Clustering and heterogeneous information fusion for social media theme discovery and associative mining

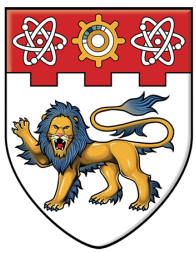
Meng, Lei

2014

Meng, L. (2014). Clustering and heterogeneous information fusion for social media theme discovery and associative mining. Doctoral thesis, Nanyang Technological University, Singapore.

https://hdl.handle.net/10356/62096

https://doi.org/10.32657/10356/62096



NANYANG
TECHNOLOGICAL
UNIVERSITY

Clustering and Heterogeneous Information Fusion for Social Media Theme Discovery and Associative Mining

A thesis
Submitted to the School of Computer Engineering
Nanyang Technological University

by

MENG LEI

in partial fulfilment of the requirement for the
degree of Doctor of Philosophy

August, 2014

Abstract

The emergence of social networking web sites has created numerous interactive sharing platforms for users to upload, comment, and share multimedia content online within their social circles. It has led to the massive number of web multimedia documents, together with their rich meta-information, such as category information, user tagging and description, and user comments. Such interconnected but heterogeneous social media data has provided opportunities for understanding traditional multimedia data, such as images and text documents. More importantly, the different types of activities and interactions of social users could be utilized to understand and analyze user behaviors, and discover social trends in social networks.

Clustering is an important approach to the analysis and mining of social media data. However, different from traditional multimedia data, the social media data are typically massive, diverse, heterogeneous and noisy. Those characteristics of social media data raise new challenges for existing clustering techniques, including the scalability to big data, the ability to automatically recognize the number of clusters in data sets, the strategies to effectively integrate data from heterogeneous resources for clustering, and the robustness to noisy features. Moreover, considering that different social users may have different preferences for categorizing the social media data, incorporating user preferences into the clustering framework to produce personalized data clusters is also a challenge.

In order to address the above issues, in this thesis, we investigate and develop novel clustering algorithms for the fast and robust clustering of large-scale social media data by integrating their multiple but different types of features and user preferences, and explore their applications to the associative social media mining tasks.

Towards this goal, we have completed four key tasks. First, we developed a two-step semi-supervised hierarchical clustering algorithm, termed Personalized Hierarchical

Theme-based Clustering (PHTC), for personalized web image organization by exploiting the surrounding text of web images. Our experiments have shown that PHTC can identify high quality clusters of web images under user supervision using the proposed semi-supervised clustering algorithm, called Probabilistic Fusion Adaptive Resonance Theory (PF-ART). In addition, it can order the clusters into a systematical hierarchy with a higher quality and lower time cost than several existing hierarchical clustering algorithms.

Secondly, we proposed a semi-supervised heterogeneous data co-clustering algorithm, termed Generalized Heterogeneous Fusion Adaptive Resonance Theory (GHF-ART), for multimedia data co-clustering by integrating different types of features from inter-related but heterogeneous data resources and user preferences. Compared with existing approaches, GHF-ART has the advantages of strong noise immunity, adaptive feature weighting, low computational cost, and incremental clustering in handling the dynamic social media data.

Thirdly, we investigated the feasibility of GHF-ART to clustering social network data for discovering user communities in heterogeneous social networks, and demonstrated its capability for analyzing the correlation among different social links and mining the potential themes of user communities.

Lastly, we studied the geometrical dynamics of Fuzzy ART and proposed three methods to adapt the vigilance parameter of Fuzzy ART. This leads to clustering algorithms insensitive to the input parameters for dealing with large and complex social media data. Our experiments have demonstrated the effectiveness of the proposed methods. Furthermore, the geometrical study of Fuzzy ART may also benefit further research.

While our completed studies has provided the base technologies for social media mining, the future directions for this thesis may focus on the following aspects: 1) Modeling of short and noisy text; 2) Automated selection of vigilance parameter in Fuzzy ART; 3) Improvement of clustering mechanism of ART; 4) Extension work on multimedia data indexing, annotation and retrieval; 5) Exploiting temporal factor for multimedia data storage and mining; and 6) Associative applications to social media mining tasks.

Keywords: Clustering, Semi-supervised clustering, Heterogeneous data co-clustering, Social media data mining, Social network analysis.

Acknowledgments

First and foremost, I wish to gratefully acknowledge my supervisor, Prof. Ah-Hwee Tan, who has supported me throughout this project with his patient guidance and invaluable advices. He has cultivated my research skills step by step, from research problem discovery, literature review, to writing skill improvement. Without his consistent support and supervision, my research would not have been so enriching and fulfilling. My thanks also go to my co-supervisor, Prof. Dong Xu, for his kind advices and support to my research work.

I would like to thank people who helped me in my research work. I deeply thank Prof. Chunyan Miao for her kindly help to my life in Singapore and inspirable suggestions in my starting research work. I thank Prof. Donald C. Wunsch II in Missouri University of Science and Technology for the generous sharing of his brilliant ideas to me and his careful and valuable comments on my research work.

I would like to thank Ms. Wenwen Wang and Ms. Yilin Kang, also the Ph.D. students of my supervisor, for their kindly help in assisting me to prepare research related materials and their prompt responses to my numerous inquiries. I also thank Mr. Kiong Wee Tan, the technician of our lab, for his patient help in fixing the problems of my computer and printing the posters for our projects.

I am really grateful to my family for their unconditional love and tremendous support. They care about my health, mood and daily life all the time and make me happy. I also want to express my thanks to my close friends, who stand firmly behind me and share my happiness and sorrow. Particularly, my special thanks go to Mr. Xi Chen, who has been my friend for 13 years. Great thanks to you for talking with me almost everyday that makes me happy and get rid of loneliness.

Last but not least, many thanks go to people in Lily Research Center for their kindly help in my daily life and nice attitude that makes me feel welcome.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Social networking applications, such as Flickr and Facebook, have transformed the World Wide Web into an interactive sharing platform, where users upload, comment and share media content within their social circles. Their popularity has led to an explosive growth of multimedia documents, together with their associated rich meta-information, such as category, keywords, user description, and comments. The availability of such a massive amount of interconnected heterogeneous social media data, on one hand, facilitates the semantic understanding of web multimedia documents, and the mining of the associations among heterogeneous data resources. On the other hand, social users are connected by plentiful types of interactions, which provide novel ways to analyze and understand user behaviors and social trends in social networks.

Clustering is a key and commonly used technique for knowledge discovery and mining from unstructured data resources. Given a collection of data, by representing the data objects into feature vectors, i.e. patterns, clustering is a process of identifying the natural groupings of the data patterns in the feature space according to their measured similarities, so that the data objects in the same cluster are more similar to each other than to those in other clusters. Thus, given a social network data repository with user records in terms of friendship, images, blogs, subscriptions and joint activity groups, clustering techniques could be utilized to analyze individual type of data, such as identifying the underlying categories of web images and discovering the hot topics from recent blogs of social users. Moreover, the multiple types of data could be treated as a whole

to discover groups of social users that have similar social behaviors, which can further benefit various applications, such as mining characteristics of different groups of users, detecting certain groups of users, and recommending friends and activity groups to the users having common interests.

However, different from traditional data sets, the social media data sets have several distinguishing characteristics. First, the social media data are usually large-scale, which may contain millions of data objects. Secondly, the social media data typically cover diverse content across a large number of topics. Thirdly, the social media data may involve data from heterogeneous resources. For example, the social network data of users may involve relational records, images and texts. Fourthly, in view that social users are free to upload data according to their minds, the social media data, especially the text data, typically involve much useless or noisy information, so that the obtained feature vectors of the data objects may be ill-featured noisy patterns. Those noisy features provide spurious relations between data patterns and may result in irregular or even overlapped shapes of data groups belonging to different classes in the feature space. Those characteristics of social media data raise new challenges for existing clustering techniques, including the scalability to big data, the ability to automatically recognize the number of clusters in a data set, the strategies to effectively integrate the data from heterogeneous resources for clustering, and the robustness to noisy features.

Besides, due to the individual life experience, different social users may have different preferences for organizing their social media data. Therefore, in this thesis, we also treat the user preferences as one type of descriptions or prior knowledge to the social media data, and explore feasible ways for integrating such user-provided information to guide the clustering process.

Considering the above issues in the context of social media data mining, we are motivated to develop novel clustering techniques from the following perspectives:

- The developed clustering algorithms should require low computer memory and computational cost to meet the scalability to big data;
- The developed clustering algorithms should be able to automatically identify the number of clusters in a data set, instead of using a pre-determined value, so as to reduce the sensitivity of clustering algorithms to the input parameter values;

- The developed clustering algorithms should be capable of effective understanding of composite data objects, which are represented by multiple types of data from heterogeneous resources;
- The developed clustering algorithms should have effective methods to identify the key features of patterns to alleviate the side-effect of noisy features;
- The developed clustering algorithms should be able to incorporate user preferences in order to generate personalized cluster structure for different social users.

1.2 Research Issues and Challenges

Taking into account the distinctive characteristics of the social media data, the task of clustering social media data encounters challenges in terms of six aspects, discussed as follows.

1.2.1 Information Representation

There are at least four common types of social media data, namely the relational data of social users, the uploaded images, the published articles, and the descriptive meta-information from social users. Note that videos are typically processed as a set of key frames/images, and in practice, the meta-information of videos, such as captions and comments, are much more effective than the video content in feature representation. The representation issue of each type of data is described as follows.

(i) **Relational Data:** The relational data illustrate the relations or similar behavior among social users, such as friendship network and co-subscription network. The feature representation of a data object of this type, i.e. a user, is usually by constructing a feature vector, of which the length equals the number of users and the elements are valued by the strength of interaction between the user and other users [1]. For example, given a data set of the friendship network of N users, the feature vector of the ith user can be denoted by $\mathbf{x}_i = [x_{i,1}, ..., x_{i,N}]$, where $x_{i,n} = 1$ if the nth user is a friend of the ith user and $x_{i,n} = 0$ otherwise. Similarly,

regarding the co-subscription network, the elements can be valued by the number of co-subscription.

The representation of relational data has a problem of requiring much computer memory to construct the relational matrix of users when the number of users is large. Additionally, the high dimensionality may also incur problems for clustering algorithms in learning the similarities between user patterns because of noisy features.

(ii) **Images:** The visual representation of image content is still a challenge nowadays. Current techniques for visual feature extraction [2, 3] are usually based on color histogram, edge detection, texture orientation and scale-invariant points, so that the visual features are inadequate to represent the images at the semantic level, a problem known as semantic gap. It leads to difficulties to group the images of the same class but with very different appearance or to distinguish those of different classes but with similar background.

(iii) **Articles:** The representation issue of text documents has been well studied in literature. Typically, articles are represented by the "Bag of Words" model, which is a feature vector containing all the keywords in the document collection. The selection of keywords is usually based on the occurrence frequencies of words or co-occurrence frequencies of groups of words, and the most commonly used algorithm to weight the selected keywords is based on the frequencies of the keywords in and cross the documents, known as term frequency-inverse document frequency (tf-idf). However, the web articles of social users typically have a large number of typos, personalized words and words that cannot reveal the semantics of the articles. Therefore, the representation of articles suffers from the issues of high dimensionality and noisy words.

(iv) **Meta-information:** In the context of social media data, the meta-information usually refers to the surrounding text of images and articles, such as titles and user comments, which provides additional knowledge to the data objects from other perspectives. However, the feature representation of meta-information meets two

problems. First, similar to the short text representation problem [4], the meta-information is usually very short so that the extracted tags cannot be effectively weighted by traditional statistical methods, such as tf-idf. Secondly, the meta-information typically involves several key tags that reveal the characteristics of the data objects, and much more noisy tags which are meaningless or even indicate incorrect relations between the data objects. Therefore, how to identify the key tags from a large number of noisy tags is also a problem for the feature construction of meta-information, which is also related to the tag ranking problem [5, 6] in the multimedia domain.

1.2.2 Scalability to Big Data

As aforementioned, social media data usually appear in a large scale. For example, the Google search engine has indexed billions of web documents, such as web pages and images. Besides, each search query to the search engine usually results in over ten millions of results. Therefore, the clustering techniques should be able to deal with a big data set with a reasonable running time.

Existing clustering techniques are usually based on K-means clustering, hierarchical clustering, spectral clustering, probabilistic clustering, density-based clustering and matrix factorization algorithms. However, most of them incur heavy mathematical computation. For example, hierarchical clustering and spectral clustering algorithms usually have a cubic time complexity of $O(n^3)$, and density-based clustering algorithms usually has a quadratic time complexity of $O(n^2)$, where n is the number of data patterns. Although K-mean clustering and matrix factorization algorithms have a linear time complexity of $O(n)$ with respect to the size of the data set, their computational cost also linearly increases with respect to the settings of the number of clusters and the number of iterations.

Recent studies for clustering large-scale social media data, especially for the social network data [7], explore methods for simplifying the data structure to achieve approximate results or parallel computation. However, the first approach usually requires assumptions to the data so as to reduce weak relations among data patterns, and the second approach needs one or more high-performance computers. Therefore, developing efficient

clustering algorithms or effective methods to accelerate existing clustering algorithms is necessary for clustering social media data.

1.2.3 Robustness to Noisy Features

As aforementioned in Section 1.2.1, the social media data usually suffer from the representation issues due to the large amount of useless or noisy features, making the produced patterns to have a high dimensionality and irregular shapes of clusters in the high-dimensional feature space.

Most of existing clustering algorithms, as discussed in Section 1.2.2, do not consider the problem of noisy features. As such, they may make incorrect correlation evaluation of patterns when calculating the similarities between patterns or doing mathematical mappings to investigate the characteristics of patterns.

Under such situation, the clustering algorithms for social media data are required to be capable of learning to identify the key features of patterns in order to alleviate the side-effect of noisy features.

1.2.4 Heterogeneous Information Fusion

The rich but heterogeneous social media data provide multiple descriptions of the data objects. However, a new challenge arises for traditional clustering algorithms on simultaneously integrating multiple but different types of data for clustering.

In recent years, many heterogeneous data co-clustering algorithms [8–12] have been proposed to extend traditional clustering algorithms to be capable of evaluating the similarity of data objects in and across different types of feature data. However, most of them perform heterogeneous data fusion by simply combining the objective functions of individual type of features. In view that different types of features have their own meanings and levels of feature values, this approach actually provides different weights to different type of features when achieving the global optimization. Although some of the algorithms consider the weighting problem of features, they usually use equal or empirical weights for different types of features. Therefore, developing effective weighting algorithms for the fusion of heterogeneous features remains a challenge.

1.2.5 Sensitivity to Input Parameters

Existing clustering algorithms typically require settings of one or more input parameters by the users, in most cases the number of clusters in the data set. Those pre-determined parameters may significantly affect the performance of clustering algorithms but usually vary in terms of different data sets, making it difficult for the users to empirically choose suitable values for them.

Although the parameter selection for clustering algorithms [13–15], especially the number of clusters in a data set, has been studied in a large body of literature, existing works are usually based on experimentally evaluating the quality of clusters, such as the intra-cluster and between-cluster distance, generated under different parameter settings of the same clustering algorithms.

In view that the social media data are typically large-scale and involve a diverse range of topics, it is not desired to enumerate the values of the input parameters to identify the fittest ones, which is time consuming and may not be accurate. Therefore, the parameter selection for specific clustering algorithms is still an open problem.

1.2.6 Ability to Incorporate User Preferences

Clustering is an automated process of discovering groups of patterns purely based on their distance evaluation in the feature space. Therefore, users have no control on the clustering results. As different users may have different preferences to organize the data, the discovered information sometimes may not match the user's requirements.

Semi-supervised clustering is an approach of incorporating user-provided information as prior knowledge to guide the clustering approach. However, existing algorithms [16–18] typically require users to specify the relations of pairs of patterns, such as whether two patterns should or should not be in the same cluster. Those relations are subsequently used as constraints in order to enhance the clustering accuracy. But such user-provided knowledge are usually very implicit in the resulting clusters.

Therefore, different methods for receiving and incorporating the user preferences into clustering algorithms are expected to be exploited to guide the clustering process, in order to not only enhance the clustering performance, but also make the clustering algorithms be capable of discovering interesting clusters for the users.

1.3 Approach and Methodology

In this thesis, we investigate and develop clustering algorithms for analyzing social media data based on Adaptive Resonance Theory (ART) and Fusion Adaptive Resonance Theory (Fusion ART).

ART [19] is a neural theory on how a human brain captures, recognizes and memorizes information about objects and events, and has led to the development of a family of clustering models. The ART-based clustering algorithms perform unsupervised learning by modeling clusters as memory prototypes and incrementally encoding the input patterns one at a time, through a real-time searching and matching mechanism. More importantly, they do not require a pre-determined number of clusters. Instead, a user-input parameter, the vigilance parameter, is used to control to which degree an input pattern can be determined similar to a selected cluster. In this way, the clusters can be automatically identified by incrementally generating new clusters to encode novel patterns that are determined dissimilar to existing clusters.

Fusion ART [20] extends ART from single input feature channel to multiple ones, and serves as a general architecture for the simultaneously learning from multi-modal feature mappings. Beside the advantage of fast learning and low computational cost, this approach recognizes similar clusters to the input pattern according to both the overall similarity across all of the feature channels, and the individual similarity of each feature channel.

We develop ART-based clustering algorithms as a solution for the aforementioned challenges of clustering social media data with the following considerations:

- Regarding information representation, as discussed in Section 1.2.1, existing methods usually suffer from the problems of high dimensionality and noisy features. Especially for the meta-information which is essentially short and noisy, there are still no established statistical methods that can effectively discover the key tags that reveal the semantics of the meta-information.

  ART uses a weight vector to model the characteristics of the patterns in the same cluster which is updated by a learning function that suppresses the values of noisy features while preserving the key features, and evaluates the similarity between

the input pattern and cluster weight vector by the intersection of their feature distributions. Taking the advantages of ART, the developed ART-based algorithms are able to identify the key features of the patterns in the same cluster through the learning algorithm of ART in order to alleviate the problem of noisy features. In Chapter 3, we propose a clustering algorithm, called Probabilistic Fusion ART (PF-ART), for handling meta-information. PF-ART uses the tag presence in a data object to construct the feature vector and incorporates a novel learning function that models the weight vector of a cluster using the probabilistic distribution of tag occurrences. Thus, the similarity measure for the meta-information becomes a match of key features between the input patterns and clusters. In this way, PF-ART resolves the representation problem of meta-information by transforming the task of identifying the features of meta-information in the feature construction stage to that of identifying the semantics of data clusters during the learning stage.

- Regarding the scalability to big data, ART is an incremental clustering algorithm. Thus, the feature vectors of all patterns are not required to be presented into the computer memory at the same time when processing a very large data set. Besides, ART incrementally processes input patterns one at a time by performing real-time searching and matching of suitable clusters, which ensures its linear time complexity of $O(n)$. Additionally, ART can finish the cluster membership evaluation and assignment of patterns in one round of presentation, which will incur a small increase in time cost with respect to the increase in the magnitude of the data set. Therefore, the ART-based algorithms will inherit the above advantages of ART and be able to handle big data.

- Regarding the sensitivity to input parameters, the performance of ART mainly depends on a single parameter, namely, the vigilance parameter, which controls the minimum intra-cluster similarity. As the vigilance parameter is a ratio value, determining to which degree the input pattern can be deemed similar to the clusters, it is easier for users to understand and decide its value than the parameters in other algorithms, such as the number of clusters and the distances between patterns. In Chapter 5, we experimentally demonstrated that a reasonable value of

vigilance parameter can be empirically chosen by tuning the value of the vigilance parameter until a few small clusters are generated. In addition, in Chapter 6, we further propose three methods for making the vigilance parameter self-adapted for individual clusters so that the performance of the ART-based clustering algorithms are more robust to the input parameters.

- Regarding the fusion of heterogeneous information, Fusion ART provides a general framework for integrating the multi-modal features. Specifically, Fusion ART allows input patterns to be represented by multiple feature vectors and interprets the heterogeneous data co-clustering task as a mapping from the multiple feature spaces to the category space. Besides, Fusion ART employs a vigilance parameter for each input channel so that the patterns in the same cluster should be consistently similar to each other in every feature space. In Chapter 4, we propose a Generalized Heterogeneous Fusion ART (GHF-ART) that extends Fusion ART to allow different feature channels to have different feature representation with different learning functions. More importantly, by incorporating an adaptive function to adjust the weights across the feature channels in the choice function, GHF-ART offers an effective approach to unify and synchronize multiple types of features for similarity measure.

- Regarding the robustness to noisy features, the learning function of ART adapts a cluster weight vector by incremental learning from the patterns categorized into this cluster to decrease the feature values. In this way, the key features of this cluster can be identified by suppressing the inconsistent features while preserving the key and consistent ones, and the matching between input patterns and clusters essentially measures the matching of shared key features. Also, with a reasonable value of vigilance parameter, ART will generate small clusters to encode the ill-featured noisy patterns that are isolated in the feature space. Therefore, the well-formed clusters will not be affected by the noisy patterns. By preserving those mechanisms, the developed ART-based clustering algorithms also have a strong immunity to noisy features.

- Regarding the incorporation of user preferences, in Chapter 3, we propose the PF-ART that extends ART to receive three forms of user preferences. First, users are allowed to figure out groups of data objects belonging to the same class. Taking into account the incremental clustering manner of ART, PF-ART is able to create a set of pre-defined clusters in the category space for modeling each group of patterns before clustering. This method can be viewed as a partitioning of the category space where the interesting regions to the users are discovered. Those clusters will be incrementally expanded and generalized by encoding the subsequent input patterns. In this way, users are likely to obtain interesting groups of data objects from the clustering results. Second, users are allowed to provide additional information, such as short sentences and tags, to describe the data objects. Those user preferences can be modeled as a feature vector for describing the data patterns, which can be received by an additional feature channel in PF-ART. Third, PF-ART allows the users to tune the vigilance parameter to produce personalized cluster structure of data sets. As PF-ART utilizes the vigilance parameter to control the intra-cluster similarity of patterns, a larger value of vigilance parameter results in the generation of clusters with more specific semantics.

1.4 Contributions

In this thesis, we have carried out investigations in terms of the tag-based semi-supervised hierarchical clustering for personalized web image organization, the heterogeneous data co-clustering for web multimedia data co-clustering, the community discovery of social users in heterogeneous social networks, and the adaptive scaling of cluster boundaries for large-scale social media data clustering. The four works are described below:

(i) We proposed a two-step semi-supervised hierarchical clustering algorithm, termed Personalized Hierarchical Theme-based Clustering (PHTC), for organizing large-scale web image collections based on the surrounding text of images. The algorithm involves 1) a semi-supervised clustering algorithm, called Probabilistic Fusion Adaptive Resonance Theory (PF-ART), to group images into semantic clusters under user supervision; and 2) an agglomerative algorithm to reveal the discovered theme structure.

In the first step, PF-ART groups images with similar themes into clusters and identifies the key tags of clusters at the same time. Additionally, it may receive user preferences, including the indication of groups of patterns belonging to the same class and user annotations, to enhance the clustering quality and produce clusters according to user preferences. Also, by tuning the vigilance parameter, the users can flexibly and directly control the degree of generalization of the discovered themes from the clusters. In the second step, an agglomerative algorithm based on Cluster Semantic Relevance (CSR) is proposed to further associate the generated clusters with similar themes together and generate a multi-branch tree hierarchy, so that father nodes contain more general themes than their children nodes. This help to provide a compact and systematic interface for image organization.

The performance of the proposed algorithms is evaluated on two web image data sets, namely the NUS-WIDE and Flickr data sets. The experimental results, in terms of the clustering quality and time cost, have demonstrated that the proposed PF-ART usually performs better than many of existing clustering algorithms, and the incorporation of user preferences consistently improves the clustering performance. By employing the proposed agglomerative algorithm, PHTC achieves a significantly better performance and much faster running speed than some of existing hierarchical clustering algorithms.

(ii) We proposed a heterogeneous data co-clustering algorithm, termed Generalized Heterogeneous Fusion Adaptive Resonance Theory (GHF-ART), for the co-clustering of multimedia web documents with the attached rich meta-information.

GHF-ART, extended from Fusion ART, allows different representation and learning functions for different feature channels. Thus it can process multiple types of features in order to handle multimedia data with an arbitrarily rich level of meta-information. For handling short and noisy text, GHF-ART does not directly learn the cluster prototypes from the textual features. Instead, it identifies key tags by learning the probabilistic distribution of tag occurrences in the clusters. More importantly, GHF-ART incorporates an adaptive method for the effective fusion of heterogeneous features, which weights the features extracted from different data

sources by incrementally measuring the importance of different feature modalities through the intra-cluster scatters.

Extensive experiments on two web image data sets, namely, the NUS-WIDE and Corel data sets, and one text document data set, known as the 20 Newsgroups data set, have shown that GHF-ART achieves a significantly better clustering performance and is much faster than many existing state-of-the-art algorithms.

(iii) We explored the feasibility of GHF-ART on the clustering of social users through their association in heterogeneous social networks, each of which describes one type of links among the users. To this end, we categorized commonly used social links into three categories, including relational links, textual links in articles and textual links in short text. Based on the characteristics of different types of social link data, we developed a set of specific pattern representation and learning rules for GHF-ART to handle various types of heterogeneous social links. Compared with existing heterogeneous data co-clustering algorithms, GHF-ART has the advantages of scalability to big data, requiring no pre-determined number of clusters and a well-defined weighting function for heterogeneous feature fusion.

