoP-based suited learning is forced by the need to equip learners with skills to deal with real-world problems and to reduce the current education-market disparities [28, 209].

**CoPs Development and Sustainability**

To qualify as a virtual CoP, the description and characteristics of a conventional CoP apply (i.e. common interest domain, notion of interaction and shared resources of practice) etc. CoP should include active members who are practitioners, or "experts," in a specific domain of interest; or who share general area of interest). Members must actively engage in a collective learning process within their domain through social structures that assist in knowledge creation and sharing. There should be multiple mechanisms to facilitate the long-term management of support as well as to enable immediate synchronous interactions. The distinct structuring characteristic of CoPs lie in its technological environment, such as using the Internet to be called virtual; and the ICT-based interaction and knowledge creation tools. CoPs are enabled in higher education by the increasing ICT solutions that enable communication, interactivity of participants and incorporation of pedagogical models. A typical technological platform of CoPs is the forum-based within which content is organized into topics, and topics consist of threads including posts and related discussions. Members of
the community typically post information to suitable thread, and other members respond or comment on the post.

Active participation of the community members is determinate by several factors that include self-interest (i.e. career development, material gain), normative considerations (i.e. shared values and vision, reciprocity), community-related considerations (i.e. a sense of belonging, a common sense of purpose, cultural dimensions), usability of technology, and leadership [24, 210, 211]. More censorious factors that contribute to the participants willingness to exchange knowledge and so to the success of CoPs are trust and acceptance [198, 212, 213, 197]. The fact is that even after initial acceptance, most of VCoPs fail to stimulate members to exchange knowledge, and suffer from the lack of continuous member participation, and which eventually threat their success. Further, most of CoPs members are knowledge consumers rather than producers, which also threat the CoPs suitability [214, 215]. This is mainly because the development of knowledge within a CoP is an essential feature of it, particularly when considering the relationship between VCoPs and professional expertise and competence [216, 112].

Given that the VCoPs survival and sustainability depend on ongoing member participation and, and voluntary knowledge creation contribution [217, 218], there should be a mechanism to retain the active members who are the most motivated to participate, and who most contribute by posting or commenting for other posts[214]. Members of CoPs joined based on their practice interest and their ability to contribute to the domain. However, in order for the community to grow and have meaning, the individual members must be motivated to join and to engage with it actively to create and maintain information flow. In this essence, trust building is crucial as without trust, members of CoPs may be reluctant to share knowledge [219, 220]. The lack of face-to-face interaction in online environment as well as the hidden identities may lead individuals to fail to engage in the CoP, preferring to work autonomously[24]. Shifting individual membership from a peripheral participation to full membership through a process of enculturation (the process whereby individuals learn their group's culture, through experience, observation, and instruction) is another barrier as
described earlier by Wenger et al. [27]. Peripheral participation is described as "a way to speak about the relations between newcomers and old-timers, and about activities, identities, artifacts, and communities of knowledge and practice. A person's intentions to learn are engaged and the meaning of learning is configured through the process of becoming a full participant in a socio-cultural practice"[25, 27].

On the other hand, one of the most critical success factors CoPs is communication that is fundamental in the development of trust and the community. Trust is built through continued interactions to develop common values and a shared understanding[219, 213]. In addition, identifying group members with prior knowledge of each other help to consolidate membership and develop trust [221]. Individuals' commitment to continuous learning and is another factor to sustain the collaborative culture in the CoPs and -thus- the practice improvement [222].

In our research work, we investigate sustainability factor at early stage of the community construction. We propose a predictive data mining technique to semantically analyze Career Profile in order to cluster individuals into appropriate classes of clusters representing CoP and predict their career paths. While building our model, we utilize the individual preferences or career dispositions that indicate the potential active members in order to assign them to the core of the community and relay on their ability to start knowledge exchange frequently, and thus ensure the sustainability of constructed CoPs.

4.3 Fuzzy Pairwise-constraints K-Means (FPKM)

We propose a career prediction module that analyzes data from Career Profile (including career disposition values) in order to predict a hypothetical career practice and bring learners with similar career patterns together into a common cluster. This process leads to a social structure made up of CoPs that are identified to specifically respond to imminent industrial needs. To solve the cold-start problem of CoP construction, we use the career readiness data warehouse discussed earlier as a source for initializing groups (or clusters) of learners and denote each such cluster as a CoP (see Fig. . In order to conduct this initial grouping
process, we apply a semi-supervised clustering technique that brings a seed set of learners into an initial set of CoPs. The seed set consists of learners who achieved high scores in career disposition values that are above a given parameter threshold. There is typically at least one seed member in each cluster (CoP) for which his/her career profile matches the definition suggested by the career ontology that yielded the CoP. The rationale of privileging highly ranked learners in their career dispositions to create dedicated CoPs is driven by the prospects to sustain CoPs. From this initial stage, we infer the use of career disposition values only to provide seed set of new CoPs (including the initial ones).

In our proposed semi-clustering algorithm, we assume the followings:

- We have seeds and each class will have at least one seed. The seed labels are always correct.
- We have pairwise constraints, must-links and cannot-links. These constraints could be wrong.
- We allow fuzzy labeling, namely each instance can be in more than one cluster.
- All labels are assigned to both seeds and constraints.

One challenging problem occurs when and whether a violation of the link constraint should be penalized. In traditional semi-supervised clustering algorithms, a violation of the link constraint is always penalized. Now, as we allow the instances to be associated with multiple labels, a constraint can be violated legitimately. For example, as shown in Fig.4.5, the must-link between B and C is only within Cluster C2. If we use label C2 for B, and label C3 for C, the must-link can be violated legitimately. On the contrary, for the cannot-link between A and D, there is no way that it can be violated legitimately. Thus, the penalty function needs to be re-designed to allow fuzzy labeling and to estimate if a constraint violation could be legitimate or not.
According to this logic, we develop the Fuzzy Pairwise-constraints K-Means (FCKM) algorithm that is presented in Algorithm 4; while notions and symbols are described in Table 4.2. The main steps of the FCKM algorithm are as follows:

1. Define overlapped clusters using the side-information: seeds and pairwise constraints

2. Initialize the centroids of each cluster as the average of the seeds belonging to that cluster

3. Assign instances to minimize the new objective function $O_{new}$ shown in Equation (1)

4. Update the cluster centroids to minimize the objective function as shown in Equation (2)

5. Repeat until convergence

For each cluster $C$, we first identify all the seeds $S_{c_1}, S_{c_2}, \ldots, S_{c_t}$ belonging to the cluster. Then we initialize the centroids of each cluster as the average of the seeds belonging to that cluster $\mu_c = \sum_{i=1}^{t} \frac{S_{ci}}{t}$. As we allow soft-constraints, namely the pairwise constraints could be wrong, we apply a penalty function on each constraint violation. As we showed in the above example, not every violation should receive a penalty. We need to determine when a violation should not receive a penalty. Assuming we are assigning the instance $x_a$, we develop the following new objective function, Equation (1), which is an updated version of CVQE objective function (see Algorithm 2):
\[ O_{new} = \frac{1}{2} \sum_{x_a \in C_j} D(x_a, \mu_j)^2 \]

\[ + I(label((x_a, x_b) \in C_-) \neq j) \frac{1}{2} \sum_{(x_a, x_b) \in C_-, y_a \neq y_b, y_a = j} D(x_b, \mu_j)^2 \]

\[ + I(label((x_a, x_b) \in C_\neq) \neq j) \frac{1}{2} \sum_{(x_a, x_b) \in C_\neq, y_a = y_b, D(x_a, \mu_j) < D(x_b, \mu_j), j = h'(x_b)} D(x_b, \mu_j)^2 \]

For instances that are not part of constraints, perform a nearest cluster centroid calculation. For pairs of instances in a constraint, for each possible combination of cluster assignments, the function is calculated and the instances are assigned to the clusters that minimally increases the error term \( h^* = \text{arg} \min_h O_{new} \). \( I(A) \) is an indicator function defined as follows: \( I(A) = 0 \) if \( A = \text{True} \) and \( I(A) = 1 \) if \( A = \text{False} \), and \( label(x_a, x_b) \) is the label of the constraint. Thus when a link is violated, we check if its associated label is different from the label that \( x_a \) is assigned to. If yes, the violation is not penalized. Notice the above objective function is computed incrementally. When a new instance \( x_a \) is to be assigned, we just need to increase \( O_{new} \) by \( D(x_a, \mu_j)^2 \) and the corresponding penalties, based on the assignment of \( x_a \).

For an instance \( x \in X \), and \( x \) is not a seed, assign \( x \) to the cluster \( h^* = \text{arg} \min_h O_{new} \).

Once we assign an instance to a cluster \( C_j \), we update the cluster centroid \( \mu_j \) as follows, Equation (2) [152]:

\[
\begin{align*}
\mu_j &= \frac{\sum_{x_i \in C_j} x_i + \text{sum}_1 + \text{sum}_2}{|\mu_j| + \text{total}_1 + \text{total}_2} \\
\text{sum}_1 &= I(label((x_a, x_b) \in C_-) \neq j) \sum_{(x_a, x_b) \in C_-, y_a \neq y_b, y_a = j} x_b \\
\text{sum}_2 &= I(label((x_a, x_b) \in C_\neq) \neq j) \sum_{(x_a, x_b) \in C_\neq, y_a = y_b, D(x_a, \mu_j) < D(x_b, \mu_j), j = h'(x_b)} x_b \\
\text{total}_1 &= I(label((x_a, x_b) \in C_-) \neq j) \sum_{(x_a, x_b) \in C_-, y_a \neq y_b, y_a = j} 1 \\
\text{total}_2 &= I(label((x_a, x_b) \in C_\neq) \neq j) \sum_{(x_a, x_b) \in C_\neq, y_a = y_b, D(x_a, \mu_j) < D(x_b, \mu_j), j = h'(x_b)} 1
\end{align*}
\]
The update rule applies that if a must-link constraint is violated, the cluster centroid is moved towards the other cluster containing the other instance. Similarly, the interpretation of the update rule for a cannot-link constraint violation is that cluster centroid containing both constrained instances should be moved to the nearest cluster centroid so that one of the instances eventually gets assigned to it, thereby satisfying the constraint.

An instance in an overlapped region should be assigned to multiple clusters. If \( x \) is close to the centroids of multiple clusters, namely \( |\text{argmin}_{h_1} O_{new} - \text{argmin}_{h_2} O_{new}| < \delta \), where \( \delta \) is a pre-defined threshold, we assign \( x \) with both \( h_1 \) and \( h_2 \) and update the centroids of the two clusters simultaneously. However, as the clusters are of different shape, it's not possible to pre-determine an appropriate \( \delta \). And for more than two clusters, we need to conduct multiple comparisons, making the problem expensive. To address the issue, we consider overlapped region as one cluster and we call them overlapped – cluster (OC).

If we have seeds in these OC (any seed with multiple labels), we could initialize its centroids as \( \mu_{OC} = \frac{\sum_{S \in OC} S}{|OC|} \). In our scenario, we always have seed information for the overlapped – clusters, either we have seeds in these regions, or we have link-constraints in the region. The nodes in the link-constraints, when all labels are given, are equivalent to seeds. Therefore, if the overlapped – cluster centroid is the closest centroid to an instance, the instance is assigned to the overlapped – cluster, meaning the instance belongs to all clusters that overlap.

**Algorithm 4 Fuzzy Pairwise-constraints K-Means (FCKM)**

**Input**
A dataset \( X = \{x_a, \ldots, x_n\} \) to cluster, \( C \) : the number of clusters, \( S \) : set of seeds, set of \( C = \{(x_a, x_b)\} \), set of \( C_\neq \{(x_a, x_b)\} \)

**Output**
A partition of \( X \) into \( C \) clusters that is a local optima of the Equation (1).

**Method**
1. Initialize clusters: \( \mu_c = \frac{\sum_{l=1}^{C} x_{il}}{C} \)
2. Repeat until convergence:
   (a) Assign each data point \( x_a \) to the nearest cluster to optimize Equation (1)
   (b) Update centroids \( \mu_1 \ldots \mu_c \) according to Equation (2)
3. Return \( C \) clusters.
<table>
<thead>
<tr>
<th>Object</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$</td>
<td>the input domain</td>
</tr>
<tr>
<td>$C$</td>
<td>number of clusters</td>
</tr>
<tr>
<td>$\mu_c$</td>
<td>initial centroids of clusters</td>
</tr>
<tr>
<td>$i, j$</td>
<td>indices running over clusters</td>
</tr>
<tr>
<td>$a, b$</td>
<td>indices running over instances or output clusters’ labels</td>
</tr>
<tr>
<td>$x_a$</td>
<td>input data instance $x_a \in X$</td>
</tr>
<tr>
<td>$y_a$</td>
<td>output cluster label $y_a \in {C}$</td>
</tr>
<tr>
<td>$D(x_a, \mu_j)$</td>
<td>distance between instance $x_a$ and center of cluster</td>
</tr>
<tr>
<td>$C_m$</td>
<td>must-link constraints</td>
</tr>
</tbody>
</table>
| $C
\ne$  | cannot-link constraints                          |

$h^* = \arg\min_{h \in \text{new}} \text{instance assignment that minimally increases the error terms}$

Table 4.2: Notions and Symbols

### 4.4 Experiment and Performance Analysis

#### 4.4.1 Experiment Environment and Metrics

We used MATLAB environment to design three major experiments utilizing: (1) 2-dimensional artificial data set; (2) benchmarked data set (IRIS) [55]; and (3) multidimensional data set simulated from career-related real world scenario. Each major experiment consists of several experiments or rounds to test different parameters such as number of seeds and degree of overlap between clusters. We compare the results of our proposed methods along two $K$-means candidate methods: (1) Seeded $K$-Means (SKM); and (2) CVQE-based Pairwise-constrained $K$-Means algorithm (denoted as PKM) that allows constraints violation with certain penalties. To evaluate the performance of the clustering algorithms, we employ external metrics that utilize a priori knowledge about the classification information of the data set. External metrics rely on the true class memberships in the data set. For soft clustering, the most used external evaluation metric is the F-Score. F-Score is a weighted combination of precision and recall to reflect the overall quality of the resulting clusters. These performance metrics are:

- **Precision**: is the ratio: $\frac{tp}{(tp+fp)}$, where $tp$ is the number of true positives and $fp$ the
number of false positives. The precision is intuitively the ability of the classifier not to label as positive a sample that is negative.

- Recall: is the ratio \( \frac{tp}{tp + fn} \) where \( tp \) is the number of true positives and \( fn \) the number of false negatives. The recall is intuitively the ability of the classifier to find all the positive samples.

- F-Score: is a combination (harmonic mean) of precision and recall to reflect the overall quality of the resulting clusters. The F-Score is defined as follows:

\[
F = \frac{(1+\alpha) \cdot \text{precision} \cdot \text{recall}}{\alpha \cdot \text{precision} + \text{recall}}
\]

- Total F-Score: typically, precision and recall are given equal weight with \( \alpha = 1 \). Varying the coefficient \( \alpha \) provides a mean of biasing F-Score towards precision or recall (e.g., \( \alpha = 0.5 \) biases it towards precision; \( \alpha = 2.0 \) biases it towards recall). The total F-Score is calculated as the average of the largest F-Score of each cluster given the total number of objects \( N \) in the sample as follows:

\[
F = \frac{\sum_{c=1}^{C} \frac{tp + fp}{N} \cdot F - \text{Score}(c)}{N}
\]

- Accuracy of clustering: is another evaluation measure that discovers the one-to-one relationship between real clusters and the ground-truth categories. Accuracy measures the extent to which each cluster contains the objects from the corresponding ground-truth category. It is defined as follows:

\[
\text{Accuracy} = \sum_{c=1}^{C} \frac{tp}{N}
\]

### 4.4.2 Artificial Data

#### Experimental Setup

In order to simulate overlapped clusters, we generate uniformly distributed data within a circle seen as a cluster, as follows:

- Randomly generate the center of the clusters. Then for each cluster, take a radius as
input and randomly sample a given number of data points in the circle.

- To determine if a data point belongs to multiple clusters, consider the distance of the data point to each cluster center. If the distance is no greater than the radius of the cluster, the point belongs to the cluster.

**Experimental Parameters**

We simulated a two-dimensional \((2-d)\) artificial data. The centers of clusters are generated randomly \((\mu = 0; \sigma = 1)\) within the range, which is a circle with \((0,0)\) as the center and \(R = 15\) as the radius of the circle in which cluster centers are generated. Then, for each cluster we consider its radius as input and then randomly sample a given number of data points within that circle (following a uniform distribution). To determine if a data point belongs to multiple clusters, we consider the distance of data points to each cluster center. If the distance is no greater than the radius of the cluster, then we consider that the point belongs to the cluster. The generated data set consists of three clusters \((C = 3)\) with 200 samples in each cluster. The constraints used in our algorithm are generated as follows: for each constraint, we randomly pick two instances from the data (following a uniform distribution) and then we check their labels (which are made available for the evaluation purpose but not visible to the clustering algorithm). If they exhibit any common label, we generated a must-link constraint. Otherwise, we generate a cannot-link constraint.

In order to determine the effectiveness of the proposed algorithm and the reliability of the experiment results, we designed three data sets with three different levels of overlapping degree. Fig.4.6 shows an example of three overlapping degree: a) all three clusters overlap, b) only two clusters overlapping, and c) one cluster is entirely within another cluster. The overlap degree is controlled by the radius formula discussed earlier, and which also controls the number of instances within the overlap region from its minimum value in the first set to its maximum in the third set of experiments. For the same number of clusters and overlap degree, we generate different sets of seeds and constraints along the following
ratios of the total number of nodes $[1\%,5\%,10\%]$.

![All three clusters overlap](image1)

![Only two clusters overlap](image2)

![One cluster in another cluster](image3)

Figure 4.6: Different levels of overlap degree

**Results and Discussion**

For the the Fuzzy Pairwise- constraints K-means (FPKM) algorithm, the results showed it achieved higher accuracy than the baseline methods (see Fig 4.7). This is because the recall of FPKM is generally very high, much higher than those of the baseline algorithms, as the baseline algorithms do not consider overlaps and thus the assignment for the nodes in the overlapped region is relatively random. Many true positives are missed. The recall of the fuzzy algorithm is, however, affected by the degree of overlap: the more the clusters overlap, the lower the recall is. This is obvious because with more overlap, there are more true positives we need to capture and the more true positives the algorithm tends to miss. Thus the recall decreases, and so the overall accuracy, see Table 4.3
Table 4.3: Overlap degree Vs Accuracy of FPKM on 2-d Artificial Data

<table>
<thead>
<tr>
<th>Index</th>
<th># of nodes in overlap region</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>18</td>
<td>0.997778</td>
</tr>
<tr>
<td>2</td>
<td>19</td>
<td>0.996667</td>
</tr>
<tr>
<td>3</td>
<td>33</td>
<td>0.99625</td>
</tr>
<tr>
<td>4</td>
<td>83</td>
<td>0.994074</td>
</tr>
<tr>
<td>5</td>
<td>133</td>
<td>0.94037</td>
</tr>
<tr>
<td>6</td>
<td>180</td>
<td>0.752593</td>
</tr>
<tr>
<td>7</td>
<td>25</td>
<td>0.997778</td>
</tr>
<tr>
<td>8</td>
<td>14</td>
<td>0.996667</td>
</tr>
<tr>
<td>9</td>
<td>6</td>
<td>0.996111</td>
</tr>
<tr>
<td>10</td>
<td>103</td>
<td>0.902593</td>
</tr>
<tr>
<td>11</td>
<td>146</td>
<td>0.952963</td>
</tr>
<tr>
<td>12</td>
<td>173</td>
<td>0.914074</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>0.993889</td>
</tr>
<tr>
<td>14</td>
<td>16</td>
<td>0.995556</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>0.997222</td>
</tr>
<tr>
<td>16</td>
<td>128</td>
<td>0.858889</td>
</tr>
<tr>
<td>17</td>
<td>173</td>
<td>0.90963</td>
</tr>
<tr>
<td>18</td>
<td>188</td>
<td>0.767778</td>
</tr>
</tbody>
</table>

Figure 4.7: Accuracy of three clustering methods (2-d Artificial Data)

Fig. 4.8 shows F-score curves when $\alpha$ is $[0.5, 1, 2]$ for the three methods as the
overlap increases. As the figure indicates, when degree of overlap increases, the performance of the fuzzy algorithm becomes better than those of the baseline algorithms. It is noticed that when the overlap degree is low, the performance of our proposed method is less than the baseline methods. This is can be justified by the lower precision value achieved by the fuzzy algorithm. The denominator of the precision is the number of nodes assigned to the cluster. For the fuzzy algorithm, as it considers overlaps, it usually assigns more nodes to each cluster, which makes the denominator larger. However, when overlap degree increases, it is often the case that all three clusters overlap with each other - the baseline methods then tend to make many mistakes which makes the precision poor. We can see the precision of the baseline methods and so the F-score generally drops when overlap degree increases. On the contrary, the fuzzy algorithm returns better precision as overlap increase. This is because the fuzzy algorithm generally tends to assign more nodes to the overlapped region. When the clusters overlap more, more nodes assigned to the overlapped regions are correct, leading to higher precision.

