THÈSE / UNIVERSITÉ DE RENNES 1
sous le sceau de l'Université Bretagne Loire

En Cotutelle Internationale avec
Northwestern Polytechnical University, Xi'an, Chine

pour le grade de

DOCTEUR DE L'UNIVERSITÉ DE RENNES 1

Mention : Informatique

Ecole doctorale MATISSE

présentée par

Kuang Zhou

préparée à l'unité de recherche UMR 6074 IRISA
Institut de Recherche en Informatique et Systèmes Aléatoires
Université de Rennes 1

Belief relational clustering and its application to community detection

Thèse soutenue à l'université de Rennes 1
le 05/07/2016

devant le jury composé de :

Thierry DENOEUX
Professeur, Université Technologique de Compiègne / *rapporteur*

Christine LARGERON
Professeur, Université Jean Monet / *rapporteur*

Frédéric DAMBREVILLE
Expert senior HDR, DGA-MI / *examinateur*

Zhun-ga LIU
Associate professor, Northwestern Polytechnical University / *examinateur*

Jacques NICOLAS
Directeur de recherche, INRIA / *examinateur*

Arnaud MARTIN
Professeur, Université de Rennes 1 / directeur de thèse

Quan PAN
Professeur, Northwestern Polytechnical University / directeur de thèse

BELIEF RELATIONAL CLUSTERING AND ITS APPLICATION TO COMMUNITY DETECTION

Kuang Zhou

A thesis submitted for the degree of Doctor of Philosophy

July 2016

Abstract

Clustering, also called unsupervised learning, is an important technique in the field of data mining. According to the type of data sets, clustering algorithms can be divided into two kinds. One is for object data in the distance space, where the objects to be clustered are described by feature vectors. The other is for proximity data, where only the relationship values such as similarities or dissimilarities between objects are known. The latter is a more general case, as the relationship could also be got for the data represented by feature vectors. On the contrary, many real-world data sets can only be represented by relational data for which object-based clustering algorithms could not be applied directly.

Communities are groups of nodes (vertices) which probably share common properties and/or play similar roles within the graph. They can extract specific structures from complex networks, and consequently community detection has attracted considerable attention crossing many areas where systems are often represented as graphs. Community detection is in fact a clustering problem on graphs, and the available information in this problem is often in the form of similarities or dissimilarities (between nodes).

We consider in this work to represent graphs as relational data, and propose models for the corresponding relational data clustering. Four approaches are brought forward to handle the community detection problem under different scenarios.

We start with a basic situation where nodes in the graph are clustered based on the dissimilarities and propose a new c-partition clustering approach named Median Evidential C-Means (MECM) algorithm. This approach extends the median clustering methods in the framework of belief function theory. Moreover, a community detection scheme based on MECM is presented. The proposed approach could provide credal partitions for data sets with only known dissimilarities. The dissimilarity measure could be neither symmetric nor fulfilling any metric requirements. It is only required to be of intuitive meaning. Thus it expands application scope of credal partitions. In addition, some practical issues about how to apply the method into community detection problems such as how to determine the initial prototypes and the optimum community number in the sense of credal partitions are discussed. This makes the approach appropriate for graph partitions and enables us to gain a better understanding of the analysed networks, especially for the uncertain and imprecise structures.

In MECM, one single representative object in the original data set is used to describe each of the individual classes. However, in some cases the way of using only one node to describe a community may not be sufficient enough. In order to capture various aspects of the community structures, more members rather than one should be referred as the prototypes of an individual group. Motivated by this idea, a Similarity-based Multiple Prototype (SMP) community detection approach is proposed. The centrality values are

used as the criterion to select multiple prototypes to characterize each community. The prototype weights are derived to describe the degree of representativeness of objects for their own communities. Then the similarity between each node and community is defined, and the nodes are partitioned into divided communities according to these similarities. Crisp and fuzzy partitions could be obtained by the application of SMP.

Following, we extend SMP in the framework of belief functions to get credal partitions so that we can gain a better understanding of the data structure. The prototype weights are incorporate into the objective function of evidential clustering. The mass membership and the prototype weights could be updated alternatively during the optimization process. In this case, each cluster could be described using multiple weighted prototypes. As we will show, the prototype weights could also provide us some useful information for structure analysis of the data sets.

With the increasing size of social networks in real world, community detection approaches should be fast. The Label Propagation Algorithm (LPA) is known to be one of the near-linear solutions and benefits of easy implementation, thus it forms a good basis for efficient community detection methods. We extend the original update rule and propagation criterion of LPA in the framework of belief functions. A new community detection approach, called Semi-supervised Evidential Label Propagation (SELP), is proposed as an enhanced version of the conventional LPA. One of the advantages of SELP is that it can take use of the available prior knowledge about the community labels of some individuals. This is very common in real practice. For instance, in the co-authorship network, some domain experts are very easy to be labeled as their research interests are well-known to everyone. In SELP, the nodes are divided into two parts. One contains the labeled nodes, and the other includes the unlabeled ones. The community labels are propagated from the labeled nodes to the unlabeled ones step by step according to the proposed evidential label propagation rule.

The performance of the proposed approaches is evaluated using benchmark graph data sets and generated graphs. Our experimental results illustrate the effectiveness of the proposed clustering algorithms and community detection approaches.

Résumé

Le *clustering*, également appelé classification non supervisée, est une technique importante dans le domaine de l'exploration de données. Selon le type d'ensembles de données, les algorithmes de classification peuvent être divisés en deux types. Le premier concerne les objets dans un espace muni d'une distance, où les objets devant être groupés sont décrits par des vecteurs de caractéristiques. Le second concerne les données de proximité, où seuls les valeurs relationnelles telles que les similitudes ou les différences entre les objets sont connus. Ce dernier est un cas plus général, la relation peut également être obtenue pour les données représentées par vecteurs de caractéristiques, mais de nombreux ensembles de données du monde réel peuvent être représentés seulement par des données relationnelles pour lequel des algorithmes de clustering à base d'objets ne peuvent pas être appliquées directement.

Les communautés sont des groupes de œuds (sommets) qui partagent probablement des propriétés communes et/ou jouent des rôles similaires dans le graphe. Ils peuvent extraire des structures spécifiques des réseaux complexes, et par conséquent la détection de ces communautés a été étudiée dans de nombreux domaines où les systèmes sont souvent représentés sous forme de graphes. La détection de communautés est en fait un problème de classification (ou *clustering*) sur les graphes, et linformation disponible dans ce problème est souvent sous la forme de similitudes ou de différences (entre les nœuds).

Nous considérons dans ce travail les graphes comme des données relationnelles et proposons des modèles pour la classification de ces données relationnelles correspondantes. Quatre approches sont présentées pour traiter le problème de la détection de clusters selon différents scénarios.

Nous commençons par une situation de base où les nœuds dans le graphe sont regroupés selon leurs similarités et proposons une nouvelle approche de clustering en c-partition nommée algorithme Median Evidential C-Means (MECM). Cette approche étend la méthode de classifciation par médiane dans le cadre de la théorie des fonctions de croyance. En outre, une détection de communautés fondée sur l'approche MECM est également présentée. L'approche proposée permet de fournir des partitions credales selon des similarités avec seulement des données connues. La mesure de dissimilarité pourrait être ni symétrique et même ne comporter aucune exigences de métriques. Elle est simplement intuitive. Ainsi, elle élargit la portée d'applications des partitions credales. De plus, certaines questions pratiques sur la façon d'appliquer la méthode dans la détection de communautés à des problèmes tels que la façon de déterminer les premiers prototypes et le nombre optimal de communautés dans le sens de partition credale sont discutées. Cela rend l'approche appropriée pour les partitions de graphes et nous donne une meilleure compréhension des réseaux analysés, en particulier pour les structures incertaines et imprécises.

Nous considérons dans ce travail les graphes comme des données relationnelles et proposons des modèles pour la classification de ces données relationnelles correspondantes. Quatre approches sont présentées pour traiter le problème de la détection de clusters selon différents scénarios.

Pour l'approche MECM, un seul objet représentatif de l'ensemble de données d'origine est utilisé pour décrire une classe individuelle. Cependant, dans certains cas, le fait d'utiliser un seul nœud (individu) pour décrire une communauté peut ne pas être suffisant. Afin de saisir les divers aspects des structures de communautés, nous pouvons avoir besoin de plusieurs nœuds plutôt qu'un seul pour représenter un prototype représantant un groupe d'individus. Motivée par cette idée, une approche de détection de communautés fondée sur le Similarity-based Multiple Prototype (SMP) est proposée. Les valeurs de centralité sont utilisées comme critère pour sélectionner plusieurs nœuds (prototypes) pour caractériser chaque communauté, et les poids des prototypes sont considérés pour décrire le degré de représentativité des objets liés à leur propre communauté. Ensuite, la similarité entre chaque nœud et les communautés est définie. Les nœ)uds sont divisés pour former des communautés selon leurs similarités. Les partitions nettes et floues peuvent être obtenues par l'approche SMP.

Ensuite, nous étendons l'approche SMP au cadre des fonctions de croyance pour obtenir des partitions credales de sorte que l'on puisse obtenir une meilleure compréhension de la structure des données. Les poids du prototype sont incorporés dans la fonction dobjectif de la communauté. La composition de masse et les poids des prototypes ont pu être mis à jour alternativement pendant le processus d'optimisation. Dans ce cas, chaque groupe peut être décrit en utilisant de multiples prototypes pondérés. Comme nous allons le montrer, les poids des prototypes peuvent également nous fournir des informations utiles pour l'analyse des données.

Avec la taille croissante des réseaux sociaux dans le monde réel, lapproche de la détection de communautés doit être rapide et précise. L'approche *Label Propagation Algorithm* (LPA) est connue pour être l'une des solutions quasi-linéaire avec les avantages d'une mise en œuvre facile. Ainsi elle forme une bonne base pour les méthodes de détection de communautés. Dans le dernier chapitre, la règle de mise à jour et le critère de propagation du LPA sont étendus aux fonctions de croyance. Une nouvelle approche de détection de communautés, appelée *Semisupervised Evidential Label Propagation* (SELP) est proposée comme une version améliorée de la méthode LPA conventionnelle. L'un des avantages de l'approche SELP est quelle permet de tenir compte de la connaissance préalable disponible sur les étiquettes des communautés de certains individus. Ceci est très courant dans la pratique réelle. Par exemple, dans le réseau des co-auteurs, certains experts du domaine sont très facilement labélisables selon leurs intérêts de recherche bien connus de tous. Dans la méthode SELP, les nœuds sont divisés en deux partis. Certains contiennent des nœuds labélisés et les autres des nœuds non labélisés. Les labels sont propagés depuis les nœuds labélisés à ceux non labélisés,

étape par étape en utilisant la règle crédibiliste de propagation de labels proposée.

Les performances des approches proposées sont évaluées en utilisant les graphes de référence des ensembles de données et des graphes générés. Nos résultats expérimentaux illustrent l'efficacité des algorithmes de classification proposés et des méthodes de détection de communautés.

Introduction

Malgré des études approfondies sur la détection de communautés pour regrouper et gérer efficacement divers types d'ensembles de données sous forme de graphes, cela reste encore une tâche très difficile. Les principaux défis pour développer un moteur de *clustering* efficace pour les graphes sont discutés ci-dessous :

Le recouvrement Dans de nombreux algorithmes de classification, tels que C-Means (CM) et les approches de classification hiérarchique, un objet est uniquement affecté à une seul classe (*custer*). Toutefois, dans des cas réels, il n'est pas toujours clair qu'un objet appartienne à une classe plutôt qu'ue autre. Les valeurs d'adhésion floue sont adoptées dans *Fuzzy C-Means* (FCM) pour décrire l'imprécision de la classe des objets. La composition d'un objet est en fait une probabilité de distribution sur le cadre de discernement Ω. Cependant, il existe un certain nombre de problèmes qui ont déjà été traités dans des partitions floues. Par exemple, il ne pourrait pas distinguer les valeurs aberrantes des objets incertains en raison de l'utilisation de la distance relative lors du calcul de l'appartenance floue. Supposons $c = 2$ (voir Figure 1). Les objets x_k et x_j sont équidistants des deux classes et la composition dans chaque classe pour les deux objets seront identiques (0.5), indépendamment de la valeur absolue de la distance de x_k à partir des deux centres de gravité (ainsi que des autres objets de l'ensemble des données d'origine). Le problème est que cela peut créer des valeurs aberrantes (objets bruités), loin mais équidistantes de la structure centrale des deux classes. Ces objets peuvent néanmoins être donnés avec une égale appartenance aux deux classes, mais il semble beaucoup plus naturel que ces objets aient une appartenance très faible à lune ou l'autre des classes. Pour remédier à ce problème, d'autres méthodes de raisonnement dincertitudes doivent être adoptées pour être une alternative de la théorie des ensembles flous pour effectuer l'analyse de la classification. La théorie des fonctions de croyance peut être un bon choix.

Différente de la théorie des probabilités et des ensembles flous, la théorie des fonctions de croyance est définie sur un ensemble des disjonctions de Ω, le cadre de discernement. En conséquence, la partition obtenue par les fonctions de croyance permet à chaque objet d'avoir une appartenance à plusieurs classes avec divers degrés, pouvant ainsi représenter le recouvrement entre les classes. L'ensemble vide peut être pris en considération pour détecter des valeurs aberrantes. Par ailleurs, il est possible de décrire le degré qu'un objet appartienne à une méta-classe (classe imprécise, la classe, comprenant plus d'une classe spécifique) en exprimant nos connaissances sur l'ensemble des disjonction de Ω. Dans ce travail, nous proposons différents algorithmes de classification dans le cadre des fonctions de croyance pour modéliser la structure incertaine des réseaux analysés.

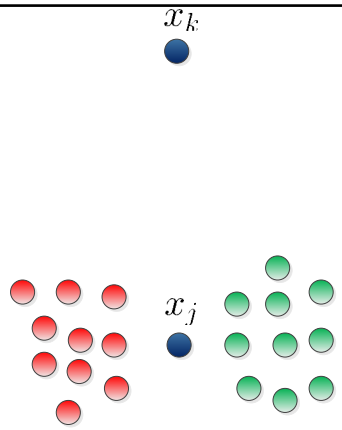
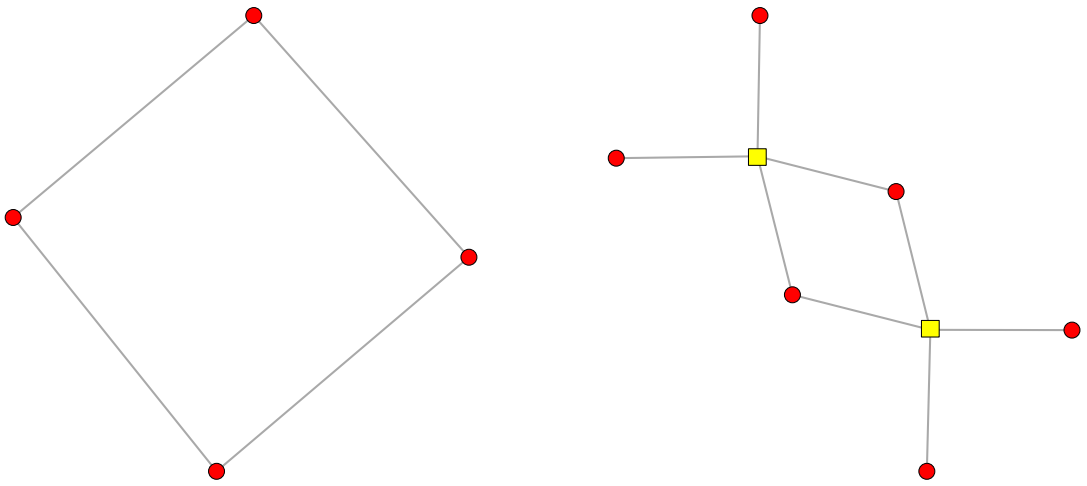


Figure 1. Le problème des partitons floues.

Les prototypes multiples Pour les réseaux sociaux avec de bonnes structures de communauté, le centre d'un groupe est susceptible d'être une personne, qui joue le rôle de chef de file de la communauté. Autrement dit, l'un des membres du groupe est mieux d'être sélectionné comme l'initiateur, plutôt que le centre de toutes les personnes. Cependant, dans certains cas, la manière d'utiliser un seul nœud pour décrire une communauté peut ne pas être suffisante. Pour illustrer cette limitation d'avoir un seul prototype représentant une classe, nous utilisons deux structures communautaires simples présentées dans la Figure 2.



a. Communauté 1 b. Communauté 2

Figure 2. Deux petite structures communautaires.

La premièr communauté se compose de quatre membres tandis que la seconde en a huit. On peut constater que, sur la communauté 1, il est déraisonnable de décrire la structure du cluster en utilisant un des quatre nœuds du groupe, puisquaucun des quatre nœuds ne pourrait être considéré en tant que représentant plus approprié que les trois autres. Dans la communauté 2 de la Figure 2, deux membres (marquée par des carrés jaunes) sur les huit sont égaux et sont raisonnables pour être sélectionnés en tant que représentant de la communauté. Cela signifie que choisir l'un deux peut conduire ne pas détecter l'ensemble complet de tous les nœuds représentatifs. À partir de ces exemples, nous pouvons voir que pour certains réseaux, nous pouvons avoir besoin de plusieurs membres plutôt qu'un seul à renvoyer comme prototype d'un groupe d'individus et ainsi mieux saisir les différents aspects de la structure de la communauté. Dans ce travail, nous proposons deux sortes de classification fournissant plusieurs prototypes, fondées sur la structure relationnelle des données. Les prototypes représentatifs sont configurés pour être les objets des ensembles de données d'origine. Ces modèles proposés peuvent tre utilisés pour obtenir des partitions nettes, floues et crédales.

Les outliers Bien que la plupart des nœuds dans un graphe suivent une distribution commune dans une communauté, certains objets peuvent dévier significativement du modèle. Les *outliers* (ou anomalies) font souvent référence à des objets dont les caractéristiques aberrantes dévient significativement de la majorité des données. Ils ne sont pas nécessairement solitaires, et ils pourraient avoir un lien négligeable avec certaines communautés. Il est d'un grand intérêt de détecter ces valeurs aberrantes dans les réseaux pour débruiter les données améliorant ainsi la qualité de la structure de la communauté pour une analyse ultérieure. Trouver les valeurs aberrantes d'une communauté est un problème important, mais na pas reçu suffisamment d'attention dans le domaine de l'analyse des réseaux sociaux. Dans ce travail, certaines classes sont spécialement conçues pour les valeurs aberrantes ou des données bruitées. Les objets qui sont significativement différents des autres peuvent être regroupés dans la classe des valeurs aberrantes.

L'information préalable Avec la disponibilité croissante des informations sur le réseau, il y a une quantité significative de connaissances antérieures disponibles sur les communautés dans les réseaux sociaux. Spécifiquement, la vrai classe d'appartenance de certains nœuds peut être connue à l'avance. Par exemple, dans un réseau d'une communauté de coauteurs, il peut être possible d'étiqueter un petit sous-ensemble de chercheurs en fonction de leurs intérêts de recherche. Dans une application de réseau social, il peut être souhaitable d'étiqueter certains nœuds selon leur affinité à certains produits par exemple. Utiliser ce type d'information pour améliorer les résultats de la détection de communauté est un problème important dans le champ de l'analyse des données d'un réseau social. Dans ce travail, une approche de détection de communautés

semi-supervisée, qui peut tirer profit de la connaissance préalable disponible, est mise en œuvre pour améliorer la précision des modèles de détection de la communauté l'aide de quelques échantillons marqués.

Les contributions Le but de cette étude est donc d'améliorer l'analyse de la classification des réseaux complexes en trouvant une représentation appropriée mathématique des données issues de graphes, et le développement efficace dalgorithmes de classification sur la base des représentations de données correspondantes. Nous traitons les données issues de graphes sous forme de données relationnelles, qui peuvent impliquer des relations entre paires de nœuds. Ce modèle fondé sur la relation offre une plus grande flexibilité et efficacité que le traditionnel modèle vectoriel pour représenter plusieurs groupes de données qui pourraient décrire les données analysées définies à partir des différentes perspectives. Foné sur les fonctions d'objectif, les modèles de classification crédibiliste et des algorithmes d'optimisation alternatifs, nous proposons différents modèles de classification des données relationnelles qui permettent d'améliorer les performances de classification tout en fournissant une meilleure compréhension de la structure interne des ensembles de données.

Nous avons ainsi :

- Proposer une nouvelle approche de classification des données relationnelles appelée MECM qui caractérise chaque groupe en utilisant un objet représentatif. L'approche proposée peut fournir des partitions crédales des donnée uniquement à partir des dissimilarité entre les données. La mesure de dissimilarité peut ne pas ê tre une métrique. Elle est simplement intuitive. Ainsi, elle élargit la portée d'applications des partitions credales. De plus, certaines questions pratiques sur la façon d'appliquer la méthode dans la détection de communautés à des problèmes tels que la façon de déterminer les premiers prototypes et le nombre optimal de communautés dans le sens de partition credale sont discutées. Cela rend l'approche appropriée pour les partitions de graphes et nous donne une meilleure compréhension des réseaux analysés, en particulier pour les structures incertaines et imprécises.

- Proposer une approche de détection de communautés fondée sur les similarités permettant d'obtenir des prototypes multiples nommée *Similarity-based Multiple Prototype* (SMP). Bien qu'il existe des méthodes de classification multi-prototypes pour les ensembles de données classiques, il y a peu d'approches adaptées aux problèmes de détection de communautés. Une nouvelle approche de représentation des communautés en utilisant de multiples prototypes est proposée ici. De plus, le concept de pondération par prototype est présenté, qui décrit le degré de représentativité du membre correspondant de son groupe. À laide du poids du prototype, l'approche SMP fournit une description plus détaillée et plus

efficace pour chaque communauté. Cela nous permet d'acquérir une connaissance approfondie de la structure interne de la communauté, qui est également très important et utile pour l'analyse du réseau. Dans l'approche proposée de détection de la communautés, différents types de mesures de similarité et de centralité pourraient être adoptées, ce qui est plus pratique et flexible dans des applications réelles.

- Proposer l'approche SMP étendue dans le cadre des fonctions de croyance. Deux versions de l'approche de classification *Evidential C-Medoids* (ECMdd), nommées sECMdd et wECMdd sont proposées pour produire une partition credale optimale, en utilisant un seul médoïde et de multiples medoïdes pondérés respectivement pour représenter une classe. Comme la méthode MECM, l'approche ECMdd est dédiée aux ensembles de données de proximités. Cependant, l'approche ECMD permet de plus à chaque classe d'être décrite par plus dun motif représentatif. Cela permet au modèle de représenter des structures de classes complexes. Les poids des prototypes sont incorporés dans la fonction objective du modè de classification, et les deux fonctions de masse représentant les poids d'adhésion et de prototypes sont mises à jour alternativement dans le processus d'optimisation.

- Proposer une approche de classification semi-supervisée en utilisant une nouvelle approche nommée *Evidential Label Propagation strategy* (SELP) pour les ensembles de données sous forme de graphes.LE principal avantage de l'approche SELP est qu'elle permet de tenir compte de certaines connaissens *a priori* afin de guider le processus de détection de communautés. Ces informations *a priori* sur les communautés sont représentées par des labels exprimées sous la forme de fonctions de masse initiales. Ainsi une nouvelle règle de propagation de labels crédibilistes est proposée pour propager les labels initiaux aux données aux labels inconnus. Les valeurs aberrantes peuvent être identifiées par l'ensemble vide. L'approche pourrait aussi être appliquée sur des ensembles de données classiques non-sphériques.

Acknowledgements

I am indebted to many people for their help and support during my Ph.D. study and research at team DRUID, IRISA, University of Rennes 1 in France and at the key Laboratory of Information Fusion Technology (LIFT) of Northwestern Polytechnical University (NPU) in China. I would like to thank the China Scholarship Council for supporting my study in France.

I would like to express my sincere thanks to my supervisor in the University of Rennes 1, Prof. Arnaud Martin. He has provided me with a lot of invaluable guidance, frequent meetings and discussions. Moreover, I have also learned a great deal from his instructions for developing a rigorous way of research. Furthermore, I thank you for introducing me the best and the brightest in our field, and for showing me that being a professor is the best job in the world. I will forever treasure my experiences in Lannion.

I would also like to gratefully thank my supervisor at NPU, Prof. Quan Pan. He always gives me some far-sighted suggestions for my study and life. He has provided me with great support for my study in France. He leads me to see the research fronts in our field. He has taught me a lot not only about knowledge but also how to be a researcher. Special thanks for you, for your productive comments, fruitful discussions and advices.

This dissertation would not have been possible without the contributions, help, and friendship of all the researchers in team DRUID and laboratory LIFT. Thanks for their help and kindness. A special thank for all colleagues in IUT of Lannion for their encouragement, kindness and support. I also thank all my friends in Lannion and Xi'an for their presence and help when I encounter difficulties in work or in life. I really appreciate their advices and jokes.

I am grateful for the ceaseless support and love of my parents. Their unwavering faith and confidence in me and my abilities have shaped me into the person I am today. I wish to express my deepest appreciation to my wife, for her love, continuous encouragement and understanding these last years.

Contents

List of Tables

List of Figures

Abbreviations and notations

In the following, a list as exhaustive as possible of abbreviations and notations used in this thesis:

Clustering

- $\boldsymbol{X}$: the set of samples in the analyzed data set;
- x_i or $\boldsymbol{x}_i$: denote the samples in the analyzed data set. To make it clear, for object data sets, we use $\boldsymbol{x}_i$ to represent the sample, while x_i for relational data sets;
- c: the number of classes/communities in the analyzed data set;
- n: the number of objects in the analyzed data set;
- $d_{ij} = d(x_i, x_j)$: denotes the dissimilarity between objects x_i and x_j (or nodes n_i and n_j).
- $\overline{d}_{ij}$: For crisp or fuzzy clustering algorithms, it is the dissimilarity between object x_i and class ω_j; While for credal partitions, it is the dissimilarity between object x_i and class with focal set A_j;
- $\overline{s}_{ij}$: the similarity between objects x_i and class ω_j in SMP algorithm;
- $m_{ij}, m(\{\omega_j\})$: is the mass membership of object x_i on the focal set A_j;
- $\boldsymbol{m}_i$: is the mass membership vector of object x_i;
- u_{ij}: is the fuzzy membership of object x_i on class ω_j;
- v_j or $\boldsymbol{v}_j$: is the prototype of specific class ω_j; For object data sets, it is usually a vector on the metric space, so we use the bolded variable $\boldsymbol{v}_j$. While for relational data sets, v_j is adopted.
- $\overline{\boldsymbol{v}}_j$: is the prototype of the class that is associated with focal set A_j in ECM;
- V_{rj}: the prototype weight of object x_j for class ω_r in SMP algorithm;
- v_k^{Ω}: the prototype of specific class $A_j = \{\omega_k\}$ in sECMdd algorithm;

- $v_k^{2^\Omega}$: the prototype of the (imprecise or specific) class that associated with A_k in sECMdd algorithm;
- $v_{ki}^{2^\Omega}$: the prototype weight of object x_i for the class that associated with A_k in wECMdd algorithm;

Belief functions

- Ω: is the frame of discernment;
- S: is the number of sources;
- ω_1, ω_2, ..., ω_c: hypothesis in Ω; they are singletons;
- $\omega_{ij} \triangleq \{\omega_i, \omega_j\}$: the compound focal set including ω_i and ω_j;
- m, m_j: is a mass function, m is a mass function defined on any frame of discernment Ω; m_j is the mass function provided by a source j;

Graph

- $G(V, E)$: is the graph. V denotes the set of n nodes, while E is the set of edges;
- $n_1, n_2, \cdots, n_n$: is the n nodes in the graph;
- s_{ij}: is the similarity between nodes n_i and n_j;
- $\boldsymbol{A}$: is the adjacent matrix;
- a_{ij}: is the elements in the adjacent matrix, indicating whether there is a direct edge between nodes n_i and n_j in the graph;
- d_i: is the degree of node n_i;
- Q: is the modularity of a graph with given community structure;

Semi-supervised community detection

- m^x: the mass function for node n_x;
- m_j^x: the mass function for node n_x according to the information provided by node n_j;

1 Introduction

1.1 Overview

With the development of computer and Internet technologies, networks are everywhere in our common life. Graph models are useful in describing and analyzing many different kinds of relationships and interdependencies. In order to have a better understanding of organizations and functions in real-world networked systems, the community structure in the graph is a primary feature that should be taken into consideration. Communities, also called clusters or modules, are groups of nodes (vertices) which probably share common properties and/or play similar roles within the graph. They can extract specific structures from complex networks, and consequently community detection has attracted considerable attention crossing many areas from physics, biology, and economics to sociology, where systems are often represented as graphs. For instance, in social networks, society offers a wide variety of possible group organizations like families, working and friendship circles, villages, towns, nation and so on. In protein-protein networks, communities are likely to group proteins having the same specific function with the cell. In the co-authorship network, scholars dealing with the same or related topics may form a community.

Community detection is in fact a clustering problem on graphs. Clustering, also called unsupervised learning, is an essential and frequently performed task in pattern recognition for exploring underlying structures of data sets. The goal of clustering is to partition a set of objects $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\}$ into c (small value) groups $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$ in such a way that objects in the same cluster are as similar as possible while objects in different clusters are as dissimilar as possible. To measure the similarities (or dissimilarities), the patterns are described by either object data or relational data. Object data are described explicitly by a p-dimensional vector. For relational data, the available information arises from the pairwise similarities or dissimilarities, which are usually stored in an $n \times n$ matrix named the similarity (or dissimilarity) matrix. Relational clustering is more general in the sense that it is applicable no matter whether the objects to be clustered could be represented by numerical features or not. Even if the object could be represented by a feature vector, in relational clustering, more kinds of dissimilarity measures could be considered according to the characters of the analyzed data sets.

Until now, various clustering approaches have been proposed for different applications. A clustering algorithm is usually formulated by taking the requirements of the particular task and the nature of the data set to be handled into consideration. Current graph clustering approaches for networks mainly partition nodes based on the topological graph structure. In real applications of community detection, there are lots of uncertain or imprecise information in the data sets. For example, the boundary between clusters usually overlaps, and the links between some nodes may be imperfect. To improve the community detection results, new clustering methods should be developed to effectively take advantage of the imprecise information and deal with the uncertain parts of the data set carefully.

1.2 Motivations

Despite extensive studies on community detection, to effectively cluster and handle various kinds of graph data sets still remains as a very challenging task. Some of the main challenges to develop an effective clustering engine for graph data sets are discussed as follows:

Overlap In many clustering algorithms, such as C-Means (CM) and hierarchical clustering, one object is only assigned to a single cluster. However, in real cases, one pattern might span multiple classes; in other words, more than one theme or topic could be used to describe the object.

Fuzzy membership values are adopted in Fuzzy C-Means (FCM) to describe the uncertainty of the objects' class. The membership of an object is in fact a probability distribution over the frame of discernment Ω. However, there are some problems that have already been found in fuzzy partitions. For example, it could not distinguish outliers from uncertain objects due to the use of relative distance when calculating the fuzzy membership. Suppose $c = 2$ (see Figure 1.1). Both x_k and x_j are equidistant from the two classes, the membership of each cluster for the two objects will be the same (0.5), regardless of the large absolute value of the distance of x_k from the two centroids (as well as from the other objects in the original data set). The problem this creates is that outliers (noise objects), far but equidistant from the central structure of the two clusters, can nonetheless be given equal membership in both, when it seems far more natural that such objects should be given very low (or even no) membership in either cluster. To overcome this problem, other uncertainty reasoning methods should be adopted to be an alternative of fuzzy set theory to perform the task of clustering analysis. The theory of belief functions can be a good choice.

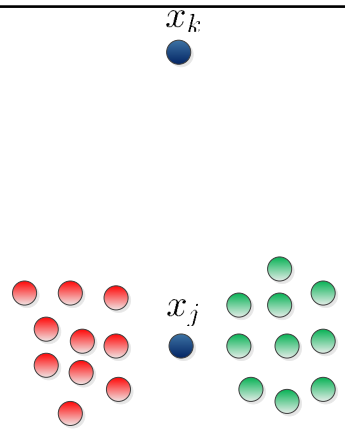
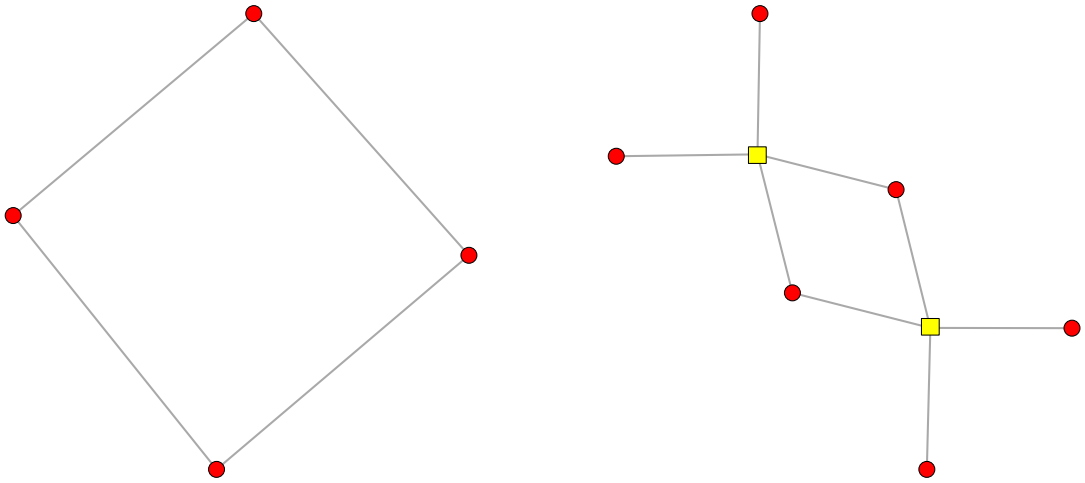


Figure 1.1: The problem of fuzzy partitions.

Different from the probability theory and fuzzy membership, belief functions are defined on the power set of the frame Ω. As a result, the partition provided by belief functions allows each pattern to belong to not only specific classes but also imprecise ones (a disjunctive combination of more than one single class) with various degrees so that overlap between clusters is able to be captured. The empty set can be considered for detecting outliers. In this work, we suggest different clustering algorithms in the framework of belief functions to capture the uncertain/overlap structure of the analyzed data sets.

Multiple prototypes For social networks with good community structures, the center of one group is likely to be one person, who plays the leader role in the community. That is to say, one of the members in the group is better to be selected as the seed, rather than the center of all the objects. However, in some cases the way of using only one node to describe a community may not be sufficient enough. To illustrate the limitation of one-prototype community representation, we use two simple community structures shown in Figure 1.2. The first community consists of four members while the second has eight. It can be seen that in the left community, it is unreasonable to describe the cluster structure using any one of the four nodes in the group, since no one of the four nodes could be viewed as a more proper representative than the other three. In the right community in Figure 1.2, two members (marked with yellow squares) out of the eight are equal reasonable to be selected as the representative of the community. This means choosing any one of them may fail to detect the complete set of all the candidate representative nodes. From these examples, we can see that for some networks, in order to capture various aspects of the community structures, we may need more members rather than one to be referred as the prototypes of an individual group. In this work, we suggest two kinds of multi-prototype based relational

clustering algorithms, where the representative prototypes are set to be the objects in the original data sets. These proposed models could be used to get hard, fuzzy and credal partitions.



a. Community 1 b. Community 2

Figure 1.2: Two small community's structures.

Outliers Although most of the nodes in a graph follow a common community distribution pattern, some certain objects may deviate significantly from the pattern. Outliers (or anomalies) often refer to aberrant objects whose characteristics deviate significantly from the majority of the data (Gao et al., 2010). They are not necessarily solitary, and they might have some negligible connection with some communities (Cao et al., 2013). It is of great value to detect such outliers in networks for de-noising data thereby improving the quality of the detected community structure and also for further analysis. Finding community outliers is an important problem but has not received enough attention in the field of social network analysis. In this work, some classes are specially designed for outliers or noisy data. The objects which are significantly different from other samples can be clustered into the outlier class.

Prior information With the growing availability of network information, there is a significant amount of prior knowledge available about the communities in social, communication and several other networks (Subbian et al., 2013). Specifically, the true community assignments of certain nodes may be known in advance. For instance, in a co-authorship community network, it may be possible to label a small subset of scholars based on their research interests. In a social network application, it may be desirable

to label some nodes according to their affinity to some products. How to use such kind of information to improve the community detection results is an important problem in the filed of network data analysis. In this work, a semi-supervised community detection approach, which can take advantage the available prior knowledge, is put forward to improve the accuracy of community detection models using few labeled samples.

1.3 Objectives

The goal of this study is to improve clustering analysis of complex networks by finding a suitable mathematical representation of the graph data, and developing effective clustering algorithms based on the corresponding data representations. We treat the graph data as relational data, which may involve pairwise relationships of nodes. This relation-based model provides a more flexible and effective way than the traditional vector-based model for representing multiple groups of data which could describe the analyzed data set from different perspectives. Based on the solid foundation of objective-function based belief clustering model and the alternating optimization algorithm, our proposed belief relational clustering model aim to achieve improvement in clustering performance while gain a better understanding of the internal structure of the data sets.

1.4 Contributions

In the process of achieving our objective, we have made several contributions to the field of clustering and community detection as follows:

Belief relational clustering A new belief relational clustering approach named MECM which characterizes each cluster using one representative object is proposed. The proposed approach could provide credal partitions for data sets with only known dissimilarities. The dissimilarity measure could be neither symmetric nor fulfilling any metric requirements. It is only required to be of intuitive meaning. Thus it expands application scope of credal partitions. In addition, some practical issues about how to apply the method into community detection problems such as how to determine the initial prototypes and the optimum community number in the sense of credal partitions are discussed. This makes the approach appropriate for graph partitions and gives us a better understanding of the analysed networks, especially for the uncertain and imprecise structures.

Similarity-based multiple prototype community detection A Similarity-based Multiple Prototype (SMP) community detection approach is proposed. Although there are some multi-prototype clustering methods for the classical data sets, there is little

such work for community detection problems. Here a new community representation mechanism using multiple prototypes is proposed. Moreover, the concept of prototype weights is presented, which describes the degree of representativeness of a member in its own group. With the help of prototype weights, SMP provides more sufficient description for each individual community. This enables us to gain a deep insight into the internal structure of a community, which we believe is also very important and useful for network analysis. In the proposed community detection approach, different kinds of similarity and centrality measures could be adopted, which makes it more practical and flexible in real applications.

Evidential multi-prototype clustering The SMP algorithm is extended in the framework of belief functions. The Evidential C-Medoids (ECMdd) clustering algorithm which can utilize multiple weighted medoids to represent a class is proposed to produce the optimal credal partition. Like MECM, ECMdd is for proximity data sets. However, ECMdd allows each class to be described by more than one representative pattern. This enables the model to capture complex class structure. The prototype weights are incorporated into the objective function of the clustering model, and both the mass membership and prototype weights are updated alternately in the optimization process.

Semi-supervised evidential community detection A Semi-supervised clustering approach using a new Evidential Label Propagation strategy (SELP) is proposed for graph data sets. The main advantage of SELP is that it can take limited supervised knowledge to guide the detection process. The prior information of community labels is expressed in the form of mass functions initially. Then a new evidential label propagation rule is adopted to propagate the labels from labeled data to unlabeled ones. The communities can be identified as well as the outliers. The approach could also be applied in non-spherical classical data sets.

1.5 Structure of the thesis

The thesis is organized in the following six chapters:

In Chapter 2, some related preliminary knowledge, including the theory of belief functions and other uncertainty theories, some classical clustering and community detection algorithms, the basic concepts of semi-supervised learning, will be introduced.

In Chapter 3, the belief relational clustering, MECM, will be presented. The dissimilarity between samples and classes (specific classes and imprecise classes) is defined, and then the objective function is given. The strategy for minimizing the cost function

is described. Moreover, several problems about the application of MECM on graph data sets are discussed.

Chapter 4 and Chapter 5 are about two multi-prototype based community detection approaches. The difference between the proposed two methods is that the former is for hard and fuzzy partitions, while the latter is for credal partitions. The multi-prototype representativeness mechanism in both methods will be described respectively.

Chapter 6 will discuss about how to utilize the limited supervised information when performing the task of community detection. The method for modeling the available prior knowledge and the strategy for prorogating labels are introduced in the framework of belief functions.

Finally, conclusions are drawn and some perspectives of this thesis are presented in Chapter 7.

2
Background

2.1 Overview

Due to the importance of clustering in various tasks of exploratory data mining, such as community discovery in social network analysis, distinguish between different types of tissue and blood in three-dimensional medical images, gene expression data analysis in bioinformatics, grouping of shopping items in market basket analysis, spatial database applications and so on, considerable efforts have been made on the development of clustering algorithms by scholars or engineers from multiple disciplines. Common open issues in clustering analysis including how to determine the number of clusters, robustness and scalability, together with additional challenges in clustering of some specific types of data sets, such as the problem of high-dimensionality for text data, imprecise and uncertain data, the difficulty of taking use of the available prior background knowledge, have been extensively investigated in the literature. Instead of discussing general issues and providing an exhaustive survey of various clustering approaches, here we concentrate specially on issues and clustering approaches that are related to community detection problems (specially partitioning-based clustering).

Specifically, the review of some related background knowledge in this chapter contains the following four components:

Belief function theory and other uncertainty theories

Among the variety of mathematical frameworks for modeling and managing uncertainty, probability theory is the predominant one and is very widely-applied. Other frameworks include fuzzy set theory (Zadeh, 1965), possibility theory (Dubois and Prade, 1988), rough set theory (Pawlak, 1982), and belief function theory (Shafer, 1976) – the model adopted in this thesis. The reason why there are multiple frameworks is that there are different types of uncertainty.

Generally, there are two kinds of uncertainty. One refers to epistemic uncertainty because it corresponds to beliefs held by an agent about the world. The other is aleatory uncertainty which is related to randomness and chance. Probability theory could only deal with the latter one. This former type of uncertainty is the result of ignorance rather than randomness. The Bayesian view is that ignorance can be adequately represented using probability theory by applying the principle of indifference, which assigns equal

probabilities to all possibilities. Consequently, equiprobability handles both equiprobable hypotheses and ignorant sources. In contrast, belief function theory distinguishes between these types of uncertainty and thus makes ignorance explicit. There are other types of uncertainty (*e.g.,* vagueness of natural language which can be described by fuzzy sets) but these are not considered in this thesis. Based on the review study of the different uncertainty theories, we will see the need of the introduction of belief function theory to effectively deal with the uncertainty information.

Clustering methods

This is the core technique in unsupervised data learning. Among the existing approaches to clustering, the objective function-driven or prototype-based clustering such as c-means and Gaussian mixture modeling is one of the most widely applied paradigms in statistical pattern recognition. These methods are based on a fundamentally very simple, but nevertheless very effective idea, namely to describe the data under consideration by a set of prototypes. They capture the characteristics of the data distribution (like location, size, and shape), and classify the data set based on the similarities (or dissimilarities) of the objects to their prototypes (Borgelt, 2006). Most of the proposed community detection approaches in this thesis belong to the family of prototype-based clustering. Thus we will review and compare some prototype-based clustering algorithms handling uncertainty.

Classical community detection algorithms

The current literature on the community identification problem and other closely related problems will also be surveyed in this section. These methods are used to compare in the conducted experiments in this thesis.

Semi-supervised learning

Semi-Supervised Learning (SSL) has grown into a large research area within machine learning. Unlike unsupervised learning, SSL could take advantage of the available prior information. But the supervised information is limited compared with the traditional supervised learning scheme. We will present the principle of SSL, and give a literature review of semi-supervised learning algorithms. Based on the review study of SSL, we then discuss the significance to introduce a semi-supervised community detection approach for graph data sets using the given labels of some (limited) individuals.

2.2 The theory of belief functions

The theory of belief functions is a mathematical theory that generalizes the the theory of probabilities by giving up the additivity constraint. As mentioned before, probability

theory could not distinguish equally probable events from the case of ignorance. In the theory of belief functions, uncertainty, incompleteness and ignorance are well modeled differently. In this theory, justified degrees of support are assessed according to an evidential corpus. Evidential corpus is the set of all evidential pieces of evidence held by a source that justifies degrees of support assigned to some subsets.

Let $\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}$ be the finite domain of reference, called the discernment frame. The c elements in Ω are non empty and mutually exclusive hypotheses related to a given problem. The belief functions are defined on the power set $2^\Omega = \{A : A \subseteq \Omega\}$. The function $m : 2^\Omega \to [0, 1]$ is said to be the *Basic Belief Assignment* (bba) on 2^Ω, if it satisfies:

$$\sum_{A \subseteq \Omega} m(A) = 1. \tag{2.1}$$

Every $A \in 2^\Omega$ such that $m(A) > 0$ is called a *focal element*. The difference with probability models is that masses can be given to any subsets of Ω instead of only to the atomic element of Ω. The *credibility* and *plausibility* functions are defined as in Eqs. (2.2) and (2.3) respectively:

$$\mathrm{Bel}(A) = \sum_{B \subseteq A, B \neq \emptyset} m(B), \quad \forall A \subseteq \Omega, \tag{2.2}$$

$$\mathrm{Pl}(A) = \sum_{B \cap A \neq \emptyset} m(B), \quad \forall A \subseteq \Omega. \tag{2.3}$$

Each quantity $\mathrm{Bel}(A)$ measures the total support given to A, while $\mathrm{Pl}(A)$ represents potential amount of support to A. Functions Bel and Pl are linked by the following relation:

$$\mathrm{Pl}(A) = 1 - m(\emptyset) - \mathrm{Bel}(\overline{A}), \tag{2.4}$$

where $\overline{A}$ denotes the complement of A in Ω. Some particular mass functions will be introduced in the following.

A *categorical mass function* is a normalized mass function which has a unique focal element $A^\star$. This kind of mass functions can be defined as:

$$m(A) = \begin{cases} 1 & \text{if } A = A^* \subset \Omega \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

A *vacuous mass function* is a particular categorical mass function focused on Ω. It is a special kind of categorical mass functions with a unique focal element Ω. This type of mass functions is defined as follows:

$$m(A) = \begin{cases} 1 & \text{if } A = \Omega \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

Vacuous mass function emphasizes the case of *total ignorance*.