The performance of GHF-ART is analyzed on two public heterogeneous social networks, namely, the YouTube data set and the BlogCatalog data set, in terms of the parameter selection, the clustering performance comparison, the performance analysis of the weighting function and the time cost comparison. Experimental results have shown that GHF-ART usually performs better and has a much lower time cost than many of existing heterogeneous data co-clustering algorithms. Additionally, using the case studies on the generated clusters on both data sets, we have shown that, by the simultaneously co-clustering of multi-modal feature data, GHF-ART is capable of identifying the key features and revealing the correlations between different types of feature data.

(iv) To fit clusters of different sizes in complex social media data, we investigated the geometrical dynamics of the clustering process of Fuzzy ART, and proposed three heuristic methods, namely the Activation Maximization Rule (AMR), the Confliction Minimization Rule (CMR) and the Hybrid Integration Rule (HIR), to make

the vigilance parameter in Fuzzy ART self-adaptable. Consequently, the clusters in the Fuzzy ART system will have individual vigilance levels that are able to adaptively tune their boundaries to accept similar patterns during the clustering process.

Through the experiments on two social media data sets, namely, the NUS-WIDE data set and the 20 Newsgroups data set, we demonstrate that, by incorporating AMR, CMR and HIR into Fuzzy ART, the resulting AM-ART, CM-ART and HI-ART consistently perform better and are more robust to the initial vigilance value than Fuzzy ART. Particularly, AM-ART significantly reduces the number of small clusters when the vigilance value is large. In contrast, CM-ART performs much better than Fuzzy ART when the vigilance value is small. HI-ART, which incorporates the ideas from both AMR and CMR, takes advantage of AM-ART and CM-ART. More importantly, the ART-based clustering algorithms usually perform better and require significantly less time cost than several existing clustering algorithms that require no pre-defined number of clusters.

1.5 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 reviews previous related work on clustering, semi-supervised clustering, heterogeneous data co-clustering, and associative tasks on social media mining. Chapter 3 presents the proposed hierarchical clustering algorithm, the Personalized Hierarchical Theme-based Clustering (PHTC), and its application to web image organization. Chapter 4 describes the Generalized Heterogeneous Fusion ART (GHF-ART) and its application to web multimedia data co-clustering. Chapter 5 illustrates the application of GHF-ART to the user community discovery task in heterogeneous social networks. Chapter 6 reports the investigation on self-adapting the vigilance parameter in Fuzzy ART for the adaptive scaling of cluster boundaries in the feature space and its applications to clustering complex social media data. The final chapter concludes the thesis and highlights the future work.

Chapter 2

Literature Review

This thesis focuses on developing novel clustering algorithms for addressing the problems of effectively discovering the underlying themes of the social media data, including the scalability to big data, the robustness to noisy features, the fusion of rich but heterogeneous information, the sensitivity to input parameters, and the ability to incorporate user preferences. To this end, we summarize existing studies on clustering techniques, in terms of clustering, semi-supervised clustering, and heterogeneous data co-clustering. Besides, in view that most clustering techniques require settings of the number of clusters in the data set, which is difficult to tune when applied to a large and complex social media data set, we also review existing techniques that help to decide the value of this parameter automatically and highlight the clustering algorithms that do not require such a parameter.

Furthermore, we present a survey of further applications of clustering algorithms to social media mining tasks, such as web image organization, multi-modal information fusion, user community detection, user sentiment analysis, social event detection, and community question answering.

2.1 Clustering

Clustering, aimed at identifying natural groupings of a data set, is a commonly used technique for statistical data analysis in many fields, such as machine learning, pattern recognition, image and text analysis, information retrieval, and social network analysis. In this section, we present a literature review on important clustering techniques for multimedia data analysis in terms of different theoretical basis.

2.1.1 K-Means Clustering

K-means clustering [21] is a centroid-based partitional algorithm, which partitions the data objects, represented by feature vectors, into k clusters. It iteratively seeks for k cluster centers in order to minimize the intra-cluster squared error. K-means clustering is widely used due to its easy implementation and well-founded objective function, and many variations have been proposed such as Fuzzy C-means Clustering [22] and Kernel K-means Clustering [23]. However, it suffers from two fundamental drawbacks: 1) the number of clustering k is difficult to determine; and 2) the clustering result is sensitive to the initialization of cluster centers. Accordingly, a large number of research efforts have been done to tackle these problems, such as [24–26].

Although such problems are still unsolved, the standard K-means clustering, in practice, frequently finds reasonable solutions quickly and is widely used in various applications, such as image segmentation [27], image organization [28], and graph theoretic clustering [29, 30].

2.1.2 Hierarchical Clustering

Hierarchical clustering algorithms attempt to generate a hierarchy of clusters for data objects. Typically, hierarchical clustering techniques fall into two types:

- **Agglomerative clustering**: Each data object is a leaf cluster of the hierarchy, and pairs of clusters are merged iteratively according to certain similarity measures.
- **Divisive clustering**: All the data objects start in one cluster, and cluster splitting is performed recursively according to some dissimilarity measures.

The similarity or dissimilarity between clusters is usually measured by the Linkage criteria, such as single-linkage [31], average-linkage [32], and complete-linkage [33].

Although hierarchical clustering has been widely used in image and text domains [34, 35], three major problems remain: 1) High time complexity, usually of $O(n^3)$, limits its scalability for big data sets; 2) The generated hierarchy can be very complex for a data set containing diverse contents; and 3) Deciding the stop criteria is difficult.

In recent years, some hierarchical clustering algorithms have been developed for web image organization [28, 36], which successively use different types of features, such as textual and visual features, to build a multi-layer hierarchy. However, this approach cannot provide a semantic hierarchy of clusters. Also, it suffers from the problem of error propagation, because the clustering result of data objects in one layer is based on that of the previous layers.

2.1.3 Graph Theoretic Clustering

Graph theoretic clustering models the relations between data objects by a graph structure, where each data object is a vertex and an edge between a pair of vertices indicates their relation. This approach is intended to group the vertices into clusters according to some optimization criteria. Graph theoretic clustering is widely studied in the literature because of its well-defined objective functions which can be easily utilized to formulate a wide range of clustering problems.

Spectral clustering, one of the most well-known graph theoretic clustering methods, refers to a type of clustering techniques, such as normalized cut [29] and minimum cut [37]. They make use of the eigenvalues of the similarity matrix of the data to project the data from the feature space with high dimensions to the one with low dimensions before clustering. Subsequently, a traditional clustering algorithm, such as K-means clustering algorithm, is invoked to obtain the final clusters.

To decrease the computation cost and avoid the effect caused by different similarity measures, a bipartite spectral graph partitioning [38] is proposed. It directly models the relations between data and features using a bipartite graph and finds the solution by solving a singular value decomposition problem [39]. A similar idea has been applied on the image domain [40].

2.1.4 Latent Semantic Analysis

Latent Semantic Analysis (LSA) [41] is initially proposed to analyze the relationships between a set of documents and the words therein. Given a term-document matrix, LSA decomposes the matrix into three matrices via singular value decomposition (SVD) [39]. The key idea behind LSA is to map the high-dimensional term vectors of documents to

a lower dimensional representation in a so-called latent semantic space. Analogous to spectral clustering, a traditional clustering algorithm should be employed to obtain the cluster assignment of data objects in the latent semantic space. LSA has been applied to a wide range of topics including text summarization [42, 43], face recognition [44], and image retrieval and annotation [45].

2.1.5 Non-Negative Matrix Factorization

Non-negative Matrix Factorization (NMF) [46], similar to the Latent Semantic Analysis (LSA), is also a technique based on matrix factorization. In contrast, NMF iteratively decomposes the matrix into two matrices based on an objective function in order to minimize the reconstruction error. Different from spectral clustering and LSA that are equivalent to the feature reduction process, NMF derives a cluster indicator matrix that directly reveals the relations between each document and a pre-defined number of clusters (dimensions). As such, the cluster membership of each document is determined by the largest projection value among all of the dimensions. A study [47] indicates that NMF outperforms spectral methods in text document clustering in terms of both accuracy and efficiency.

Recently, a tri-factorization objective function [48] has been proposed for a general framework of data clustering, which has been extended to perform document-word co-clustering [49] and semi-supervised document clustering [16].

2.1.6 Probabilistic Clustering

Probabilistic clustering, usually referred to as mixture models, is a model-based approach, which uses statistical distributions to model clusters and achieves the cluster assignment of data objects by optimizing the fit between the data and the models. Specifically, this approach assumes that data objects are generated from a set of probabilistic distributions, so the data points in different clusters should follow different probabilistic distributions. Typically, this approach requires the users to specify the number and the functional forms of the distributions, such as the Gaussian distribution [50]. As such, the clustering process is equivalent to estimating the parameters of the probabilistic distributions.

The most popular method for the parameter estimation task of probabilistic distributions is the Expectation-Maximization (EM) algorithm [51–53], which estimates the maximum likelihood of parameters based on the Bayes's theorem. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters by maximizing the expected log-likelihood found in the E step.

2.1.7 Genetic Clustering

The use of genetic algorithms [54, 55] to identify the best clustering typically depends on the evolution of cluster structures, as evaluated by certain cluster validity indices. In [54], a symmetry-based genetic clustering algorithm, called VGAPS-clustering, was proposed, in which each pattern in the population pool is a concatenation of the weight values of cluster centers, and different patterns may have different numbers of centers. After the maximum number of generations, the pattern with the highest fitness is selected as the best cluster structure.

Genetic clustering can identify clusters of arbitrary shapes and achieve a global optimum. However, genetic clustering algorithms are usually quite slow due to the stochastic evolution of patterns. The experiments presented in [54] were conducted only on a few small data sets with several hundred patterns, each of which also had a small number of dimensions. A review of genetic clustering algorithms is provided in [55].

2.1.8 Density-Based Clustering

Density-based clustering identifies dense regions of patterns as clusters in the feature space. DBSCAN [13] forms the degree of density using two parameters, namely, the maximum distance for the search of neighbors and the minimum number of neighbors; patterns and their neighbors that satisfy the above requirements are called core points and are deemed to be in the same cluster. The patterns that do not satisfy the requirements and are not neighbors of any core point are considered noise.

In addition, DBSCAN has several extensions. GDBSCAN [56] extends DBSCAN so that it can cluster point objects and spatially extended objects according to both their

spatial and non-spatial attributes. OPTICS [57] provides a hierarchical view of the data structure, which is equivalent to the density-based clusterings corresponding to a broad range of parameter settings. DECODE [58] composes clusters with different densities in the data set; more specifically, it computes the mth nearest distance of each pattern and uses reversible jump Markov Chain Monte Carlo (MCMC) to identify the clusters of patterns in terms of their different densities. Tran et al. [59] proposed a density-based clustering algorithm, KNNCLUST, in which the density is measured by a KNN-kernel table. With a pre-defined number of neighbors, all of the patterns in the data set are assigned to clusters according to the proposed KNN-kernel Bayes' class-condition. The cluster memberships of all of the patterns are recalculated until their cluster assignments stop changing.

Density-based algorithms have a number of advantages, including their ability to form clusters with arbitrary shapes and their insensitivity to initialization. However, they require several pre-defined parameters that are difficult to decide, such as the minimum number of neighbors in DBSCAN, the value of m and the parameter for deciding the probability mixture distribution in DECODE, the number of neighbors in the KNN table and the choice of kernel in KNNCLUST. In addition, density-based clustering algorithms typically require a quadratic time complexity of $O(n^2)$, which may be reduced to $O(n\,log\,n)$ when a spatial index structure is used to speed up the search process of neighbors [56]. A review of density-based clustering algorithms can be found in [60].

2.1.9 Affinity Propagation

Affinity Propagation [61] is an exemplar-based clustering algorithm that identifies a set of representative patterns as "exemplars" to the other patterns in the same cluster. Exemplars are identified by recursively updating two messages of patterns, namely, the "availability" to indicate the qualification of a pattern to be an exemplar, and the "responsibility" to indicate the suitability of a pattern to be a member of the exemplars' clusters. The algorithm stops when the exemplars for all of the patterns remain for a number of iterations, or upon reaching a maximum number of iterations.

Two algorithms [62, 63] have been proposed to improve the efficiency of Affinity Propagation. Fast Sparse Affinity Propagation (FSAP) [62] generated a sparse graph using

the K-nearest neighbor method, rather than the original similarity matrix, in order to reduce the computation of message transmission in Affinity Propagation. In [63], the proposed fast algorithm for Affinity Propagation reduced the computation by pruning the edges that can be directly calculated after the convergence of Affinity Propagation.

Affinity Propagation has shown better performance than K-means in terms of the average squared error. However, it has a quadratic time complexity of $O(tn^2)$ where t is the number of iterations. Even the fastest one [63] also has a quadratic time complexity of $O(n^2 + tm)$, where m is the number of edges. In addition, Affinity Propagation usually requires the tuning of four parameters, including the preference vector "preference" which controls the number of generated clusters and impacts the speed of convergence, the damping factor "dampfact", and the maximum and minimum number of iterations "maxits" and "convits" which ensure convergence.

2.1.10 Adaptive Resonance Theory

Adaptive Resonance Theory (ART) [19] is a learning theory on how a human brain memorizes events and objects, and leads to a series of real-time unsupervised learning models capable of fast and stable category recognition, such as ART 1 [64], ART 2 [65], ART 2-A [66], ART 3 [67], and Fuzzy ART [68], as well as supervised learning models, such as ARTMAP [69] and Fuzzy ARTMAP [70].

ART 1, the basic ART-based clustering model, incrementally performs real-time searching and matching between input patterns and existing clusters (memory prototypes) in the category space one at a time. When an input pattern is presented, the best-matching cluster, called the winner, in the category field is selected using the choice function, whose similarity to the input pattern is evaluated by the match function. If the similarity is larger than a certain threshold, as controlled by the vigilance parameter, the input pattern is categorized into the winner, called a resonance. Otherwise, a reset occurs, leading to the search for another winner in the rest of the clusters. If all of the clusters in the category space incur a reset, a new cluster is generated to encode this novel pattern. ART has advantages of fast and stable learning as well as incremental clustering manner, and has been successfully applied to many applications, such as pattern recognition and document organization [71].

Fuzzy ART replaces the intersection operator ($\cap$) used in ART 1 with the min operator ($\wedge$) used in fuzzy set theory. More importantly, Fuzzy ART incorporates complement coding, which normalizes the input patterns and alleviates the cluster proliferation problem that occurs in ART 1. Fuzzy ART has been used in different ART-based variants to resolve many image and text mining problems, such as web document management [72], tag-based web image organization [73], image-text association analysis [74], multimedia data co-clustering [75] and social community detection in heterogeneous social networks [1].

2.2 Semi-Supervised Clustering

Whereas clustering organizes data objects into groups according to purely similarity (or distance) measures in the feature space, semi-supervised clustering exploits the available prior knowledge, also called side information, to guide the clustering process. Typically, the prior knowledge is given by the information about the related and/or irrelevant data objects. Group label constraint and pairwise constraint are two commonly used methods for providing such information.

Group label constraint requires users to indicate subsets of documents in the data set that belong to the same class. Semi-supervised learning is usually achieved by learning a metric for adjusting the similarity measure [76, 77] or incorporating such constraints to adjust the objective function of the original clustering algorithms [78, 79].

Pairwise constraint is the most widely used method in practice, because of its easy access to users and not requiring users to have much prior knowledge. Using this method, users need to provide a set of must-link and cannot-link constraints to indicate if pairs of documents should be associated with the same cluster or not. Chen et al. developed two methods for incorporating the pairwise constraints into the Non-negative Matrix Tri-Factorization (NMF) algorithm [48]. The first method [16] adds the constraints into the objective function as rewards and penalties to balance the clustering. The other method [17] computes new relational matrices for documents through a distance metric learning algorithm such that, in the derived feature space, documents with must-link are moved closer while those with cannot-link are moved farther apart. Beside the NMF, spectral

constrained clustering algorithms for incorporating pairwise constraints also have been widely studied [18, 80]. Other notable works include Semi-supervised Kernel K-means (SS-KK) [81] and Semi-supervised Spectral Normalized Cuts (SS-SNC) [82].

2.3 Heterogeneous Data Co-Clustering

Heterogeneous data co-clustering addresses the problem of clustering composite objects, which are described by the data from heterogeneous resources. Typically, the data objects, such as images, text documents, and social users, and their associated descriptive information are modeled as a star structure [10]. By simultaneously integrating those different types of data as the multi-modal features of the composite objects, the heterogeneous data co-clustering task is to find the best partitioning of the composite objects, considering their similarities in terms of each feature modality.

In this section, we illustrate existing heterogeneous data co-clustering algorithms in terms of different model formulations, which can be organized into six categories, discussed as follows.

2.3.1 Graph Theoretic Models

A large body of recent literature on heterogeneous data co-clustering is based on graph theoretic models. Gao et al. [83] proposed a web image co-clustering algorithm, named Consistent Bipartite Graph Co-partitioning (CBGC). This algorithm interprets the image-text co-clustering task as a tripartite graph and transforms the partitioning of the tripartite graph into the simultaneous partitioning of the visual and textual graphs. In this way, CBGC models the solution as a multi-objective optimization problem which is solved by the semi-definite programming (SDP). This work has been generalized to process multi-modal heterogeneous data in [84]. However, CBGC requires empirical settings of three parameters, and should employ traditional clustering algorithms on the embedding vectors produced to obtain the final clusters.

A similar work [30] to CBGC, called Consistent Isoperimetric Highorder Co-clustering (CIHC), also considers the problem of integrating visual and textual features as the partitioning of a tripartite graph. Different from CBGC, CIHC solves the problem by extending the Isoperimetric Co-clustering Algorithm (ICA) [38], which can be solved by

a sparse system of linear equations. CIHC has been demonstrated to be more effective and has much lower time cost than CBGC. However, it also requires an additional clustering algorithm to partition the obtained embedding vectors, and is only applicable to distinguish data of two classes.

Long et al. [85] proposed the Spectral Relational Clustering (SRC) for the clustering of multi-type relational data. They first proposed a collective clustering based on minimizing the reconstruction error of both the object affinity matrices and the feature matrices, and then derived an iterative spectral clustering algorithm accordingly for the factorization of these relational matrices. However, SRC requires solving the eigenvalue decomposition problem which is inefficient for large-scale data sets. In addition, a separate clustering algorithm, in this case K-means, is used to obtain the final clustering.

Zhou et al. [9] proposed a multi-view spectral clustering for clustering data with multiple views. This method generalizes the normalized cut from a single view to multiple views by forming a mixture of Markov random walks on each graph, and aims to divide the data objects into two clusters, which should be a good partitioning for each of the graphs. Therefore, this method is not suitable for clustering data sets with many underlying clusters.

Cai et al. [86] proposed a Multi-Modal Spectral Clustering (MMSC) to simultaneously integrate five types of visual features for image clustering. In order to obtain the final cluster indicator matrix, MMSC uses a unified objective function to simultaneously optimize the clustering results of each feature modality and their combination. This objective function is finally solved by eigenvalue decomposition and a spectral rotation algorithm.

A Multi-Modal Constraint Propagation (MMCP) [80] has been proposed for the semi-supervised clustering of multi-modal image sets. MMCP first defines the random walk on the multiple graphs, each of which corresponding to one type of modalities. Subsequently, by decomposing the problem of label propagation on multiple graphs into a set of independent multi-graph based two-class label propagation sub-problems, MMCP deduces the refined similarity matrix of data objects through a series of quadratic optimization procedures. A spectral clustering algorithm is applied to obtain the final clustering results.

In view of the above issues, the graph theoretic models typically utilize a unified objective function to realize the fusion of multi-modal features, and require a series of matrix operations to deduce a vector or matrix that reveals the features of data objects. It is notable that the graph theoretic models deal with the similarity matrix of the data objects instead of the feature matrix. So, in practice, evaluating the similarities between data objects should be considered first. A drawback of this approach is the computational complexity due to the mathematical computation. Also, the clustering performance depends on the traditional clustering algorithms that are used to obtain the final results.

2.3.2 Non-Negative Matrix Factorization Models

The non-negative matrix tri-factorization (NMF) approach, as illustrated in Section 2.1.5, iteratively factorizes the data matrix into three sub-matrices, one of which, called the cluster indicator matrix, reveals the projection values of data objects to the dimensions (clusters).

Chen et al. [10] proposed a symmetric nonnegative matrix tri-factorization algorithm, called Semi-Supervised NMF (SS-NMF), which attempts to find a partitioning of the data objects in order to minimize the global reconstruction error of the relational matrices for each type of data. Similar to the NMF, the cluster membership of each data object is determined by the largest projection value among all clusters. Moreover, by incorporating the user-provided pairwise constraints, SS-NMF derives new relational matrices through a distance learning algorithm to enhance the clustering performance.

Linked Matrix Factorization (LMF) [87] has an objective function similar to that of SS-NMF. However, LMF minimizes the overall reconstruction error and maximizes the sparsity of the factorized sub-matrices at the same time. Also, a semi-supervised version using pairwise constraints is proposed for metric learning.

The NMF approach has the advantage of a linear time complexity of $O(tn)$, where t is the number of iterations and n is the number of data objects. However, it requires users to set the number of clusters for the data objects and each type of features to construct the sub-matrices, and its performance may vary with different initializations of the sub-matrices.

2.3.3 Markov Random Field Model

Bekkerman et al. [88] proposed the Combinatorial Markov Random Fields (Comrafs) for the multi-modal information co-clustering based on the information bottleneck theory, and applied it to various applications, such as semi-supervised clustering [89], multi-modal image clustering [12] and cluster analysis [90].

Comrafs constructs a set of Markov random fields for each type of data, wherein each data modality is modeled as a combinatorial random variable which takes values from all the possible partitions, and the edges between pairs of variables are represented using mutual information. The approach of Comrafs is to maximize the information-theoretic objective function, which is resolved by the hierarchical clustering algorithm with either agglomerative or divisive strategies.

One potential problem of this approach is the heavy computational cost, having the time complexity of $O(n^3 \log n)$. As Comrafs needs to traverse all subsets of the data samples for each data modality, the computational cost increases significantly with respect to the size of data sets.

2.3.4 Multi-View Clustering Models

The multi-view clustering models consider the clustering of data objects with two types of features. Typically, two clustering algorithms are employed for each set of features. Subsequently, the learnt parameters of two clustering models are refined by learning from each other iteratively. However, this approach is restricted to two types of data.

In [8], three types of traditional clustering algorithms, namely the EM, K-means, and agglomerative clustering algorithms, are extended to fit the multi-view clustering framework. Besides, the extended EM and K-means algorithms have been applied for discovering communities in linked data [91].

Recent studies also developed multi-view clustering models based on Canonical Correlation Analysis [92] and spectral clustering [93].

2.3.5 Aggregation Based Models

The aggregation approach follows the similar idea of first identifying the similarity between the data objects through each type of features, and subsequently integrating them

to produce the final results.

Principal Modularity Maximization (PMM) [94] first obtains a fixed number of eigenvectors from the modularity matrices which are produced with each type of the relational matrix. Then, those eigenvectors are concatenated into one matrix, and singular value decomposition is employed to obtain the final embedding vectors for each data object. Finally, K-means is used to obtain the final clustering.

MVSIM [11] is an iterative algorithm based on a co-similarity measure termed X-SIM, which, given a relational matrix, evaluates both the similarity between the data patterns and also the features. In each iteration, MVSIM runs X-SIM on the relational matrices for each feature modality to obtain the similarity matrices and then aggregates them to form an integrated similarity matrix using an update function.

2.3.6 Fusion ART

As discussed in Section 2.1.10, Adaptive Resonance Theory (ART) is an incremental clustering algorithm, which processes input patterns one at a time, and employs a two-way similarity measure to perform real-time searching and matching of suitable clusters to the input patterns.

Fusion ART [20] extends ART from a single input field to multiple ones and learns multi-channel mappings simultaneously across multi-modal pattern channels in an online and incremental manner. As a natural extension of ART, Fusion ART is composed of multiple input feature channels, each of which corresponds to one type of features. Thus, each type of features of the data objects is processed independently, and the output similarities of each feature channel are integrated through a choice function.

Different from existing heterogeneous data co-clustering algorithms, Fusion ART allows the flexibility of using different learning methods for different types of features, and considers both the overall similarity across feature channels and the individual similarity of each modality. More importantly, Fusion ART has a very low computational complexity of $O(n)$, so it is suitable for clustering large-scale data sets. Successful applications on the multimedia domain [74, 95] have demonstrated the viability of Fusion ART for the multimedia data analysis.

2.4 Automatic Recognition of Clusters in a Data Set

Existing clustering algorithms typically require settings of the number of clusters in data sets. However, different from traditional image and text document data sets, the social media data sets are usually large-scale and may cover diverse content across different topics, making it difficult to manually evaluate the number of underlying topics in the data sets. Therefore, automatically identifying the number of clusters in data sets becomes a key challenge for clustering social media data sets.

In this section, we introduce existing approaches on the automatic recognition of clusters in a data set.

2.4.1 Cluster Tendency Analysis

Cluster tendency analysis aims to identify the number of clusters in a data set before clustering. Most recent studies [14, 96, 97] have focused on investigating the dissimilarity matrix of patterns.

Visual Assessment of Tendency (VAT) [96] reorders the dissimilarity matrix of patterns so that patterns in nearby rows will have low dissimilarity values. When displaying the reordered matrix as an intensity image, referred to as a "reordered dissimilarity image" (RDI), the number of clusters may be determined by counting the dark blocks along the diagonal pixels in the image. However, in complex data sets, the boundaries between dark blocks may be indistinct, making it difficult to correctly identify the number of clusters.