4.4.3 Benchmarked Data

Experimental Setup

For the benchmarked data evaluation phase, we selected the IRIS data set. IRIS data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant (Setosa, Versicolour, Virginica). The latter two classes are somewhat overlapped as seen in the visualized figures in Chapter 1. It has 4 features (sepal length, sepal width, petal length, and petal width).

We first need to generate the ground-truth, which instances are in the overlapped region. Our criterion is: if the distance of an instance to the center of the other cluster (class) is larger than the maximum internal distance of the other cluster (maximum in-group pairwise distance), then this instance falls into the overlapped region. It turns out that 29 out of 50 instances in class Versicolour belongs to the overlapped region, and 13 out of
50 instances in class Virginica is in the overlapped region. Therefore, we consider these 42 instances as all in the overlapped region.

**Experimental Parameters**

For performance evaluation purpose, we consider the following two scenarios:

1. Vary the number of seeds in the overlapped region: here we vary the number of seeds in the overlapped region as 5, 10, 15, 20, 25 and evaluate the performance of our proposed FPKM clustering algorithm.

2. Vary the size of the overlapped region with fixed number of seeds: here we vary the size of the overlapped region as 10, 15, 20, 25, 30, 35 and 40, with a fixed number of seeds that is 5. We then evaluate the performance of all methods.
<table>
<thead>
<tr>
<th>Number of Seeds</th>
<th>Setosa</th>
<th>Versicolour</th>
<th>Virginica</th>
<th>[Versicolour, Virginica]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>0.736698</td>
<td>0.857113</td>
<td>0.388589</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.734242</td>
<td>0.862598</td>
<td>0.444580</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>0.736743</td>
<td>0.866021</td>
<td>0.482571</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>0.742481</td>
<td>0.871530</td>
<td>0.520833</td>
</tr>
<tr>
<td>25</td>
<td>1</td>
<td>0.739036</td>
<td>0.871514</td>
<td>0.536115</td>
</tr>
</tbody>
</table>

Table 4.4: Precision of FPKM for different number of seeds (IRIS)

<table>
<thead>
<tr>
<th>Number of Seeds</th>
<th>Setosa</th>
<th>Versicolour</th>
<th>Virginica</th>
<th>[Versicolour, Virginica]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.854</td>
<td>0.857143</td>
<td>0.759493</td>
<td>0.333333</td>
</tr>
<tr>
<td>10</td>
<td>0.84</td>
<td>0.868254</td>
<td>0.794937</td>
<td>0.416667</td>
</tr>
<tr>
<td>15</td>
<td>0.838</td>
<td>0.884127</td>
<td>0.818988</td>
<td>0.485719</td>
</tr>
<tr>
<td>20</td>
<td>0.838</td>
<td>0.901587</td>
<td>0.841772</td>
<td>0.554762</td>
</tr>
<tr>
<td>25</td>
<td>0.826</td>
<td>0.903175</td>
<td>0.859494</td>
<td>0.590476</td>
</tr>
</tbody>
</table>

Table 4.5: Recall of FPKM for different number of seeds (IRIS)

For each parameter setting, we randomly simulate 3 data sets and we show the average performance of the 3 data sets. We set the number of seeds for each class (excluding the overlapped region) to be 5, and give 5 must-link constraints as well as 5 cannot-link constraints. The seeds and constraints are randomly sampled and they do not overlap with each other. Notice for the two baseline methods used in our experiments, we randomly assign the instances in the overlapped region to each over-lapped cluster, and take those assignments as ground-truth. Therefore, we could expect these two methods to have poor performance in the overlapped region.

Results and Discussion

Scenario (1): with different numbers of seeds in the overlapped region, we compute the precision (Table 4.4), recall (Table 4.5) and F-score (Table 4.6) for each cluster, including the overlapped region. Total accuracy and Total F-Score of our proposed FPKM algorithm for different number of seeds in the overlapped region size is presented in Fig. 4.9(\(\alpha = 0.5\)).

We can see that overall our algorithm achieves good performance on IRIS data. The precision of the algorithm is in general high. By increasing the number of seeds in the overlapped region, the precision does not change much. However, the recall increases
Table 4.6: F-Score of FPKM for different number of seeds (IRIS)

<table>
<thead>
<tr>
<th>Number of Seeds</th>
<th>Setosa</th>
<th>Versicolour</th>
<th>Virginica</th>
<th>[Versicolour, Virginica]</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>0.946060</td>
<td>0.772876</td>
<td>0.821861</td>
<td>0.368188</td>
</tr>
<tr>
<td>10</td>
<td>0.940223</td>
<td>0.774042</td>
<td>0.838744</td>
<td>0.434764</td>
</tr>
<tr>
<td>15</td>
<td>0.939349</td>
<td>0.780043</td>
<td>0.849600</td>
<td>0.483769</td>
</tr>
<tr>
<td>20</td>
<td>0.939280</td>
<td>0.788824</td>
<td>0.861320</td>
<td>0.531568</td>
</tr>
<tr>
<td>25</td>
<td>0.934125</td>
<td>0.786598</td>
<td>0.867293</td>
<td>0.552867</td>
</tr>
</tbody>
</table>

as the number of seeds increases and the same for F-score, especially for the overlapped region. This indicates that the number of seeds in the overlapped region does affect the clustering performance and it has more significant impacts on the overlapped region. On the other hand, we can see in Fig.4.9 that both the total accuracy and total F-score for our fuzzy algorithm increases with respect to the number of seeds from the overlapped region. This shows the importance of the seeds in the overlapped region: these seeds could support our algorithm to recover the overlapped region effectively.

![Graph showing the performance of FPKM method for different number of seeds in the overlapped region size (IRIS)](image)

Figure 4.9: The performance of FPKM method for different number of seeds in the overlapped region size (IRIS)

Scenario (2) : to vary the size of the overlapped region we randomly sample instances from the overlapped region. We then compare the performance of all three methods by precision, recall and F-score as presented in Table4.7, Table4.8, and Table 4.9, respec-
We can see that the precision of our constrained fuzzy algorithm increases as the size of the overlapped region increases. While for the two baseline methods, the precision drops when the size of the overlapped region increases. This indicates that when the overlapped region is small, the effect of the seeds is not significant as the two baseline methods would not make many mistakes anyway. However, when the overlapped region is large, the two baseline methods might make more mistakes and thus the seeds in the overlapped region is critical for more accurate clustering.

<table>
<thead>
<tr>
<th>Method</th>
<th>Overlap Region Size</th>
<th>Setosa</th>
<th>Versicolor</th>
<th>Virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10</td>
<td>1</td>
<td>0.769994</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1</td>
<td>0.786584</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1</td>
<td>0.794635</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1</td>
<td>0.862567</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>0.86507</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>1</td>
<td>0.87963</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1</td>
<td>0.915046</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.986928</td>
<td>0.973162</td>
<td>0.945861</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.974108</td>
<td>0.963606</td>
<td>0.93083</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.980644</td>
<td>0.911905</td>
<td>0.939683</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.974596</td>
<td>0.904614</td>
<td>0.93879</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.955728</td>
<td>0.84451</td>
<td>0.8971</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.967823</td>
<td>0.831032</td>
<td>0.913624</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.980644</td>
<td>0.840688</td>
<td>0.903943</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.897972</td>
<td>0.590325</td>
<td>0.934837</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.931373</td>
<td>0.763794</td>
<td>0.958516</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.955491</td>
<td>0.79386</td>
<td>0.973447</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.95775</td>
<td>0.677205</td>
<td>0.950411</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.926573</td>
<td>0.741497</td>
<td>0.981187</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.949905</td>
<td>0.806834</td>
<td>0.92632</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.944081</td>
<td>0.603008</td>
<td>0.892416</td>
</tr>
</tbody>
</table>

Table 4.7: Precision of all methods for different overlap region-size (IRIS)
The recall of our algorithm is in general much better than that of the two baseline methods, due to the seeds in the overlapped region. The recall drops as the size of the overlapped region increases, as it becomes harder to assign the instances in the overlapped region to the correct clusters. The recall of the two baseline methods vary significantly with respect to the size the overlapped region, indicating their performances are not very stable.

Table 4.8: Recall of all methods for different overlap region-size (IRIS)

<table>
<thead>
<tr>
<th>Method</th>
<th>Overlap Region Size</th>
<th>Setosa</th>
<th>Versicolour</th>
<th>Virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPKM</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>0.978723</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1</td>
<td>1</td>
<td>0.961538</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1</td>
<td>1</td>
<td>0.953216</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1</td>
<td>1</td>
<td>0.962366</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>1</td>
<td>0.930348</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>1</td>
<td>1</td>
<td>0.925926</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1</td>
<td>1</td>
<td>0.917749</td>
</tr>
<tr>
<td>SKM</td>
<td>10</td>
<td>1</td>
<td>0.893651</td>
<td>0.983333</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1</td>
<td>0.872727</td>
<td>0.968411</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1</td>
<td>0.910354</td>
<td>0.916724</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1</td>
<td>0.912019</td>
<td>0.903504</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>0.836508</td>
<td>0.858998</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>1</td>
<td>0.89694</td>
<td>0.811022</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1</td>
<td>0.899592</td>
<td>0.814059</td>
</tr>
<tr>
<td>PKM</td>
<td>10</td>
<td>1</td>
<td>0.775397</td>
<td>0.515789</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1</td>
<td>0.829293</td>
<td>0.749031</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1</td>
<td>0.940972</td>
<td>0.789031</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1</td>
<td>0.902548</td>
<td>0.629351</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>0.804048</td>
<td>0.725242</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>1</td>
<td>0.889694</td>
<td>0.783097</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1</td>
<td>0.92585</td>
<td>0.346514</td>
</tr>
</tbody>
</table>
The $F$-score of our method is in general significantly better than that of the two baseline methods. Again, we observed that the performance of the two baseline methods are not very stable, especially for the second baseline method (PKM).

Fig. 4.10a and Fig. 4.10bshow the total accuracy and total $F$-Score of all methods when the size of the overlapped region increases. We can see that when the size of the overlapped region increases, in general the performance of the fuzzy algorithm increases or remain stable while the performance of the two baseline methods decrease. This again indicates the effects of seeds for accurately recovering the overlapped region. Without seeds, the clustering becomes harder when the overlapped region becomes larger, and thus the performance of the two baseline methods drops. On the contrary, with the help of seeds, the performance of our fuzzy algorithm does not drop even when the size of the overlapped

<table>
<thead>
<tr>
<th>Method</th>
<th>Overlap Region Size</th>
<th>Setosa</th>
<th>Versicolour</th>
<th>Virginica</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPKM</td>
<td>10</td>
<td>1</td>
<td>0.833621</td>
<td>0.992806</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>1</td>
<td>0.845956</td>
<td>0.986842</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>1</td>
<td>0.852787</td>
<td>0.983787</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>1</td>
<td>0.903964</td>
<td>0.986954</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>1</td>
<td>0.905664</td>
<td>0.975486</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>1</td>
<td>0.916388</td>
<td>0.974015</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>1</td>
<td>0.941703</td>
<td>0.970982</td>
</tr>
<tr>
<td>SKM</td>
<td>10</td>
<td>0.991228</td>
<td>0.943916</td>
<td>0.957693</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.98257</td>
<td>0.931053</td>
<td>0.943005</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.986956</td>
<td>0.910874</td>
<td>0.931412</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.982794</td>
<td>0.90667</td>
<td>0.926618</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.969973</td>
<td>0.841491</td>
<td>0.883459</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.978298</td>
<td>0.851617</td>
<td>0.876319</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.986956</td>
<td>0.857826</td>
<td>0.869679</td>
</tr>
<tr>
<td>PKM</td>
<td>10</td>
<td>0.928487</td>
<td>0.63301</td>
<td>0.669153</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.952012</td>
<td>0.777648</td>
<td>0.838368</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>0.969863</td>
<td>0.835889</td>
<td>0.899005</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>0.970923</td>
<td>0.738492</td>
<td>0.809792</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>0.949675</td>
<td>0.786128</td>
<td>0.877659</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>0.965916</td>
<td>0.832429</td>
<td>0.872869</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>0.961859</td>
<td>0.680689</td>
<td>0.555197</td>
</tr>
</tbody>
</table>

Table 4.9: $F$-Score of all methods for different overlap region-size (IRIS)
region increases.

![Graph](image)

(a) Accuracy of the three methods for different overlapped region size (Number of seeds = 5)

![Graph](image)

(b) F Score of the three methods for different overlapped region size (Number of seeds = 5)

Figure 4.10: Performance of three methods on IRIS

As a summary, the seeds are critical for the clustering of the instances in the overlapped region and it is crucial to allow the seeds or constraints from the overlapped region to be associated with multiple cluster labels. This is indeed the advantage of our method over other non-fuzzy algorithms, or other membership score based fuzzy algorithms which are not able to specify the membership scores for the seeds or constraints.
4.4.4 Data Simulated from Real World Scenario

Experimental Setup

The data set used in this experiment represent attributes from Career Profile. This data has 15 attributes, which describes learner’s career interests and competencies. There are 5 outcome classes (C1(SS/AD)= software system/application developer, C2(SA)=security analyst, C3(CSA)=computer system analyst, C4(NA)=network architect, and C5(DA)=data analyst). Based on a real world scenario, we assume there is some overlap between C1 and C5, as well as C2 and C4, namely there is one overlapped region between C1 and C5, one overlapped region between C2 and C4. Each individual has 15 binary features and each feature has a probability in [0,1]. Therefore, we have a probability table of 15 x 5 (Table 4.10).

The values in the table show the %-based probability of Career Profile features. For example, when an individual is in Cluster “Security Analyst”, he has 80% probability to have a career interest in “Networks”. Here we consider probability 0 as “not possible”, probability 20% as “not likely”, probability 50% as “random”, probability 80% as very likely, probability 1 as “must be”.
### Table 4.10: Career Profile %-based probability of attributes

<table>
<thead>
<tr>
<th>Category</th>
<th>Attribute</th>
<th>SS/AD</th>
<th>SA</th>
<th>CSA</th>
<th>NA</th>
<th>DA</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>C1</td>
<td>C2</td>
<td>C3</td>
<td>C4</td>
<td>C5</td>
</tr>
<tr>
<td>Career Interests</td>
<td>Artificial Intelligence</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Software Engineering</td>
<td>80</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>System Architecture</td>
<td>50</td>
<td>20</td>
<td>1</td>
<td>80</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Hardware Architecture</td>
<td>20</td>
<td>80</td>
<td>1</td>
<td>50</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Enterprise Architecture</td>
<td>50</td>
<td>50</td>
<td>80</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Network</td>
<td>20</td>
<td>80</td>
<td>50</td>
<td>20</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Security</td>
<td>20</td>
<td>1</td>
<td>50</td>
<td>80</td>
<td>0</td>
</tr>
<tr>
<td>Competencies</td>
<td>CompTIAA+</td>
<td>50</td>
<td>1</td>
<td>50</td>
<td>1</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Cisco Certified Technician</td>
<td>20</td>
<td>50</td>
<td>80</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Data Management</td>
<td>20</td>
<td>0</td>
<td>50</td>
<td>20</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Web Design</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Java Programming</td>
<td>1</td>
<td>50</td>
<td>20</td>
<td>0</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>Operating System</td>
<td>20</td>
<td>80</td>
<td>80</td>
<td>50</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>Business Intelligence</td>
<td>80</td>
<td>20</td>
<td>20</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Data Modeling</td>
<td>80</td>
<td>20</td>
<td>50</td>
<td>20</td>
<td>1</td>
</tr>
</tbody>
</table>

Then the data set is simulated using the above Career Profile probability table to mimic the proposed real world scenario. For each cluster, we simulate $n1$ instances and the feature values are generated using the probability table. For overlapped region, we re-calculate the probability of each feature to simulate the instances in the overlapped region.

For example, as C1 overlaps with C5, the probability of "Artificial Intelligence" with value 1 is $(20,$ the probability of “Networks” with value 1 is $(20,$ etc.).

### Experimental Parameters

To simulate the data, we consider the following parameters: number of seeds in each cluster $s1$; number of link-constraints $l$; number of seeds in each overlapped region $s2$; size of each cluster $n1$ (excluding non-overlapped region); size of overlapped region $n2$. In each run of our simulation, we generate 100 instances within each cluster (excluding non-overlapped region), and 40 instances within the 2 overlapped region respectively. The number of seeds of each cluster (excluding non-overlapped region) is 20. For our experiment, we used the following parameters: $n1 = 100, l = 30, n2 = 40, s1 = 20$, then we vary $s2$ as 15, 20, 25,
30. For the 30 link-constraints, and we generate 15 must-link and cannot-link constraint respectively. Notice the seeds and constraints are randomly sampled and they do not overlap with each other.

Results and Discussion

The performance of the three algorithms for different $s_2$ is presented in Table ($s_2 = 15$). Table ($s_2 = 20$). Table ($s_2 = 25$). Table ($s_2 = 30$). From these tables, we can see that both precision and recall of the fuzzy algorithm increase as the number of seeds in the overlapped region increases. However, when the number of seeds is too large in the overlapped region, the performance of the fuzzy algorithm drops. This again indicates that the number of seeds does affect the performance of the fuzzy algorithm significantly. We can also see that in most of the cases both the precision and recall of our fuzzy algorithm is better that those of the baseline methods, which then lead to a better F-score for our fuzzy algorithm over the baseline methods. This indicates the importance of allowing fuzzy assignment when the clusters overlap.