A *Bayesian mass function* is a mass function which all focal elements are elementary hypotheses, *i.e.* the focal elements are all singletons. It is defined as follows:

$$m(A) = \begin{cases} a \in [0,1] & \text{if } |A| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2.7}$$

As all focal elements are single points, this mass function is a *probability distribution* over frame Ω. Specially, if a Bayesian mass function is categorical, it describes that there is no uncertainty at all and we are completely sure about the state of the concerned variable.

When a source of evidence is not completely reliable, a discounting operation proposed by Shafer (1976) and justified by Smets (1993) can be applied to the associated bba. Suppose the reliability degree of mass function m is represented by $\alpha \in [0,1]$, then the discounting operation is given by:

$$m^{'}(A) = \begin{cases} \alpha \times m(A) & \forall A \subset \Theta, \\ 1 - \alpha + \alpha * m(\Omega) & \text{if } A = \Omega. \end{cases} \tag{2.8}$$

If $\alpha = 1$, the evidence is completely reliable and the bba will remain unchanged. On the contrary, if $\alpha = 0$, the evidence is completely unreliable, and we have $m(\Omega) = 1$. This is the so-called vacuous belief function representing total ignorance.

A belief function on the credal level can be transformed into a probability function by Smets method (Smets, 2005). In this algorithm, each mass of belief $m(A)$ is equally distributed among the elements of A. This leads to the concept of *pignistic probability*, BetP, defined by

$$\mathrm{BetP}(\omega_i) = \sum_{\omega_i \in A \subseteq \Omega} \frac{m(A)}{|A|(1 - m(\emptyset))}, \tag{2.9}$$

where $|A|$ is the number of elements of Ω in A. Pignistic probabilities, which is a probability distribution over the frame Ω, can easily help us make a decision. In fact, belief functions provide us many decision-making techniques not only in the form of probability measures. For instance, a pessimistic decision can be made by maximizing the credibility function, while maximizing the plausibility function could provide an optimistic one (Martin and Quidu, 2008). Another criterion (Appriou's rule) (Martin and Quidu, 2008) considers the plausibility functions and consists in attributing the class A_j for object i if

$$A_j = \arg\max_{X \subseteq \Omega} \{m_i(X)\mathrm{Pl}_i(X)\}, \tag{2.10}$$

where

$$m_i(X) = K_i \lambda_X \left(\frac{1}{|X|^r} \right). \tag{2.11}$$

In Eq. (2.10) $m_i(X)$ is a weight on $\mathrm{Pl}_i(X)$, and r is a parameter in $[0,1]$ allowing a decision from a simple class ($r=1$) until the total ignorance Ω ($r=0$). The value λ_X allows the integration of the lack of knowledge on one of the focal sets $X \subseteq \Omega$, and it can be set to be 1 simply. Coefficient K_i is the normalization factor to constrain the mass to be in the closed world:

$$K_i = \frac{1}{1 - m_i(\emptyset)}. \tag{2.12}$$

How to combine efficiently several bbas coming from distinct sources is a major information fusion problem in the belief function framework. Many rules have been proposed for such a task. Here we just briefly recall how some most popular rules are mathematically defined.

When information sources reliable, the used fusion operators can be based on the conjunctive combination. If bbas $m_j, j = 1, 2, \cdots, S$ describing S distinct items of evidence on Ω, the included result of the *conjunctive rule* (Smets and Kennes, 1994) is defined as

$$m_{\mathrm{conj}}(A) = (\textcircled{\cap}_{j=1,\cdots,S} m_j)(X) = \sum_{Y_1 \cap \cdots \cap Y_S = A} \prod_{j=1}^{S} m_j(Y_j), \tag{2.13}$$

where $m_j(Y_j)$ is the mass allocated to Y_j by expert j. To apply this rule, the sources are assumed reliable and cognitively independent.

Another kind of conjunctive combination is *Dempster's rule* (Dempster, 1967). Assuming that $m_{\mathrm{conj}}(\emptyset) \neq 1$, the result of the combination by Dempster's rule is

$$m_{\mathrm{DS}}(A) = \begin{cases} 0 & \text{if } A = \emptyset, \\ \dfrac{m_{\mathrm{conj}}(A)}{1 - m_{\mathrm{conj}}(\emptyset)} & \text{otherwise.} \end{cases} \tag{2.14}$$

The item

$$\kappa \triangleq m_{\mathrm{conj}}(\emptyset) = \sum_{Y_1 \cap \cdots \cap Y_S = \emptyset} \prod_{j=1}^{S} m_j(Y_j)$$

is generally called Dempster's degree of conflict of the combination or the inconsistency of the combination.

2.3 Some links with other uncertainty theories

In this section some other uncertainty theories (including possibility theory, fuzzy set theory, and rough set theory) and the links with belief function theory are presented.

2.3.1 Possibility theory

Possibility theory is another popular choice for representing uncertain information (Dubois and Prade, 1988; Liu, 2006). At the semantic level, a basic function in possibility theory is a possibility distribution denoted as π which assigns each possible world in the frame of discernment Ω with a value in $[0, 1]$ (or a set of graded values). From a possibility distribution, two measures are derived, a possibility measure (demoted as Π) and a necessity measure (denoted as N). The former estimates to what extent the true event is believed to be in the subset and the latter evaluates the degree of necessity that the subset is true. The relationships between π, Π, and N are as follows.

$$\Pi(A) = \max\left(\{\pi(\omega)|\omega \in \Omega\}\right), \quad N(A) = 1 - \Pi(A). \tag{2.15}$$

$$\Pi(2^{\Omega}) = 1, \quad \Pi(\emptyset) = 0. \tag{2.16}$$

$$\Pi(A \cup B) = \max\left(\Pi(A), \Pi(B)\right), \quad N(A \cap B) = \min\left(N(A), N(B)\right) \tag{2.17}$$

A belief function is called a consonant function if its focal elements are nested (Shafer, 1976). That is, if $A_1, A_2, \cdots, A_n$ are focal elements then $A_1 \subset A_2 \subset \cdots \subset A_n$. Suppose the credibility function and plausibility function of the consonant bba are Bel and Pl, we have

$$\mathrm{Bel}(A \cap B) = \min\left(\mathrm{Bel}(A), \mathrm{Bel}(B)\right), \quad \mathrm{Pl}(A \cup B) = \max\left(\mathrm{Pl}(A), \mathrm{Pl}(B)\right). \tag{2.18}$$

These two properties are exactly the requirements of necessity and possibility measures in possibility theory. Necessity and possibility measures are special cases of credibility and plausibility functions when the focal sets of bba are nested. The theory of belief functions thus, in a sense, is more general than the theory of possibilities.

2.3.2 Fuzzy set theory

In mathematics, fuzzy sets are sets whose elements have degrees of membership. In classical set theory, the membership of an element in a set is either 1 or 0 (either belongs or does not belong to the set). By contrast, fuzzy set theory permits the gradual assessment of the membership of elements in a set with the help of a membership function valued in the interval $[0, 1]$. The classical bivalent sets are usually called crisp (hard) sets. Fuzzy sets generalize classical sets, since the indicator functions of classical sets are special cases of the membership functions of fuzzy sets, if the latter only take values 0 or 1.

Fuzzy sets represent vagueness of concepts as typically expressed by natural language, while belief functions adequately model uncertainty induced by partial evidence. They are not competing but complementary theories that model different aspects of imperfect information.

2.3.3 Rough set theory

In computer science, a rough set (Pawlak, 1982) is a formal approximation of an imprecise concept or knowledge by available information or knowledge databases in terms of a pair of sets which give the lower and the upper approximation of the original set.

Let Ω be a finite non-empty universe of discourse and $R \subseteq \Omega \times \Omega$ a binary equivalence relation on Ω. Given an arbitrary set $X \subseteq \Omega$, it may be impossible to describe X precisely using the equivalence classes of R. In this case, one may characterize X by a pair of lower and upper approximations:

$$\underline{R}(X) = \{x \in \Omega \mid [x] \subseteq X\} = \bigcup \{[x] \mid [x] \subseteq X\}, \tag{2.19}$$

$$\overline{R}(X) = \{x \in \Omega \mid [x] \cap X \neq \emptyset\} = \bigcup \{[x] \mid [x] \cap X \neq \emptyset\}, \tag{2.20}$$

where $[x]$ is the R-equivalence class containing x. The lower approximation $\underline{R}(X)$ is the union of all elementary sets which are subsets of X, and the upper approximation $\overline{R}(X)$ is the union of all elementary sets which have a non-empty intersection with X. It is equivalent to say that an element of Ω necessarily belongs to X if all of its equivalent elements belong to X, whereas it possibly belongs to X if at least one of its equivalent elements belongs to X. The pair $\left(\underline{R}(X), \overline{R}(X)\right)$ is the representation of X in the approximation space (U, R), or simply called the Pawlak rough set of X with respect to (U, R).

Let P be a probability measure on $\left(\Omega/R, 2^{\Omega/R}\right)$. The corresponding inner and outer measures are defined by

$$\underline{P}(X) = P\left(\underline{R}(X)\right), \quad \overline{P}(X) = P\left(\overline{R}(X)\right), \quad \forall X \subseteq \Omega. \tag{2.21}$$

We can see that the theory of belief functions can be regarded as a particular case. In fact, $\underline{P}$ can be regarded as the credibility function and $\overline{P}$ is the dual plausibility function. The focal sets are the equivalence classes of R, and we have

$$m(F) = P(F), \qquad \forall F \in \Omega/R. \tag{2.22}$$

2.4 Clustering algorithms

Clustering, or unsupervised learning, has been widely studied in the data mining and machine learning literature due to its numerous applications to information retrieval, climate, segmentation, and business (Grossman and Frieder, 2012; Clifton and Lundquist, 2012; Dubey et al., 2013; Chen et al., 2012). The basic problem of clustering may be stated as follows (Aggarwal and Reddy, 2013): Given a set of data points, partition them into a set of groups which are as similar as possible.

Clustering algorithms can be broadly divided 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) (Jain, 2010). The most well-known hierarchical algorithms are single-link and complete-link (Murtagh and Contreras, 2012). The recent development of hierarchical clustering could be found in the work of Horng et al. (2005); Dong et al. (2006); Zhang and Xu (2015); Maalel et al. (2014), et.al.

Compared to hierarchical clustering algorithms, partitional clustering algorithms find all the clusters simultaneously as a partition of the data and do not impose a hierarchical structure. The most popular and the simplest partitional algorithm is c-means (Jain, 2010). The basic c-means algorithm has been extended in many different ways. In original c-means, each data point is assigned to a single cluster (called hard or crisp partition). To deal with the uncertainty or ambiguity of the membership of objects, Dunn (1973) proposed fuzzy c-means and later Bezdek (1981) improved it. Unlike c-means, in fuzzy c-means each data point can be a member of multiple clusters with a membership value (fuzzy partition). In the past years many other soft computing methodologies like possibility theory (Krishnapuram and Keller, 1993), rough sets (Lingras and West, 2004), and shadowed sets (Mitra et al., 2010) have been adopted to produce soft assignments in clustering.

Since partitional algorithms are preferred in pattern recognition due to the nature of available data, our coverage here focuses on these algorithms, especially the c-means type clustering. The following will provide a brief review of some soft prototype-based clustering methods based on uncertainty theories.

Let $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n\}$ be a collection of vectors in $\mathbb{R}^p$ describing n objects, and $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$ be the frame of discernment of classes. The objective function of c-means clustering is that:

$$J_{\text{CM}} = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij} d_{ij}^2, \tag{2.23}$$

where c is the number of clusters. As CM is based on crisp partitions, u_{ij} is either 0 or 1 depending whether $\boldsymbol{x}_i$ is in cluster ω_j. The value d_{ij} [1] is the distance between object $\boldsymbol{x}_i$ and class ω_j, *i.e.* the distance between $\boldsymbol{x}_i$ and class ω_j' s prototype vector $\boldsymbol{v}_j$. The algorithm proceeds by alternating between two steps

(1) Assignment update: Assign each observation to the cluster whose mean yields the least within-cluster sum of squares. Mathematically, we have

$$u_{ik} = 1, \quad \text{and } u_{ij} = 0, \ j \neq k, \tag{2.24}$$

[1] To make the notations consistent with those in the original paper, in this chapter, d_{ij} is used to define the distance between object $\boldsymbol{x}_i$ and class ω_j (or focal set A_j in ECM in Section 2.4.4).

with

$$k = \arg\min \{d_{il} \mid l = 1, 2, \cdots, c\}. \tag{2.25}$$

(2) Prototype update: Calculate the new means to be the centroids of the observations in the new clusters.

$$\boldsymbol{v}_k = \frac{\sum_{i=1}^{n} u_{ik}^{\beta} \boldsymbol{x}_k}{\sum_{i=1}^{n} u_{ik}^{\beta}}. \tag{2.26}$$

Note here that u_{ij} in the equation is either 0 or 1.

2.4.1 Fuzzy clustering

Fuzzy C-Means (FCM) is a fuzzification of the classical crisp C-Means (CM) algorithm (Dunn, 1973; Bezdek, 1981), and it looks for a best partition of the n patterns in $\boldsymbol{X}$ by miming the objective function

$$J_{\text{FCM}} = \sum_{k=1}^{c} u_{ik}^{\beta} d_{ik}^2, \tag{2.27}$$

subjects to the constraints

$$\sum_{k=1}^{c} u_{ik} = 1, \forall i \in \{1, 2, \cdots, n\} \tag{2.28}$$

and

$$\sum_{i=1}^{n} u_{ik} > 0, \forall k \in \{1, 2, \cdots, c\}. \tag{2.29}$$

In Eq. (2.27), $\beta \in [1, \infty)$ is the fuzzifier, $u_{ik} \in [0, 1]$ is the membership of the i^{th} pattern, and d_{ik} is the distance between object $\boldsymbol{x}_i$ and class ω_k, *i.e.* the distance between $\boldsymbol{x}_i$ and class ω_k's center (prototype) $\boldsymbol{v}_k$. The objective function could be minimized using an iterative algorithm obtained by the use of Lagrange multiplier method. The object assignment and prototype selection are preformed by the following alternating update steps:

(1) Assignment update:

$$u_{ik} = \frac{d_{ik}^{-2/(\beta-1)}}{\sum_{j=1}^{c} d_{ij}^{-2/(\beta-1)}}. \tag{2.30}$$

(2) Prototype update: the new prototype of cluster k is set to be $\boldsymbol{v}_k$ with

$$\boldsymbol{v}_k = \frac{\sum_{i=1}^{n} u_{ik}^{\beta} \boldsymbol{x}_k}{\sum_{i=1}^{n} u_{ik}^{\beta}}. \tag{2.31}$$

2.4.2 Possibilistic clustering

FCM uses the probabilistic constraint (Eq. (2.28)) that the memberships of an object across classes must sum to 1, which is responsible for the inability to detect noisy data and outliers (Krishnapuram and Keller, 1993; Gabrys and Bargiela, 2000). In fact, the fuzzy membership of $\boldsymbol{x}_i$ for cluster ω_k is based on the relative distance between object $\boldsymbol{x}_i$ and prototype $\boldsymbol{v}_k$, depending on not only d_{ik} but also the distances to other classes. In the Possibilistic version of C-Means (PCM) type clustering introduced by Krishnapuram and Keller (1993), the probabilistic constraint is dropped and the objective function of PCM is

$$J_{\text{PCM}} = \sum_{k=1}^{c} u_{ik}^{\beta} d_{ik}^2 + \sum_{k=1}^{c} \eta_k \sum_{i=1}^{n} (1 - u_{ik})^{\beta}, \tag{2.32}$$

where η_k is user-specified positive weight balancing the opposite effects of the two terms in J_{PCM}. As it can be seen, a penalty term is added in PCM to avoid the trivial solution $u_{ij} = 1, \forall i, j \in \{1, 2, \cdots, n\}$. The alternative update procedure of PCM is similar to FCM. The update equation for membership values is

$$u_{ik} = \frac{1}{1 + (d_{ik}^2/\eta_i)^{1/(\beta-1)}}. \tag{2.33}$$

As the added term is independent of class prototypes, the update formula for class centers is the same as that in FCM.

2.4.3 Rough clustering

Lingras and West (2004) proposed an extension of CM algorithm using the theory of rough set (RCM) and successfully applied it to web mining. In RCM, each cluster is regarded as an interval or rough set. A rough set W is characterized by its lower and upper approximations $\underline{W}$ and $\overline{W}$ respectively, with the following properties.

(1) Property 1. An object $\boldsymbol{x}_i$ can be part of as most one lower approximation.

(2) Property 2. If $\boldsymbol{x}_i \in \underline{W}$ of cluster W, then simultaneously $\boldsymbol{x}_i \in \overline{W}$.

(3) Property 3. If $\boldsymbol{x}_i$ is not a part of any lower approximation, then it belongs to two or more upper approximations.

This permits overlaps between clusters. The patterns in the set $\overline{W} - \underline{W} \triangleq W^B$ ($\in \overline{W}$ but $\notin \underline{W}$) locate in the boundary of class W (we call this meta approximation), and those belonging to the shared boundary ares of more than one cluster are overlapping objects. The Lingras and West (2004) rough cluster algorithm goes as follows:

Step 1: Initialization. Randomly partition each object to exactly one lower approximation. By Property 2, the pattern also belongs to the upper approximation of the same cluster.

Step 2: Update of prototypes. Computation of the prototypes of classes is modified in the rough framework by incorporating the concepts of upper and lower approximations. The means of the new class could be calculated by

$$\boldsymbol{v}_k = \begin{cases} w_l \dfrac{\sum_{\boldsymbol{x}_i \in \underline{\omega_k}} \boldsymbol{x}_i}{|\underline{\omega_k}|} + w_b \dfrac{\sum_{\boldsymbol{x}_i \in \omega_k^B} \boldsymbol{x}_i}{|\omega_k^B|} & \text{for} \quad \omega_k^B \neq \emptyset, \\ w_l \dfrac{\sum_{\boldsymbol{x}_i \in \underline{\omega_k}} \boldsymbol{x}_i}{|\underline{\omega_k}|} & \text{otherwise,} \end{cases} \tag{2.34}$$

where the parameters w_l and w_b define the importance of the lower approximation and boundary area of the class, and $|\cdot|$ is the cardinality of the set indicating the number of included elements.

Step 3: Update the partition. For a given object $\boldsymbol{x}_i$, determine its closest prototype $\boldsymbol{v}_h$, *i.e.*

$$d_{ih} = d(\boldsymbol{x}_i, \boldsymbol{v}_h) = \min_{k=1,2,\cdots,c} d(\boldsymbol{x}_i, \boldsymbol{v}_k). \tag{2.35}$$

Object $\boldsymbol{x}_i$ can be assigned to the upper approximation of class ω_h, *i.e.* $\boldsymbol{x}_i \in \overline{\omega_h}$.

The classes with prototypes that are also close to $\boldsymbol{x}_i$ should be considered then. Specially, given a threshold ϵ, those classes with centers that are not farther away from $\boldsymbol{x}_i$ than $d(\boldsymbol{x}_i, \boldsymbol{v}_h) + \epsilon$ are in the set

$$T = \{t : d(\boldsymbol{x}_i, \boldsymbol{v}_k) - d(\boldsymbol{x}_i, \boldsymbol{v}_h) \leq \epsilon \wedge h \neq k\}. \tag{2.36}$$

If $T = \emptyset$, pattern $\boldsymbol{x}_i$ is partitioned into $\underline{\omega_h}$. Otherwise, $\boldsymbol{x}_i$ is also close to at least one other mean $\boldsymbol{v}_t$ besides $\boldsymbol{v}_h$, and $\boldsymbol{x}_i$ could be assigned to the upper approximation of ω_t, *i.e.* $\boldsymbol{x}_i \in \overline{\omega_t}, \forall t \in T$.

Step 4: Iterate Step 2 and Step 3 until some given criterion functions converge, such as there are no more new assignments of objects.

It is observed that RCM clustering described above depends on the choice of the three parameters w_l, w_b, and ϵ. Mitra (2004) proposed an evolutionary rough c-means clustering algorithm, where genetic algorithms are employed to tune the threshold, and relative importance of upper and lower approximations of the rough sets modeling the clusters. Peters (2006) discussed various refinements of Lingras and West (2004) original proposal. These included calculation of rough centroids and the use of ratios

of distances as opposed to differences between distances similar to those used in the rough set based Kohonen algorithm described in (Lingras and West, 2004).

Peters (2006) first pointed out that the algorithm as presented by Lingras and West (2004) is numerical instable since there are data constellations where $|\underline{\omega_k}| = 0$, and suggested a solution by considering the three types of generated clusters by RCM, such as those having objects (Mitra et al., 2006):

(1) in both the lower and meta approximations;

(2) only in lower approximation;

(3) only in meta approximation.

We can modify Eq. (2.34) accordingly as follows

$$\boldsymbol{v}_k = \begin{cases} w_l \dfrac{\sum_{\boldsymbol{x}_i \in \underline{\omega_k}} \boldsymbol{x}_i}{|\underline{\omega_k}|} + w_b \dfrac{\sum_{\boldsymbol{x}_i \in \omega_k^B} \boldsymbol{x}_i}{|\omega_k^B|} & \text{for} \quad \omega_k^B \neq \emptyset \wedge \underline{\omega_k} \neq \emptyset, \\ \dfrac{\sum_{\boldsymbol{x}_i \in \underline{\omega_k}} \boldsymbol{x}_i}{|\underline{\omega_k}|} & \text{for} \quad \omega_k^B = \emptyset \wedge \underline{\omega_k} \neq \emptyset, \\ \dfrac{\sum_{\boldsymbol{x}_i \in \omega_k^B} \boldsymbol{x}_i}{|\omega_k^B|} & \text{for} \quad \omega_k^B \neq \emptyset \wedge \underline{\omega_k} = \emptyset. \end{cases} \tag{2.37}$$

The properties of the set T were also investigated and interpreted in (Peters, 2006). In Lingras and West (2004) RCM, the absolute distance $d(\boldsymbol{x}_i, \boldsymbol{v}_k) - d(\boldsymbol{x}_i, \boldsymbol{v}_h)$ is considered to determine the elements of T. But in RCM proposed by Peters (2006), the following relative distance measure instead of the absolute one was suggested

$$T' = \{t : \frac{d(\boldsymbol{x}_i, \boldsymbol{v}_k)}{d(\boldsymbol{x}_i, \boldsymbol{v}_h)} \leq \epsilon \wedge h \neq k\}. \tag{2.38}$$

2.4.4 Evidential clustering

Evidential c-means (Masson and Denoeux, 2008) is a direct generalization of FCM in the framework of belief functions, and it is based on the credal partition first proposed by Denœux and Masson (2004). The credal partition takes advantage of imprecise (meta) classes to express partial knowledge of class memberships. The principle is different from another belief clustering method put forward by Schubert (2004), in which conflict between evidence is utilized to cluster the belief functions related to multiple events. In ECM, the evidential membership of object $\boldsymbol{x}_i = \{x_{i1}, x_{i2}, \cdots, x_{ip}\}$ is represented by a bba $\boldsymbol{m}_i = (m_i(A_j) : A_j \subseteq \Omega)$ $(i = 1, 2, \cdots, n)$ over the given frame of discernment $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$. The set $\mathcal{F} = \{A_j \mid A_j \subseteq \Omega, m_i(A_j) > 0\}$ contains all the focal elements. The optimal credal partition $\boldsymbol{M} = (\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_n)$ and the matrix $\boldsymbol{V}$ of size $(c \times p)$ of cluster centres can be obtained by minimizing the following objective

function:

$$J_{\mathrm{ECM}}(\boldsymbol{M},\boldsymbol{V})=\sum_{i=1}^{n}\sum_{A_j\subseteq\Omega,A_j\neq\emptyset}|A_j|^{\alpha}m_i(A_j)^{\beta}d_{ij}^2+\sum_{i=1}^{n}\delta^2 m_i(\emptyset)^{\beta} \tag{2.39}$$

constrained on

$$\sum_{A_j\subseteq\Omega,A_j\neq\emptyset} m_i(A_j)+m_i(\emptyset)=1, \tag{2.40}$$

and

$$m_i\left(A_j\right)\geq 0,\quad m_i\left(\emptyset\right)\geq 0, \tag{2.41}$$

where $m_i(A_j)\triangleq m_{ij}$ is the bba of x_i given to the nonempty set A_j, while $m_i(\emptyset)\triangleq m_{i\emptyset}$ is the bba of $\boldsymbol{x}_i$ assigned to the empty set. Parameter α is a tuning parameter allowing to control the degree of penalization for subsets with high cardinality, parameter β is a weighting exponent and δ is an adjustable threshold for detecting the outliers. Here d_{ij} denotes the distance (generally Euclidean distance) between x_i and the barycenter (*i.e.* prototype, denoted by $\overline{v}_j$) associated with A_j:

$$d_{ij}^2=\|\boldsymbol{x}_i-\overline{\boldsymbol{v}}_j\|^2, \tag{2.42}$$

where $\overline{\boldsymbol{v}}_j$ is defined mathematically by

$$\overline{\boldsymbol{v}}_j=\frac{1}{|A_j|}\sum_{h=1}^{c}s_{hj}v_h,\quad\text{with}\quad s_{hj}=\begin{cases}1 & \text{if } \omega_h\in A_j,\\ 0 & \text{else.}\end{cases} \tag{2.43}$$

The notation $\boldsymbol{v}_h$ is the geometrical center of points in cluster ω_h. In fact the value of d_{ij} reflects the distance between object x_i and class A_j. Note that a "noise" class $\emptyset$ is considered in ECM. If $A_j=\emptyset$, it is assumed that the distance between object x_i and class A_j is $d_{ij}=\delta$. As we can see for credal partitions, the label of class ω_j is not from 1 to c as usual, but ranges in $1,2,\cdots,f$ where f is the number of the focal elements *i.e.* $f=|\mathcal{F}|$. The update process with Euclidean distance is given by the following two alternating steps.

(1) Assignment update:

$$m_{ij}=\frac{|A_j|^{-\alpha/(\beta-1)}d_{ij}^{-2/(\beta-1)}}{\sum\limits_{A_h\neq\emptyset}|A_h|^{-\alpha/(\beta-1)}d_{ih}^{-2/(\beta-1)}+\delta^{-2/(\beta-1)}},\forall i,\ \forall j/A_j(\neq\emptyset)\subseteq\Omega \tag{2.44}$$

$$m_{i\emptyset}=1-\sum_{A_j\neq\emptyset}m_{ij},\quad\forall i=1,2,\cdots,n. \tag{2.45}$$

(2) Prototype update: The prototypes (centers) of the classes are given by the rows of the matrix $\boldsymbol{V}_{c\times p}$, which is the solution of the following linear system:

$$\boldsymbol{HV} = \boldsymbol{B}, \tag{2.46}$$

where $\boldsymbol{H}$ is a matrix of size $(c \times c)$ given by

$$\boldsymbol{H}_{lt} = \sum_{i} \sum_{A_h \supseteq \{\omega_t, \omega_l\}} |A_h|^{\alpha-2} m_{ih}^{\beta}, \quad t, l = 1, 2, \cdots, c, \tag{2.47}$$

and $\boldsymbol{B}$ is a matrix of size $(c \times p)$ defined by

$$\boldsymbol{B}_{lq} = \sum_{i=1}^{n} x_{iq} \sum_{A_k \ni \omega_l} |A_k|^{\alpha-1} m_{ik}^{\beta}, \quad l = 1, 2, \cdots, c, \quad q = 1, 2, \cdots, p. \tag{2.48}$$

There is another credal clustering algorithm named EVCLUS (Denœux and Masson, 2004), which is dedicated to proximity data. It does not use any explicit geometrical model of the data, so that it is applicable to both metric and non-metric data. The only required data consists of a $n \times n$ dissimilarity matrix $\Delta = (\delta_{ij})$ where δ_{ij} represents the dissimilarity between objects x_i and x_j [2]. Matrix Δ is only supposed to be symmetric with null diagonal elements. EVCLUS minimizes an error function inspired from Sammon's stress function (Sammon, 1969) defined as

$$J_{\text{EVCLUS}} = \frac{1}{cons} \sum_{i<j} \frac{(ak_{ij} + b - \delta_{ij})^2}{\delta_{ij}}, \tag{2.49}$$

where a and b are two coefficients, $cons$ is a normalizing constant, and

$$k_{ij} = \sum_{A \cap B = \emptyset} m_i(A) m_j(B) \tag{2.50}$$

is the global conflict between the mass membership of objects x_i and x_j. This criterion can be minimized with respect to $\boldsymbol{M}$, a and b using an iterative procedure. To control the model complexity, the author suggested to add to the stress function a penalization term that favors "simple" and "informative" masses. The informativeness of each bba m_i could be measured through the following entropy measure:

$$\mathrm{E}(m_i) = \sum_{A \neq \emptyset} \log_2 \left(\frac{|A|}{m_i(A)} \right) + m_i(\emptyset) \log_2 \left(\frac{|\Omega|}{m_i(\emptyset)} \right). \tag{2.51}$$

[2] For the clustering methods on relational data sets, the objects may not be vectors. Thus we do not use the bold element in these contexts.

This measure tends to be small when the mass is assigned to few focal sets with small cardinality. Finally, the objective function to be minimized is:

$$J_{\text{EVCLUS1}} = J_{\text{EVCLUS}} + \lambda \sum_{i=1}^{n} \text{E}(m_i), \tag{2.52}$$

where λ is the penalization coefficient balancing the two items.

The semi-supervised versions of ECM and EVCLUS have been developed by Antoine et al. (2012) and Antoine et al. (2014) recently. The must-link and cannot-link constraints are considered in the clustering task, and these additional information could help us to improve the performance of the clustering algorithms. Other works about clustering using belief functions could be found in the work of Liu et al. (2012, 2015), where a modified version of ECM is designed; Masson and Denoeux (2011), where the ensemble clustering in the framework of belief functions are presented; Masson and Denœux (2009) where the relational version of ECM is proposed; Masson and Denœux (2004) where the problem of clustering objects based on interval-valued dissimilarities is tackled in the framework of the Dempster–Shafer theory of belief functions; Hariz et al. (2006) where a new clustering technique handling uncertainty in the attribute values of objects using belief functions is provided.

Recently, a new decision-directed clustering algorithm for relational data sets is put forward based on the Evidential K Nearest-Neighbor (EK-NN) rule (Denœux et al., 2015). Starting from an initial partition, the algorithm, called EK-NNclus, iteratively reassigns objects to clusters using the EK-NN rule (Denoeux, 1995), until a stable partition is obtained. After convergence, the cluster membership of each object is described by a mass function which assigns a belief to each specific cluster and to the whole set of clusters.

2.5 Community detection

In this section some background knowledge related to community detection problems and social networks, including centrality and similarity measures, modularity and some classical existing algorithms, will be presented.

2.5.1 Node centrality and similarity

Generally speaking, the person who is the center of a community in a social network has the following characteristics: he has relation with most of the members of the group and the relationships are stronger than usual; he may directly contact with other persons who also play an important role in their own communities. Therefore, the centers of the community should be set to the ones not only with high degree and weight strength, but also with neighbors who also have high degree and strength. The degree of node is

the number of its connections with other nodes, and the strength describes the levels of these connections. Gao et al. (2013) proposed an evidential centrality measure, named Evidential Semi-local Centrality (ESC), based on the theory of belief functions. In the application of ESC, the degree and strength of each node are first expressed by basic belief assignments (BBA), and then the fused importance is calculated using the combination rule in the theory of belief functions. The higher the ESC value is, the more important the node is. Gao et al. (2013) pointed out that it is more efficient than the existing centrality measures such as Degree Centrality (DC), Betweenness Centrality (BC) and Closeness Centrality (CC). The detail computation process of ESC can be found in (Gao et al., 2013).

The similarity measures the closeness between any pair of nodes in the graph. In (Zhou et al., 2009) several node similarity metrics on basis of local information were described and the performance of different measures applied to community detection was discussed. Here we give a brief description of some measures. Let $G(V, E)$ be an undirected network, where V is the set of n nodes and E is the sets of edges. Let $\boldsymbol{A} = (a_{ij})_{n\times n}$ denote the adjacency matrix, where $a_{ij} = 1$ represents that there is an edge between nodes n_i and n_j. The degree of node n_i, d_i, can be defined by

$$d_i = \sum_{j=1}^{n} a_{ij}. \tag{2.53}$$

(1) Common neighbors. This measure is based on the idea that more common neighbors the pair shares, more similar they are. Thus the similarity can be simply proportional to the number of their shared neighbors:

$$s^{\mathrm{C}}(x, y) = |N(x) \cap N(y)|, \tag{2.54}$$

where $N(x) = \{w \in V \setminus x : a(w, x) = 1\}$ denotes the set of vertices that are adjacent to x.

(2) Jaccard Index. This index was proposed by Jaccard over a hundred years ago, and is defined as

$$s^{\mathrm{J}}(x, y) = \frac{|N(x) \cap N(y)|}{|N(x) \cup N(y)|}. \tag{2.55}$$

(3) Zhou-Lü-Zhang Index. Zhou et al. (2009) also proposed a new similarity metric which is motivated by the resource allocation process:

$$s^{\mathrm{Z}}(x, y) = \sum_{z\in N(x)\cap N(y)} \frac{1}{d(z)}, \tag{2.56}$$

where $d(z)$ is the degree of node z.

(4) Pan's Index. Pan et al. (2010) pointed out that the similarity measure proposed by Zhou et al. (2009) may bring about inaccurate results for community detection on the networks as the metric can not differentiate the tightness relation between a pair of nodes whether they are connected directly or indirectly. In order to overcome this defect, in his presented new measure the similarity between unconnected pair is simply set to be 0:

$$S^P(x,y) = \begin{cases} \sum_{z \in N(x) \cap N(y)} \frac{1}{d(z)}, & \text{if } x, y \text{ are connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.57}$$

(5) Signal similarity. A similarity measure considering the global graph structure is put forward by Hu et al. (2008) based on signaling propagation in the network. For a network with n nodes, every node is viewed as an excitable system which can send, receive, and record signals. Initially, a node is selected as the source of signal. Then the source node sends a signal to its neighbors and itself first. Afterwards, the nodes with signals can also send signals to their neighbors and themselves. After a certain T time steps, the amount distribution of signals over the nodes could be viewed as the influence of the source node on the whole network. Naturally, compared with nodes in other communities, the nodes of the same community have more similar influence on the whole network. Therefore, similarities between nodes could be obtained by calculating the differences between the amount of signals they have received.

2.5.2 Modularity

Modularity is a recently introduced quality measure for graph clusterings (Newman and Girvan, 2004). By assumption, high values of modularity indicate good partitions. However, the true maximum is out of reach, as it has been recently proved that modularity optimization is an NP-complete problem (Brandes et al., 2006; Newman, 2006b). Existing modularity maximization algorithms are all try to find fairly good approximations of the modularity maximum in a reasonable time, for example the algorithms based on greedy techniques (Clauset et al., 2004; Newman, 2004; Blondel et al., 2008), spectral optimization (White and Smyth, 2005), simulated annealing (Guimera et al., 2004; Reichardt and Bornholdt, 2006) extremal optimization (Duch and Arenas, 2005) and other optimization strategies (Amiri et al., 2013; Xu et al., 2007).

Newman and Girvan (2004) proposed a modularity measure, also called Q function, which has been widely used. Let $G(V, E, W)$ be an undirected network, V is the set of n nodes, E is the set of edges, and $\boldsymbol{W}$ is a $N \times N$ edge weight matrix with elements $w_{ij}, i, j = 1, 2, \cdots, N$. Given a hard partition with K groups $\boldsymbol{U} = (u_{ik})_{N \times K}$, where u_{ik} is one if vertex i ($i = 1, 2, \cdots, N$) belongs to the k^{th} ($k = 1, 2, \cdots, K$) community,

0 otherwise. Denote the K crisp subsets of vertices by $\{C_1, C_2, \cdots, C_K\}$, then the modularity can be defined as (Fortunato, 2010):

$$Q_h = \frac{1}{\|\boldsymbol{W}\|} \sum_{k=1}^{K} \sum_{i,j \in C_k} \left(w_{ij} - \frac{k_i k_j}{\|\boldsymbol{W}\|} \right), \tag{2.58}$$

where $\|\boldsymbol{W}\| = \sum_{i,j=1}^{N} w_{ij}$, $k_i = \sum_{j=1}^{N} w_{ij}$. The node subsets $\{V_k, k = 1, 2, \cdots, c\}$ are determined by the hard partition $\boldsymbol{U}_{n \times c}$, but the role of $\boldsymbol{U}$ is somewhat obscured by this form of modularity function. To reveal the role played by the partition $\boldsymbol{U}$ explicitly, Havens et al. (2013) rewrote the equations in the form of $\boldsymbol{U}$. Let $\boldsymbol{k} = (k_1, k_2, \cdots, k_n)^T$, $\boldsymbol{B} = \boldsymbol{W} - \boldsymbol{k}^T \boldsymbol{k} / \|W\|$, then

$$\begin{aligned} Q_h &= \frac{1}{\|\boldsymbol{W}\|} \sum_{k=1}^{c} \sum_{i,j=1}^{n} \left(w_{ij} - \frac{k_i k_j}{\|\boldsymbol{W}\|} \right) u_{ik} u_{jk} \\ &= \frac{1}{\|\boldsymbol{W}\|} \sum_{k=1}^{c} \boldsymbol{u}_k \boldsymbol{B} \boldsymbol{u}_k^{\mathrm{T}} \\ &= \operatorname{trace}\left(\boldsymbol{U}^{\mathrm{T}} \boldsymbol{B} \boldsymbol{U} \right) / \|\boldsymbol{W}\|, \end{aligned} \tag{2.59}$$

where $\boldsymbol{u}_k = (u_{1k}, u_{2k}, \cdots, u_{nk})^{\mathrm{T}}$.

Havens et al. (2013) pointed out that an advantage of Eq. (2.59) is that it is well defined for any partition of the nodes not just crisp ones. The fuzzy modularity of $\boldsymbol{U}$ was derived as

$$Q_f = \operatorname{trace}\left(\boldsymbol{U}^{\mathrm{T}} \boldsymbol{B} \boldsymbol{U} \right) / \|\boldsymbol{W}\|, \tag{2.60}$$

where $\boldsymbol{U}$ is the membership matrix and u_{ik} represents the membership of community k for node n_i. If u_{ik} is restricted in $[0, 1]$, the fuzzy partition degrades to the hard one, and so Q_f equals to Q_h at this time.

The Q measure has been proved highly effective in practice for community evaluation, although Fortunato and Barthelemy (2007) claim resolution limits of modularity-based division methods. Besides, some other problems of Newman's modularity have also been found (Chen et al., 2009). To solve these problems, some new modularity measures have been proposed (Scripps et al., 2007; Chen et al., 2009). The Max–Min (MM) modularity function proposed by Chen et al. (2009) is utilized as the index to determine the optimal number of communities. MM modularity attempts to maximize the number of edges within groups and minimize the number of unrelated pairs from the user-defined unrelated pair set within groups at the same time:

$$Q_{\mathrm{MM}} = Q_{\max} - Q_{\min}, \tag{2.61}$$

where $Q_{\max}$ is the Q modularity of the original graph, while $Q_{\min}$ is that of the complement graph $G^{'}$. Graph $G^{'} = (Y, E^{'})$ is created based on the user-defined criteria $\mathcal{M}$ which defines whether two disconnected nodes i, j are related $(i, j) \in \mathcal{M}$ or unrelated $(i, j) \notin \mathcal{M}$, *i.e.* $(i, j) \in E^{'}$ if $(i, j) \notin E$ and $(i, j) \notin \mathcal{M}$. The related pairs $\mathcal{M}$ can be given by experts, or defined according to the original structure (Chen et al., 2009).

2.5.3 Some classical methods of community detection

Here we give a short presentation of some classical approaches for community detection.

MMO is a heuristic method through modularity optimization. This algorithm is now known as the "Louvain method" because it was devised when the authors all were at the University Catholique of Louvain. Here we use the name MMO to indicate that it is based on Multi-level Modularity Optimization. The algorithm is divided into two phases repeated iteratively. In the beginning of the first phase, the network is thought to have N groups each of which consists of only one node. Then for each node n_i, it may be placed into a new community (it must be a community that one of its neighbors belongs to) for which the gain of modularity is maximum. The first phase is not completed until no further improvement of the modularity can be achieved. The second phase consists in building a new network whose nodes are the communities detected in the last phase, and then the first phase can be reapplied on this newly created graph. Blondel et al. (2008) pointed out that MMO outperformed all other known community detection methods in terms of computation time.

Newman (2006a) demonstrated that the modularity can be succinctly expressed as a function of the eigenvalues and eigenvectors of the modularity matrix and derived a competitive Leading Eigenvector (**LE**) algorithm for identifying communities. The graph is first divided into two groups according to the signs of the elements of the eigenvector corresponding to the most positive eigenvalue of the modularity matrix, and then can be partitioned into more communities depending on the requirement analogously. It is showed that LE works better than the standard spectral partitioning method as it is unconstrained by the need to find groups of any particular size (Newman, 2006a) .

LPA is investigated by Raghavan et al. (2007) and it only uses the network structure and requires neither optimization of a predefined objective function nor prior information about the communities. It starts from an initial configuration where every node has a unique label. Then at every step one node (in asynchronous version) or each node (in a synchronous version) updates its current label to the label shared by the maximum number of its neighbors. For node v, its new label can be updated to ω_j with

$$j = \arg\max_{l}\{|u : c_u = l, u \in N_v|\}, \tag{2.62}$$

where $|X|$ is the cardinality of set X, and N_v is the set of node v's neighbors. When

there are multiple maximal labels among the neighbors labels, the new label is picked randomly from them. By this iterative process densely connected groups of nodes form consensus on one label to form communities, and each node has more neighbors in its own community than in any of other community. Communities are identified as a group of nodes sharing the same label.

InfoMap uses the probability flow of random walks on a network as a proxy for information flows in the real system, and graph clustering turns then into the coding problem of finding the partition that yields the minimum description length of an infinite random walk (Fortunato, 2010). The network is optimally decomposed into modules by compressing the information needed to describe of the process of information diffusion across the graph (Rosvall and Bergstrom, 2008). The regularities in the community structure and their relationships are reflected by a map.

The *c*-**rank** algorithm is proposed by Jiang et al. (2012), and it uses an alternate iteration strategy like *c*-means. Firstly, the top–*c* nodes with the highest rank centrality is selected as initial seeds. This initialization mechanism could overcome the problem brought by the random initial centers in the application of prototype-based clustering methods like *c*-means. Then the seeds and cluster labels are updated alternately by using an iterative technique. As illustrated before, the way of selecting c representative members with each to totally represent one individual community may be insufficient to fully characterize a community. This in turn indicates that multiple nodes should be utilized in order to capture each group in the network more accurately.

2.6 Semi-supervised learning

Unlike unsupervised clustering, the semi-supervised approach to clustering has a short history (Zhu and Goldberg, 2009). As the name suggests, semi-supervised learning is somewhere between unsupervised and supervised learning. The key point of semi-supervised clustering is how to combine both sources of information, *i.e.* the dissimilarities between data objects and the user-provide information. In fact, most semi-supervised learning strategies are based on extending either unsupervised or supervised learning to include additional information typical of the other learning paradigm (Zhu and Goldberg, 2009).

There are two kinds of prior information. One is the pairwise constrains, and the other is the labels of some individuals. This leads to two paradigms of semi-supervised learning methods: constrained clustering and semi-supervised classification. Constrained clustering is an extension to unsupervised clustering. The supervised information can be so-called must-link constraints, that two instances x_i, x_j must be in the same class; and cannot-link constraints, that x_i, x_j cannot be in the same cluster.

For semi-supervised classification, it is also known as classification with labeled and unlabeled data (or partially labeled data). This is an extension to the supervised

classification problem. One typically assumes that there is much more unlabeled data than labeled data. The supervised classifier does not work well since the limited amount of training data. Semi-supervised could effectively take advantage of the unlabeled data to help training the classifier and consequently improve the performance.

Recently, a few methods for semi-supervised clustering methods have been proposed for the community detection task in graph data sets. A semi-supervised spin-glass model that enables current community detection methods to incorporate background knowledge in the forms of pairwise constraints was developed in (Eaton and Mansbach, 2012). The proposed method has addressed the problem for applications that impose constraints on the acceptable solutions. Ma et al. (2010) developed a new community detection algorithm, named the so-called SNMF-SS, by combining the Symmetric Nonnegative Matrix Factorization (SNMF) and a semi-supervised clustering approach. The conducted experiments have shown that the proposed approach performs well particularly in the cases when community structure is obscure. Zhang (2013) put forward another framework which could implicitly encodes the must-link and cannot-link constraints by modifying the adjacency matrix of network. The author stated that the proposed method enhances the interpretability of the detection results. All these methods described above can improve the accuracy of community detection with prior knowledge in the form of pairwise constrains.

Liu et al. (2014) considered the individual labels as prior knowledge, *i.e.* the true community assignments of certain nodes are known in advance. In their work the traditional LPA is adapted, allowing a few nodes to have true community labels, but the rest nodes are unlabeled. In face the presented semi-supervised community detection approach is an application of the semi-supervised classification algorithm proposed by Wang and Zhang (2008) on graph data sets.

2.7 Summary

In this chapter, we provided some background knowledge on uncertainty theories, clustering and community detection. We started this chapter by introducing the theory of belief functions. Then we described the links between belief function theory and other uncertainty theories. Next, we introduced some classical clustering algorithms, especially some soft prototype-based clustering approaches. Finally, some concepts in networks and classical community detection models were presented.

This chapter recalls some basics of belief function theory. In the sequel of this report, uncertainty is modeled with the theory of belief functions. The following chapters are the core of this thesis. The key idea is to find imprecise clustering algorithms for relational data where only dissimilarities or similarities between objects are available. Firstly we try to use one "most" representative object to describe each class and proposed the median evidential c-means clustering (Chapter 3). To capture the cluster

structure more effectively, in Chapters 4 and 5, we extend the class representativeness mechanism so that each class could be described by more than one prototype. The clustering models for hard/fuzzy partitions (Chapter 4) and credal partitions (Chapter 5) are presented respectively. In Chapter 6 another semi-supervised community detection method that can take advantage of the available prior information is proposed.