Therefore, Cluster Count Extraction (CCE) [97] and Dark Block Extraction (DBE) [14] are further proposed to objectively identify the number of clusters without relying on manual counting. CCE attempts to remove noise in the RDI obtained by VAT through two rounds of Fast Fourier Transform (FFT) with a filter that transforms the RDI to and from the frequency domain. The number of clusters equals the number of spikes in the histogram constructed by the off-diagonal pixel values of the filtered image. In contrast, after obtaining the RDI, DBE employs several matrix transformation steps to project all of the pixel values of the RDI to the main diagonal axis in order to obtain a projection signal. The number of clusters equals the number of major peaks in the signal.

In practice, a traditional clustering algorithm, such as K-means, can be employed to obtain the clusters using the identified number of clusters. However, such methods have several limitations when applied to web multimedia data. First, because these data sets typically involve noise, the dissimilarity matrix may not represent the structure of the data in the input space well, which may result in an RDI with low quality. Second, such methods employ heavy computation, so their performance is only measured on small data sets containing several thousand patterns.

2.4.2 Clustering Validation

Cluster validation aims to quantitatively evaluate the quality of different cluster structures, usually based on intra-cluster compactness and between-cluster separation, so as to find the best clustering.

Liang et al. [98] proposed a modified K-means algorithm with a validation method based on the intra-cluster and between-cluster entropies. This algorithm requires K-means to run multiple times, starting with a pre-defined maximum number of clusters. During each iteration, the "worst cluster" is removed using information entropy, and the quality of the clusters is evaluated according to the proposed validation method. Upon reaching the pre-defined minimum number of clusters, the clustering with the best quality is identified.

In [99], Sugar et al. proposed a "jump method", which generates a transformed distortion curve based on the clustering results of K-means with different numbers of clusters. The highest peak, or "jump", in the curve represents the best number of clusters.

Kothari et al. [100] proposed a scale-based algorithm in which a "neighborhood" serves as the scale parameter. By varying the value of the neighborhood, the proposed algorithm may identify clusterings with different numbers of clusters. The best number of clusters is identified based on the persistence across a range of the neighbors.

A meta-learning-based algorithm was proposed in [101]. Given a data set, multiple new data sets are first generated by distorting the original patterns. Subsequently, for each data set, a traditional clustering method is employed to generate clusterings with different numbers of clusters, the quality of which is measured by the disconnectivity and

compactness. After identifying the elbows of both the disconnectivity and the compactness plots for each data set, the true number of clusters is decided by a vote.

The above methods typically are designed for hard clustering algorithms. For fuzzy clustering algorithms, a summary of existing cluster validity indices can be found in [15, 102].

2.4.3 Algorithms Without Pre-Defined Number of Clusters

As discussed above, the cluster tendency analysis requires heavy computation and is not robust to noise. Similarly, the cluster validation approach attempts to select the best number of clusters by evaluating the quality of clusterings with different numbers of clusters. As such, they are not feasible for the large-scale social media data sets.

Fortunately, there are clustering algorithms that do not require a pre-defined number of clusters, including the hierarchical clustering based algorithms, genetic clustering algorithms, density-based clustering algorithms, Affinity Propagation, and ART-based clustering algorithms.

As discussed in Section 2.1.2, hierarchical clustering algorithms either merge small clusters with individual data objects into big clusters or split the data set into individual data objects step by step. Therefore, existing studies typically incorporate a cluster validity index to measure the cluster quality during each merging or splitting iteration.

Li et al. [103] proposed an Agglomerative Fuzzy K-means algorithm that introduces a penalty term to the objective function of the standard Fuzzy K-means and requires a maximum number of clusters. The modified Fuzzy K-means runs multiple times with a gradually increased penalty parameter; during these runs, the clusters that share centers are merged according to a validation method. The algorithm stops when the number of cluster centers remains stable over a certain number of iterations.

Leung et al. [104] proposed a scale-based algorithm based on the scale space theory, in which a data set is considered an image, and each pattern is considered a light point on the image. The generation of a hierarchy is then simulated by blurring the image such that the light points gradually merge together. Several cluster validity indices, including lifetime, compactness, isolation and outlierness, are used to select the best cluster structure in the hierarchy.

In [105], an agglomerative clustering algorithm was proposed for transactional data. Based on the intra-cluster dissimilarity measure, referred to as the "coverage density", a "Merge Dissimilarity Index" is presented to find the optimal number of clusters.

Detailed illustration of genetic clustering algorithms, density-based clustering algorithms, Affinity Propagation, and ART-based clustering algorithms can be found in Section 2.1.7, Section 2.1.8, Section 2.1.9, and Section 2.1.10 respectively.

2.5 Social Media Mining and Related Clustering Techniques

The social media data refer to data that are generated by the social users on the social web sites, such as the tweets on Twitter, the blogs published on Facebook, the images shard on Flickr, the questions and answers on Yahoo! answers, and the user comments and descriptions for the above user-generated multimedia data.

As aforementioned, the big social media data record user behaviors and activities on the social web sites, and provide rich information for multimedia data understanding and social behavior analytics. However, different from traditional data sets for data mining tasks, they are large scale, noisy, multi-modal, unstructured, and dynamic in nature, due to the diverse communication tools provided by the social web sites.

Therefore, those distinguishing characteristics of social media data post new challenges for developing novel techniques to utilize the rich but noisy information for traditional multimedia data understanding and mining tasks, such as tag-based web image organization [28, 106], comment-based video organization [34], image retrieval assisted by web images and their surrounding text [107], short text understanding [4, 108], and multi-modal feature integration for social media data understanding [10, 75]. Additionally, numerous new problems and requirements arise, which are important for social media research and development, such as social community discovery [1, 109–111], user sentiment analysis [112, 113], influential user detection [114, 115], social link prediction and recommendation [116–118], question answering system analysis [119, 120], and emergent social event recognition and prediction [121–123]. A brief introduction of social media mining can be found in [124].

In the following sections, we illustrate the social media tasks that utilize clustering techniques as a solution.

2.5.1 Web Image Organization

The vast number of web images online motivates the requirement of effective image organization, especially the search results from web engines. Due to the diverse nature of web image contents, it is difficult to group images with similar semantics solely based on the visual features. Therefore, early efforts are usually based on the clustering of the textual features extracted from the surrounding text of web images [34, 106].

Additionally, there are some studies [28, 36] that make use of both the visual content and the surrounding text of web images in order to generate a two-layer hierarchical structure. Those methods typically apply clustering algorithms to the textual features to generate the first layer of clusters, and subsequently group the images in each cluster according to their visual features.

Beside the tag-based image organization techniques, there are also studies on improving the organization of the image search results using purely visual features. Leuken et al. [125] developed three clustering algorithms that can incorporate multiple types of visual features for partitioning images with different visual appearances. A weighting function is proposed to dynamically evaluate the distinguishing power to the images.

Recently, crowdsourcing has been incorporated into the clustering techniques as a solution to improve the clustering performance of web images [126]. By asking the web users to judge the cluster membership of some images, this type of clustering models utilizes such information as relevance constraint to learn a new distance metric, in order to refine the clustering performance.

2.5.2 Multi-Modal Information Fusion

The images and text documents in social media are usually attached with rich meta-information, such as category information, user description and user comments. Multi-modal information fusion, therefore, is aimed at processing those interrelated data modalities in a unified way and identifying their underlying interactions.

Image-text fusion for image clustering is widely studied for alleviating the semantic gap [73]. Early studies attempt to integrate the visual and textual features by either concatenating them into a single vector [64] or using them in a sequential manner [28]. However, the first approach usually cannot achieve desired results. The second method suffers from the problem of error propagation and the sequential usage of textual and visual features does not help to improve the clustering quality. Jiang et al. [74] interpret the fusion of visual and textual features as identifying pairs of related images and texts, and propose two methods, based on vague transformation [127] and Fusion ART [20], for learning the image-text associations. A large number of clustering techniques proposed for the fusion of multi-modal features are discussed in Section 2.3.

The fusion of multi-modal features is also an important research task for various applications, such as multi-document summarization [128, 129] and multi-modal multimedia data indexing and retrieval [130–133].

2.5.3 User Community Detection in Social Networks

A user community is formed by a group of social users having similar interests or behaviors, or interact with each other more frequently that those out of the group on the Web. The user community detection task is thus to identify different underlying communities in social networks, which may further benefit relevant research tasks, such as collective social behavior analysis [134] and social link prediction and recommendation [116–118].

A social network of users is typically modeled as a graph, where each node corresponds to a user and each edge indicates the strength of connection between two users, such as the frequency of contact or the number of co-subscription. Clustering is commonly used for the community detection task, especially the graph theoretic clustering algorithms. However, there are two challenges for applying traditional clustering algorithms to clustering social networks. The first challenge is the large-scale size of the social network data. To overcome this problem, existing studies attempt to reduce the computational cost of their algorithms by obtaining approximate solution from the simplified network graphs [135, 136] or developing parallel clustering models [7]. Despite the problem of big data, the other problem is the lack of ground-truth. Existing studies on assessing the

quality of the discovered clusters are usually based on the internal similarities or distances between nodes. Yang et al. [109] presented a comparative study of 13 evaluation measures for discovering the densely connected users as communities.

In recent years, a large body of studies focus on discovering user communities in the heterogeneous social networks. That is, users are connected with different types of links. Some of recent studies on this topic are based on multi-view clustering approach [91], matrix factorization approach [87] and aggregation approach [87]. Additionally, this task is closely related to heterogeneous data co-clustering, as discussed in Section 2.3.

2.5.4 User Sentiment Analysis

The analysis of user sentiment is aimed at understanding the sentiment of users from their comments on certain things like products, services and events.

Most of existing studies are based on supervised learning while those based on unsupervised learning are inadequate [137, 138]. Clustering algorithms, in this task, are typically performed to identify groups of users or comments that reveal similar sentiment, such as positive, negative and neutral. Hu et al. [138] incorporated emotional signals, such as emoticons and sentiment lexicon, into a non-negative matrix tri-factorization clustering algorithm to discover groups of users with similar sentiment. Zhu et al. [137] also developed a non-negative matrix tri-factorization model for clustering user and user comments. Moreover, an online framework has been proposed to receive dynamic online streams. A review of unsupervised sentiment analysis methods can be found in [138].

2.5.5 Social Event Detection

Clustering-based social event detection aims to identify the social events that attract collective attention through the massive number of posts and comments of users on the social web sites.

There are two directions for social event detection. One type of studies focuses on detecting real-time social events through online clustering algorithms. Becker et al. [123] developed an online clustering model with a set of cluster-level event features to group Twitter messages, and subsequently trained a classification model to judge whether the generated clusters are related to events.

The other type of studies focuses on detecting social events from a set of user messages collected from a given time period, also known as retrospective event detection [139]. Chen et al. [139] utilized the tags, time stamps, and location information of the images collected from Flickr to cluster these images and simultaneously obtain the key tags of clusters as events. Papadopoulos et al. [140] developed a clustering algorithm to cluster tagged images using their visual and textual features, and subsequently used a classifier to determine whether the clusters of images represent events or landmarks. Petkos et al. [141] developed a multi-modal spectral clustering algorithm to clustering multimedia data with different attributions, such as time, location, visual features, and tags.

2.5.6 Community Question Answering

The community question answering task attempts to resolve the problem of automatically providing answers to user's questions based on a question-answer database.

In this task, the user's question is typically treated as a query, and clustering is usually adopted to identify the question-answer pairs that are similar to the user query. Subsequently, answer ranking is further employed to produce relevant answers. In an early work, Kwok et al. [142] developed a question answering system, called Mulder. It first obtains a set of answers by sending the user's query to several search engines, and then uses a clustering algorithm to group similar answers together. Finally, a voting procedure is conducted to select the best-matching answer. Blooma et al. [143] modeled the question-answer pairs as a question-answer-asker-answerer quadripartite graph, and proposed an agglomerative algorithm to merge similar question-answer pairs. A review of question answering studies can be found in [144].

The community question answering task is also closely related to the task of query clustering [145–147], which addresses the problem of identifying and organizing similar user queries to web search engines.

Chapter 3

Semi-Supervised Hierarchical Clustering for Personalized Web Image Organization

Existing efforts on web image organization usually transform the task into surrounding text clustering. However, current text clustering algorithms do not address the problem of insufficient statistical information for image representation and noisy tags that greatly decreases the clustering performance while increasing the computational cost. In this chapter, we present our study on the proposed two-step semi-supervised hierarchical clustering algorithm, termed Personalized Hierarchical Theme-based Clustering (PHTC), for tag-based web image organization. First, a Probabilistic Fusion ART (PF-ART) is proposed to group images having similar semantics while simultaneously learning the probabilistic distribution of tag occurrences for mining the key tags/topics of clusters. Besides, PF-ART can incorporate user preference for semi-supervised learning and provide users a direct control of the clustering process. Secondly, an agglomerative merging strategy is used for organizing the clusters into a semantic hierarchy. The proposed merging strategy can provide a multi-branch tree structure rather than the traditional binary tree structure through a cluster distance measure called inner cluster scatter. Extensive experiments on two real world web image data sets, namely NUS-WIDE and Flickr, demonstrate the effectiveness of our algorithm for large web image data sets.

3.1 Introduction

Along with the explosive popularity of social web sites, a massive number of web images has appeared in diverse content online. It leads to the need for effective image organization to make the information more systematic and manageable. Two research challenges have been identified. The first challenge is how to learn the semantics (i.e. themes/topics) from images. Most of the existing applications [28, 34, 36, 106] are based on text clustering techniques, in which the tags of images extracted from their surrounding text (titles, categories information and user descriptions etc.) are used for image representation. This is because current state-of-the-art visual feature extraction techniques cannot fully represent the image content at the semantic level, a problem known as semantic gap. Thus, the problem of image organization is usually transformed into short text categorization. However, similar to the short document categorization problem [4], the tags cannot provide sufficient statistic information for effective similarity measure, i.e. the key tags that are useful for image topic representation cannot be revealed by traditional word weighting strategies, like term frequency-inverse document frequency (tf-idf). Besides, as users usually give descriptions based on their own views, the tags for images in one topic may be diverse, which is known as the problem of noisy tags. Therefore, traditional text clustering algorithms [16, 38, 47] may fail to achieve reasonable results when they are directly applied on this task. Besides, as existing algorithms are based on computational models, the noisy tags will significantly increase their computational cost.

The second challenge is how to associate the discovered topics. For a real world web image collection, there should be a large number of topics and sub-topics. Some of them may be relevant (e.g. "white tiger" and "Indian tiger"), and some of them may belong to a more general topic (e.g. "tiger" and "bird" belong to "animal"). It may result in the generation of too many categories. Therefore, a semantic hierarchy that can reveal the relationship between topics is necessary. However, existing hierarchical clustering approaches, such as [34], follow the agglomerative strategy which merges two clusters in one round. It leads to the problem that the generated binary tree structure becomes too complex when the number of the generated clusters is large.

In this study, we present a two-step hierarchical clustering algorithm termed Personalized Hierarchical Theme-based Clustering (PHTC) for organizing large-scale web

image collection. PHTC can incrementally discover the semantic categories and the key themes according to user preferences at the same time, and further organize the generated clusters into a multi-branch tree hierarchy. In the first step, we propose a novel semi-supervised clustering algorithm called Probabilistic Fusion Adaptive Resonance Theory (PF-ART), a variant of Fusion ART [20], for generating semantic clusters according to user preferences. Different from Fusion ART, PF-ART represents each cluster using the probabilistic distribution of tag occurrences. Beyond existing semi-supervised clustering algorithms [17, 18], PF-ART not only incorporates the user-provided information to enhance the clustering quality, but also provides the flexibility for users to directly control the degree of topic mining. That is, users can decide whether the clusters are generated according to general topics like "lion" and "bird", or more specific topics like "lion in zoo" and "lion in Africa". In the second step, we propose a similarity measure between categories called Cluster Semantic Relevance (CSR) and an agglomerative merging strategy based on CSR for generating the semantic hierarchy. Different from typical agglomerative algorithms [31–34], the proposed algorithm can recognize if the relationship between selected clusters is father and child, according to the inner CSR of children categories of the given category. Therefore, the generated hierarchy provides a multi-branch tree structure which is more systematic and clear.

3.2 Problem Statement and Approach

We define the problem of discovering semantics from web images as the problem of mining key themes from the surrounding text of web images. Given a set of web images and their surrounding text in the original web pages, such as titles, categories, and user descriptions, as the raw textual information obtained from the web pages is typically noisy, the first task is to filter the noisy words, including stop-words, typos, and chronic slangs. However, after removing the noisy words and stemming the variation form of words to the root form, the remaining tags are usually diverse, because of the diverse views of different users. It leads to the difficulties in identifying the underlying topics of the web images.

From the perspective that the semantically related images typically hold similar textual description, we can apply clustering algorithms to group similar images and identify

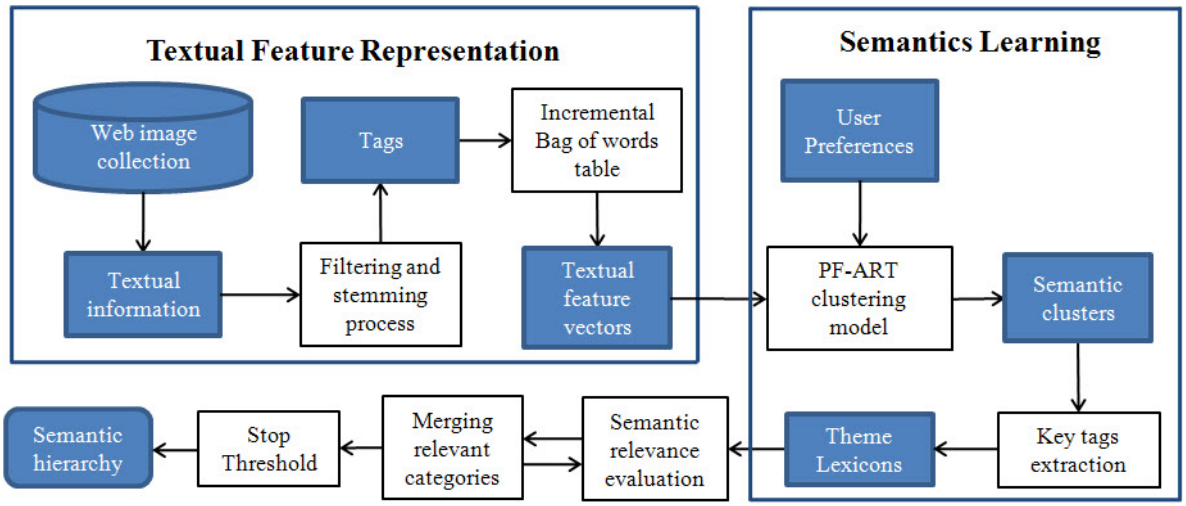


Figure 3.1: Procedures of the proposed clustering framework.

the key tags of each category. This work can also be treated as a tag refining procedure. To provide a systematic view of clusters, we further associate the semantic clusters by a semantic hierarchy.

The proposed clustering framework (Fig. 3.1) comprises three key modules: 1) Textual feature representation; 2) Semantic categories and theme lexicon generation; and 3) Semantic hierarchy construction. In the first module, given a collection of web images, the associated textual information goes through a pre-processing step, in order to obtain salient tags that are meaningful for representing the semantics of the respective images. Then, the bag-of-words method is applied for acquiring textual features that represent the presence of tags of each image. Subsequently, in the second module, PF-ART categorizes the images and simultaneously learns the probabilistic distribution of tag occurrences of each category. In addition, user preferences can be incorporated for improving the clustering quality and the degree of topic mining in the final results. The probability distribution is then used to identify the potential key tags (i.e. themes) which constitute the theme lexicons for the respective categories. In the last module, cluster semantic relevance (CSR) is used for evaluating the semantic relevance between categories and their children categories, such that the merging strategy may determine if two categories can be grouped into a new category or one category should be a child category of the other one. When the highest semantic relevance score of categories reaches the stop

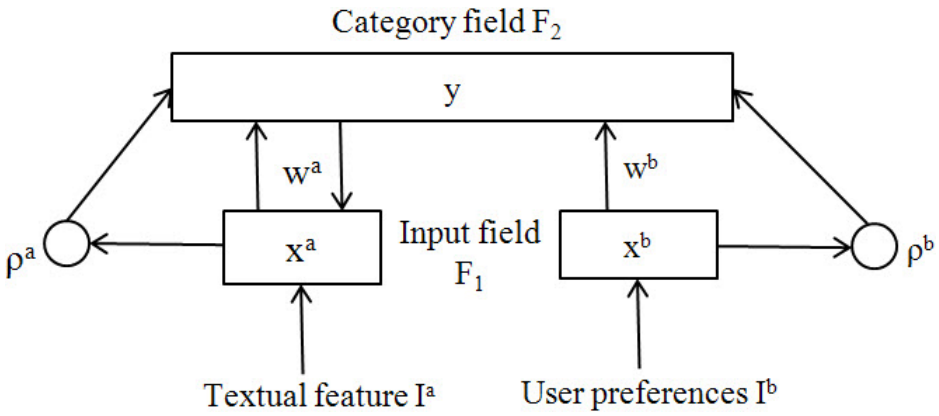


Figure 3.2: The architecture of Probabilistic Fusion ART.

threshold, we can obtain a semantic hierarchy, where the relationship between categories is revealed by a multi-branch tree structure and the themes of father categories are more general than their children. The details of PF-ART and agglomerative merging strategy are described in the following sections.

3.3 Probabilistic Fusion ART

Probabilistic Fusion ART (Fig. 3.2), a variant of Fusion ART [20], employs a two-channel Fusion ART model, where feature vector $\mathbf{I}^a$ encodes the textual features and preference vector $\mathbf{I}^b$ encodes the user-provided tags. Note that the preference vector is only used for incorporating user preferences. $\mathbf{x}^a$ and $\mathbf{x}^b$ are the normalized input vectors in the input fields respectively. $\mathbf{w}^a$ and $\mathbf{w}^b$ are the weight vector of the respective prototypes of the clusters in the category field. Different from Fusion ART, PF-ART models the cluster prototypes/weights by the probabilistic distribution of tag occurrence, because the original learning strategy of ART cannot preserve sub-topics. The details of PF-ART are described in the following sub-sections.

3.3.1 Textual Feature Representation

We construct the textual feature vector based on a textual table, consisting of the distinct tags in the whole image set, expressed by $\mathbf{t} = [t_1, \ldots, t_m]$. We denote the textual feature for the k^{th} image img_k as $\mathbf{t}^k = [t_1^k, \ldots, t_m^k]$, where t_m^k corresponds to the m^{th} tag in $\mathbf{t}$.

Assuming the tag list of img_k is φ_k, the value of t_m^k is given by the impulse response, defined by:

$$t_m^k = \begin{cases} 1 & if\, t_m \in \varphi_k \\ 0 & otherwise \end{cases}.$$

We do not follow traditional methods like tf-idf to weigh each word, because the extracted tags cannot provide sufficient statistical information [4]. In practice, the key tags are usually buried by noise tags which results in feature vectors with a flat distribution and low values.

The feature vector indicates a point in the textual feature space of m dimensions constructed by all tags. Therefore, more common tags in two given images leads to a shorter distance in the feature space of PF-ART.

3.3.2 Similarity Measure

We adopt the two-way similarity measure of Fusion ART [20] to select the best matching category for the input patterns of images. Given an image img_k with its textual feature vector $\mathbf{t}^k$, the similarity measure goes through two steps: 1) category choice and 2) template matching. In the first step, a choice function is applied to evaluate the overall similarity between the input pattern $\mathbf{t}^k$ of the image img_k and the weight vector $\mathbf{w}_j^a$ of each category c_j in the category field. The choice function is defined by

$$T_j = \frac{|\mathbf{t}^k \wedge \mathbf{w}_j^a|}{\alpha + |\mathbf{w}_j^a|}, \tag{3.1}$$

where $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, the norm $|.|$ is defined by $|p| \equiv \sum_i p_i$, and $\alpha \approx 0$ is used to avoid the case when $|w_j| \approx 0$. The weight vector $\mathbf{w}_j^a$ is the textual feature prototype of the j^{th} cluster c_j.

In the second step, the cluster having the highest value of choice function is selected as the winner c_J. Subsequently, we use the match function to evaluate if the similarity between the input pattern $\mathbf{t}^k$ and the weight prototype $\mathbf{w}_j^a$ of c_J meets the lower bound, i.e. the vigilance parameter, which is defined by

$$M_J^a = \frac{|\mathbf{t}^k \wedge \mathbf{w}_J^a|}{|\mathbf{t}^k|} > \rho^a, \tag{3.2}$$

where ρ^a, handling the similarity threshold, is the vigilance parameter for the textual feature channel.

From Equation 3.1, we note that the similarity is the intersection between the input feature vector $\mathbf{t}^k$ and the cluster prototype $\mathbf{w}_j^a$, and the choice function assesses to which degree the prototype $\mathbf{w}_j^a$ is a subset of the input vector $\mathbf{t}^k$. Therefore, if we interpret the feature vector using a histogram, the category choice procedure selects the cluster whose interaction with the input image possesses the biggest proportion of the prototypes.

However, it doesn't mean the winner category fit the input image, because if the prototypes of a given cluster are covered by the features of the input image, this category can also be chosen as winner. Therefore, the similarity measure is not symmetric, so the winner matching procedure is subsequently used to evaluate the fitness of selected cluster for the input image. The vigilance parameter ρ^a determines the lower bound for acceptance. If Equation 3.2 is satisfied, the input image is clustered into the winner category c_J. Otherwise, another winner category is selected from the rest of the clusters for winner matching process. If no fit category is found for the input image, a new category is generated and the prototypes are set by the visual and textual features of the input image.