<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPKM</td>
<td>Precision</td>
<td>0.927112</td>
<td>0.825615</td>
<td>0.909562</td>
<td>0.754854</td>
<td>0.923568</td>
</tr>
<tr>
<td>SKM</td>
<td>Precision</td>
<td>0.892830</td>
<td>0.856429</td>
<td>0.846753</td>
<td>0.796212</td>
<td>0.886683</td>
</tr>
<tr>
<td>PKM</td>
<td>Precision</td>
<td>0.895199</td>
<td>0.857443</td>
<td>0.868173</td>
<td>0.772398</td>
<td>0.875206</td>
</tr>
<tr>
<td>FPKM</td>
<td>Recall</td>
<td>0.952381</td>
<td>0.935714</td>
<td>0.873333</td>
<td>0.907143</td>
<td>0.971429</td>
</tr>
<tr>
<td>SKM</td>
<td>Recall</td>
<td>0.872219</td>
<td>0.800222</td>
<td>0.896667</td>
<td>0.813008</td>
<td>0.898613</td>
</tr>
<tr>
<td>PKM</td>
<td>Recall</td>
<td>0.858156</td>
<td>0.802654</td>
<td>0.88</td>
<td>0.832534</td>
<td>0.887470</td>
</tr>
<tr>
<td>FPKM</td>
<td>F-Score</td>
<td>0.934997</td>
<td>0.859307</td>
<td>0.896843</td>
<td>0.799343</td>
<td>0.938851</td>
</tr>
<tr>
<td>SKM</td>
<td>F-Score</td>
<td>0.884728</td>
<td>0.835440</td>
<td>0.861886</td>
<td>0.799683</td>
<td>0.889813</td>
</tr>
<tr>
<td>PKM</td>
<td>F-Score</td>
<td>0.881883</td>
<td>0.837065</td>
<td>0.871757</td>
<td>0.789989</td>
<td>0.878203</td>
</tr>
</tbody>
</table>

Table 4.11: Performance of all Methods (real scenario, $s_2 = 15$)
<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPKM</td>
<td>Precision</td>
<td>0.928489</td>
<td>0.825646</td>
<td>0.906245</td>
<td>0.781950</td>
<td>0.921130</td>
</tr>
<tr>
<td>SKM</td>
<td>Precision</td>
<td>0.921380</td>
<td>0.850769</td>
<td>0.843897</td>
<td>0.817566</td>
<td>0.901657</td>
</tr>
<tr>
<td>PKM</td>
<td>Precision</td>
<td>0.902325</td>
<td>0.858607</td>
<td>0.869662</td>
<td>0.792983</td>
<td>0.893434</td>
</tr>
<tr>
<td>FPKM</td>
<td>Recall</td>
<td>0.957143</td>
<td>0.940476</td>
<td>0.886666</td>
<td>0.907143</td>
<td>0.973809</td>
</tr>
<tr>
<td>SKM</td>
<td>Recall</td>
<td>0.868162</td>
<td>0.821765</td>
<td>0.92</td>
<td>0.809209</td>
<td>0.927021</td>
</tr>
<tr>
<td>PKM</td>
<td>Recall</td>
<td>0.857390</td>
<td>0.827151</td>
<td>0.89</td>
<td>0.828034</td>
<td>0.915767</td>
</tr>
<tr>
<td>FPKM</td>
<td>F-Score</td>
<td>0.937819</td>
<td>0.860331</td>
<td>0.898997</td>
<td>0.819388</td>
<td>0.938021</td>
</tr>
<tr>
<td>SKM</td>
<td>F-Score</td>
<td>0.902862</td>
<td>0.839813</td>
<td>0.867658</td>
<td>0.812805</td>
<td>0.909953</td>
</tr>
<tr>
<td>PKM</td>
<td>F-Score</td>
<td>0.886411</td>
<td>0.847304</td>
<td>0.876333</td>
<td>0.803550</td>
<td>0.900612</td>
</tr>
</tbody>
</table>

Table 4.12: Performance of all Methods (real scenario, s² = 20)

<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPKM</td>
<td>Precision</td>
<td>0.938953</td>
<td>0.844374</td>
<td>0.953310</td>
<td>0.787179</td>
<td>0.872916</td>
</tr>
<tr>
<td>SKM</td>
<td>Precision</td>
<td>0.899793</td>
<td>0.844600</td>
<td>0.833456</td>
<td>0.810675</td>
<td>0.892941</td>
</tr>
<tr>
<td>PKM</td>
<td>Precision</td>
<td>0.91215</td>
<td>0.840778</td>
<td>0.890582</td>
<td>0.771424</td>
<td>0.891261</td>
</tr>
<tr>
<td>FPKM</td>
<td>Recall</td>
<td>0.95</td>
<td>0.978571</td>
<td>0.876667</td>
<td>0.933333</td>
<td>0.969048</td>
</tr>
<tr>
<td>SKM</td>
<td>Recall</td>
<td>0.857538</td>
<td>0.805157</td>
<td>0.916666</td>
<td>0.803215</td>
<td>0.909752</td>
</tr>
<tr>
<td>PKM</td>
<td>Recall</td>
<td>0.871223</td>
<td>0.807393</td>
<td>0.88</td>
<td>0.834267</td>
<td>0.901276</td>
</tr>
<tr>
<td>FPKM</td>
<td>F-Score</td>
<td>0.942563</td>
<td>0.884680</td>
<td>0.926050</td>
<td>0.830476</td>
<td>0.902558</td>
</tr>
<tr>
<td>SKM</td>
<td>F-Score</td>
<td>0.885106</td>
<td>0.830910</td>
<td>0.859183</td>
<td>0.807985</td>
<td>0.898262</td>
</tr>
<tr>
<td>PKM</td>
<td>F-Score</td>
<td>0.897945</td>
<td>0.828711</td>
<td>0.886458</td>
<td>0.791262</td>
<td>0.894572</td>
</tr>
</tbody>
</table>

Table 4.13: Performance of all Methods (real scenario, s² = 25)

<table>
<thead>
<tr>
<th>Method</th>
<th>Metric</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>FPKM</td>
<td>Precision</td>
<td>0.937658</td>
<td>0.830683</td>
<td>0.901168</td>
<td>0.757886</td>
<td>0.901082</td>
</tr>
<tr>
<td>SKM</td>
<td>Precision</td>
<td>0.905261</td>
<td>0.853363</td>
<td>0.792420</td>
<td>0.788385</td>
<td>0.868901</td>
</tr>
<tr>
<td>PKM</td>
<td>Precision</td>
<td>0.895773</td>
<td>0.856707</td>
<td>0.825470</td>
<td>0.769128</td>
<td>0.853874</td>
</tr>
<tr>
<td>FPKM</td>
<td>Recall</td>
<td>0.961905</td>
<td>0.940476</td>
<td>0.86</td>
<td>0.921429</td>
<td>0.973810</td>
</tr>
<tr>
<td>SKM</td>
<td>Recall</td>
<td>0.842655</td>
<td>0.805813</td>
<td>0.893333</td>
<td>0.783248</td>
<td>0.888509</td>
</tr>
<tr>
<td>PKM</td>
<td>Recall</td>
<td>0.823500</td>
<td>0.789057</td>
<td>0.87</td>
<td>0.833815</td>
<td>0.880010</td>
</tr>
<tr>
<td>FPKM</td>
<td>F-Score</td>
<td>0.945549</td>
<td>0.863985</td>
<td>0.886017</td>
<td>0.805355</td>
<td>0.924027</td>
</tr>
<tr>
<td>SKM</td>
<td>F-Score</td>
<td>0.883221</td>
<td>0.836356</td>
<td>0.823267</td>
<td>0.785567</td>
<td>0.875328</td>
</tr>
<tr>
<td>PKM</td>
<td>F-Score</td>
<td>0.869649</td>
<td>0.832421</td>
<td>0.839465</td>
<td>0.789405</td>
<td>0.862221</td>
</tr>
</tbody>
</table>

Table 4.14: Performance of all Methods (real data, s² = 30)
Figure 4.11: Performance of all Methods on real scenario

In addition, we can see in Fig. 4.11b that the total F-score of our fuzzy method is better than those of the baseline methods, indicating again that allowing fuzzy assignment is critical. Also we observe that the total F-score in general increases when the number of
seeds increases, as we get more information on the overlapped region from the seeds. The performance drops when the number of seeds is too large. This might be because with too many seeds, we obtain more false positives in the overlapped region. We observed similar patterns for the total accuracy as well (see Fig.4.11a).

4.5 Summary

We devised a semi-supervised clustering method (FPKM) to bring learners with similar professional traits that match a typical career pattern together into the same cluster to construct a CoP. Our method aims to initially form the CoP with a seed set of learners who can drive the CoP activities and sustain its effectiveness. We emphasized the natural overlap nature of industrial needs and career paths by allowing each learner to be in more than one cluster. Using three types of data sets of different nature, we experimentally showed the improved performance of the proposed fuzzy pairwise-constraints clustering approach when the overlap degree increases, in comparison with baseline line methods of seeded and pairwise-constraints K-means algorithm. Hence, our method has the potential to serve as a learning predictive tool to reveal hidden patterns of common traits among learners viewed as future candidates of the job market. These patterns could evolve into social communities of learners with shared career interests, that evolve socially rather than individually. We present our work to contract Cop-Network in next chapter.
Chapter 5: Social Learning Analytics for Career Development

5.1 Introduction

Social Networks (SNs) drive new forms of collaborations and contacts, and provide a fruitful platform for social learning as well. In SNs, people develop social relationships or ties related to their domain of interests. These ties are leveraged for gaining access to new knowledge and learning opportunities [223]. The impact of online social networks on education has been addressed considerably in the literature [224, 225, 226, 227]. For higher education in particular, online social networking with peers and faculty presents a dynamic platform for gaining information and knowledge which influences students' learning outcomes and academic achievements [228, 229, 230]. Some studies reported that students' social networking behavior is positively associated with their academic success and grade performance [231, 232]. Furthermore, a link has been revealed between social networking and college students' social well-being [233, 234, 235, 236]. A comprehensive literature review and research directions pertaining to social networking in higher education has been presented in [31]. Moreover, SNs research has shown that having an extended social relationship is crucial for personal and professional development [237, 238, 239]. Individuals could gain advantage from their personal social networks to enhance their opportunity to influence entrepreneurs, to improve their job performance to achieve higher mobility and build career-related aspirations [240, 21]. In job business, newcomers can benefit from social networks to learn organizational and tasks knowledge; and to enhance their social integration [241, 242].

Social networks (SNs) are formally defined as a set of actors which are tied by one or more types of relations (e.g., friendship, partnership, etc.); and they are most often modeled as graphs. Relations are represented by the edges in the graph connecting the actors and may have a direction indicating the flow from one actor to the other; and a strength denoting how much, how often, or how important the relationship is. With the raising of online social network and its impact on collaborative learning, the topics of social learning
analytics and community detection become absolutely interesting research areas. Social
learning analytics (SLA) [12, 41] is a recent emerging field which focuses on the application of LA in social learning networks to investigate the underlying patterns of learning interactions and behaviors that signify effective learning processes. Recent studies on SLA have been focused on the discovery of relevant network structures and visualization of interactions in order to understand social learning; and provide new means to assess learners online participations [223, 243, 244]. Community is considered to be a significant property of social network structure as it often accounts for the functionality of the whole social networking system; and thus community detection is a valuable tool for social network analysis [245, 246]. Community detection aims to find groups of actors who have more connections to each other than to the rest of the network. Community detection is important in social learning networks as it contributes in identifying powerful individuals and initiating influence propagation in the network [247, 248].

The objective of this chapter is to construct a CoP-Network of a dense community structures. We introduce a Reciprocal-Weighted Euclidean (RWE) similarity function that considers order and weight assigned by learners to their attributes (career and social interests) and which are employed to match learners of high similarity. We explain the hierarchal representation of career profile to capture the order and weight of each single attribute. Learners of a match score that is above certain threshold are linked by a weighted edge (weight = matchscore) in the CoP-Network. We also devise a Triadic Closure approach (TC-CNM) for community detection enhancement in the generated CoP-Network.
5.2 Background and Related Work

5.2.1 Social Network

Social Network Structure

The idea of "social network" has roots in psychology and sociology where scientists investigated ways in which small community structures could influence individual perceptions and action choices [249]. Network structure was initially described by Moreno [250] who introduced the idea of representing social structure as a network diagram of points and lines, labelled as "socimetry". The individual who can establish links or contacts with other individuals in the network represent a "social atom" that is the smallest unit of the social structure in a community. Fig. 5.1 illustrates a generic "sociogram" (i.e. social digram) and density table of a network in which individuals are variably connected to one another as a function of prior contact, exchange, and attendant emotions. Individuals are dots, relations are lines while solid (dashed) lines connect pairs of individuals who have a strong (weak) relationship [251].

Network models use different structures to indicate how information flows in a system of individuals based on two socio-psychological facts [252, 251]: (a) individuals "cluster" themselves into groups or communities as a result of interaction opportunities defined by the places where they meet; and (b) communication is more frequent and influential between communities as individuals in the same group tend more likely to develop similar views. Individuals of the same group create a unique system of opinions and behaviors (relevant to the group) which ultimately defines what it means to be a member of a common group. Thus, they share implicit and explicit understanding constituting the knowledge attached to their cluster. In Fig.5.1, individuals are clustered into groups or communities that are marked by lines which density is greater within clusters than between clusters. Density of a group is defined by the percentage (ranging from 0 to 1) of aggregation of its individual members calculated based on the sum of direct contacts that each individual of the group
has activated by or received from others. For example, the diagram and the density table at the bottom of Fig.5.1 show three groups (A, B, C), and the generic pattern of in-group relations stronger than relations between groups (diagonal elements of the density table are higher than the off-diagonal ones, and each cell of the density table is the average of relationships between individuals within the row and individuals in the column).

<table>
<thead>
<tr>
<th></th>
<th>.65</th>
<th>.05</th>
<th>.00</th>
</tr>
</thead>
<tbody>
<tr>
<td>Group</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.00</td>
<td>.01</td>
<td>.65</td>
</tr>
<tr>
<td>Group</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>.00</td>
<td>.01</td>
<td>.65</td>
</tr>
<tr>
<td>Group</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 5.1: Social Network Structure

Social Network Mechanisms

As for network mechanisms, individuals can play either of two roles: (1) Closure: specialize within cluster (as James in Fig.5.1); or (2) Brokerage or structural hole: build bridges between clusters (as Robert in Fig.5.1). Closure is about strengthening the connections to gain advantage by getting better information about what is already known, while brokerage is about connecting across groups to engage diverse information[252, 251]. The two mechanisms made the foundation for “social capital” as they do not assume that networks replace information so much as affect the flow of information and what people can do with
Social capital refers to the advantages or resources that individuals or groups enjoy because of their position in a social structure. The underlying metaphor is that individuals with high social capital are those who perform better because they are better connected to other nodes. Certain individuals or certain groups are connected to certain others, trusting certain others, dependent on exchange with certain others, obligated to support certain others. Holding a certain position in the structure of these interactions can be an asset in its own and that asset is the social capital, or as defined by Bourdieu: "...social capital is the sum of the resources, actual or virtual, that accrue to an individual or group by virtue or possessing a durable network of more or less institutionalized relationships of mutual acquaintance and recognition" (129). By definition, social capital involves various features of social structure, such as trust, norms, and connections which all can improve the efficiency of society or group by facilitating the coordinated actions.

Both closure and brokerage mechanisms begin with the assumption that communication matures over time, so prior relationships affect who knows what early. Information can be expected to spread across the individuals in a network, but will circulate within groups before it circulates between groups. Networks with closure - operationally described as a dense network in which everyone is connected such that no one can escape the notice of others - are argued to be the source of social capital mainly because "closure" is of two benefits: (1) it affects the access to information; and (2) it facilitates restrictions that make it less risky for individuals in the network to trust one another. For example, strong relationships among James’ contacts give him more reliable communication channels than Robert, and so more social capital.

On the other hand, structural holes present the opportunities to broke the information flow between clusters, and control the metrics that bring individuals from opposite side of the hole. For example, Robert was advantaged three ways by his broker position: (1) access to less redundant and more broad information (information breath); (2) early access/know of information due the crossroads position in the flow of information between groups (information
formation timing) that in many occasions be the individual to deliver information from one group to another; and (3) play important role in groups' merge. Thus, Robert enjoys what is identified in diffusion research by "an opinion leader" who is responsible for the spread of new ideas and behaviors [254, 255].

5.2.2 Community Detection

Community Definition

A community within networks can be defined as subgraph of vertices which are more densely connected to each other than to the rest of the graph [256]. Let $G = (V, E)$ be a graph with adjacency matrix $(A_{i,j})$. The graph $G' = (V', E' = E|_{V'})$ is a subgraph of $G$. The degree $deg(i)$ of a vertex $i$ is expressed as:

$$deg(i) = \sum_j A_{i,j}.$$

The number of edges that connect vertex $i$ to the other vertices within the subgraph $G'$ is represented by:

$$deg_{in}^{G'}(i) = \sum_{j \in G'} A_{i,j}$$

and the number of edges which connect vertex $i$ to vertices which are not in $G'$ is represented by:

$$deg_{out}^{G'}(i) = \sum_{j \notin G'} A_{i,j}.$$

Thus, total edges for every vertex $i$ belongs $G'$ can be expressed by the equation:

$$deg^{G'}(i) = deg_{in}^{G'}(i) + deg_{out}^{G'}(i)$$

To formalize the density of a community[257], the subgraph $G'$ is a community of strong sense if the number of edges connecting vertex $i$ to vertices within $G'$ is greater than the number of edges connecting vertex $i$ to vertices outside $G'$:

$$deg_{in}^{G'}(i) > deg_{out}^{G'}(i), \forall i \in G'.$$
Instead of requiring each vertex of a community to have more connections within
the community than outside the community, the subgraph $G'$ is a community in a weak
sense if the sum of all vertex degrees within the community is larger than the sum of all
degrees with the rest of the graph:

$$\sum_{i \in G} \text{deg}_{G'}^\text{in}(i) > \sum_{i \in G} \text{deg}_{G'}^\text{out}(i).$$

Accordingly, every community in a strong sense is a community of weak sense as
well while the converse is not true[141].

Community can be also defined as a group of vertices which are similar to each
other. Similarity can be computed between each pair of vertices with respect to some ref-
ience property, local or global, no matter whether they are connected by an edge or not
[258]. Each vertex ends up in the community or cluster whose vertices are most similar to
it. Similarity measures are at the basis of traditional clustering methods, like hierarchical,
partitional and spectral clustering. For example, If the graph vertices are embedded= in
an n-dimensional Euclidean space, in given positions, then the distance between a pair of
vertices can be used as a measure of their similarity. If the graph cannot be embedded in
space, the similarity could be inferred from the adjacency relationships between vertices
(based on the concept of structural equivalence)[258, 259].

**Community Detection Algorithms**

Identifying communities and structural positions of vertices within communities contribute
to understand and then exploit the functions of control and influence that mediate and lead
relationships and exchanges among nodes. These nodes belong to a tight-knit community
and are more than likely to have many properties in common. For example, identifying
communities and powerful nodes in = popular social networks (e.g. Facebook and Twitter)
can be used to initiate influential marketing campaigns [260].

A variety of community detection algorithms have been proposed in recent years,
involving different types of networks and community structures optimization[246, 141].
Most of the proposed algorithms are intended to detect communities in un-directed and un-weighted networks as experimental networks may not have much information for vertices and edges themselves, such as attributes and weights. However, it is possible to enhance the community structure of a network with extra information (e.g. structure- or attribute-based weights) computed based on vertex similarity measures [261]. Next, we describe some community detection methods selected out of a huge number of existing algorithms reflecting recent research trends and applications:

**A) GN Algorithm**

A hierarchical divisive (top-down) algorithm of of Girvan and Newman [262], in which links are iteratively removed based on the value of their betweenness. Vertex betweenness represents a measure of the centrality and influence of a node over the flow of information between other nodes, especially in cases where information flow over a network primarily follows the shortest available path. The betweenness centrality of a vertex \( i \) is defined as the number of shortest paths between pairs of other vertices that run through \( i \). Thus, if a network contains communities that are only loosely connected by a few inter-group edges, then all shortest paths between different communities must go along one of these few edges which essentially have the highest betweenness. By removing these edges, the algorithm separates groups from one another and so reveal the underlying community structure of the graph. The GN algorithm runs in worst-case time \( O(m^2n) \), while \( m \) is the number of edges in a graph of \( n \) vertices. The authors also introduce a measure (Modularity) [256] for the strength of the community structure found by the algorithm. Modularity is a quality function that estimates the goodness of a partition based on the comparison between the graph at hand and a null model, which is a class of random graphs with the same expected degree sequence of the original graph. Further, modularity measure can be used as an objective metric for choosing the number of communities into which a network should be divided.

**B) CNM Algorithm**
Clauset, Newman and Moore developed a greedy modularity optimization algorithm (CNM) [245], which is a fast implementation of a technique proposed by Newman in [263] to simply optimize modularity over all possible divisions. CNM is a hierarchical agglomeration (bottom-up) algorithm, which starts from a set of isolated nodes and then communities are constructed by iteratively adding the links of the original graph to produce the largest possible increase of the modularity at each step. Fig. shows a graphical representation of divisive and agglomerative algorithms. CNM algorithm -as a candidate algorithm used in our experimental evaluation- is further described in Section . The fast version of CNM has a complexity of $O(n \log^2 n)$.