3
Median evidential c-means clustering

3.1 Overview

Median clustering is of great value for partitioning relational data. In this chapter, a new prototype-based clustering method, called Median Evidential C-Means (MECM), which is an extension of median c-means and median fuzzy c-means on the theoretical framework of belief functions is proposed (Zhou et al., 2015c). The median variant relaxes the restriction of a metric space embedding for the objects but constrains the prototypes to be in the original data set. Due to these properties, MECM could be applied to graph clustering problems. A community detection scheme for networks based on MECM is investigated. The credal partitions of graphs provided by MECM, which are more refined than crisp and fuzzy ones, enable us to have a better understanding of the graph structures. An initial prototype-selection scheme based on evidential semi-centrality is presented to avoid local premature convergence. The concept of evidential modularity function is introduced to choose the optimal number of communities. Finally, experiments in synthetic and real data sets illustrate the performance of MECM and show its difference to other methods.

3.2 Median c-means and median fuzzy c-means

Median c-means is a variant of the traditional c-means method (Cottrell et al., 2006; Geweniger et al., 2010) on relational data sets. We assume that n data objects in $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\}$ are given. The dissimilarity between objects x_i and x_j is denoted by $d_{ij} = d(x_i, x_j)$. The objective function of MCM is similar to that in CM:

$$J_{\text{MCM}} = \sum_{j=1}^{c} \sum_{i=1}^{n} u_{ij} \overline{d}_{ij}^2, \tag{3.1}$$

where c is the number of clusters. As MCM is based on crisp partitions, u_{ij} is either 0 or 1 depending whether x_i is in cluster ω_j. The value $\overline{d}_{ij}$ is the dissimilarity between object x_i and cluster ω_j ($i = 1, 2, \cdots, n, j = 1, 2, \cdots, c$), *i.e.* the dissimilarity between

x_j and prototype v_j

$$\overline{d}_{ij} = d(x_i, v_j). \tag{3.2}$$

The dissimilarity measure is not assumed to be fulfilling any metric properties but should reflect the common sense of dissimilarity. Due to these weak assumptions, object x_i itself may be a general choice and it does not have to live in a metric space (Geweniger et al., 2010). The main difference between MCM and CM is that the prototypes of MCM are restricted to the data objects.

Median fuzzy c-means (MFCM) merges MCM and the standard fuzzy c-means (FCM). As in MCM, it requires the knowledge of the dissimilarity between data objects, and the prototypes are restricted to the objects themselves (Geweniger et al., 2010). MFCM also performs a two-step iteration scheme to minimize the cost function

$$J_{\text{MFCM}} = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij}^{\beta}\overline{d}_{ij}^{2}, \tag{3.3}$$

subject to the constrains

$$\sum_{k=1}^{c} u_{ik} = 1, \forall i \in \{1, 2, \cdots, n\}, \tag{3.4}$$

and

$$\sum_{i=1}^{n} u_{ik} > 0, \forall k \in \{1, 2, \cdots, c\}, \tag{3.5}$$

where each number $u_{ik} \in [0, 1]$ is interpreted as a degree of membership of object i to cluster ω_k, and $\beta > 1$ is a weighting exponent that controls the fuzziness of the partition. Again, MFCM is preformed by alternating update steps as for MCM:

- Assignment update:

$$u_{ij} = \frac{\overline{d}_{ij}^{-2/(\beta-1)}}{\sum_{k=1}^{c}\overline{d}_{ik}^{-2/(\beta-1)}}. \tag{3.6}$$

- Prototype update: the new prototype of cluster ω_j is set to be $v_j = x_{l^*}$ with

$$x_{l^*} = \arg\min_{\{v_j : v_j = x_l (\in X)\}} \sum_{i=1}^{n} u_{ij}^{\beta}\overline{d}_{ij}^{2}. \tag{3.7}$$

3.3 Median Evidential C-Means (MECM)

We introduce here median evidential c-means in order to take advantages of both median clustering and credal partitions. Like all the prototype-based clustering methods, for MECM, an objective function should first be found to provide an immediate mea-

sure of the quality of clustering results. Our goal then can be characterized as the optimization of the objective function to get the best credal partition.

3.3.1 The objective function of MECM

To group n objects in $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\}$ into c clusters $\omega_1, \omega_2, \cdots, \omega_c$, the credal partition $\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_n\}$ defined on $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$ is used to represent the class membership of objects, as in (Denœux and Masson, 2004; Masson and Denoeux, 2008). The quantities $m_{ij} = m_i(A_j)$ ($A_j \neq \emptyset, A_j \subseteq \Omega$) are determined by the dissimilarity between object x_i and focal set A_j which has to be defined first.

Let the prototype set of specific (singleton) clusters be $\boldsymbol{V} = \{v_1, v_2, \cdots, v_c\}$, where v_i is the prototype vector of cluster ω_i ($i = 1, 2, \cdots, c$) and it must be one of the n objects. If $|A_j| = 1$, *i.e.* A_j is associated with one of the singleton clusters in Ω (suppose to be ω_j with prototype vector v_j), then the dissimilarity between x_i and A_j is defined by

$$\overline{d}_{ij}^2 = d^2(x_i, v_j), \tag{3.8}$$

where $d(x_i, x_j)$ represents the given dissimilarity between objects x_i and x_j. When $|A_j| > 1$, it represents an imprecise (meta) cluster. If object x_i is to be partitioned into a meta cluster, two conditions should be satisfied. One is the dissimilarity values between x_i and the included singleton classes' prototypes are similar. The other is the object should be close to the prototypes of all these specific clusters. The former measures the degree of uncertainty, while the latter is to avoid the pitfall of partitioning two data objects irrelevant to any included specific clusters into the corresponding imprecise classes. The dissimilarity between x_i and A_j can be defined as:

$$\overline{d}_{ij}^2 = \frac{\gamma \frac{1}{|A_j|} \sum\limits_{\omega_k \in A_j} d^2(x_i, v_k) + \rho_{ij} \min\{d(x_i, v_k) : \omega_k \in A_j\}}{\gamma + 1}, \tag{3.9}$$

with

$$\rho_{ij} = \frac{\sum\limits_{\omega_x, \omega_y \in A_j} \sqrt{\left(d\left(x_i, v_x\right) - d(x_i, v_y)\right)^2}}{\eta \sum\limits_{\omega_x, \omega_y \in A_j} d(v_x, v_y)}. \tag{3.10}$$

In Eq. (3.9), parameter γ weights the contribution of the dissimilarity of the objects from the consisted specific clusters and it can be tuned according to the applications. If $\gamma = 0$, the imprecise clusters only consider our uncertainty. Discounting factor ρ_{ij} reflects the degree of uncertainty. If $\rho_{ij} = 0$, it means that all the dissimilarity values between x_i and the included specific classes in A_j are equal, and we are absolutely uncertain about which cluster object x_i is actually in. Parameter η ($\in [0, 1]$) can be tuned to control of the discounting degree. In credal partitions, we can distinguish between "equal evidence" (uncertainty) and "ignorance". The ignorance reflects the

indistinguishability among clusters. In fact, imprecise classes take both uncertainty and ignorance into consideration, and we can balance the two types of imprecise information by adjusting γ. Therefore, the dissimilarity between x_i and $A_j (A_j \neq \emptyset, A_j \subseteq \Omega)$, $\overline{d}_{ij}$, can be calculated by

$$\overline{d}_{ij}^2 = \begin{cases} d^2(x_i, v_j) & |A_j| = 1, \\ \gamma \frac{1}{|A_j|} \sum\limits_{\omega_k \in A_j} d^2(x_i, v_k) + \rho_{ij} \min\{d(x_i, v_k) : \omega_k \in A_j\} & \\ \hline \gamma + 1 & |A_j| > 1 \end{cases}. \tag{3.11}$$

Like ECM, we propose to look for the credal partition

$$\boldsymbol{M} = \{\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_n\} \in \mathcal{R}^{n \times 2^c}$$

and the prototype set $\boldsymbol{V} = \{v_1, v_2, \cdots, v_c\}$ of specific (singleton) clusters by minimizing the objective function:

$$J_{\text{MECM}}(\boldsymbol{M}, \boldsymbol{V}) = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \overline{d}_{ij}^2 + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{3.12}$$

constrained on

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1, \tag{3.13}$$

where $m_{ij} \triangleq m_i(A_j)$ is the bba of x_i given to the nonempty set A_j, $m_{i\emptyset} \triangleq m_i(\emptyset)$ is the bba of x_i assigned to the empty set, and $\overline{d}_{ij}$ is the dissimilarity between x_i and focal set A_j. Parameters α, β, δ are adjustable with the same meanings as those in ECM. Note that J_{MECM} depends on the credal partition $\boldsymbol{M}$ and the set $\boldsymbol{V}$ of all prototypes.

3.3.2 The optimization

To minimize J_{MECM}, an optimization scheme via an Expectation-Maximization (EM) algorithm as in MCM (Cottrell et al., 2006) and MFCM (Geweniger et al., 2010) can be designed. The detailed derivation process can be found in the appendix. The alternate update steps are as follows:

Step 1. Credal partition ($\boldsymbol{M}$) update.

- $\forall A_j \subseteq \Omega, A_j \neq \emptyset$,

$$m_{ij} = \frac{|A_j|^{-\alpha/(\beta-1)} \overline{d}_{ij}^{-2/(\beta-1)}}{\sum\limits_{A_k \neq \emptyset} |A_k|^{-\alpha/(\beta-1)} \overline{d}_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}} \tag{3.14}$$

- if $A_j = \emptyset$,

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij} \tag{3.15}$$

Step 2. Prototype ($\boldsymbol{V}$) update.

The prototype v_i of a specific (singleton) cluster ω_i ($i = 1, 2, \cdots, c$) can be updated first and then the dissimilarity between the object and the prototype of each imprecise (meta) clusters associated with subset $A_j \subseteq \Omega$ can be obtained by Eq. (3.11). For singleton clusters ω_k ($k = 1, 2, \cdots, c$), the corresponding new prototypes v_k ($k = 1, 2, \cdots, c$) are set to be sample x_l orderly, with

$$x_l = \arg\min_{v_k^{'}} \left\{ L(v_k^{'}) \triangleq \sum_{i=1}^{n} \sum_{\omega_k \in A_j} |A_j|^{\alpha} m_{ij}^{\beta} \overline{d}_{ij}^2(v_k^{'}), v_k^{'} \in \{x_1, x_2, \cdots, x_n\} \right\}, \tag{3.16}$$

The dissimilarity between x_i and A_j, $\overline{d}_{ij}^2$, is a function of $v_k^{'}$, which is the prototype of $\omega_k (\in A_j)$, and it should be one of the n objects in $\boldsymbol{X} = \{x_1, x_2, \cdots, x_n\}$.

The bbas of the objects' class membership are updated identically to ECM (Masson and Denoeux, 2008), but it is worth noting that $\overline{d}_{ij}$ has different meanings and less constraints as explained before. For the prototype updating process the fact that the prototypes are assumed to be one of the data objects is taken into account. Therefore, when the credal partition matrix $\boldsymbol{M}$ is fixed, the new prototypes of the clusters can be obtained in a simpler manner than in the case of ECM application. The MECM algorithm is summarized as Algorithm 1.

Algorithm 1 : Median evidential c-means algorithm

Input	dissimilarity matrix $\boldsymbol{D} \triangleq [d(x_i, x_j)]_{n \times n}$ for the n objects $\{x_1, x_2, \cdots, x_n\}$
Parameters	c: number clusters $1 < c < n$ α: weighing exponent for cardinality $\beta > 1$: weighting exponent $\delta > 0$: dissimilarity between any object to the emptyset $\gamma > 0$: weight of dissimilarity between data and prototype vectors $\eta \in [0, 1]$: control of the discounting degree
Initialization	Choose randomly c initial cluster prototypes from the objects
Loop	$t \leftarrow 0$ **Repeat** (1). $t \leftarrow t + 1$ (2). Compute $\boldsymbol{M}_t$ using Eq. (3.14), Eq. (3.15) and $\boldsymbol{V}_{t-1}$ (3). Compute the new prototype set $\boldsymbol{V}_t$ using Eq. (3.16) **Until** the prototypes remain unchanged

The convergence of MECM algorithm can be proved in the following lemma, similar

to the proof of median neural gas (Cottrell et al., 2006) and MFCM (Geweniger et al., 2010).

Lemma 3.1 *The MECM algorithm (Algorithm 1) converges in a finite number of steps.*

Proof *Suppose* $\theta^{(t)} = (\boldsymbol{M}_t, \boldsymbol{V}_t)$ *and* $\theta^{(t+1)} = (\boldsymbol{M}_{t+1}, \boldsymbol{V}_{t+1})$ *are the parameters from two successive iterations of MECM. We will first prove that*

$$J_{MECM}(\theta^{(t)}) \geq J_{MECM}(\theta^{(t+1)}), \tag{3.17}$$

which shows MECM always monotonically decreases the objective function. Let

$$\begin{aligned} J_{MECM}^{(t)} &= \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^{\alpha} (m_{ij}^{(t)})^{\beta} (\overline{d}_{ij}^{(t)})^2 + \sum_{i=1}^{n} \delta^2 (m_{i\emptyset}^{(t)})^{\beta} \\ &\triangleq \sum_i \sum_j f_1(\boldsymbol{M}_t) f_2(\boldsymbol{V}_t) + \sum_i f_3(\boldsymbol{M}_t), \end{aligned} \tag{3.18}$$

where $f_1(\boldsymbol{M}_t) = |A_j|^{\alpha}(m_{ij}^{(t)})^{\beta}$, $f_2(\boldsymbol{V}_t) = (d_{ij}^{(t)})^2$, *and* $f_3(\boldsymbol{M}_t) = \delta^2(m_{i\emptyset}^{(t)})^{\beta}$. $\boldsymbol{M}_{t+1}$ *is then obtained by maximizing the right hand side of the equation above. Thus,*

$$J_{MECM}^{(t)} \geq \sum_i \sum_j f_1(\boldsymbol{M}_{t+1}) f_2(\boldsymbol{V}_t) + \sum_i f_3(\boldsymbol{M}_{t+1}) \tag{3.19}$$

$$\geq \sum_i \sum_j f_1(\boldsymbol{M}_{t+1}) f_2(\boldsymbol{V}_{t+1}) + \sum_i f_3(\boldsymbol{M}_{t+1}) \tag{3.20}$$

$$= J_{MECM}^{(t+1)}. \tag{3.21}$$

This inequality (3.19) *comes from the fact* $\boldsymbol{M}_{t+1}$ *is determined by differentiating of the respective Lagrangian of the cost function with respect to* $\boldsymbol{M}_t$. *To get Eq.* (3.20), *we could use the fact that every prototype* v_k $(k = 1, 2, \cdots, c)$ *in* $\boldsymbol{V}_{t+1}$ *is orderly chosen explicitly to be*

$$\arg\min_{v_k'} \left\{ L(v_k') \triangleq \sum_{i=1}^{n} \sum_{\omega_k \in A_j} |A_j|^{\alpha} m_{ij}^{\beta} \overline{d}_{ij}^2(v_k'), v_k' \in \{x_1, x_2, \cdots, x_n\} \right\},$$

and thus this formula evaluated at $\boldsymbol{V}_{t+1}$ *must be equal to or less than the same formula evaluated at* V_t.

Hence MECM causes the objective function to converge monotonically. Moreover, the bba $\boldsymbol{M}$ *is a function of the prototypes* $\boldsymbol{V}$ *and for given* $\boldsymbol{V}$ *the assignment* $\boldsymbol{M}$ *is unique. Because MECM assumes that the prototypes are original object data in* $\boldsymbol{X}$*, so there is a finite number of different prototype vectors* $\boldsymbol{V}$ *and so is the number of corresponding credal partitions* $\boldsymbol{M}$*. Consequently we can get the conclusion that the MECM algorithm converges in a finite number of steps.* □

Remark 1. Although the objective function of MECM takes the same form as that in ECM (Masson and Denoeux, 2008), we should note that in MECM, it is no longer assumed that there is an underlying Euclidean distance. Thus the dissimilarity measure d_{ij} has few restrictions such as the triangle inequality or the symmetry. This freedom distinguishes the MECM from ECM and RECM, and it leads to the constraint for the prototypes to be data objects themselves. The distinct difference in the process of minimization between MECM and ECM lies in the prototype-update step. The purpose of updating the prototypes is to make sure that the cost function would decrease. In ECM the Lagrange multiplier optimization is evoked directly while in MECM a search method is applied. As a result, the objective function may decline more quickly in ECM as the optimization process has few constraints. However, when the centers of clusters in the data set are more likely to be the data object, MECM may converge with few steps.

Remark 2. Although both MECM and MFCM can be applied to the same type of data set, they are very different. This is due to the fact that they are founded on different models of partitioning. MFCM provides fuzzy partition. In contrast, MECM gives credal partitions. We emphasize that MECM is in line with MCM and MFCM: each class is represented by a prototype which is restricted to the data objects and the dissimilarities are not assumed to be fulfilling any metric properties. MECM is an extension of MCM and MFCM in the framework of belief functions.

3.3.3 The parameters of the algorithm

As in ECM, before running MECM, the values of the parameters have to be set. Parameters α, β and δ have the same meanings as those in ECM, and γ weighs the contribution of uncertainty to the dissimilarity between nodes and imprecise clusters. The value β can be set to be $\beta = 2$ in all experiments for which it is a usual choice. The parameter α aims to penalize the subsets with high cardinality and control the amount of points assigned to imprecise clusters in both ECM and MECM. As the measures for the dissimilarity between nodes and meta classes are different, thus different values of α should be taken even for the same data set. But both in ECM and MECM, the higher

α is, the less mass belief is assigned to the meta clusters and the less imprecise will be the resulting partition. However, the decrease of imprecision may result in high risk of errors. For instance, in the case of hard partitions, the clustering results are completely precise but there is much more intendancy to partition an object to an unrelated group. As suggested in (Masson and Denoeux, 2008), a value can be used as a starting default one but it can be modified according to what is expected from the user. The choice δ is more difficult and is strongly data dependent (Masson and Denoeux, 2008).

For determining the number of clusters, the validity index of a credal partition defined by Masson and Denoeux (2008) could be utilised:

$$N^*(c) \triangleq \frac{1}{n\log_2(c)} \times \sum_{i=1}^{n} \left[\sum_{A\in 2^\Omega \setminus \emptyset} m_i(A)\log_2|A| + m_i(\emptyset)\log_2(c) \right], \tag{3.22}$$

where $0 \leq N^*(c) \leq 1$. This index has to be minimized to get the optimal number of clusters. When MECM is applied to community detection, a different index is defined to determine the number of communities. We will describe it in the next section.

3.4 Application and evaluation issues

In this section, we will discuss how to apply MECM to community detection problems in social networks and how to evaluate credal partitions.

3.4.1 Evidential modular function

Assume the obtained credal partition of the graph is

$$\boldsymbol{M} = [\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_n]^{\mathrm{T}},$$

where $\boldsymbol{m}_i = (m_{i1}, m_{i2}, \cdots, m_{ic})^{\mathrm{T}}$. Similarly to the fuzzy modularity by Havens et al. (Havens et al., 2013), here we introduce an evidential modularity (Zhou et al., 2014):

$$Q_e = \frac{1}{\|\boldsymbol{W}\|} \sum_{k=1}^{c} \sum_{i,j=1}^{n} \left(w_{ij} - \frac{k_i k_j}{\|\boldsymbol{W}\|} \right) pl_{ik} pl_{jk}, \tag{3.23}$$

where $\boldsymbol{pl}_i = (pl_{i1}, pl_{i2}, \cdots, pl_{ic})^{\mathrm{T}}$ is the contour function associated with m_i, which describes the upper value of our belief to the proposition that the i^{th} node belongs to the k^{th} community.

Let $\boldsymbol{PL} = (pl_{ik})_{n\times c}$, then Eq. (3.23) can be rewritten as:

$$Q_e = \frac{\mathrm{trace}(\boldsymbol{PL}^{\mathrm{T}}\ \boldsymbol{B}\ \boldsymbol{PL})}{\|\boldsymbol{W}\|}. \tag{3.24}$$

Q_e is a directly extension of the crisp and fuzzy modularity functions in Eq. (2.59). When the credal partition degrades into the hard and fuzzy ones, Q_e is equal to Q_h and Q_f respectively.

3.4.2 The initial prototypes for communities

Generally speaking, the person who is the center in the community in a social network has the following characteristics: he has relation with all the members of the group and their relationship is stronger than usual; he may directly contact with other persons who also play an important role in their own community. For instance, in Twitter network, all the members in the community of the fans of Real Madrid football Club (RMC) are following the official account of the team, and RMC must be the center of this community. RMC follows the famous football player in the club, who is sure to be the center of the community of his fans. In fact, RMC has 19015495 followers and 48 followings (the data on April 22, 2016). Most of the followings have more than 500000 followers. Therefore, the centers of the community can be set to the ones not only with high degree and strength, but also with neighbors who also have high degree and strength. Thanks to the theory of belief functions, the evidential semi-local centrality ranks the nodes considering all these measures. Therefore the initial c prototypes of each community can be set to the nodes with largest ESC values.

Note that there is usually more than one center in one community. Take Twitter network for example again, the fans of RMC who follow the club official account may also pay attention to Cristiano Ronaldo, the most popular player in the team, who could be another center of the community of RMC's fans to a great extent. These two centers (the accounts of the club and Ronaldo) both have large ESC values but they are near to each other. This situation violates the rule which requires the chosen seeds as far away from each other as possible (Arthur and Vassilvitskii, 2007; Jiang et al., 2012).

The dissimilarity between the nodes could be utilised to solve this problem. Suppose the ranking order of the nodes with respect to their ESCs is $n_1 \geq n_2 \geq \cdots \geq n_n$. In the beginning n_1 is set to be the first prototype as it has the largest ESC, and then node n_2 is considered. If $d(n_1, n_2)$ (the dissimilarity between nodes 1 and 2) is larger than a threshold μ, it is chosen to be the second prototype. Otherwise, we abandon n_2 and turn to check n_3. The process continues until all the c prototypes are found. If there are not enough prototypes after checking all the nodes, we should decrease μ moderately and restart the search from n_1. In this chapter we test the approach with the dissimilarity measure proposed in (Zhou, 2003). Based on our experiments, $[0.7, 1]$ is a better experiential range of the threshold μ. This seed choosing strategy is similar to that in (Jiang et al., 2012).

3.4.3 The community detection algorithm based on MECM

The whole community detection algorithm in social networks based on MECM is summarized in Algorithm 2.

Algorithm 2 : Community detection algorithm based on MECM

Input: $\boldsymbol{A}$, the adjacency matrix; $\boldsymbol{W}$, the weight matrix (if any); μ, the threshold controlling the dissimilarity between the prototypes; $c_{\min}$, the minimal number of communities; $c_{\max}$, the maximal of communities; the required parameters in original MECM algorithm

Initialization: Calculate the dissimilarity matrix of the nodes in the graph.

repeat

(1). Set the cluster number c in MECM be $c = c_{\min}$.

(2). Choose the initial c prototypes using the strategy proposed in Section 3.4.2.

(3). Run MECM with the corresponding parameters and the initial prototypes got in (2).

(4). Calculate the evidential modularity using Eq. (3.24).

(5). Let $c = c + 1$.

until c reaches at $c_{\max}$.

Output: Choose the number of communities at around which the modular function peaks, and output the corresponding credal partition of the graph.

In the algorithm, $c_{\min}$ and $c_{\max}$ can be determined based on the original graph. Note that $c_{\min} \geq 2$. It is an empirical range of the community number of the network. If c is given, we can get a credal partition based on MECM and then the evidential modularity can be derived. As we can see, the modularity is a function of c and it should peak at around the optimal value of c for the given network.

3.4.4 Performance evaluation

The objective of the clustering problem is to partition a similar data pair to the same group. There are two types of correct decisions by the clustering result: a true positive (TP) decision assigns two similar objects to the same cluster, while a true negative (TN) decision assigns two dissimilar objects to different clusters. Correspondingly, there are two types of errors we can commit: a false positive (FP) decision assigns two dissimilar objects to the same cluster, while a false negative (FN) decision assigns two similar objects to different clusters. Let a (respectively, b) be the number of pairs of objects simultaneously assigned to identical classes (respectively, different classes) by the stand reference partition and the obtained one. Actually a (respectively, b) is the number of TP (respectively, TN) decisions. Similarly, let c and d be the numbers of FP and FN decisions respectively. Two popular measures that are typically used to evaluate the performance of hard clusterings are precision and recall. Precision (P) is the fraction of relevant instances (pairs in identical groups in the clustering benchmark) out of those

retrieved instances (pairs in identical groups of the discovered clusters), while recall (R) is the fraction of relevant instances that are retrieved. Then precision and recall can be calculated by

$$\mathrm{P} = \frac{a}{a+c} \quad \text{and} \quad \mathrm{R} = \frac{a}{a+d} \tag{3.25}$$

respectively. The Rand index (RI) measures the percentage of correct decisions and it can be defined as

$$\mathrm{RI} = \frac{2(a+b)}{n(n-1)}, \tag{3.26}$$

where n is the number of data objects. In fact, precision measures the rate of the first type of errors (FP), recall (R) measures another type (FN), while RI measures both.

For fuzzy and evidential clusterings, objects may be partitioned into multiple clusters with different degrees. In such cases precision would be consequently low (Mendes and Sacks, 2003). Usually the fuzzy and evidential clusters are made crisp before calculating the measures, using for instance the maximum membership criterion (Mendes and Sacks, 2003) and pignistic probabilities (Masson and Denoeux, 2008). Thus in the work presented in this chapter, we have hardened the fuzzy and credal clusters by maximizing the corresponding membership and pignistic probabilities and calculate precision, recall and RI for each case.

The introduced imprecise clusters can avoid the risk to group a data into a specific class without strong belief. In other words, a data pair can be clustered into the same specific group only when we are quite confident and thus the misclassification rate will be reduced. However, partitioning too many data into imprecise clusters may cause that many objects are not identified for their precise groups. In order to show the effectiveness of the proposed method in these aspects, we use the evidential precision (EP) and evidential recall (ER):

$$\mathrm{EP} = \frac{n_{er}}{N_e}, \quad \mathrm{ER} = \frac{n_{er}}{N_r}. \tag{3.27}$$

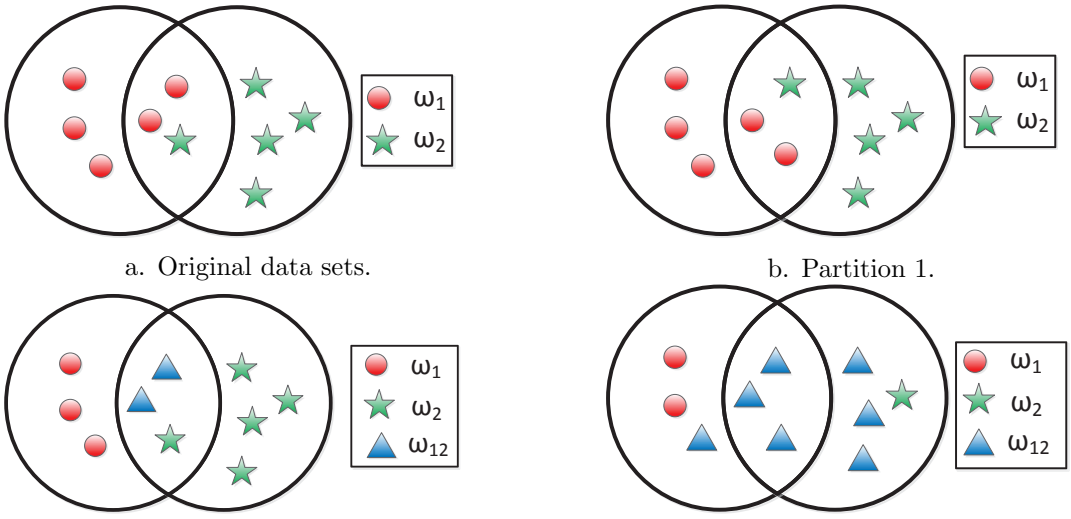
In Eq. (3.27), the notation N_e denotes the number of pairs partitioned into the same specific group by evidential clusterings, and n_{er} is the number of relevant instance pairs out of these specifically clustered pairs. The value N_r denotes the number of pairs in the same group of the clustering benchmark, and ER is the fraction of specifically retrieved instances (grouped into an identical specific cluster) out of these relevant pairs. When the partition degrades to a crisp one, EP and ER equal to the classical precision and recall measures respectively. EP and ER reflect the accuracy of the credal partition from different points of view, but we could not evaluate the clusterings from one single term. For example, if all the objects are partitioned into imprecise clusters except two relevant data object grouped into a specific class, EP = 1 in this case. But we could not say this is a good partition since it does not provide us with any information of great value. In this case ER ≈ 0. Thus ER could be used to express the efficiency of

the method for providing valuable partitions. Certainly we can combine EP and ER like RI to get the evidential rank index (ERI) describing the accuracy:

$$\mathrm{ERI} = \frac{2(a^* + b^*)}{n(n-1)}, \tag{3.28}$$

where a^* (respectively, b^*) is the number of pairs of objects simultaneously clustered to the same specific class (*i.e.* singleton class, respectively, different classes) by the stand reference partition and the obtained credal one. Note that for evidential clusterings, precision, recall and RI measures are calculated after the corresponding hard partitions are got, while EP, ER and ERI are based on hard credal partitions (Masson and Denoeux, 2008).

Example 3.1. In order to show the significance of the above performance measures, an example containing only ten objects from two groups is presented here. The three partitions are given in Figures 3.1-b – 3.1-d. The values of the six evidential indices (P,R,RI,EP,ER,ERI) are listed in Table 3.1.



a. Original data sets. b. Partition 1.

c. Partition 2. d. Partition 3.

Figure 3.1: A small data set with imprecise classes.

We can see that if we simply partition the nodes in the overlapped area, the risk of misclassification is high in terms of precision. The introduced imprecise cluster $\omega_{12} \triangleq \{\omega_1, \omega_2\}$ could enable us to make soft decisions, as a result the accuracy of the specific partitions is high. However, if too many objects are clustered into imprecise classes, which is the case of partition 3, it is pointless although EP is high. Generally, EP denotes the accuracy of the specific decisions, while ER represents the efficiency

Table 3.1: Evaluation Indices of the obtained partitions.

	EP	ER	ERI	P	R	RI
Partition 1	0.6190	0.6190	0.6444	0.6190	0.6190	0.6444
Partition 2	1.0000	0.6190	0.8222	–	–	–
Partition 3	1.0000	0.0476	0.5556	–	–	–

of the approach. We remark that the evidential indices degrade to the corresponding classical indices (*e.g,* evidential precision degrades to precision) when the partition is crisp.

3.5 Experiments

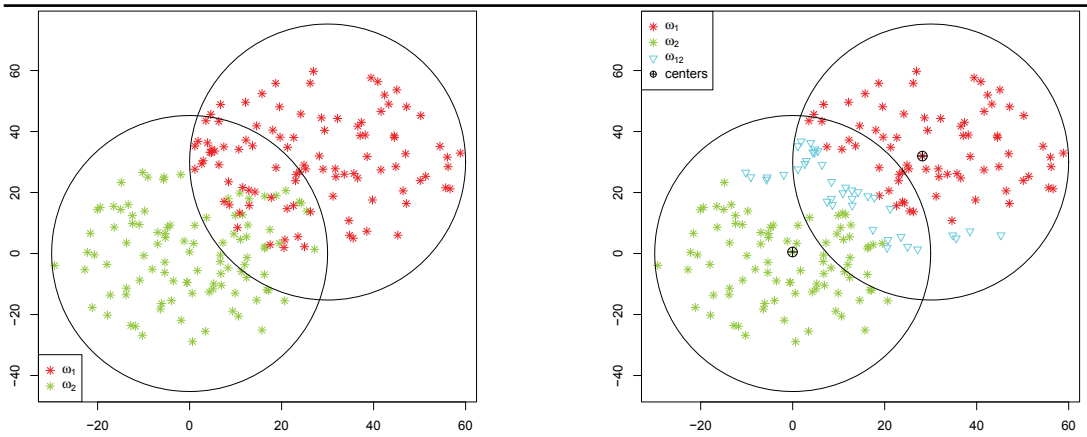
In this section a number of experiments are performed on classical data sets in the distance space and on graph data for which only the dissimilarities between nodes are known. The obtained credal partitions are compared with hard and fuzzy ones using the evaluation indices proposed in Section 3.4.4 to show the merits of MECM.

3.5.1 Overlapped data set

Clustering approaches to detect overlap objects which leads to recent attentions are still inefficiently processed. Due to the introduction of imprecise classes, MECM has the advantage to detect overlapped clusters. In the first example, we will use overlapped data sets to illustrate the behavior of the proposed algorithm.

We start by generating 2×100 points uniformly distributed in two overlapped circles with a same radius $R = 30$ but with different centers. The coordinates of the first circle's center are $(0, 0)$ while the coordinates of the other circle's center are $(30, 30)$. The data set is displayed in Figure 3.2-a.

In order to show the influence of parameters in MECM and ECM, different values of γ, α, η and δ have been tested for this data set. Figure 3.3-a displays the three evidential indices varying with γ (α is fixed to be 2) by MECM, while Figure 3.3-b depicts the results of MECM with different α but a fixed $\gamma = 0.4$ (η and δ are set 0.7 and 50, respectively, in the tests). For fixed α and γ, the results with different η and δ are shown in Figure 3.3-c. The effect of α and δ on the clusterings of ECM is illustrated in Figure 3.3-e. As we can see, for both MECM and ECM, if we want to make more imprecise decisions to improve ER, parameter α can be decreased. In MECM, we can also reduce the value of parameter γ to accomplish the same purpose. Although both α and γ have effect on imprecise clusters in MECM, the mechanisms



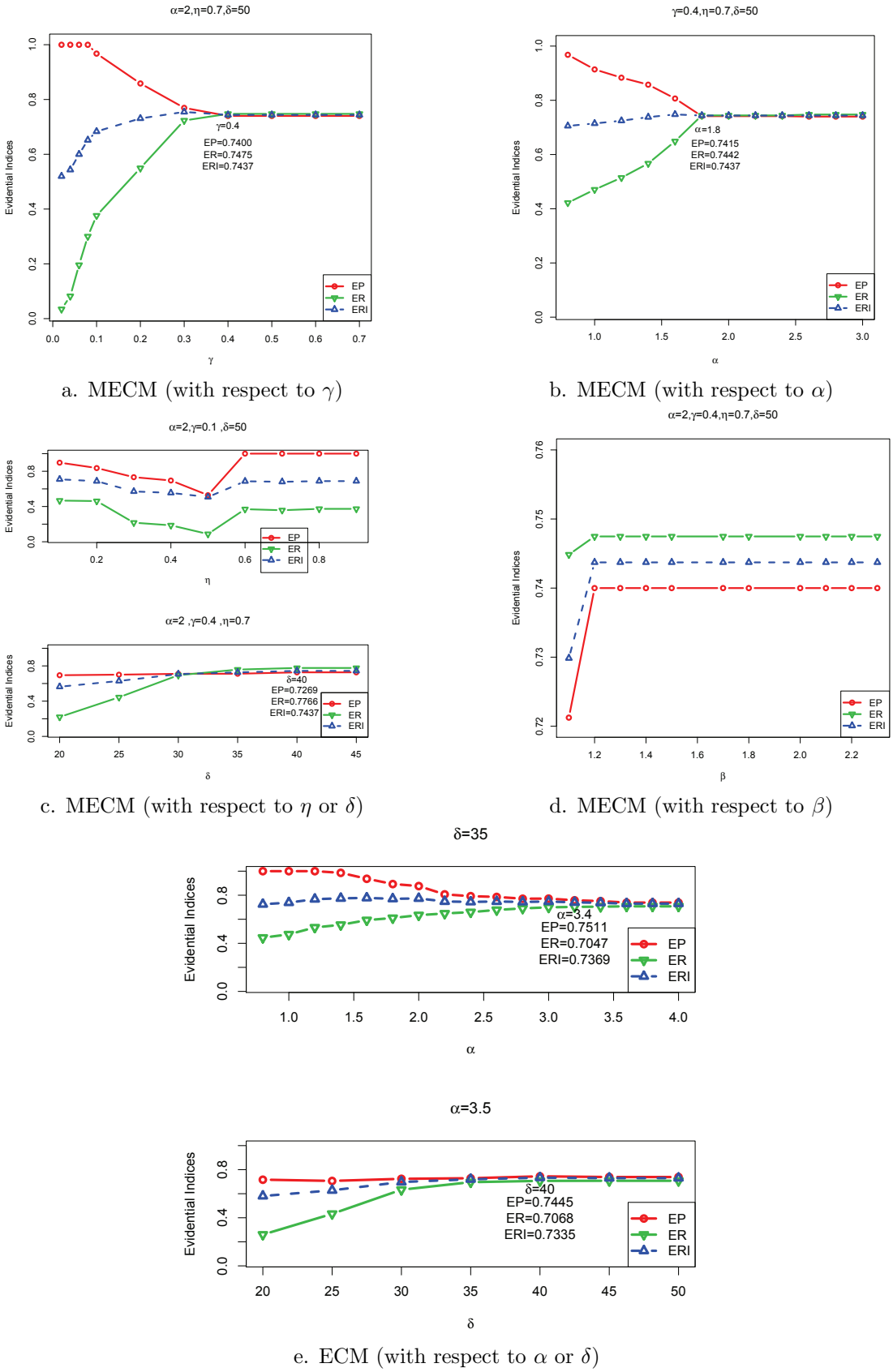
a. Overlapped data b. Clustering result of MECM

Figure 3.2: Clustering of overlapped data set

they work are different. Parameter α tries to adjust the penalty degree to control the imprecise rates of the results. However, for γ, the same aim could be got by regulating the uncertainty degree of imprecise classes. It can be seen from the figures, the effect of γ is more conspicuous than α. Moreover, although α may be set too high to obtain good clusterings, "good" partitions can also be got by adjusting γ in this case. For both MECM and ECM, the stable limiting values of evidential measures are around 0.7 and 0.8. Such values suggest the equivalence of the two methods to a certain extent. Parameter η is used for discounting the distance between uncertain objects and specific clusters. As pointed out in Figure 3.3-c, if γ and α are well set, it has little effect on the final clusterings. The same is true in the case of δ which is applied to detect outliers. The effect of the different values of parameter β is illustrated in Figure 3.3-d. We can see that it has little influence on the final results as long as it is larger than 1. As in FCM and ECM, for which it is a usual choice, we use $\beta = 2$ in all the following experiments.

The improvement of precision will bring about the decline of recall, as more data could not be clustered into specific classes. What we should do is to set parameters based on our own requirement to make a tradeoff between precision and recall. For instance, if we want to make a cautious decision in which EP is relatively high, we can reduce γ and α. Values of these parameters can be also learned from historical data if such data are available.

For the objects in the overlapped area, it is difficult to make a hard decision *i.e.* to decide about their specific groups. Thanks to the imprecise clusters, we can make a soft decision. As analysed before, the soft decision will improve the precision of



a. MECM (with respect to γ)

b. MECM (with respect to α)

c. MECM (with respect to η or δ)

d. MECM (with respect to β)

e. ECM (with respect to α or δ)

Figure 3.3: Clustering of overlapped data set with different parameters

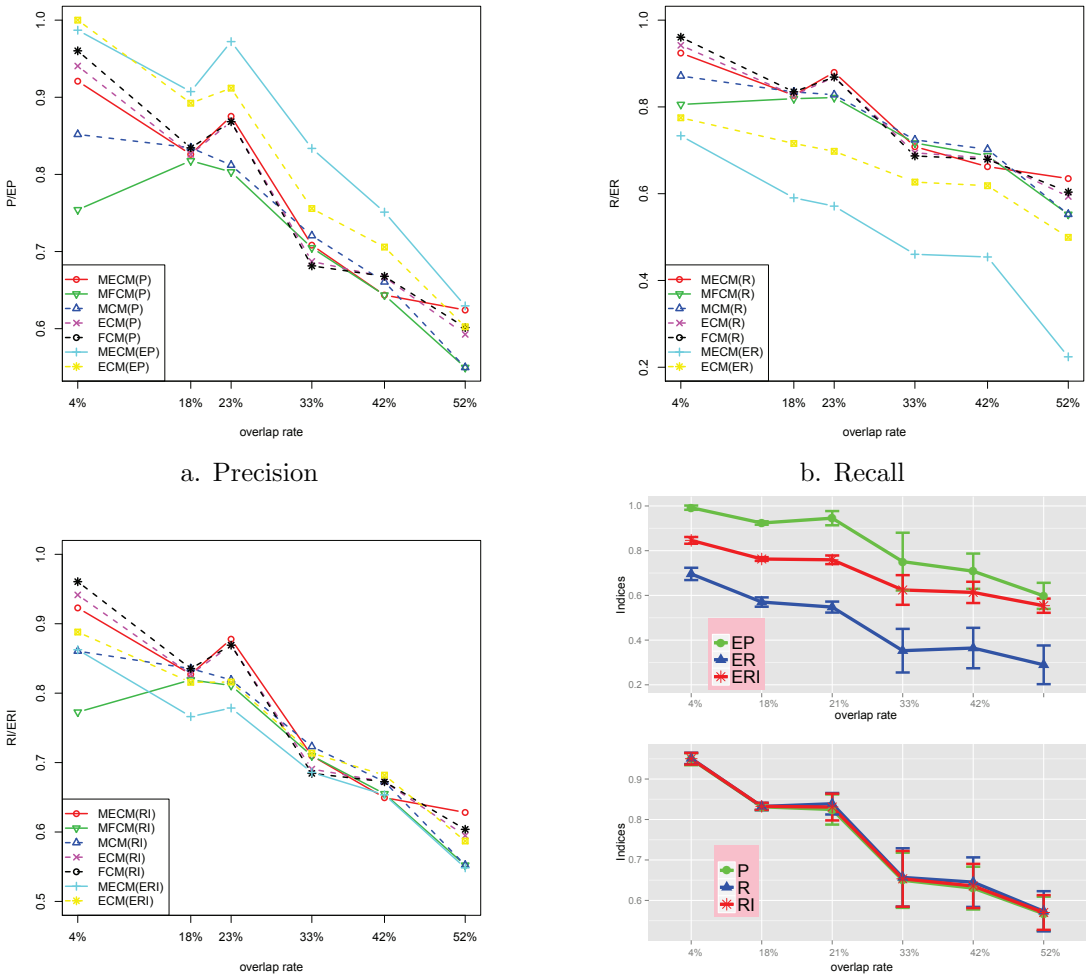
total results and reduce the risk of misclassifications caused by simply partitioning the overlapped objects into specific class. However, too many imprecise decisions will decrease the recall value. Therefore, the ideal partition should make a compromise between the two measures. Set $\alpha = 1.8, \gamma = 0.2, \eta = 0.7$ and $\delta = 50$, the "best" (with relatively high values on both precision and recall) clustering result by MECM is shown in Figure 3.2-b. As we can see, most of the data in the overlapped area are partitioned into imprecise cluster $\omega_{12} \triangleq \{\omega_1, \omega_2\}$ by the application of MECM. We adjust the coordinates of the center of the second circle to get overlapped data with different proportions (overlap rates), and the validity indices of the clustering results by different methods are illustrated in Figure 3.4. For the application of MECM, MCM and MFCM, each algorithm is evoked 20 times with randomly selected initial prototypes for the same data set and the mean values of the evaluating indices are reported. Figure 3.4-d shows the average values of the indices by MECM (plus and minus one standard deviation) for 20 repeated experiments as a function of the overlap rates. As we can see the initial prototypes indeed have effects on the final results, especially when the overlap rates are high. Certainly, we can avoid the influence by repeating the algorithm many times. But this is too expensive for MECM. Therefore, we suggest to use the prototypes obtained in MFCM or MCM as the initial. In the following experiments, we will set the initial prototypes to be the ones got by MFCM.

As it can be seen, for different overlap rates, the classical measures such as precision, recall, and RI are almost the same for all the methods. This reflects that pignistic probabilities play a similar role as fuzzy membership. But we can see that for MECM, EP is significantly high, and the increasing of overlap rates has least effects on it compared with the other methods. Such effect can be attributed to the introduced imprecise clusters which enable us to make a compromise decision between hard ones. But as many points are clustered into imprecise classes, the evidential recall value is low.

Overall, this example reflects one of the superiority of MECM that it can detect overlapped clusters. The objects in the overlapped area could be clustered into imprecise classes by this approach. Other possible available information or special techniques could be utilised for these imprecise data when we have to make hard decisions. Moreover, partitions with different degree of imprecision can be got by adjusting the parameters of the algorithm based on our own requirement.

3.5.2 Classical data sets from Gaussian mixture model

In the second experiment, we test on a data set consisting of $3 \times 50 + 2 \times 5$ points generated from different Gaussian distributions. The first 3×50 points are from Gaussian



a. Precision

b. Recall

c. RI

d. Repeated experiments for MECM

Figure 3.4: Clustering of overlapped data set with different overlap rates. Figure d shows the average values of the indices (plus and minus one standard deviation) for 20 repeated experiments, as a function of the overlap rates. For MECM $\alpha = 1.8, \gamma = 0.2, \eta = 0.7, \delta = 50$.

distributions $G(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)(k = 1, 2, 3)$ with

$$\begin{aligned} \boldsymbol{\mu}_1 =&(0,0)^{\mathrm{T}}, \boldsymbol{\mu}_2 = (40,40)^{\mathrm{T}}, \boldsymbol{\mu}_3 = (80,80)^{\mathrm{T}} \\ \boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 &= \begin{pmatrix} 120 & 0 \\ 0 & 120 \end{pmatrix}, \end{aligned} \tag{3.29}$$

and the last 2×10 data are noisy points follow $G(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)(k = 4, 5)$ with

$$\begin{aligned} \boldsymbol{\mu}_4 =&(-50,90)^{\mathrm{T}}, \boldsymbol{\mu}_5 = (-10,130)^{\mathrm{T}} \\ \boldsymbol{\Sigma}_4 = \boldsymbol{\Sigma}_5 &= \begin{pmatrix} 80 & 0 \\ 0 & 80 \end{pmatrix}. \end{aligned} \tag{3.30}$$

MECM is applied with the following settings: $\alpha = 1, \delta = 100, \eta = 0.7$, while ECM has been tested using $\alpha = 1.7, \delta = 100$ (The appropriate parameters can be determined similarly as in the first example). One can see from Figures 3.5-b and 3.5-c , MCM and MFCM can partition most of the regular data in ω_1, ω_2 and ω_3 into their correct clusters, but they could not detect the noisy points correctly. These noisy data are simply grouped into a specific cluster by both approaches. As can be seen from Figure 3.5-d, for the points located in the middle part of ω_2, ECM could not find their exact group and misclassify them into imprecise cluster ω_{13}. In the figures $\omega_{ij} \triangleq \{\omega_i, \omega_j\}$ denotes imprecise clusters.