We can have a further interpretation for the similarity measure. Note that the textual feature vector of an image indicates the presence of tags in the surrounding text of the image by setting the corresponding elements all ones, and the weight prototype of textual features of a cluster is modeled by the frequency distribution. Therefore, the similarity measure tends to evaluate whether the input image contains key tags in the given category, and the hit of more key tags means a better fit.

3.3.3 Learning Strategy for Topic Mining

The original learning function of Fusion ART is defined by

$$\hat{\mathbf{w}}_J^a = \beta(\mathbf{I}^a \wedge \mathbf{w}_J^a) + (1-\beta)\mathbf{w}_J^a, \tag{3.3}$$

where $\beta \in [0,1]$ is the learning rate. Therefore, the cluster prototype learns from the textual features by stably depressing the rare and unstable components while preserving the key and frequent ones. However, a set of mismatch induced by noise tags will erode

the values of the key tags in the prototype. Besides, the sub-key tags cannot be preserved, which may lead to the generation of extra clusters that represent the same topics.

Based on the above consideration, we propose to model the cluster prototype of textual feature by the probabilistic distribution of tag occurrence. In this way, the weights of noisy tags are depressed while the key and sub-key tags can be preserved.

Consider a group of images belonging to cluster c_j, denoted as $Img^j = \{img_1^j, \ldots, img_l^j\}$. By denoting the textual features for img_l^j as $\mathbf{t}^{j,l} = [t_1^{j,l}, \ldots, t_m^{j,l}]$ and the weight vector for textual feature of cluster c_j as $\mathbf{w}_j^a = [w_{j,1}^a, \ldots, w_{j,m}^a]$, the probability of occurrence of the k^{th} tag t_k^j of the given cluster c_j can be calculated by the frequency:

$$w_{j,k}^a = p(t_k^j|c_j) = \frac{\sum_{i=1}^{l} t_k^{j,i}}{l}. \tag{3.4}$$

Therefore, the prototype for the textual features of cluster c_j can be represented by $\mathbf{w}_j^a = [p(t_1^j|c_j), \ldots, p(t_m^j|c_j)]$.

Subsequently, we introduce the sequential factor and denote Equation 3.4 by $p_l(t_k^j|c_j)$ as the state for time l. Assuming a new image is grouped into cluster c_j, the relationship between the states of time l and $l+1$ can be derived by

$$p_{l+1}(t_k^j|c_j) = \frac{\sum_{i=1}^{l+1} t_k^{j,i}}{l+1} = \frac{l}{l+1} p_l(t_k^j|c_j) + \frac{t_k^{j,l+1}}{l+1}. \tag{3.5}$$

Therefore, the general form of learning function for $w_{j,k}^a$ is defined by

$$\hat{w}_{j,k}^a = \frac{n_j}{n_j+1} w_{j,k}^a + \frac{t_k^I}{n_j+1}, \tag{3.6}$$

where n_j is the number of images in cluster c_j, and t_k^I is the k^{th} element of the feature vector of the input image. Considering t_k^I equals either 0 or 1, the learning function for $\mathbf{w}_j^a = [w_{j,1}^a, \ldots, w_{j,m}^a]$ can be further simplified as

$$\hat{\mathbf{w}}_{j,k}^a = \begin{cases} \eta \mathbf{w}_{j,k}^a & t_k^I = 0 \\ \eta(\mathbf{w}_{j,k}^a + \frac{1}{n_j}) & t_k^I = 1 \end{cases}, \quad \eta = \frac{n_j}{n_j+1}. \tag{3.7}$$

3.3.4 Incorporating User Preferences

The above sections describe how PF-ART works in an unsupervised manner, i.e. only the textual features are used for clustering. In this section, the semi-supervised version

Algorithm 3.1 The clustering procedures of PF-ART

Input: Input patterns $\{\mathbf{t}^k\}_{k=1}^{k=N}$, α, β and ρ.

1: Receive user preference for generating pre-defined clusters as initial network. If no user preference, create an uncommitted category with all weight values equal to 1's.
2: Given an input image, present its textual feature vector $\mathbf{t}$ into the input field.
3: For each category c_j in category field F_2, calculate the choice function T_j using Equation 3.1.
4: Identify the winner c_J such that $T_J = \max_{c_j \in F_2}\{T_j\}$.
5: Calculate the match function M_J^a using Equation 3.2.
6: If $M_J^a < \rho^a$, set $T_J = 0$ and go to 3; else, go to 7.
7: If the selected c_J is uncommitted, set $\mathbf{w}_J^a = \mathbf{t}$ and create a new uncommitted node; else, resonance occurs, go to 8.
8: Update the weight vector $\mathbf{w}_J^a$ using Equation 3.7.
9: If all images have been presented, algorithm stops. Otherwise, go to 2.

Output: Cluster Assignment Array $\{A_n|_{n=1}^N\}$.

of PF-ART that employs both the textual feature vector and the user preference vector is presented.

With the incremental nature of Fusion ART, our method receives the user preferences by sending the user-provided relevant images as pre-defined cluster prototypes. Each element of the textual prototype (weight vector) is derived from its frequency of occurrence. The preference vector $\mathbf{I}^b$, as shown in Fig. 3.2, is a channel for encoding user-provided labels. The preference vector does not contribute to the clustering process, but makes a tradeoff for the pre-defined categories, because of two possible cases that may decrease the clustering performance: 1) for two categories that are equal in textual vector, user may give different labels; 2) conversely, for two categories that are different in textual feature, user may give them the same label. For the first case, we combine the user-provided labels and merge them into one category. For the second case, we deem that the two categories of images are the same and represent them in one category whose textual prototype is calculated by the frequency of occurrence. Besides, the user-provided labels represent the key topics of the pre-defined categories and contribute to the generation of semantic hierarchy.

Beside the user-provided information, users can also have a direct control of the clustering results by changing the value of the vigilance parameter ρ^a, as used in Equation 3.2. As aforementioned, the vigilance parameter constrains the dissimilarity between

the images in the same category. As the similarity of textual features directly reflects the common topics, a low vigilance parameter results in a few clusters whose key topics are few and general. In contrast, a high value leads to the generation of relatively more clusters such that the clusters belonging to one general topic are also discriminated due to detailed sub-topics.

3.4 Semantic Hierarchy Generation

After the clustering process, the key tags of each cluster can be extracted as a theme lexicon to represent the underlying topics of the clusters, and each tag should be associated with a weight to indicate its importance in different clusters. As the textual feature prototype represents the probability of tag occurrences, the top valued tags are extracted as key tags and weighed by their respective probability of occurrence. Besides, tags in the preference vector are all considered key tags with weights of 1s. Then, we propose an agglomerative approach for merging the clusters according to their semantic relevance.

3.4.1 Measuring Cluster Semantic Relevance

Given two clusters c_i and c_j, their similarity $S(c_i, c_j)$ can be expressed as the semantic relevance of key tags in their respective theme lexicons denoted as $L_i = \{l_{i,1}, \ldots, l_{i,m}\}$ and $L_j = \{l_{j,1}, \ldots, l_{j,m}\}$. Traditional measures for assessing the semantic similarities between two concepts are usually based on the path length according to a well-structured corpus such as WordNet [148]. But such methods are not suitable for web resources as the diversity of words used for the description of web images. Here, we follow the idea of measuring the semantic similarity of two concepts based on their co-occurrence [5]. We first define the semantic distance between the two tags x and y. Similar to the definition of Google distance [149], the semantic distance is estimated as follows:

$$d(x, y) = \frac{\max(\log f(x), \log f(y)) - \log f(x, y)}{\log G - \min(\log f(x), \log f(y))}, \tag{3.8}$$

where G is the total number of the searched images, $f(x)$ is the number of images returned by Google image search engine using keyword x and $f(x, y)$ is the number of images by searching with both x and y. Then, their semantic relevance is defined by

$$\theta(x, y) = \exp(-d(x, y)), \tag{3.9}$$

where $\theta(x, y) \in [0, 1]$ amd $d(x, y)$ is the semantic distance between the two tags x and y. If two concepts never occur in one image, their semantic distance becomes infinite, so their semantic relevance is 0; conversely, if two concepts always occur in one image, their semantic distance becomes 0, so their semantic relevance is 1. Finally, we define the cluster semantic relevance between categories which can be represented as a sum of the semantic relevance between each tag in c_i and all tags in c_j weighted by their weights in the respective categories:

$$S(c_i, c_j) = \sum_{r=1}^{m} \sum_{k=1}^{n} p_{i,r} p_{j,k} \theta(l_{i,r}, l_{j,k}), \tag{3.10}$$

where $p_{i,r} = w_{i,r}^a$ is the frequency of the r^{th} tag in the category c_i.

3.4.2 Agglomerative Strategy

In the process of having the semantic relevance $S(c_i, c_j)$ for each pair of categories, we simultaneously obtain an upper triangular matrix $v = \{v_{ij}\}$ recording the semantic relevance between pairs of categories, such that

$$v_{ij} = \begin{cases} S(c_i, c_j) & i > j \\ 0 & otherwise \end{cases}. \tag{3.11}$$

For each category c_i, we denote the set of its children categories as ξ_i. Then we define its inner scatter as:

$$\Delta_i = \max\{S(c_p, c_q) - S(c_m, c_n) | c_p, c_q, c_m, c_n \in \xi_i\}. \tag{3.12}$$

The merging process starts by checking if c_j is a child of c_j. Specifically, c_j is a child of c_i if and only if

$$S(c_i, c_j) + \Delta_i \geq \min S(c_p, c_q) | c_p, c_q \in \Delta_i. \tag{3.13}$$

If Equation 3.13 is satisfied, we set c_j's father category as c_i and update the matrix using 3.15. Otherwise, we check if c_i is a child of c_j. If both conditions are not satisfied, a new category c_{new} is generated as the father category of c_i and c_j, assigned with a new lexicon $L_{new} = \{L_i \cup L_j\}$. L_{new} contains all distinct tags in L_i and L_j. Let the k^{th} tag in

L_{new} be the i^{th} tag in L_i and the j^{th} tag in L_j, its weight is determined by the following equation:

$$p_k = \frac{n_i}{n_i + n_j} p_{i,i} + \frac{n_j}{n_i + n_j} p_{j,j} = \alpha p_{i,i} + \beta p_{j,j}, \tag{3.14}$$

where N_i and N_j is the number of images in c_i and c_j respectively. The equation U_i for updating the relevance score of cluster c_i in the semantic relevance matrix is defined by

$$U_i = \begin{cases} \hat{v}_{k,i} = \alpha v_{k,i} + \beta v_{k,j} & k < i \\ \hat{v}_{i,k} = \alpha v_{i,k} + \beta v_{k,j} & i < k < j \\ \hat{v}_{i,k} = \alpha v_{i,k} + \beta v_{j,k} & k > j \end{cases}, \tag{3.15}$$

namely, the semantic relevance between c_{new} and other categories are the weighted average of its children.

We illustrate the merging process using a simplified case of the resulting hierarchy shown in Fig. 3.3. The clusters a and b at the bottom are generated by PF-ART. As their key tags share "animal" and "dog", they are likely to be merged into one cluster. The cluster c is the father cluster of a set of semantically related clusters including clusters a and b. Relevant clusters are merged into a father cluster and we finally obtain a series of clusters with distinct general themes. The leaf categories should be of more constrained semantics than their father categories and the categories with the same father category should have at least one common general theme. For the visualization purpose, top tags of each cluster can be extracted as the cluster name.

Algorithm 3.2 The proposed agglomerative algorithm

Input: Cluster similarity matrix v and stop criterion $\bar{S}$.

1: Select the largest $S(c_i, c_j)$ in v, if $S(c_i, c_j) < \bar{S}$, algorithm stops; else go to 2.
2: Check if c_j is a child of c_i using Equation 3.13, if satisfied, set c_j as a child of c_i and go to 4; else check if c_i is a child of c_j. If satisfied, set c_i is a child of c_j and go to 5; else go to 3.
3: Merge c_i and c_j into c_{new} by merging L_i and L_j into L_{new}. Set $c_i = c_{new}$. Go to 4.
4: Remove the j^{th} row and j^{th} column of v and update v using Equation 3.15. Go to 1.
5: Remove the i^{th} row and i^{th} column of v and update v using Equation 3.15. Go to 1.

Output: The hierarchy of clusters.

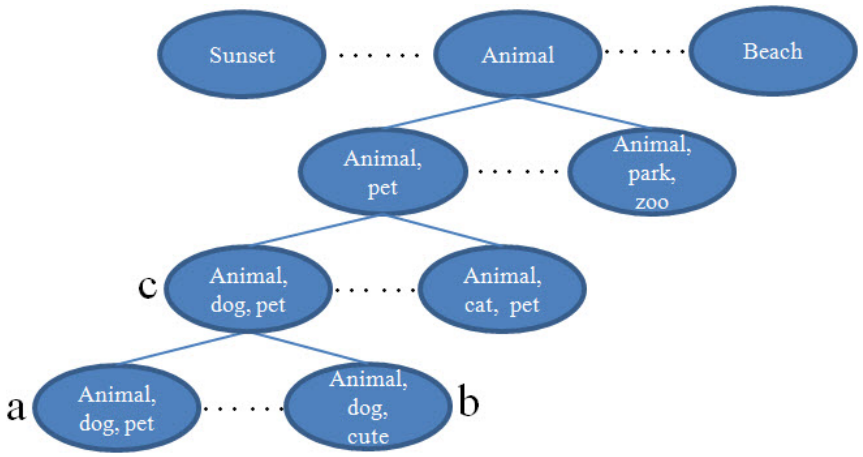


Figure 3.3: A toy example for the generation procedures of semantic hierarchy.

3.5 Experiments

We have conducted experiments on the NUS-WIDE and Flickr data sets to evaluate the performance of the proposed two-step hierarchical clustering method in three aspects: 1) the clustering quality of Probabilistic Fusion ART (PF-ART); 2) the quality of the semantic hierarchy generated by the cluster merging strategy; and 3) time cost of the whole algorithm.

3.5.1 Evaluation Measures

With the understanding that a high quality cluster maximizes the number of images of the same class in it, and an ideal clustering is to group all images of the same class into one cluster, we use precision and F-score score to evaluate the quality of clusters. F-score is defined as

$$F = \frac{2(recall * precision)}{recall + precision}.$$

It has an overall assess of the quality of a cluster where a high value indicates a high quality clustering in terms of both precision and recall.

Besides, as our goal is to discover the key topics of groups of images, we also evaluate the quality of clusters through their cohesion and scatter in terms of key tags, which are assessed by the cluster entropy and class entropy [150]. The cluster entropy of a cluster

c_j is computed by

$$e_{c_j} = -\sum_j \frac{n_{(l_i,c_j)}}{\sum_i n_{(l_i,c_j)}} \log \frac{n_{(l_i,c_j)}}{\sum_i n_{(l_i,c_j)}}, \tag{3.16}$$

where $n_{(l_i,c_j)}$ denotes the number of patterns in cluster c_j with key tag l_i. It evaluates how well a cluster distinguishes images with different topics/key tags. If all of the patterns in a cluster having the same key tags, the cluster entropy is zero.

On the other hand, the class entropy evaluates whether the images having the same topics are represented by a minimal number of clusters. For each key tag l_i in cluster c_j, its class entropy is calculated by

$$\bar{e}_{l_i} = -\sum_i \frac{n_{(l_i,c_j)}}{\sum_j n_{(l_i,c_j)}} \log \frac{n_{(l_i,c_j)}}{\sum_j n_{(l_i,c_j)}}. \tag{3.17}$$

The overall class entropy of cluster c_j is obtained by averaging the class entropies of all the key tags. A low value of class entropy indicates a high recall of images of the same topics.

3.5.2 NUS-WIDE Data Set

The NUS-WIDE data set [2] consists of 269,648 images and ground-truth labels from 81 concepts. The images are downloaded from the famous photo sharing website Flickr.com. We choose this data set because it is the largest well-annotated web image set with filtered surrounding text. To test the clustering performance of our method on large scale image set, we collected 23,284 images of nine categories, including dog (2,504 images), bear (1,271 images), cat (2,376 images), bird (3,780 images), flower (3,000 images), lake (3,000 images), sky (3,000 images), sunset (3,000 images) and wedding (1,353 images), according to its ground truth labels. We choose the nine categories as they are widely used in research works and they are also the most popular tags recorded by Flickr.com.

We construct the texture feature vector by considering all distinctive and high frequency tags. Specifically, we extract all words in the raw text of selected images. After removing the stop-words, misspellings and personalized words, and stemming the variation form of words, there are 3,684 remaining tags. We further filter the infrequency tags which are not in the top 2,000 tags sorted by the tag frequency. Finally, we obtain 1,142 tags (features) and each image is associated with seven tags on average.

3.5.2.1 Performance of Probabilistic Fusion ART

We compare the performance of PF-ART with existing widely used text clustering algorithms, including Fuzzy ART [68], K-means clustering algorithm, Isoperimetric Co-clustering Algorithm (ICA) [38], Non-negative Matrix Factorization (NMF) [47] and Semi-supervised NMF (SS-NMF) [16]. Note that the Fusion ART with one input channel is a Fuzzy ART. Therefore, we compare PF-ART with Fuzzy ART to investigate the effectiveness of the learning function of PF-ART. All algorithms are implemented by C language and experiments are performed on the computer with Intel Core2 Duo CPUs 2.66GHz and 3.25GB of RAM.

To initialize PF-ART and Fuzzy ART, we fix the choice parameter α^a at 0.01 and the learning parameter β^a at 0.6. We test their performance on the nine categories of NUS-WIDE data set in terms of average precision (AP), F-score, the overall cluster entropy e and the overall class entropy entropy $\bar{e}$. For calculating the entropies, the key tags of each cluster are extracted according to top occurrence frequencies. The performance on each category is obtained by averaging the performance of its key clusters in which the number of images of the category is the majority. The overall performance is calculated by averaging the performance of all clusters of the hierarchy. In view that the clustering results of all of the above algorithms depends on a fixed parameter, including the vigilance parameter of PF-ART and Fuzzy ART, the iteration threshold for ICA and the number of generated clusters of K-means, NMF and SS-NMF, we calcuate the final results by averaging the performance under different settings. Specifically, the vigilance parameter ρ^a of PF-ART and Fuzzy ART is set from 0.4 to 0.9, and the iteration of ICA for bi-partitioning and the number of clusters for K-means, NMF and SS-NMF is set from 9 to 20. For the semi-supervised version of PF-ART and SS-NMF, three images of each category are used as user preferences. For a fair comparison, no labels are provided for PF-ART.

The results are shown in Table 3.1 and the best results are bold. PF-ART outperforms others in terms of average precision, F-score and cluster entropy in both unsupervised and semi-supervised cases. The clustering quality of PF-ART has a great improvement after receiving user preferences. Compared with Fuzzy ART, PF-ART without user preferences obtains similar results in terms of precision and cluster entropy, but performs better in

Table 3.1: The clustering performance comparison of different clustering algorithms in terms of average precision (AP), F-score, cluster entropy e and class entropy $\bar{e}$ on NUS-WIDE data set.

NUS-WIDE	K-means	ICA	NMF	SS-NMF	Fuzzy ART	PF-ART	PF-ART (semi)
AP	0.6859	0.7947	0.7412	0.8327	0.7739	0.7832	**0.8636**
$F-score$	0.5748	0.6823	0.6175	0.6917	0.6573	0.7391	**0.7624**
e	0.5882	0.4426	0.4794	0.4027	0.3842	0.3614	**0.3350**
$\bar{e}$	0.4834	0.4177	0.4136	**0.3729**	0.4364	0.3826	0.3764

terms of F-score and class entropy, which indicates a higher recall. The reason should be that Fuzzy ART cannot preserve sub-topics. As with a high vigilance parameter, more clusters are generated due to the mismatch of sub-topics. SS-NMF obtains the best result in class entropy. One possible reason for PF-ART is still the side-effect of noisy tags which increases the difference between the cluster prototype and the input pattern. However, the performance of PF-ART is still comparable to the best result.

3.5.2.2 Performance of the proposed PHTC

As the proposed two-step Personalized Hierarchical Theme-based Clustering approach (PHTC) is an agglomerative clustering algorithm in nature, we compare its performance with four related methods. The first method, referred to as hierarchical theme-based clustering (HTC), directly applies our merging strategy on the input patterns without the clustering step. Specifically, each image is regarded as one cluster with the associated tags as key tags, and then the cluster semantic evaluation and merging strategies are performed to obtain the semantic hierarchy. The second method is the traditional agglomerative method (HC). As different merging strategies vary largely on performance, we test three popular merging strategies termed single-linkage (HC-SL)[31], average-linkage (HC-AL) [32] and complete-linkage (HC-CL) [33]. The third and fourth methods are Hierarchical Fuzzy Clustering (HFC) [151] and the hierarchical clustering algorithm (HCC) used in Hierarchical Comments-based Clustering [34].

We follow the parameter settings of PF-ART in PHTC in the above section. The vigilance parameter ρ^a is fixed at 0.9 and no user preferences are provided to PF-ART. HFC requires two parameters including the number of nearest neighbors k and the number of clusters s for stop criterion. We empirically set $k = 100$ and $s = 19$ according to the size

Table 3.2: The performance of PHTC and other hierarchical clustering algorithms on NUS-WIDE data set.

NUS-WIDE	HTC	HC_SL	HC_AL	HC_CL	HFC	HCC	PHTC
AP	0.6692	0.5977	0.6512	0.5485	0.6309	0.7248	**0.7634**
$F-score$	0.4977	0.5291	0.5047	0.4628	0.4811	0.4631	**0.5883**
e	0.4642	0.4726	0.4873	0.5581	0.5385	0.4468	**0.4434**
$\bar{e}$	0.5258	0.5873	0.5131	0.6127	0.4871	0.5235	**0.4604**
e_{max}	0.5471	0.6272	0.6894	0.7284	0.5813	0.5707	**0.5137**
$\bar{e}_{max}$	0.6963	0.7642	0.6826	0.7535	0.6427	0.7364	**0.6355**
$Time(sec.)$	108.1504	165.5269	182.8592	151.1495	136.4930	86.1498	**32.2217**

and the number of topics of our data set. To make a fair comparison, the stop criterion $s = 19$ is applied to all of the algorithms.

We evaluate the quality of the generated hierarchy by evaluating the quality of all clusters in the hierarchy. Beside the overall cluster and class entropies e and $\bar{e}$, the maximum entropies e_{max} and $\bar{e}_{max}$ are studied for revealing the worst merging in the hierarchy. We also consider the time cost for generating the hierarchy. The results are shown in Table 3.2, from which we observe that PHTC obtains the best results for all evaluation measures. Compared with HTC, the quality of generated hierarchy has a great improvement in terms of average precision and F-score. Note that HTC is a special case of PHTC when $\rho^a = 1$, it demonstrates the effectiveness of PF-ART in grouping semantically similar images and mining the key tags. We also observe that the performance of HTC is comparable to the best results of other algorithms. It indicates that the proposed cluster semantic relevance measure and merging strategy are effective for text clustering. In terms of the time cost, PHTC is much faster than other algorithms. It is due to the rapid nature of PF-ART. Thus, our algorithm is suitable for large image data sets.

3.5.3 Flickr Data Set

To evaluate the robustness of our algorithm for universal web image resources, we conduct experiments on another image set used in [107], which is also crawled from Flickr.com. Although the two data sets are collected from the same website, the images are totally d-ifferent as there is a long interval between the collections of these two data sets. This data

Table 3.3: The clustering performance comparison of different clustering algorithms on Flickr data set.

Flickr	K-means	ICA	NMF	SS-NMF	Fuzzy ART	PF-ART	PF-ART (semi)
AP	0.7559	0.8347	0.8147	0.8793	0.8439	0.8363	**0.8812**
$F-score$	0.6644	0.7025	0.6892	0.7636	0.7350	0.7731	**0.7848**
e	0.3022	0.2491	0.2685	0.2106	0.2411	0.2317	**0.1934**
$\bar{e}$	0.4454	0.4136	0.4329	0.3801	0.4284	0.4013	**0.3704**

Table 3.4: The performance of PF-ART and other hierarchical clustering algorithms on Flickr data set.

Flickr	HTC	HC_SL	HC_AL	HC_CL	HFC	HCC	PHTC
AP	0.7269	0.6168	0.6823	0.5578	0.6581	0.7155	**0.8128**
$F-score$	0.5366	0.5594	0.4989	0.5168	0.5267	0.4814	**0.6933**
e	0.3942	0.4462	0.4131	0.5083	0.4325	0.3768	**0.2434**
$\bar{e}$	0.4712	0.4296	0.4328	0.4056	0.4203	0.4648	**0.3869**
e_{max}	0.5714	0.6420	0.6359	0.6821	0.5329	0.4966	**0.4137**
$\bar{e}_{max}$	0.6424	0.6342	0.6593	0.6684	0.6341	0. 6139	**0.5944**
$Time(sec.)$	41.8445	48.4286	59.8663	51.3578	46.8524	36.3776	**22.9314**

set contains 11,589 images of 19 categories (animal, baby, beach, birthday, boat, crowd, graduation, museum, night, parade, park, people, picnic, playground, show, skiing, sport, sunset and wedding) and each image is associated with filtered textual description (i.e. tags). Therefore, they can be seen as two different resources. In total, 894 tags are extracted and each image is associated with six tags on average.

Similar to the experiments on NUS-WIDE data set, PF-ART with user preferences, in Table 3.3, achieves the best performance in terms of all evaluation criteria and has an great improvement, compared with the unsupervised one, on average precision. We can also observe that PF-ART without user preferences obtains comparable performance with the best results of other unsupervised methods.