C) Self-contained GN algorithm

A divisive hierarchical method by Radicchi et al. [257] based on GN method, where links are iteratively removed according to the value of their edge clustering coefficient. The edge clustering coefficient is the ratio between triangles to which a given edge belongs and the number of triangles that might potentially include it, given the degrees of the adjacent nodes. More formally, the the edge-clustering coefficient is (when the edge connects node $i$ to node $j$):

$$C_{i,j}^{(3)} = \frac{z_{i,j}^{(3)}}{\min[(k_i - 1), (k_j - 1)]}$$

where $z_{i,j}^{(3)}$ is the number of triangles built on that edge and $\min[(k_i - 1), (k_j - 1)]$ is the maximum possible number of them. The edge clustering co-efficient is a local measure, where its computation cost is less expensive than the edge betweenness, yielding a significant improvement in the complexity of the algorithm, that is $O(n^2)$. Another major difference from GN algorithm is the stopping criterion of the procedure, which depends on the properties of the communities themselves, not on the values of a quality function (e.g. modularity).

D) Guillaume

A multistep technique based on a local optimization of modularity in the neighborhood of each node [264]. The algorithm considers only the local properties of the graph
during the maximization process to deliver a hierarchical tree of communities. It is divided into two phases that are repeated iteratively. Starting with a weighted network of \( n \) nodes, each node of the network is assigned to a different community and so, there are as many communities as there are nodes. Then, for each node \( i \), the algorithm considers the neighbors \( j \) of \( i \) and evaluates the gain of modularity that would take place by removing \( i \) from its community and by placing it in the community of \( j \). The node \( i \) is then placed in the community for which this gain is maximum (a breaking rule is used in case of tie), but only if this gain is positive. If no positive gain is possible, \( i \) stays in its original community. This process (see Fig. 5.2) is applied repeatedly and sequentially for all nodes until no further improvement can be achieved. The computational complexity is linear along the number of links in the graph.

Figure 5.2: Visualization of the steps of Guillaume algorithm

**E) Clique Percolation Method (CPM)**

A method introduced in [265] detects overlapping communities based on the concept that internal edges of a community are likely to form cliques while inter-community edges are unlikely to form cliques (see Fig.5.3). The algorithm finds \( k \)-cliques percolation cluster (or \( k \)-clique community) that is a maximal \( k \)-clique-connected subgraph (i.e.,
the union of all \( k \)-cliques that can be reached from each other through a series of adjacent \( k \)-cliques. A clique is a complete graph and so a \( k \)-clique is a subgraph of \( k \) vertices are all connected with each other. Two \( k \)-cliques are \( k \)-clique adjacent if they have \( k - 1 \) vertices in common. The combination of a sequence of adjacent \( k \)-cliques is called a \( k \)-clique chain; and the \( k \)-cliques which are members of a \( k \)-clique chain are \( k \)-clique-connected.In the first step any \( k \)-clique of the original graph can be selected. In the second step this \( k \)-clique is rolled to an adjacent \( k \)-clique by changing only one vertex and keeping the other \( k - 1 \) vertices as before. Then this \( k \)-clique itself is rolled to another adjacent \( k \)-clique. If there is no possibility left to reach unvisited vertices by rolling through adjacent \( k \)-cliques a \( k \)-clique percolation cluster is found.

Figure 5.3: Sketches of two graphs (\( N=20 \)). The left one is below the the 3-clique (triangle) percolation threshold and so only two small 3-clique percolation clusters (distinguished by black and dark gray edges) can be observed. In the right graph, most 3-cliques accumulate in a “giant” 3-clique percolation cluster (black edges).

An extension of CPM was developed, the clique percolation method with weights CPMw [266] that allows detecting overlapping communities in weighted graphs. The running time of CPM is non-polynomial while the running time of CPMw is exponential. The major disadvantages of CPM and CPMw is that optimal value of \( k \) is not known in advance; and if the input graph does not consist of many \( k \)-cliques, then both algorithms are not applicable. Further, the results of the two methods differs strongly if edges with a high weight tend to have neighbors with a small weight (see Fig.5.4)
F) Fortunato

An algorithm introduced in [259] to detect both overlapping communities and a hierarchical structure. Instead of optimizing modularity, within Fortunato a fitness function is optimized \( (f_{Ci}) \) with underlying assumption that communities are essentially local structures that involve the nodes belonging to the communities themselves plus at most an extended neighborhood of them. Thus, a community is a subgraph identified by the maximization of a property or fitness function of its vertices. The fitness function of community \( C_i \in C \) can be expressed as follows:

\[
f_{Ci} = \frac{\text{deg}_{in}^{C_i}}{\left(\text{deg}_{in}^{C_i} + \text{deg}_{out}^{C_i}\right)^\alpha}
\]

Where \( \text{deg}_{in}^{C_i} \) and \( \text{deg}_{out}^{C_i} \) are the total internal and external degrees of the nodes in \( C_i \) and \( \alpha \) is a positive real-valued parameter which controls the size of the communities. The aim is to determine the natural community of the node \( a \), such that the inclusion of a new node, or the elimination of one node from the subgraph would lower \( f_{Ci} \) (see Fig. 5.5). This amounts to determine local maxima for the fitness function for a given \( \alpha \) which forms the natural community of the node. The true maximum for each node trivially corresponds to the whole network, because in this case \( \text{deg}_{out}^{C_i} = 0 \) and \( f_{Ci} \) reaches the largest value that can be attained for a given \( \alpha \). Given \( f_{Ci} \), the fitness of a node \( a \) with respect to subgraph \( C_i \), is defined as the variation of the fitness of subgraph \( C_i \) with and without node \( a \), i.e.
The fitness \( f_{C_i+\{a\}} \) describes the fitness of the subgraph \( C_i \), which is merged with the node \( a \), whereas the fitness \( f_{C_i-\{a\}} \) is the fitness of the subgraph \( C_i \) without the node \( a \). With each iteration performed the same manner, the natural community of node \( a \) is detected through the following steps starting with \( C_i \) contains only node \( a \) (that means \( \text{deg}_{in}^{C_i}= 0 \)): (1) for every neighbor node \( b \) of \( C_i \), which is not part of \( C_i \), the fitness \( f_{C_i}^{b} \) is calculated, (2) the neighbor node, which delivers the largest fitness is added to \( C_i \), (3) the fitness of each node in \( C_i' = C_i \cup \{w\} \) is recalculated, and (4) if the fitness of a node is negative, it is removed from \( C_i' \) and a new subgraph \( C'' \) is obtained. If Step (4) occurs the procedure is repeated from Step (3), otherwise it is repeated from Step (1) for \( C'' \). The whole procedure stops when all neighbor \( i \) nodes in Step (1) have negative fitness. The complexity of Fortunato algorithm depends on the size of the detected communities and the extent of the overlaps. These characteristics are strongly influenced by the structure of the graph and the value of \( \alpha \). The worst-case complexity amounts to \( O(n^2) \) for any single \( \alpha \), where \( n \) is the number of nodes, this means the size of the communities is about \( n^{[141]} \).

Figure 5.5: Schematic example of natural community for a node (sky-blue point in the figure). The blue nodes are the other members of the group and have positive fitness within the group, while the red nodes have all negative fitness with respect to the group.
In Table 5.1, an overview of community detection algorithms is given. In conclusion, the running time of the algorithm is especially important if large graphs are analyzed, and that detecting hierarchical structures is desirable since the granularity can play an important role. In light of this review, we adopt CNM as the baseline method for our research work in enhancing community detection on CoP-Network.

### 5.2.3 Social Learning Analytics (SLA)

Social Learning Analytics (SLA) is a distinctive subset of LA, which highlights the social perspective of learning. SLA draws on the significant educational research work evidencing that new skills and ideas are developed and passed on through interactions and collaboration, and that learning cannot be understood without reference to context. As a group of learners engaged in a joint activity, their success is related to a combination of individual knowledge and skills, environment, use of tools, and ability to work together [267, 39].

SLA develops potential to make use of data generated by learners' traces through online activities in order to identify behaviors within learning environments that indicate their learning performance. A good discussion of different drives behind the emergence of SLA is provided in [41], concluding that LA in general must be reframed to place a special focus on online social interaction and social construction of knowledge. The literature identifies five SLA approaches (summarized in Tables 5.2 and 5.3, the first two approaches are inherently social (i.e., make sense in a collective analytics) while the other three can be socialized (i.e., use personal analytics that have important attributes in a collective context) [41, 40].

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Weighted Graph</th>
<th>Allow Overlap</th>
<th>Hierarchy is detected</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>GN</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>$O(m^2n)$</td>
</tr>
<tr>
<td>CNM</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>$O(n\log^2n)$</td>
</tr>
<tr>
<td><strong>Self-contained GN</strong></td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>$O(n^2)$</td>
</tr>
<tr>
<td>Guillaume</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>CPM</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td><strong>Exponential</strong></td>
</tr>
<tr>
<td>CPMw</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td><strong>Exponential</strong></td>
</tr>
<tr>
<td>Fortunato</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>$O(n^2)$</td>
</tr>
</tbody>
</table>

Table 5.1: Features of community detection algorithms.
In this chapter, we mainly focus on social learning analysis (SNA) to be applied as an example of SLA on CoP-Network.

**Social Network Analysis (SNA)**

Social Networks are modeled as a graph $G$ which is defined as a pair of sets $G = (V, E)$, where $V$ is a set of $N$ nodes $V = \{v_1, v_2, \ldots, v_n\}$ and $E \subseteq V \times V$ is a set of edges that connect pairs of nodes $v_i, v_j$ within $V$. In other words, $V \times V$ is an adjacency matrix $E = [E_{ij}]_{i, j \in V}$, where $E_{ij} \in \{0, 1\}$ represents the availability of an edge from node $i$ to node $j$. The edge weight $E_{ij} > 0$ represents the intensity of interaction and the graph $G(V, E)$ in that case is called weighted graph. The graph is directed if $E_{ij} \neq E_{ji}$ and undirected if $E_{ij} = E_{ji}$ for all $i, j \in V$. The graph theory then provides a set of abstract concepts and methods for the analysis of graphs using graph-based measures which, in combination with other analytical tools developed specifically for the visualization and analysis of social networks, form the basis of structural analysis or what is referred to as social network analysis (SNA) [268].

SNA, in general, focuses on the characteristics of ties connecting individuals in a network rather than on the characteristics of the individuals themselves; and views personal communities as networks of individual relations that people foster, maintain, and use in the course of their daily lives [269]. It also evaluates how structural regularities influence individuals’ behaviors and actions [268]. There are two main forms of SNA: the ego network analysis, and the whole network analysis. In ‘ego’ network analysis, the network of one individual (ego) and his connections (alter) is studied using individual elements metrics; while in whole network analysis, SNA tries to find all relations between the participants in the network using network level metrics (see Fig.5.6).
<table>
<thead>
<tr>
<th>Type</th>
<th>Base</th>
<th>Description</th>
<th>Tools</th>
<th>Potentials in the context of social learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social Networks Analysis (SNA)</td>
<td>Interpersonal relationships define social platforms</td>
<td>Investigates networking process and properties of ties, relationships, roles and network formation and how people develop and maintain these relationships to support learning. Involves egocentric network and whole network analysis.</td>
<td>Mzinga, SNAPP, Gephi</td>
<td>Define what counts as learning ties. Defines which interactions promote learning. Accounts for interaction with resources (objects of knowledge) to identify &amp; strength indirect relationships between people.</td>
</tr>
<tr>
<td>Discourse Analytics</td>
<td>Language is a primary tool for knowledge negotiation and construction</td>
<td>Analysis of series of communicative events. Three social modes of thinking: Disputational: disagreement and individual decisions; Cumulative: speakers build on the contributions of others without critiquing or challenging them; Exploratory: speakers explaining their reasoning, challenging ideas, evaluating evidence and developing understanding together.</td>
<td>Wordle, TagCrowd, NVivo, Cohere</td>
<td>Identify learner’s attention, rhetorical attitude to discourse contribution, distribution of learning topics and learners relationships. Help learners to develop their conversations into reasoned arguments and educational dialogue. Employs a structured deliberation/argument mapping platform to study what learners are paying attention to, what they focus on, which viewpoints they take up, how learning topics are distributed amongst participants, how learners are linked by semantic relationships such as support and challenge, and how learners react to different ideas and contributions.</td>
</tr>
</tbody>
</table>

Table 5.2: Five Approaches of Social Learning Analytics A (SLA)
<table>
<thead>
<tr>
<th>Type</th>
<th>Base</th>
<th>Description</th>
<th>Tools</th>
<th>Potentials in the context of social learning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning dispositions</td>
<td>Intrinsic motivation to learn is a defining feature of online social media, and lies at the heart of engaged learning, and innovation</td>
<td>Qualities that make up the individual capacity for lifelong learning.</td>
<td>ELLI</td>
<td>Draw learners’ attention to the importance of relationships and interdependence</td>
</tr>
<tr>
<td>Analytics</td>
<td></td>
<td></td>
<td></td>
<td>Encourage learners to reflect on their ways of perceiving, processing and reacting to learning interactions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Support mentors ability to engage group of learners in meaningful and engaging education.</td>
</tr>
<tr>
<td>Content Analytics</td>
<td>User-generated content is one of the defining characteristics of Web 2.0</td>
<td>Examines, indexes and filters online media assets to guide learners through available resources</td>
<td>Web-based search engine</td>
<td>Analyze semantic relations and provide feedback for content authors to improve their materials.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Support the effective distribution of key resources through learning network.</td>
</tr>
<tr>
<td>Context Analytics</td>
<td>Seeks to understand contexts in relation to learning (Individuality context, Time context, Location context, Activity context, and Relations context)</td>
<td>Seeks to understand contexts in relation to learning</td>
<td>MOBlearn, Active Campus</td>
<td>Provide context-focused recommendations</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Highlight the activity of other learners in a community or network, through tag clouds, hash tags, data visualizations, activity streams and emergent folksonomies.</td>
</tr>
</tbody>
</table>

Table 5.3: Five Approaches of Social Learning Analytics B (SLA)
There are several SNA indicators in the literature that describe both the individual and network levels [268, 270, 271]. Here we describe some of the elementary indicators.

- **Density**

The density (D) is a measure of the general level of connectedness of the graph. If node i is directly connected to every other node, we have a complete graph. The density of a graph is defined as a ratio of the number of edges E to the number of possible edges. For an undirected graph $G$ with $N$ nodes, the density $D$ is defined:

$$D = \frac{2E}{N(N-1)}$$

A directed graph will have half the density of its undirected equivalent, because there are twice as many possible edges (see Fig. 5.7). A perfectly connected network is called a clique and has $D = 1$. 

Figure 5.6: Ego networks and “whole” networks
• Coreness (k-core)

The degree-based k-cores or coreness is a simple tool for identifying well-connected structures within a graph. The core number \( k \) of vertex \( v \) is the value of the highest-value core containing \( v \). Then, vertices with high core numbers belong to relatively well-connected sets. The k-cores can be thought of as unions of relatively cohesive subgroups. As k-cores are nested, each k-core can be think of as representing a "slice" through a hypothetical "cohesion surface" on \( G \) (see Fig. 5.8). For a given undirected graph \( G = (V, E) \), where \( V \) is the vertices set and \( E \) is the edges set:

A subgraph \( H = (C, E|C) \) induced by the set \( C \subseteq V \) is a k-core or a core of order \( k \) iff \( \forall v \in C : \text{degree}_H(v) \geq k \), and \( H \) is the maximum subgraph with this property. Then, a minimal degree \( k \) is imposed to the core of order \( k \).

![Figure 5.8: K-core decomposition](image)

• Centrality

The most popular centrality measures are: degree centrality, betweenness centrality, and closeness centrality (see Fig. 5.9).
a) **Degree centrality** of a node is defined as the number of ties this node has (the number of edges adjacent to this node). Degree centrality, \( d(i) \), of node \( i \) is defined as:

\[
d(i) = \sum_j m_{ij}
\]

Where \( m_{ij} = 1 \) if there is a link between nodes \( i \) and \( j \), and \( m_{ij} = 0 \) if there is no such link.

b) **Closeness centrality** of a node is equal to the total distance (in the graph) of this node from all other nodes. Closeness centrality, \( c(i) \), of node \( i \) can be expressed as:

\[
c(i) = \sum_i d_{ij}
\]

where \( d_{ij} \) is the number of links in a shortest path from node \( i \) to node \( j \).

c) **Betweenness centrality** is defined as the number of times a node needs a given node to reach another node; or the number of shortest paths that pass through a given node. The betweenness centrality \( b(i) \) of node \( i \) is defined as:

\[
b(i) = \sum_{j,k} \frac{g_{jk} - g_{jik}}{g_{jk}}
\]

where \( g_{jk} \) is the number of shortest paths from node \( j \) to node \( k \), and \( g_{jik} \) is the number of shortest paths from node \( j \) to node \( k \) passing through node \( i \).
Nodes 3 and 5 have the highest degree centrality (4)

(b) Closeness centrality

Node 5 has higher betweenness centrality than 3

(c) Betweenness centrality

Figure 5.9: Centrality Measures

Example of how these measures are used to identify the set of key players in a given network is illustrated in Fig. 5.10. According to degree centrality, node 10 is the most central, but nodes 3 and 5 together will reach more nodes according to their betweenness and closeness degree. Moreover the tie between them is critical: if severed, the network will break into two isolated sub-networks. Thus, with other things being equal, players 3 and 5 together are more ‘key’ to this network than 10.
Players 3 and 5 together are more 'key' to this network than 10 of higher centrality degree

Figure 5.10: Identifying set of key players

• **Tie Strength**

Adding weights to the edges in a graph network structure represents the relationship strength that may reflect the frequency of interactions, the amount of flow, the degree of similarities or other attributes of the nodes or ties. As Fig. 5.11 illustrates, there are three main concepts of tie strength: homophily, transitivity and bridges.

1. Homophily is the tendency to relate to people with similar characteristics (status, beliefs, etc.) It leads to the formation of homogeneous groups (clusters or communities) where forming relations is easier. Homophilous ties can be strong or weak. As extreme homogenization can act counter to innovation and idea generation, heterophily is desirable in some contexts.

2. Transitivity applies that if there is a tie between A and B and one between B and C, then in a transitive network A and C will also be connected. Strong ties are more often transitive than weak ties; transitivity is therefore evidence for the existence of strong ties (but not a necessary or sufficient condition). Transitivity and homophily together lead to the formation of cliques (fully connected clusters).

3. Bridges are nodes and edges that connect across groups. Bridges facilitate inter-
group communication and increase social cohesion. They are usually weak ties, but not every weak tie is a bridge.

In this essence, SNA facilitates: (1) identification of individuals’ social circles; (2) identification of individuals and groups playing central roles; and (3) identification of isolated individuals and groups and information flow bottlenecks. Analysis output could be then used to detect communities within networks in order to improve and accelerate information and knowledge flows across the network; and to enhance information exchange for different strategical purposes (i.e. learning, business, marketing) [272, 273, 274, 41, 244].

5.3 CoP-Network

5.3.1 Distance and Similarity

A distance space \((X, d)\) is a set \(X\) equipped with a distance \(d\). A function \(d : X \times X \rightarrow R\), where \(R\) is the set of non-negative real numbers, is called a distance (or dissimilarity) on \(X\).
if, for all $x, y \in X$, it holds [275]:

1. $d(x, y) \geq 0$ (nonnegativity);
2. $d(x, y) = d(y, x)$ (symmetry);
3. $d(x, x) = 0$.

A distance with $d(x, y) = 0$ implies that $x = y$, whereby $x$ and $y$ are deemed symmetric.