As mentioned before, imprecise classes in MECM can measure ignorance and uncertainty at the same time, and the degree of ignorance in meta clusters can be adjusted by γ. We can see that MECM does not detect many points in the overlapped area between two groups if γ is set to 0.6. In such a case the test objects are partitioned into imprecise clusters mainly because of our ignorance about their specific classes. These objects attributed to meta classes mainly belong to noisy data in ω_4 and ω_5. The distance of these points to the prototypes of specific clusters is large (but not too large or they could be regarded to be in the emptyset, see Figure 3.5-e). Thus the distance between the prototype vectors is relatively small so that these specific clusters are indistinguishable. Decreasing γ to be 0.2 would make imprecise class denoting more uncertainty, as it can be seen from Figure 3.5-f, where many points located in the margin of each group are clustered into imprecise classes. In such a case, meta classes rather reflect our uncertainty on the data objects' specific cluster.

Table 3.2 lists the indices for evaluating the different methods. Bold entries in each column of this table (and also other tables in the following) indicate that the results are significant as the top performing algorithm(s) in terms of the corresponding evaluation index. We can see that the precision, recall and RI values for all approaches are similar except from those obtained for ECM which are significantly lower. As these classical measures are based on the associated pignistic probabilities for evidential clusterings, it

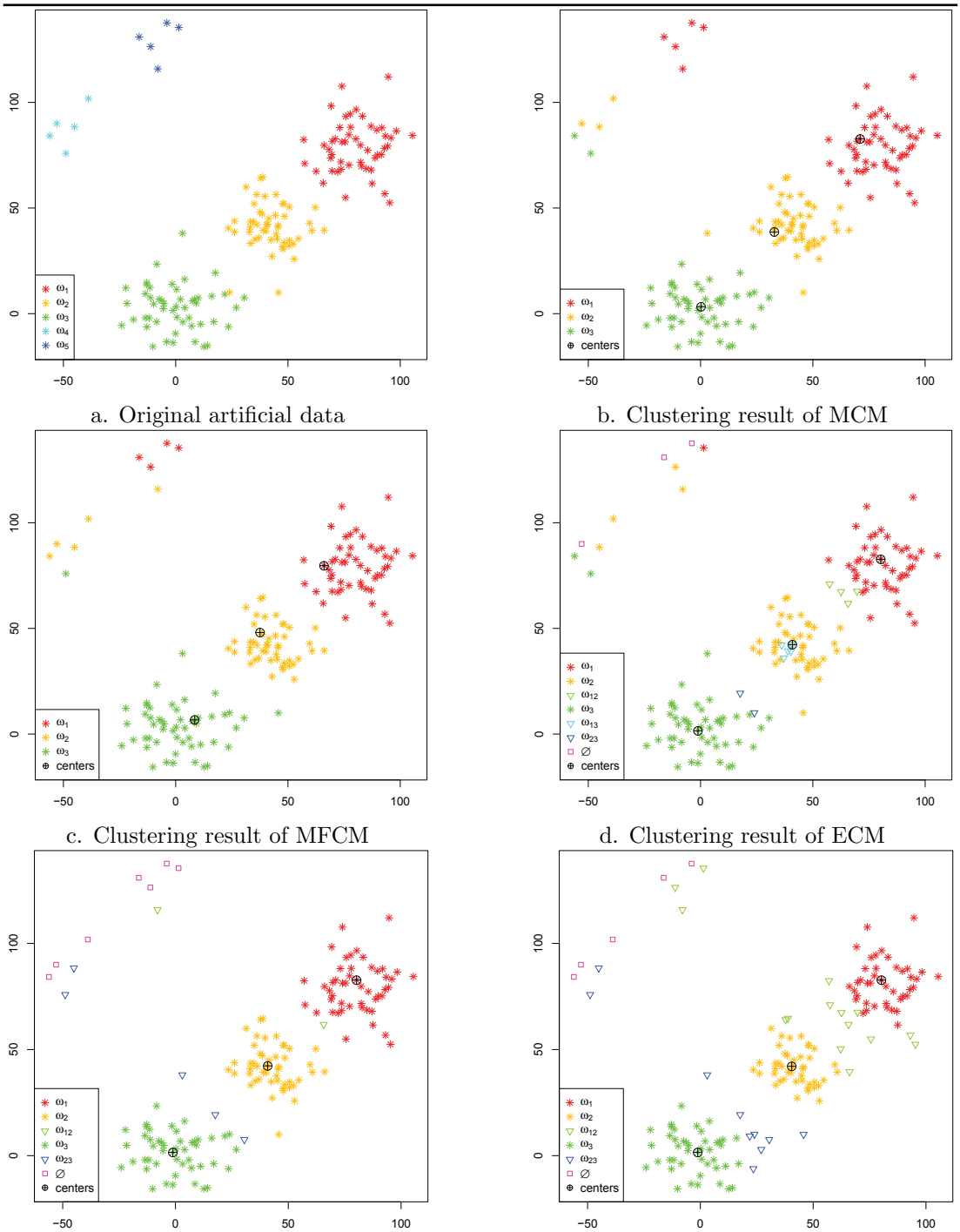
seems that credal partitions can provide the same information as crisp and fuzzy ones. But from the same table, we can also see that the evidential measures EP and ERI obtained for MECM are higher (for hard partitions, the values of evidential measures equal to the corresponding classical ones) than the ones obtained for other methods. This fact confirms the accuracy of the specific decisions *i.e.* decisions clustering the objects into specific classes. The advantage can be attributed to the introduction of imprecise clusters, with which we do not have to partition the uncertain or unknown objects into a specific cluster. Consequently, it could reduce the risk of misclassification. However, although ECM also deals with imprecise clusters, the accuracy is not improved as much as in the case of applying MECM. As illustrated before in the case of ECM application, many objects of a specific cluster are partitioned into an irrelevant imprecise class and, as a result, the evidential precision value and ERI decrease as well.

Table 3.2: The clustering results for Gaussian data sets by different methods. For each method, we generate 20 data sets with the same parameters and report the mean values of the evaluation indices for all the data sets.

	Precision	Recall	RI	EP	ER	ERI
MCM	0.7802	0.9570	0.9002	0.7802	0.9570	0.9002
MFCM	0.8616	0.9797	0.9484	0.8616	0.9797	0.9484
FCM	0.8644	0.9820	0.9500	0.8644	**0.9820**	0.9500
ECM	0.8215	0.9353	0.9222	0.9069	0.8436	0.9294
MECM ($\gamma = 0.2$)	**0.8674**	**0.9855**	**0.9520**	**0.9993**	0.7721	0.9336
MECM ($\gamma = 0.6$)	0.8662	0.9851	0.9515	0.9958	0.9586	**0.9868**

We also test on "Iris flower", "cat cortex" and "protein" data sets (Fisher, 1936; Graepel et al., 1999; Hofmann and Buhmann, 1997). The first is object data while the other two are relational data sets. Thus we compare our method with FCM and ECM for the Iris data set, and with RECM and NRFCM (Non-Euclidean Relational Fuzzy Clustering Method (Hathaway and Bezdek, 1994)) for the last two data sets. The results are displayed in Figure 3.6.

Presented results allow us to sum up the characteristics of MECM. Firstly, one can see that the behavior of MECM is similar to ECM for traditional data. Besides, credal partitions provided by MECM allow to recover the information of crisp and fuzzy partitions. Moreover, we are able to balance influence of our uncertainty and ignorance according to the actual needs. The examples utilised before deal with classical data sets. But the superiority of MECM makes it applicable in the case of data sets for which only dissimilarity measures are known *e.g.* social networks. Thus in the following experiments, we will use some graph data to illustrate the behaviour of the proposed



a. Original artificial data

b. Clustering result of MCM

c. Clustering result of MFCM

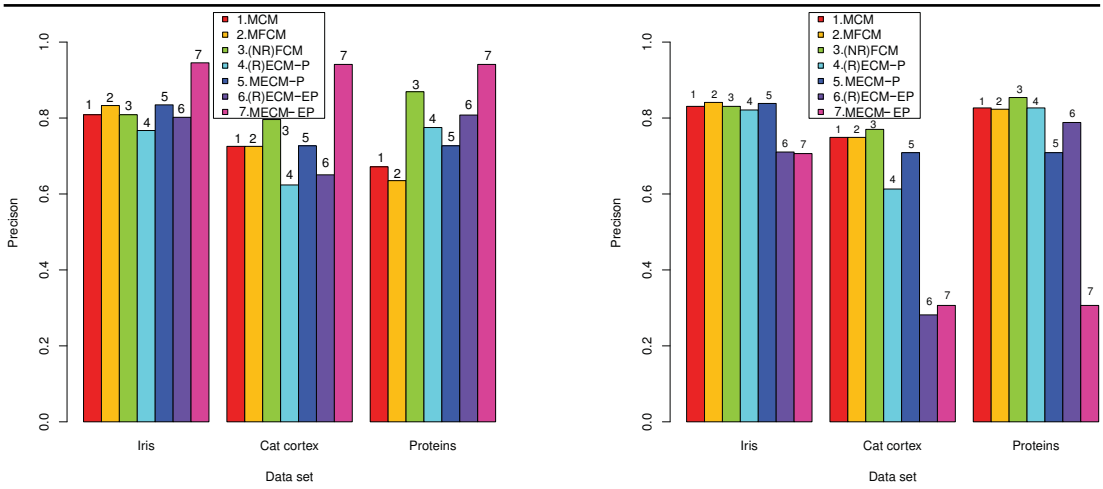
d. Clustering result of ECM

e. Clustering result of MECM ($\gamma = 0.6$)

f. Clustering result of MECM ($\gamma = 0.2$)

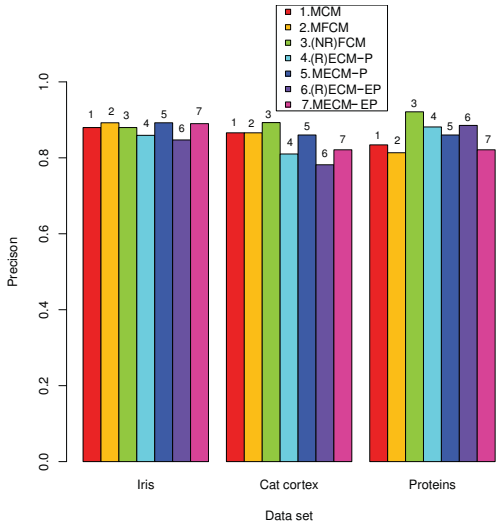
Figure 3.5: Clustering of Gaussian data set by different methods.

method on the community detection problem in social networks. The dissimilarity index used here is the one brought forward by Zhou (2003). To have a fair comparison, in the following experiments, we also compare with three classical algorithms for community detection *i.e.* BGLL (Blondel et al., 2008), LPA (Raghavan et al., 2007) and ZFCM (a fuzzy c-means based approach proposed by Zhang et al. (2007)). The obtained community structures are compared with known performance measures, *i.e.* NMI (Normalized Mutual Information), VI (Variation of Information) and Modularity. Note that the NMI and VI are calculated using the detected community structure and the benchmark. The more the result is similar to the benchmark, the larger the value of NMI is, and the smaller the value of VI is.



a. Precision

b. Recall



c. RI

Figure 3.6: Clustering results of different UCI data sets.

3.5.3 Artificial graphs and generated benchmarks

To show the performance of the algorithm in detecting communities in networks, we first apply the method to a sample network generated from Gaussian mixture model. This model has been used for testing community detection approaches by Liu and Liu (2010).

The artificial graph is composed of 3×50 nodes, $n_1, n_2, \cdots, n_{150}$, which are represented by 150 sample points, $x_1, x_2, \cdots, x_{150}$, in two-dimensional Euclidean space. There are 3×50 points generated from Gaussian distributions $G(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)(k = 1, 2, 3)$ with

$$\boldsymbol{\mu}_1 = (1.0,4.0)^{\mathrm{T}}, \boldsymbol{\mu}_2 = (2.5, 5.5)^{\mathrm{T}}, \boldsymbol{\mu}_3 = (0.5, 6.0)^{\mathrm{T}} \tag{3.31}$$
$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \begin{pmatrix} 0.25 & 0 \\ 0 & 0.25 \end{pmatrix}.$$

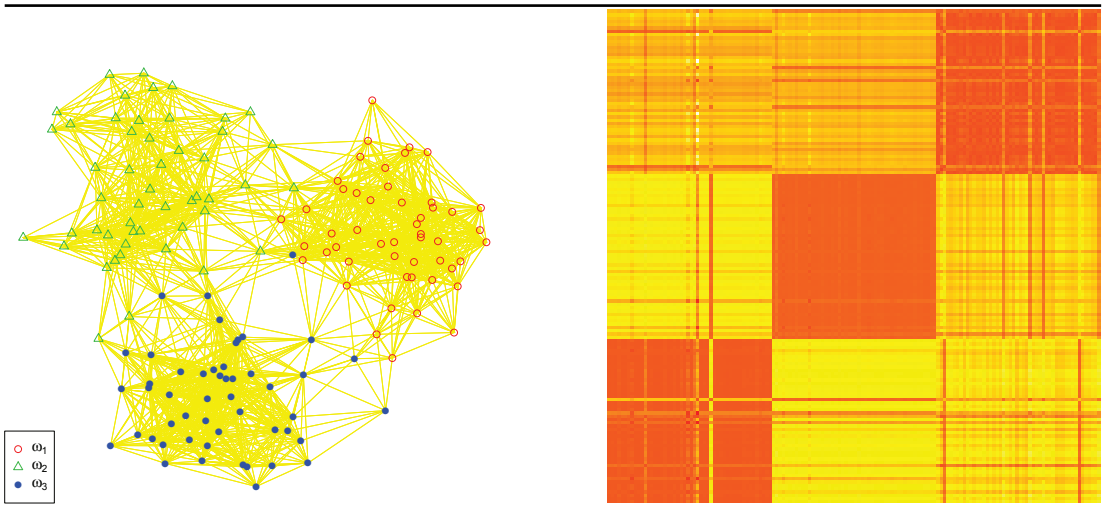
Then, the edges of the graph are generated by the following thresholding strategy: if $|x_i - x_j| \leq dist$, we set an edge between nodes n_i and n_j; Otherwise the two nodes are not directly connected. The graph is shown in Figure 3.7-a (with $dist = 0.8$) and the dissimilarity matrix of the nodes is displayed in Figure 3.7-b. From the figures we can see that there are three significant communities in the graph, and some nodes in the bordering of their groups seem to be in overlapped classes as they contact with members in different communities simultaneously.

Table 3.3 lists the indices for evaluating the results. It shows that MECM performs well as the evidential precision resulting from its application is high. MECM utilization also results in decreasing the probabilities of clustering failure thanks to the introduction of imprecise clusters. This makes the decision-making process more cautious and reasonable.

Table 3.3: The results for Gaussian graph by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.9049	0.9110	0.9392	0.9049	0.9110	0.9392	0.8282	0.3769	0.6100
MFCM	0.9067	0.9099	0.9396	0.9067	0.9099	0.9396	0.8172	0.4013	0.6115
ZFCM	0.9202	0.9224	0.9482	0.9202	0.9224	0.9482	0.8386	0.3545	0.6118
MECM	**0.9470**	**0.9472**	**0.9652**	**0.9789**	0.6060	0.8661	**0.8895**	**0.2428**	0.6072
BGLL	0.9329	0.9347	0.9564	0.9329	0.9347	**0.9564**	0.8597	0.3081	**0.6119**
LPA	0.3289	1.0000	0.3289	0.3289	1.0000	0.3289	0.0000	1.0986	0.0000

The algorithms are also compared by means of Lancichinetti et al. (2008) benchmark



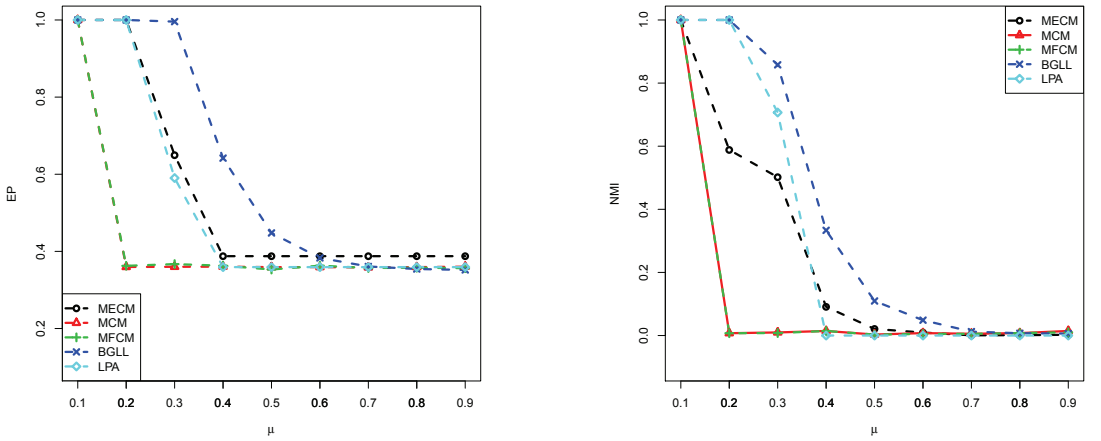
a. Gaussian network b. Dissimilarity matrix

Figure 3.7: Artificial network from Gaussian mixture model.

(LFR) networks. The results of different methods in two kinds of LFR networks with 500 and 1000 nodes are displayed in Figures 3.8–3.9 respectively. The parameter μ showed in the x-axis in the figures identifies whether the network has clear communities. When μ is small, the graph has well community structure. In such a case, almost all the methods perform well. But we can see that when μ is large, the results by MECM have the largest values of precision. It means that the decisions which partition the nodes into a specific cluster are of great confidence. In terms of NMI, the results are similar to those by BGLL and LPA, but better than those of MCM and MFCM. This fact well explains that the hard or fuzzy partitions could be recovered when necessary.

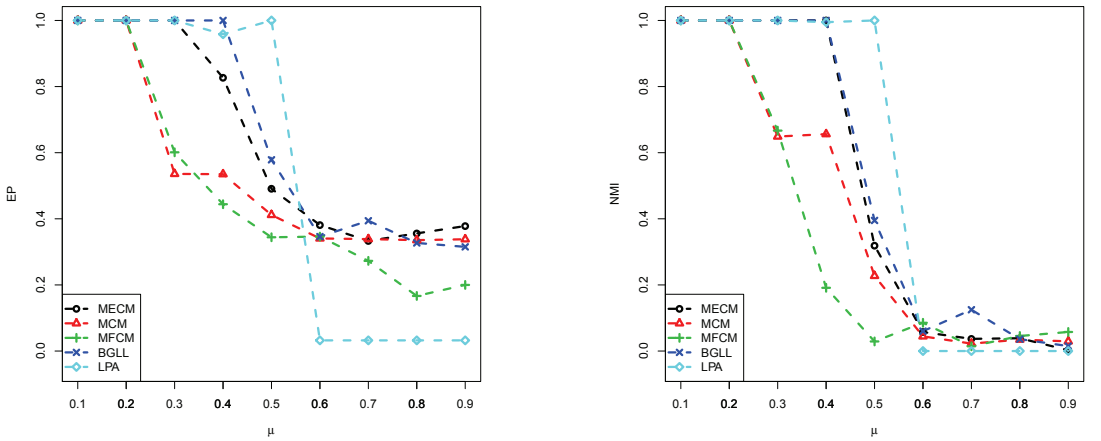
3.5.4 Some real-world networks

A. Zachary's Karate Club. The Zachary's Karate Club data (Zachary, 1977) is an undirected graph which consists of 34 vertices and 78 edges. The edges describe the friendship between the members of the club observed by Zachary in his two-year study. During the course of the study, a disagreement developed between the administrator of the club and the club's instructor, which ultimately resulted in the instructor's leaving and starting a new club, taking about a half of the original club's members with him. The original graph and the dissimilarity of the nodes are shown in Figures 3.10-a and 3.10-b respectively.



a. Precision b. NMI

Figure 3.8: Comparison of MECM and other algorithms in LFR networks. The number of nodes is $n = 500$. The average degree is $|k| = 15$, and the pair for the exponents is $(\gamma, \beta) = (2, 1)$.



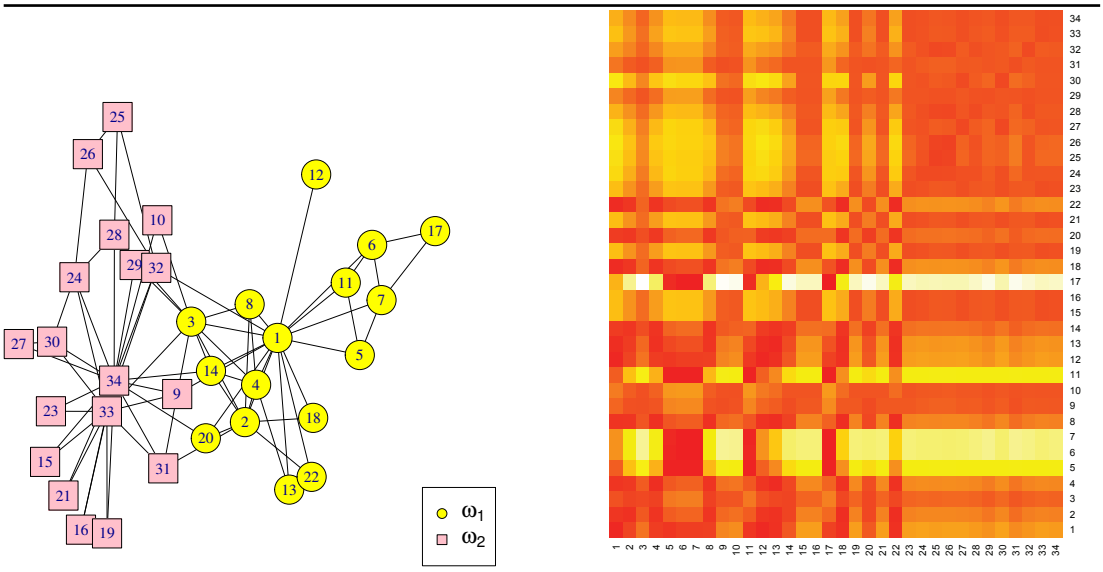
a. Precision b. NMI

Figure 3.9: Comparison of MECM and other algorithms in LFR networks. The number of nodes is $n = 1000$. The average degree is $|k| = 20$, and the pair for the exponents is $(\gamma, \beta) = (2, 1)$.

Let the parameters of MECM be $\alpha = 1.5, \delta = 100, \eta = 0.9, \gamma = 0.6$. The modularity functions by MECM, MCM, MFCM and ZFCM (Figure 3.12-a) peak around $c = 2$ and $c = 3$. Let $c = 2$, all the methods can detect the two communities exactly. If we set $c = 3$, a small community, which can also be found in the dissimilarity matrix (Figure 3.10-b), is separated from ω_1 by all the approaches (see Figure 3.11). But ZFCM assigns the maximum membership to ω_1 for node 9, which is actually in ω_2. It seems that the loss of accuracy in the mapping process may cause such results.

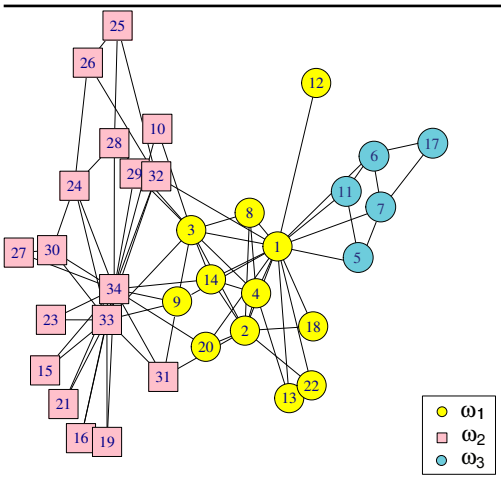
MECM does not find imprecise groups when $\gamma = 0.6$ as the network has apparent community structure, and this reflects the fact that the communities are distinguishable for all the nodes. But there may be some overlap between two communities. The nodes in the overlapped cluster can be detected by decreasing γ (increasing the uncertainty for imprecise communities). As is displayed in Figure 3.11-c and d, by declining γ to 0.1 and 0.05 respectively (the other parameters remain unchanged), nodes 3 and 9 are clustered into both ω_1 and ω_2 (ω_{12}) one after another.

From the results, we can see that MECM takes both the ignorance and the uncertainty into consideration while introducing imprecise communities. The degree of ignorance and uncertainty could be balanced through adjusting γ. The analysis shows that there appears only uncertainty without ignorance in the original club network. In order to show the performance of MECM when there are noisy conditions such that some communities are indistinguishable, two noisy nodes are added to the original graph in the next experiment.

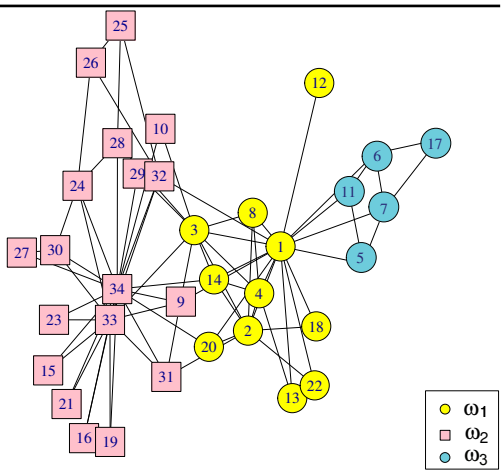


a. Original Karate Club network b. Dissimilarity matrix

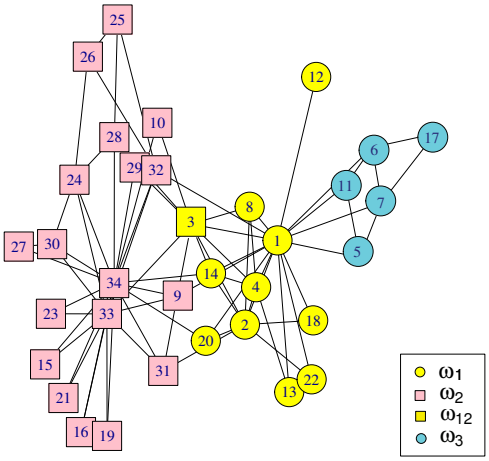
Figure 3.10: Original Karate Club network.



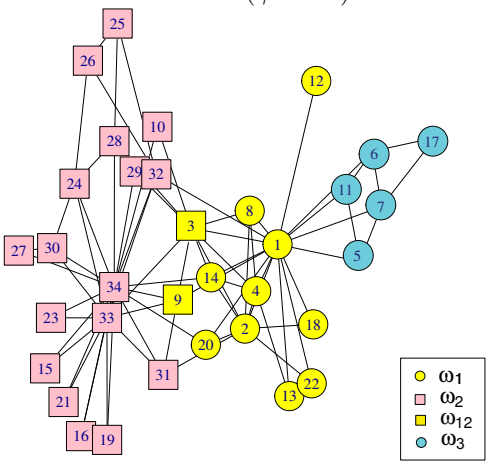
a. Clustering result of ZFCM



b. Clustering result of MCM, MFCM, and MECM ($\gamma = 0.1$)



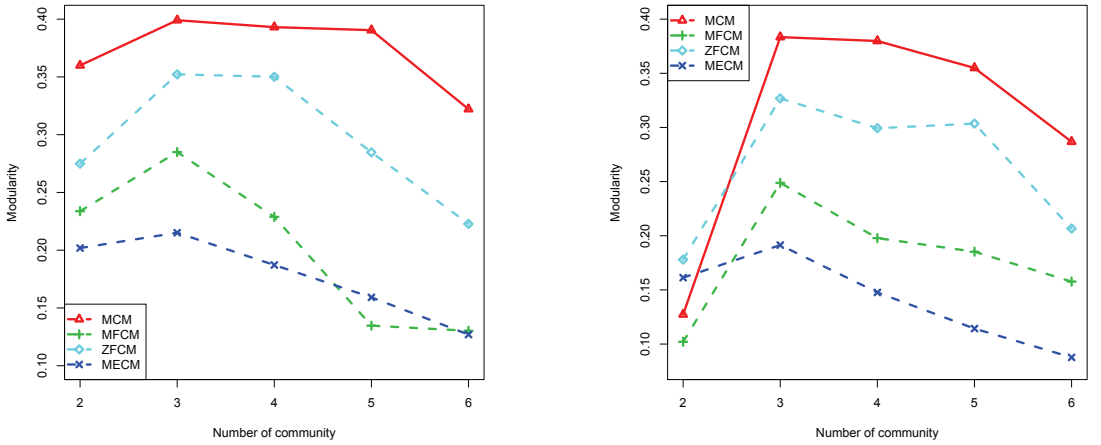
c. Clustering result of MECM ($\gamma = 0.1$)



d. Clustering result of MECM ($\gamma = 0.05$)

Figure 3.11: Detected communities of Karate Club network by different methods.

B. Karate Club network with some added noisy nodes. In this test, two noisy nodes are added to the original Karate Club network (see Figure 3.13-a). The first one is node 35, which is directly connected with nodes 18 and 27. The other one is 36, which is connected to nodes 1 and 33. It can be seen from the dissimilarity matrix that node 36 has stronger relationships with both communities than node 35. This is due to the fact that the nodes connected to node 36 play leader roles in their own



a. Original Karate Club b. Karate Club with noisy nodes

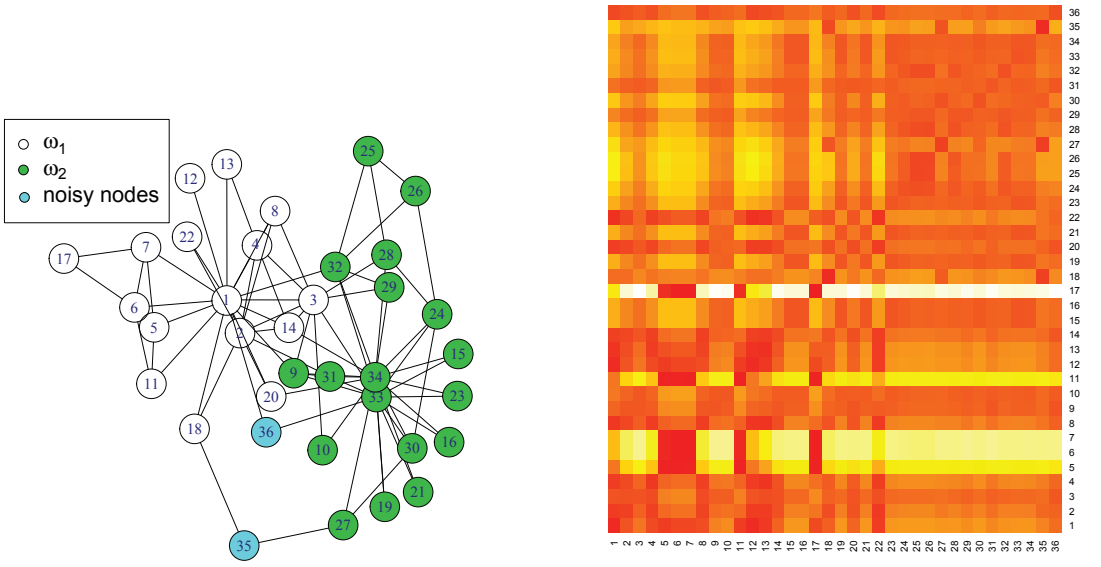
Figure 3.12: Modularity functions of Karate Club network by different methods.

group, but node 35 contacts with two marginal nodes with "small" or insignificant roles in their own groups only.

The results obtained by the application of different methods are shown in Figure 3.14. The MECM parameters are set as follows: $\alpha = 1.5, \delta = 100, \eta = 0.9$ and γ is tuned according to the extent that the imprecise communities reflect our ignorance. As we can see, MCM, MFCM and ZFCM simply group the two noisy nodes into ω_1. With $\gamma = 0.4$, MECM regards node 36 as a member of ω_1 while node 35 is grouped into imprecise community ω_{12}. And ω_{12} mainly reflects our ignorance rather than uncertainty on the actual community of node 36. This is why node 36 is not clustered into ω_{12} since ω_1 and ω_2 are distinguishable for him but we are just not sure for the final decision. The increase in the extent of uncertainty in imprecise communities results from the decrease of γ value. We can see that more nodes (including nodes 36,9,1,12,27, see Figures 3.14-e and 3.14-f) are clustered into ω_{12} or ω_{13} due to uncertainty. The imprecise communities consider both ignorance (node 35) and uncertainty (other nodes).

These results reflect the difference between ignorance and uncertainty. As node 35 is only related to one outward node of each community, thus we are ignorant about which community it really belongs to. On the contrary, node 36 connects with the key members (playing an important role in the community), and in this case the dissimilarity between the prototypes of ω_1 and ω_2 is relatively large so they are distinguishable. Thus there is uncertainty rather than ignorance about which community node 36 is inside. In this network, node 36 is a "good" member for both communities, whereas node 35 is a "poor" member. It can be seen from Figure 3.15-a that the fuzzy partition by MFCM also gives large similar membership values to ω_1 and ω_2 for node 35, just like in the case of such good members as nodes 36 and 9. The obtained results show the

problem of distinguishing between ignorance and the "equal evidence" (uncertainty) for fuzzy partitions. But Figure 3.15-b shows that the credal partition by MECM assigns small mass belief to ω_1 and ω_2 for node 35, indicating our ignorance on its situation.



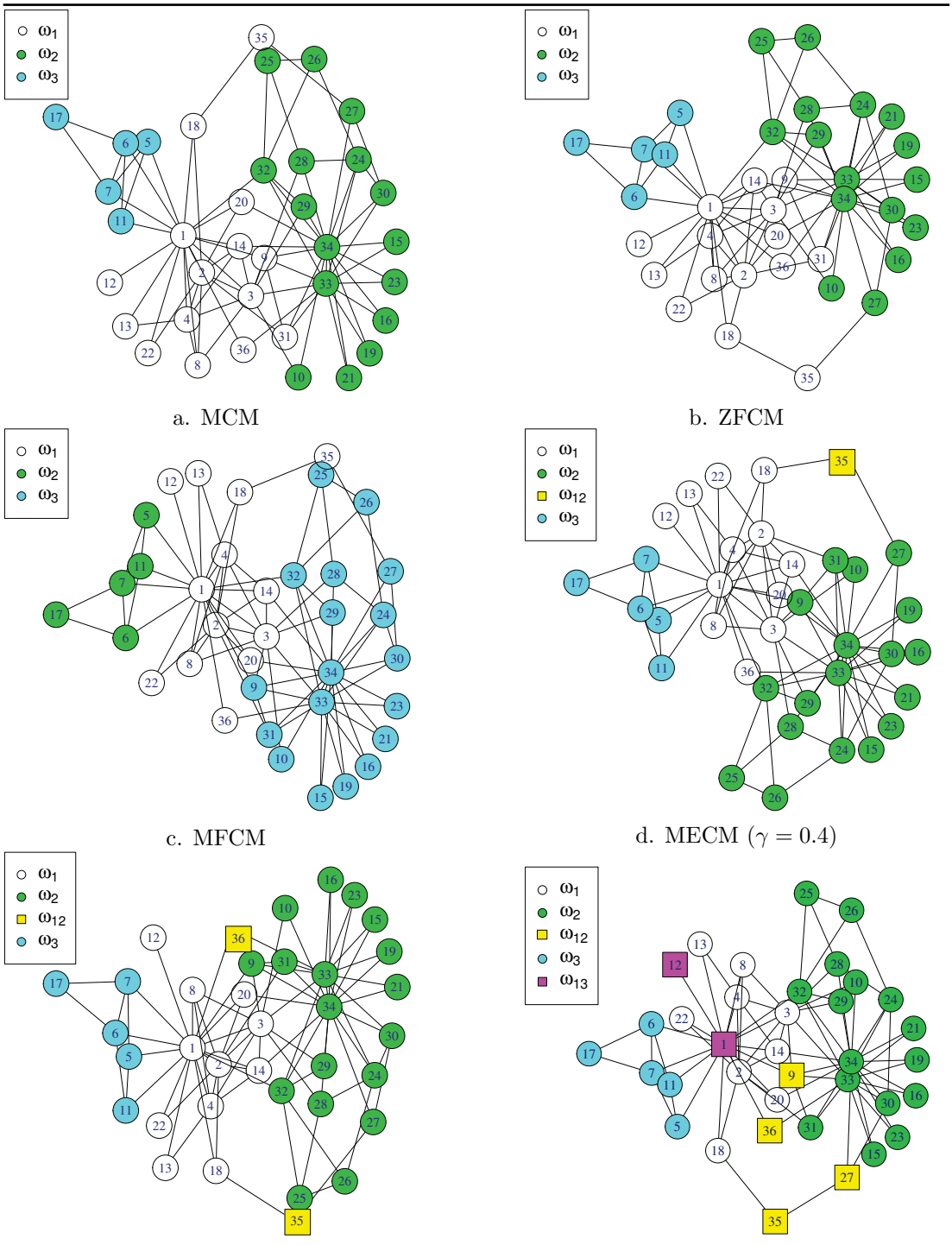
a. Karate Club network with added nodes

b. Dissimilarity matrix

Figure 3.13: Karate Club network with two noisy nodes.

We also test our method on four other real-world graphs: American football network, Dolphins network, Lesmis network and Political books network[3]. The measures applied to evaluate the performance of different methods are listed in Tables 3.4–3.7. It can been seen from the tables, for all the graphs MECM application results in a community structure with high evidential precision level. The precision results from a cautious decision making process which clusters the noisy nodes into imprecise communities. In terms of classical performance measures like NMI, VI and modularity, MECM slightly outperforms the other algorithms. Note that these classical measures for hard partitions are calculated by the pignistic probabilities associated with the credal partitions provided by MECM. Therefore, we can also see the possibility to recover the hard decisions here when using the proposed evidential detection approach.

[3]These data sets can be found in http://networkdata.ics.uci.edu/index.php



a. MCM

b. ZFCM

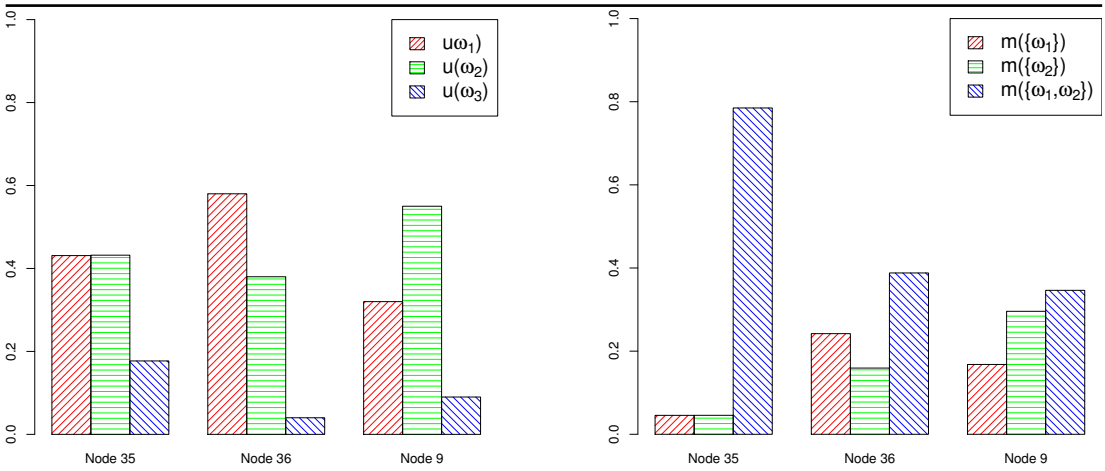
c. MFCM

d. MECM ($\gamma = 0.4$)

e. MECM ($\gamma = 0.1$)

f. MECM ($\gamma = 0.02$)

Figure 3.14: Detected communities in Karate Club network with noisy nodes.



a. Fuzzy membership by MFCM b. Mass belief by MECM

Figure 3.15: Fuzzy membership and mass assignments for the nodes in Karate Club network with noisy nodes.

Table 3.4: The results for American football network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.7416	0.8834	0.9661	0.7416	0.8834	0.9661	0.8637	0.6467	0.5862
MFCM	0.7583	0.8757	0.9678	0.7583	0.8757	0.9678	0.8715	0.6160	0.5745
ZFCM	0.8176	0.9082	0.9765	0.8176	0.9082	0.9765	0.9035	0.4653	0.6022
MECM	**0.8232**	0.9082	**0.9771**	**0.9303**	0.8681	**0.9843**	**0.9042**	**0.4625**	0.5995
BGLL	0.7512	**0.9120**	0.9689	0.7512	**0.9120**	0.9689	0.8903	0.5195	**0.6046**
LPA	0.6698	0.8298	0.9538	0.6698	0.8298	0.9538	0.8623	0.6580	0.5757

3.5.5 Discussion

We will discuss for which application MECM is designed here. As analysed before, for MECM only dissimilarities between objects are required and only the intuitive assumptions need to be satisfied for the dissimilarity measure. Therefore, the algorithm could be appropriate for many clustering tasks for non-metric data objects. This type of data is very common in social sciences, psychology, etc, where any metric assumptions about the similarities/dissimilarities could not be assured. The freedom for the data set leads to the restriction that the prototypes should be the objects themselves. Nevertheless, this constraint seems reasonable for social networks as the center of a community is usually the person (node) frequently contacting with others. Thus the approach can be applied to community detection problems. Thanks to the introduction of imprecise classes, it could reduce the risk of partitioning the objects which we are uncertain or ignorant into an incorrect cluster. For this reason the algorithm can help us make soft decisions when clustering the data set without distinct cluster/community structures

Table 3.5: The results for Dolphins network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	1	1	1	1	1	1	1	0	0.3787
MFCM	1	1	1	1	1	1	1	0	0.3787
ZFCM	1	1	1	1	1	1	1	0	0.3787
MECM	1	1	1	1	1	1	1	0	0.3787
BGLL	0.9271	0.3583	0.6351	0.9271	0.3583	0.6351	0.4617	1.1784	0.5185
LPA	0.9250	0.5029	0.7070	0.9250	0.5029	0.7070	0.5595	0.8354	0.5070

Table 3.6: The results for Lesmis network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.6109	0.5522	0.9005	0.6109	0.5522	0.9005	0.7381	1.1295	0.4732
MFCM	0.5774	0.6456	0.8971	0.5774	0.6456	0.8971	0.7743	0.9555	0.4705
ZFCM	**0.7368**	0.5769	0.9217	0.7368	0.5769	0.9217	0.7805	0.9666	0.4983
MECM	0.7065	0.7473	**0.9299**	**0.9298**	0.4368	**0.9258**	**0.7977**	**0.8531**	0.4884
BGLL	0.5796	0.8104	0.9033	0.5796	0.8104	0.9033	0.7551	0.9435	**0.5556**
LPA	0.4594	**0.9643**	0.8544	0.4594	**0.9643**	0.8544	0.7500	0.8637	0.5428

or with overlap.

Due to the computational complexity, the proposed algorithm is not well directly adapted to handle very large data sets. However, here we discuss the possibility to apply the evidential community detection approach to large-scale networks. Firstly, the number of parameters to be optimized is exponential and depends on the number of clusters (Masson and Denoeux, 2008). For the number of classes larger than 10, calculations are not tractable. But we can consider only a subclass with a limited number of focal sets (Masson and Denoeux, 2008). For instance, we could constrain the focal sets to be composed of at most two classes (except Ω). Secondly, for the network with millions of nodes, MCM or MFCM could be evoked as a first step to merge some nodes into small clusters. After that we can apply MECM to the "coarsened" network. But how to define the edges or connections of the new graph should be studied. Lastly we emphasize that the evidential community detection algorithm could be utilised for gaining a better insight into the network structure and detecting the imprecise classes. For the large-scale network, it is difficult to make specific decisions for all of nodes due to the limitation of time, money or techniques. In this case, we can use the proposed approach to make some "soft" decisions first and then use some techniques special for the imprecise parts of the graph.

Table 3.7: The results for Political books network by different methods.

	Precision	Recall	RI	EP	ER	ERI	NMI	VI	Modularity
MCM	0.8109	0.8030	0.8482	0.8109	0.8030	0.8482	0.5721	0.8426	0.4979
MFCM	0.8020	0.8187	**0.8485**	0.8020	0.8187	**0.8485**	**0.5755**	0.8256	0.4962
ZFCM	0.7928	0.7487	0.8234	0.7928	0.7487	0.8234	0.5301	0.9407	0.5048
MECM	0.7880	0.8081	0.8383	**0.8458**	0.6435	0.8128	**0.5755**	0.8247	0.4725
BGLL	**0.8244**	0.6203	0.7978	0.8244	0.6203	0.7978	0.5121	1.0987	**0.5205**
LPA	0.7331	**0.8558**	0.8200	0.7331	**0.8558**	0.8200	0.5612	**0.7925**	0.4604

3.6 Conclusion

We introduced a Median variant of Evidential C-means (MECM) as a new prototype-based clustering algorithm in this chapter. The proposed approach is an extension of median c-means and median fuzzy c-means. It is based on the framework of belief function theory. The applied median-based clustering requires the definition of the dissimilarity between objects only. Therefore, it is not restricted to a metric space application. The prototypes of the clusters are constrained to the data objects themselves. MECM provides us with not only credal partitions but also hard and fuzzy partitions as by-products through computing pignistic probabilities. Moreover, it could distinguish ignorance from uncertainty while the fuzzy or crisp partitions could not. By the introduced imprecise clusters, we could find some overlapped and indistinguishable clusters for related nodes. Thanks to the advantages of belief function theory and median clustering, MECM could be applied to community detection problems in social networks. As other median clustering approaches, MECM tends to get stuck in local minima such that several runs have to be performed to obtain good performance. However, we propose an initial prototype-selection scheme using the evidential semi-centrality for the application of MECM in community detection to solve the problems brought by the initial prototypes. Results of presented experiments on artificial and real-world networks show that the credal partitions on graphs provided by MECM application are more refined than crisp and fuzzy ones. Therefore, they could enable us to gain a better understanding of analysed community structure. Some examples on the classical metric space are also given to illustrate the interest of MECM and to show its difference with respect to the existing methods.