Table 3.4 shows similar results with that observed in Table 3.2. PHTC outperforms other hierarchical methods in both clustering quality and time cost. Interestingly, it is shown that the number of images of the NUS-WIDE is twice as much as that of the Flickr data set, while the time cost is only one and a half times. It demonstrates that the mining of semantic groups can enhance the quality of generated hierarchy and reduces the computational cost. Therefore, PHTC is scalable and efficient for large image collections.

A snapshot of the resulting hierarchy in the experiment is shown in Fig. 3.4. Each folder denotes a cluster and the folder name includes the top tags of that cluster. A better interface can be achieved by simple post-processing such as the name pruning of sub-clusters.

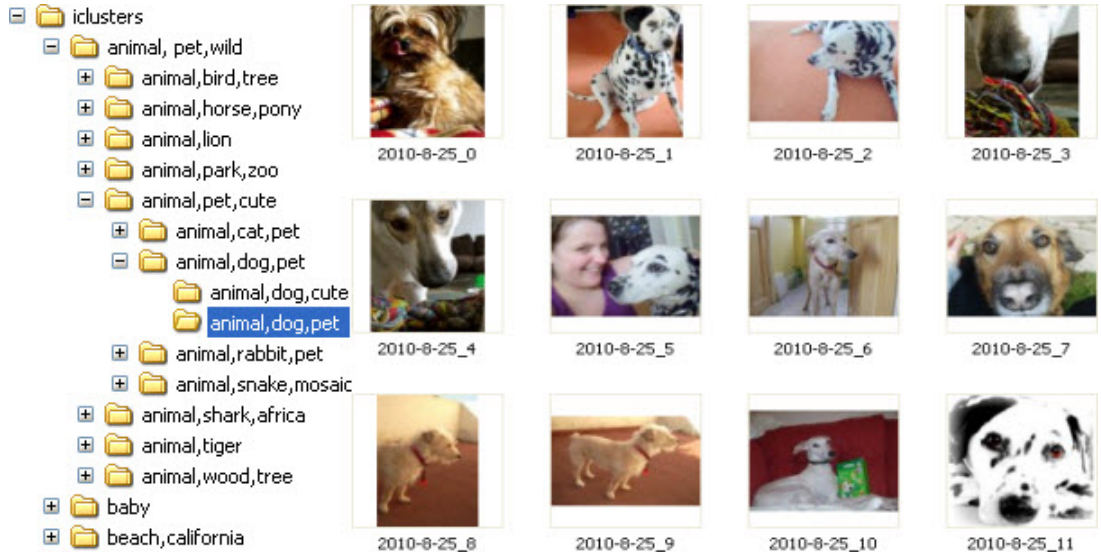


Figure 3.4: A snapshot of the generated hierarchy on Flickr data set.

Chapter 4

Semi-Supervised Heterogeneous Fusion for Multimedia Data Co-Clustering

Heterogeneous data co-clustering is a commonly used technique for tapping the rich meta-information of multimedia web documents, including category, annotation, and description, for associative discovery. However, most co-clustering methods proposed for heterogeneous data do not consider the representation problem of short and noisy text and their performance is limited by the empirical weighting of the multi-modal features. In this chapter, we propose a generalized form of Heterogeneous Fusion Adaptive Resonance Theory (HF-ART), called GHF-ART, for co-clustering of large-scale web multimedia documents. By extending the two-channel Heterogeneous Fusion ART to multiple channels, GHF-ART is designed to handle multimedia data with an arbitrarily rich level of meta-information. For handling short and noisy text, GHF-ART employs the representation and learning methods of PF-ART described in Chapter 3, which identify key tags for cluster prototype modeling by learning the probabilistic distribution of tag occurrences of clusters. More importantly, GHF-ART incorporates an adaptive method for effective fusion of multi-modal features, which weights the features of multiple data sources by incrementally measuring the importance of feature modalities through the intra-cluster scatters. Extensive experiments on two web image data sets and one text document set have shown that GHF-ART achieves significantly better clustering performance and is much faster than many existing state-of-the-art algorithms.

4.1 Introduction

The increasingly popularity of social networking websites, such as Flickr and Facebook, has led to the explosive growth of multimedia web documents sharing online. In order to provide easy access for users to browse and manage large-scale repositories, effective organization of those documents with common subjects is desired. Clustering techniques, designed to identify groupings of data in multi-dimensional feature space based on measured similarity, are often applied to this task. As web multimedia resources are often attached with rich meta-information, for example, category, annotation, description, images and surrounding text, how to utilize the additional information to enhance the clustering performance poses a challenge to traditional clustering techniques.

In recent years, the heterogeneous data co-clustering approach, which advances from the clustering of one data type to the co-clustering of multiple data types, has drawn much attention and been applied to the image and text domains [10, 12, 30, 84, 85]. However, the algorithms follow the similar idea of linearly combining the objective functions of each feature modality and subsequently minimizing the global cost. For the co-clustering of multimedia data, existing algorithms face three challenges elaborated as follows. Firstly, similar to the short text clustering problem [4], meta-information is usually very short and therefore the extracted tags cannot be effectively weighted by traditional data mining techniques such as term frequency-inverse document frequency (tf-idf). Secondly, the weights of features in the objective function still rely on empirical settings, which usually leads to a sub-optimal result. Finally, this approach requires an iterative process to ensure the convergency, which leads to high computational complexity. Thus, existing methods are only applicable to small data sets consisting of up to a thousand of documents but become very slow and not scalable to big data.

In view of the above issues, a self-organizing neural network, called Heterogeneous Fusion Adaptive Resonance Theory (HF-ART) [152], has been recently proposed for web image co-clustering, which performs fusion of visual and textual features as a mapping across two feature spaces. HF-ART achieves effective representation of the surrounding text by modeling the cluster prototype of textual features using probabilistic distribution of tag occurrences, and addresses the problem of feature weighting by employing a robustness measure to weight the features by learning from the intra-cluster scatters. Moreover,

HF-ART is semi-supervised as it is able to take in prior knowledge by initializing the network with pre-defined clusters, indicating regions of interests to users. Different from traditional semi-supervised clustering techniques, such as [10], in which the user-provided knowledge is rarely reflected by the resulting clusters, HF-ART can incrementally generalize and preserve the learnt knowledge by identifying and learning from relevant input patterns, and present the resulting clusters, reflecting user preferences, directly to the users.

Whereas HF-ART is restricted to two pattern channels, in this study, we propose a generalized heterogeneous data co-clustering algorithm, termed Generalized Heterogeneous Fusion Adaptive Resonance Theory (GHF-ART), for fast and robust web multimedia data co-clustering. By extending HF-ART from a two-channel model to multiple feature channels wherein each channel may receive different types of data patterns, GHF-ART is designed to handle multimedia data with an arbitrarily rich level of meta-information. Accordingly, the adaptive feature weighting algorithm has also been generalized by evaluating a robustness measure for each of the multiple feature channels.

The performance of GHF-ART has been evaluated on two public web image data sets, namely the NUS-WIDE [2] and Corel data sets, and a public text document set, known as the 20 Newsgroups data set [153]. Our empirical results show that GHF-ART consistently achieves better cluster quality and is much faster than many state-of-the-art heterogeneous data co-clustering algorithms.

4.2 Problem Formulation

Considering a set of documents $\mathcal{D} = \{doc_n|_{n=1}^{N}\}$ with the associated meta-information, which may be tags, category information and surrounding text, each document doc_n may be represented by a multi-channel input pattern $\mathbf{I} = \{\mathbf{x}^k|_{k=1}^{K}\}$, where $\mathbf{x}^k$ is a feature vector extracted from the document or one type of meta-information. The goal of the heterogeneous data co-clustering task, as defined in this study, is to partition the set of N documents into a set of clusters $\mathcal{C} = \{c_j|_{j=1}^{J}\}$ by evaluating the similarity between the input patterns of the documents according to their corresponding feature vectors such that the documents belonging to the same cluster should be more similar to each other

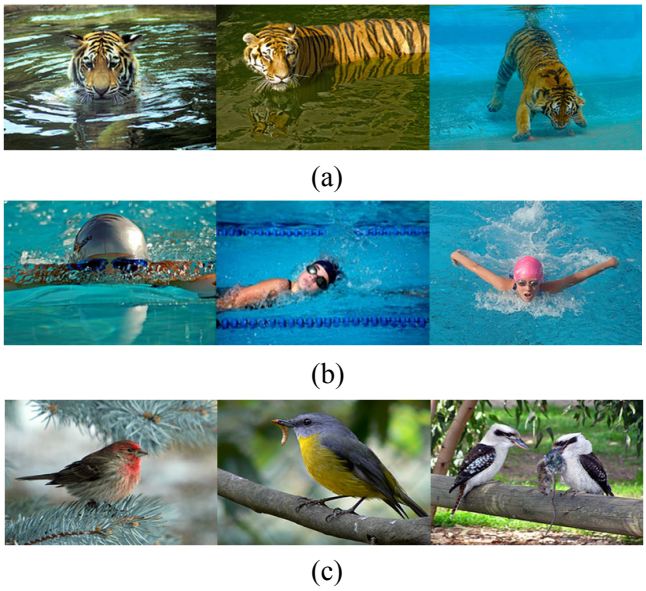


Figure 4.1: Examples of web images sharing similar visual content and high-level semantics.

than to the documents of the other clusters. For example, in the image domain, the co-clustering task may be to identify similar images according to both the visual content and the surrounding text. In each cluster, the images therein are similar in image content and the high-level semantics reflected from the image content are consistent. Similarly, in the text domain, the co-clustering task is to consider both the features of the text document and the meta-information, such as category information and authors.

As reviewed in the previous section, the heterogeneous data co-clustering task presents a number of issues and challenges, especially for multimedia data set. We discuss the key challenges in three aspects as follows.

(i) **Representation of document content:** The representation issue of text documents has been well studied in literature. Typically, text documents are represented by the keywords appearing in the document collection, each of which is weighted based on its frequency in and cross the documents, known as tf-idf. On the other hand, visual representation of images is still a challenge nowadays. Current techniques for visual feature extraction are based on color histogram, edge detection, texture orientation and scale-invariant points so that the visual features

are inadequate to represent the images at the semantic level, a problem known as semantic gap. It leads to difficulties to group the images with very different appearance (Fig. 4.1(c)) or to distinguish those with similar background (Fig. 4.1(a) and 4.1(b)).

(ii) **Representation of meta-information:** The meta-information of documents provides additional knowledge which indicates the relations between documents from another perspective. However, in both image and text domains, the problem of noisy tags exists. Specifically, although the extracted tags from the meta-information of documents usually contain the key tags that are helpful for identifying the correct groupings of documents, a large number of noisy tags exist which contribute nothing or even indicate incorrect relations between documents. How to identify key tags from noisy text is also an open problem in tag ranking [5, 6].

(iii) **Integrating multiple types of features:** It is the key challenge which is related to heterogeneous data utilization for clustering. Existing works, as described in Section 2.3, typically rely on some global optimization methods for the partitioning of each feature modality. However, they do not address the problem of weighting the feature modalities in their objective functions. Instead, either a uniform weighting or some empirical settings are used, which may not yield the desirable results.

4.3 Heterogeneous Fusion ART

Adaptive Resonance Theory (ART) [154] is a neural theory of cognitive information processing. ART performs unsupervised learning by modeling clusters as memory prototypes and encodes each input pattern incrementally through a two-way similarity measure, which simulates how human brains capture, recognize and memorize information of objects and events. As long as the difference between the input pattern and the selected prototype does not exceed a threshold called vigilance parameter, the input pattern is considered a member of the selected cluster. ART has the advantage of fast and stable learning as well as the incremental manner, and has shown strong noise immunity [72].

By extending ART from a single input field to multiple ones, Fusion ART [20] provides a general architecture for simultaneously learning of multi-modal feature mappings.

Specifically, Fusion ART performs real-time search for suitable clusters and learns to encode the mappings of multi-modal features in an incremental manner. A previous work [95] shows the viability of Fusion ART for integrating visual and textual features for image-text co-clustering. However, its performance is limited by the textual feature representation and learning.

The architecture of HF-ART is a two-channel Fusion ART. Different from Fusion ART, HF-ART employs heterogeneous learning for the features of documents and meta-information respectively to achieve effective cluster prototypes. The proposed learning method models the corresponding cluster prototype by the probability distribution of tag occurrences, which helps to address the problem of noisy tags, especially for data type of which insufficient statistic information is provided. Besides, by employing the robustness measure, the contribution parameter is adaptively adjusted by learning the intra-class scatter of clusters. In this way, the problem of weighting multi-modal features is solved by learning from the cluster structure of input patterns.

4.4 Generalized Heterogeneous Fusion ART

Generalized Heterogeneous Fusion ART (GHF-ART) (Fig. 4.2) extends the HF-ART from two channels to multiple channels so that GHF-ART can be applied to the clustering of more than two modalities wherein each channel may receive a different type of data patterns. More importantly, by generalizing the feature construction methods for multimedia documents and incorporating an adaptive channel weighting algorithm, GHF-ART is able to effectively integrate different types of features across multiple pattern channels. Whereas most current works [10, 12, 30, 85] employ statistical methods, the proposed GHF-ART model performs heterogeneous data co-clustering using a self-organizing neural network. In essence, GHF-ART simultaneously learns the multi-dimensional mappings across multiple feature spaces to the category space. The clustering process of GHF-ART thus partitions the category space into regions of clusters by incrementally learning the cluster prototypes from the input patterns and identifying the key features.

Moreover, with the incremental characteristics of Adaptive Resonance Theory, GHF-ART may perform semi-supervised learning by taking in the user preferences in the form

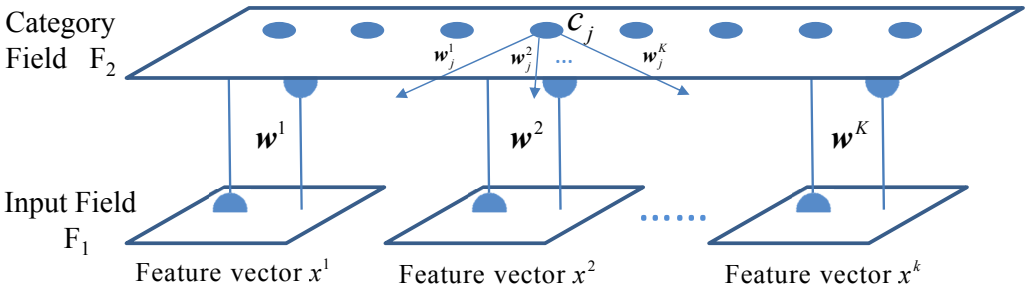


Figure 4.2: The architecture of Generalized Heterogeneous Fusion ART.

of prior knowledge to initialize the cluster structure before clustering. Essentially, the user may identify groupings of documents wherein documents in the same group are deemed to be similar to each other. During the network initialization step, GHF-ART first generates clusters with averaged or other carefully designed cluster prototypes for these user-specified groups of documents. These pre-defined clusters can then be treated as user-defined projection from the feature space to the category space. During the subsequent clustering process, these user-defined clusters can be further generalized by recognizing and learning from similar input patterns, while new clusters can still be created automatically for novel patterns dissimilar to existing clusters. By incorporating user preferences, the predefined clusters help to construct better cluster structure comparing to one using pure data driven clustering.

The dynamics of GHF-ART algorithm is summarized as follows.

Input vectors: Let $\mathbf{I} = \{\mathbf{x}^k|_{k=1}^K\}$ denote the multi-channel input pattern, where $\mathbf{x}^k$ is the feature vector for the kth feature channel. Note that, with complement coding [68], $\mathbf{x}^k$ is further augmented with a complement vector $\bar{\mathbf{x}}^k$ such that $\bar{\mathbf{x}}^k = 1 - \mathbf{x}^k$ in the input field F_1.

Weight vectors: Let $\{\mathbf{w}_j^k|_{k=1}^K\}$ denote the weight vectors associated with the jth cluster c_j in the category field F_2.

Parameters: The GHF-ART's dynamics is determined by choice parameter $\alpha > 0$, learning parameter $\beta \in [0, 1]$, contribution parameters $\gamma^k \in [0, 1]$ and vigilance parameters $\rho^k \in [0, 1]$ for $k = 1, ..., K$.

The clustering process of GHF-ART comprises four key steps: 1) Network initialization: if the user preferences are provided, generate a cluster for each group of documents.

Specifically, each cluster c_j has K weight vectors $\{\mathbf{w}_j^k|_{k=1}^K\}$, obtained by averaging the values of the feature vectors of the documents it contains. Otherwise, generate an uncommitted cluster with all the feature values of its weight vectors set to 1's; 2) Category choice: for each input pattern $\mathbf{I} = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\}$, select the most suitable cluster (winner cluster) c_{j^*}, which has the maximum score calculated by a choice function $T(c_j, \mathbf{I})$ $(j = 1, \ldots, J)$; 3) Template matching: evaluate the similarity between the input pattern $\mathbf{I}$ and the winner c_{j^*} using a match function $M(c_{j^*}, \mathbf{I})$ and a vigilance parameter ρ. If the winner satisfies the vigilance criteria, a resonance occurs which leads to the learning step. Otherwise, a new winner is selected from the rest of the clusters in the category field. If no winner satisfies the vigilance criteria, a new cluster is generated to encode the input pattern; and 4) Prototype learning: if c_{j^*} satisfies the vigilance criteria, its corresponding weight vectors $\mathbf{w}_{j^*}^k$ $(k = 1, \ldots, K)$ are updated through the respective learning functions (see Section 4.4.3). The algorithm stops when all the input patterns are presented.

4.4.1 Feature Extraction

4.4.1.1 Feature Extraction for Document Content

In our work, a document can be either an image or an article. For an image, the feature vector is the concatenation of multiple types of visual features. For an article, we extract the term frequency-inverse document frequency (tf-idf) features. Since the ART-based algorithm requires the input values to be in the interval of [0,1], we further apply min-max normalization on the feature values.

4.4.1.2 Feature Extraction for Meta-Information

As the meta-information (e.g. the surrounding text for a web image or the author information for an article) is usually short and noisy, traditional text mining techniques cannot effectively weight the tags. For example, the tf-idf features usually lead to feature vectors with a flat distribution of low values [4]. Therefore, we model the textual features to indicate the presence of tags such that the probabilistic distribution of tag occurrences in the given clusters can be subsequently learnt as the cluster prototype of textual features through the proposed learning function defined by Equation 4.9.

We first construct the textual feature vector for the meta-information based on a textual table consisting of all distinct tags in the whole image set expressed by $\mathcal{G} = \{g_1, \ldots, g_M\}$. Subsequently, we denote the textual feature vector for the nth document doc_n as $\mathbf{t}_n = [t_n^1, \ldots, t_n^M]^\top$, where t_n^m corresponds to the mth tag g_m in $\mathcal{G}$. The value of t_n^m is given by:

$$t_n^m = \begin{cases} 1, & if\ g_m \in doc_n \\ 0, & otherwise \end{cases}. \tag{4.1}$$

The feature vector indicates a point in the textual feature space of M dimensions constructed by all tags. Therefore, more common tags in two given images lead to a shorter distance in the feature space of GHF-ART.

4.4.2 Similarity Measure

We adopt the similarity measure of Fusion ART [20] to select the best matching cluster for the input pattern. Considering a document doc_n with its corresponding multi-channel input pattern $\mathbf{I} = \{\mathbf{x}^1, ..., \mathbf{x}^K\}$, the cluster selection process consists of two stages, namely category choice and template matching. In the first step, a choice function is applied to evaluate the overall similarity between the input pattern and the template pattern of each cluster in the category field. Specifically, the choice function for each cluster c_j is defined by

$$T(c_j, \mathbf{I}) = \sum_{k=1}^{K} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha + |\mathbf{w}_j^k|}, \tag{4.2}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the ℓ_1 norm $|.|$ is given by $|\mathbf{p}| = \sum_i |p_i|$.

After identifying the cluster having the highest value as the winner c_{j^*}, we use a match function to evaluate if the similarity between the input pattern $\mathbf{I}$ and the winner c_{j^*} meets the vigilance criteria. The match function, for the kth feature channel, is defined by

$$M(c_{j^*}, \mathbf{x}^k) = \frac{|\mathbf{x}^k \wedge \mathbf{w}_{j^*}^k|}{|\mathbf{x}^k|}. \tag{4.3}$$

If, for all the K feature channels, the corresponding match function satisfies the vigilance criteria $M(c_{j^*}, \mathbf{x}^k) \geq \rho^k$ $(k = 1, ..., K)$, a resonance occurs and the input pattern is categorized into the winner cluster. Otherwise, a reset occurs to select a new winner from the rest of the clusters in the category field.

Property 1 *Using the category choice and template matching functions, each input pattern is categorized into the cluster with the best matching feature distribution.*

Proof: From Equation 4.2, we observe that, for each feature channel k, the similarity is calculated by the ratio of the intersection $|\mathbf{x}^k \wedge \mathbf{w}_j^k|$ and the corresponding cluster prototype $|\mathbf{w}_j^k|$. If we interpret the feature vector using a histogram, the most similar feature distribution produces the largest value of $\frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha + |\mathbf{w}_j^k|}$. Taking into account all of the feature channels, the choice function measures the overall similarity between the input pattern $\mathbf{I}$ and the cluster c_j across all of the K feature channels. Thus, the category choice procedure selects the cluster whose feature distribution across all features is the most similar to that of the input pattern.

Subsequently, the template matching procedure defined by Equation 4.3 evaluates if the selected winner matches well with the feature distribution of the input pattern, controlled by the vigilance parameter ρ^k. With a reasonable setting of ρ^k, the clusters that do not match the feature distribution of the input pattern are rejected.

If all the existing categories are not fit for the input pattern, a new cluster is generated and the prototypes are set by the features of the input pattern. In this way, each input pattern will be grouped into the best matching cluster.

4.4.3 Learning Strategies for Multi-Modal Features

4.4.3.1 Learning key features of document content

We use the learning function of Fusion ART [20] to learn the cluster prototype for the document content. Given an input document with its multi-channel input pattern $\mathbf{I} = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\}$ and the winner cluster c_{j^*}, if $\mathbf{x}^k$ is the feature vector for document content, then the learning function for the corresponding weight vector $\mathbf{w}_{j^*}^k$ is defined by

$$\hat{\mathbf{w}}_{j^*}^k = \beta(\mathbf{x}^k \wedge \mathbf{w}_{j^*}^k) + (1 - \beta)\mathbf{w}_{j^*}^k, \tag{4.4}$$

Property 2 *The learning function defined by Equation 4.4 incrementally identifies the key features from the input patterns.*

Proof: The learning function defined by Equation 4.4 consists of two components $\mathbf{x}^k \wedge \mathbf{w}_{j^*}^k$ and $\mathbf{w}_{j^*}^k$, in which the first component is the intersection between the input pattern and the cluster prototype and the second one is the cluster prototype. We observe that whatever the value of the learning rate β, the value of the new cluster prototype, for each element of the feature vector, will not exceed the old one. That is, if the elements of the feature vector are inconsistent in values, the prototype learns a small value. In this way, the cluster prototype learns from the input pattern by suppressing the inconsistent features while preserving the key and consistent ones.

4.4.3.2 Learning key features of meta-information

Directly learning the key tags of clusters from individual documents represented by traditional weighting techniques is usually biased by the limited tag lexicon and statistical information. Based on the above consideration, we propose to model the cluster prototype of textual features by the probabilistic distribution of tag occurrences. In this way, the weights of noisy tags are depressed while the key and sub-key tags can be preserved.

Assuming the winner c_{j^*} contains L documents, denoted as $c_{j^*} = \{doc_1, \ldots, doc_L\}$. Recall that in Section 4.4.1.2, we denote the feature vector for the meta-information of doc_l as $\mathbf{t}_l = [t_l^1, \ldots, t_l^M]^\top$, so the weight vector for the kth feature channel of cluster c_{j^*} can be represented as $\mathbf{w}_{j^*}^k = [w_{j^*,1}^k, \ldots, w_{j^*,M}^k]^\top$. Then, the probability of occurrences of the mth tag in $\mathcal{G}$ in the winner cluster c_{j^*} having L documents is calculated by:

$$w_{j^*,m}^k = p_L(g_m|c_{j^*}) = \frac{\sum_{l=1}^{L} t_l^m}{L}. \tag{4.5}$$

Therefore the prototype for the textual features of cluster c_{j^*} can be represented by

$$\mathbf{w}_{j^*}^k = [p_L(g_1|c_{j^*}), \ldots, p_L(g_M|c_{j^*})]^\top. \tag{4.6}$$

Now we introduce the sequential factor. We treat $p_L(g_m|c_{j^*})$ in Equation 4.5 as the state for time L. Assuming a new document doc_{L+1} is grouped into c_{j^*}, we derive the relationship between the probabilities of occurrence of the mth tag at time L and $L+1$ by

$$p_{L+1}(g_m|c_{j^*}) = \frac{\sum_{l=1}^{L+1} t_l^m}{L+1} = \frac{L}{L+1} p_L(g_m|c_{j^*}) + \frac{t_{L+1}^m}{L+1}. \tag{4.7}$$

Therefore, the general form of learning function for $w^k_{j^*,m}$ is defined by

$$\hat{w}^k_{j^*,m} = \frac{L}{L+1} w^k_{j^*,m} + \frac{t^m_{L+1}}{L+1}. \tag{4.8}$$

Considering t^m_{L+1} equals either 0 or 1, we further simplify the learning function for $\mathbf{w}^k_{j^*} = [w^k_{j^*,1}, \ldots, w^k_{j^*,M}]^\top$, such that

$$\hat{w}^k_{j^*,m} = \begin{cases} \eta w^k_{j^*,m}, & if\ t^m_{L+1} = 0 \\ \eta(w^k_{j^*,m} + \frac{1}{L}), & otherwise \end{cases}. \tag{4.9}$$

where $\eta = \frac{L}{L+1}$.