Let $X$ be a set. A function $s : X \times X \rightarrow R$ is called a similarity on $X$ if $s$ is nonnegative, symmetric and the inequality: $s(x, y) \leq s(x, x)$ holds for all $x, y \in X$, with equality if and only if $x = y$. The main transforms used to obtain a distance (dissimilarity) $d$ from a similarity $s$ bounded by 1 from above are:

$$d = 1 - s \text{ or normalized alternatives such as } d = \sqrt{1 - s} \text{ or } d = \sqrt{2(1 - s^2)}$$

The choice of distance/similarity measures depends on the measurement type or representation of objects [276]. The discrete probability function, $p(x)$, and the the continuous probability function, $f(x)$, [277] are the most popular pattern representations [276]. These are illustrated next.

Let $X$ be a set of $n$ elements which possible values are discrete (not necessarily finite), $p(x)$ is a function that satisfies the following properties:

1. The probability that $x$ can take a specific value is $p(x)$ that is $P[X = x] = p(x) = px$
2. $p(x)$ is non-negative for all real $x$.
3. The sum of $p(x)$ over all possible values of $x$ is 1, that is $\sum_j p(j) = 1$ where $j$ represents all possible values that $x$ can have and $p(j)$ is the probability at $x_j$. One consequence of properties 2 and 3 is that $0 \leq p(x) \leq 1$.

While a continuous probability function, $f(x)$, is a function that satisfies the following properties:

1. The probability that $x$ is between two points $a$ and $b$ is $p[a \leq x \leq b] = \int_a^b f(x)dx$. 


2. It is non-negative for all real \( x \).

3. The integral of the probability function is one, that is:
\[
\int_{-\infty}^{\infty} f(x) \, dx = 1
\]

Discrete probability functions are referred to as probability mass functions and continuous probability functions are referred to as probability density functions (see Fig. 5.12). When referring to probability functions in generic terms, the term probability density functions (PDF) is used to describe both functions. The purpose of distance/similarity is to compare two lists of numbers (i.e. vectors), and compute a single number which evaluates their similarity. There are various distance/similarity measures that are applicable to compare two PDFs. One of the most popular methods is to consider PDF as a vector (i.e. a point in the Euclidean space) and apply geometrical distances to compare two PDFs. A comprehensive survey on distance/similarity measures between PDFs can be found in [278, 275]. In our research work, we apply the Euclidean distance/similarity measure, discussed next.
Euclidean Similarity

Euclidean similarity is the basis of many distance/similarity. The Euclidean distance between n-dimensional vectors $X$ and $Y$ is the square root of the sum of squared differences between corresponding elements of the two vectors, defined as follows:

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Euclidean distance is mostly appropriate for data measured on the same scale and thus it is most often used to compare profiles of respondents across variables used in our Career Profile model. For example, considering our data that consists of career interests information on a sample of individuals, arranged as a respondent-by-variable matrix. Each row of the matrix is a vector of $n$ numbers, where $n$ is the number of variables. A common similarity $s$ (or, the distance $d$) is evaluated between any pair of rows to reflect similar career aspirations.

Hybrid Weighted Euclidean Similarity Function

The weighted euclidean distance ($WED$) and the ordered weighted euclidean distance ($OWD$) are the main two types of distance measures used in existing literature to reflect the importance of each difference value of vector elements under consideration [279, 280].

$WED$ is defined as:

$$WED(x, y) = \sqrt{\sum_{i=1}^{n} w_i (x_i - y_i)^2},$$

Where $w_i$ represents the importance of each argument (i.e. absolute difference values $|x_i - y_i|$).

On the other hand, the ordered weighted distance ($OWD$) emphasizes the order or position of each argument, as defined next:

$$OWD(x, y) = \sqrt{\sum_{i=1}^{n} v_i (x_{\sigma(i)} - y_{\sigma(i)})^2}$$

where $\sigma(1), \sigma(2), \ldots, \sigma(n)$ is any permutation of $(1, 2, \ldots, n)$, such that:
\[|x_{\sigma(i-1)} - y_{\sigma(i-1)}| \geq |x_{\sigma(i)} - y_{\sigma(i)}|, \quad i = 2, 3, \ldots, n\]

and \(v = (v_1, v_2, \ldots, v_n)\) is the weighting vector associated with the \(OWD\) measures (i.e., the weighting vector of the ordered positions of the arguments \(|x_i - y_i|\)); such that \(v_i \geq 0\) and \(\sum^n_i v_i = 1\).

To reflect the importance of both the value and the position order of an argument, a hybrid weighted distance (\(HDW\)) [281] is developed as follows:

\[
HDW(x, y) = \sqrt{\sum^n_i v_i \Delta(x_{\sigma(i)} - y_{\sigma(i)})^2}
\]

where \(\Delta(x_{\sigma(i)} - y_{\sigma(i)})\) is the \(i\)th largest of the weighted arguments \(\Delta(x_i - y_i) = \max(|x_i - y_i|^2, i = 1, 2, \ldots, n)\). \(v = (v_1, v_1, \ldots, v_n)\) is the weighting vector associated with the \(HWD\) measures, \(v_i \geq 0\), \(\sum^n_i v_i = 1\). \((w_1, w_2, \ldots, w_n)\) is the weight vector of the arguments \(|x_i - y_i|\), \(w_i \geq 0\), \(\sum^n_i w_i = 1\), and \(n\) is the balancing coefficient such that:

- If \(v = (1/n, 1/n, \ldots, 1/n)\), then \(HWD\) is reduced to the weighted distance (\(WD\))
- If \(w = (1/n, 1/n, \ldots, 1/n)\), then \(HWD\) is reduced to \(OWD\)

### 5.3.2 Reciprocal-Weighted Euclidean Similarity

We propose a reciprocal-weighted euclidean similarity function (\(RWD\)), which is inspired by the hybrid weighted euclidean function and existing research works on matching user profiles of a network [282, 283], to consider self-assigned weights and position orders to profile attributes, when determining similarity measure across profiles. Let \(L\) and \(U\) be two learner profiles represented by \(n\)-dimensional attribute vector, \(L = (l_1, l_2, \ldots, l_n)\) and \(U = (u_1, u_2, \ldots, u_n)\) depicting \(n\) pre-established measurements made associated with the learner from \(n\) attributes, respectively \(A_1, A_2, \ldots, A_n\) which represent common social interests. \(RWD\) calculates similarity or (the match score used later) of \(L\) and \(U\) as follows:

\[
RWD(L, U) = \sqrt{\sum^n_i rw_i(l_i - rw_i u_i)^2}
\]
where \( rw(l) = (w_1, w_2, \ldots, w_n) \) is the weighting vector assigned to learner vector \( L \); and \( rw(u) = (w_1, w_2, \ldots, w_n) \) is the weighting vector assigned to learner vector \( U \). The simplest and most convenient method to allow learners to assign order and weight to their attributes is to include it in their Career Profile then measure the attributes or interests of a given learner using RW. In such representation, interests may be associated with an interest level, such as a numeric or graded rating that represents the strength or weakness of the learner’s interest in a particular topic. Example of such representation is illustrated in Fig. 5.13. The learner provides a level of interest data in a hierarchical manner, scoring topics and subtopics as shown in Fig. 5.13 (In this figure, football is the highest interest at the sub-topic level of sports while history is the least interest in the sub-sub-topic level of entertainment. A vector could represent interests in [sports, football, UAE league, Brazil fashion, reading, history, ME history, fiction], and so a learner profile could be represented as: \([1, 2, 5, 8, 0, 3, 0, 4]\). This means that the learner is generally interested in sports (1), especially football (2) and most of all the UAE league (5), and least of all Brazil (8). has no opinion regarding fashion (0), is generally interested in reading (3), more into fiction (4) than history (5) and definitely not so interested in ME history (4).
Table 5.4: Vector Representation of Learners Interests in Fig.5.13

In order to obtain the reciprocal-weighted \((rw)\) interest accumulation, we calculate the ranked weight of each attribute as follows:

For example, a measure of learner’s interest for history may be determined as: \(1/2 \times 1 \times 1/2 = 0.250\) (it is ranked second at the first level \((r_1 = 2)\), ranked first at the second level \((r_2 = 1)\); and again ranked second at the third level \((r_3 = 2)\)). Thus, the representation of the measure of learner interests in the previous example may be converted into a vector representation as follows: \(L = (1, 1, 1, 0.33, 0.0, 0.50, 0.25, 0.25, 0.083, 0.50)\) (see Table 5.4). We may apply one or more thresholds, such as by considering anything below 0.05 (or another value) as too low and thus simply expressed as zero.

Our proposed Reciprocal-Weighted Interest Accumulation Networking (RWIAN) algorithm (see Algorithm 5) constructs a CoP-Network that employs the above profile structure to generate vector attributes of learners within a common CoP, resulting in an overall CoP network. This process calculates the social similarity using RWD function to generate the similarity matrix \((SM)\) in order to establish (weighted) links between learners with highest similarity. We use the Modularity \((Q)\) metric to evaluate the goodness of clustering learners into communities within the constructed network by applying the CNM community detection algorithm. Using our proposed RWIAN algorithm, we can generate several weighted networks \(G_w\) with different similarity thresholds, calculate their \(Qs\) and estimate the maximum \(Q\) associated with a given similarity threshold \(t\).

5.4 Triadic Closure to Enhance Community Detection

Transitivity of a relation means that when there is a tie between \(l\) and \(u\) in a network, and also between \(u\) and \(v\) nodes within the same network, then there is also a tie between \(l\) and \(v\). As discussed in the SNA subsection and illustrated in Fig. , strong ties that con-
Algorithm 5 Reciprocal-Weighted Interest Accumulation Networking Algorithm (RWIAN)

**Input:** A set of learners: \( L = (l_1, l_2, \ldots, l_n) \); \( t \): Similarity threshold

**Output:** \( SM \) (similarity matrix); \( G_w \): weighted directed graph where: nodes are users \( L \) & the edges weights are calculated based on \( n \)-attributes \( RWD \) similarity ; \( Q \): Modularity value

**Begin**

//Initialize the similarity matrix(\( MS \)) & weighted graph matrix (\( G_w \)) to zeros
\( SM = 0 \)
\( G_w = 0 \)

//For every \( l \) in \( L \) ; find the highest similar learners from \( L/l \) by calculating \( RWD \) match score (\( S \))
For every \( l, u \in L \)
   Calculate \( S(l, u) = RWD(l, u) \)
   \( SM(l, u) = S(l, u) \)
   If \( SM(l, u) \geq t \)
      Establish a link between learner \( l \) and \( u \); and assign a weight (\( w \)) to the link using \( S \)
      \( G_w(l, u) = s(l, u) \)
end for

Apply CNM algorithm on \( G_w \) and generate \( Q \)

End

Connect individuals with similar attributes are more often transitive than weak ties. Similarity and transitivity together lead to the formation of cliques (fully connected clusters), which enhance the network structure and community detection as internal edges of community likely to form cliques while inter-community edges unlikely to form cliques. Thus, we propose a triadic closure approach to enhance the constructed (RWIAN) social networks among learners. In our research work scope, we aim at introducing a systematic approach to build dense social networks in which learners can benefit from the power of collaborative knowledge towards achieving career common goals and objectives. Given as input, a graph \( G \), we propose to apply two different types of network enhancements: Structure based and Attribute based (see. Fig.5.14) enhancements. Structure based enhancement methods modify the network based on the existing structure, i.e. the edges of the network. On the other hand, Attribute based enhancements use socio-demographic features and interests of learners (represented by their nodes in the network) and use these properties to augment the network. Both of these enhancements increase edges between nodes to build tightly connect groups of learners in order to optimize collaborative learning opportunities.
and knowledge-exchange.

Figure 5.14: Overview of the proposed process to study network enhancements

A number of methods can be used to identify the structural similarity of nodes [284]. One of the most widely used network metric to calculate the structural similarity of two nodes is the Jaccard – Distance which uses the number of common neighbors as the principal driver to determine how similar two nodes are in the network [285]. Once, this is calculated, two nodes which are not connected in the network but may have high similarity can be connected to form a denser network (see Fig.5.15).

Figure 5.15: A new edge (shown in red) is created between nodes 3 and 4 as they have a high similarity. Similarity is calculated using Jaccard Index.
Another common structural enhancement could use the calculation of quasi-cliques (see Fig. 5.16). A Quasi-clique is a well-connected group of nodes but missing a few edges to form a complete clique. Edges can be added to these quasi-cliques to form tightly connected group of nodes leading to a denser network [286].

Figure 5.16: Two new edges (shown in red) are created between nodes 3-4 and 1-2 as the four nodes belong to a quasi-clique, missing only two edges.

While structure based methods are widely used in the literature to enhance community detection, we propose an attribute method that utilizes Career Profile records to optimize the performance of a group through connecting individuals who share common attributes in the profile such as social interests, and pursuing similar career goals and objectives. One way to achieve this is by using a Triadic-Closure approach based on Social interests and Career Interests. Triadic-closure refers to the property that if strong ties exist between nodes $I - u$ and nodes $u - v$, this can potentially lead to the formation of a new tie between $I - v$. Analysis of triad formation in networks has attracted tremendous interest from both academic and industrial communities studying how different patterns of triad formations influence the dynamic evolution of networks [287, 288].

However, since we filter the constructed social network by removing edges of similarity score below a given threshold ($S >= t$), and to avoid re-establishing these links, we propose to use a second similarity measure. This measure acts as a decisive metric to form the triadic tie if it holds, and only if $I$ and $v$ nodes have strong common interests that are different than the initial interests used to construct the network (see Fig. 5.17). For example, if we use Career Interests as the first similarity measure ($SI = Career - Interests$) to construct
the (RWIAN) network: $G_w = (V, E)$, where $V$ is a set of learners and $E \subseteq V \times V$ is a set of relationships connecting learners of highest $S1$. Three users $(l, u, v)$, where links between $l$ and $u$ ($e_{l,u} = 1$) and between $u$ and $v$ ($e_{u,v} = 1$) exist, form a candidate open triad $OT(l, u, v)$. We then use Social Interests as the second similarity measure ($S2 = Social - Interests$), to study whether the $OT(l, u, v)$ will (or not) become a closed triad $CT(l, u, v)$ to establish a link between $l$ and $v$ ($e_{l,v} = 1$). The strength of the newly established link $e_{l,v}$ is then a result of a combination of $S1$ and $S2$. A community detection algorithm (CNM) is applied to demonstrate that enhanced connected communities are indeed created as a result yielding improved modularity value ($Q$). The proposed Triadic Closure for enhanced community detection (TC_CNM) is depicted in Algorithm 6.

Figure 5.17: A new edge (shown in red) is created between nodes 3 - 4 as they have similar interests. The color encoding represents the interests of an individual.
**Algorithm 6** Triadic Closure for Enhanced Community Detection (TC_CNM)

**Input:** Graph $G$ constructed using $S1 = \text{Career - Interests}$; $S2 = \text{Social - Interests}$; Similarity Matrix: $SM$; Similarity Threshold: $t$

**Output:** Graph $G'$ with enhanced community structure

**Begin:**

For each node $n$ in $G$

\[ \text{neighbors} = \text{getNeighbors}(n, \text{distance} = 2) \]

// getNeighbors: gets all the neighbors of a node at given distance, which in this example is 2

For each $\text{neighborNode}$ in $\text{neighbors}$

\[ \text{if edgeExists}(n, \text{neighborNode}) = \text{False} \land SM[n, \text{neighborNode}] > t \]

\[ \text{createEdge}(G, n, \text{neighborNode}) \]

// edgeExists: returns a "True", if there is an edge between given nodes otherwise it returns "False"

// createEdge: creates a new edge between given nodes and adds it to Graph $G$

End For

End For

Apply CNM algorithm on $G'$ and generate $Q$

End

Further, in the attribute based network enhancements, more attributes (i.e. Career Goals) can also be a driving force to build new ties among learners of the same CoP showing high similarities with respect to the attributes set. New edges can be created among learners who are pursuing common career goals, for example, if two individuals are looking towards a teaching career, this common goal can motivate the formation of a new tie, which in turn can result in the formation of focused communities where individuals are pursuing common career goals (see Fig. 5.18).

![Figure 5.18: A new node is connected (shown in red) to all other nodes pursing similar career goals and becomes part of their community.](image-url)
5.5 Performance Evaluation

5.5.1 Experiment Setup

We generated a dataset \( L \) of a group of learners: \( L = \{ l_i \}_{i=1}^{N} \) while \( N \) is the size of \( L \). Each learner \( l_i \) is described by a vector of \( n \) attributes: \( l_i = \{ x_{i_1}, \ldots, x_{i_n} \} \) that represents his interests and is used to match with other likeminded learners. The value of each attribute \( x_{i_k} \) is a value within a given range: \( x_{i_k} \in [\min(x_{i_k}), \max(x_{i_k})] \) and that is generated following a normal distribution: \( x_{i_k} \sim N(\mu_{x_{i_k}}, \sigma_{x_{i_k}}) \) where \( \mu_{x_{i_k}} = 1/2(\min(x_{i_k}) + \max(x_{i_k})) \) and \( \sigma_{x_{i_k}} = 1/3(\max(x_{i_k}) - \mu_{x_{i_k}}) \) (about 99.7% are within three standard deviations).

Each attribute is assigned a weight \( w_{i_k} \) that is a value within a given range:

\[
w_{i_k} \in [\min(w_{i_k}), \max(w_{i_k})]
\]

and that is generated following a normal distribution: \( w_{i_k} \sim N(\mu_{w_{i_k}}, \sigma_{w_{i_k}}) \)

where \( \mu_{w_{i_k}} = 1/2(\min(w_{i_k}) + \max(w_{i_k})) \) and \( \sigma_{w_{i_k}} = 1/3(\max(w_{i_k}) - \mu_{w_{i_k}}) \). Hence, each learner is represented by a vector of weighted attributes, illustrated next where \( w_{i_k} \) and \( x_{i_k} \) are independent random values:

\[
l = w_{i_1}x_{i_1} + w_{i_2}x_{i_2} + \ldots + w_{i_n}x_{i_n} \text{ or } l = \sum_{i=1}^{n} w_{i_k}x_{i_k}
\]

To simplify the data generation process, we used a joint probability distribution function \( f_i(w_{i_k}, x_{i_k}) \) (c denoted \( f_{w_i x_i}(w_{i_k}, x_{i_k}) \)) to produce the weighted vector \( l \) where \( \mu_{w_{i_k}x_{i_k}} = \mu_{w_{i_k}}\mu_{x_{i_k}} \) and \( \sigma_{w_{i_k}x_{i_k}} = \sqrt{\sigma_{w_{i_k}}^2 + \sigma_{x_{i_k}}^2} \).

5.5.2 Experiment Parameters

Network Setup

In our experiment, \( N = 100 \) and \( n = 15 \), based on which we generate a dataset of 100 learners' vectors consisting of 15 attributes following a joint probability distribution func-
tion ($\mu = 0.53 : \sigma = 0.16$), to represent hobbies/social interests and sub-interests in areas such as sports, travel, and entertainment (see the example in Fig.). These vectors are used to generate a static network of similar profiles utilizing the $100 \times 100$ similarity matrix ($SM$) such that $l$ and $u$ are similar and thus linked in the network, only when $SM(l,u) = [S_{lu}] \geq$ similarity threshold ($\Delta$), where $S_{ij}$ indicates the similarity weight between $l_i$ and $l_j$.

Network Enhancement

In the network enhancement step, we created a sparse matrix of a specified size ($n \times n$). To generate a matrix of size $n \times n$, $n$ vectors each of length $n$ are initialized with input of minimum and maximum values. Once these vectors are generated, they are combined together to form a matrix. Further, we use two similarity measures ($S1, S2$). $S1$ is used to setup the initial network ($G$) through a set of attributes (e.g. career interests). $S2$ is further used to add edges to $G'$ using another set of attributes (for example: social interests) where these edges are enforced form triads in ($G$) to produce the enhanced network ($G'$).