In MECM, we ignore "multi-center" to avoid the troubles brought by the need for an initial seed using ESC and the definition of a threshold to control the distance between prototypes. Nevertheless, many real-world networks may have more than one center. Therefore, we will include the feature of multi-center clustering in the sequel of this report.

4
Similarity-based community detection with multiple prototypes

4.1 Overview

Some existing community detection algorithms, including the model based on MECM proposed in the last chapter, use a single prototype to represent an individual group. As we mentioned, in real applications this may not adequately model the different types of communities and hence limits the clustering performance on graphs. To address this problem, a Similarity-based Multi-Prototype (SMP) community detection approach is proposed in this chapter (Zhou et al., 2015a). In SMP, vertices in each community carry various weights to describe their degree of representativeness. This mechanism enables each community to be represented by more than one node. The centrality of nodes is used to calculate prototype weights, while similarity is utilized to guide us to partitioning the graph. Experimental results on computer generated and real-world networks clearly show that SMP performs well for detecting communities. Moreover, the method can provide richer information for the inner structure of the detected communities with the help of prototype weights compared with the existing community detection models.

4.2 The multi-prototype community detection approach

We propose our method in this section . After an introduction of the concept of representative weights (also called prototype weights) in Section 4.2.1, the whole algorithm will be presented in detail in Section 4.2.2. The problem of determining the optimum community number and the complexity of the algorithm will be discussed in Section 4.2.3 and Section 4.2.4 respectively.

4.2.1 The prototype weights

Suppose $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$ is a partition of a graph $G(V, E)$, where V is the set of nodes and E is the set of edges. The n nodes in the graph can be denoted by $\{n_1, n_2, \cdots, n_n\}$. The matrix $\boldsymbol{V}_{c\times n}$ denotes the prototype weights of n nodes with respect to all the c communities. As analyzed before, the centrality value of a node

can be used to express the belief that the node plays the center role in its community. Therefore, the weight of node n_j's degree of representativeness in cluster ω_r can be derived as below:

$$V_{rj} = \begin{cases} \dfrac{Cen(j)}{\sum\limits_{\{h:n_h \in \omega_r\}} Cen(h)} & n_j \in \omega_r \\ 0 & n_j \notin \omega_r, \end{cases} \qquad r = 1, 2, \cdots, c,\ j = 1, 2, \cdots, n, \tag{4.1}$$

where $Cen(j)$ is the centrality of node n_j in the subgraph corresponding community ω_r. Then, for a given node n_i, the similarity between n_i and community ω_j, denoted by $\bar{s}_{ij}$, can be obtained as

$$\bar{s}_{ij} = \sum_{h=1}^{n} v_{jh} s_{ih}, \tag{4.2}$$

where s_{ih} is the similarity between nodes n_i and n_h. From Eqs. (4.1) and (4.2) we can see that $\bar{s}_{ij}$ is a weighted sum of the similarity between node n_i and all the nodes in community ω_j, and the weights used in the summation depend on the contribution of the nodes to their own community.

4.2.2 The detection algorithm

The whole SMP algorithm to detect communities in social networks is summarized as Algorithm 3. In fact SMP is a variation of c-means, c-medoids and c-rank. The difference between SMP and the other three clustering algorithms lies in the manner of updating the prototypes. c-means uses the average value to represent every class while c-medoids and c-rank uses one "most possible" object. On the contrary, SMP adopts an effective multi-prototype representation based on the determined prototype weights of each member in the group. Due to the various types of community structures, the way to represent a cluster using multiple prototypes is more reasonable in real applications. Moreover, SMP often needs fewer iterations than c-means to make the algorithm convergent.

Remark 3. As we can see, SMP provides us a crisp (hard) partition of the analyzed network. Also the similarity between node n_i and community ω_j could be obtained by Eq. (4.2). Then the node n_i's membership with regard to community ω_j can be defined as follows:

$$u_{ij} = \frac{\bar{s}_{ij}}{\sum\limits_{h=1}^{c} \bar{s}_{ih}}, \quad i = 1, 2, \cdots, n,\ j = 1, 2, \cdots, c. \tag{4.3}$$

This form of membership measure is in line with that got by FCM algorithm, where the membership values assigned to an object are inversely related to the relative distance to the cluster. Similarly here the memberships in Eq. (4.3) are determined by the relative similarities. One of the problem of fuzzy membership has been reported is that it could

Algorithm 3 : The Similarity-based Multi-Prototype (SMP) community detection algorithm

Input: c, the number of communities; $\boldsymbol{A}$, the adjacency matrix; $\boldsymbol{W}$, the weight matrix (if any); $N_{\max}$, the maximum number of iterations.
Initialization:
(1). Select the top c nodes with highest centralities as the initial c prototypes.
(2). Calculate the similarity matrix between any two nodes in the graph.
(3). Extract the similarity matrix between the nodes and the prototypes. Partition the node into the community to which its nearest prototype belongs, and get the initial c classes of the graph: $\omega_1, \omega_2, \cdots, \omega_c$.
repeat
(4). Update the matrices $\boldsymbol{V}_{c\times n}$ recording prototype weights of n nodes with respect to all the c communities based on the current partitions using Eq. (4.1).
(5). Calculate the similarity between node n_i and community ω_j, $\bar{s}_{ij}$, using Eq. (4.2), and then cluster the vertices into k communities with every node being in the community it is most similar to.
until All the detected communities remain unchanged or the number of iterations comes to $N_{\max}$.
Output: The membership of each node and the prototype weights of all the members in each community.

not distinguish between "equal evidence" (membership values are large and equal for a number of alternatives) and "ignorance" (all the membership values are equal but very close to zero) (Krishnapuram and Keller, 1993; Pal et al., 2005). If node n_i is equidistant from more than one community, the membership of each cluster will be the same, regardless of the absolute values of the similarity to the communities. Consequently, the fuzzy membership could not be applied to detect noise objects (outliers) which are far but equidistant to some communities (Pal et al., 2005). In SMP, the prototype weights can help us solve this problem, which we will show in detail in Section 4.3.2.

4.2.3 Determining the number of communities

In the first step of SMP algorithm, the additional information about the number of communities should be specified. This is also a fundamental issue in classical c-means and FCM clusterings. In fact, to determine the optimal number of clusters is an open problem for prototype-based clustering methods. Most of the methods to solve this problem consist in computing a validity index from several community structures detected with different values of c and looking for a minimum or maximum of a given criterion (Hu et al., 2008; Zhang et al., 2007; Nepusz et al., 2008). In this chapter MM–modularity (Eq. (2.61)) is used to estimate a proper c. The modularity values signify the quality of the detected communities. When the modularity achieves the maximum, we can get the best c.

4.2.4 The complexity of SMP algorithm

The complexity of SMP consists of calculating similarities and centralities of nodes and iterative process. If we use signal similarity and evidential semi-local centrality measures, as we will see in Section 4, the corresponding time complexity if $O(p(|k|+1)n^2)$ (Hu et al., 2008) and $O(n|k|^2)$ (Gao et al., 2013), where p is the number of propagations, $|k|$ is the average degree of vertices in the network, and n is the number of nodes. The iterative technique is similar to that in c-means. The only difference is the strategy of updating the prototypes. The c-means computes the average value of all the members in the cluster, while SMP tries to find prototype weights of all the members. As the communities are subgraphs which are much smaller than the original network, the updating prototype weights process of SMP does not cost much. If the number of communities c is fixed, the time complexity of c-means clustering is $O(nct)$, where t is the number of iterations. Consequently, the total complexity of SMP is $O(p(|k|+1)n^2+n|k|^2+nct)$. It is worth noting that SMP often needs fewer iterations.

4.3 Experiments

In this section some experiments are performed on both computer-generated graphs and real-world networks whose community structure is known in advance. Apart from c-rank (Jiang et al., 2012), we also compare SMP with four other classical methods: Multi-level Modularity Optimization (MMO) algorithm (Blondel et al., 2008), Leading Eigenvector (LE) algorithm (Newman, 2006a), Label Propagation Algorithm (LPA) (Raghavan et al., 2007), and Information Map algorithm (InfoMap) (Rosvall and Bergstrom, 2008). The obtained community structures are evaluated with known performance measures, *i.e.* accuracy and NMI (Normalized Mutual Information). As the benchmarks and the real-world data sets used in this chapter are with known community structure, accuracy and NMI measure the similarity between the planted partitions (ground truth) and the results of the algorithms. Both accuracy and NMI measure the proportion of the nodes that have been grouped correctly, and represent the consistence between the found community structure and the presumed one (Fan et al., 2007; Hu et al., 2008). The influence of different similarity and centrality measures in the application of SMP will be discussed in the first experiment. After that we will use the evidential semi-local centrality and signal similarity in the following tests based on the experimental results.

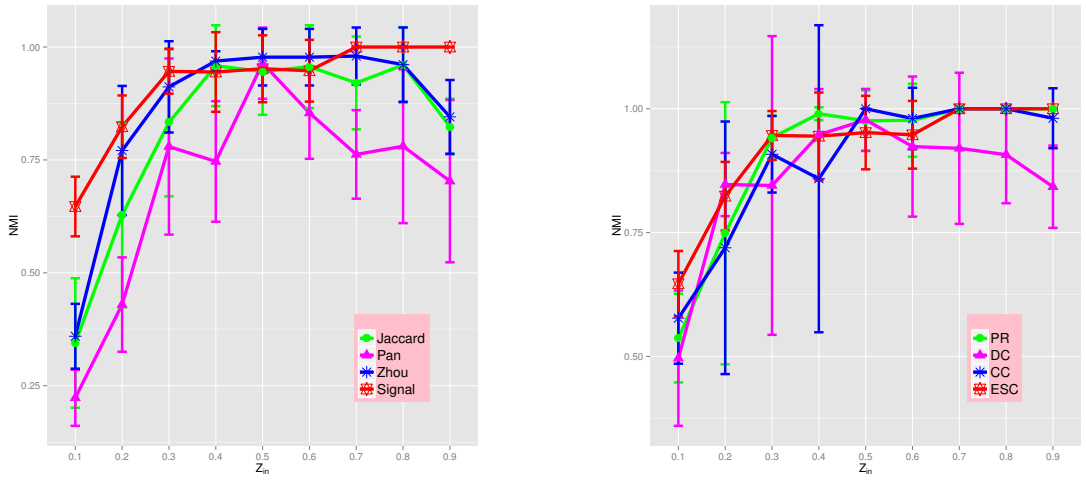
4.3.1 Computer-generated graphs

The algorithm is first compared by means of two classes of computer-generated artificial benchmark networks, namely, Girvan and Newman (Girvan and Newman, 2002) (GN) and Lancichinetti et al. (2008) benchmark (LFR) networks. For the former, each net-

work has $n = 128$ nodes in total and 32 nodes in each of the four divided communities. The average degree of each vertex is set to 16. For a given node, the average number of links to its fellows in the inner community, denoted by Z_{in}, is varied from 8 to 16. The average number of edges between communities, denoted by Z_{out}, is varied from 8 to 0. The larger Z_{in} is, the more apparent community structure the network has.

It is noteworthy that in the application of SMP algorithm, different similarity and centrality measures could be adopted instead of the signal similarity and evidential semi-local centrality suggested in this chapter. When using ESC for calculating the centrality, results by four different similarity metrics, *i.e.* signal similarity, the simple Jaccard index and the measures proposed by Pan et al. (2010) (denoted by Pan in the figure) and Zhou et al. (2009) (denoted by Zhou in the figure), are shown in Figure 4.1-a. As can be seen from the figure, the results by signal similarity are better than the other indices in terms of NMI values. Here we could conclude that global similarity measures like signal similarity are more applicable for SMP than local ones. Figure 4.1-b depicts the behavior of SMP with difference centrality measures but the same (signal) similarity index. It can be seen that ESC and PR are better among the four measures, *i.e.* ESC, PageRank (PR) (Brin and Page, 1998), Degree Centrality (DC), and Closeness Centrality (CC). Although there is no significant difference between ESC and PR, the performance of ESC is more stable than PR. This paper is not focusing on the comparison of different similarity and centrality measures, thus in the following experiment we only consider the signal similarity and evidential semi-local centrality.

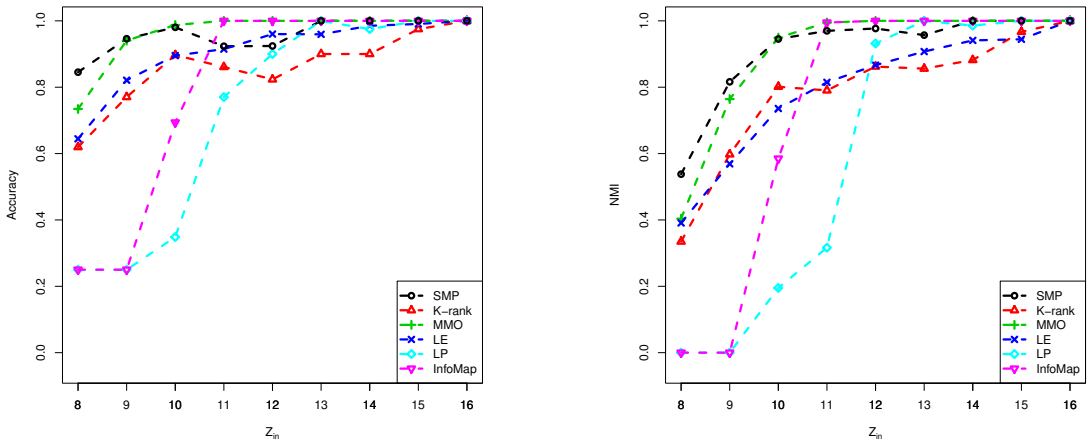
For each Z_{in}, the experiment is repeated 20 times and the mean values of the evaluating measures are reported. The average values of the indices by accuracy and NMI using SMP and the other five algorithms with different values of Z_{in} are displayed in Figure 4.2-a and Figure 4.2-b respectively. The results show that in terms of accuracy and NMI, all the methods perform well when Z_{in} is large. However, when Z_{in} is smaller than 10, they have different performances. LPA and InfoMap have the worst results as they could not work when $Z_{\text{in}} < 10$. SMP and MMO are best in general among all the methods. Although MMO is superior to SMP when $Z_{\text{in}} = 11$ and $Z_{\text{in}} = 12$, the superiority is not obvious. SMP is significantly better than MMO when Z_{in} is small (especially when $Z_{\text{in}} = 8$). Moreover, with the decreasing of Z_{in}, the performance of SMP does not drop so dramatically as the case in other methods. This demonstrates that using multiple members with various prototype weights is able to characterize the structure of clusters more precisely no matter whether the network has clear community structure or not, which in turn helps to produce a partition of the graph with good quality.



a. Different similarity measures

b. Different centrality measures

Figure 4.1: Comparison of similarity and centrality measures in the application of SMP algorithm. Average NMI values (plus and minus one standard deviation) for 20 repeated experiments, as a function of the average degree.

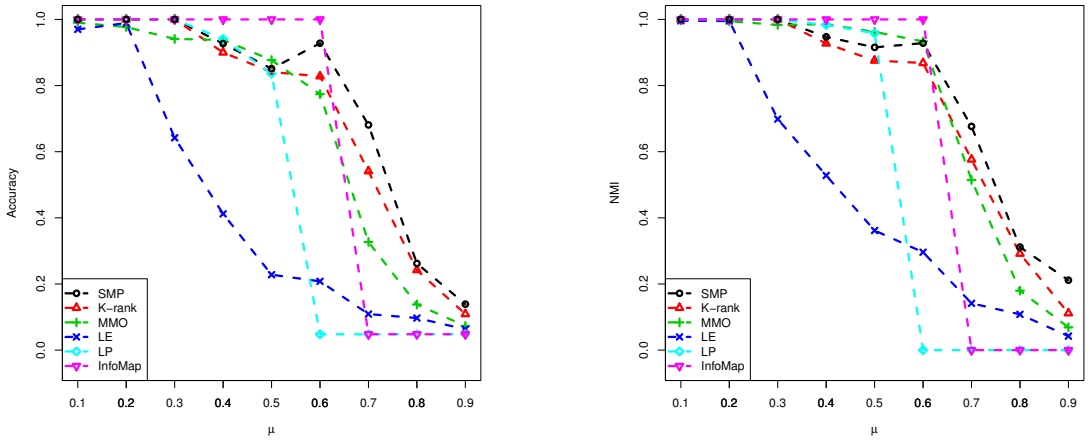


a. Accuracy

b. NMI

Figure 4.2: Comparison of SMP and other algorithms in Girvan and Newman's networks.

The LFR benchmark network (Lancichinetti et al., 2008) is an artificial network for community detection, which is claimed to process some basic statistical properties found in real networks, such as heterogeneous distributions of degree and community size. The results of different methods in three kinds of LFR networks with 1000, 2000 and 5000 nodes are displayed in Figures 4.3–4.5 respectively. The parameter μ illustrated in x-axis in the figures identifies whether the network has clear communities. When μ is small, the graph has well community structure. In such a case, almost all the methods perform well. But we can see that when μ is large, the results by SMP have relatively large values of NMI, and the performance of SMP and c-rank do not drop dramatically as the case in other methods. SMP slightly outperforms c-rank especially when μ is large, this could be attributed to the multi-prototype representation of communities. Overall, from the two types of benchmarks, SMP fits for the networks no matter whether they have clear community structures or not.

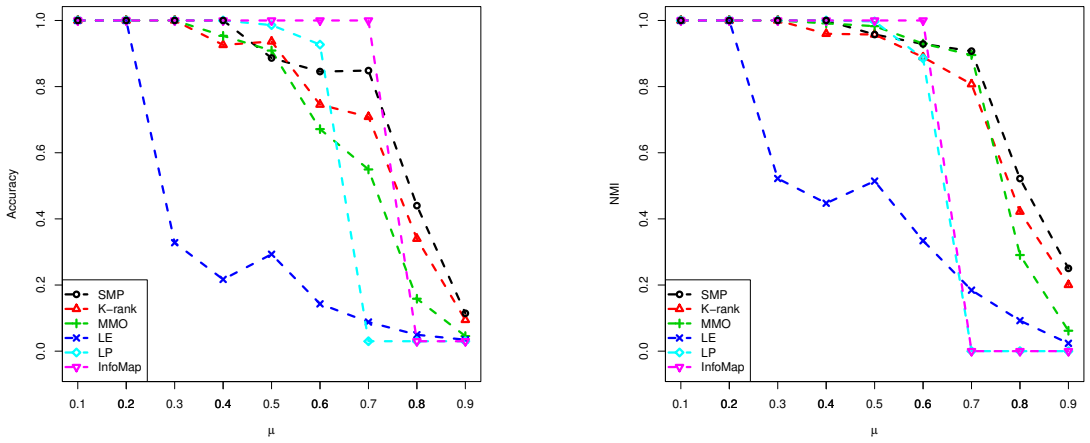


a. Accuracy b. NMI

Figure 4.3: Comparison of SMP and other algorithms in LFR networks. The number of nodes is $n = 1000$. The average degree is $|k| = 20$, and the pair for the exponents is $(\gamma, \beta) = (2, 1)$.

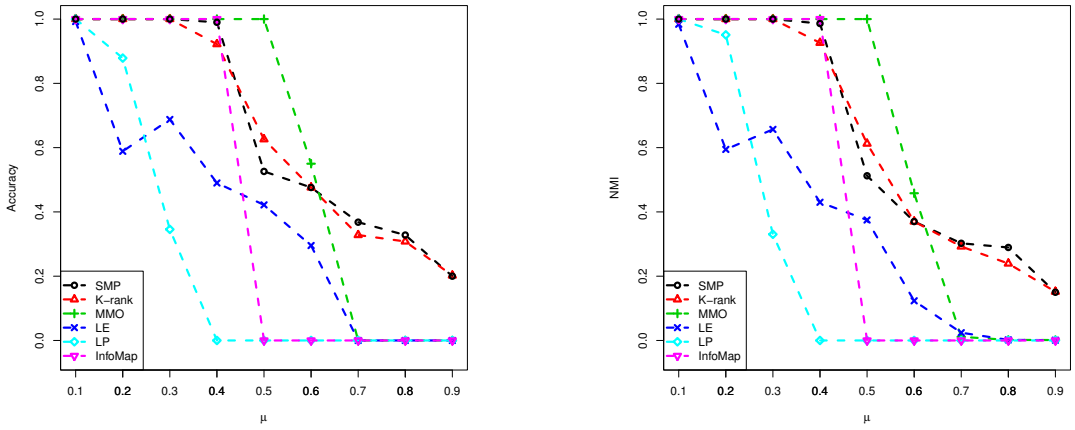
4.3.2 Real world networks

A. Zachary's Karate Club. To evaluate the effectiveness of the proposed method applied on real-world networks, we first test on the widely used benchmark in detecting



a. Accuracy b. NMI

Figure 4.4: Comparison of SMP and other algorithms in LFR networks. The number of nodes is $n = 2000$. The average degree is $|k| = 30$, and the pair for the exponents is $(\gamma, \beta) = (2, 1)$.



a. Accuracy b. NMI

Figure 4.5: Comparison of SMP and other algorithms in LFR networks. The number of nodes is $n = 5000$. The average degree is $|k| = 30$, and the pair for the exponents is $(\gamma, \beta) = (2, 1)$.

community structures, "Karate Club" (Zachary, 1977). This data set has been studied in Section 3.5.4.

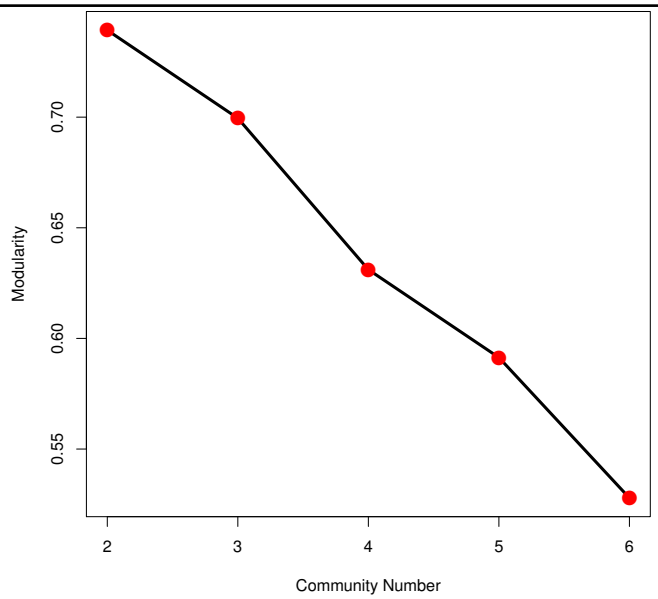


Figure 4.6: The modularity values on Karate Club network varying with community numbers.

The values of the modularity with different number of communities are displayed in Figure 4.6. The modularity function peaks when $c = 2$. This is in consistent with the fact that the network has two groups. The discovered communities are illustrated in Table 4.1. The table also shows the prototype weights in each of the found group. As we can see, node 1 makes the most contribution to community 1, while node 34 is most important to community 2. This confirms the center role of the two persons in their own communities. On the contrary, nodes 17 and 25 seem not very important in their group in terms of their prototype weights. We can see that in Figure 4.6-a, these two nodes locate in the marginal parts. Therefore, the proposed SMP detection approach enables us to have a better understanding of the graph structure with the help of prototype weights.

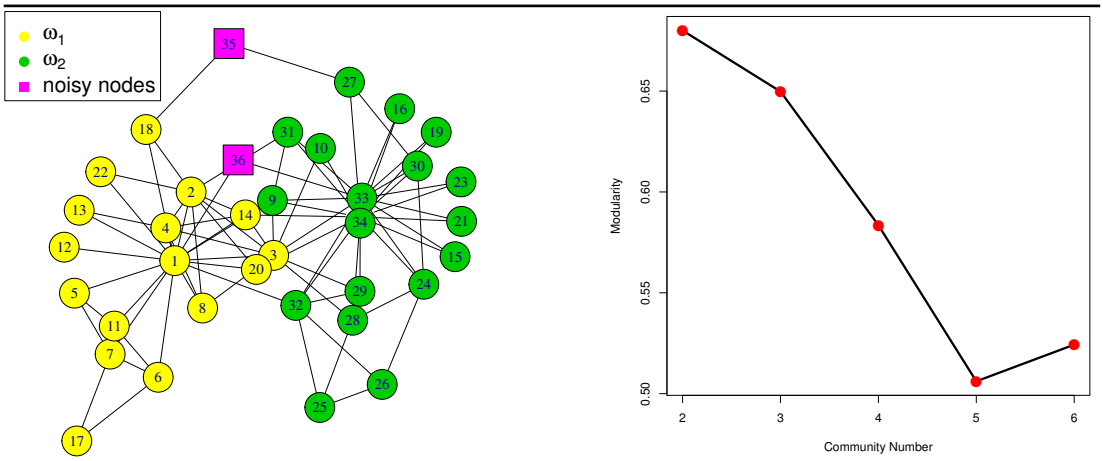
B. Karate Club network with some added noisy nodes. In this test two noisy nodes are added to the original Karate Club network (see Figure 4.7-a). The detail of this data set can be seen in Experiment B in Section 3.5.4. The modularity values varying with different community numbers are depicted in Figure 4.7-b and the detected results are displayed in Table 4.2.

Table 4.1: The results for Karate Club network. The notation u_{ij} denotes the fuzzy membership of node n_i to community j, and PW is short for prototype weights. The nodes are order by prototype weights in each community.

Community 1				Community 2			
Node ID	u_{i1}	u_{i2}	PW	Node ID	u_{i1}	u_{i2}	PW
1	0.5324	0.4676	0.1166	34	0.4607	0.5393	0.1025
2	0.5305	0.4695	0.0929	33	0.4582	0.5418	0.0940
4	0.5385	0.4615	0.0881	24	0.4469	0.5531	0.0738
3	0.5091	0.4909	0.0857	32	0.4798	0.5202	0.0698
8	0.5404	0.4596	0.0786	30	0.4424	0.5576	0.0679
14	0.5175	0.4825	0.0786	9	0.4882	0.5118	0.0595
6	0.5576	0.4424	0.0536	31	0.4772	0.5228	0.0595
7	0.5576	0.4424	0.0536	15	0.4464	0.5536	0.0532
18	0.5486	0.4514	0.0524	16	0.4464	0.5536	0.0532
20	0.5109	0.4891	0.0524	19	0.4464	0.5536	0.0532
22	0.5486	0.4514	0.0524	21	0.4464	0.5536	0.0532
5	0.5564	0.4436	0.0488	23	0.4464	0.5536	0.0532
11	0.5564	0.4436	0.0488	28	0.4707	0.5293	0.0474
13	0.5513	0.4487	0.0476	29	0.4788	0.5212	0.0408
12	0.5488	0.4512	0.0334	27	0.4420	0.5580	0.0392
17	0.5734	0.4266	0.0164	10	0.4802	0.5198	0.0307
				26	0.4582	0.5418	0.0268
				25	0.4671	0.5329	0.0223

From Table 4.2 we can see that the fuzzy membership values of nodes 35 and 36 are almost the same for both communities (approximatively equal to 0.5). These results could not reflect the difference between ignorance and uncertainty. As node 35 is only related to one outward node of each community, thus we are ignorant about which community it really belongs to, or we say node 35 is an outlier. On the contrary, node 36 connects with the key members (playing an important role in the community) in both communities. Thus there is uncertainty rather than ignorance about which community node 36 is inside. In this network, node 36 is a "good" member for both communities, whereas node 35 is a "poor" member. As mentioned before, the inability to distinguish the outliers from the uncertain nodes with equal memberships is caused by the relative similarity used in fuzzy memberships. In SMP, the prototype weights could be utilized to solve this problem and to detect the outliers. As shown in Table 4.2, the prototype weight of node 35 is the least in the community, but node 36 contributes much more than node 35. Therefore, node 35 has no contribution to both communities (the prototype weight of node 35 for community 1 is 0.0052, and 0 for community 2), and it could be recognized as an outlier. This example further demonstrates the fact that prototype weights indeed enable us to gain a better understanding of the graph

structure, especially for detecting outliers in the network.



a. Karate Club network with two added nodes

b. Modularity function

Figure 4.7: The Karate Club network with added nodes and the modularity values varying with community numbers.

We also test our method on four other real-world graphs: American football network, Dolphins network, Lesmis network and Political books network[4]. The values of the two indices, accuracy and NMI, applied to evaluate the performance of different methods are listed in Table 4.3 and Table 4.4 respectively[5]. It can been seen from the tables, SMP application results in a community structure with highest accuracy level in most cases. In terms of the performance measure NMI, SMP also outperforms the other algorithms. It should be noted that some methods provide partitions with high accuracy but low NMI. This may be caused by the fact that they cluster the nodes into too many small communities. The partition rules of both c-rank and SMP are based on node similarity. These two approaches are better than the others in general, and the effectiveness could be attributed to the high performance of vertex similarities. But the reason that SMP works better than c-rank in these real-world networks is largely because of the application of multiple prototype representation of communities.

From the above extensive experimental results, we can summarize the compelling properties of SMP as follows:

(1) In the partition process, SMP uses multiple prototypes to represent the communities. This is a useful extension of the existing community detection methods

[4]These data sets can be found in http://networkdata.ics.uci.edu/index.php

[5]All these real-world graphs are with known community structure, thus the accuracy and NMI are calculated based on the ground truth and the partition got by different algorithms.

Table 4.2: The results for Karate Club network with added nodes. The notation u_{ij} denotes the fuzzy membership of node n_i to community j, and PW is short for prototype weights. The nodes are order by PW in each community.

Community 1				Community 2			
Node ID	u_{i1}	u_{i2}	PW	Node ID	u_{i1}	u_{i2}	PW
1	0.5278	0.4722	0.1111	34	0.4656	0.5344	0.1028
2	0.5271	0.4729	0.0888	33	0.4651	0.5349	0.0944
4	0.5344	0.4656	0.0836	24	0.4534	0.5466	0.0737
3	0.5084	0.4916	0.0814	32	0.4824	0.5176	0.0696
8	0.5360	0.4640	0.0747	30	0.4506	0.5494	0.0680
14	0.5158	0.4842	0.0747	9	0.4899	0.5101	0.0598
18	0.5399	0.4601	0.0528	31	0.4801	0.5199	0.0598
6	0.5511	0.4489	0.0520	15	0.4533	0.5467	0.0534
7	0.5511	0.4489	0.0520	16	0.4533	0.5467	0.0534
20	0.5099	0.4901	0.0506	19	0.4533	0.5467	0.0534
22	0.5427	0.4573	0.0506	21	0.4533	0.5467	0.0534
5	0.5498	0.4502	0.0475	23	0.4533	0.5467	0.0534
11	0.5498	0.4502	0.0475	28	0.4740	0.5260	0.0471
13	0.5454	0.4546	0.0462	29	0.4813	0.5187	0.0404
12	0.5427	0.4573	0.0330	27	0.4539	0.5461	0.0395
36	0.5016	0.4984	0.0330	10	0.4826	0.5174	0.0309
17	0.5658	0.4342	0.0154	26	0.4628	0.5372	0.0258
35	0.5020	0.4980	0.0052	25	0.4705	0.5295	0.0212

where only one prototype is allowed, especially when the analyzed graph has some complex community structures.

(2) The prototype weights, as a by-product of the detection results, provide us with some valuable information about the community structure from another point of view, and enable us to gain a better understanding of the analyzed graph.

(3) SMP works well even for the graphs without clear community structures. It could avoid the problem of inability to distinguish the outliers from uncertain data for fuzzy membership.

(4) Last but not the least, the experiments on both synthetic and real-world graph data sets demonstrate that the proposed approach is a competitive candidate for community detection tasks compared with other five existing methods.

Table 4.3: Comparison of SMP and other algorithms by accuracy in real-world networks.

	Karate	Football	Dolphins	Lesmis	Books
SMP	**1.0000**	**0.9345**	**1.0000**	0.7792	**0.8667**
c-rank	**1.0000**	0.9320	**1.0000**	0.8052	0.8537
MMO	**1.0000**	0.8000	0.9516	0.7922	0.7276
LE	**1.0000**	0.6261	0.9677	0.7273	0.8476
LPA	0.9706	0.9043	**1.0000**	0.7273	0.8476
InfoMap	**1.0000**	0.9043	0.9839	**0.8701**	0.7854

Table 4.4: Comparison of SMP and other algorithms by NMI in real-world networks.

	Karate	Football	Dolphins	Lesmis	Books
SMP	**1.0000**	**0.9235**	**1.0000**	0.7444	**0.5938**
c-rank	**1.0000**	0.9211	**1.0000**	0.7818	0.5741
MMO	0.6873	0.8550	0.4617	0.7551	0.5121
LE	0.6552	0.6952	0.5094	0.7182	0.5201
LPA	0.8255	0.9095	0.8230	0.7381	0.5485
InfoMap	0.8255	0.8937	0.5629	**0.8198**	0.4935

4.4 Conclusion

In this chapter, a new type of similarity-based community detection algorithm called SMP is proposed. SMP could find not only communities of each node but also weighted representative members of each group. In the task of network structure analysis, information on both community labels and internal structure of each of the detected communities are important. One distinctive characteristic of the proposed method is that each community is presented by multiple prototypes, rather than by single one object. The experiments on synthetic networks show the effectiveness of the proposed method and the tests on real-world networks have further pointed out our method preforms better than the existing ones. The results show that the way of using prototype weights to represent a cluster enables SMP to capture the various types of community structures more precisely and completely hence improves the quality of the detected communities. Moreover, more detail information on the discovered clusters may be obtained with the help of prototype weights.

In real applications, the signal similarity measure and ESC centrality utilized in the work could be replaced by any other index. For instance, if we want to apply the method to directed networks, the similarity and centrality measures for directed networks could be adopted. Therefore, we intend to study on the comparison of difference measures and on the application into directed networks in our future research work. Meanwhile,

not only centrality but also more other factors should be considered for determining the prototype weights. Hence the way to optimize the prototype weights using the available information as much as possible will also be included in our further study.

As we see, SMP could only be used to produce hard and fuzzy partitions. This is not enough to describe the uncertain or overlapping graph structure. Therefore, in the next chapter, we will extend SMP in the framework of belief functions to obtain credal partitions.

5
Evidential c-medoids clustering with multiple weighted prototypes

5.1 Overview

In this chapter, we will extend SMP algorithm in the framework of belief functions to capture the uncertain class structures in the data set more effectively. The prototype weights are incorporated into the concept of medoid-based clustering, which is of great value for partitioning proximity data sets. A new prototype-based clustering method named Evidential C-Medoids (ECMdd) is proposed (Zhou et al., 2015b). ECMdd can be seen an extension of Fuzzy C-Medoids (FCMdd) on the theoretical framework of belief functions. In the application of FCMdd and original ECMdd, a single medoid (prototype), which is supposed to belong to the object set, is utilized to represent one class. To make it clearly, this kind of ECMdd using a single medoid is denoted by sECMdd. As we mentioned before, in real clustering applications, using only one pattern to capture or interpret a class may not adequately model different types of group structure and hence limits the clustering performance. In order to address this problem, a variation of ECMdd using multiple weighted medoids, denoted by wECMdd, is presented. Unlike sECMdd, in wECMdd objects in each cluster carry various weights to represent their degree of representativeness for that class. This mechanism enables each class to be represented by more than one object. Experimental results in synthetic and real data sets clearly demonstrate the superiority of sECMdd and wECMdd. Moreover, the clustering results by wECMdd could provide richer information for the inner structure of the detected clusters with the help of prototype weights.

5.2 Hard and fuzzy c-medoids clustering

The hard C-Medoids (CMdd) clustering is a variant of the traditional c-means algorithm. Both of them produce a crisp partition of the analyzed data set. Let $\boldsymbol{X} = \{x_i \mid i = 1, 2, \cdots, n\}$ be the set of n objects and $d(x_i, x_j) \triangleq d_{ij}$ denote the dissimilarity between objects x_i and x_j. Each object may or may not be represented by a feature vector. Let $\boldsymbol{V} = \{v_1, v_2, \cdots, v_c\}$, $v_i \in \boldsymbol{X}$ represent a subset of $\boldsymbol{X}$. The objective

function of CMdd is similar to that in CM:

$$J_{\text{CMdd}} = \sum_{j=1}^{c}\sum_{i=1}^{n} u_{ij} d(x_i, v_j), \tag{5.1}$$

where c is the number of clusters. As CMdd is based on crisp partitions, u_{ij} is either 0 or 1 depending whether x_i is in cluster ω_j. The notation v_j is the prototype of class ω_j, and it is supposed to be one of the objects in the data set. Due to the fact that exhaustive search of medoids is an NP hard problem, Kaufman and Rousseeuw (2009) proposed one approximate search algorithm called PAM, where the c medoids are found efficiently. After the selection of the prototypes, object x_i is assigned the closest class ω_f, the medoid of which is most similar to this pattern, *i.e.*

$$x_i \in \omega_f, \quad \text{with} \quad f = \arg\min_{l=1,2,\cdots,c} d(x_i, v_l). \tag{5.2}$$

Fuzzy C-Medoids (FCMdd) is a variation of CMdd designed for relational data (Krishnapuram et al., 2001). The objective function of FCMdd is given as

$$J_{\text{FCMdd}} = \sum_{i=1}^{n}\sum_{j=1}^{c} u_{ij}^{\beta} d(x_i, v_j) \tag{5.3}$$

subject to

$$\sum_{j=1}^{c} u_{ij} = 1, i = 1, 2, \cdots, n, \tag{5.4}$$

and

$$u_{ij} \geq 0, i = 1, 2, \cdots, n, \quad j = 1, 2, \cdots, c. \tag{5.5}$$

In fact, the objective function of FCMdd is similar to that of FCM. The main difference lies in that the prototype of a class in FCMdd is defined as the medoid, *i.e.* one of the object in the original data set, instead of the centroid (the average point in a continues space) for FCM. The object assignment and prototype selection are preformed by the following alternating update steps:

(1) Assignment update:

$$u_{ij} = \frac{d_{ij}^{-1/(\beta-1)}}{\sum_{k=1}^{c} d_{ik}^{-1/(\beta-1)}}. \tag{5.6}$$

(2) Prototype update: the new prototype of cluster ω_j is set to be $v_j = x_{l^*}$ with

$$x_{l^*} = \arg\min_{\{v_j : v_j = x_l (\in X)\}} \sum_{i=1}^{n} u_{ij}^{\beta} d(x_i, v_j). \tag{5.7}$$

In a recent work of Mei and Chen (2011), a generalized medoids-based Fuzzy clustering with Multiple Medoids (FMMdd) has been proposed. For a data set $\boldsymbol{X}$ given the dissimilarity matrix $\{d_{ij}\}_{n\times n}$, where d_{ij} records the dissimilarity between each two objects x_i and x_j. The objective of FMMdd is to minimize the following criterion:

$$J_{\text{FMMdd}} = \sum_{k=1}^{c}\sum_{i=1}^{n}\sum_{j=1}^{n} u_{ik}^{\beta} v_{kj}^{\psi} d_{ij} \tag{5.8}$$

subject to

$$\sum_{k=1}^{c} u_{ik} = 1, \forall i = 1, 2, \cdots n; \quad u_{ik} \geq 0, \forall i \text{ and } k \tag{5.9}$$

and

$$\sum_{j=1}^{n} v_{kj} = 1, \forall k = 1, 2, \cdots, c; \quad v_{kj} \geq 0, \quad \forall k \text{ and } j, \tag{5.10}$$

where u_{ik} denotes the fuzzy membership of x_i for cluster ω_k, and v_{kj} denotes the prototype weights of x_j for cluster ω_k. The constrained minimization problem of finding the optimal fuzzy partition could be solved by the use of Lagrange multipliers and the update equations of u_{ik} and v_{kj} are derived as below:

$$u_{ik} = \frac{\left(\sum\limits_{j=1}^{n} v_{kj}^{\psi} d_{ij}\right)^{-1/(\beta-1)}}{\sum\limits_{f=k}^{c}\left(\sum\limits_{j=1}^{n} v_{fj}^{\psi} d_{ij}\right)^{-1/(\beta-1)}} \tag{5.11}$$

and

$$v_{kj} = \frac{\left(\sum\limits_{i=1}^{n} u_{ik}^{\beta} d_{ij}\right)^{-1/(\psi-1)}}{\sum\limits_{h=1}^{n}\left(\sum\limits_{i=1}^{n} u_{ik}^{\beta} d_{ih}\right)^{-1/(\psi-1)}}. \tag{5.12}$$

The FMMdd algorithm starts with a non-negative initialization, then the membership values and prototype weights are iteratively updated with Eqs. (5.11) and (5.12) until convergence.

5.3 ECMdd clustering with multiple prototypes

In order to introduce ECMdd clustering with multiple prototypes, we start with the introduction of evidential c-medoids clustering algorithm using a single medoid, sECMdd, which takes advantages of both medoid-based clustering and credal partitions. Based on sECMdd, the method where multiple prototypes are adopted will be presented then.

5.3.1 sECMdd with a single medoid

The principle of sECMdd is similar to that of fuzzy c-medoids. Like all the prototype-based clustering methods, for sECMdd, an objective function should first be found to provide an immediate measure of the quality of the partitions. Hence our goal can be characterized as the optimization of the objective function to get the best credal partition.

The objective function

As before, let $\boldsymbol{X} = \{x_i \mid i = 1, 2, \cdots, n\}$ be the set of n objects, and $d_{ij} = d(x_i, x_j)$ records the dissimilarity between objects x_i and x_j. The pairwise dissimilarity is the only information required for the analyzed data set. The objective function of sECMdd is similar to that in ECM:

$$J_{\text{sECMdd}}(\boldsymbol{M}, \boldsymbol{V}) = \sum_{i=1}^{n} \sum_{A_j \subseteq \Omega, A_j \neq \emptyset} |A_j|^{\alpha} m_{ij}^{\beta} \overline{d}_{ij} + \sum_{i=1}^{n} \delta^2 m_{i\emptyset}^{\beta}, \tag{5.13}$$

constrained on

$$\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} + m_{i\emptyset} = 1, \tag{5.14}$$

where $m_{ij} \triangleq m_i(A_j)$ is the bba of x_i given to the nonempty set A_j, $m_{i\emptyset} \triangleq m_i(\emptyset)$ is the bba of x_i assigned to the empty set, and $\overline{d}_{ij} \triangleq \overline{d}(x_i, A_j)$ is the dissimilarity between x_i and focal set A_j. Parameters α, β, δ are adjustable with the same meanings as those in ECM. Note that J_{sECMdd} depends on the credal partition $\boldsymbol{M}$ and the set $\boldsymbol{V}$ of all prototypes.

Let v_k^{Ω} be the prototype of specific cluster (whose focal element is a singleton) $A_j = \{\omega_k\}$ $(k = 1, 2, \cdots, c)$ and assume that it must be one of the objects in $\boldsymbol{X}$. The dissimilarity between object x_i and cluster (focal set) A_j can be defined as follows. If $|A_j| = 1$, *i.e.* A_j is associated with one of the singleton clusters in Ω (suppose to be ω_k with prototype v_k^{Ω}, *i.e.* $A_j = \{\omega_k\}$), then the dissimilarity between x_i and A_j is defined by

$$\overline{d}_{ij} = \overline{d}(x_i, A_j) = d(x_i, v_k^{\Omega}). \tag{5.15}$$

When $|A_j| > 1$, it represents an imprecise (meta) cluster. If object x_i is to be partitioned into a meta cluster, two conditions should be satisfied (Zhou et al., 2015c). One condition is the dissimilarity values between x_i and the included singleton classes' prototypes are small. The other condition is the object should be close to the prototypes of all these specific clusters. The former measures the degree of uncertainty, while the latter is to avoid the pitfall of partitioning two data objects irrelevant to any included specific clusters into the corresponding imprecise classes. Therefore, the medoid (prototype) of an imprecise class A_j could be set to be one of the objects locating with similar

dissimilarities to all the prototypes of the specific classes $\omega_k \in A_j$ included in A_j. The variance of the dissimilarities of object x_i to the medoids of all the involved specific classes could be taken into account to express the degree of uncertainty. The smaller the variance is, the higher uncertainty we have for object x_i. Meanwhile the medoid should be close to all the prototypes of the specific classes. This is to distinguish the outliers, which may have similar dissimilarities to the prototypes of some specific classes, but obviously not a good choice for representing the associated imprecise classes. Let $v_j^{2^\Omega}$ denote the medoid of class A_j[6]. Based on the above analysis, the medoid of A_j should set to $v_j^{2^\Omega} = x_p$ with

$$p = \arg\min_{i:x_i \in \boldsymbol{X}} \Big\{ f\left(\{d(x_i, v_k^\Omega); \omega_k \in A_j\}\right) + \eta \frac{1}{|A_j|} \sum_{\omega_k \in A_j} d(x_i, v_k^\Omega) \Big\}, \tag{5.16}$$

where ω_k is the element of A_j, v_k^Ω is its corresponding prototype and f denotes the function describing the variance among the related dissimilarity values. The variance function could be used directly:

$$\mathrm{Var}_{ij} = \frac{1}{|A_j|} \sum_{\omega_k \in A_j} \left[d(x_i, v_k^\Omega) - \frac{1}{|A_j|} \sum_{\omega_k \in A_j} d(x_i, v_k^\Omega) \right]^2. \tag{5.17}$$

In this chapter, we use the following function to describe the variance ρ_{ij} of the dissimilarities between object x_i and the medoids of the involved specific classes in A_j

$$\rho_{ij} = \frac{1}{\mathrm{choose}(|A_j|, 2)} \sum_{\omega_x, \omega_y \in A_j} \sqrt{\left(d(x_i, v_x^\Omega) - d(x_i, v_y^\Omega)\right)^2}, \tag{5.18}$$

where $\mathrm{choose}(a, b)$ is the number of combinations of the given a elements taken b at a time. Then the dissimilarity between objects x_i and class A_j can be defined as

$$\overline{d}_{ij} = \frac{d(x_i, v_j^{2^\Omega}) + \gamma \frac{1}{|A_j|} \sum_{\omega_k \in A_j} d(x_i, v_k^\Omega)}{1 + \gamma}. \tag{5.19}$$

As we can see from the above equation, the dissimilarity between object x_i and meta class A_j is the weighted average of dissimilarities of x_i to the all involved singleton cluster medoids and to the prototype of the imprecise class A_j with a tuning factor γ. If A_j is a specific class with $A_j = \{\omega_k\}$ ($|A_j| = 1$), the dissimilarity between x_j and A_j degrades to the dissimilarity between x_i and v_k^Ω as defined in Eq. (5.15), *i.e.* $v_j^{2^\Omega} = v_k^\Omega$.