4.4.4 Self-Adaptive Parameter Tuning

The settings of vigilance parameter ρ and contribution parameter γ affect the clustering results greatly. Using some fixed values will certainly limit the robustness of GHF-ART for a diverse range of data sets. Therefore, self-adaptive tuning of the two parameters is desirable.

4.4.4.1 Match Tracking Rule

The original match tracking rule was first used in ARTMAP [69] to maximize generalization with a minimum number of cluster nodes. GHF-ART utilizes a generalized form of match tracking rule, wherein the vigilance value of each feature channel can be adapted.

At the beginning of each input pattern presentation, the vigilance parameters of all feature channels $\{\rho^1, ..., \rho^K\}$ are set to a baseline ρ_0. A change in the vigilance values is triggered when the template matching process causes a reset. The process is formalized as:

$$\hat{\rho}^k = M(c_{j^*}, \mathbf{x}^k) + \varepsilon. \quad (k = 1, ..., K) \tag{4.10}$$

where $\varepsilon > 0$ is a very small value and $M(c_{j^*}, \mathbf{x}^k)$ is defined as in Equation 4.3.

4.4.4.2 Robustness Measure of Features

The contribution parameter specifies the weighting factor given to each feature channel during the category choice process. Intuitively, the feature channel which is more robust

in distinguishing the classes of the patterns should have a higher weight. Therefore, we want to scale the robustness of the feature channels by learning from the input patterns rather than following an empirical setting. In view that a robust feature channel represents the documents belonging to the same class stably, namely with a small scatter in the cluster weights, it can be measured by the difference between the intra-cluster patterns and the cluster prototypes (weights). Consider a cluster c_j and the intra-cluster documents $\{doc_1,\ldots, doc_L\}$. By denoting the features vectors of doc_l as $\mathbf{I}_l = \{\mathbf{x}_l^1,..., \mathbf{x}_l^K\}$ for $l = 1, ..., L$ and the weight vectors of the cluster c_j as $\{\mathbf{w}_j^1, ..., \mathbf{w}_j^K\}$, we define the Difference for the kth feature vector in c_j as follows:

$$D_j^k = \frac{\frac{1}{L}\sum_l |\mathbf{w}_j^k - \mathbf{x}_l^k|}{|\mathbf{w}_j^k|}. \tag{4.11}$$

Subsequently, the overall difference of one feature vector can be evaluated by averaging the difference of all clusters, defined by:

$$D^k = \frac{1}{J}\sum_j D_j^k, \tag{4.12}$$

where J is the number of clusters. Therefore, the robustness of the kth feature modality can be measured by

$$R^k = \exp(-D^k). \tag{4.13}$$

When D^k is 0, R^k becomes 1, which means that this feature can well represent the patterns belonging to the same class. In contrast, when D^k is very large, R^k approaches zero. The expression implies that the feature with higher difference is not robust and has lower reliability. Thus, in a normalized form, the contribution parameter γ^k for the kth feature channel can be expressed by

$$\gamma^k = \frac{R^k}{\sum_{k=1}^K R^k}. \tag{4.14}$$

This equation shows the rule for tuning the contribution parameter during the clustering process. Initially, the contribution parameter is given by equal weights based on the intuition that the powers of all features are the same. Subsequently, the value of γ^k changes along with the encoding of input patterns.

The tuning of contribution parameters occurs after each resonance, i.e. the clustering epoch for each input pattern, which can be computationally expensive. For efficiency purpose, we further derive a method to incrementally update the contribution parameter values, according to the learning functions defined in Equation 4.4 and Equation 4.9. We consider the update equations in two cases:

- **Resonance in existing cluster**: Assuming the input pattern is assigned to an existing cluster c_j. In this case, only the change of D_j^k should be considered. For the k-th feature channel, the update equations for document content and meta-information are defined by Equation 4.15 and Equation 4.16 respectively:

$$\hat{D}_j^k = \frac{\eta}{|\hat{\mathbf{w}}_j^k|}(|\mathbf{w}_j^k|D_j^k + |\mathbf{w}_j^k - \hat{\mathbf{w}}_j^k| + \frac{1}{L}|\hat{\mathbf{w}}_j^k - \mathbf{x}_{L+1}^k|) \tag{4.15}$$

$$\hat{D}_j^k = \frac{\eta}{|\hat{\mathbf{w}}_j^k|}(|\mathbf{w}_j^k|D_j^k - |\hat{\mathbf{w}}_j^k - \eta\mathbf{w}_j^k| + \frac{1}{L}|\hat{\mathbf{w}}_j^k - \mathbf{x}_{L+1}^k|). \tag{4.16}$$

  After the update for all of the feature channels, the new contribution parameter can then be obtained by calculating Equation 4.12 - Equation 4.14. In this way, the computational complexity reduces from $O(n_i n_f)$ to $O(n_f)$, where n_f denotes the dimension of the feature channels and n_i denotes the number of documents.

- **Generation of new cluster**: When generating a new cluster, the differences of other clusters remain unchanged. Therefore, it just introduces a proportionally change of the robustness. Considering the robustness R^k $(k = 1, ..., K)$ for all of the feature channels, the update equation for the kth feature channel is derived as:

$$\hat{\gamma}^k = \frac{\hat{R}^k}{\sum_{k=1}^K \hat{R}^k} = \frac{(R^k)^{\frac{J}{J+1}}}{\sum_{k=1}^K (R^k)^{\frac{J}{J+1}}}, \tag{4.17}$$

4.4.5 Summary of GHF-ART Algorithm

The complete algorithm of GHF-ART is summarized as follows. First, in step 1, the network structure of GHF-ART is initialized by either pre-defined clusters or a uncommitted cluster. Second, when an input pattern is presented, a best-matching cluster, called the winner, is selected from the category field using the choice function, as described in steps

Algorithm 4.1 Clustering algorithm of GHF-ART

Input: Documents $\mathcal{D} = \{doc_n|_{n=1}^N\}$, α, β, ρ_0.

1: Generate pre-defined clusters for the initial network based on user preferences. If no prior knowledge is received, create an uncommitted cluster with all weight vectors containing 1's.
2: For each document, present its corresponding input pattern $\mathbf{I} = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\}$ into the input field F_1.
3: For each cluster c_j in the category field F_2, calculate the choice function $T(c_j, \mathbf{I})$ defined in Equation 4.2.
4: Identify the winner c_{j^*} with the largest value of the choice function such that $j^* = \arg\max_{j:c_j \in F_2} T(c_j, \mathbf{I})$.
5: Calculate the match function $M(c_{j^*}, \mathbf{x}^k)$ $(k = 1, ...K)$ defined in Equation 4.3.
6: If $\exists k$ such that $M(c_{j^*}, \mathbf{x}^k) < \rho^k$, set $T(c_{j^*}, \mathbf{I}) = 0$, update ρ^k $(k = 1, ..., K)$ according to Equation 4.10, go to 4; else, go to 7.
7: If the selected c_{j^*} is uncommitted, set each cluster prototype to the corresponding feature vector of the input pattern such that $\mathbf{w}_{j^*}^k = \mathbf{x}^k$ $(k = 1, ..., K)$, update γ according to Equation 4.17, and create a new uncommitted node, go to 9; else, go to 8.
8: Update $\mathbf{w}_{j^*}^k$ $(k = 1, ..., K)$ according to Equation 4.4 and Equation 4.9 respectively and update γ according to Equation 4.12 - Equation 4.16.
9: If no input pattern exist, algorithm stops. Otherwise, go to 2.

Output: Cluster Assignment Array $\{A_n|_{n=1}^N\}$.

2-4. Third, the match function is used in step 5 to evaluate if the similarity between the input pattern and the winner satisfies a threshold, called the vigilance criteria. The last steps 6-8 illustrate that GHF-ART will iteratively identify the winner that satisfies the threshold; Otherwise, a new cluster will be created to encode the input pattern. The weight vector of the winner and the corresponding algorithm parameters will be updated accordingly after the categorization of the input pattern. The algorithm stops when all input patterns have been presented.

4.4.6 Time Complexity

The time complexity of GHF-ART depends on the search of suitable categories and the update of contribution parameter. The first step calculates the choice and match function defined in Equation 4.2 and Equation 4.3, which is $O(n_c n_f)$, where n_c denotes the number of clusters and n_f denotes the number of feature dimension of both visual and textual

features. The second step contains two cases: 1) the input pattern is grouped into one of existing clusters; and 2) a new cluster is generated for the input pattern. For the first case, the new contribution parameter is calculated by Equation 4.12 - Equation 4.16. The time complexity of Equation 4.15 and Equation 4.16 is $O(n_f)$ and that of Equation 4.12 - Equation 4.14 is $O(1)$. For the second case, the contribution parameter is updated according to Equation 4.17, whose time complexity is $O(1)$. Assuming there are n_i input patterns, the overall time complexity is $O(n_i n_c n_f)$.

In comparison, the time complexity of CIHC co-cluster algorithm is $O(QR\{n_i n_f\} + (n_i + n_f)\log(n_i + n_f))$, where QR{.} is the time for QR matrix decomposition. The time complexity of NMF is $O(tn_c n_i n_f)$, SRC is $O(t(max(n_i^3, n_f^3) + n_c n_i n_f))$, Comrafs is $O(t(max(n_i^3, n_f^3))$, where t is the number of iterations in the algorithm. We observe that GHF-ART requires the least time cost and maintains a linear increase of running time with respect to the increase in the size of the data set.

4.5 Experiments

4.5.1 NUS-WIDE Data Set

The NUS-WIDE data set [2] is the largest well-annotated web image set with filtered surrounding text, which consists of 269,648 images and their ground-truth annotations from 81 concepts. The images are downloaded from the famous photo sharing website *Flickr.com*. To evaluate the clustering performance of our method on large scale image sets, we collect a total of 23,284 images belonging to nine biggest classes of NUS-WIDE data set, including dog, bear, cat, bird, flower, lake, sky, sunset and wedding, each of which contains nearly 3000 images, except bear (1,271 images) and wedding (1,353 images).

We utilize the visual content and surrounding text of images for clustering. For the visual features, we use a concatenation of Grid Color Moment (225 features), Edge Direction Histogram (73 features) and Wavelet Texture (128 features). We use the above three types of global features as they can be efficiently extracted and have been shown to be effective for image content representation [2]. Finally, each image is represented as a vector of 426 features. We construct the texture feature vector by considering all

distinctive and high frequency tags in the surrounding text of images. After filtering the infrequency tags, we have a total of 1,142 textual features and each image is associated with seven tags on average.

4.5.1.1 Performance of Robustness Measure

In the experiments, we set the choice parameter $\alpha = 0.01$, the learning parameter $\beta = 0.6$ and the baseline vigilance parameter $\rho_0 = 0.1$. Small choice parameter of $\alpha = 0.01$ is commonly used as it has been shown that the clustering performance is generally robust to this parameter [155]. We empirically use $\beta = 0.6$ to tune the cluster weight towards the geometric center of the cluster. In our experiments, the performance of GHF-ART remains roughly the same when the learning parameter changes from 0.3 to 0.8. In view that the vigilance parameter has a direct effect on the number of generated clusters, we use $\rho_0 = 0.1$ which produces a small number of small clusters containing less than 1% of the data patterns. In our experiments, we find that the performance of GHF-ART improves significantly when ρ_0 increases to 0.1. Beyond that, the performance improvement is rather small but the number of clusters increases almost linearly. Therefore, we use $\rho_0 = 0.1$ consistently in all our experiments. Other vigilance values may still work, but a higher vigilance value may lead to a better performance in precision but may create many more clusters resulting in poorer generalization.

We evaluate the performance of robustness measure by comparing the clustering performance of GHF-ART using the self-adapted contribution parameter γ_{SA} with that of fixed values. Since we utilize two channels for visual and textual features respectively, we vary the contribution parameter of textual features and calculate that of visual features by Equation 4.14. The result of average precision weighted by cluster sizes is shown in Fig. 4.3(a). We observe that, without prior knowledge, the self-adaptive tuning method always has comparable performance with the best settings and even slightly improves the results in several classes. The weighted average precision across all classes shows that the overall performance of the robustness measure is slightly better than the best results of the fixed settings of the contribution parameter. Besides, the time cost of GHF-ART with fixed settings is 9.610 seconds and that with the robustness method is 9.832 seconds. Therefore, this method is effective and efficient for solving the tuning problem of contribution parameter and is also scalable to big data.

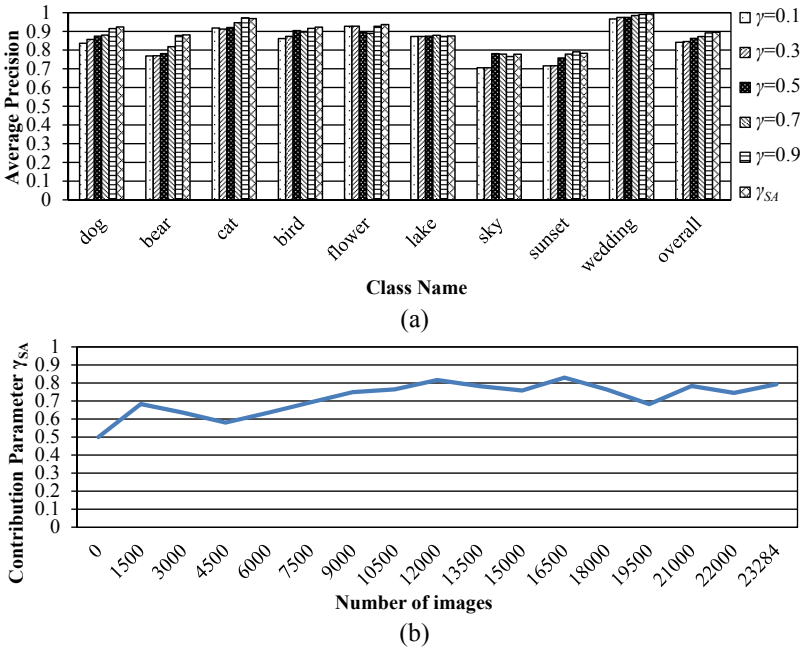


Figure 4.3: (a) Clustering performance using fixed contribution parameters (γ) and self-adapted contribution parameter (γ_{SA}); (b) Tracking of γ_{SA} of textual feature channel on NUS-WIDE data set.

To understand how the robustness measure works, we show the value tracking of γ_{SA} of the textual feature channel in Fig. 4.3(b). We observe that, despite the initial fluctuation, the value of γ_{SA} climbs from 0.5 to 0.8 and then stabilizes in the interval of [0.7, 0.8]. The initial fluctuation should be due to the order of input pattern presentation. As the robustness measure adjusts the contribution parameters along with the learning from input patterns, a large number of images with similar image content or tags may result in such a change in values. However, with the learning from massive input patterns, the value of γ_{SA} becomes stable. It demonstrates the convergency of the robustness measure.

4.5.1.2 Clustering Performance Comparison

We compare the performance of GHF-ART with Fusion ART (the base model of GHF-ART), the baseline algorithm K-means, and existing co-clustering algorithms, namely, CIHC, SRC, Comrafs, NMF and SS-NMF.

To make a fair comparison, since the ART-based algorithms need normalized features, we have conducted experiments to evaluate if the normalized features will benefit the

Table 4.1: Clustering performance on NUS-WIDE data set using visual and textual features in terms of nine classes.

Average Precision	dog	bear	cat	bird	flower	lake	sky	sunset	wedding	Overall
K-means	0.8065	0.7691	0.8964	0.6956	0.7765	0.4873	0.5278	0.5836	0.9148	0.7175
CIHC	0.8524	0.8343	0.9167	0.8942	0.8756	0.6544	0.7466	0.6384	0.9127	0.8139
SRC	0.8184	0.7831	0.8193	0.8302	0.8713	0.6852	0.7132	0.5684	0.8723	0.7735
Comrafs	0.8292	0.6884	0.9236	0.8541	0.8667	0.6719	0.7240	0.6562	0.9065	0.7959
NMF	0.8677	0.8133	0.8623	0.7845	0.8259	0.7848	0.7134	0.6956	0.8648	0.8014
SS-NMF	0.8913	0.8272	0.9149	0.8366	0.8723	0.8213	0.7274	0.7346	0.9174	0.8381
Fusion ART	0.8139	0.7914	0.8500	0.9131	0.8368	0.7448	0.7039	0.6829	0.9653	0.8111
GHF-ART	0.9339	0.8814	0.9685	0.9231	0.9368	0.8755	0.7782	0.7829	0.9932	0.8971
GHF-ART(SS)	**0.9681**	**0.9023**	**0.9719**	**0.9655**	**0.9593**	**0.8864**	**0.8132**	**0.8482**	**0.9961**	**0.9234**

other algorithms. The results show that the performance using the normalized features is similar to those using original features. Thus, we use the original features for the other algorithms. For K-means, we concatenate the visual and textual features and use the Euclidean distance. For K-means, SRC and NMF which require a fixed number of clusters and iterations, we average their performance with different numbers of clusters ranging from 9 to 15 and set the number of iteration to 50. The parameter settings of Fusion ART are the same as those of GHF-ART. For Fusion ART and SRC which need to set the weights for multi-modal features, we use a weight of 0.7, which is the best setting in our empirical study. For the semi-supervised algorithms SS-NMF and GHF-ART(SS), three images of each class are used as user preferences. As CIHC applies ratio cut which only divides the data set into two clusters, we calculate the precision of each class by clustering with each of other classes and calculating the average. As two-class clustering is easier than our nine-class one, the effectiveness of GHF-ART can still be demonstrated if their performance are comparable.

Table 4.1 shows the clustering performance in weighted average precision for each class using the visual content of images and the corresponding surrounding text. We observe that GHF-ART outperforms the others in all cases. K-means usually achieves the worst result especially for the classes “bird”, “lake” and “sky”. The reason should be that the sample mean in the concatenated feature space cannot well represent the common characteristics of features for some classes. CIHC, Comrafs and NMF usually achieve comparable performance and outperform SRC. For the semi-supervised algorithms, we can see that SS-NMF and GHF-ART(SS) achieve better performance than their unsupervised version. Besides, GHF-ART outperforms Fusion ART in all classes, which shows the effectiveness of the proposed methods in addressing the limitations of Fusion ART.

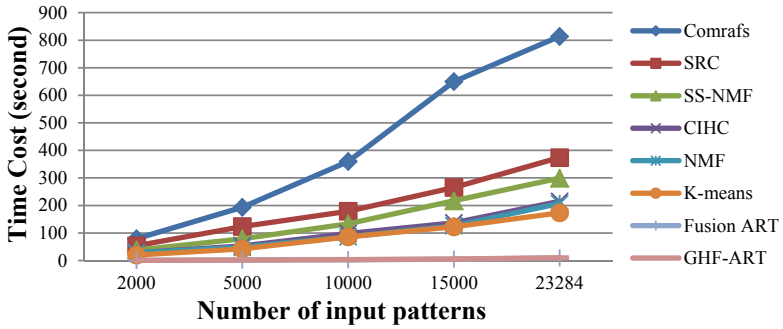


Figure 4.4: Time cost of eight algorithms on NUS-WIDE data set along with the increase in the number of input patterns.

To evaluate the scalability of GHF-ART to big data, we study the time cost of each algorithm with the increase in the number of input patterns. All the algorithms are performed on the computer with 2.66GHz Intel Core2 Duo CPUs and 3.25GB RAM. Since the user preferences for GHF-ART are given before the clustering, the time cost of GHF-ART(SS) is almost the same as that of GHF-ART. As shown in Fig. 4.4, along with the increase in the number of patterns, Comrafs has the highest time cost among all the algorithms. CIHC and NMF have a similar time cost and are slower than K-means. Fusion ART and GHF-ART incur a very small increase of time cost while those of other algorithms increase greatly. Althrough GHF-ART employs the robustness measure, Their time costs are similar. For over 20,000 images, GHF-ART needs less than 10 seconds to complete the clustering process.

To further evaluate the performance of GHF-ART under more complex problems, we run experiments with more classes and noisier data. To this end, we choose nine new classes, including beach, boat, bridge, car, cloud, coral, fish, garden and tree, each of which contains 1500 images. Three classes "car", "cloud" and "tree" are deemed as noisy classes since all the algorithms achieve lower performance. In addition to weighted average precision, we further utilize cluster and class entropies [150], purity [156] and rand index [157] as performance measures. For those algorithms which need a pre-defined number of clusters, we set the number from 18 to 30 and calculate the average performance. For K-means, Fusion ART, GHF-ART and GHF-ART(SS), which are sensitive to initialization, we repeat the experiments for ten times and calculate the means and standard deviations. For the other algorithms that are not sensitive to the initialization, we keep their default settings and report their performances on a single run.

Table 4.2: Clustering results on NUS-WIDE data set with 9 and 18 classes in terms of weighted average precision (AP), cluster entropy ($H_{cluster}$), class entropy (H_{class}), purity and rand index (RI).

	K-means	CIHC	SRC	Comrafs	NMF	SS-NMF	Fusion ART	GHF-ART	GHF-ART(SS)
AP	0.6582 ± 0.036	0.8139	0.7735	0.7959	0.8014	0.8381	0.8047 ± 0.031	0.8663 ± 0.022	$\mathbf{0.9035 \pm 0.016}$
$H_{cluster}$	0.4792 ± 0.037	0.3924	0.4169	0.4386	0.3779	0.3761	0.3744 ± 0.016	0.3583 ± 0.019	$\mathbf{0.3428 \pm 0.013}$
H_{class}	0.5317 ± 0.034	0.4105	0.4462	0.4367	0.4189	0.3922	0.4124 ± 0.024	0.3692 ± 0.018	$\mathbf{0.3547 \pm 0.019}$
Purity	0.7118 ± 0.029	0.8307	0.7891	0.8036	0.8167	0.8498	0.8352 ± 0.027	0.8863 ± 0.018	$\mathbf{0.9085 \pm 0.021}$
RI	0.6291 ± 0.031	0.7806	0.7485	0.7340	0.7615	0.7759	0.7467 ± 0.018	0.7961 ± 0.023	$\mathbf{0.8216 \pm 0.013}$

(a) Clustering on 9 classes

	K-means	CIHC	SRC	Comrafs	NMF	SS-NMF	Fusion ART	GHF-ART	GHF-ART(SS)
AP	0.4528 ± 0.042	0.7739	0.6812	0.6583	0.7209	0.7637	0.7379 ± 0.024	0.7933 ± 0.023	$\mathbf{0.8366 \pm 0.024}$
$H_{cluster}$	0.3892 ± 0.029	0.4161	0.4497	0.4667	0.4018	0.3894	0.4125 ± 0.021	0.3849 ± 0.016	$\mathbf{0.3624 \pm 0.018}$
H_{class}	0.6355 ± 0.024	0.4203	0.4726	0.4639	0.4491	0.4215	0.4378 ± 0.024	0.4109 ± 0.018	$\mathbf{0.3921 \pm 0.019}$
Purity	0.4682 ± 0.033	0.7795	0.6944	0.6727	0.7279	0.7346	0.7193 ± 0.018	0.8054 ± 0.022	$\mathbf{0.8433 \pm 0.023}$
RI	0.4677 ± 0.028	0.7049	0.6728	0.6496	0.7105	0.7488	0.7245 ± 0.022	0.7523 ± 0.012	$\mathbf{0.7681 \pm 0.014}$

(b) Clustering on 18 classes

Table 4.2 shows the results on the original data set with 9 classes and the new data set with 18 classes. In Table 4.2(a), we observe that GHF-ART(SS) achieves the best results in all the evaluation measures in terms of the means. Without supervision, GHF-ART still obtains better performance than all other algorithms. Comparing Table 4.2(b) with Table 4.2(a), we find that all algorithms perform worse when the number of classes increases. This is expected as the increase in the number of classes makes it more difficult to partition the feature spaces. However, GHF-ART still obtains the best results.

To evaluate the statistical significance of performance difference, we conduct t-test among Fusion ART, GHF-ART and GHF-ART(SS). The results show that the performance levels of Fusion ART and GHF-ART are significantly different at 0.05 level of significance in all of the evaluation measures except cluster entropy, of which the difference is at 0.1 level. For GHF-ART and GHF-ART(SS), the difference between their performance in weighted average precision, purity and rand index is significant at 0.05 level of significance. For cluster entropy and class entropy, the performance difference is at 0.1 level.

4.5.1.3 Evaluation on Incremental Property

To evaluate the incremental property of GHF-ART, as described in Section 4.4, we divide the original data set with nine classes into four smaller subsets and apply GHF-ART to them sequentially. Then, we compare the clustering performance of GHF-ART with that

Table 4.3: Clustering performance on the NUS-WIDE data set using the whole set and the subsets.

		dog	bear	cat	bird	flower	lake	sky	sunset	wedding
Whole Set	Average Precision	0.9339	0.8814	0.9685	0.9231	0.9368	0.8755	0.7782	0.7829	0.9932
	# of clusters	3	2	3	4	2	3	3	1	1
Subsets	Average Precision	0.9273	0.9036	0.9512	0.9039	0.9368	0.8622	0.7694	0.8315	0.9967
	# of clusters	2	2	3	3	2	2	3	2	1

for the whole data set. To make a fair comparison, we randomize the sequence of input patterns in all the subsets.

As shown in Table 4.3, we observe that, for all the classes, the number of clusters and the performance on weighted average precision are similar for clustering the whole data set and the subsets. This shows that, given several sequential data sets with random patten sequences, the cluster structure obtained by clustering the whole data set and the subsets are similar. This demonstrates that GHF-ART is able to cluster the new patterns of the updated data set by incrementally adapting the cluster structure learnt from the original data set.