5.5.3 Candidate Algorithms

CNM Algorithm

CNM community analysis algorithm is proposed by Clauset, Newman, and Moore [245] to infer community structure from network topology. It is basically a bottom-up agglomerative clustering method which works by greedily optimizing the modularity function that states the quality of graph partitioning or the quality of clustering. The goal of modularity optimization is to efficiently find a decomposition of $G$ into a set of $c$ modules $C_i$, $i = 1, 2, ..., c$ that maximizes the modularity function $Q$ (denoted as $\text{mod}(C)$ in Algorithm 7). Fig. 5.19 shows a visualization of the community structure at maximum modularity. The running time of CNM is essentially linear, $O(n\log^2 n)$; and this is considerably faster than most of community detection algorithms [246]. The algorithm starts with singletons, that
means every vertex is the only member of a community. Thus there are as many communi-
ties as vertices and every community has size one at the beginning. Subsequently a greedy
optimization of \( \text{mod}(C) \) is applied. The two communities whose amalgamation yields the
highest increase in \( \text{mod}(C) \) are merged repeatedly until no further increase in \( \text{mod}(C) \) is
possible. The CNM pseudocode is given in Algorithm 7

**Algorithm 7: CNM Algorithm**

**Input:** Graph \( G = (V, E) \)

**Output:** Clustering \( C \) of graph \( G \)

\[
C := \forall v \in V \, |v;
\]

**Method**

while \((true)\) do

\( \forall C_i, C_j \in C. \)

\[
\Delta Q_{C_i,C_j}^C = Q(G, C - C_i - C_j + (C_i \cup C_j)) - Q(G, C);
\]

Find \((C_i, C_j) \in C^2\) that has maximum \(\Delta Q_{C_i,C_j}^C\).

If (max \((\Delta Q_{C_i,C_j}^C) < 0)\) then

break;

\( C := C - C_i - C_j + (C_i \cup C_j); \)

---

**Figure 5.19:** A visualization of the community structure at maximum modularity

**K-Nearest Neighbor Algorithm (K-NN)**

The K-Nearest Neighbor (K-NN) algorithm [289] finds a group of \( k \) objects in a training set
that are closest to the test object, and bases the assignment of a label on the predominance of
a particular class in this neighborhood. There are three key elements in this approach: a set of labeled objects, a distance or similarity metric to compute distance between objects, and the value of \( k \), the number of nearest neighbors. To classify an unlabeled object, the distance of this object to the labeled objects is computed. Its \( k \)-nearest neighbors are identified, and the class labels of these nearest neighbors are then used to determine the class label of the object.

K-NN pseudocode is given in Algorithm 8. Given a training set \( D \) and a test object \( x = (x, y) \), the algorithm computes the distance (or similarity) between \( x \) and all the training objects \( (x, y) \in D \) to determine its nearest-neighbor list, \( D_x \), where \( x \) is the data of a training object, and \( y \) is its class. Likewise, \( x' \) is the data of the test object and \( y' \) is its class. Once the nearest-neighbor list is obtained, the test object is classified based on the majority class of its nearest neighbors, as illustrated next:

\[
y' = \arg\max_v \sum_{(x_i, y_i) \in D} I(v = y_i),
\]

Where \( v \) is a class label, \( y_i \) is the class label for the \( i \)th nearest neighbors, and \( I(\cdot) \) is an indicator function that returns the value 1 if its argument is true and 0 otherwise.

In our experiment, we use the K-NN strategy to produce a K-NN graph that is a directed graph \( G = (V, E) \), where \( V \) is the set of nodes and \( E \) is the set of links. Node \( v \) is connected to node \( u \) if \( u \) is one of the K-NNs of \( v \). Throughout the K-NN filtering process, the algorithm uses the distance function \( \rho \) (or similarity measure) to return the first \( u \leq k \) neighbors of node \( v \) in \( G \):

\[
u = \arg\min_{u \leq k} \rho(v, u)
\]

**Algorithm 8 K-NN Algorithm**

**Input:** \( D \), the set of \( k \) objects, and test object \( z = (x', y') \)

**Output:** \( y' = \arg\max_v \sum_{(x_i, y_i) \in D} I(v = y_i) \)

**Method:**
Compute \( d(x', x) \), the distance between \( z \) and every object, \( (x, y) \in D \)
Select \( D_z \subseteq D \), the set of \( k \) closest objects to \( z \).
5.5.4 Performance Metrics

We use the modularity $Q$ as the basic evaluation metric to show the performance of our method. Modularity is a property of a network that specifies the degree of division of that network into coherent communities [256]. When the division is a good one, in the sense that there are many edges within communities and only a few between them, $Q$ reflects a high value. Thus, modularity for unweighted and undirected networks is defined as the ratio of the difference between the actual and expected number of edges within the community (in a randomized graph with the same number of nodes and the same degree sequence). For the given community partition of a network $G = (V,E)$ with $|E|$ edges, modularity ($Q$) is given by:

$$Q = \sum_{c_i \in C} \left[ \frac{|E_{c_i}^{in}|}{|E|} - \left( \frac{2|E_{c_i}^{in}| + |E_{c_i}^{out}|}{2|E|} \right)^2 \right]$$

where $C$ is the set of all the communities, $c_i$ is a specific community in $C$, $|E_{c_i}^{in}|$ is the number of edges between nodes within a community $c_i$, and $|E_{c_i}^{out}|$ is the number of edges from the nodes in the community $c_i$ to the nodes outside $c_i$. $Q$ is always less than 1, and equal to 0 only if all nodes are put in the same community. High values of $Q$ indicate a strong community structure.
5.5.5 Experiment Results

Given the $100 \times 100$ similarity matrix ($SM$), we generate the frequency distribution of the similarity weights ($S_1$) as shown in Fig. 5.20. X-axis represents the similarity values (rounded to 2 decimal places) and Y-axis shows the respective frequencies of similarity among edges. Then we build several networks by filtering out edges using different methods. The goodness of the generated network is evaluated using CNM clustering and Modularity ($Q$) to assess the obtained communities.

Network Filtering Methods

**Method 1: Using an absolute similarity threshold**  The edges in the original similarity matrix (generated based on career interests i.e. $S_1$) are filtered using a threshold $\Delta$ to remove weak edges. Different values of $\Delta$ were used and the modularity metrics of the obtained networks are provided in Table 5.5. As indicated in the obtained results, $Q$ value decreases when we include more links of poor similarity weight ($0.60 \leq S_1 \leq 0.85$).
Table 5.5: Metrics obtained using Absolute Similarity Threshold filtering

<table>
<thead>
<tr>
<th>Threshold $\Delta$</th>
<th>Nodes</th>
<th>Edges</th>
<th>Clusters using CNM</th>
<th>Modularity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.85</td>
<td>100</td>
<td>387</td>
<td>6</td>
<td>0.777</td>
</tr>
<tr>
<td>0.80</td>
<td>100</td>
<td>627</td>
<td>4</td>
<td>0.697</td>
</tr>
<tr>
<td>0.75</td>
<td>100</td>
<td>853</td>
<td>4</td>
<td>0.627</td>
</tr>
<tr>
<td>0.70</td>
<td>100</td>
<td>1149</td>
<td>3</td>
<td>0.553</td>
</tr>
<tr>
<td>0.65</td>
<td>100</td>
<td>1492</td>
<td>3</td>
<td>0.487</td>
</tr>
<tr>
<td>0.60</td>
<td>100</td>
<td>1800</td>
<td>3</td>
<td>0.419</td>
</tr>
</tbody>
</table>

Table 5.6: Metrics obtained using $k$-Nearest Neighbors

Method 2: Using $k$-Nearest Neighbors  The edges in the original similarity matrix are further filtered using the best $k$-nearest neighbors strategy. For every node, its nearest $k$ neighbors are kept in the network, whereas the other edges are removed. Different values of $k$ were used and the metrics for the obtained networks are provided in Table 5.6. As expected, the more neighbors ($k$) we include, the more chances to add weak links, resulting in more inter-communities edges than inner-communities edges and so decreasing $Q$ value. On the other hand, a small number of ($k$) connects nodes which are highly similar in increasingly dense communities and, thus improving $Q$. Fig.5.21 shows the network built using K-NN method. ($n=100$, $k=5$). Different colors signify different communities found in the network using CNM algorithm.

<table>
<thead>
<tr>
<th>$k$</th>
<th>Nodes</th>
<th>Edges</th>
<th>Clusters using CNM</th>
<th>CNM_Modularity (Q)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>100</td>
<td>250</td>
<td>8</td>
<td>0.819</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>304</td>
<td>7</td>
<td>0.772</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>354</td>
<td>7</td>
<td>0.786</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>408</td>
<td>6</td>
<td>0.754</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>461</td>
<td>6</td>
<td>0.737</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>515</td>
<td>6</td>
<td>0.708</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>678</td>
<td>6</td>
<td>0.698</td>
</tr>
</tbody>
</table>

Table 5.6: Metrics obtained using $k$-Nearest Neighbors
Network Enhancement Method

The results of averaging a total of 150 simulations (for different network sizes with different number of edges) of our enhanced method that uses a Social interest-Based Triadic-Closure (TC_CNM) are presented in Table 5.7. The results consistently show an increase in modularity (Q) as triadic closures increase the intra-cluster coherence among nodes forming densely connected community structures (see Fig.5.22). Average (AVG), Maximum (MAX) and Minimum (MIN) of Q value obtained out of 25 simulation rounds per each network size N is presented in Table 5.8.

As indicated in the obtained results, social interests attributes (S2) can also be a driving force to build new ties among learners who are connected based only on similar career attributes (S1). Common social interests can motivate formation of new ties, which in turn result in the formation of more connected communities than structures from an expected random graph that essentially results in increasing Q value for different N-size networks (see Fig.5.23). Further, Fig.5.24 shows improvement of Q of networks which
## Table 5.7: Average results of different network sizes for TC_CNMP

<table>
<thead>
<tr>
<th>Serial</th>
<th>Nodes</th>
<th>Edges</th>
<th>Avg. Total Edges</th>
<th>Avg. (Q) TC_CNMP</th>
<th>Avg.(Q) CNM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>50</td>
<td>79</td>
<td>127</td>
<td>0.660</td>
<td>0.488</td>
</tr>
<tr>
<td>2</td>
<td>50</td>
<td>69</td>
<td>117</td>
<td>0.663</td>
<td>0.471</td>
</tr>
<tr>
<td>3</td>
<td>50</td>
<td>59</td>
<td>107</td>
<td>0.660</td>
<td>0.450</td>
</tr>
<tr>
<td>4</td>
<td>50</td>
<td>49</td>
<td>97</td>
<td>0.727</td>
<td>0.480</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>39</td>
<td>87</td>
<td>0.756</td>
<td>0.484</td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>329</td>
<td>377</td>
<td>0.794</td>
<td>0.741</td>
</tr>
<tr>
<td>7</td>
<td>100</td>
<td>279</td>
<td>327</td>
<td>0.807</td>
<td>0.713</td>
</tr>
<tr>
<td>8</td>
<td>100</td>
<td>239</td>
<td>287</td>
<td>0.811</td>
<td>0.711</td>
</tr>
<tr>
<td>9</td>
<td>100</td>
<td>189</td>
<td>237</td>
<td>0.813</td>
<td>0.667</td>
</tr>
<tr>
<td>10</td>
<td>100</td>
<td>149</td>
<td>197</td>
<td>0.820</td>
<td>0.644</td>
</tr>
<tr>
<td>11</td>
<td>150</td>
<td>749</td>
<td>797</td>
<td>0.871</td>
<td>0.829</td>
</tr>
<tr>
<td>12</td>
<td>150</td>
<td>639</td>
<td>687</td>
<td>0.863</td>
<td>0.817</td>
</tr>
<tr>
<td>13</td>
<td>150</td>
<td>539</td>
<td>587</td>
<td>0.856</td>
<td>0.800</td>
</tr>
<tr>
<td>14</td>
<td>150</td>
<td>429</td>
<td>477</td>
<td>0.856</td>
<td>0.781</td>
</tr>
<tr>
<td>15</td>
<td>150</td>
<td>329</td>
<td>377</td>
<td>0.843</td>
<td>0.754</td>
</tr>
<tr>
<td>16</td>
<td>200</td>
<td>1339</td>
<td>1387</td>
<td>0.882</td>
<td>0.861</td>
</tr>
<tr>
<td>17</td>
<td>200</td>
<td>1449</td>
<td>1197</td>
<td>0.890</td>
<td>0.852</td>
</tr>
<tr>
<td>18</td>
<td>200</td>
<td>959</td>
<td>1007</td>
<td>0.879</td>
<td>0.843</td>
</tr>
<tr>
<td>19</td>
<td>200</td>
<td>769</td>
<td>817</td>
<td>0.875</td>
<td>0.830</td>
</tr>
<tr>
<td>20</td>
<td>200</td>
<td>579</td>
<td>627</td>
<td>0.859</td>
<td>0.802</td>
</tr>
<tr>
<td>21</td>
<td>250</td>
<td>2019</td>
<td>2157</td>
<td>0.894</td>
<td>0.871</td>
</tr>
<tr>
<td>22</td>
<td>250</td>
<td>1809</td>
<td>1857</td>
<td>0.888</td>
<td>0.865</td>
</tr>
<tr>
<td>23</td>
<td>250</td>
<td>1509</td>
<td>1557</td>
<td>0.886</td>
<td>0.865</td>
</tr>
<tr>
<td>24</td>
<td>250</td>
<td>1209</td>
<td>1257</td>
<td>0.884</td>
<td>0.854</td>
</tr>
<tr>
<td>25</td>
<td>250</td>
<td>909</td>
<td>957</td>
<td>0.884</td>
<td>0.842</td>
</tr>
</tbody>
</table>

## Table 5.8: Average, maximum and minimum $Q$ value of N-size Networks

<table>
<thead>
<tr>
<th>N</th>
<th>Method</th>
<th>50</th>
<th>100</th>
<th>150</th>
<th>200</th>
<th>250</th>
</tr>
</thead>
<tbody>
<tr>
<td>$AVG(Q)$</td>
<td>CNM</td>
<td>0.475</td>
<td>0.695</td>
<td>0.796</td>
<td>0.838</td>
<td>0.861</td>
</tr>
<tr>
<td></td>
<td>TC_CNMP</td>
<td>0.693</td>
<td>0.809</td>
<td>0.858</td>
<td>0.877</td>
<td>0.887</td>
</tr>
<tr>
<td>$MAX(Q)$</td>
<td>CNM</td>
<td>0.617</td>
<td>0.808</td>
<td>0.857</td>
<td>0.879</td>
<td>0.887</td>
</tr>
<tr>
<td></td>
<td>TC_CNMP</td>
<td>0.832</td>
<td>0.860</td>
<td>0.886</td>
<td>0.893</td>
<td>0.896</td>
</tr>
<tr>
<td>$MIN(Q)$</td>
<td>CNM</td>
<td>0.318</td>
<td>0.496</td>
<td>0.618</td>
<td>0.703</td>
<td>0.796</td>
</tr>
<tr>
<td></td>
<td>TC_CNMP</td>
<td>0.525</td>
<td>0.712</td>
<td>0.820</td>
<td>0.820</td>
<td>0.870</td>
</tr>
</tbody>
</table>
built using different $S_1$ values ($0.65 \leq S_1 \leq 0.85$) and enhanced by the power of $S_2$ ($S_2 = 0.85$).

(a) Expected random graph using one similarity measure

(b) Graph using the second similarity measure-based Triadic-Closure

Figure 5.22: Differences in graph structures using similarity-based Triadic-Closure

Figure 5.23: $Q$ Vs Network Size
On the other hand, linking more learners of poor $S_2$ results in groups of vertices that have more of inter-group edges than the inner-group edges and thus less dense connected communities. As Fig.5.25 shows, $Q$ values decreases as $S_2$ threshold decrease ($0.50 \leq S_1 \leq 0.90$) while creating the new edges in the original networks of different sizes ($N = 100, N = 50$). This is consistent with the result of decreasing $Q$ value against increasing number of added edges of weak similarity weights (see Fig. 5.26 and Fig.5.27)

![Figure 5.24: Q Vs Similarity Threshold ($S_1$) ($N = 100, S_2 = 85$)](image1)

![Figure 5.25: Q by TC_CNM Vs Similarity Threshold $S_2$ of different network size](image2)
Figure 5.26: Q Vs Number of added edges (N = 100)

Figure 5.27: Q Vs Number of added edges (N = 50)

5.6 Examining CoP-Network: A SNA Example

In this section, we apply SNA metrics to examine the structure of CoP Network example in a way that allows visual mapping and quantitative analysis. Network visualization views the mapping of the individual's ties and the strength of these ties; while quantitative side determines key players and key ties within a network. The first step to begin a SNA is to create visual mappings of the CoP Network. Fig.5.28 is a visual representation of the
networks created between learners of similar career interests. The circles represent learners, or nodes, and the lines between them indicate that two nodes share also some social interests. SNA tools allow changing the size, shape, color and name of the nodes to create the most efficient visualization of the network. The lines, or ties, can also be changed to be different colors, or thickness. For example, Fig. 5.29 displays the connections between local and expatriate learners. Fig. displays the connections of the faculty members or mentors (denoted as Prof.) with learners.

Figure 5.28: CoP network of learners
Figure 5.29: Visualize CoP Network as connections between local and expatriate

Figure 5.30: Visualize connections between learners and mentors in CoP Network
<table>
<thead>
<tr>
<th>Learner</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hashim</td>
<td>9</td>
</tr>
<tr>
<td>Tariq</td>
<td>7</td>
</tr>
<tr>
<td>Fadi</td>
<td>6</td>
</tr>
<tr>
<td>Ibrahim</td>
<td>6</td>
</tr>
<tr>
<td>Nadim</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 5.9: Selected key learners using Degree Centrality measure

<table>
<thead>
<tr>
<th>Learner</th>
<th>Betweenness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tariq</td>
<td>909.32</td>
</tr>
<tr>
<td>Talal</td>
<td>868.05</td>
</tr>
<tr>
<td>Amir</td>
<td>706.56</td>
</tr>
<tr>
<td>Fadi</td>
<td>660.83</td>
</tr>
<tr>
<td>Fisal</td>
<td>639.30</td>
</tr>
</tbody>
</table>

Table 5.10: Selected key learners using Betweenness Centrality measure

Further, one of the most practical uses of SNA is to identify the most important or key nodes or users in a network. There are several metrics that can be used to identify key users and key ties depending on the purpose of the analysis. For our example, learners of high degree centrality are listed in Table 5.9; while Table 5.10 shows learners of high betweenness centrality. Fig. 5.31 and Fig. 5.32 visualize the scores of learners with respect to degree and betweenness. High betweenness centrality for nodes suggests that the network has pockets of densely communities while low values of betweenness centrality suggest that nodes of the entire network are well connected to each other representing the absence of well defined boundary structure for communities.
Figure 5.31: Visualization of the Degree Centrality scores in CoP Network
The eccentricity measure (Table 5.11) is another metric that may be used to identify key nodes. The eccentricity of a node is the maximal shortest path distance between the node and any other node. We can see that Tariq appears in the three tables with a powerful position in the network at different measures. This suggests that Tariq have the ability to hinder or change information passed along him; and to play the role of a bridge between different communities. Thus, Tariq according to brokerage mechanisms has the most potential to be an opinion leader who can influence the behaviors of others.