[6]The notation v_k^Ω denotes the prototype of specific class ω_k, indicating it is in the framework of Ω. Similarly, $v_j^{2^\Omega}$ is defined on the power set 2^Ω, representing the prototype of the focal set $A_j \in 2^\Omega$. In fact $\boldsymbol{V}$ is the set of all the prototypes, *i.e.* $\boldsymbol{V} = \{v_j^{2^\Omega} : j = 1, 2, \cdots, 2^c - 1\}$. It is easy to see $\{v_k^\Omega : k = 1, 2, \cdots, c\} \subseteq \boldsymbol{V} \subseteq \boldsymbol{X}$.

And if $|A_j| > 1$, its medoid is decided by Eq. (5.16).

Remark 4. sECMdd is similar to Median Evidential C-Means (MECM) (Zhou et al., 2015c) algorithm in Chapter 3. MECM is in the framework of median clustering, while sECMdd consists with FCMdd in principle. Another difference of sECMdd and MECM is the way of calculating the dissimilarities between objects and imprecise classes. Although both MECM and sECMdd consider the dissimilarities of objects to the prototypes for specific clusters, the strategy adopted by sECMdd is more simple and intuitive, hence makes sECMdd run faster in real time. Moreover, there is no representative medoid for imprecise classes in MECM. In addition, sECMdd is designed as the base of the multiple prototype evidential clustering presented later.

The optimization

To minimize J_{sECMdd}, an optimization scheme via an Expectation-Maximization (EM) algorithm can be designed, and the alternate update steps are as follows:
Step 1. Credal partition ($\boldsymbol{M}$) update.

The bbas of objects' class membership for any subset $A_j \subseteq \Omega$ and the empty set $\emptyset$ representing the outliers are updated identically to ECM (Masson and Denoeux, 2008):

(1) $\forall A_j \subseteq \Omega, A_j \neq \emptyset$,

$$m_{ij} = \frac{|A_j|^{-\alpha/(\beta-1)} \overline{d}_{ij}^{-1/(\beta-1)}}{\sum\limits_{A_k \neq \emptyset} |A_k|^{-\alpha/(\beta-1)} d_{ik}^{-1/(\beta-1)} + \delta^{-1/(\beta-1)}} \tag{5.20}$$

(2) If $A_j = \emptyset$,

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij} \tag{5.21}$$

Step 2. Prototype ($\boldsymbol{V}$) update.

The prototype v_i^{Ω} of a specific (singleton) cluster ω_i $(i = 1, 2, \cdots, c)$ can be updated first and then the prototypes of imprecise (meta) classes could be determined by Eq. (5.16). For singleton clusters ω_k $(k = 1, 2, \cdots, c)$, the corresponding new prototype v_k^{Ω} $(k = 1, 2, \cdots, c)$ could be set to $x_l \in \boldsymbol{X}$ such that

$$x_l = \arg\min_{v_k'} \left\{ \sum_{i=1}^{n} \sum_{A_j = \{\omega_k\}} m_{ij}^{\beta} \overline{d}_{ij}(v_k') : v_k' \in X \right\}. \tag{5.22}$$

The dissimilarity between object x_i and cluster A_j, $\overline{d}_{ij}$, is a function of v_k', which is the potential prototype of class ω_k.

The bbas of the objects' class assignment are updated identically to ECM (Masson and Denoeux, 2008), but it is worth noting that $\overline{d}_{ij}$ has different meanings as that in

ECM although in both cases it measures the dissimilarity between object x_i and class A_j. In ECM $\overline{d}_{ij}$ is the distance between object i and the centroid point of A_j, while in sECMdd, it is the dissimilarity between x_i and the most "possible" medoid. For the prototype updating process the fact that the prototypes are assumed to be one of the data objects is taken into consideration. Therefore, when the credal partition matrix $\boldsymbol{M}$ is fixed, the new prototype of each cluster can be obtained in a simpler manner than in the case of ECM application. The sECMdd algorithm can be summarized as Algorithm 4.

Algorithm 4 : sECMdd algorithm

Input: Dissimilarity matrix $[d_{ij} \triangleq d(x_i, x_j)]_{n\times n}$ for the n objects $\{x_1, x_2, \cdots, x_n\}$.
Parameters:
c: number clusters $1 < c < n$
α: weighing exponent for cardinality
$\beta > 1$: weighting exponent
$\delta > 0$: dissimilarity between any object to the empty set
$\eta > 0$: to distinguish the outliers from the possible medoids
$\gamma \in [0, 1]$: to balance of the contribution for imprecise classes
Initialization:
Choose randomly c initial prototypes from the object set
repeat
(1). $t \leftarrow t + 1$
(2). Compute $\boldsymbol{M}_t$ using Eq. (5.20), Eq. (5.21) and $\boldsymbol{V}_{t-1}$
(3). Compute the new prototype set $\boldsymbol{V}_t$ using Eqs. (5.22) and (5.16)
until the prototypes remain unchanged.
Output: The optimal credal partition.

Remark 5. The assignment update process will not increase J_{sECMdd} since the new mass matrix is determined by differentiating of the respective Lagrangian of the cost function with respect to $\boldsymbol{M}$. Also J_{sECMdd} will not increase through the medoid-searching scheme for prototypes of specific classes. If the prototypes of specific classes are fixed, the medoids of imprecise classes determined by Eq. (5.16) are likely to locate near to the "centroid" of all the prototypes of the included specific classes. If the objects are in Euclidean space, the medoids of imprecise classes are near to the centroids found in ECM. Thus it will not increase the value of the objective function also. Moreover, the bba $\boldsymbol{M}$ is a function of the prototypes $\boldsymbol{V}$ and for given $\boldsymbol{V}$ the assignment $\boldsymbol{M}$ is unique. Because sECMdd assumes that the prototypes are original object data in $\boldsymbol{X}$, so there is a finite number of different prototype vectors $\boldsymbol{V}$ and so is the number of corresponding credal partitions $\boldsymbol{M}$. Consequently we can conclude that the sECMdd algorithm converges in a finite number of steps.

5.3.2 ECMdd with multiple weighted medoids

This section presents evidential c-medoids algorithm using multiple weighted medoids. The approach to compute the relative weights of the medoids are based on both the computation of the membership degree of objects belonging to specific classes and the computation of the dissimilarities between objects.

The objective function

The objective function of wECMdd, J_{wECMdd}, has the same form as that in sECMdd (see Eq. (5.13)). In wECMdd, we use multiple weighted medoids to represent each specific class instead of a single medoid. Thus the method to calculate $\overline{d}_{ij}$ in the objective function is different from sECMdd. Let $\boldsymbol{V}^{\Omega} = \{v_{ki}^{\Omega}\}_{c\times n}$ be the weight matrix for specific classes, where v_{ki}^{Ω} describes the weight of object i for the k^{th} specific class. Then, the dissimilarity between object x_i and cluster $A_j = \{\omega_k\}$ could be calculated by

$$\overline{d}(x_i, A_j) \triangleq \overline{d}_{ij} = \sum_{l=1}^{n} \left(v_{kl}^{\Omega}\right)^{\psi} d_{il}, \tag{5.23}$$

with

$$\sum_{l=1}^{n} v_{kl}^{\Omega} = 1, \forall k = 1, 2, \cdots, c. \tag{5.24}$$

Parameter ψ controls the smoothness of the distribution of prototype weights. The weights of imprecise class A_j ($|A_j| > 1$) could be derived according to the involved specific classes. If object x_i has similar weights for specific classes ω_m and ω_n, it is most probable that x_i lies in the overlapped area of two classes. Thus the variance of the weights of object x_i for all the included specific classes of A_j, Var_{ji}, could be used to express the weights of x_i for A_j (denoted by $v_{ji}^{2^{\Omega}}$, and $\boldsymbol{V}$ is used to denote the corresponding weight matrix[7]). The smaller Var_{ji} is, the higher $v_{ji}^{2^{\Omega}}$ is. However, we should pay attention to the outliers. They may hold similar small weights for each specific class, but have no contribution to the imprecise classes at all. The minimum of x_i's weights for all the associated specific classes could be taken into consideration to distinguish the outliers. If the minimal weight is too small, we should assign a small weight value for that object. Based on the discussion, the weights of object x_i for class A_j ($A_j \subseteq \Omega$) could be calculated as

$$v_{ji}^{2^{\Omega}} = \frac{f_1\left(\text{Var}\left(\{v_{ki}^{\Omega}; \omega_k \in A_j\}\right)\right) * f_2\left(\min\left(\{v_{ki}^{\Omega}; \omega_k \in A_j\}\right)\right)}{\sum_{l} f_1\left(\text{Var}\left(\{v_{kl}^{\Omega}; \omega_k \in A_j\}\right)\right) * f_2\left(\min\left(\{v_{kl}^{\Omega}; \omega_k \in A_j\}\right)\right)}, \tag{5.25}$$

[7] In sECMdd, $\boldsymbol{V}$ denotes the set of prototypes of all the classes. Here $\boldsymbol{V}$ represents the weights of prototypes. We use the same notation to show the similar role of $\boldsymbol{V}$ in sECMdd and wECMdd. In fact sECMdd can be regarded as a special case of wECMdd, where the weight values are restricted to be either 0 or 1.

where f_1 is a monotone decreasing function while f_2 is an increasing function. The two functions should be determined according to the application under concern. Based on our experiments, we suggest adopting the simple directly and inversely proportion functions, *i.e.*

$$v_{ji}^{2^\Omega} = \frac{[\min\left(\{v_{ki}^\Omega; \omega_k \in A_j\}\right)]^\xi / \mathrm{Var}\left(\{v_{ki}^\Omega; \omega_k \in A_j\}\right)}{\sum_l [\min\left(\{v_{kl}^\Omega; \omega_k \in A_j\}\right)]^\xi / \mathrm{Var}\left(\{v_{kl}^\Omega; \omega_k \in A_j\}\right)}. \tag{5.26}$$

Parameter ξ is used to balance the contribution of f_1 and f_2. It is remarkable that when $A_j = \{\omega_k\}$, that is to say $|A_j| = 1$, $v_{ji}^{2^\Omega} = v_{ki}^\Omega$. Therefore, the dissimilarity between object x_i and cluster A_j (including both specific and imprecise classes) could be given by

$$\overline{d}_{ij} = \sum_{l=1}^{n} \left(v_{jl}^{2^\Omega}\right)^\psi d_{il}, \quad A_j \subseteq \Omega, A_j \neq \emptyset. \tag{5.27}$$

Optimization

The problem of finding optimal cluster assignments of objects and representatives of classes is now formulated as a constrained optimization problem, *i.e.* to find optimal values of $\boldsymbol{M}$ and $\boldsymbol{V}$ subject to a set of constrains. As before, the method of Lagrange multipliers could be utilized to derive the solutions. The Lagrangian function is constructed as

$$L_{\mathrm{wECMdd}} = J_{\mathrm{wECMdd}} - \sum_{i=1}^{n} \lambda_i \left(\sum_{A_j \subseteq \Omega, A_j \neq \emptyset} m_{ij} - 1 \right) - \sum_{k=1}^{c} \beta_k \left(\sum_{i=1}^{n} v_{ki}^\Omega - 1 \right), \tag{5.28}$$

where λ_i and β_k are Lagrange multipliers. By calculating the first order partial derivatives of L_{wECMdd} with respect to m_{ij}, v_{ki}^Ω, λ_i and β_k and letting them to be 0, the update equations of m_{ij} and v_{ki}^Ω could be derived. It is easy to see that the update equations for m_{ij} are the same as Eqs. (5.20) and (5.21) in the application of sECMdd, except that in this case $\overline{d}_{ij}$ should be calculated by Eq. (5.27). The update strategy for the prototype weights v_{ki}^Ω is difficult to get since it is a non-linear optimization problem. Some specifical techniques may be adopted to solve this problem. Here we use a simple approximation scheme to update v_{ki}^Ω.

Suppose the class assignment $\boldsymbol{M}$ is fixed and assume that the prototype weights for imprecise class A_j ($A_j \subseteq \Omega, |A_j| > 1$), $v_{ji}^{2^\Omega}$, are dependent of the weights for specific classes (v_{ki}^Ω). Then the first order necessary condition with respect to v_{ki}^Ω is only related

to $\overline{d}_{ij}$ with $A_j = \{\omega_k\}$. The update equations of v_{ki}^{Ω} could then derived as

$$v_{ki}^{\Omega} = \frac{\left(\sum_{l=1}^{n} m_{lj}^{\beta} d_{li}\right)^{-1/(\psi-1)}}{\sum_{h=1}^{n}\left(\sum_{l=1}^{n} m_{lj}^{\beta} d_{lh}\right)^{-1/(\psi-1)}} \quad k = 1, 2, \cdots, c,\ A_j = \{\omega_k\}. \tag{5.29}$$

After obtaining the weights for specific classes, the weights for imprecise classes could be got by Eq. (5.26) and the dissimilarities between objects and classes could then calculated by Eq. (5.27). The update of cluster assignment $\boldsymbol{M}$ and prototype weight matrix $\boldsymbol{V}$ should be repeated until convergence. The wECMdd algorithm is summarised in Algorithm 5.

Algorithm 5 : wECMdd algorithm

Input: Dissimilarity matrix $[d(x_i, x_j)]_{n\times n}$ for the n objects $\{x_1, x_2, \cdots, x_n\}$.
Parameters:
c: number clusters $1 < c < n$
α: weighing exponent for cardinality
$\beta > 1$: weighting exponent
$\delta > 0$: dissimilarity between any object to the empty set
$\xi > 0$: balancing the weights of imprecise classes
ψ: controlling the smoothness of the distribution of prototype weigths
Initialization:
Choose randomly c initial prototypes from the object set
repeat
(1). $t \leftarrow t + 1$
(2). Compute $\boldsymbol{M}_t$ using Eq. (5.20), Eq. (5.21) and $\boldsymbol{V}_{t-1}$
(3). Compute the prototype weights for specific classes using Eq. (5.29)
(4). Compute the prototype weights for imprecise classes using Eq. (5.26) and get the new $\boldsymbol{V}_t$.
until the prototypes remain unchanged.
Output: The optimal credal partition.

Remark 6. Existing work has studied the convergence properties of the partitioning clustering algorithms, such as C-Means, and C-Medoids. As we can see, wECMdd follows a similar clustering approach. The optimization process consists of three steps: cluster assignment update, prototype weights of specific classes update and then prototype weights of imprecise classes update. The first two steps improve the objective function value by the application of Lagrangian multiplier method. The third step tries to find good representative objects for imprecise classes. If the method to determine the weights for imprecise classes is of practical meaning, it will also keep the objective function increasing. In fact the approach of updating the prototype weights is similar to the idea of one-step Gaussian-Seidel iteration method, where the computation of the

new variable vector uses the new elements that have already been computed, and the old elements that have not yet to be advanced to the next iteration. In Section 5.5, we will demonstrate through experiments that wECMdd could converge in a few number of iterations.

5.4 Application issues

In this section, some problems when applying the ECMdd algorithms, such as how to adjust the parameters and how to select the initial prototypes for each class, will be discussed.

5.4.1 The parameters of the algorithm

As in ECM, before running ECMdd, the values of the parameters have to be set. Parameters α, β and δ have the same meanings as those in ECM. The value β can be set to be $\beta = 2$ in all experiments for which it is a usual choice. The parameter α aims to penalize the subsets with high cardinality and control the amount of points assigned to imprecise clusters for credal partitions. The higher α is, the less mass belief is assigned to the meta clusters and the less imprecise will be the resulting partition. However, the decrease of imprecision may result in high risk of errors. For instance, in the case of hard partitions, the clustering results are completely precise but there is much more intendancy to partition an object to an unrelated group. As suggested in (Masson and Denoeux, 2008), a value can be used as a starting default one but it can be modified according to what is expected from the user. The choice δ is more difficult and is strongly data dependent (Masson and Denoeux, 2008).

In sECMdd, parameter γ weighs the contribution of uncertainty to the dissimilarity between objects and imprecise clusters. Parameter η is used to distinguish the outliers from the possible medoids when determining the prototypes of meta classes. It could be set 1 by default and it has little effect on the final partition results. Parameters ξ and ψ are for specially for wECMdd. Similar to β, ψ is used to control the smoothness of the weight distribution. Parameter ξ is used for not assigning the outliers large weights for imprecise classes. If there are few outliers in the data set, it could be set to be near 0.

As in MECM, the validity index for credal partitions defined by Masson and Denoeux (2008) can be used to determine the number of clusters:

$$N^*(c) \triangleq \frac{1}{n \log_2(c)} \times \sum_{i=1}^{n} \left[\sum_{A \in 2^\Omega \setminus \emptyset} m_i(A) \log_2 |A| + m_i(\emptyset) \log_2(c) \right], \tag{5.30}$$

where $0 \leq N^*(c) \leq 1$. This index has to be minimized to get the optimal number of

classes in the data set.

5.4.2 The initial prototypes

The c-means type clustering algorithms are sensitive to the initial prototypes. In this work, we follow the initialization procedure as the one used in (Krishnapuram et al., 2001; Mei and Chen, 2010) to generate a set of c initial prototypes one by one. The first medoid, σ_1, is randomly picked from the data set. The rest of medoids are selected successively one by one in such a way that each one is most dissimilar to all the medoids that have already been picked. Suppose $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_j\}$ is the set of the first chosen j ($j < c$) medoids. Then the $j+1$ medoid, σ_{j+1}, is set to the object x_p with

$$p = \arg \max_{1 \leq i \leq n; x_i \notin \sigma} \left\{ \min_{\sigma_k \in \sigma} d(x_i, \sigma_k) \right\}. \tag{5.31}$$

This selection process makes the initial prototypes evenly distributed and locate as far away from each other as possible. It is noted that another scheme is that the first medoid is set to be the object with the smallest total dissimilarity to all the other objects, *i.e.* $\sigma_1 = x_r$ with

$$r = \arg \min_{1 \leq i \leq n} \left\{ \sum_{j=1}^{n} d(x_i, x_j) \right\}, \tag{5.32}$$

and the remaining prototypes are selected the same way as before. Krishnapuram et al. (2001) have pointed out that both initialization schemes work well in practice. But based on our experiments, for credal partitions, a bit of randomness of the first prototype might be desirable.

5.4.3 Making the important objects more important

In wECMdd, a matrix $\boldsymbol{V} = \{v_{ji}^{2^\Omega}\}$ is used to record prototype weights of n objects with respect to all the clusters, including the specific classes and imprecise classes. All objects are engaged in describing clusters information with some weights assigned to each detected classes. This seems unreasonable since it is easy to understand that when an object does not belong to a cluster, it should not participate in describing that cluster (Gao et al., 2014). Therefore, in each iteration of wECMdd, after the weights $v_{ki}^{\Omega}, k = 1, 2, \cdots, c, i = 1, 2, \cdots, n$ of x_i for all the specific classes ω_k are got by

Eq. (5.29), the normalized weights w_{ki}^{Ω} could be calculated by [8]

$$w_{ki}^{\Omega} = \frac{v_{ki}^{'}}{\sum_{i=1}^{n} v_{ki}^{'}} \ , i = 1, 2, \cdots, n, \text{ and } k = 1, 2, \cdots, c, \tag{5.33}$$

where $v_{ki}^{'}$ equals to v_{ki}^{Ω} if x_i belongs to ω_k, 0 otherwise. Remark that x_i is regarded as a member of class ω_k if $m_i(\{\omega_k\})$ is the maximum of the masses assigned to all the focal sets at this iteration. In fact, if we want to make the important "core" objects more important in each cluster, a subset of fixed cardinality $1 \leq q \ll n$ of objects X could be used. The q objects constitute core of each cluster, and collaborate to describe information of each class. This kind of wECMdd with q medoids in each class is denoted by wECMdd-q. More generally, q could be different for each cluster. However, how to determine q or the number of cores in every class should be considered. This is not the topic of this work and we will study that in the future work.

5.5 Experiments

In this section some experiments on generated and real data sets will be performed to show the effectiveness of sECMdd and wECMdd. The results are compared with other relational clustering approaches PAM (Kaufman and Rousseeuw, 2009), FCMdd (Krishnapuram et al., 2001), FMMdd (Mei and Chen, 2011) and MECM (Zhou et al., 2015c) to illustrate the advantages of credal partitions and multi-prototype representativeness of classes. The Precision (P), Recall (R) and Rand Index (RI), and the indices for evaluating credal partitions, Evidential Precision (EP), Evidential Recall (ER) and Evidential Rank Index (ERI) presented in Section 3.4.4 will be used to compare.

5.5.1 Overlapped data sets

Due to the introduction of imprecise classes, credal partitions have the advantage to detect overlapped clusters. In the first example, we will use overlapped data sets to illustrate the behavior of the proposed algorithms. We start by generating 3×361 points distributed in three overlapped circles with a same radius $R = 5$ but with different centers. The coordinates of the first circle's center are $(5, 6)$ while the coordinates of the other two circles' centers are $(0, 0)$ and $(9, 0)$. The data set is displayed in Figure 5.1-a.

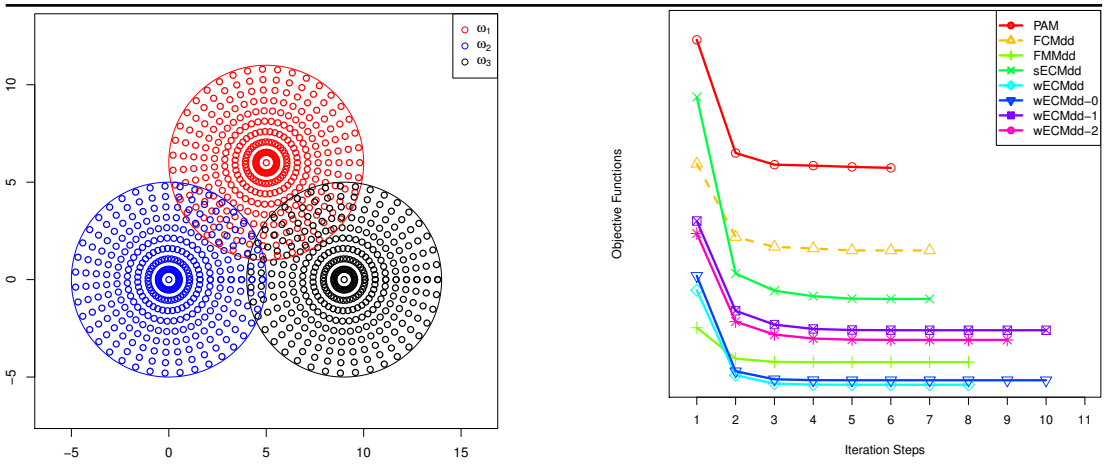
Figure 5.1-b shows the iteration steps for different methods. For ECMdd clustering algorithms, there are three alternative steps to optimize the objective function (assign-

[8] In the following we call this type of prototype weights "normalized weights", and wECMdd with normalized weights is denoted by wECMdd-0. The standard wECMdd with multiple weights on all the objects described in the last section is still denoted by wECMdd.

ment update, and the update for medoids of specific and imprecise classes), while only two steps (update of membership and specific classes' prototypes) are required for the existing methods (PAM, FCMdd and FMMdd). But we can see from the figure, the added third step for calculating the new prototypes of imprecise classes in ECMdd clustering has no effects on the convergence.

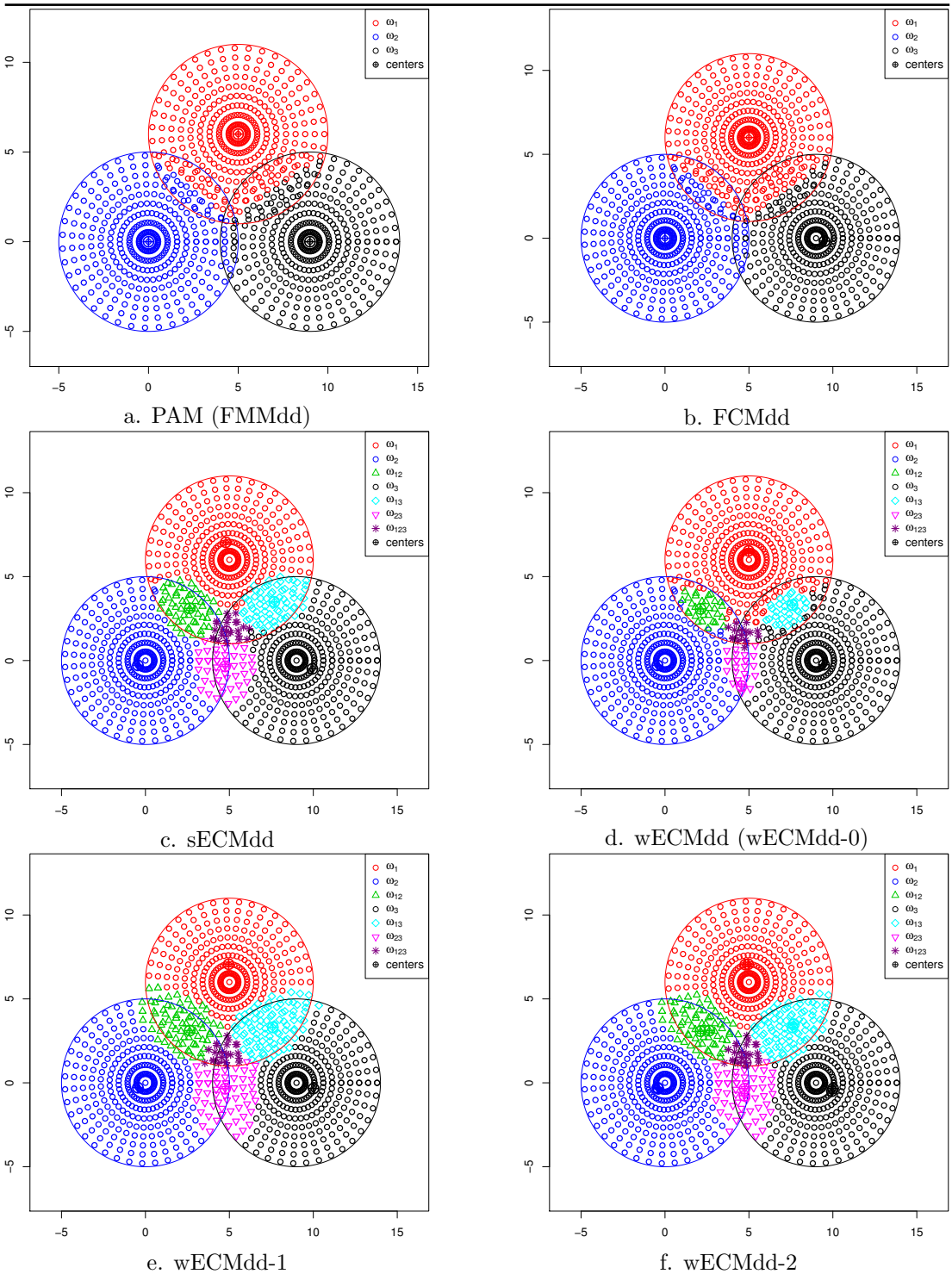
The fuzzy and credal partitions by different methods are shown in Figure 5.2, and the values of the evaluation indices are listed in Table 5.1. The objects are clustered into the class with the maximum membership values for fuzzy partitions (by FCMdd, FMMdd), while for credal partitions (by different ECMdd algorithms), with the maximum mass assignment. As a result, imprecise classes, such as $\{\omega_1, \omega_2\}$ (denoted by ω_{12} in the figure), are produced by ECMdd clustering to accept the objects for which it is difficult to make a precise (hard) decision. Consequently, the EP values of the credal partitions by ECMdd algorithms are distinctly high, which indicates that such soft decision mechanism could make the clustering result more "cautious" and decrease the misclassification rate.

In this experiment, all the ECMdd algorithms are run with: $\alpha = 2, \beta = 2, \delta = 100$. For sECMdd, $\eta = 1$ and for wECMdd $\gamma = 1.2, \xi = 3$. The results by wECMdd and wECMdd-0 are similar, as they both use weights of objects to describe the cluster structure. The ECMdd algorithms using one (sECMdd, wECMdd-1) or two (wECMdd-2) objects to represent a class are sensitive to the detected prototypes. More objects that are not located in the overlapped area are inclined to be partitioned into the imprecise classes by these methods.



a. Original data set b. Iteration steps

Figure 5.1: The overlapped data sets.



a. PAM (FMMdd) b. FCMdd

c. sECMdd d. wECMdd (wECMdd-0)

e. wECMdd-1 f. wECMdd-2

Figure 5.2: Clustering on overlapped data sets. All the methods are evoked with the same initial medoids. The prototypes in the detected classes by each method are marked with $\oplus$. For wECMdd and wECMdd-0, the object with maximum weight in each class is marked as medoid. The results of PAM and FMMdd are similar, so we only display the figure of PAM to save space. And so also are the results for wECMdd and wECMdd-0.

Table 5.1: The clustering results on the overlapped data set.

	P	R	RI	EP	ER	ERI
PAM	0.8701	0.8701	0.9136	0.8701	0.8701	0.9136
FCMdd	0.8731	0.8734	0.9156	0.8731	0.8734	0.9156
FMMdd	0.8703	0.8702	0.9136	0.8703	0.8702	0.9136
sECMdd	0.8715	0.8730	0.9149	0.9889	0.6799	0.8910
wECMdd	0.8703	0.8705	0.9137	0.9726	0.7181	0.8994
wECMdd-0	0.8737	0.8738	0.9159	0.9405	0.7732	0.9083
wECMdd-1	0.8746	0.8764	0.9171	1.0000	0.6015	0.8674
wECMdd-2	0.8763	0.8780	0.9182	1.0000	0.6213	0.8740

The running time of sECMdd, wECMdd, MECM, PAM, FCMdd, FMMdd is calculated to show the computational complexity[9]. Each algorithm is evoked 10 times with different initial parameters, and the average elapsed time is displayed in Table 5.2. As we can see from the table, ECMdd is of higher complexity compared with fuzzy or hard medoid based clustering. This is easy to understand, as in the partitions there are imprecise classes and the membership is considered on the extended frame of the power set 2^{Ω}. But credal partitions by the use of ECMdd will improve the precision of the clustering results. This is also important in some applications, where cautious decisions are more welcome to avoid the possible high risk of misclassification.

Table 5.2: The average running time of different algorithms.

	sECMdd	wECMdd	MECM	PAM	FCMdd	FMMdd
Elapsed Time (s)	19.1100	14.2260	330.4680	1.3000	1.3480	6.9080

In order to show the influence of parameters in ECMdd algorithms, different values of α, η, ξ, δ and β have been tested for this data set. Figure 5.3-a displays the three evidential indices varying with α by sECMdd, while Figure 5.3-b depicts the results of wECMdd with different α. As we can see, for both sECMdd and wECMdd, if we want to make more imprecise decisions to improve ER, parameter α can be decreased, since α tries to adjust the penalty degree to control the imprecise rates of the results. Keeping more soft decisions will reduce the misclassification rate and makes the specific decisions more accurate. But the partition results with few specific decisions have low ER values and they are of limited practical meaning. In application we should determine α based on the requirement. Parameter η in sECMdd and ξ in wECMdd are both for distinguish

[9]All the algorithms in this work are implemented with R 3.2.1

the outliers in imprecise classes. As pointed out in Figures 5.3-c and 5.3-d, if η and ξ are well set, they have little effect on the final clusterings. The same is true in the case of δ which is applied to detect outliers (see Figure 5.3-f). The effect of various values of β is illustrated in Figure 5.3-e. We can see that it has little influence on the final results as long as it is larger than 1. Similar to FCM and ECM, the value of β could also be set to be 2 as a usual choice here. Compared with MECM (the discussion about the parameters of MECM could be seen in (Zhou et al., 2015c)), the parameters of ECMdd are much easier to adjust and control.

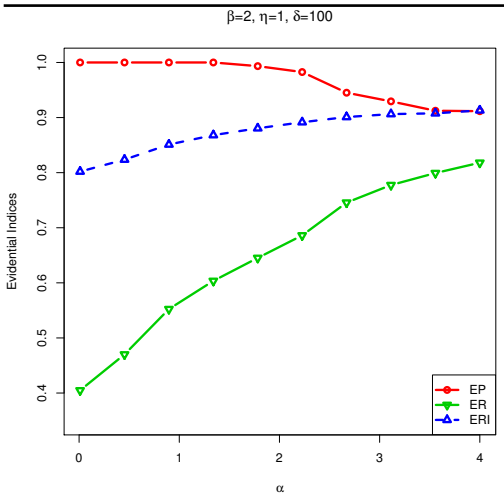
5.5.2 Gaussian data set

In the second experiment, we test on a data set consisting of 10000 points generated from different Gaussian distributions. The points are from 10 Gaussian distributions, the mean values of which are uniformly located in a circle. The data set is displayed in Figure 5.4.

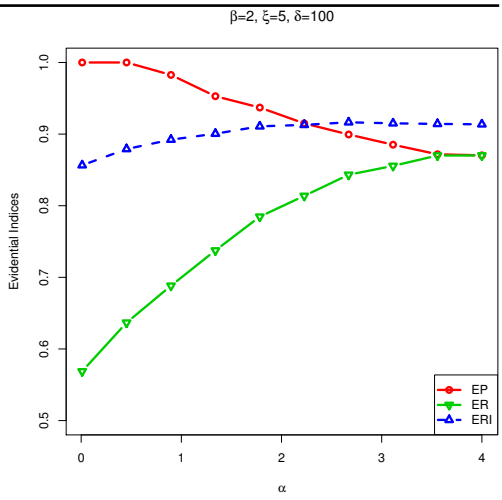
Table 5.3: The clustering results on Gaussian data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.8939	0.8940	0.8988	0.8939	0.8940	0.8988	118.2097
FCMdd	0.8960	0.8960	0.8992	0.8960	**0.8960**	0.8992	152.4320
FMMdd	0.8928	0.8980	0.8996	0.8980	0.8928	0.8996	197.5340
MECM	**0.8980**	0.8940	0.8921	0.9932	0.3173	0.9321	19430.1560
sECMdd	0.8931	**0.8992**	**0.9043**	**1.0000**	0.4468	0.9452	8987.7390
wECMdd	0.8923	0.8914	0.8908	**1.0000**	0.5623	**0.9566**	8534.8740

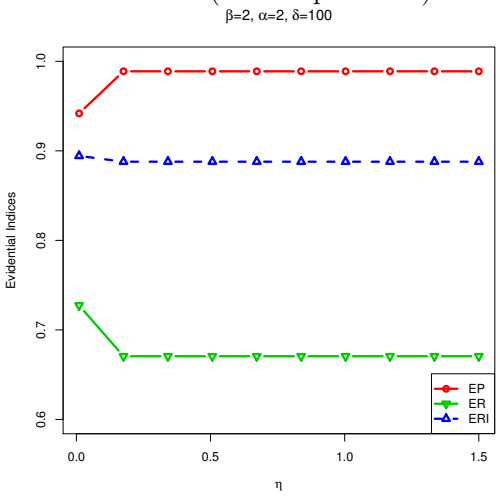
Table 5.3 lists the indices for evaluating the different methods. Bold entries in each column of this table (and also other tables in the following) indicate that the results are significant as the top performing algorithm(s) in terms of the corresponding evaluation index. We can see that the precision, recall and RI values for all approaches are similar. As the objects are from gaussian distributions, it is intuitive that there is only one geometrical center in each class. That's why the one-prototype based clustering sECMdd is a little better than wECMdd. For evidential clusterings, *e.g.,* MECM, sECMdd and wECMdd, the three classical measures are based on the associated pignistic probabilities. It indicates that credal partitions can provide the same information as crisp and fuzzy ones (PAM, FCMdd, and FMMdd). Most of the misclassifications in this experiment come from the objects lying in the overlapped area between two classes.



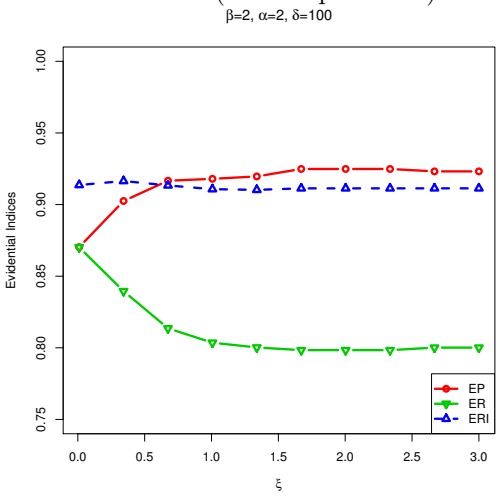
a. sECMdd (with respect to α)



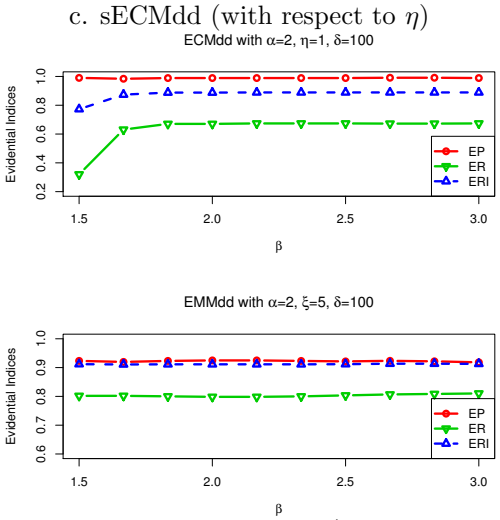
b. wECMdd (with respect to α)



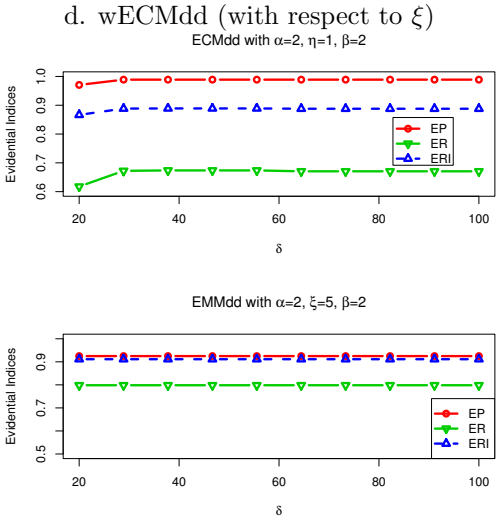
c. sECMdd (with respect to η)



d. wECMdd (with respect to ξ)



e. sECMdd and wECMdd (with respect to β)



f. sECMdd and wECMdd (with respect to δ)

Figure 5.3: Clustering of overlapped data with different parameters.

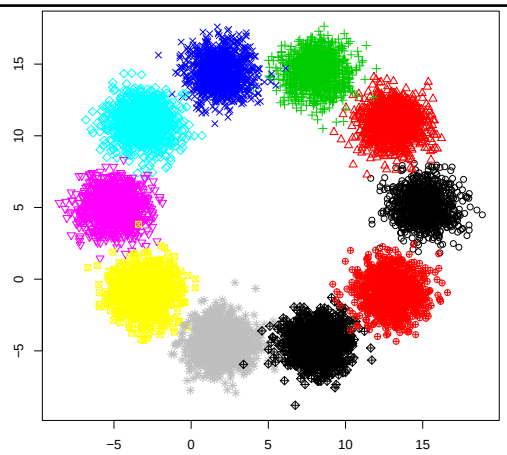


Figure 5.4: Gaussian data set.

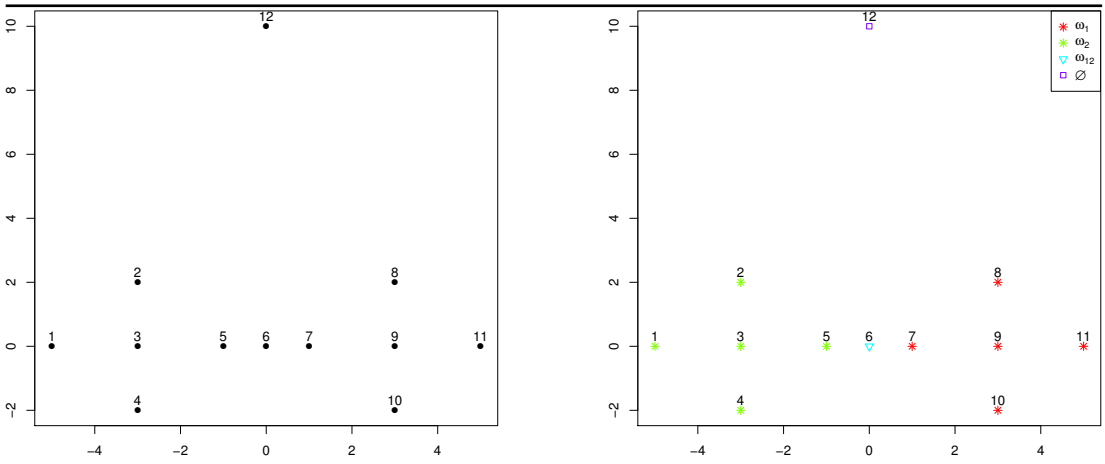
However, from the same table, we can also see that the evidential measures EP and ERI by sECMdd and wECMdd are higher (for hard partitions, the values of evidential measures equal to the corresponding classical ones) than the ones obtained by other methods. This fact confirms the accuracy of the specific decisions *i.e.* decisions clustering the objects into specific classes. The advantage can be attributed to the introduction of imprecise clusters, with which we do not have to partition the uncertain or unknown objects lying in the overlap into a specific cluster. Consequently, it could reduce the risk of misclassification. For the computational time, the same conclusion as in the first experiment can be obtained. Evidential clustering algorithms (sECMdd, wECMdd and MECM) are more time-consuming than hard or fuzzy ones. But we can see that wECMdd is the fastest one among the three, and it is significantly better than MECM in terms of complexity.

5.5.3 X_{12} data set

In this test, a simple classical data set composed of 12 objects represented in Figure 5.5-a is considered. As we can see from the figure, objects 1 - 11 are clearly dived into two groups whereas object 12 is an outlier. The results by sECMdd and wECMdd are shown in Figure 5.5-b. Object 6 is clustered into imprecise class $\omega_{12} \triangleq \{\omega_1, \omega_2\}$ while object 12 is regarded as an outlier (belonging to $\emptyset$).

In this data set, object 6 is a "good" member for both classes, whereas object 12 is a "poor" point. It can be seen from Table 5.4 that the fuzzy partition by FCMdd also gives large equal membership values to ω_1 and ω_2 for object 12, just like in the case of such good members as point 6. The same is true for PAM and FMMdd. The obtained results show the problem of distinguishing between ignorance and the "equal evidence" (uncertainty) for fuzzy partitions. But the table shows that the credal partition by

wECMdd assigns largest mass belief to $\emptyset$ for object 12, indicating it is an outlier. Moreover, the values $v_{ji}^{2^\Omega}$ in the table are the weights of object i for class A_j, from which it can be seen that object 3 and object 9 play the center role in their own class, while object 6 contributes most to the overlapped parts of the two classes. Thus the prototype weights indeed could provide us some rich information about the cluster structure.



a. Original data set b. sECMdd & wECMdd

Figure 5.5: A simple data set of 12 objects.

5.5.4 X_{11} data set

In this experiment, we will show the effectiveness of the application of multiple weighted prototypes using the data set displayed in Figure 5.6. The X_{11} data set has two obvious clusters, one containing objects 1 to 4 and the other including objects 5 to 10. Object 11 locates slightly biased to the cluster on the right side. It can be seen that in the left class, it is unreasonable to describe the cluster structure using any one of the four objects in the group, since no one of the four points could be viewed as a more proper representative than the other three. The clustering results by FCMdd, sECMdd, wECMdd are listed in Table 5.5. The result by MECM is not listed here as it is similar to that by sECMdd.

From the table we can see that the two clustering approaches, FCMdd and sECMdd, which using a single medoid cluster to represent a cluster, partition object 11 to cluster 1 for mistake. This is resulted by the fact that both of them set object 4 to be the center of class ω_1. On the contrary, in wECMdd, the four objects in cluster ω_1 are thought

Table 5.4: The clustering results of X_{12} data set using FCMdd and wECMdd. The objects marked with * are the medoids found by FCMdd. Values $m_{ij}, j = 1, 2, 3, 4$ are the mass assigned to x_i for class $\emptyset$, ω_1, ω_2 and imprecise class $\omega_{12} \triangleq \{\omega_1, \omega_2\}$. Values $v_{ij}^{2^\Omega}, j = 1, 2, 3$ are the weights of object x_i for class ω_1, ω_2 and ω_{12}.

	FCMdd		wECMdd								
id	u_{i1}	u_{i2}	m_{i1}	m_{i2}	m_{i3}	m_{i4}	$BetP_{i1}$	$BetP_{i2}$	$v_{1i}^{2^\Omega}$	$v_{2i}^{2^\Omega}$	$v_{3i}^{2^\Omega}$
1	0.9412	0.0588	0.1054	0.7242	0.1599	0.0105	0.8154	0.1846	0.1123	0.0230	0.0000
2	0.9091	0.0909	0.0749	0.7282	0.1825	0.0144	0.7950	0.2050	0.1396	0.0359	0.0000
3	1.0000	0.0000*	0.0502	0.8005	0.1354	0.0140	0.8501	0.1499	**0.1829**	0.0382	0.0000
4	0.9091	0.0909	0.0821	0.7083	0.1938	0.0158	0.7803	0.2197	0.1117	0.0337	0.0000
5	0.8000	0.2000	0.0438	0.5969	0.2498	0.1095	0.6815	0.3185	0.1386	0.0709	0.0001
6	0.5000	0.5000	0.0000	0.0000	0.0000	1.0000	0.5000	0.5000	0.0997	0.0999	**0.9998**
7	0.2000	0.8000	0.0437	0.2463	0.6006	0.1094	0.3147	0.6853	0.0707	0.1388	0.0001
8	0.0909	0.9091	0.0753	0.1813	0.7289	0.0145	0.2039	0.7961	0.0358	0.1395	0.0000
9	0.0000	1.0000*	0.0507	0.1351	0.8001	0.0141	0.1497	0.8503	0.0381	**0.1823**	0.0000
10	0.0909	0.9091	0.0825	0.1927	0.7089	0.0159	0.2186	0.7814	0.0336	0.1115	0.0000
11	0.0588	0.9412	0.1063	0.1596	0.7235	0.0106	0.1845	0.8155	0.0230	0.1119	0.0000
12	0.5000	0.5000	0.3803	0.3042	0.3060	0.0095	0.4986	0.5014	0.0142	0.0143	0.0001

to have nearly the same contribution to the class. Consequently, object 11 is clustered into ω_2 correctly. FMMdd could also get the exactly accurate results as it is also take use of multiple weighted medoids. This experiment shows that the multi-prototype representation of classes could capture some complex data structure and consequently enhance the clustering performance. It is remarkable that the hard partition could be recovered from pignistic probability (BetP) for credal partitions. And the results of these experiments reflects that pignistic probabilities play a similar role as fuzzy membership.