4.5.1.4 Case Study Analysis of Performance

We present a case study to analyze why GHF-ART outperforms other algorithms. Since one major difference between GHF-ART and the other algorithms is the adaptive weighting method of GHF-ART, we evaluate the performance when all the algorithms employ equal weights for the visual and textual features. The results are summarized in Table 4.4. The performance of GHF-ART with adaptive weights (GHF-ART$_{aw}$) is also listed below for comparison.

Table 4.4: Clustering performance on NUS-WIDE data set in terms of weighted average precision (AP) using equal weights to visual and textual features in all the algorithms. GHF-ART$_{ew}$ indicates GHF-ART using equal weights and GHF-ART$_{aw}$ indicates GHF-ART using adaptive weights.

AP	dog	bear	cat	bird	flower	lake	sky	sunset	wedding	Overall
K-means	0.8065	0.7691	0.8964	0.6956	0.7765	0.4873	0.5278	0.5836	0.9148	0.7175
CIHC	0.8524	**0.8343**	0.9167	0.8942	0.8756	0.6544	0.7466	0.6384	0.9127	0.8139
SRC	0.7629	0.7781	0.7667	0.8352	0.8274	0.6903	0.7095	0.5971	0.8566	0.7326
Comrafs	0.8292	0.6884	**0.9236**	0.8541	0.8667	0.6719	0.7240	0.6562	0.9065	0.7959
NMF	0.8677	0.8133	0.8623	0.7845	0.8259	0.7848	0.7134	0.6956	0.8648	0.8014
Fusion ART	0.7960	0.7835	0.8376	0.8891	0.8267	0.7614	0.6850	0.7035	0.9661	0.8037
GHF-ART$_{ew}$	**0.8746**	0.7812	0.9211	**0.9046**	**0.8952**	**0.8748**	**0.7814**	**0.7585**	**0.9746**	**0.8629**
GHF-ART$_{aw}$	0.9339	0.8814	0.9685	0.9231	0.9368	0.8755	0.7782	0.7829	0.9932	0.8971

Comparing with GHF-ART$_{aw}$, the performance of GHF-ART with equal weights (GHF-ART$_{ew}$) has an obvious decrease in most classes, especially for the class "bear". Similarly, the performance of Fusion ART and SRC also have a decrease when using the equal weights. It demonstrates the importance of weighting the feature modalities in clustering. However, GHF-ART$_{ew}$ still obtains the best results in seven out of nine classes.

In addition, suppose we use the learning function of Fuzzy ART instead of the proposed learning method for the meta-information, GHF-ART degenerates to the original Fusion ART. We see that Fusion ART achieves comparable performance with NMF and a little bit lower than CIHC in the overall performance. For specific classes, Fusion ART obtains the best result in "wedding" and usually achieves a comparable performance for the other classes. However, with our proposed meta-information learning method, GHF-ART$_{ew}$ outperforms Fusion ART in most classes and has a relatively big improvement in "lake", "sky" and "cat". This also demonstrates that the proposed learning method of meta-information enables GHF-ART to be robust in handling noisy text.

In comparison, we find all the other algorithms achieve a low level of performance on these noisy classes. This, we reckon, is due to the differences between various methods in handling the patterns. For example, K-means generates hyperspherical clusters in the feature space which are sensitive to noise. Therefore, K-means performs poorly in the noisy classes but obtains comparable performance in classes such as "wedding". CIHC and SRC, which employ spectral clustering, derive eigenvectors from the graph affinity matrices. As such, the noisy features may lead to spurious correlations between patterns. This is why CIHC obtains reasonable performance in all the classes except the three noisy classes. Since SRC employs K-means to get the final clusters, it also suffers from the drawbacks of K-means. NMF derives the cluster indicator matrix from the relational matrices which maps the data into a non-negative latent semantic space. Similar to spectral clustering, noisy features should also be the main reason for the poor performance in the noisy classes. Comrafs performs clustering by finding a cluster structure of patterns that maximizes the Most Probable Explanation based on mutual information. Therefore, noisy features affect the calculation of mutual information and lead to incorrect categorization of patterns.

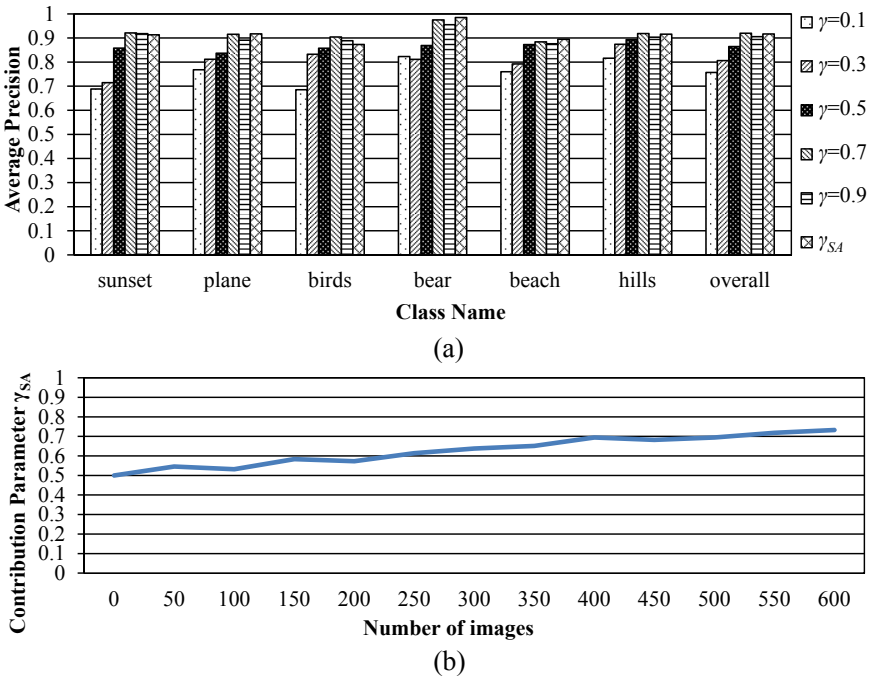


Figure 4.5: (a) Clustering performance using fixed contribution parameters (γ) and self-adapted contribution parameter (γ_{SA}); (b) Tracking of γ_{SA} on Corel data set.

Based on the above analysis, we may conclude that GHF-ART outperforms the other algorithms when the surrounding text is noisy and when the desired weights for different feature modalities are not equal.

4.5.2 Corel Data Set

Corel data set is a subset of Corel CDs data set and consists of 5,000 images from 50 Corel Stock Photo CDs, each of which contains 100 images on the same topic. Each image is annotated by an average of 3-5 keywords from a dictionary of 374 words. We utilize the images of six classes including "sunset", "plane", "birds", "bear", "beach" and "hills". Similar to the NUS-WIDE data set, we extract the 426 visual features and build the textual features using 374 words.

4.5.2.1 Performance of Robustness Measure

Similar to the NUS-WIDE data set, we test the performance of GHF-ART with different settings of contribution parameter of textual features on Corel data set. In Fig. 4.5(a), we

observe that robustness measure achieves the best results for most classes except "sunset" and "birds" and the best overall performance is achieved by $\gamma = 0.7$. However, it still outperforms the other settings and achieves performance very close to the best setting. The value tracking of γ is shown in Fig. 4.5(b). In contrast to that for NUS-WIDE, the result shows a relatively smooth change in the contribution parameter value. The reason should be that the Corel data set contains less noisy tags. We can see that the value gradually increases and stabilizes at $\gamma = 0.7$. It demonstrates that the robustness measure can effectively adjust the contribution parameter to the best setting.

4.5.2.2 Clustering Performance Comparison

Similar to the NUS-WIDE data set, we evaluate the performance of GHF-ART in terms of weighted average precision, cluster and class entropies, purity and rand index. We set the number of clusters ranging from 6 to 15 for those algorithms which need a predefined number of clusters. As shown in Table 4.5, firstly, we observe that all algorithms achieve better clustering performance than that of NUS-WIDE data set. One possible reason is that the visual content of the images belonging to the same category is more similar and the tags of Corel data set is relatively cleaner. We can also see that GHF-ART and GHF-ART(SS) outperform the other algorithms in all the performance measures. Particularly, GHF-ART obtains a mean result close to CIHC and SS-NMF in weighted average precision, cluster entropy, purity and rand index but a much better performance in class entropy. With supervisory information, GHF-ART(SS) has a further improvement over GHF-ART. In addition, GHF-ART has a big improvement over Fusion ART, which demonstrates the effectiveness of our proposed adaptive feature weighting and meta-information learning methods in improving the performance and robustness of Fusion ART.

Table 4.5: Clustering results on Corel data set using visual content and surrounding text.

	K-means	CIHC	SRC	Comrafs	NMF	SS-NMF	Fusion ART	GHF-ART	GHF-ART(SS)
AP	0.7245 ± 0.023	0.8940	0.8697	0.8115	0.8794	0.8960	0.8525 ± 0.027	0.8944 ± 0.018	$\mathbf{0.9168 \pm 0.019}$
$H_{cluster}$	0.3538 ± 0.025	0.2566	0.2714	0.2972	0.2703	0.2667	0.2793 ± 0.022	0.2521 ± 0.018	$\mathbf{0.2366 \pm 0.015}$
H_{class}	0.3816 ± 0.024	0.2614	0.2803	0.3316	0.2771	0.2592	0.2409 ± 0.019	0.2184 ± 0.016	$\mathbf{0.1960 \pm 0.014}$
Purity	0.7263 ± 0.026	0.9031	0.8725	0.8304	0.8862	0.8997	0.8628 ± 0.023	0.8975 ± 0.021	$\mathbf{0.9176 \pm 0.015}$
RI	0.6635 ± 0.024	0.8347	0.8051	0.7734	0.8172	0.8416	0.8116 ± 0.015	0.8342 ± 0.018	$\mathbf{0.8533 \pm 0.014}$

Table 4.6: Clustering results on Corel data set using visual content, surrounding text and category information.

	K-means	CIHC	SRC	Comrafs	NMF	SS-NMF	Fusion ART	GHF-ART	GHF-ART(SS)
AP	0.7254 ± 0.020	0.9014	0.8782	0.8279	0.8865	0.9047	**1**	**1**	**1**
$H_{cluster}$	0.3251 ± 0.026	0.2467	0.2682	0.2543	0.2489	0.2466	**0**	**0**	**0**
H_{class}	0.3688 ± 0.022	0.2544	0.2758	0.3263	0.2709	0.2537	0.1727 ± 0.023	0.1496 ± 0.016	$\mathbf{0.1362 \pm 0.014}$
Purity	0.7284 ± 0.020	0.9106	0.8721	0.8463	0.8917	0.9044	**1**	**1**	**1**
RI	0.6775 ± 0.021	0.8428	0.8147	0.8045	0.8276	0.8315	0.9061 ± 0.019	0.9297 ± 0.021	$\mathbf{0.9485 \pm 0.016}$

Similar to the NUS-WIDE data set, we further conduct t-test between the performance of Fusion ART, GHF-ART and GHF-ART(SS) reported in Table 4.5. The results show that the performance differences between Fusion ART, GHF-ART and GHF-ART(SS) are significant at 0.05 level of significance across all evaluation measures.

4.5.2.3 Clustering Performance Comparison with Category Information

We further conduct the experiments by incorporating the category information for clustering. The category information is used in the same way of surrounding text. In view that the category information for each image is exactly one word, it therefore can also be used as the noiseless tag of the image. Generally speaking, the category information cannot be obtained for all the images under the clustering setting. We use it as an additional tag feature to evaluate all the methods in an ideal case, and show that Fusion ART, GHF-ART and GHF-ART(SS) achieve perfect results in terms of weighted average precision, cluster entropy and purity, while the other algorithms cannot obtain such excellent results (see Table 4.6). It is because the ART-based algorithms not only evaluate the overall similarity across all of the feature channels, but also have constraints for each of them. Therefore, with category labels, the ART-based algorithms can effectively identify the classes of images. Besides, we can also observe an improvement of GHF-ART(SS) over Fusion ART and GHF-ART in class entropy and rand index, which also consider how the patterns with the same label are grouped together.

Comparing the results with those in Table 4.5, we can find that Fusion ART, GHF-ART and GHF-ART(SS) also obtain a big improvement in terms of class entropy and rand index, while the other algorithms have a relatively small improvement. The reason should be that the global optimization considers the overall similarity across all the feature channels so that the noisy features still contribute to incorrect categorization. It demonstrates the importance of taking in the fitness of patterns in terms of the overall similarity and also similarity in individual modality.

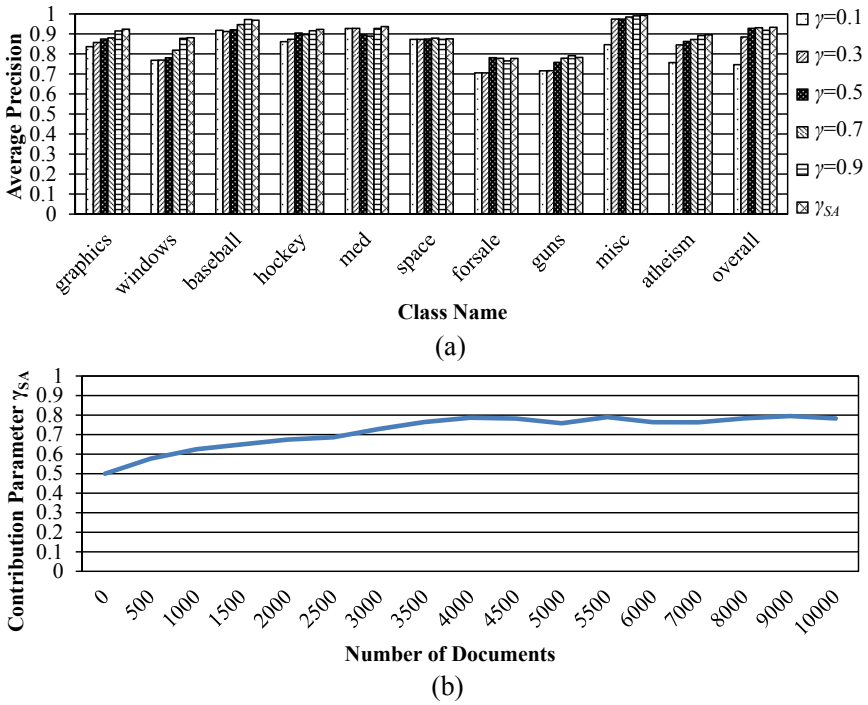


Figure 4.6: (a) Clustering performance using fixed contribution parameters (γ) and self-adapted contribution parameter (γ_{SA}); (b) Tracking of γ_{SA} on 20 newsgroups data set.

4.5.3 20 Newsgroups Data Set

The 20 Newsgroups data set [153] is a popular public data set which comprises nearly 20,000 newsgroup documents across 20 different newsgroups and is widely used for the experiments of text clustering techniques. We directly collect ten classes from the processed matlab version of the 20news-bydate data set[1], and each of them contains nearly 1,000 documents. For the ease of discussion, we refer the ten categories by the abbreviations as follows: comp.graphics (graphics), comp.windows.x (windows), rec.sport.baseball (baseball), rec.sport.hockey (hockey), sci.med (med), sci.space (space), misc.forsale (forsale), talk.politics.guns (guns), talk.politics.misc (misc) and alt.atheism (atheism). We use the traditional text mining algorithm tf-idf to extract the features of documents and use the words in the category information to construct the category features.

[1] `http://qwone.com/~jason/20Newsgroups/`

Table 4.7: Clustering results on 20 Newsgroups data set using document content and category information.

	K-means	CIHC	SRC	Comrafs	NMF	SS-NMF	Fusion ART	GHF-ART	GHF-ART(SS)
AP	0.6386 ± 0.027	0.7583	0.7246	0.6547	0.7357	0.7869	0.7566 ± 0.021	0.8071 ± 0.023	$\mathbf{0.8452 \pm 0.018}$
$H_{cluster}$	0.4833 ± 0.025	0.4246	0.4432	0.4679	0.4267	0.3938	0.4016 ± 0.016	0.3822 ± 0.018	$\mathbf{0.3642 \pm 0.018}$
H_{class}	0.5284 ± 0.031	0.4573	0.4630	0.5162	0.4487	0.4296	0.4469 ± 0.015	0.4131 ± 0.017	$\mathbf{0.3824 \pm 0.019}$
Purity	0.6826 ± 0.027	0.7711	0.7348	0.6950	0.7503	0.7836	0.7538 ± 0.021	0.7994 ± 0.018	$\mathbf{0.8435 \pm 0.021}$
RI	0.6670 ± 0.025	0.7284	0.6867	0.6136	0.7019	0.7458	0.7268 ± 0.017	0.7759 ± 0.022	$\mathbf{0.8013 \pm 0.019}$

4.5.3.1 Performance of Robustness Measure

Fig. 4.6 shows the clustering results with different settings of contribution parameter of the category features. In Fig. 4.6(a), we observe that the robustness measure works well for all classes and usually produces the best results. From Fig. 4.6(b), we can observe that the contribution parameter of category features gradually increases from 0.5 to over 0.6 after 1,500 input patterns. Despite the small fluctuation, the value stabilizes at around 0.8, which indicates that the category information is more robust during the clustering process.

4.5.3.2 Clustering Performance Comparison

Similar to the NUS-WIDE data set, we evaluate the clustering performance of GHF-ART using weighted average precision, cluster and class entropies, purity and rand index. Since the number of classes in 20 Newsgroups data set is 10, we set the number of clusters ranging from 10 to 15. In Table 4.7, we can see that GHF-ART and GHF-ART(SS) outperform the other algorithms in all the performance measures. Moreover, both of them achieve higher than 80 percent in weighted average precision and purity while the other algorithms typically obtain less than 75% except CIHC and SS-NMF. Similarly, a gain of more than 3% over the best performance by the other algorithms is achieved in rand index. The t-test results further show that the performance of Fusion ART, GHF-ART and GHF-ART(SS) are significantly different at 0.05 level of significance in all evaluation measures. In fact, we observe that GHF-ART has a big improvement over Fusion ART. This demonstrates that the proposed feature weighting algorithm and meta-information learning method can help to improve the performance of Fusion ART in the heterogeneous data co-clustering task.

Chapter 5

Community Discovery in Social Networks via Heterogeneous Link Association and Fusion

Discovering social communities of web users through clustering analysis of heterogeneous link associations has drawn much attention. However, existing approaches typically require the number of clusters a prior, do not address the weighting problem for fusing heterogeneous types of links, and have a heavy computational cost. In this chapter, we study the commonly used social links of users and explore the feasibility of the proposed heterogeneous data clustering algorithm GHF-ART, proposed in Chapter 4, for discovering user communities in social networks. Different from existing algorithms proposed for this task, GHF-ART performs real-time matching of patterns and one-pass learning, which guarantee its low computational cost. With a vigilance parameter to restrain the intra-cluster similarity, GHF-ART does not need the number of clusters a prior. To achieve a better fusion of multiple types of links, GHF-ART employs a weighting function to incrementally assess the importance of all the feature channels. Extensive experiments have been conducted to analyze the performance of GHF-ART on two heterogeneous social network data sets. The promising results comparing with existing methods demonstrate the effectiveness and efficiency of GHF-ART.

5.1 Introduction

Clustering [7] for discovering communities of users in social networks [109] has been an important task for the understanding of collective social behavior [134] and associative mining such as social link prediction and recommendation [116, 117]. However, with the popularity of social websites such as Facebook, users may communicate and interact with each other easily and diversely, such as posting blogs and tagging documents. The availability of those social media data, on one hand, enables the extraction of rich link information among users for further analysis. On the other hand, new challenges have arised for traditional clustering techniques to perform community discovery of social users from heterogeneous social networks in which the users are associated by multiple but different types of social links, such as the scalability to large social networks, techniques for link representation, and methods for fusing heterogeneous types of links.

In recent years, many works have been done on the clustering of heterogeneous data. Existing methods may be considered in four categories: multi-view clustering approach [8, 91–93], spectral clustering approach [9, 30, 85, 158], matrix factorization approach [10, 87] and aggregation approach [11, 94]. However, they have several limitations for clustering heterogeneous social network data in practice. Firstly, existing algorithms typically involve iterative optimization which does not scale well to big data sets. Secondly, most of them need the number of clusters a prior, which is hard to decide in practice. Thirdly, most of those algorithms do not consider the weighting problem when fusing multiple types of links. Since different types of links have their own meanings and levels of feature values, equal or empirical weights for them may bias their importance in similarity measure and may not yield satisfactory performance.

In this study, we explore the feasibility of Generalized Heterogeneous Fusion Adaptive Resonance Theory (GHF-ART) for identifying user groups in heterogeneous social networks. GHF-ART [75], extended from Fusion ART [20], has been proposed for clustering web multimedia data through the fusion of an arbitrary rich level of heterogeneous data resources such as images, articles, and surrounding text. For clustering data patterns of social networks, we develop a set of specific feature representation and learning rules for GHF-ART to handle various heterogeneous types of social links, including relational links, textual links in articles, and textual links in short text.

GHF-ART has several key properties different from existing approaches. Firstly, GHF-ART performs online and one-pass learning so that the clustering process can be done in just a single round of pattern presentation. Secondly, GHF-ART does not need the number of clusters a prior. Thirdly, GHF-ART employs a weighting function, termed *Robustness Measure* (RM), which adaptively tunes the weights for different feature channels according to their importance in pattern representation, in order to achieve a satisfactory level of the overall similarity across all the feature channels. Besides, GHF-ART not only globally considers the overall similarity across all the feature channels, but also locally evaluates the similarity obtained from each channel. This helps to handle cases when users share some common interests but behave differently in some other aspects.

We analyze the performance of GHF-ART on two public social network data sets, namely the YouTube data set [94] and the BlogCatalog data set [159], through the parameter sensitivity analysis, the clustering performance comparison, the effectiveness evaluation of *Robustness Measure* and the time cost comparison. The experimental results show that GHF-ART outperforms and is much faster than many existing heterogeneous data clustering algorithms.

5.2 Problem Statement

The community discovery problem in heterogeneous social networks is to identify a set of social user groups by evaluating different types of links between users, such that members in the same group interact with each other more frequently and share more common interests than those outside the group.

Considering a set of users $\mathcal{U} = \{u_1, \ldots, u_N\}$ and their associated multiple types of links $\mathcal{L} = \{l_1, \ldots, l_K\}$, such as contact links and subscription links, each user u_n therefore can be represented by a multi-channel input pattern $\mathcal{I} = \{\mathbf{x}^1, \ldots, \mathbf{x}^K\}$, where $\mathbf{x}^k$ is a feature vector extracted from the kth link.

Consequently, the community discovery task is to identify a set of clusters $\mathcal{C} = \{c_1, \ldots, c_J\}$ according to the similarities among the user patterns evaluated within and across different types of links. As a result, given a user $u_N \in c_J$ and two users $u_p \in c_J$ and $u_q \notin c_J$, for $\forall p, q$ such that $u_p, u_q \in \mathcal{U}$, we have $S_{u_N,u_p} > S_{u_N,u_q}$, where S_{u_N,u_p} denotes

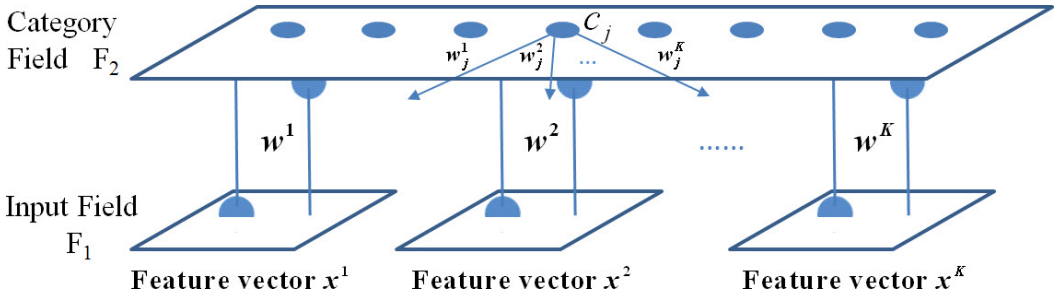


Figure 5.1: The architecture of GHF-ART for integrating K types of feature vectors.

the overall similarity between u_N and u_p. Namely, users in a cluster may consistently have a higher degree of similarity in terms of all types of links than those belonging to other clusters.

5.3 GHF-ART for Clustering Heterogeneous Social Links

GHF-ART [75] is designed for clustering composite patterns which are represented by multiple types of features. As shown in Fig. 5.1, GHF-ART consists of K independent feature channels in the input field for handling an arbitrarily rich level of heterogeneous links, and a category field consisting of clusters. GHF-ART processes input patterns one at a time, during which each of them is either identified as a novel template/prototype which incurs the generation of a new cluster or categorized into an existing cluster of similar patterns. In this way, the category space of GHF-ART is incrementally partitioned into regions of clusters.

The clustering process of GHF-ART comprises four key steps: 1) **Category choice**: select a best-matching cluster, called a winner, across all the feature channels; 2) **Template matching**: Evaluate if the degree of similarity between the input pattern $\mathcal{I}$ and the winner satisfies a threshold, called the vigilance criteria, for each feature channel; 3) **Resonance and Reset**: If the vigilance criteria is violated, a reset occurs so that a new winner is selected from the rest of the clusters in the category field; Otherwise, a resonance occurs which leads to the learning of winner from the input pattern for all the feature channels. 4) **Network Expansion**: If no cluster meets the vigilance criteria, a new cluster is generated to encode the new pattern.