Hubs and authorities are another important measures to identity important nodes. Good hubs are those nodes which point to many good authorities; and good authorities are those pointed to by many good hubs. Since our network is indirected network, we calculated one metric for both measures as presented in Table5.12. Fig.5.33 shows a visualization of
Table 5.11: Selected key learners using Degree Centrality measure A

<table>
<thead>
<tr>
<th>Learner</th>
<th>Eccentricity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tariq</td>
<td>8</td>
</tr>
<tr>
<td>Hadman</td>
<td>8</td>
</tr>
<tr>
<td>Malik</td>
<td>8</td>
</tr>
<tr>
<td>Fikri</td>
<td>9</td>
</tr>
<tr>
<td>Ibrahim</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 5.12: Selected key learners using Degree Centrality measure B

<table>
<thead>
<tr>
<th>Learner</th>
<th>Hubs and Authorities (H&amp;A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hashim</td>
<td>1</td>
</tr>
<tr>
<td>Omar</td>
<td>0.44</td>
</tr>
<tr>
<td>Alaa</td>
<td>0.41</td>
</tr>
<tr>
<td>Mustafa</td>
<td>0.41</td>
</tr>
<tr>
<td>Mohammed</td>
<td>0.38</td>
</tr>
</tbody>
</table>

H&A scores. Notice that Hashim ranks at the top of important nodes; and he also scores the highest degree centrality as shown in Table 5.9. This suggests that Hashim acts as a definitive source of information in the network with respect to closure mechanisms. In next chapter, we will explore more about the role and structural position of individuals in networks; and how to determine the most influential individuals to spread information or to affect behaviors.
We introduced a similarity function ($RW_D$) to enable self-assigned weights and position orders to profile attributes, when determining similarity measure across learners in CoP-Network. We employed the CNM algorithm and Modularity ($Q$) metric to evaluate the goodness of clustering learners into communities within the constructed CoP-Network using two methods: an absolute similarity threshold ($\Delta$) and $k$-Nearest Neighbors. While the initial results showed a relatively good modularity of generated networks using high values for $\Delta$ ($\geq 80$) and $k$ ($\geq 6$), we further implemented a community detection enhancement method using a triadic closure approach (TC_CNM) based on two similarity measures (career interests and social interests). The results of our proposed method indicated an increase in
modularity ($Q$) for network of different sizes. This is because triadic closures increase the intra-cluster coherence among nodes forming densely connected community structures. Allowing more edges to be added when handling large networks ($\geq 250$) will lead in more improved results. In next chapter, we explore social influence in CoP-Network.
Chapter 6: Social Influence

6.1 Introduction

The notion of how to persuade others has been a popular and profitable subject long before the computers were invented. In 350 BC, Aristotle described three modes of persuasion that are important to be proficient in, not only as a politician, but also as a citizen in everyday life [290]. These modes are: the spoken word (logos), speakers' personal character (ethos), and audience emotions triggers (pathos). Since then, persuasion and social influence have become the most important functions of communication [291, 292, 293, 294, 295, 296]. In the persuasion paradigm, influence is presented as a detailed argumentation delivered to individual recipients which impact is limited to minimal social interactions. Social influence—in contrast—includes simple information (e.g. “Likes” in FaceBook) about the source’s behaviour in the network, and delivered across wider social interactions [295]. With the rise of media and technology, understanding audience and triggering desired emotions and behaviors can be achieved easier than in ancient Greek. Persuasive technological strategies have been applied widely in the context of commercial and healthcare promotion, but not so much for educational purposes [297] [298, 299]. Social context is crucial in educational environments to define important properties of persuasion power that could lead to increased learning retention. There are still few projects of applying persuasive technology in educational context, such as the EuroPLOT (2010) project. This project was initiated to improve technology enhanced teaching and learning using persuasive elements to embrace successful pedagogical and instructional methods in different domains such as academic business computing, language learning, and chemical industrial training [300, 301]. Authors in [302] developed a persuasive method to change students' attitude towards mathematic learning into more positive levels.

We discussed earlier the role of social networks to provide an interactive platform for social learning, as individuals develop relationships related to their career interests. In

1http://www.eplot.eu
this chapter, we extend our work to apply persuasion and social influence approaches in tandem to career development in higher education as follows. Our proposed model empowers future recruiters to first identify a small group of learners (labelled persuasive agents or key users) with special personal abilities, who also are eligible to increase diffusion of desired career behaviours across the social network. The features of a successful persuasive model involve the prominent role of a higher recruiting authority which facilitates positive interactions between key users and market environments about career prospects. To build this environment, we incorporate Fogg Behavioral Model (FBM) and Persuasive System Design Model (PSD), and key principles from personal traits to propose our Behavioral Change Support System for Career-Adoption (BCSS-CA) model. BCSS-CA aims to encourage learners in higher education to adopt and diffuse desired career behaviours according to current "local" market needs advocated by recruiting authorities, in support to national economic growth objectives.

6.2 Background

6.2.1 Social Influence

Social influence is defined as a "change in an individual's thoughts, feelings, attitudes, or behaviors that results from interaction with another individual or a group" [303]. Social influence occurs when an individual changes his/her behavior after interacting with other individuals who tend to be similar or superior. Identifying social influence in networks is critical to understanding how behaviors spread [304]. Users in social networks connect with each others by different type of relationships (i.e. friendships, memberships, and common interests); to share information. Information in social networks reflects users’ online activities, and plays an essential role in diffusion analysis [305]. Users influence each others to form different networking structures or crowds which directly, explicitly or implicitly, promote information diffusion [306, 305]. Given a network represented by a graph $G = (V, E)$ with the set of vertices $V$ representing individuals and the set of edges $E$ representing links among individuals, if any node $v_i \in V$, replicates the action of another node $v_j \in V$, we
may assume that \( v_2 \) has influence on \( v_1 \) [248]. When the influential node performs an action, nodes connected to it will replicate the action starting information propagation. The influence of a node on other connected nodes can be due to some external factors like trust [307]; or to the popularity of the information or action [308]; or to the prestige or celebrity of the influential node [309]. The relational structure in the network (i.e. role and structural position of individuals) also plays an important part in determining how influential an individual is, in this network[310, 311, 305]. Therefore, understanding the structure of social network provides important further insights into how individuals influence each other’s behavior. The underlying social mechanisms are present both at the level of the whole network, and also at a ego level in the network, between an individual and his or her set of friends or colleagues. In many cases, individuals tend to align their behavior with the behavior of their immediate neighbors in the social network, rather than with the network as a whole [248].

A very important task for the maximization of diffusion in social networks is to identify the influential nodes or the adopters who can exploit social network effects. Researchers in social influence target to identify a small set of influential individuals (referred to as a seed set) who can ideally maximize the influence across the entire network in minimal time[247, 312, 313]. The problem of influence maximization can be expressed as follows: “given a network with influence estimates, how to select an initial set of \( k \) users such that they eventually influence the largest number of users in the social network” [314].

Given a social network, a positive integer \( k \), and information diffusion model, the goal is to find such a target set \( A_k^* \) of \( k \) nodes that maximizes the expected number of adopters of the information if \( A_k^* \) initially adopts it. The expected number of nodes influenced by a target set is referred to as its influence degree, and this combinatorial optimization problem is called the influence maximization problem of size \( k \) [247]. Information diffusion models used in such problems are probabilistic models for the process by which a certain information spreads through the network. They provide theoretical basis and technical support for understanding the crucial factors associated with the spread process by visualizing
the spread process, inferring the possible paths, and predicting trends [315, 305]. Information diffusion models can be classified into three main categories: structural models, staged models, and feature models, depending on the type of insight provided to the analyst [305]. The structural models are built on the assumption that interactions at the micro level between pairs of nodes and the topology of their interconnections (i.e., network structure) can explain the dynamics of the spreading process at the macro level. Staged models divide the information adoption process into several states and investigate the flow of information based on switching among these states. Feature models study dynamic patterns of information diffusion based on characteristics of these patterns (such as temporal rhythms).

Social influence is becoming a complex and subtle force that governs the dynamics of all social networks. Therefore, there is a need for methods and techniques to analyze and quantify social influences. In this chapter, we examine the influence maximization problem on the CoP network for—particularly—the structural influence diffusion models. SNA measures (addressed in Chap) help in addressing the sources and distribution of influence power in the social networks, based on the structure of the network. Many research studies discussed the power of a specific node in the network by addressing the attributes of centrality using SNA such as degree, closeness, and betweenness centralities [311, 316, 317, 248].

6.2.2 Structural Influence Diffusion Models

Two widely-used fundamental information diffusion models are the independent cascade (IC) model [317, 318, 319] and the linear threshold (LT) model [320, 317]. In both these models, at a given timestamp, each node is either active (has already adopted the information) or inactive, and each node's tendency to become active increases monotonically as more of its neighbors become active. Nodes can switch from being inactive to being active, but cannot switch from being active to being inactive. Given an initial set $\mathcal{A}$ of active nodes, both models assume that the nodes in $\mathcal{A}$ first become active and all the other nodes remain inactive at time-step 0. The diffusion process of active nodes unfolds in discrete time-steps $t \geq 0$. As time unfolds, more and more of neighbors of an inactive node $u$ become active.
eventually making $u$ become active, and $u$'s decision may in turn trigger further decisions by nodes to which $u$ is connected [321]. Thus, the spread of the information through the network $G$ is represented as the spread of active nodes on $G$.

IC and LT are both independent of time. They based on the discrete Bernoulli distribution which have two possible outcomes: $n = 1$ ("success") occurs with probability $p$; and $n = 0$ ("failure") occurs with probability $q = 1 - p$, where $0 < p < 1$. In social influence a specific node $u$ has fixed probability to influence its inactive neighbor $v$. If it activates the neighbor then this is a successful attempt and failure otherwise. Each attempt can be shown as Bernoulli trial. Figure 6.1 shows a sample illustration to explain Bernoulli trials. The influence probability can be estimates using Maximum Likelihood Estimator (MLE) [314] as the ratio of successful attempts over the total number of trials: $p_{u,v} = \frac{A_{u,v}}{A_{u,v} + A_{v,u}}$. Next, we define each model with more details.

![Figure 6.1: Bernoulli distribution: node $u$ will have fixed probability to influence its inactive neighbor $v_1$, $v_2$, and $v_3$. If $u$ attempt is "successful" node $v$ will be activated otherwise node $v$ will remain inactive.](image)

**Independent Cascade (IC) Model**

In IC Model, we specify a real value $p_{u,v} \in [0, 1]$ for each directed link $(u,v)$ in advance. Here, $p_{u,v}$ is referred to as the propagation probability through link $(u,v)$. The model starts with an initial set of active nodes $A_0$. This set of individuals should be chosen to generate the maximum influence during the cascade diffusion process. The process occurs in discrete
steps according to the following randomized rule: when node $u$ becomes active for the first time in timestep $t$, it is given one chance to activate each of its currently inactive neighbor $v$; in that case $u$ is called contagious that means it has the ability to affect other nodes as shown in Fig.6.2a. Node $u$ succeeds to influence its neighbor $v$ with a probability $p_{u,v}$ independently of past history. If $u$ succeeds, then $v$ will become active in timestep $t+1$ as shown in Fig.6.2b; but whether or not $u$ succeeds, it cannot make any further attempts to activate $v$ in future rounds[317]. The same process continues until $u$’s communicate with all neighbors for influence attempts. The process terminates if no more activations are possible.

Figure 6.2: Example of Independent Cascade Model process

**Linear Threshold (LT) Model**

In LT model, a weight $w_{u,v}$ is used to measure the tendency of a node $u$ to be influenced by each neighbor $v$ such that $\sum_{\text{neighbour of } u} w_{u,v} \leq 1$. Starting with initial set of active nodes $A_0$, the influence propagation process continues according to the following randomized rule: each node $u$ is assigned a threshold $\theta_u$ randomly from the interval $[0, 1]$; the threshold represents the weight fraction of $u$'s neighbors that must adopt the behavior (be active) in order for $u$ to become active and adopt the same behavior. At timestamp $t$, all nodes that were active in time $t - 1$ remain active, and we then activate any node $u$ for which the total weight of its active neighbors is at least $\theta_u$; where

$$\sum_{\text{active neighbour of } u} w_{u,v} \geq \theta_u$$
The thresholds $\theta_n$ represents the tendency of nodes to adopt the new behavior when their neighbors do [317]. The process terminates if no more activations are possible. Fig. 6.3a-b shows an example of the Linear Threshold model process.

![Figure 6.3: Example of Linear Threshold Model process](image)

Note that node thresholds are random in LT model, while in the IC model it is the propagations through links that are random.

**The influence maximization problem**

The influence maximization problem of size $k$ on a network $G = (V, E)$ for the IC and LT models is defined as follows [247]:

Find a set $A_k^* \subseteq V$ of $k$ nodes to target for initial activation such that $\sigma(A_k^*) \geq \sigma(S)$ for any set $S$ of $k$ nodes, that is, find

$$A_k^* = \arg\max_{A \subseteq V, |A| = k} \sigma(A)$$

where $|S|$ stands for the number of elements of set $S$ and $\sigma(A)$ represents the influence degree of $A$. Kempe et al [317] experimentally showed -on large social networks- that the greedy algorithm can give a good approximate solution to this problem, and that it is NP-Complete problem [317]. The efficient extracting of influential nodes on a social network has become one of the central problems in social network analysis. Kimura, Satio and Nakano [322] proposed a method of efficiently estimating all the marginal gains $\nabla \sigma(A)$ for the influence degree $\sigma(A)$ of a given target set $A$, and applied it to approximately solving the
problem. A tractable model [323] was proposed using two natural special cases of IC model for extracting influential nodes and experimentally demonstrate small propagation probabilities through links can give good approximations for discovering influential node sets. In additions, methods based on bond percolation (e.g., [324]) and submodularity (e.g., [325]) were proposed for efficiently estimating the greedy solution. Succeeding work further improved the efficiency by approximating the solution using a heuristic [316]. A recent work [313] proposed a method for identifying influential nodes in complex networks with community structure using the information transfer probability between any pair of nodes and the k-medoid clustering algorithm. The experimental results show that the influential nodes identified by the k-medoid method can influence a larger scope in networks with obvious community structure than the greedy algorithm without reducing the expected number of influenced nodes.

In our research work, we apply social influence mining to identifying the most influential individuals in the CoP network in order employ them as “persuasive agents” in our model for career-adopter persuasive technology.

6.2.3 Persuasive Technology

Persuasive technology is a recent proposal for using computers to change human behavior [326, 327]. It aims to alter the mindset, attitudes and behaviors of individuals through technologies which create opportunities for persuasive interactions such as those enabled by Web and social networks. Persuasion is defined as “human communication designed to influence the autonomous judgments and actions of others” [294]. The very early work of Bernays “Manipulating Public Opinion: The Why and The How” [328] describes how to use the means and insights of social theories to change attitudes and behaviors of larger groups of individuals in an intended direction (i.e. adopting certain career paths). According to this study, the main process used in the psychology of public persuasion involves: (1) collecting facts and opinions of the public and the object of interest; (2) applying diagnostic procedures and statistics to interpret this collected data, and (3) applying various techniques
of persuasion (described next) to guide the target group towards adopting desired ideas or behaviors.

The literature of persuasion science ([292, 291, 327, 329]) describes different techniques to change attitudes and behaviors. We selected the following five of these techniques for their relevance to our study:

- Authority: authority is considered a form of social influence as individuals are inclined to follow recommendations and suggestions originating from authorities.

- Social proof: when individuals observe multiple instances of other users presenting the same belief or behavior, they are more likely to believe and behave similarly.

- Scarcity: scarcity (or assumed scarcity) increases the perceived value of products and opportunities (to incite its exclusive value).

- Liking: individuals say "yes" to people they like. When a request is made by someone we like, we are more inclined to act accordingly. Increased interpersonal similarity is proved to also increase linking.

- Reciprocity: when a the receiver (or persuadee) is in debt to the source (persuaders), he or she will comply with persuasive requests to even out this discrepancy.

A key construct for research in the field of persuasive technology in related to public persuasion is the Behavioral Change Support Systems (BCSSs) described next.

**Behavioral change support systems (BCSSs)**

A BCSS is defined as "information system designed to form, alter or reinforce attitudes, behaviors or an act of complying without using deception, coercion or inducements" [330]. By definition, BCSS may utilize either computer-human persuasion or computer-mediated persuasion to achieve three outcomes: (1) reinforcement of current attitudes or behaviors; (2) changing an individual's response to an issue; and (3) shaping attitudes and behaviors by formulation of a pattern for a situation when one does not exist beforehand.
Computer-human persuasion utilizes some patterns of interaction similar to social communication, whereas computer-mediated persuasion means that individuals are persuading others through computers (e.g. discussion forums, instant messages, blogs, or social networks) [331]. There are examples of BCSSs developed to support people to change their behavior in healthcare issues to become more physically active, or to shop and eat healthier (e.g. [332, 333, 334]). In the energy sector also, BCSSs have been created to help people become aware of their energy consumption and to reduce it (e.g., [335, 336]). However, BCSSs also hold potential for other domains, e.g. in the context of the higher education and career development. For achieving better outcomes from BCSSs, they should be designed by using persuasive system design lifecycle [330], described next.

**Persuasive System Design Lifecycle**

The Persuasive System Design (PSD) lifecycle proposed by Fogg [327] is a widely utilized framework that provides a systematic analysis and design methods to develop BCSS software solutions. In PSD, the *persuasion context* module comprises elements analyzing whether persuasion can take place. This context exhibits intent, event, and strategy elements (see Fig. 6.4). The intent includes the persuader and the deliberate target behavior (change type). The event contains the: use, user, and technology sub-contexts. The “use” context refers to the problem domain dependent features (e.g., education, healthcare). The user context includes, for example, interests, goals, commitments, motivations, abilities, pre-existing attitudes, and lifestyles. The technology context refers to the features of the technological platform (e.g., interactive display, visualized feedback). In strategy dimension, PSD emphasizes two strategic elements, namely the message and the route. The message refers to the form and/or content selected to deliver the persuasive message for the planned (behavior or attitude) change (e.g. statistical data, dialogue system, or raw text). The route refers to the provision of one or a few arguments represented by convincing facts.
The second module in PSD focuses on the persuasion itself or system qualities, describing how an application may be persuasive on an operational level, though four dimensions that are: primary task support, dialogue support, system credibility support, and social support. Primary task support dimension addresses the target behaviors (i.e. reduction, personalization, self-monitoring). Dialogue support dimension deals with the feedback that the system offers in guiding the user to reach the intended behavior (i.e. rewards, similarity, liking). System credibility support dimension deals with the requirements of credibility as a key of persuasive processes (i.e. trustworthiness, expertise, real world feel, third party involvement). Finally, social support dimension describes how to design the system so that it motivates users by leveraging social influence via incorporating elements that affect the overall persuasiveness of the system (i.e. normative influence, social learning, cooperation, recognition). PSD model is described in more details in [337, 331].

Further, PSD model suggests three phases before implementing the persuasive systems: (1) understand key issues; (2) carefully analyze the persuasion context and the selection of the pervasive design principles; and (3) design the system qualities. Having thoroughly evaluated the issues in career readiness and career success; and the need for persuasion to motivate learners in higher education to adopt a given career path in response to market needs, we have established the first two phases of PSD thus far. Next, we present our proposed persuasive approach for career adoption as step toward the third phase of PSD.
6.3 Persuasive Approach for Career Adoption

6.3.1 Behavior Change Support Model

In this section, we describe the underlying structure of our proposed persuasive approach for Behavioral Change Support System for Career-Adoption (BCSS-CA). As discussed in the introduction of this chapter, influence in persuasive strategies is presented as persuasive message that is delivered to individual recipients in a setting of minimum social interaction; whereas social influence deliver messages from influential individuals in more complex social context that may involve interactions among participants. Accordingly, we propose to start the persuasion process by identifying the most influential individuals in the CoP-Network to be hired as persuasive agents. Persuasive agents be will treated by different persuasive strategies according to their "persuasive profile" to get them to adopt the desired career path. Pervasive agents are then expected to leverage their powerful structure position in the CoP network to influence other learners to adopt same career paths as required by the persuasion source (i.e. human resource planning authority, recruitment agency).

Our proposed BCSS-CA model shown in Fig.6.5 aims to stimulate persuasive agents to adopt certain careers based on the recommendations of local human resource authorities that oversee the market needs. The model is a part of career development interactive process which continues to monitor, analyze, and process career-related information; and periodically provides feedback to learners to keep them up-to-date about career opportunities and the special skills set required in the areas of career of interest. The designated authorities target first to hire this group of "persuasive agents" to help in influencing other learners to adopt the desired career path. The performance of persuasive agents to get more learners adopting the desired career is monitored by the persuaders in order to provide frequent feedback that can be also visualized (i.e. number of influenced individuals by each agent). This is in alignment to the original process of developing the tacit knowledge and professional skills required for this career. How to stimulate key users to be persuasive agents is
Figure 6.5: The proposed Behavioral Change Support System for Career-Adoption (BCSS-CA) discussed next.