5.5.5 Karate Club network

Graph visualization is commonly used to visually model relations in many areas. For graphs such as social networks, the prototype (center) of one group is likely to be one of the persons (*i.e.* nodes in the graph) playing the leader role in the community. Moreover, a graph (network) of vertices (nodes) and edges usually describes the interactions between different agents of the complex system and the pair-wise relationships between nodes are often implied in the graph data sets. Thus medoids-based relational clustering algorithms could be directly applied. In this section we will evaluate the effectiveness of the proposed methods applied on community detection problems.

Here the widely used benchmark in detecting community structures, "Karate Club", studied by Wayne Zachary is considered. There are many similarity and dissimilarity

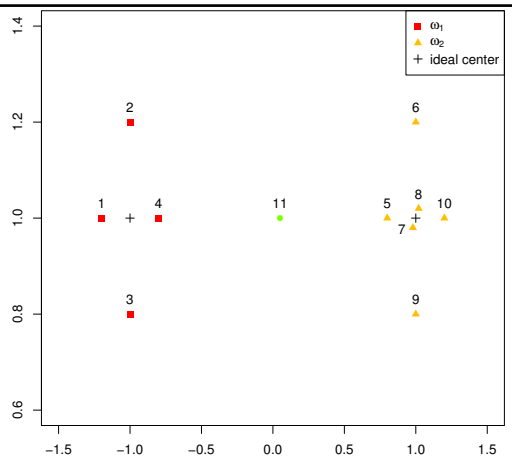


Figure 5.6: A simple data set of 11 objects. The idea centers of the two clusters are located at (-1, 1) and (1, 1). The coordinates of object 11 are (0.05, 1), which is closer to the center of cluster 2.

indices for networks, using local or global information of graph structure. In this experiment, the different similarity metrics will be compared first. The similarity indices considered here are listed in Table 5.6 [10]. It is notable that the similarities by these measures are from 0 to 1, thus they could be converted into dissimilarities simply by $dissimilarity = 1 - similarity$. The comparison results for different dissimilarity indices by FCMdd and sECMdd are shown in Table 5.7 and Table 5.8 respectively. As we can see, for all the dissimilarity indices, for sECMdd, the value of evidential precision is higher than that of precision. This can be attributed to the introduced imprecise classes which enable us not to make a hard decision for the nodes that we are uncertain and consequently guarantee the accuracy of the specific clustering results. From the table we can also see that the performance using the dissimilarity measure based on signal prorogation is better than those using local similarities in the application of both FCMdd and sECMdd. This reflects that global dissimilarity metric is better than the local ones for community detection. Thus in the following experiments, we only consider the signal dissimilarity index.

The detected community structures by different methods are displayed in Figures 5.7-b – 5.7-d. FCMdd could detect the exact community structure of all the nodes except nodes 3, 14, 20. As we can see from the figures, these three nodes have connections with both communities. They are partitioned into imprecise class $\omega_{12} \triangleq \{\omega_1, \omega_2\}$, which describing the uncertainty on the exact class labels of the related nodes, by the application of sECMdd. The medoids found by FCMdd of the two specific communities are node 5 and node 29, while by sECMdd node 5 and node 33. The uncertain nodes

[10] A more detailed description could be found in Section 2.5.1.

Table 5.5: The clustering results of X_{11} data set. The objects marked with * are the medoids found by FCMdd and sECMdd. Values $v_{ij}, j = 1, 2, 3$ are the weights of object x_i for class ω_1, ω_2 and imprecise class $\omega_{12} \triangleq \{\omega_1, \omega_2\}$.

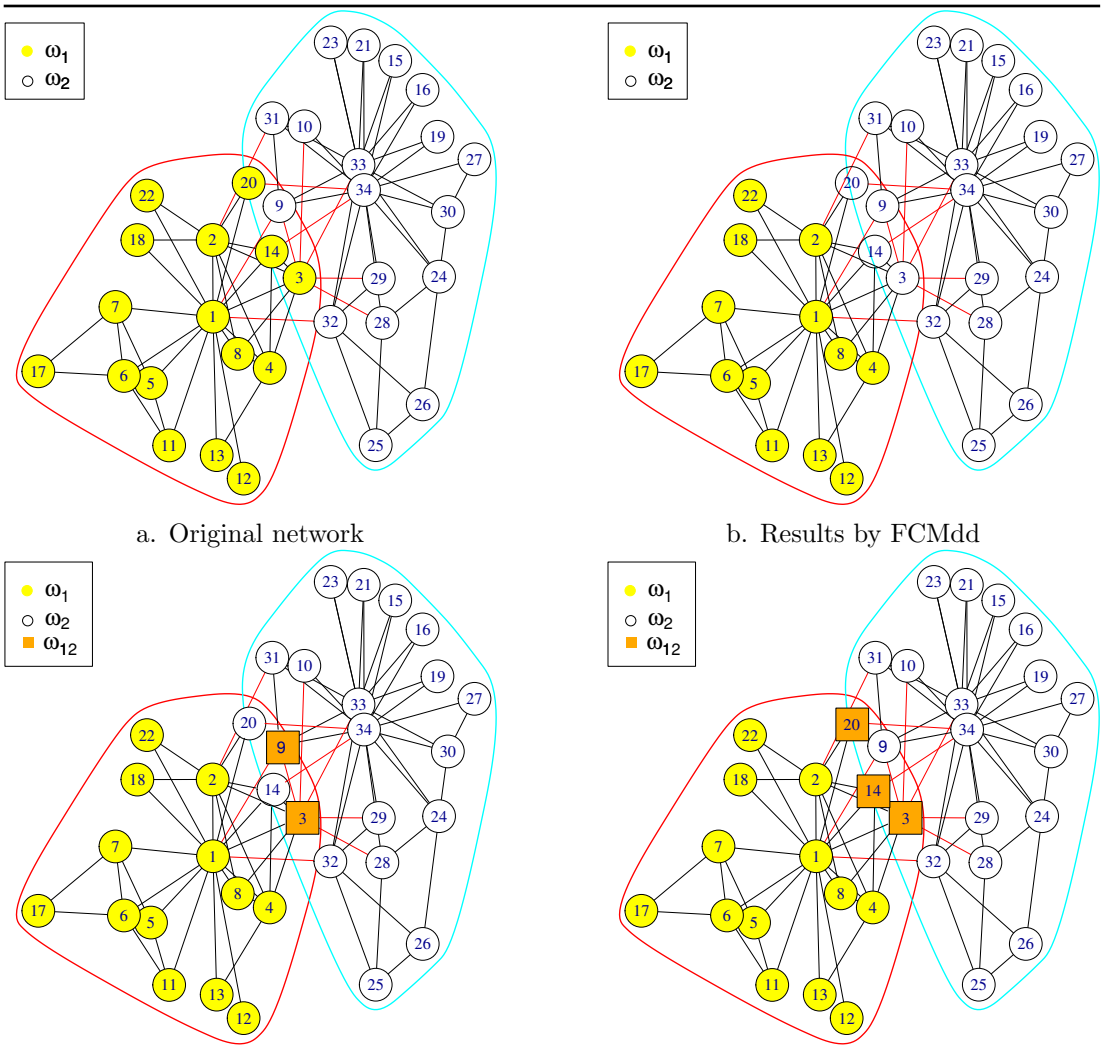
	FCMdd		sECMdd		wECMdd				
id	u_{i1}	u_{i2}	$BetP_{i1}$	$BetP_{i2}$	$BetP_{i1}$	$BetP_{i2}$	v_{i1}	v_{i2}	v_{i3}
1	0.9674	0.0326	0.9510	0.0490	0.9620	0.0380	**0.1477**	0.0414	0.0018
2	0.9802	0.0198	0.9671	0.0329	0.9578	0.0422	0.1476	0.0433	0.0024
3	0.9802	0.0198	0.9667	0.0333	0.9578	0.0422	0.1476	0.0433	0.0024
4	1.0000	0.0000*	1.0000	0.0000*	0.9517	0.0483	0.1475	0.0457	0.0033
5	0.0127	0.9873	0.0958	0.9042	0.0169	0.9831	0.0585	0.1190	0.0320
6	0.0147	0.9853	0.0383	0.9617	0.0145	0.9855	0.0554	0.1187	0.0223
7	0.0000	1.0000*	0.0327	0.9673	0.0073	0.9927	0.0558	**0.1447**	0.0117
8	0.0010	0.9990	0.0198	0.9802	0.0072	0.9928	0.0553	0.1445	0.0111
9	0.0099	0.9901	0.5000	0.5000	0.0144	0.9856	0.0554	0.1187	0.0223
10	0.0121	0.9879	0.0000	1.0000*	0.0128	0.9872	0.0530	0.1183	0.0167
11	0.5450	0.4550	0.5723	0.4277	0.4990	0.5010	0.0761	0.0625	**0.8739**

Table 5.6: Different local and global similarity indices.

Index name	Global metric	Ref.
Jaccard	No	(Jaccard, 1912)
Pan	No	(Pan et al., 2010)
Zhou	No	(Zhou et al., 2009)
Signal	Yes	(Hu et al., 2008)

found by MECM are node 3 and node 9.

The results by wECMdd algorithms are similar to that by sECMdd. Table 5.9 lists the prototype weights obtained by FMMdd and wECMdd. The nodes in each community are ordered by prototype weights in the table. We just display the first ten important members in every class. From the weight values by FMMdd and wECMdd in the table we can get the same conclusion: nodes 1 and 12 play the center role in community ω_1, while node 33 and 34 consists the two cores in community ω_2. But by wECMdd more information about the overlapped structure of the network is available. As we can see from the last two columns of the table, node 9 contributes most to the overlapped community ω_{12}, which is a good reflection of its "bridge" role for the two classes. Therefore, the prototype weights provide us some information about the cluster structure from another point of view, which could help us gain a better understanding of the inner structure of a class.



a. Original network

b. Results by FCMdd

c. Results by MECM

d. Results by sECMdd

Figure 5.7: The Karate Club network. The parameters of MECM are $\alpha = 1.5, \beta = 2, \delta = 100, \eta = 0.9, \gamma = 0.05$. In sECMdd, $\alpha = 0.05, \beta = 2, \delta = 100, \eta = 1, \gamma = 1$, while in FCMdd, $\beta = 2$.

Table 5.7: Comparison of different similarity indices by FCMdd.

Index	P	R	RI	EP	ER	ERI
Jaccard	0.6364	0.7179	0.6631	0.6364	0.7179	0.6631
Pan	0.4866	1.0000	0.4866	0.4866	1.0000	0.4866
Zhou	0.4866	1.0000	0.4866	0.4866	1.0000	0.4866
Signal	0.8125	0.8571	0.8342	0.8125	0.8571	0.8342

Table 5.8: Comparison of different similarity indices by sECMdd.

Index	P	R	RI	EP	ER	ERI
Jaccard	0.6458	0.6813	0.6631	0.7277	0.5092	0.6684
Pan	0.6868	0.7070	0.7005	0.7214	0.6923	0.7201
Zhou	0.6522	0.6593	0.6631	0.7460	0.3443	0.6239
Signal	1.0000	1.0000	1.0000	1.0000	0.6190	0.8146

5.5.6 Countries data

In this section we will test on a relational data set, referred as the benchmark data set Countries Data (Kaufman and Rousseeuw, 2009; Mei and Chen, 2010). The task is to group twelve countries into clusters based on the pairwise relationships as given in Table 5.10, which is in fact the average dissimilarity scores on some dimensions of quality of life provided subjectively by students in a political science class. Generally, these countries are classified into three categories: Western, Developing and Communist. The parameters are set as $\beta = 2$ for FCMdd, and $\beta = 2, \alpha = 0.95, \eta = 1, \gamma = 1$ for sECMdd. We test the performances of FCMdd and sECMdd with two different sets of initial representative countries: $\Delta_1 = \{$C10: USSR; C8: Israel; C7: India$\}$ and $\Delta_2 = \{$C6: France; C4: Cuba; C1: Belgium$\}$. The three countries in Δ_1 are well separated. On the contrary, for the countries in Δ_2, Belgium is similar to France, which makes two initial medoids of three are very close in terms of the given dissimilarities.

The results of FCMdd and sECMdd are given in Table 5.11 and Table 5.12 respectively. It can be seen that FCMdd is very sensitive to initializations. When the initial prototypes are well set (the case of Δ_1), the obtained partition is reasonable. However, the clustering results become worse when the initial medoids are not ideal (the case of Δ_2). In fact two of the three medoids are not changed during the update process of FCMdd when using initial prototype set Δ_2. This example illustrates that FCMdd is quite easy to be stuck in a local minimum. For sECMdd, the credal partitions are the same with different initializations. The pignistic probabilities are also displayed in Table 5.12, which can be regarded as membership values in fuzzy partitions. The country Egypt is clustered into imprecise class $\{1, 2\}$, indicating that Egypt is not so well

Table 5.9: The prototype weights by FMMdd and wECMdd. Community ω_{12} denotes the imprecise community $\{\omega_1, \omega_2\}$. Only the first 10 nodes with largest weight values in each community are listed.

FMMdd				wECMdd					
Community ω_1		Community ω_2		Community ω_1		Community ω_2		Community ω_{12}	
Node	Weights	Node	Weights	Node	Weights	Node	Weights	Node	Weights
1	0.0689	33	0.0607	12	0.0707	33	0.0606	9	0.3194
12	0.0663	34	0.0565	1	0.0659	34	0.0562	3	0.1348
22	0.0590	28	0.0556	13	0.0588	24	0.0557	20	0.1254
18	0.0590	24	0.0551	18	0.0584	28	0.0549	25	0.0989
13	0.0583	15	0.0512	22	0.0584	15	0.0519	10	0.0493
2	0.0548	16	0.0512	5	0.0519	16	0.0519	32	0.0453
4	0.0544	19	0.0512	11	0.0519	19	0.0519	26	0.0429
8	0.0537	21	0.0512	4	0.0506	21	0.0519	29	0.0379
14	0.0469	23	0.0512	8	0.0503	23	0.0519	14	0.0351
5	0.0436	31	0.0504	2	0.0500	30	0.0509	31	0.0306

belongs to Developing or Western alone, but belongs to both categories. This result is consistent with the fact shown from the dissimilarity matrix: Egypt is similar to both USA and India, but has the largest dissimilarity to China. The results by wECMdd and MECM algorithms are not displayed here, as they product the same clustering result with sECMdd. From this experiment we can conclude that ECMdd is more robust to the initializations than FCMdd.

5.5.7 UCI data sets

Finally the clustering performance of different methods will be compared on eight benchmark UCI data sets (Lichman, 2013) summarized in Table 5.13. Euclidean distance is used as the dissimilarity measure for the object data sets, and the Signal dissimilarity is adopted for the graph data sets.

Same as ECM, the number of parameters to be optimized in ECMdd is exponential and depends on the number of clusters (Masson and Denoeux, 2008). For the number of classes larger than 10, calculations are not tractable. But we can only consider a subclass with a limited number of focal sets (Masson and Denoeux, 2008). Here we constrain the focal sets to be composed of at most two classes (except Ω). The evaluation results are listed in Tables 5.14–5.21.

It can be seen that generally wECMdd works better than the other approaches on all of the data sets, except for Iris data set where sECMdd works best. This may be explained by the fact that, Iris is a small data set and each class can be well represented by one prototype. wECMdd has better performance for the other complex

Table 5.10: Countries data: dissimilarity matrix.

	Countries	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12
1	C1: Belgium:	0.00	5.58	7.00	7.08	4.83	2.17	6.42	3.42	2.50	6.08	5.25	4.75
2	C2: Brazil	5.58	0.00	6.50	7.00	5.08	5.75	5.00	5.50	4.92	6.67	6.83	3.00
3	C3: China	7.00	6.50	0.00	3.83	8.17	6.67	5.58	6.42	6.25	4.25	4.50	6.08
4	C4: Cuba	7.08	7.00	3.83	0.00	5.83	6.92	6.00	6.42	7.33	2.67	3.75	6.67
5	C5: Egypt	4.83	5.08	8.17	5.83	0.00	4.92	4.67	5.00	4.50	6.00	5.75	5.00
6	C6: France	2.17	5.75	6.67	6.92	4.92	0.00	6.42	3.92	2.25	6.17	5.42	5.58
7	C7: India	6.42	5.00	5.58	6.00	4.67	6.42	0.00	6.17	6.33	6.17	6.08	4.83
8	C8: Israel	3.42	5.50	6.42	6.42	5.00	3.92	6.17	0.00	2.75	6.92	5.83	6.17
9	C9: USA	2.50	4.92	6.25	7.33	4.50	2.25	6.33	2.75	0.00	6.17	6.67	5.67
10	C10: USSR	6.08	6.67	4.25	2.67	6.00	6.17	6.17	6.92	6.17	0.00	3.67	6.50
11	C11: Yugoslavia	5.25	6.83	4.50	3.75	5.75	5.42	6.08	5.83	6.67	3.67	0.00	6.92
12	C12: Zaire	4.75	3.00	6.08	6.67	5.00	5.58	4.83	6.17	5.67	6.50	6.92	0.00

Table 5.11: Clustering results of FCMdd for countries data. The prototype (medoid) of each class is marked with *.

		FCMdd with Δ_1					FCMdd with Δ_2				
	Countries	u_{i1}	u_{i2}	u_{i3}	Label	Medoids	u_{i1}	u_{i2}	u_{i3}	Label	Medoids
1	C1: Belgium	0.4773	0.2543	0.2685	1	-	1.0000	0.0000	0.0000	1	*
2	C6: France	0.4453	0.2719	0.2829	1	-	0.0000	1.0000	0.0000	2	*
3	C8: Israel	1.0000	0.0000	0.0000	1	*	0.4158	0.3627	0.2215	1	-
4	C9: USA	0.5319	0.2311	0.2371	1	-	0.4078	0.4531	0.1391	2	-
5	C3: China	0.2731	0.3143	0.4126	3	-	0.2579	0.2707	0.4714	3	-
6	C4: Cuba	0.2235	0.2391	0.5374	3	-	0.0000	0.0000	1.0000	3	*
7	C10: USSR	0.0000	0.0000	1.0000	3	*	0.2346	0.2312	0.5342	3	-
8	C11: Yugoslavia	0.2819	0.2703	0.4478	3	-	0.2969	0.2875	0.4156	3	-
9	C2: Brazil	0.3419	0.3761	0.2820	2	-	0.3613	0.3506	0.2880	1	-
10	C5: Egypt	0.3444	0.3687	0.2870	2	-	0.3558	0.3493	0.2948	1	-
11	C7: India	0.0000	1.0000	0.0000	2	*	0.3257	0.3257	0.3485	3	-
12	C12: Zaire	0.3099	0.3959	0.2942	2	-	0.3901	0.3321	0.2778	1	-

data sets, since the single prototype seems not enough to capture a cluster in these cases, whereas the cluster can be properly characterized by the multiple prototypes as done in wECMdd. From the tables we can see that the EP values for credal partitions by sECMdd and wECMdd are significantly higher than those for hard or fuzzy partitions, which indicates the accuracy of specific decisions. Consequently it will avoid the risk of misclassification by the concept of imprecise decisions.

The value of ER describes the fraction of instances grouped into an identical specific cluster out of those relevant pairs in the ground-truth. If the objects are located in the overlap, they are likely to be clustered into imprecise classes by ECMdd. This will increase the value of EP. However, as few objects are partitioned into specific classes, the value of ER will decrease. That's why for Iris data set the partitional result by wECMdd has the highest EP value following with a low ER value. The value of ERI can be regarded as a compromise between EP and ER, and it is an integration of EP and

Table 5.12: Clustering results of sECMdd for countries data. The prototype (medoid) of each class is marked with *. The Label {1, 2} represents the imprecise class expressing the uncertainty on class 1 and class 2.

		sECMdd with Δ_1					sECMdd with Δ_2				
	Countries	$BetP_{i1}$	$BetP_{i2}$	$BetP_{i3}$	Label	Medoids	$BetP_{i1}$	$BetP_{i2}$	$BetP_{i3}$	Label	Medoids
1	C1: Belgium	1.0000	0.0000	0.0000	1	*	1.0000	0.0000	0.0000	1	*
2	C6: France	0.4932	0.2633	0.2435	1	-	0.5149	0.2555	0.2297	1	-
3	C8: Israel	0.4144	0.3119	0.2738	1	-	0.4231	0.3051	0.2719	1	-
4	C9: USA	0.4503	0.2994	0.2503	1	-	0.4684	0.2920	0.2396	1	-
5	C3: China	0.2323	0.2294	0.5383	3	*	0.0000	0.0000	1.0000	3	*
6	C4: Cuba	0.2778	0.2636	0.4586	3	-	0.2899	0.2794	0.4307	3	-
7	C10: USSR	0.2509	0.2260	0.5231	3	-	0.3167	0.2849	0.3984	3	-
8	C11: Yugoslavia	0.3478	0.2488	0.4034	3	-	0.3579	0.2526	0.3895	3	-
9	C2: Brazil	0.0000	1.0000	0.0000	2	*	0.0000	1.0000	0.0000	2	*
10	C5: Egypt	0.3755	0.3686	0.2558	{1, 2}	-	0.3845	0.3777	0.2378	{1, 2}	-
11	C7: India	0.3125	0.3650	0.3226	2	-	0.2787	0.3740	0.3473	2	-
12	C12: Zaire	0.3081	0.4336	0.2583	2	-	0.3068	0.4312	0.2619	2	-

Table 5.13: A summary of eight UCI data sets.

Data set	No. of objects	No. of cluster	Category
Iris	150	3	object data
Cat cortex	65	4	relational data
Protein	213	4	relational data
American football	115	12	graph data
Banknote	1372	2	object data
Segment	2100	19	object data
Digits	1797	10	object data
Yeast	1484	10	object data

ER. As can be seen from the results, ECMdd performs best in terms of ERI for most of the data sets. In practice, one can adjust the value of parameter α to get partitions with different definition. The elapsed time for every clustering algorithm is illustrated in the last column of each table. In terms of computational time, as excepted, the evidential clustering algorithms take more time than hard or fuzzy clustering. But sECMdd and wECMdd are much faster than MECM. wECMdd is less time-consuming than sECMdd.

Presented results allow us to sum up the characteristics of the proposed ECMdd clustering approaches (including sECMdd and wECMdd). Firstly, credal partitions provided by all the ECMdd algorithms could recover the information of crisp and fuzzy partitions. Secondly, ECMdd is more robust to the outliers and the initialization than FCMdd. Thirdly, the imprecise classes by credal partitions enable us to make soft decisions for uncertain objects and could avoid the risk of misclassifications. Moreover, wECMdd performs best generally due to the efficient class representativeness strategy.

Table 5.14: The clustering results on Iris data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.8077	**0.8571**	0.8859	0.8077	0.8571	0.8859	0.0140
FCMdd	0.7965	0.8520	0.8797	0.7965	0.8520	0.8797	0.0160
FMMdd	0.8329	0.8411	0.8923	0.8329	**0.8411**	0.8923	0.0560
MECM	0.8347	0.8384	0.8923	0.9454	0.7064	0.8900	73.3300
sECMdd	**0.8359**	0.8471	**0.8950**	0.9347	0.7328	**0.8953**	0.2500
wECMdd	0.8305	0.8335	0.8893	**0.9742**	0.4827	0.8257	0.2000

Table 5.15: The clustering results on Proteins data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.7023	0.8246	0.8492	0.7023	0.8246	0.8492	0.0230
FCMdd	0.6405	0.8353	0.8181	0.6405	**0.8353**	0.8181	0.0200
FMMdd	0.6586	0.7735	0.8198	0.6586	0.7735	0.8198	0.1760
MECM	0.6734	0.8250	0.8348	0.8530	0.5946	0.8542	220.7700
sECMdd	0.6534	0.8150	0.7848	**0.8630**	0.5146	0.8642	0.8100
wECMdd	**0.7449**	**0.8594**	**0.8751**	0.8609	0.7527	**0.8940**	0.4700

Table 5.16: The clustering results on Cats data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.6883	0.6897	0.8438	0.6883	0.6897	0.8438	0.0040
FCMdd	0.6036	0.5747	0.7986	0.6036	0.5747	0.7986	0.0220
FMMdd	0.4706	0.6130	0.7298	0.4706	0.6130	0.7298	0.0090
MECM	0.7269	0.7088	0.8601	0.9412	0.3065	0.8212	8.8000
sECMdd	0.7569	0.7288	0.8801	**0.9512**	0.2865	0.8312	0.1700
wECMdd	**0.8526**	**0.8755**	**0.9308**	0.8774	**0.8908**	**0.9413**	0.1400

Table 5.17: The clustering results on American football network.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.8649	0.9178	0.9820	0.8649	**0.9178**	0.9820	0.0430
FCMdd	0.8649	0.9178	0.9820	0.8649	0.9178	0.9820	0.0200
FMMdd	0.8590	0.9082	0.9808	0.8590	0.9082	0.9808	0.0710
MECM	0.8232	0.9082	0.9771	0.9303	0.8681	**0.9843**	154.9300
sECMdd	0.4166	0.6826	0.8984	0.7696	0.3384	0.9391	19.4700
wECMdd	**0.8924**	**0.9197**	**0.9847**	**0.9735**	0.5621	0.9638	18.2100

Table 5.18: The clustering results on Banknote authentication data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.5252	0.5851	0.5226	0.5252	0.5851	0.5226	0.7561
FCMdd	0.5252	0.5851	0.5226	0.5252	**0.5851**	0.5226	0.8350
FMMdd	0.5225	0.5302	0.5173	0.5225	0.5302	0.5173	5.9381
MECM	0.5201	0.5618	0.5265	0.5553	0.4078	0.5353	50.0890
sECMdd	0.5211	**0.6334**	0.5202	0.5191	0.5256	0.5138	8.2880
wECMdd	**0.5259**	0.5645	**0.5793**	**0.5713**	0.4808	**0.5797**	7.1500

Table 5.19: The clustering results on Segment data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.4131	0.4910	0.8281	0.4131	0.4910	0.8281	7.8250
FCMdd	0.4380	0.5683	0.8246	0.4380	0.5683	0.8346	8.9900
FMMdd	0.5186	0.8043	0.5626	0.5186	**0.8043**	0.5626	107.3040
MECM	0.5164	0.7744	0.6160	0.6764	0.5444	0.7160	765.8800
sECMdd	0.5040	0.7738	0.6065	0.7040	0.4738	0.7255	351.0800
wECMdd	**0.5433**	**0.8350**	**0.8455**	**0.7584**	0.4856	**0.8582**	308.3100

Table 5.20: The clustering results on Digits data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.5928	0.6351	0.8203	0.5928	0.6351	**0.8203**	6.3638
FCMdd	0.5096	0.5753	0.8026	0.5096	0.5753	0.8026	4.1913
FMMdd	0.6542	0.5941	0.7861	0.6542	0.5941	0.7861	25.7530
MECM	0.6148	0.5685	0.7772	0.8137	0.7268	0.6126	524.2380
sECMdd	0.7201	0.5920	0.7566	0.8048	**0.7630**	0.6005	215.5220
wECMdd	**0.7250**	**0.6645**	**0.8232**	**0.8211**	0.5911	0.8141	206.5590

Table 5.21: The clustering results on Yeast data set.

	P	R	RI	EP	ER	ERI	Elapsed Time (s)
PAM	0.5229	0.4848	0.7322	0.5229	0.4848	0.7322	4.6414
FCMdd	0.5939	0.5151	0.7515	0.5939	0.5151	0.7515	4.7177
FMMdd	0.5938	**0.5568**	0.6345	0.5938	0.5568	0.6345	12.7288
MECM	0.3991	0.4098	0.6829	0.5723	0.5601	0.7149	212.6400
sECMdd	0.4123	0.4698	0.7050	0.6393	0.5369	0.7273	155.5300
wECMdd	**0.6329**	0.5065	**0.7712**	**0.7041**	**0.6544**	**0.7917**	134.8950

Lastly, the prototype weights provided by wECMdd algorithms are useful for our better understanding of cluster structure in real applications.

5.6 Conclusion

In this chapter, the evidential c-medoids clustering is proposed as a new medoid-based clustering algorithm. Two versions of ECMdd algorithms are presented. One uses a single medoid to represent each class, while the other adopts the multiple weighted medoids. The proposed approaches are some extensions of crisp c-medoids and fuzzy c-medoids on the framework of belief function theory. The experimental results illustrates the advantages of credal partitions by sECMdd and wECMdd. Moreover, the way of using prototype weights to represent a cluster enables wECMdd to capture the various types of cluster structures more precisely and completely hence improves the quality of the detected classes. Furthermore, more detailed information on the discovered clusters may be obtained with the help of prototype weights.

As we analyzed in this chapter, assigning weights of a class to all the patterns seems not rational since objects in other clusters make little contribution. Thus it is better to set the number of possible objects holding positive weights differently for each class. But how to determine the optimal number of prototypes is a key problem and we will study this in our future work. The relational descriptions of a data set may be given by multiple dissimilarity matrices. Thus another interesting work aiming to obtain a collaborative role of the different dissimilarity matrices to get a final consensus partition will also be investigated in the future.

Until now, we have introduced three clustering algorithms for community detection. These models only take the dissimilarities or similarities between nodes based on the topological graph structure into consideration. In some cases, there may be some available information which is also of great value for guiding us preforming the task of graph clustering. In next chapter, we will discuss this problem and present a new clustering model which can make use of the available supervised knowledge.

6
Semi-supervised evidential label propagation algorithm for graphs

6.1 Introduction

With the increasing size of networks in real world, community detection approaches should be fast and accurate. The Label Propagation Algorithm (LPA) (Raghavan et al., 2007) is known to be one of the near-linear solutions and benefits of easy implementation, thus it forms a good basis for efficient community detection methods. The behavior of LPA is not stable because of the randomness. Different communities may be detected in different runs over the same network. Moreover, by assuming that a node always adopts the label of the majority of its neighbors, LPA ignores any other structural information existing in the neighborhood of this node.

Supervised classification is one of the most popular techniques in machine learning. Generally, the goal of supervised learning is to train a classifier that reliably approximates a classification task based on a set of labeled examples from the problem of interest. The performance of the learned classifier highly depends on the proportion of labeled samples. However, in many practical applications of pattern classification, it is usually difficult to get abundant labeled samples since the task of manual labeling is time consuming and often requires expensive human labor. On the contrary, there are usually a large number of unlabeled samples which are easier to obtain. Consequently, Semi-Supervised Learning (SSL), which aims to effectively combine the information from both unlabeled and labeled data, has been proposed to perform the classification task when there are not enough training samples.

Semi-supervised classification has been widely studied for classical data sets, but there has been little work on semi-supervised community detection. In many scenarios a substantial amount of prior knowledge about the graph structure may be available. It could reflect the application-specific knowledge about cluster membership to some extent. For instance, in a co-authorship community network, it may be possible to label a small subset of scholars based on their research interests. In a social network application, it may be desirable to label some nodes according to their affinity to some products.

In this chapter, we enhance the original LPA by introducing new update and propagation strategies using the theory of belief functions. The Semi-supervised version of

Evidential Label Propagation (SELP) algorithm is presented. SELP can take advantage of the limited amount of supervised information and consequently improve the detection results.

6.2 Semi-supervised label propagation

Inspired from LPA and EK-NNclus (Denœux et al., 2015), we propose here the SELP algorithm for graph data sets with prior information. The problem of semi-supervised community detection will be first described in a mathematical way, and then the proposed semi-supervised label propagation algorithm will be presented in detail.

6.2.1 Problem restatement and notions

Let $G(V, E)$ denote the graph, where V is the set of n nodes and $E \subseteq V \times V$ is the set of edges. Generally, a network can be expressed by its adjacent matrix $\boldsymbol{A} = (a_{ij})_{n\times n}$, where $a_{ij} = 1$ indicates that there is a direct edge between nodes n_i and n_j, and 0 otherwise.

Assume that there are c communities in the graph. The set of labels is denoted by $\Omega = \{\omega_1, \omega_2, \cdots, \omega_c\}$. In addition, in order to make sure the solution is unique, we assume that there must be at least one labeled vertex in each community. The n nodes in set V can be divided into two parts:

$$V_L = \{(n_1, y_1), (n_2, y_2), \cdots, (n_l, y_l)\}, \quad y_j \in \Omega$$

for the labeled nodes, and

$$V_U = \{n_{l+1}, n_{l+2}, \cdots, n_n\}$$

for the unlabeled ones. The main task of the semi-supervised community detection is to make models propagating the labels from nodes in V_L to those in V_U, and further determine the labels of those unlabeled vertices.

6.2.2 The dissimilarities between nodes

Like the smooth assumption in the semi-supervised graph-based learning methods (Zhu et al., 2005), here we assume that the more common neighbors the two nodes share, the larger probability that they belong to the same community. Thus in this work, the index considering the number of shared common neighbors is adopted to measure the similarities between nodes.

Definition 6.1. Let the set of neighbors of node n_i be N_i, and the degree of node n_i

be d_i. The similarity between nodes n_i and n_j ($n_i, n_j \in V$) is defined as

$$s_{ij} = \begin{cases} \dfrac{|N_i \cap N_j|}{d_i + d_j}, & \text{if } a_{ij} = 1 \\ 0, & \text{otherwise.} \end{cases} \tag{6.1}$$

Then the dissimilarities associated with the similarity measure can be defined as

$$d_{ij} = \frac{1 - s_{ij}}{s_{ij}}, \ \forall \ n_i, n_j \in V. \tag{6.2}$$

6.2.3 Label propagation

For the labeled node $n_j \in V_L$ in community ω_k, the initial bba can be defined as a Bayesian categorical mass function:

$$m^j(A) = \begin{cases} 1 & \text{if } A = \{\omega_k\} \\ 0 & \text{otherwise.} \end{cases} \tag{6.3}$$

For the unlabeled node $n_x \in V_U$, the vacuous mass assignment can be used to express our ignorance about its community label:

$$m^x(A) = \begin{cases} 1 & \text{if } A = \Omega \\ 0 & \text{otherwise.} \end{cases} \tag{6.4}$$

To determine the label of node n_x, its neighbors can be regarded as distinct information sources. If there are $|N_x| = r_x$ neighbors for node n_x, the number of sources is r_x. The reliability of each source depends on the similarities between nodes. Suppose that there is a neighbor n_t with label ω_j, it can provide us with a bba describing the belief on the community label of node n_x as

$$\begin{aligned} m_t^x(\{\omega_t\}) &= \alpha * m^t(\{\omega_j\}), \\ m_t^x(\Omega) &= m^t(\Omega) + (1 - \alpha) * m^t(\{\omega_j\}), \\ m_t^x(A) &= 0, \quad \text{if } A \neq \{\omega_j\}, \Omega, \end{aligned} \tag{6.5}$$

where α is the discounting parameter such that $0 \leq \alpha \leq 1$. It should be determined according to the similarity between nodes n_x and n_t. The more similar the two nodes are, the more reliable the source is. Thus α can be set as a decreasing function of d_{xt}. In this work we suggest to use

$$\alpha = \alpha_0 \exp\{-\gamma d_{xt}^{\beta}\}, \tag{6.6}$$

where parameters α_0 and β can be set to be 1 and 2 respectively as default, and γ can be set to

$$\gamma = 1/\text{median}\left(\left\{d_{ij}^{\beta},\ i = 1, 2, \cdots, n,\ j \in N_i\right\}\right). \tag{6.7}$$

After the r_x bbas from its neighbors are calculated using Eq. (6.5), the fused bba of node n_x can be got by the use of Dempster's combination rule (see Eq. (2.14)):

$$m^x = m_1^x \oplus m_2^x \oplus \cdots \oplus m_{r_x}^x. \tag{6.8}$$

The label of node n_x can be determined by the maximal value of m^x. The main principle of semi-supervised learning is to take advantage of the unlabeled data. It is an intuitive way to add node n_x (previously in set V_U but already be labeled now) to set V_l to train the classifier. However, if the predicted label of n_x is wrong, it will have very bad effects on the accuracy of the following predictions. Here a parameter η is introduced to control the prediction confidence of the nodes that to be added in V_l. If the maximum of m^x is larger than η, it indicates that the belief about the community of node n_x is high and the prediction is confident. Then we remove node n_x in V_U and add it to set V_L. On the contrary, if the maximum of m^x is not larger than η, it means that we can not make a confident decision about the label of n_x based on the current information. Thus the node n_x should be remained in set V_U. In fact this is the idea of self-training (Li and Zhou, 2005).

In order to propagate the labels from the labeled nodes to the unlabeled ones in the graph, a classifier should be first trained using the labeled data in V_l. For each node n_x in V_U, we find its direct neighbors and construct bbas through Eq. (6.5). Then the fused bba about the community label of node n_x is calculated by Eq. (6.8). The subset of the unlabeled nodes, the maximal bba of which is larger than the given threshold η, are selected to augment the labeled data set. The predicted labels of these nodes are set to be the class assigned with the maximal mass. Parameter η can be set to 0.7 by default in practice.

After the above update process, there may still be some nodes in V_U. For these nodes, we can find their neighbors that are in V_L, and then use Eqs. (6.5) and (6.8) to determine their bbas. The whole algorithm of SELP is summarised in Algorithm 6.

6.3 Experiment

In order to verify the efficiency and effectiveness of the proposed SELP algorithm, some experiments on classification tasks will be conducted in this section, and the results by the use of different methods will be reported. The semi-supervised community detection algorithm using label propagation (SLP) (Liu et al., 2014) and the unsupervised label propagation algorithm will be used to compare the performance on graph data sets. The parameters in SELP are all set to the default values in the experiments.

Algorithm 6 : SELP algorithm

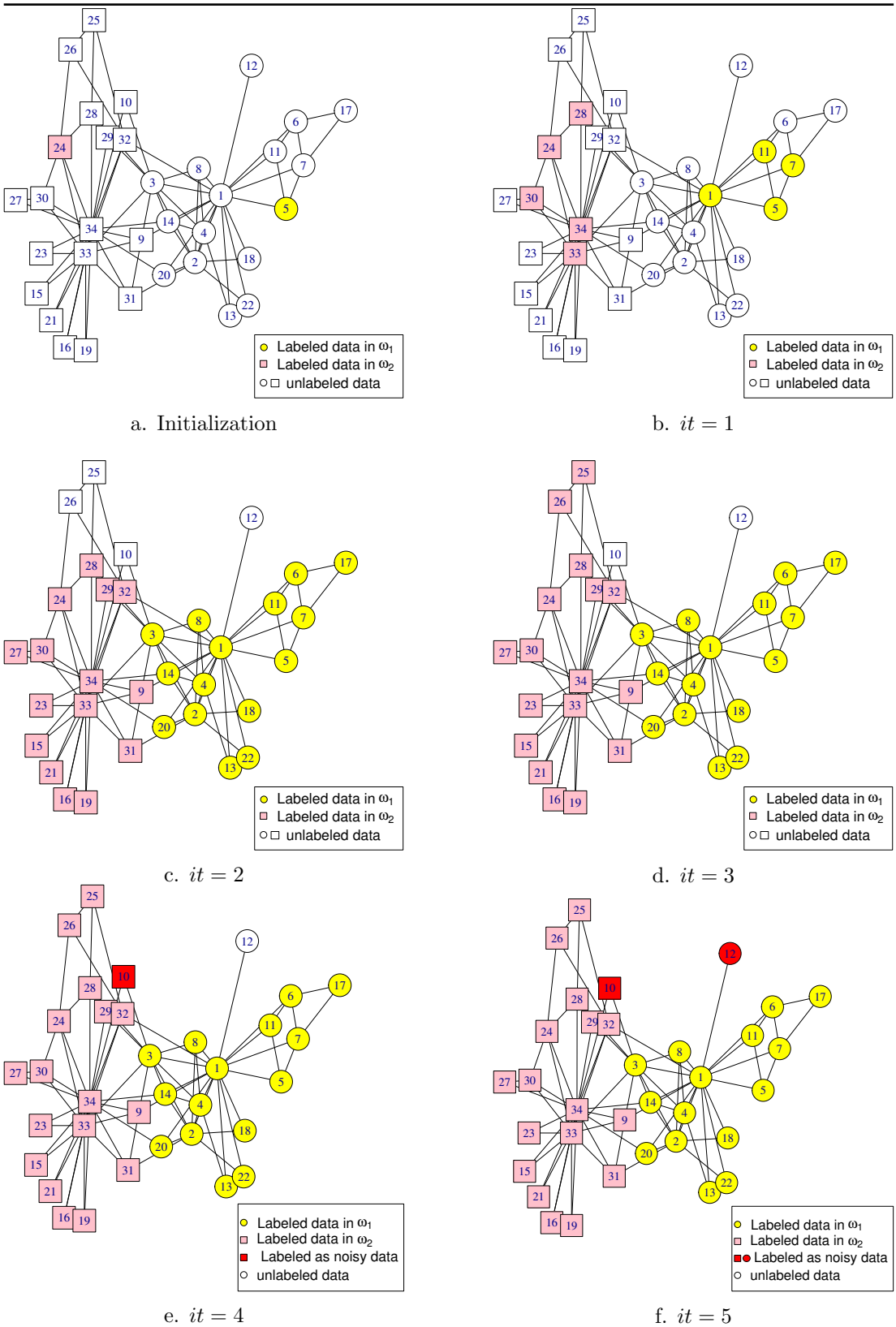
Input: Graph $G(V, E)$. The set of labeled node V_L, and the set of unlabeled node V_U.
Parameters:
η: the parameter to control the prediction confidence
α_0, β: the parameter to determine the discounting factor
$MaxIts$: the maximal update steps
$PercFul$: the percentage of the labeled data
Initialization:
(1). Initialize the bba of each node in the network using Eqs. (6.3) and (6.4).
(2). Let $it = 0$
repeat
(1). For each node $n_x \in V_U$, find its r_x direct neighbors and construct r_x bbas of n_x using Eq. (6.5).
(2). Calculate the fused bba of node n_x by Eq. (6.8).
(3). If the maximum of mass assignment of n_x is larger than η, move node n_X from set V_U to set V_L.
(4). $it = it + 1$.
until The percentage of nodes in V_L is larger than $PercFul$ or the maximal update step is reached.
If there are still some nodes in V_U, update their bbas based on the information from the neighbors using Eqs. (6.5) and (6.8).
Output: The bba matrix $\boldsymbol{M} = \{m^i\},\ i = 1, 2, \cdots, n$.

6.3.1 Real world networks

A. Karate Club network. In this experiment we test on the widely used benchmark in detecting community structures, "Karate Club".

In the first test, the labeled node in community ω_1 is set to node 5, while that in community ω_2 is node 24. After five steps, SELP algorithm stops. The detailed update process is displayed in Figure 6.1. It can be seen from the figure that two outliers, nodes 10 and 12 are detected by SELP. From the original graph, we can see that node 10 has two neighbors, nodes 3 and 34. But neither of them shares a common neighbor with node 10. For node 12, it only connects to node 1, but has no connection with any other node in the graph. Therefore, it is very intuitive that the two nodes are regarded as outliers of the graph.

The detection results on Karate Club network by SELP and SLP algorithms with different labeled nodes are shown in Table 6.1. The labeled vertices and its corresponding misclassified vertices are clearly presented in the table. As can be seen from the table, nodes 10 and 12 are detected as outliers in all the cases by SELP, and the two communities can be correctly classified most of the time. The performance of SLP is



a. Initialization

b. $it = 1$

c. $it = 2$

d. $it = 3$

e. $it = 4$

f. $it = 5$

Figure 6.1: The label propagation process on Karate Club network. The nodes marked with color red are the outliers detected by SELP.

worse than that of SELP when there is only one labeled data in each community. For the nodes which are connected to both communities and located in the overlap, such as nodes 3 and 9, they are misclassified most frequently. If the number of labeled data in each community is increased to 2, the exact community structure can be got by both methods. It is indicated that the more prior information we have, the better the performance of SELP is.

Table 6.1: Community detection results for the Karate Club network.

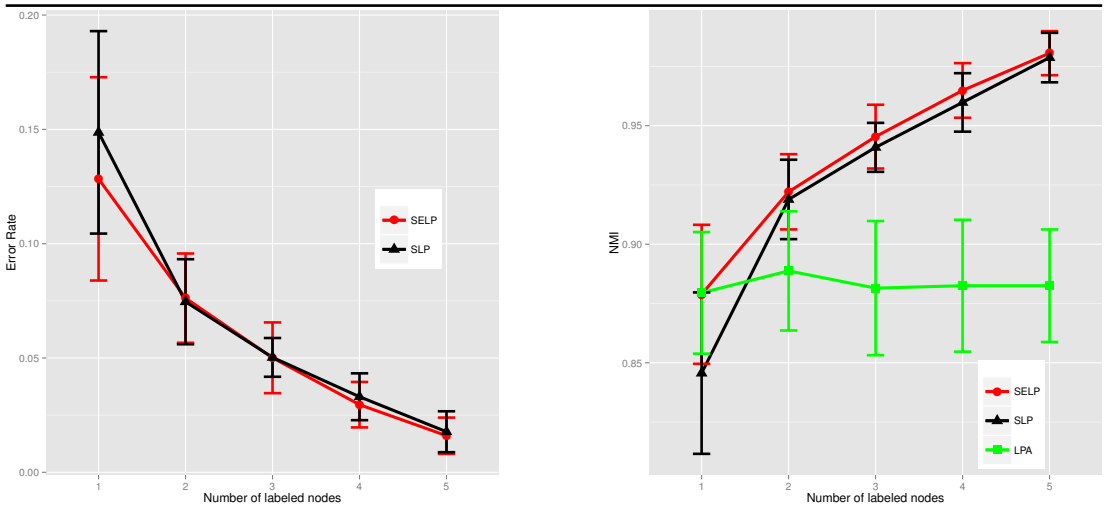
Labeled nodes in ω_1	Labeled nodes in ω_2	Misclassified nodes by SELP	Detected outliers by SELP	Misclassified nodes by SLP
1	34	None	10, 12	None
1	32	9	10, 12	9, 10, 27, 31, 34
2	33	None	10, 12	None
6	31	3	10, 12	2, 3, 8, 14, 2
8	31	None	10, 12	10
8	32	None	10, 12	None
17	31	3, 4, 8, 14	10, 12	2, 3, 4, 8, 13, 14, 18, 20, 22
1, 2	33, 34	None	10, 12	None
1, 2	33, 9	None	10, 12	None
3, 18	26, 30	None	10, 12	None
17, 4	31, 9	None	10, 12	None

B. American football network. As a further test of our algorithm, the network we investigate in this experiment is the world of American college football games.