6.3.2 Persuasive Agents Stimulation

Identifying persuasive agents involve creating the "personalized persuasive profile" to develop an understanding of individual behavior in response to persuasion; and so use suitable means to persuade each agent into adopting the desired career path. They will then naturally influence others in their circle to follow. Typically, the persuasive profile will include measures of "persuasive susceptibility": i.e. collections of expected effects of different influence strategies or principles for a specific individual such as authority, liking, and reciprocity. Having the persuasive profile of each agent contributes also in deciding the right motivator, right trigger and right time to persuade him or her. Persuasive agents can then be categorized into different behavioral modalities for the source to design tailored persuasive techniques or messages. For example, for individuals who are most susceptible to scarcity influence principle, the persuasive message can be delivered as "It is your unique opportunity to lead the data analytics career in UAE to be the first data analyst in the country". While for those who are most susceptible to authority, the message could be "Information Authority announces that there is a big demand on data analysts; and calls upon national IT students to apply for the data analysts profession."
The Fogg Behavioral Model (FBM) [338] states that for a target behavior to happen, an individual must have sufficient motivation, sufficient ability, and an effective trigger. All three factors must be present at the same time for the behavior to occur. Most specifically, as individual has increased motivation and increased ability, the more likely it is that he or she will perform the target behavior. The FBM model provides us with the insights to use the learners of highest career dispositions as the primary "persuasive agents" as they accordingly will have the required levels of motivation and ability to adopt the target career. Then, based on their structural position in the CoP network, they will be able to influence other individuals -who should also have some non-zero- level of both motivation and ability- to follow them. As illustrated in Fig.6.6, we classify motivators, abilities and triggers to apply FBM in our design approach as follows:

- **Motivators**: The goal in designing for motivation is, conceptually, to move an individual to a higher position in the FBM landscape. The individuals who have high ability but low motivation need to have motivation increased so they cross the behavior activation threshold to become persuasive agents. We identify two main types of motivators [339]: (a) intrinsic motivation which means that the individual’s motivational stimuli are coming from within based on the desire to perform a specific task (i.e. independence, power, career success); and (b) Extrinsic motivation which that the individual’s motivational stimuli are coming from outside. The desire to perform a task is controlled by an outside source who will reward the individual for performing the task (i.e. financial benefits). Persuasive agents are selected so that they already have the strongest desire to pursue a successful career path. Their structural position in the network fulfills their need to feel they are unique (independence) and to have influence on others (power).

- **Abilities**: persuasive design relies heavily on the power of individual ability that makes the targeted behavior easier to do. Accordingly, we differentiate two set of abilities for our persuasive agents: (a) career dispositions; (b) personalized persuasive profile, and (c) powerful network structure. As discussed in career dispositions
refer to the scheme of attitudes, assumptions and skills that engenders professional behaviors, and influences the ability to adapt and respond to change. We need the persuasive agents to be able to respond and adopt to the change in career needs as soon as it is delivered from the persuaders. On the other hand, personalized persuasive profile combines the insights of personal traits and persuasion principles to personalize the persuasive intervention in a way that the messages, timing, interfaces, the persuasive strategies and other factors of the BCSS are tailored to one specific individual or agent.

- **Triggers**: for behaviors where individuals are already above the activation threshold – they have sufficient motivation and ability – a trigger is all that is required to tell individual to perform the behavior now (i.e. Kairos). We identify three types of triggers: (a) interaction with industry, (b) feedback, and (c) recommendations and suggestions. Interactions with industry either via CoP networks or through industrial insights presented in Analytic Dashboards serve as triggers to encourage individuals to adopt future careers when they most in demands. Another feasible approach to prompting behavioral changes in adopting career paths is to monitor career adoption development; and provide real-time feedback to the individuals. We may enhance the strength of feedback using instantly processed and visualized results driven from the CoPs monitoring mechanisms shown in . As for recommendations and suggestions, they may come as direct messages from human resources authorities or recruitment agencies.
6.4 Social Mining and Influence Diffusion

In our proposed method to identify persuasive agents, we employ social mining techniques in order to select initial seed nodes effectively so as to maximize influence diffusion in CoP-Network. For comparative analysis, we use different SNA metrics ($sna - d$) for seeds selection. IC and LT influence diffusion models require as input, initial seed nodes ($k$) which are considered to be influenced (have performed action a already) at the start of the experiment. Algorithm 9 defines the implemented algorithm to mine influence in CoP-Network using various selection methods to select $k$. In Algorithm 9, $G$ is a graph representing the CoP social network, consisting of $V$ nodes and $E$ edges, where $sna - d$ is the structure position of a node with respect to the five measures mentioned above. First, we initiate seed set $S$ and its influence $I_S$. The while loop stores a set of nodes with highest $sna - d$ in $S$; this set is used to start influence spread process. Next, we execute the influence propagation process using LT and IC diffusion models. The maximum influence from $S$, i.e. $I_S$ is returned.
Algorithm 9 Influence mining using various SNA-based seed selection methods

Input
\( G : (V, E, sna - d) \), \( k : \) the number of target influential nodes

Output
Seed set \( (S) \) and its influence \( (I_S) \)

Method

\[
S \leftarrow NULL \\
I_S \leftarrow NULL \\
while |S| < k \\
\quad S \leftarrow S \cup \{v|(sna - d)_v = \max(sna - d): v \in V - S\}
\]

end while

while \( V \neq NULL \) do

\[
A \leftarrow NULL \\
V \leftarrow V - S \\
\quad \text{for all } v \in V \text{ do} \\
\quad \quad \text{apply IC model} \quad \text{\( \backslash \)apply LT model}
\]

end for

\[
S \leftarrow S \cup A \\
I_S = I_S + |A|
\]

end while

return \( S \)

return \( I_S \)

Kempe et al. [317] showed the effectiveness of the greedy algorithm (see Algorithm 10) for the influence maximization problem under the IC and LT models. The greedy algorithm guarantees that the influence spread is within \((1 - 1/e)\) of the optimal influence spread (where \(e\) is the base of the natural logarithm). In Algorithm 10, \(A_k\) denotes the set of \(k\) nodes obtained by the greedy algorithm, and so it is referred to as the greedy solution of size \(k\). The resulting set \(S\) activates at least \((1 - 1/e) > 63\%\) of the number of nodes that any size-\(k\) set could activate [340].
Algorithm 10 Greedy algorithm to solve the influence maximization problem under IC & LT

Input
\( G = (V, E), k, SS \): SS is a designated propagation probability;

Output
\( A_k \): the identified \( k \) influential nodes set

Method
Set \( A \leftarrow NULL \)
for all \( i = 1 \) to \( k \) do
    Choose a node \( v_i \in V \) maximizing \( \sigma(A \cup \{v_i\}), (v \in V - A) \)
    Set \( A \leftarrow A \cup \{v_i\} \)
end for

6.5 Performance Evaluation

6.5.1 Social Influence Mining Experimental Evaluation

In this section we will compare influence spread on CoP Network based on two famous influence propagation models: the Linear Threshold (LT) model and the Independent Cascade (IC) model. The algorithms are implemented using R igraph package for network analysis.

Experimental Parameters

We applied the experiment on the generated CoP network in Chapter. We used five SNA methods: degree, closeness, betweenness, closeness, coreness and eccentricity for selection of nodes as initial seed. We set the seed nodes \( k = 5\% \) of total nodes. We fix the number of steps (iterations) to be 50. The number of iterations can be considered as an indicator of the time required to influence the entire network as they represent how many iterations it takes to try to influence every node in a network. The ideal seed selection would be with maximum influence and minimum iterations for a network. We measured one primary parameter for the comparative analysis that is the percentage of vertices influenced.
Results and Discussion

Fig. 6.7 shows the results of influence diffusion using the five selection methods to select initial seed nodes \((k)\) for IC and LT models. The first observation from the graphs for the influence spread is the effects of different seed selection strategies on influence results; and that high coreness nodes have the highest impact in influencing other nodes under both models \((29\%\text{ and } 55\%\text{ respectively})\). As discussed earlier, nodes are said to have coreness \(k\) (or, equivalently, to belong to the \(k\)-shell) if they belong to the \(k\)-core but not to the \((k+1)\)-core. Thus, by definition cores are “concentric”, meaning that nodes belonging to the 3-core belong to the 2-core and 1-core, as well. High values of “coreness”, though, clearly correspond to nodes with a more central position in the network structure \([341]\).

On the other hand and among centrality measures, closeness under both models achieved the least impact to influence other nodes. This is could be attributed to the fact that closeness centrality measures implicitly assume that the underlying social network is strongly connected\([342, 248]\).

Degree and betweenness centrality measures perform well under both model; with degree performs better that betweenness under IC model; and the other way around under LT model. This is can be justified by the technical nature of each model, as that node thresholds are random in LT model, while in the IC model it is the propagations through links that are random. Since degree centrality is a local measure, the small value of node thresholds in LT results in less tendency of nodes to adopt the new behavior when their neighbors do. On the other hand, betweenness measure depends on the number of shortest paths or links that pass through a given node; while IC model operates according to the random propagation probability through these links. Finally, the strategy of using eccentricity measure to select seeds performs well in LT model which was higher by 43\% than IC model. This is also can be attributed to the technical aspects of the two models given that the eccentricity of a node is the maximal shortest path distance between the node and any other node.
Investigating the influential nodes selected by each seeding strategies (see Table 6.1), it is noticed that there are nodes appeared as influential nodes in all (i.e. Tariq) or most seeding strategies (i.e. Ibrahim). This is due to their unique position in the network which makes them potential candidates to be “career persuasive agents”. However, we target to limit the seed selection for those seeds nodes which score high in their career dispositions (Car-Disp). We need to “control” the influence as not to be diffused only by nodes which has the structural power in the CoP network, but with nodes which also demonstrate career-oriented abilities. The selection that is based on the network position and career abilities is one of the principal factor in our persuasive model.

Figure 6.7: Percentage of influenced vertices when $k = 5\%$
Influential
Degree
Ibrahim, Fadi, Tariq
Betweenness
Hamdan, Faisal, Fadi, Amir, Talal, Tariq
Closeness
Habib, Jihad, Fahd, Faras, Hamdan, Tariq
Coreness
Ibrahim, Tariq, Salem, Habib, Adnan, Nadim
Eccentricity
Tariq, Khalil, Jihad, Fahd, Ali, Ibrahim

Table 6.1: Influential nodes in CoP Network

The importance of identifying the key users as potential persuasive agents; and understanding their personal differences is supported by recent research studies that have examined individual differences in the personality of users for persuasive applications. For example, Halko and Kientz [343] explored the relationships between the Big Five personality model and a preference for individual distinct persuasive strategies. The researchers find a number of relationships between the personality of users and the preferred types of persuasive messages such as authoritative, as competitive, or reinforcement messages. In similar direction, Nov and Arazy [344] recently explored relationships between personality and interface design. They showed that user’s conscientiousness levels (the personality trait of being thorough and careful to do a task efficiently) relate to their reactions to the use of social proof messages. Authors in [345] introduced persuasion profiling as a method to personalize the persuasive messages used by a system to influence its users. Persuasion profile will contain a collection of estimates of the expected effects of different influence principles for a specific individual. Hence, an individual’s persuasion profile indicates which influence principles are expected to be most effective. Actually, the idea of creating a profile of users at the level of the persuasive techniques was for the first time introduced by Fogg in 2004 during his lectures [346, 347]; then Fogg publicity explained and discussed persuasion profiles in the 2006 U.S. Federal Trade hearing on the subject of protecting consumers in the next digital age [348]. Persuasive profile doesn’t track simple interests of users, instead it tracks by which means a user is most likely to be persuaded into doing something desired from the persuasion source [347]. Beside self-reporting survey, it is possible to gain data for individuals persuasive profile from tracking their online activities; and their interactions in social networks [346].
6.6 Summary

In this chapter, we discussed social influence and persuasion strategies and their potentials in filling the gap between industry and education through stimulating learners to adopt in-demand career. We presented our persuasive approach to support career adoption in higher education in coordination with human resource planning authorities and recruitment agencies. Our model proposed employing set of key users as a persuasive agent to propagate the desired behavior among other users. For purpose of identifying most influential key users in social network, we introduced a social mining technique to analyze social influence in CoP network using five different seeding strategies under two popular influence maximization models. Our technique allowed us to learn about important structural metrics (i.e. coreness); and to identify most influential individuals by each seeding strategy. We also introduced a model for stimulating selected key users to act as persuasive agents according to their specific persuasive profile.
Chapter 7: Conclusion and Future Works

In this chapter, we summarize the contributions of this dissertation and discuss some future research directions towards enhancing learning analytics in higher education context, and leveraging CoP-Network as a platform for collaborative learning and behavioral change.

7.1 Learning Analytics

Learning Analytics (LA) is widely viewed as important, but data use at most higher education institutions is still limited to reporting. LA focuses on measuring, collecting, analyzing data about learners and learning contexts to infer changes and improve learning outcomes. LA provide insights to support learning needs of individuals. It provides instructors and mentors with opportunities to carry out real-time analysis of learning activities. By performing retrospective analysis of learner data, predictive models can be created to evaluate future events and provide appropriate intervention, hence enabling instructors to adapt their teaching or initiate tutoring, tailored assignments and continuous assessment. Some of anticipated challenges associated with the implementation of LA in higher education include consolidating data from various resources; and developing well defined strategic objectives to guide the design and deploying of LA tools. We have discussed the conceptual and theoretical underpinnings of Big Data and LA analytics in higher education, and developed the argument to set the strategic objectives to: (1) prepare learners to meet the demands of the knowledge society; (2) ensure strong link between learning programs and industry practice; and (3) develop lifelong learning capacity to allow learners to acquire further qualifications or skills independently throughout their career path.

Toward implementing LA methods to achieve these objectives, we proposed the Career Readiness Data Warehouse as a multidimensional data structure that collects and stores career-related data from different sources inducing self-reporting, learner profile, market trends, online activities, and assessment feedbacks. We also introduced the Career Profile data structure to record both individual professional traits and career aspirations. We developed a scale to measure the lifelong learning capacity and professional skills referred
to as "Career Dispositions" of each learner as an indicator of preparation level towards practicing in knowledge-based market. We then implemented a diagnostic LA tool to analyze and visualize information on collected data; and deliver it to users through a portal design. Further, we proposed a Fuzzy Pairwise-Constraints K-Means (FPKM) algorithm as a predictive LA method to cluster learners of similar career patterns into interconnected Communities of Practice (CoPs) that are driven by current industrial needs. CoPs provide online social structure to supplement formal education with a virtual collaborative learning environment supervised by a professional mentor to bridge diverse viewpoints and instill a joint effort to leverage future workforce developments. CoP also provides indicators and means to intervene in order to positively affect career readiness.

Future work will involve developing LA methods to capture, analyze and aggregate learning data that is not interoperable such as learners administrative data, academic performance data, and classroom and online data. While integrating data sets from across a variety of unconnected systems can be extremely difficult, it offers better extensive insights that automatically lead to improved capabilities of LA models. Further and in alignment with the research work that is underway exploring data management and governance structures associated with LA in higher education, we aim to develop LA management model to explore the effectiveness of CoP on improving career preparation outcomes in relation to the set of skills required by the industry for each designated career. In order for the CoP to grow and have meaning, the individual members must be motivated to engage with it actively to create and maintain information flow; thus we expect learners to develop a changing understanding of practice over time by shifting from knowledge consumption only to knowledge creation through a social interaction process. For evaluating CoP effectiveness and supporting its sustainability, we propose developing a set of LA tools inspired by: (1) criteria to underpin the CoP of learners in the educational context (e.g. development of learners' reflective experience, encouragement of multidisciplinary knowledge sharing, and support learning through cognitive and practical apprenticeship; and (2) fundamental elements of successful online CoP (e.g. knowledge generating interactions, efficiency of
Involvement, connections to the world, and belonging and relationships).

7.2 Social Learning Communities Analytics

In this research work, we constructed a CoP-Network to support socialized learning by linking learners who share social interests beside their common career aspirations. Our similarity-based linking method allows learners to assign weight and order for their own attributes to determine how much other CoP participants are close to them. We also devised a triadic closure approach to enhance community structures in CoP-Network in order to optimize the learning performance of groups of learners. While structure based methods are widely used in the literature to enhance community detection, our method utilizes attribute-based data from Career Profile to close open triad in CoP-Network. Our community detection enhancement method (TC_CNM) showed an improved modularity value, a community dense measure, for CoP-Network of different sizes. This is because triadic closures increase the intra-cluster coherence among nodes forming enhanced densely connected community structures. Further, the networked CoP structure allows us to explore the power of Social Learning Analytics (SLA) that focuses on how learners build knowledge together in their cultural and social settings. It aims to identify social behaviors and patterns that signify effective processes in learning environments. One of the SLA categories is the social network analytics (SNA) that investigates networking process and properties of ties, relationships, roles and network formation to provide insights of how individuals develop and maintain these relationships to support learning. We discussed different SNA metrics and their usability; and provided a case study of SNA application on CoP-Network in a way that allows visual mapping and quantitative analysis.

Though our work in this area, we came to the open challenge of detecting overlap communities. This is relate to our application domain as communities in learning network are most likely to overlap as learners can belong to multiple CoPs at once. In a future work, we aim to explore community detection algorithms of overlapped communities. We also look to develop different categories of SLA to be able to define what accounts as learning
ties; and which interactions really promote learning in the social learning community as presented in the CoP-Network. One category—in particular—interests us: that is discourse analytics. Discourse analytics offers ways of exploring and understanding series of communicative events in the social environment and their influence on learning.

### 7.3 Social Influence Mining for Persuasive Technology

In this thesis, we differentiate between influence in persuasion paradigm and in social influence models. Influence in persuasion paradigm, is presented as detailed argumentation delivered to individual recipients in a context with only minimal social interaction. Social influence includes simple information about the source's position that is delivered in more complex social settings which involve interactions among participants. Accordingly, we proposed incorporating the concept of social influence into a persuasive model to develop a Behavioral Change Support System for Career Adoption (BCSS-CA). Towards this end, we reviewed the literature of persuasive technology to identify associated influence principles; and to explore recognized persuasive system design models required to develop effective BCSSs. We also surveyed social influence literature to present widely-used fundamental influence diffusion models (ICT and LT); and how they are employed to address the issue of influence maximization in social networks. We then presented our social mining method to extract the most influential nodes in CoP-Network using different strategy based on SNA metrics (i.e. centrality degree, betweenness degree, coreness). The extracted nodes by each strategy are used as initial set for influence propagation models to compare their influence performance on the CoP-Network. The results revealed the different effects of different seeding strategies on influence spread; and that nodes of highest coreness value have the highest impact in influencing other nodes under both models.

The identified most influential nodes in CoP-Network are hired as "persuasive agents" in our BCSS-CA. In order to assure that persuasive agents will deliver or influence the required behavior (i.e. career path adoption), we suggested using their Career Profile and Career Dispositions information to create their personalized Persuasive Profile. Persuasive
Profile tracks by which means an agent is most likely to be persuaded into adopting the desired behavior. In corresponding to the Fogg Behavioral Model (FBM), we provided and discussed with examples the matrix of mechanisms to ensure sufficient motivation, sufficient ability, and effective triggers for persuasive agents to adopt the target behavior. A future work aims to develop the BCSS-CA software using our persuasive agents; and track the diffusion of target behavior in CoP-Network. This will involve experimenting more strategies to find the most influential seed set for persuasive application.
Bibliography


[298] C. Malamed, “How to be effective at persuasion for learning.”


[346] C. Steiner, E. D. Mekler, and K. Opwis, “Personalised persuasion—what are the most effective user data for persuasion profiling?” 2013.