Let the number of labeled nodes in each community to be fixed. Then SELP and SLP algorithms are evoked 50 times respectively with randomly selected labeled nodes. The average error rates and NMI values (plus and minus one standard deviation) of the 50 experiments are displayed in Figures 6.2-a and 6.2-b respectively. As can be seen from the figures, with the increasing number of labeled samples, the performance of both SELP and SLP becomes better. The NMI values of the detected communities by SELP and SLP are significantly larger than those by LPA. It indicates that the semi-supervised community detection methods could take advantage of the limited amount of prior information and consequently improve the accuracy of the detection results. The behavior of SELP is better than that of SLP in terms of both error rates and NMI values.

6.3.2 LFR network

In this subsection, LFR benchmark networks will be used to test the ability of the algorithm to identify communities. The experiments here include evaluating the performance of the algorithm with various amounts of labeled nodes and different values of parameter μ in the benchmark networks. The original LPA (Raghavan et al., 2007) and the semi-supervised community detection approach SLP (Liu et al., 2014) will be used to compare.



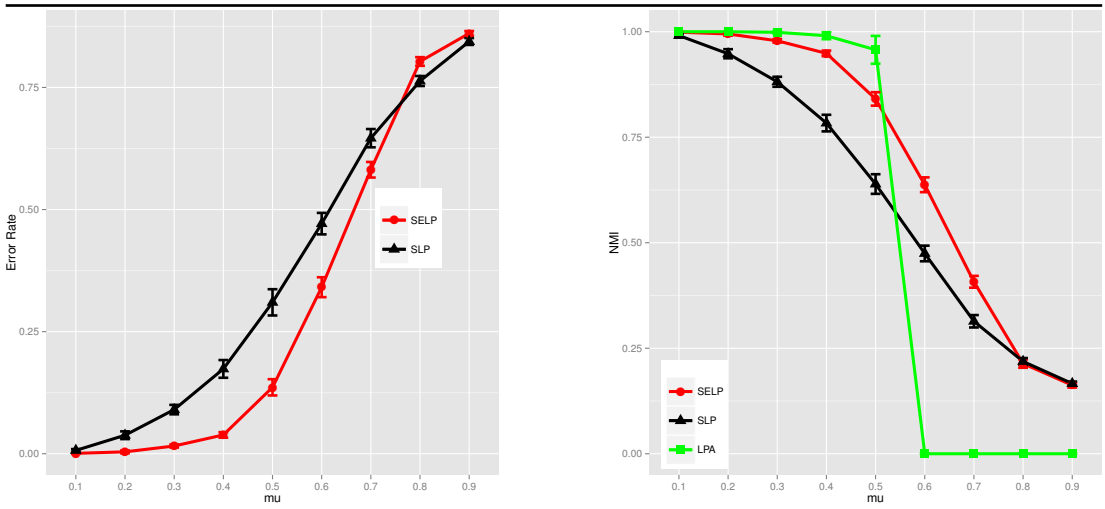
a. Error rate b. NMI

Figure 6.2: The results on American football network. The two figures show the average error rates and NMI values (plus and minus one standard deviation) for 50 repeated experiments, as a function of the number of labeled samples.

As mentioned before, in LFR networks, the mixing parameter μ represents the ratio between the external degree of each vertex with respect to its community and the total degree of the node. The larger the value of μ is, the more difficult the community structure will be correctly detected. The values of the parameters in LFR benchmark networks are set as follows: $n = 1000, \xi = 15, \tau_1 = 2, \tau_2 = 1, cmin = 20, cmax = 50$.

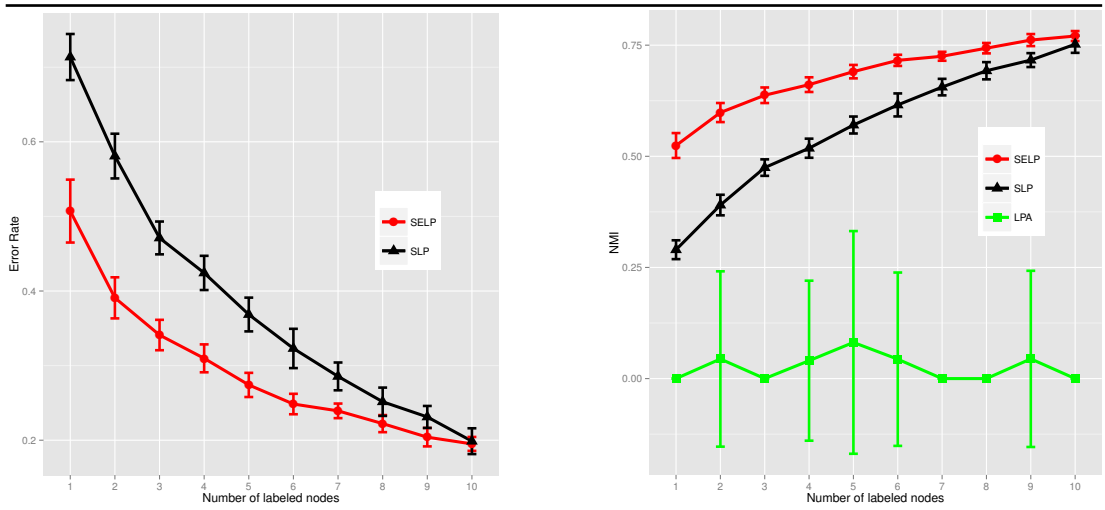
The performance of different methods with various values of μ is shown in Figure 6.3. As expected, the error rate is very high and the NMI value is low when μ is large. It demonstrates the fact that the community structure is not very clear and consequently difficult to be identified correctly. It can be seen from Figure 6.3-a that the error rates by SELP are smaller than those by SLP generally. SELP performs better than SLP. This conclusion could also be got in terms of NMI values displayed in Figure 6.3-b.

The original LPA could not work at all when μ is larger than 0.5. The results of SELP and SLP are significantly improved in these cases compared with LPA. As shown in Figure 6.4-b, even when there is only one labeled data in each community, the behavior of SELP is much better than that of LPA. This confirms the fact that the semi-supervised community detection approaches can effectively take advantage of the limited amount of labeled data. From Figure 6.4, we can also see that the performance of SELP and SLP becomes better with the increasing number of labeled nodes.



a. Error rate b. NMI

Figure 6.3: The results on LFR network. The number of labeled nodes in each community is 3.



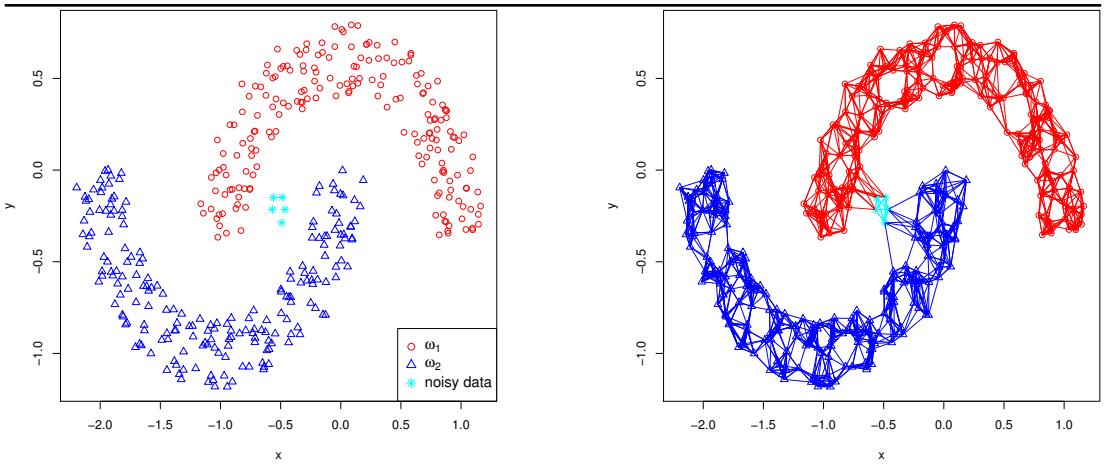
a. Error rate b. NMI

Figure 6.4: The results on LFR network. The parameter of μ is set to be 0.6.

6.3.3 Classical data sets

To apply the proposed SELP algorithm on classical data sets, we should first find an appropriate graph to model the analyzed data set. However, although the graph is at the heart of these graph-based methods, its construction has not been studied extensively (Zhu, 2006). Here a commonly used method, the K-Nearest Neighbor Graph (KNNG), is adopted to construct a graph based on the dissimilarities between objects (Maier et al., 2009). Assume that there are n objects, $x_1, x_2, \cdots, x_n$, in the data set. The dissimilarities between objects are denoted by $d_{ij}, i, j = 1, 2, \cdots, n$. The data points are used as vertices in the constructed undirected graph. As before, by N_j we denote the set of the K nearest neighbors of object x_i among $x_1, x_2, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n$. The KNNG can be defined as: x_i and x_j are connected if $x_i \in N_j$ and $x_j \in N_i$.

Example 6.1. In this experiment a simulated two-dimensional data set, often called "two-moon data set" (shown in Figure 6.5-a), is used. This data set consists of 405 objects which form two non-linearly separable semi-circle shaped clusters. There are five noisy points (marked with stars in the figure) that do not belong to either class. The constructed graph with $K = 9$ is displayed in Figure 6.5-b.

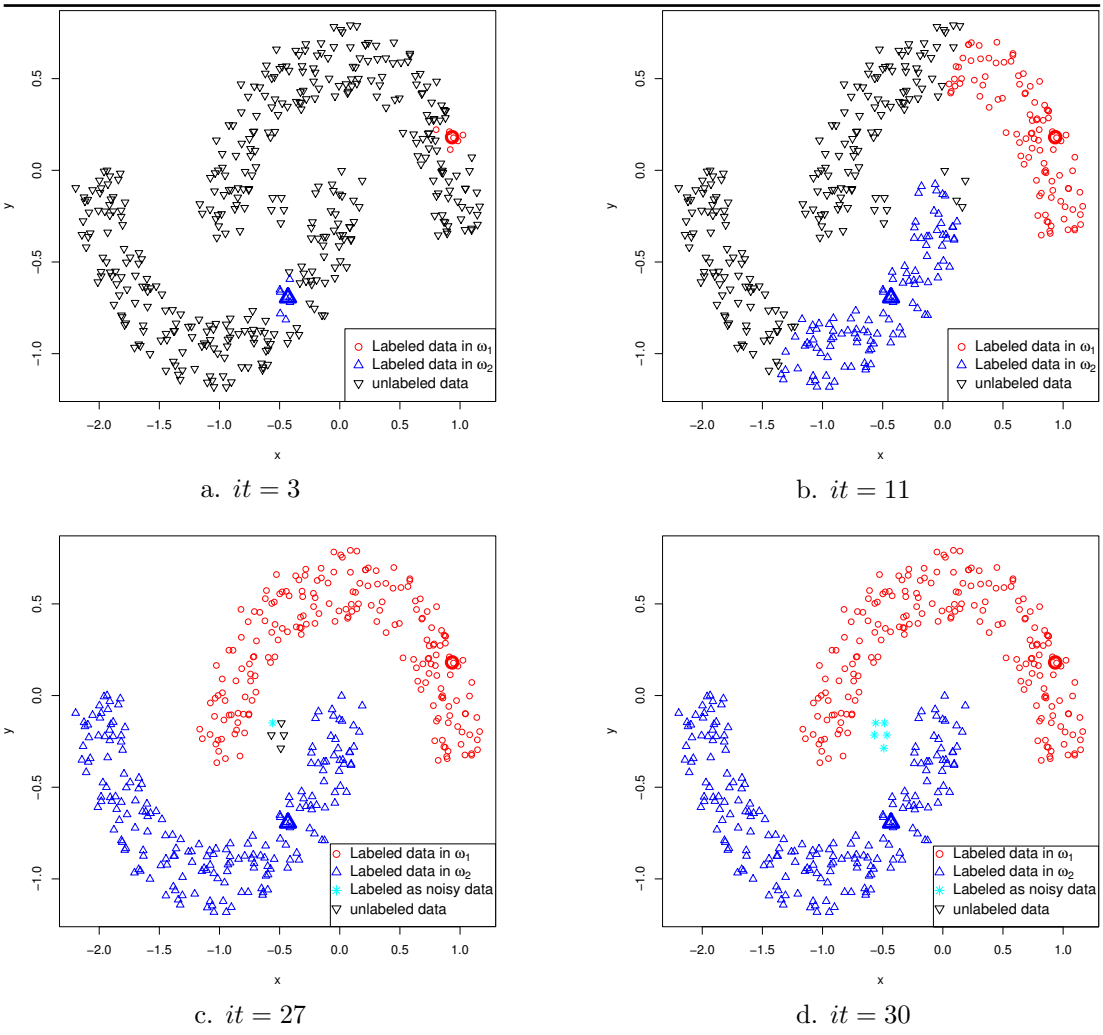


a. The original data set b. The constructed graph

Figure 6.5: The two-moon data set.

We evoke SELP algorithm on this graph. Initially we randomly select one sample from each class as the labeled data. The label propagation process is illustrated in Figure 6.6. As can be seen from the figure, the algorithm stops after 30 iterations. The two classes as well as the five noisy data are correctly classified finally. Using other semi-supervised learning algorithms such as SLP on the same constructed graph, the

two classes are easy to be detected. However, the noisy data will be partitioned into the two classes by mistake.

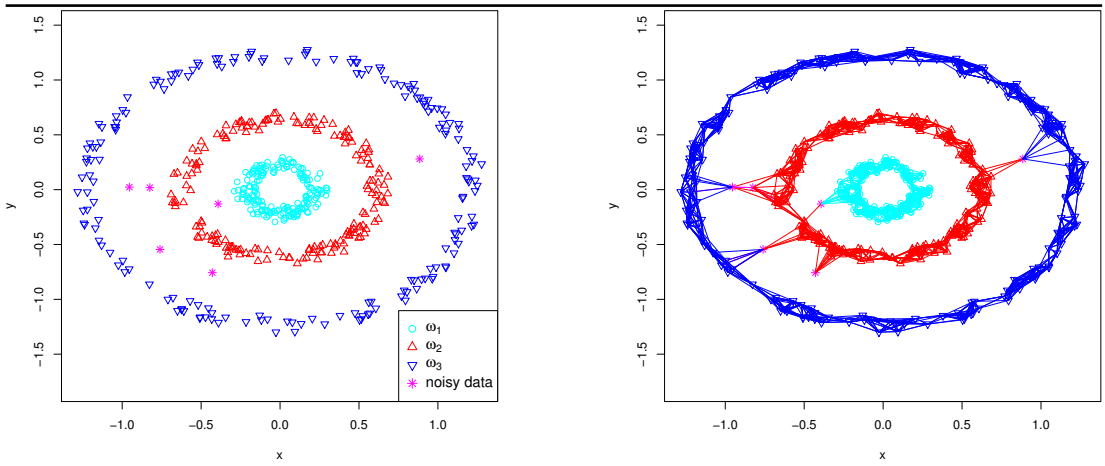


a. $it = 3$ b. $it = 11$

c. $it = 27$ d. $it = 30$

Figure 6.6: The label propagation process on two-moon data set. The initial labeled samples are marked with big size symbols in the figure.

Example 6.2. The original data set used in the example, which is a three-ring pattern with $180 \times 3 + 6$ data points, is shown in Figure 6.7-a. Each circle contains 180 patterns, and there are also six noisy points located between circles. Figure 6.7-b depicts the constructed K-NN graph with $K = 10$.



a. The original data set b. The constructed graph

Figure 6.7: The three-ring data set.

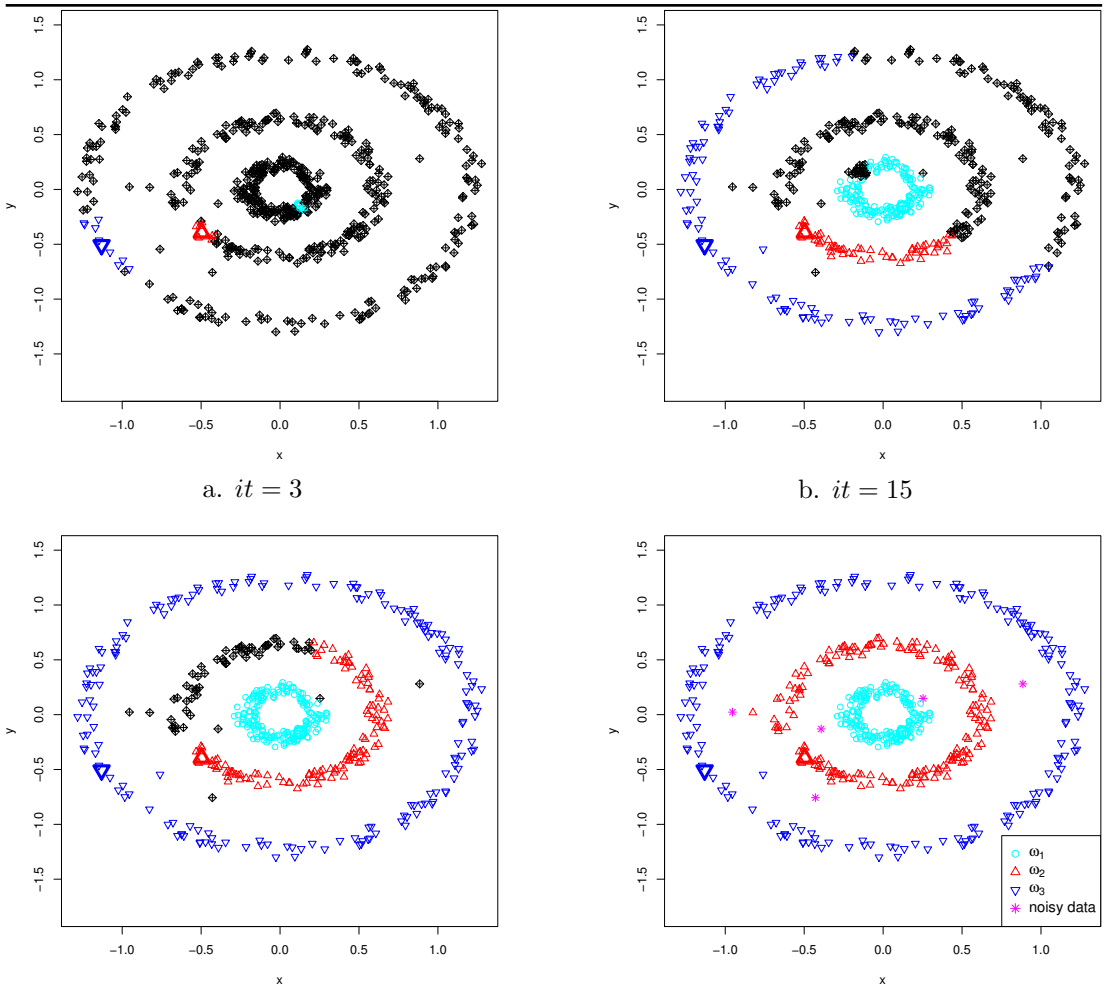
The update process and the classification results are illustrated in Figures 6.8-a – 6.8-d. From the figures we can see that SELP could detect the three classes exactly. Five outliers out of six are correctly found.

These two experiments on classical data sets are just used to show the possibility of the application of SELP on classical data sets. The results indicate that SELP works well especially it can detect the outliers. However, how to construct the graph using the dissimilarities between objects needs a further consideration.

6.4 Conclusion

In this chapter, the semi-supervised evidential label propagation algorithm is presented as an enhanced version of LPA. Different from the other chapters in this manuscript, the approach proposed here could effectively take advantage of the limited amount of supervised information. This is of practical meaning in real applications as there often exists some prior knowledge for the analyzed data sets. The experimental results show that the detection results will be significantly improved with the help of limited amount of supervised data.

At the end of the experimental part, we show the possibility to apply SELP on classical data sets. This requires the construction of graph based on the dissimilarities between objects. However, how to construct graphs should be further studied. Another problem, also can be seen from the experiments, is that the detection results are different if the labeled data are different. Thus which data should be selected as the



a. $it = 3$ b. $it = 15$

c. $it = 30$ d. $it = 47$

Figure 6.8: The label propagation process on three-ring data set. The initial labeled samples are marked with big size symbols in the figure. In the first three figures, the samples marked with black symbols denote the unlabeled data in the current step.

initial labeled samples should be considered. In fact this question is in the context of active learning. The active community detection approach using the principle of label propagation is also an interesting problem that is worth investigation in the future research work.

7 Conclusion and future work

7.1 Conclusion

Clustering is a key process in different data mining applications and it offers many potential benefits to a wide-range of fields including data mining, pattern recognition, information retrieval, bioinformatics, and business intelligence. Different requirements and challenges need to be taken into consideration in designing clustering algorithms applied to different tasks. In this thesis, we focused our study on clustering of proximity data. For community detection problems on networks, it could be seen as a graph clustering where the available data is the similarities or dissimilarities based on the topological graph structure.

With this objective in mind, we first consider a basic situation where only the pairwise relation between patterns is available, and three clustering methods, MECM, SMP, and ECMdd have been proposed. MECM uses only one prototype to describe the class, while SMP and ECMdd can take advantage of more than one representative prototype. The difference between SMP and ECMdd is that the former is for hard and fuzzy partitions, and the latter is for credal partitions. Compared with crisp and fuzzy partitions, credal partitions could also provide us with the information contained in hard and fuzzy partitions as by-products through computing pignistic probabilities. Thus it is more general than hard and fuzzy clustering. Moreover, it could distinguish ignorance from uncertainty while fuzzy or crisp partitions could not. By the introduced imprecise clusters, we could find some overlapping and indistinguishable clusters for related patterns. We also find that the way of using prototype weights to represent a cluster enables the clustering models to capture the various types of cluster structures more precisely and completely hence improves the quality of the detected classes. Furthermore, more detailed information on the discovered clusters can be got with the help of prototype weights. These clustering algorithms have been successfully applied in community detection problems.

SELP is different from the above three clustering methods since it can take some available prior information into consideration. In real applications of network data analysis, there often exists some prior information about the community labels of nodes. SELP can take use of such kind of supervised information to improve the detection results. SELP performs significantly better than the unsupervised community detection

approach. Compared with the existing semi-supervised community detection methods, SELP also has a better behavior.

To conclude, our work on evidential clustering aims to effectively make use of proximity data to improve the clustering results, provide interpretive information to describe each of the detected clusters, especially the imprecise and uncertain classes.

7.2 Future work

In this section, we try to develop some possible future works based on the current work that we have already done in this thesis. We are mostly interested in several directions as below:

- We will compare the four proposed methods on community detection problems, and discuss the principle of the selection of different methods under different conditions in real applications.

- In practice, labeled data are expensive to get. Active learning (Settles, 2010) aims to achieve high accuracy using few labeled data. There is little work on active learning for community detection problems. But it is of great value to introduce active learning in graph clustering approach since it could improve the performance of discovering community structure of complex networks. As there is often quite a large amount of uncertain information in networks, querying uncertain data during the learning procedure can reduce the uncertainty(Ma et al., 2016). It is interesting to study the strategy of selecting and querying the uncertain members in networks to improve the performance of learner.

- For credal partitions, the complexity of the algorithm is exponential to the number of clusters. This limits the application on the data sets when the class number is large. Although there are some method such as considering only a subclass with a limited number of focal sets (Masson and Denoeux, 2008), it is not efficient in practice. How to reduce the complexity of the algorithm should also be considered.

- The communities considered in this work can intuitively be defined as subsets of nodes in a graph with a dense structure in the corresponding subgraph. Only structural aspects are taken into account. Typically, no concise nor easily interpretable community description is provided in this case (Atzmueller et al., 2016). In order to provide both structurally valid and interpretable communities, the graph structure as well as additional descriptive features of the nodes in the graph should be utilized (Atzmueller et al., 2016). When nodes are described with a set of attributes, it is the concept of attributed networks (Largeron et al.,

2015). The attributes may be noisy and uncertain. It may be interesting to propose an evidential community detection approach to capture the imprecise graph structure with some uncertain information in an attributed graph.

Appendix

A. Update equations for the MECM algorithm

For minimizing J_{MECM}, the Lagrange multipliers method can be adopted. The calculation for the updating equations is similar to that in ECM (Masson and Denoeux, 2008). In the first step, the prototype set $\boldsymbol{V}$ is considered to be fixed. To solve the constrained minimization problem with respect to $\boldsymbol{M}$, the Lagrange multipliers λ_i can be introduced:

$$\mathcal{L}(\boldsymbol{M};\lambda_1,\lambda_2,\cdots,\lambda_n)=J_{\mathrm{MECM}}(\boldsymbol{M},\boldsymbol{V})-\sum_{i=1}^{n}\lambda_i\left(\sum_{A_j\subseteq\Omega,A_j\neq\emptyset}m_{ij}+m_{i\emptyset}-1\right). \quad \text{(A.1)}$$

By differentiating the Lagrangian with respect to the m_{ij}, $m_{i\emptyset}$ and λ_i and setting the derivatives to zero, we can obtain:

$$\frac{\partial L}{\partial m_{ij}}=|A_j|^{\alpha}\beta m_{ij}^{\beta-1}\overline{d}_{ij}^{2}-\lambda_i=0, \quad \text{(A.2)}$$

$$\frac{\partial L}{\partial m_{i\emptyset}}=\beta m_{ij}^{\beta-1}\delta^2-\lambda_i=0, \quad \text{(A.3)}$$

$$\frac{\partial L}{\partial \lambda_i}=\sum_{A_j\subseteq\Omega,A_j\neq\emptyset}m_{ij}+m_{i\emptyset}-1=0. \quad \text{(A.4)}$$

From Eq. (A.2) we can get

$$m_{ij}=\left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}}\left(\frac{1}{|A_j|^{\alpha}\overline{d}_{ij}^{2}}\right)^{\frac{1}{\beta-1}}. \quad \text{(A.5)}$$

And from Eq. (A.3)

$$m_{i\emptyset}=\left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}}\left(\frac{1}{\delta^2}\right)^{\frac{1}{\beta-1}}. \quad \text{(A.6)}$$

Using Eqs. (A.4)- (A.6) we have

$$\left(\frac{\lambda_i}{\beta}\right)^{\frac{1}{\beta-1}}=\left(\sum_j\frac{1}{|A_j|^{\frac{\alpha}{\beta-1}}}\frac{1}{\left(\overline{d}_{ij}\right)^{\frac{2}{\beta-1}}}+\frac{1}{\delta^{\frac{2}{\beta-1}}}\right)^{-1}. \quad \text{(A.7)}$$

Returning in Eq. (A.2), one obtains the necessary condition of optimality for $\boldsymbol{M}$:

$$m_{ij} = \frac{|A_j|^{-\alpha/(\beta-1)}\overline{d}_{ij}^{-2/(\beta-1)}}{\sum\limits_{A_k \neq \emptyset} |A_k|^{-\alpha/(\beta-1)}\overline{d}_{ik}^{-2/(\beta-1)} + \delta^{-2/(\beta-1)}}, \tag{A.8}$$

$$m_{i\emptyset} = 1 - \sum_{A_j \neq \emptyset} m_{ij}. \tag{A.9}$$

The update strategy for $\boldsymbol{V}$ is based on a global searching scheme, trough which the objective function J_{MECM} can also be decreased.

Author's Publication List

International Journals

(1) **Kuang Zhou**, Arnaud Martin, Quan Pan, and Zhun-ga Liu. ECMdd: Evidential c-medoids clustering with multiple prototypes. *Pattern Recognition*, 60:239-257, 2016.

(2) **Kuang Zhou**, Arnaud Martin, Quan Pan, and Zhun-ga Liu. Median evidential c-means algorithm and its application to community detection. *Knowledge-Based Systems*, 74:69–88, 2015.

(3) **Kuang Zhou**, Arnaud Martin, and Quan Pan. A similarity-based community detection method with multiple prototype representation. *Physica A: Statistical Mechanics and its Applications*, 438:519-531, 2015.

(4) **Kuang Zhou**, Arnaud Martin, and Quan Pan. The Belief-Noisy-OR model applied to network reliability analysis. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Accepted.

Conference Papers

(1) **Kuang Zhou**, Arnaud Martin, Quan Pan, and Zhun-ga Liu. E Evidential Label Propagation Algorithm for Graphs. In *Information Fusion (FUSION), 2016 19th International Conference on*, Heidelberg, Germany, July 2016.

(2) **Kuang Zhou**, Arnaud Martin, Quan Pan, and Zhun-ga Liu. Evidential relational clustering using medoids. In *Information Fusion (FUSION), 2015 18th International Conference on*, Washington, United States, July 2015.

(3) **Kuang Zhou**, Arnaud Martin, and Quan Pan. Evidential community detection using structural and attribute information. In *Atelier Réseaux Sociaux et Intelligence Artificielle*, Rennes, France, June 2015.

(4) **Kuang Zhou**, Arnaud Martin, and Quan Pan. Evidential communities for complex networks. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 557–566. Springer, 2014.

(5) **Kuang Zhou**, Arnaud Martin, and Quan Pan. Evidential-EM algorithm applied to progressively censored observations. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 180–189. Springer, 2014.

(6) Wiem Maalel, **Kuang Zhou**, Arnaud Martin, and Zied Elouedi. Belief hierarchical clustering. In *Belief Functions: Theory and Applications*, pages 68–76. Springer, 2014.

Bibliography

Aggarwal, C. C. and Reddy, C. K. (2013). *Data clustering: algorithms and applications*. CRC Press.

Amiri, B., Hossain, L., Crawford, J. W., and Wigand, R. T. (2013). Community detection in complex networks: Multi–objective enhanced firefly algorithm. *Knowledge-Based Systems*, 46:1–11.

Antoine, V., Quost, B., Masson, M.-H., and Denoeux, T. (2012). CECM: Constrained evidential c-means algorithm. *Computational Statistics & Data Analysis*, 56(4):894–914.

Antoine, V., Quost, B., Masson, M.-H., and Denoeux, T. (2014). CEVCLUS: Evidential clustering with instance-level constraints for relational data. *Soft Computing*, 18(7):1321–1335.

Arthur, D. and Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

Atzmueller, M., Doerfel, S., and Mitzlaff, F. (2016). Description-oriented community detection using exhaustive subgroup discovery. *Information Sciences*, 329:965–984.

Bezdek, J. C. (1981). *Pattern recognition with fuzzy objective function algorithms*. Kluwer Academic Publishers.

Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008.

Borgelt, C. (2006). *Prototype-based classification and clustering*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, Universitätsbibliothek.

Brandes, U., Delling, D., Gaertler, M., Görke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. (2006). On modularity – np-completeness and beyond.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1):107–117.

Cao, X., Wang, X., Jin, D., Cao, Y., and He, D. (2013). Identifying overlapping communities as well as hubs and outliers via nonnegative matrix factorization. *Scientific reports*, 3.

Chen, J., Zaïane, O. R., and Goebel, R. (2009). Detecting communities in social networks using max-min modularity. In *SDM*, volume 3, pages 20–24. SIAM.

Chen, Y.-S., Cheng, C.-H., Lai, C.-J., Hsu, C.-Y., and Syu, H.-J. (2012). Identifying patients in target customer segments using a two-stage clustering-classification approach: A hospital-based assessment. *Computers in biology and medicine*, 42(2):213–221.

Clauset, A., Newman, M. E., and Moore, C. (2004). Finding community structure in very large networks. *Physical review E*, 70(6):066111.

Clifton, A. and Lundquist, J. K. (2012). Data clustering reveals climate impacts on local wind phenomena. *Journal of Applied Meteorology and Climatology*, 51(8):1547–1557.

Cottrell, M., Hammer, B., Hasenfuß, A., and Villmann, T. (2006). Batch and median neural gas. *Neural Networks*, 19(6):762–771.

Dempster, A. P. (1967). Upper and lower probabilities induced by a multivalued mapping. *The annals of mathematical statistics*, pages 325–339.

Denoeux, T. (1995). A k-nearest neighbor classification rule based on dempster-shafer theory. *Systems, Man and Cybernetics, IEEE Transactions on*, 25(5):804–813.

Denœux, T., Kanjanatarakul, O., and Sriboonchitta, S. (2015). EK-NNclus: A clustering procedure based on the evidential k-nearest neighbor rule. *Knowledge-Based Systems*, 88:57–69.

Denœux, T. and Masson, M.-H. (2004). EVCLUS: evidential clustering of proximity data. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(1):95–109.

Dong, Y., Zhuang, Y., Chen, K., and Tai, X. (2006). A hierarchical clustering algorithm based on fuzzy graph connectedness. *Fuzzy Sets and Systems*, 157(13):1760–1774.

Dubey, S. R., Dixit, P., Singh, N., and Gupta, J. P. (2013). Infected fruit part detection using k-means clustering segmentation technique. *IJIMAI*, 2(2):65–72.

Dubois, D. and Prade, H. (1988). Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4(3):244–264.

Duch, J. and Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical review E*, 72(2):027104.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *J. Cybernet*, 3:32–57.

Eaton, E. and Mansbach, R. (2012). A spin-glass model for semi-supervised community detection. In *AAAI*. Citeseer.

Fan, Y., Li, M., Zhang, P., Wu, J., and Di, Z. (2007). Accuracy and precision of methods for community identification in weighted networks. *Physica A: Statistical Mechanics and its Applications*, 377(1):363–372.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, 7(2):179–188.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

Fortunato, S. and Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1):36–41.

Gabrys, B. and Bargiela, A. (2000). General fuzzy min-max neural network for clustering and classification. *Neural Networks, IEEE Transactions on*, 11(3):769–783.

Gao, C., Wei, D., Hu, Y., Mahadevan, S., and Deng, Y. (2013). A modified evidential methodology of identifying influential nodes in weighted networks. *Physica A: Statistical Mechanics and its Applications*, 392(21):5490–5500.

Gao, J., Liang, F., Fan, W., Wang, C., Sun, Y., and Han, J. (2010). On community outliers and their efficient detection in information networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 813–822. ACM.

Gao, Y., Qi, H., Liu, D., Li, J., and Li, L. (2014). A fuzzy relational clustering algorithm with q-weighted medoids. *Journal of Computational Information Systems*, 10(6):2389–2396.

Geweniger, T., Zülke, D., Hammer, B., and Villmann, T. (2010). Median fuzzy c-means for clustering dissimilarity data. *Neurocomputing*, 73(7):1109–1116.

Girvan, M. and Newman, M. E. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

Graepel, T., Herbrich, R., Bollmann-Sdorra, P., and Obermayer, K. (1999). Classification on pairwise proximity data. *Advances in neural information processing systems*, pages 438–444.

Grossman, D. A. and Frieder, O. (2012). *Information retrieval: Algorithms and heuristics*, volume 15. Springer Science & Business Media.

Guimera, R., Sales-Pardo, M., and Amaral, L. A. N. (2004). Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101.

Hariz, S. B., Elouedi, Z., and Mellouli, K. (2006). Clustering approach using belief function theory. In *Artificial Intelligence: Methodology, Systems, and Applications*, pages 162–171. Springer.

Hathaway, R. J. and Bezdek, J. C. (1994). Nerf *c*-means: Non-euclidean relational fuzzy clustering. *Pattern recognition*, 27(3):429–437.

Havens, T., Bezdek, J., Leckie, C., Ramamohanarao, K., and Palaniswami, M. (2013). A soft modularity function for detecting fuzzy communities in social networks. *Fuzzy Systems, IEEE Transactions on*, 21(6):1170–1175.

Hofmann, T. and Buhmann, J. M. (1997). Pairwise data clustering by deterministic annealing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(1):1–14.

Horng, Y.-J., Chen, S.-M., Chang, Y.-C., and Lee, C.-H. (2005). A new method for fuzzy information retrieval based on fuzzy hierarchical clustering and fuzzy inference techniques. *Fuzzy Systems, IEEE Transactions on*, 13(2):216–228.

Hu, Y., Li, M., Zhang, P., Fan, Y., and Di, Z. (2008). Community detection by signaling on complex networks. *Physical Review E*, 78(1):016115–1–8.

Jaccard, P. (1912). The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666.

Jiang, Y., Jia, C., and Yu, J. (2012). An efficient community detection method based on rank centrality. *Physica A: Statistical Mechanics and its Applications*.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons.

Krishnapuram, R., Joshi, A., Nasraoui, O., and Yi, L. (2001). Low-complexity fuzzy relational clustering algorithms for web mining. *Fuzzy Systems, IEEE Transactions on*, 9(4):595–607.

Krishnapuram, R. and Keller, J. M. (1993). A possibilistic approach to clustering. *Fuzzy Systems, IEEE Transactions on*, 1(2):98–110.

Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Physical Review E*, 78(4):046110.

Largeron, C., Mougel, P.-N., Rabbany, R., and Zaïane, O. R. (2015). Generating attributed networks with communities. *PloS one*, 10(4):e0122777.

Li, M. and Zhou, Z.-H. (2005). SETRED: Self-training with editing. In *Advances in Knowledge Discovery and Data Mining*, pages 611–621. Springer.

Lichman, M. (2013). UCI machine learning repository.

Lingras, P. and West, C. (2004). Interval set clustering of web users with rough k-means. *Journal of Intelligent Information Systems*, 23(1):5–16.

Liu, D., Bai, H.-Y., Li, H.-J., and Wang, W.-J. (2014). Semi-supervised community detection using label propagation. *International Journal of Modern Physics B*, 28(29):1450208.

Liu, J. and Liu, T. (2010). Detecting community structure in complex networks using simulated annealing with k-means algorithms. *Physica A: Statistical Mechanics and its Applications*, 389(11):2300–2309.

Liu, W. (2006). Measuring conflict between possibilistic uncertain information through belief function theory. In *Knowledge Science, Engineering and Management*, pages 265–277. Springer.

Liu, Z.-G., Dezert, J., Mercier, G., and Pan, Q. (2012). Belief c-means: An extension of fuzzy c-means algorithm in belief functions framework. *Pattern Recognition Letters*, 33(3):291–300.

Liu, Z.-g., Pan, Q., Dezert, J., and Mercier, G. (2015). Credal *c*-means clustering method based on belief functions. *Knowledge-Based Systems*, 74:119–132.

Ma, L., Destercke, S., and Wang, Y. (2016). Online active learning of decision trees with evidential data. *Pattern Recognition*, 52:33–45.

Ma, X., Gao, L., Yong, X., and Fu, L. (2010). Semi-supervised clustering algorithm for community structure detection in complex networks. *Physica A: Statistical Mechanics and its Applications*, 389(1):187–197.

Maalel, W., Zhou, K., Martin, A., and Elouedi, Z. (2014). Belief hierarchical clustering. In *Belief Functions: Theory and Applications*, pages 68–76. Springer.

Maier, M., Hein, M., and von Luxburg, U. (2009). Optimal construction of k-nearest-neighbor graphs for identifying noisy clusters. *Theoretical Computer Science*, 410(19):1749–1764.

Martin, A. and Quidu, I. (2008). Decision support with belief functions theory for seabed characterization. In *Information Fusion, 2008 11th International Conference on*, pages 1–8. IEEE.

Masson, M.-H. and Denœux, T. (2004). Clustering interval-valued proximity data using belief functions. *Pattern Recognition Letters*, 25(2):163–171.

Masson, M.-H. and Denoeux, T. (2008). ECM: An evidential version of the fuzzy *c*-means algorithm. *Pattern Recognition*, 41(4):1384–1397.

Masson, M.-H. and Denœux, T. (2009). RECM: Relational evidential *c*-means algorithm. *Pattern Recognition Letters*, 30(11):1015–1026.

Masson, M.-H. and Denoeux, T. (2011). Ensemble clustering in the belief functions framework. *International Journal of Approximate Reasoning*, 52(1):92–109.

Mei, J.-P. and Chen, L. (2010). Fuzzy clustering with weighted medoids for relational data. *Pattern Recognition*, 43(5):1964–1974.

Mei, J.-P. and Chen, L. (2011). Fuzzy relational clustering around medoids: A unified view. *Fuzzy Sets and Systems*, 183(1):44–56.

Mendes, M. and Sacks, L. (2003). Evaluating fuzzy clustering for relevance-based information access. In *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, volume 1, pages 648–653. IEEE.

Mitra, S. (2004). An evolutionary rough partitive clustering. *Pattern Recognition Letters*, 25(12):1439–1449.

Mitra, S., Banka, H., and Pedrycz, W. (2006). Rough-fuzzy collaborative clustering. *IEEE transactions on systems, man, and cybernetics. Part B, Cybernetics: a publication of the IEEE Systems, Man, and Cybernetics Society*, 36(4):795.

Mitra, S., Pedrycz, W., and Barman, B. (2010). Shadowed c-means: Integrating fuzzy and rough clustering. *Pattern Recognition*, 43(4):1282–1291.

Murtagh, F. and Contreras, P. (2012). Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97.

Nepusz, T., Petróczi, A., Négyessy, L., and Bazsó, F. (2008). Fuzzy communities and the concept of bridgeness in complex networks. *Physical Review E*, 77(1):016107.

Newman, M. E. (2004). Fast algorithm for detecting community structure in networks. *Physical review E*, 69(6):066133.

Newman, M. E. (2006a). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104.

Newman, M. E. (2006b). Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582.

Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.

Pal, N. R., Pal, K., Keller, J. M., and Bezdek, J. C. (2005). A possibilistic fuzzy *c*-means clustering algorithm. *Fuzzy Systems, IEEE Transactions on*, 13(4):517–530.

Pan, Y., Li, D.-H., Liu, J.-G., and Liang, J.-Z. (2010). Detecting community structure in complex networks via node similarity. *Physica A: Statistical Mechanics and its Applications*, 389(14):2849–2857.

Pawlak, Z. (1982). Rough sets. *International Journal of Computer & Information Sciences*, 11(5):341–356.

Peters, G. (2006). Some refinements of rough *k*-means clustering. *Pattern Recognition*, 39(8):1481–1491.

Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106.

Reichardt, J. and Bornholdt, S. (2006). Statistical mechanics of community detection. *Physical Review E*, 74(1):016110.

Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105(4):1118–1123.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 18(5):401–409.

Schubert, J. (2004). Clustering belief functions based on attracting and conflicting metalevel evidence using potts spin mean field theory. *Information Fusion*, 5(4):309–318.

Scripps, J., Tan, P.-N., and Esfahanian, A.-H. (2007). Exploration of link structure and community-based node roles in network analysis. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 649–654. IEEE.

Settles, B. (2010). Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11.

Shafer, G. (1976). *A mathematical theory of evidence*, volume 1. Princeton university press Princeton.

Smets, P. (1993). Belief functions: the disjunctive rule of combination and the generalized bayesian theorem. *International Journal of approximate reasoning*, 9(1):1–35.

Smets, P. (2005). Decision making in the tbm: the necessity of the pignistic transformation. *International Journal of Approximate Reasoning*, 38(2):133–147.

Smets, P. and Kennes, R. (1994). The transferable belief model. *Artificial intelligence*, 66(2):191–234.

Subbian, K., Aggarwal, C. C., Srivastava, J., and Philip, S. Y. (2013). Community detection with prior knowledge. In *SDM*, pages 405–413. SIAM.

Wang, F. and Zhang, C. (2008). Label propagation through linear neighborhoods. *Knowledge and Data Engineering, IEEE Transactions on*, 20(1):55–67.

White, S. and Smyth, P. (2005). A spectral clustering approach to finding communities in graph. In *SDM*, volume 5, pages 76–84. SIAM.

Xu, G., Tsoka, S., and Papageorgiou, L. G. (2007). Finding community structures in complex networks using mixed integer optimisation. *The European Physical Journal B-Condensed Matter and Complex Systems*, 60(2):231–239.

Zachary, W. W. (1977). An information flow model for conflict and fission in small groups. *Journal of anthropological research*, pages 452–473.

Zadeh, L. A. (1965). Fuzzy sets. *Information and control*, 8(3):338–353.

Zhang, S., Wang, R.-S., and Zhang, X.-S. (2007). Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Statistical Mechanics and its Applications*, 374(1):483–490.

Zhang, X. and Xu, Z. (2015). Hesitant fuzzy agglomerative hierarchical clustering algorithms. *International Journal of Systems Science*, 46(3):562–576.

Zhang, Z.-Y. (2013). Community structure detection in complex networks with partial background information. *EPL (Europhysics Letters)*, 101(4):48005.

Zhou, H. (2003). Distance, dissimilarity index, and network community structure. *Physical review E*, 67(6):061901.

Zhou, K., Martin, A., and Pan, Q. (2014). Evidential communities for complex networks. In *Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 557–566. Springer.

Zhou, K., Martin, A., and Pan, Q. (2015a). A similarity-based community detection method with multiple prototype representation. *Physica A: Statistical Mechanics and its Applications*, 438:519–531.

Zhou, K., Martin, A., Pan, Q., and Liu, Z.-g. (2015b). Evidential relational clustering using medoids. In *18th International Conference on Information Fusion*.

Zhou, K., Martin, A., Pan, Q., and Liu, Z.-g. (2015c). Median evidential c-means algorithm and its application to community detection. *Knowledge-Based Systems*, 74:69–88.

Zhou, T., Lü, L., and Zhang, Y.-C. (2009). Predicting missing links via local information. *The European Physical Journal B-Condensed Matter and Complex Systems*, 71(4):623–630.

Zhu, X. (2006). Semi-supervised learning literature survey. Technical report, Computer Sciences Technical Report 1530, University of Wisconsin, Madison.

Zhu, X. and Goldberg, A. B. (2009). Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130.

Zhu, X., Lafferty, J., and Rosenfeld, R. (2005). *Semi-supervised learning with graphs*. Carnegie Mellon University, language technologies institute, school of computer science.