In the following subsections, we illustrate the key steps of GHF-ART for clustering social network data, in terms of the representation of commonly used social links, the heterogeneous link fusion for pattern similarity measure, the learning strategies for cluster template generalization, and the weighting algorithm for heterogeneous links. The complete algorithm of GHF-ART is shown at the end of this section.

5.3.1 Heterogeneous Link Representation

In GHF-ART, each social user with multi-modal links is represented by a multi-channel input pattern $\mathcal{I} = \{\mathbf{x}^k|_{k=1}^K\}$, where $\mathbf{x}^k$ is the feature vector for the kth feature channel. When presenting to GHF-ART, the input patterns undergo two normalization procedures. Firstly, *min-max normalization* is employed to guarantee that the input values are in the interval of [0, 1]. Secondly, *complement coding* [68] normalizes the input feature vector by concatenating $\mathbf{x}^k$ with its complement vector $\bar{\mathbf{x}}^k$ such that $\bar{\mathbf{x}}^k = 1 - \mathbf{x}^k$.

To fit GHF-ART with the social network data, we categorize commonly used social links into three categories and develop the respective representation methods accordingly, as discussed below.

5.3.1.1 Density-Based Features for Relational Links

Relational links, such as contact and co-subscription links, use the number of interactions as the strength of connection between users. Considering a set of users $\mathcal{U} = \{u_1, \ldots, u_N\}$, the density-based feature vector of the nth user u_n is represented by $[f_{n,1}, \ldots, f_{n,N}]$, wherein $f_{n,i}$ reflects the density of interactions between the user u_n and the ith user u_N.

5.3.1.2 Text-Similarity Features for Articles

Text-similarity features are used to represent the articles of users with long paragraphs such as blogs. Considering a set of users $\mathcal{U} = \{u_1, \ldots, u_N\}$ and the word list $\mathcal{G} = \{g_1, \ldots, g_M\}$ of all of the M distinct keywords from their articles, the text-similarity feature vector of the nth user u_n is represented by $[f_{n,1}, \ldots, f_{n,M}]$, where $f_{n,i}$ indicates the importance of keyword g_i to represent the user u_n, which can be computed by term frequency-inverse document frequency (tf-idf).

5.3.1.3 Tag-Similarity Features for Short Text

Tag-similarity features are used to represent short text, such as tags and comments. The key difference of short text from article is that short text consists of few but meaningful words. Given a set of user $\mathcal{U} = \{u_1, \ldots, u_N\}$ and the corresponding word list $\mathcal{G} = \{g_1, \ldots, g_H\}$ of all the H distinct words, the tag-similarity feature vector of the nth user u_n is expressed by $[f_{n,1}, \ldots, f_{n,H}]$. Following the representation method for meta-information in [75], given that $\mathcal{G}_n$ is the word list of u_n, $f_{n,i}$ $(i = 1, ..., H)$ is given by

$$f_{n,i} = \begin{cases} 1, & if\ g_i \in \mathcal{G}_n \\ 0, & otherwise \end{cases}. \tag{5.1}$$

5.3.2 Heterogeneous Link Fusion for Pattern Similarity Measure

GHF-ART performs the selection of best-matching cluster to the input pattern and evaluates the fitness between them through a two-way similarity measure: a bottom-up measure to select a winner cluster by globally considering the overall similarity across all the feature channels; and a top-down measure to locally evaluate if the similarity for each feature channel meets the vigilance criteria.

5.3.2.1 Bottom-Up Similarity Measure for Category Choice

In the first step, a choice function is employed to evaluate the overall similarity between the input pattern and the template weight of each cluster in the category field, which is defined by

$$T(c_j, \mathcal{I}) = \sum_{k=1}^{K} \gamma^k \frac{|\mathbf{x}^k \wedge \mathbf{w}_j^k|}{\alpha + |\mathbf{w}_j^k|}, \tag{5.2}$$

where $\mathbf{w}_j^k$ denotes the weight vector for the kth feature channel of the jth cluster, contribution parameter $\gamma^k \in [0, 1]$ is the weight for the kth feature channel, choice parameter $\alpha \approx 0$ is a positive real value to balance the denominator, the operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and $|.|$ is given by the ℓ_1 norm. The choice function evaluates the proportion of intersection between the feature vectors of the input pattern and the prototypes of the winner across all the feature channels, so that the winner cluster with the best matching feature distribution in the category field is identified.

5.3.2.2 Top-Down Similarity Measure for Template Matching

After identifying the winner cluster c_{j^*}, a match function is used to evaluate if the selected winner matches the input pattern in each feature channel. For the kth feature channel, the match function is defined by

$$M(c_{j^*}, \mathbf{x}^k) = \frac{|\mathbf{x}^k \wedge \mathbf{w}_{j^*}^k|}{|\mathbf{x}^k|}. \tag{5.3}$$

If the match function value for each of the K feature channels satisfies the respective vigilance criterion defined by $M(c_{j^*}, \mathbf{x}^k) > \rho$ for $k = 1, ...K$, where $\rho \in [0, 1]$ is the vigilance parameter, a resonance occurs so that the input pattern is categorized into the winner cluster. Otherwise, a reset occurs to select a new winner from the rest of the clusters in the category field.

5.3.3 Learning from Heterogeneous Links

5.3.3.1 Learning from Density-Based and Text-Similarity Features

The density-based features and textual features for articles use a distribution to represent the characteristics of a user. Therefore, GHF-ART should be able to learn the generalized distribution of similar patterns in the same cluster so that the users with similar feature distribution can be identified.

To this end, we use the learning function of Fuzzy ART [68]. Assuming the kth feature channel is for density-based features, the corresponding learning function of the winner cluster c_{j^*} is therefore defined by

$$\hat{\mathbf{w}}_{j^*}^k = \beta(\mathbf{x}^k \wedge \mathbf{w}_{j^*}^k) + (1 - \beta)\mathbf{w}_{j^*}^k, \tag{5.4}$$

where $\beta \in [0, 1]$ is the learning parameter. We observe that the updated weight values will not be larger than the old ones, so that this learning function may incrementally identify the key features by preserving the key features which have stably high values while depressing the features which are unstable in values.

5.3.3.2 Learning from Tag-Similarity Features

We use the learning function for meta-information in GHF-ART [75] to model the cluster prototypes for tag-similarity features. Assuming the kth feature channel is for tag-similarity features of short text, given the kth feature vector $\mathbf{x}^k = [x_1^k, \ldots, x_H^k]$ of the input pattern $\mathcal{I}$, the winner cluster c_{j^*} with L users and the corresponding weight vector $\mathbf{w}_{j^*}^k = [w_{j^*,1}^k, \ldots, w_{j^*,H}^k]$ of c_{j^*} for the kth feature channel, the learning function for $w_{j^*,h}^k$ is defined by

$$\hat{w}_{j^*,h}^k = \begin{cases} \eta w_{j^*,h}^k & if\ x_h^k = 0 \\ \eta(w_{j^*,h}^k + \frac{1}{L}) & otherwise \end{cases}, \tag{5.5}$$

where $\eta = \frac{L}{L+1}$. Equation 5.5 models the cluster prototype for the tag-similarity features by the probabilistic distribution of tag occurrences. Thus, the similarity between tag-similarity features can be considered as the number of common words. During each round of learning, the keywords with high frequency to occur in the cluster are given high weights while those of the noisy words are incrementally decreased.

5.3.4 Adaptive Weighting of Heterogeneous Links

GHF-ART employs the *Robustness Measure* (RM) to adaptively tune γ for different feature channels, which evaluates the importance of different feature channels by considering the intra-cluster scatters.

Considering a cluster c_j with L users, each of which is denoted by $\mathcal{I}_l = \{\mathbf{x}_l^1, \ldots, \mathbf{x}_l^K\}$ for $l = 1, \ldots, L$, and the corresponding weight vectors for the K feature channels denoted by $\mathcal{W}_j = \{\mathbf{w}_j^1, \ldots, \mathbf{w}_j^K\}$, the *Difference* for the kth feature channel of c_j is defined by

$$D_j^k = \frac{\frac{1}{L}\sum_l |\mathbf{w}_j^k - \mathbf{x}_l^k|}{|\mathbf{w}_j^k|}. \tag{5.6}$$

Considering all the clusters, the *Robustness* of the kth feature channel can be measured by

$$R^k = \exp(-\frac{1}{J}\sum\nolimits_j D_j^k). \tag{5.7}$$

As the weights for the respective feature channels, the contribution parameter for the kth feature channel γ^k is defined by

$$\gamma^k = \frac{R^k}{\sum_{k=1}^K R^k}. \tag{5.8}$$

Algorithm 5.1 GHF-ART

Input: Input patterns $\mathcal{I}_n = \{\mathbf{x}^k|_{k=1}^K\}$, α, β and ρ.

1: Present $\mathcal{I}_1 = \{\mathbf{x}^k|_{k=1}^K\}$ to the input field.
2: Set $J = 1$. Create a node c_J such that $\mathbf{w}_J^k = \mathbf{x}^k$ for $k = 1, \ldots, K$.
3: set $n = 2$.
4: **repeat**
5: Present $\mathcal{I}_n$ to the input field.
6: For $\forall c_j$ $(j = 1, ..., J)$, calculate the choice function $T(c_j, \mathcal{I}_n)$ according to Equation 5.2.
7: Identify the winner cluster c_{j^*} so that $j^* = \arg\max_{j:c_j \in F_2} T(c_j, \mathcal{I}_n)$. If $j^* = 0$, go to 11.
8: Calculate the match function $M(c_{j^*}, \mathbf{x}^k)$ for $k = 1, \ldots, K$ according to Equation 5.3.
9: If $\exists k$ such that $M(c_{j^*}, \mathbf{x}^k) < \rho^k$, set $T(c_{j^*}, \mathcal{I}_n) = 0$, $j^* = 0$, go to 7.
10: If $j^* \neq 0$, update $\mathbf{w}_{j^*}^k$ for $k = 1, \ldots, K$ according to Equation 5.4 and Equation 5.5 respectively and update γ according to Equation 5.7 - Equation 5.10.
11: If $j^* = 0$, set $J = J+1$, create a new node c_J such that $\mathbf{w}_{J+1}^k = \mathbf{x}^k$ for $k = 1, \ldots, K$, update γ according to Equation 5.11.
12: $n = n + 1$.
13: **until** All the input patterns are presented.

Output: Cluster Assignment Array $\{A_n|_{n=1}^N\}$.

The respective incremental update equations for the contribution parameters are further derived for the following two cases:

- **Resonance in existing cluster**: Assume that the input pattern $\mathcal{I}_{L+1} = \{\mathbf{x}_{L+1}^1, \ldots, \mathbf{x}_{L+1}^K\}$ is assigned to an existing cluster c_j. For the kth feature channel, the corresponding update equations for the density-based and text-similarity features and tag-similarity features are defined by Equation 5.9 and Equation 5.10 respectively:

$$\hat{D}_j^k = \frac{\eta}{|\hat{\mathbf{w}}_j^k|}(|\mathbf{w}_j^k|D_j^k + |\mathbf{w}_j^k - \hat{\mathbf{w}}_j^k| + \frac{1}{L}|\hat{\mathbf{w}}_j^k - \mathbf{x}_{L+1}^k|) \tag{5.9}$$

$$\hat{D}_j^k = \frac{\eta}{|\hat{\mathbf{w}}_j^k|}(|\mathbf{w}_j^k|D_j^k - |\hat{\mathbf{w}}_j^k - \eta\mathbf{w}_j^k| + \frac{1}{L}|\hat{\mathbf{w}}_j^k - \mathbf{x}_{L+1}^k|). \tag{5.10}$$

After the update for all feature channels, the updated contribution parameter can then be obtained by calculating Equation 5.7 - Equation 5.8.

- **Generation of new cluster**: When generating a new cluster, the differences of other clusters remain unchanged. Therefore, it just introduces a proportionally change of *Difference*. Considering the *robustness* R^k $(k = 1, ..., K)$ for all of the feature channels, the update contribution parameter for the kth feature channel is derived as:

$$\hat{\gamma}^k = \frac{(R^k)^{\frac{J}{J+1}}}{\sum_{k=1}^{K}(R^k)^{\frac{J}{J+1}}}. \tag{5.11}$$

5.3.5 Time Complexity Comparison

The time complexity of GHF-ART with *Robustness Measure* has been demonstrated to be $O(n_i n_c n_f)$ in [75], where n_i is the number of input patterns, n_c is the number of clusters, and n_f is the total number of features.

In comparison with existing heterogeneous data clustering algorithms, the time complexity of LMF [87] is $O(tn_i n_c(n_c + n_f))$, PMM [94] is $O(n_i^3 + tn_c n_i n_f))$, SRC [85] is $O(tn_i^3 + n_c n_i n_f))$ and NMF [10] is $O(tn_c n_i n_f)$, where t is the number of iteration. We observe that GHF-ART has a much lower time complexity.

5.4 Experiments

5.4.1 YouTube Data Set

5.4.1.1 Data Description

The YouTube data set [1] is a heterogeneous social network data set, which is originally used to study the community detection problem via heterogeneous interactions of users. This data set contains $15,088$ users from YouTube website and involves five types of relational links, including contact network, co-contact network, co-subscription network, co-subscribed network and favorite network.

5.4.1.2 Evaluation Measure

Since there is no ground truth labels of users in this data set, we adopt the following five evaluation measures: 1) *Cross-Dimension Network Validation* (*CDNV*) [94], which

[1] http://socialcomputing.asu.edu/datasets/YouTube

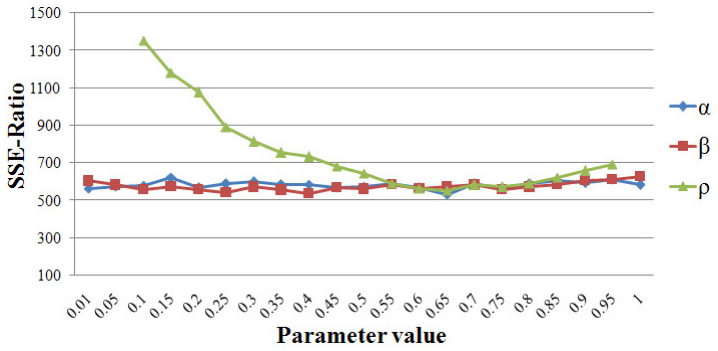


Figure 5.2: The clustering performance of GHF-ART on the YouTube data set in terms of *SSE-Ratio* by varying the values of α, β and ρ respectively.

evaluates how well the cluster structure learnt from one or more types of links fits the network of the other type of links. A larger value indicates a better performance; 2) *Average Density* (*AD*) measures the average probability of two users in the same cluster having connection, defined by $AD = \frac{1}{J}\frac{1}{K}\Sigma_j\Sigma_k\frac{2e_j^k}{n_j(n_j-1)}$, where e_j^k is the number of edges of the k-th link in cluster c_j and n_j is the number of patterns in c_j; 3) *Intra-cluster sum-of-squared error* (*Intra-SSE*) measures the weighted average of *SSE* within clusters across feature modalities, defined by $\textit{Intra-SSE} = \Sigma_j\Sigma_{\mathbf{x}_i^k\in c_j}\Sigma_k\frac{n_j}{\Sigma_j n_j}(\mathbf{x}_i^k - \bar{\mathbf{x}}_j^k)^2$, where $\mathbf{x}_i^k$ is the feature vector of the i-th pattern for the k-th link and $\bar{\mathbf{x}}_j^k$ is the mean value of all the $\mathbf{x}_i^k \in c_j$; 4) *Between-cluster SSE* (*Between-SSE*) measures the average distance between two cluster centers to evaluate how well-separated the clusters are from each other, defined by $\textit{Between-SSE} = \Sigma_j\Sigma_i\Sigma_k\frac{1}{J(J-1)}(\bar{\mathbf{x}}_j^k - \bar{\mathbf{x}}_i^k)^2$; and 5) $\textit{SSE-Ratio} = \textit{Intra-SSE}/\textit{Between-SSE}$ gives an overall performance.

5.4.1.3 Parameter Selection Analysis

We initialized $\alpha = 0.01$, $\beta = 0.6$ and $\rho = 0.6$ and studied the change in performance of GHF-ART in terms of *SSE-Ratio* by varying one of them while fixing others, as shown in Fig. 5.2. We observe that despite some small fluctuations, the performance of GHF-ART is roughly robust to the change in the values of α and β. Regarding the vigilance parameter ρ, we find the performance is improved when ρ increases up to 0.65 and degrades when $\rho > 0.85$. We further analyzed the cluster structures generated under different values ρ, as shown in Fig. 5.3. We observe that the increase of ρ leads to the generation of more clusters, which may contribute to the compactness of clusters.

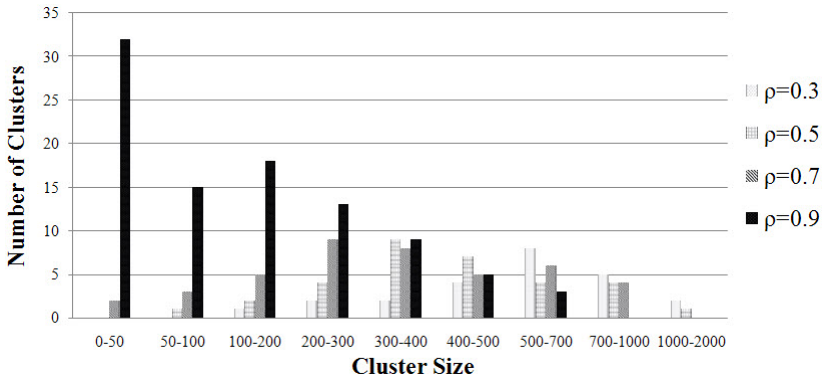


Figure 5.3: The cluster structures generated by GHF-ART on the Youtube data set in terms of different values of vigilance parameter ρ.

At $\rho = 0.9$, a significant number of small clusters are generated, which degrades the performance in terms of recall.

To study the selection of ρ, we analyzed the cluster structure at $\rho = 0.5$ and 0.7 at which the best performance is obtained. We observe that when ρ increases from 0.5 to 0.7, the number of small clusters, which contain less than 100 patterns, increases. Therefore, we assume that when a suitable ρ is reached, the number of small clusters starts to increase. If this idea works, an interesting empirical way to select a reasonable value of ρ is to tune the value of ρ until a small number of small clusters, less than 10% of the total number of clusters, are identified.

5.4.1.4 Clustering Performance Comparison

We compared the performance of GHF-ART with four existing heterogeneous data clustering algorithms, namely the Spectral Relational Clustering (SRC) [85], Linked Matrix Factorization (LMF) [87], Non-negative Matrix Factorization (NMF) [10] and Principal Modularity Maximization (PMM) [94]. Since SRC and PMM need K-means to obtain the final clusters, we also employed K-means with Euclidean distance as a baseline.

To make a fair comparison, since GHF-ART needs to perform min-max normalization, we applied the normalized data as input to the other algorithms. For GHF-ART, we fixed $\alpha = 0.01$ and $\beta = 0.6$. For K-means, we concatenated the feature vectors of the five types of links. For SRC, we use the same weight values as GHF-ART. The number of iteration for K-means, SRC, LMF, NMF and PMM was set to 50.

Table 5.1: The clustering performance of GHF-ART, K-means, SRC, LMF, NMF and PMM under the best setting of pre-defined number of clusters ("k") ($\rho = 0.6$ and 0.65 when $k = 35$ and 37 respectively for GHF-ART) in terms of *CDNV*, *Average Density* (*AD*), *Intra-SSE*, *Between-SSE* and *SSE-Ratio* on the YouTube data set.

	CDNV		*AD*		*Intra-SSE*		*Between-SSE*		*SSE-Ratio*	
	value	k	value	k	value	k	value	k	value	k
K-means	0.2446	43	0.0572	40	7372.4	41	9.366	40	774.14	41
SRC	0.2613	37	0.0691	35	6593.6	36	10.249	35	652.34	36
LMF	0.2467	39	0.0584	38	6821.3	41	9.874	37	694.72	40
NMF	0.2741	36	0.0766	35	6249.5	36	**10.746**	34	591.57	35
PMM	0.2536	36	0.0628	37	6625.8	37	9.627	34	702.25	35
GHF-ART	**0.2852**	37	**0.0834**	37	**5788.6**	37	10.579	35	**563.18**	37

We obtained the clustering results of GHF-ART with different values of ρ ranging from 0.3 to 0.9 and those of K-means, SRC, LMF, NMF and PMM with different pre-defined numbers of clusters ranging from 20 to 100. The best performance of each algorithm for each evaluation measure is reported in Table 5.1. We observe that the best performance of each algorithm is typically achieved with $34 - 41$ clusters. GHF-ART usually achieves the best performance with $\rho = 0.65$ which is more consistent than other algorithms. GHF-ART outperforms other algorithms in terms of all the evaluation measures except *between-SSE*, but the result of GHF-ART is still competitive to the best one.

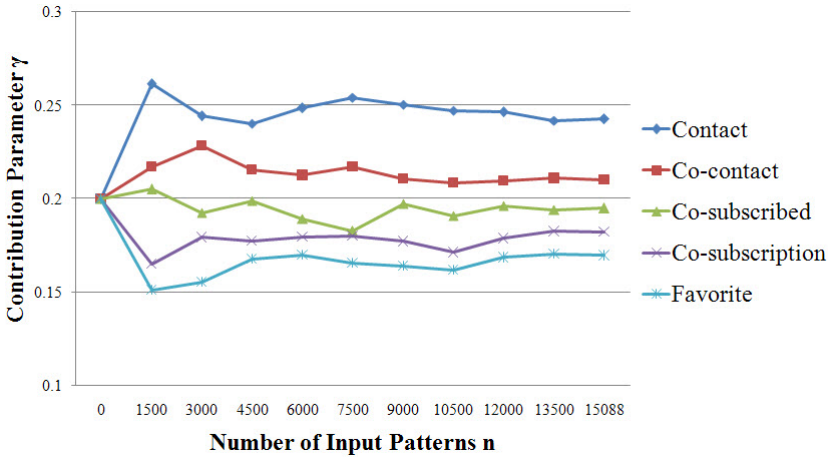


Figure 5.4: Trace of contribution parameters for five types of links during clustering with the increase in the number of input patterns.

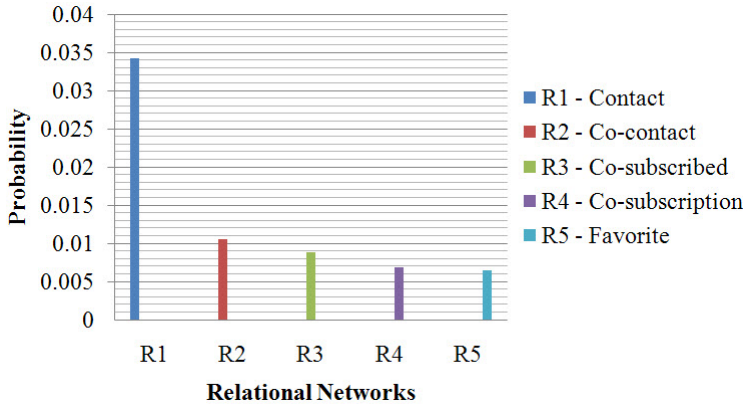


Figure 5.5: The probability that pairs of patterns falling into the same cluster are connected in each of the five relational networks.

5.4.1.5 Correlation Analysis of Heterogeneous Networks

We first ran GHF-ART under $\alpha = 0.01$, $\beta = 0.6$ and $\rho = 0.65$ and showed the trace of contribution parameters for each type of links during clustering in Fig. 5.4. We observe that the weights for all types of features begin with 0.2. The initial fluctuation at $n = 1500$ is due to the incremental generation of new clusters. After $n = 12000$, the weight values of all types of features become stable.

We further analyzed the probability of pairs of connected patterns falling into the same cluster to study how each type of relational networks affects the clustering results, as shown in Fig. 5.5. We observe that the order of relational networks is consistent with the results shown in Fig. 5.4. This demonstrates the validity of *Robustness Measure*. Among all types of links, the contact network achieves a much higher probability than other relational networks. This may be due to the fact that the contact network is much sparser than the other four networks. As such, we may expect that the links of contact network are more representative.

5.4.2 BlogCatalog Data Set

5.4.2.1 Data Description

The BlogCatalog data set[2] is crawled in [159] and used for discovering the overlapping social groups of users. It consists of the raw data of 88, 784 users, each of which involves

[2]http://dmml.asu.edu/users/xufei/datasets.html#Blogcatalog

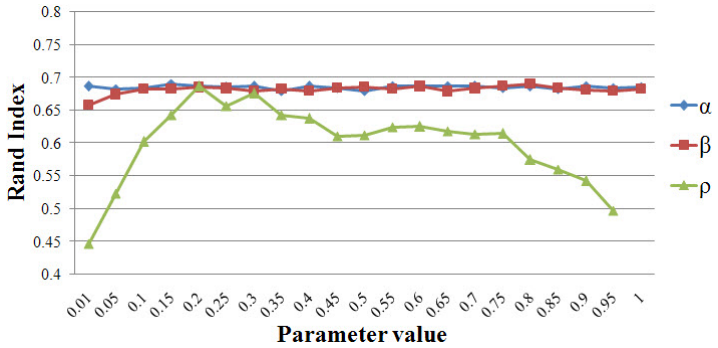


Figure 5.6: The clustering performance of GHF-ART on the BlogCatalog data set in terms of Rand Index by varying the values of α, β and ρ respectively.

the friendship to other users and the published blogs. Each blog of a user is described by several pre-defined categories, user-generated tags and six snippets of blog content.

We extracted three types of links, including a friendship network and two textual similarity networks in terms of blog content and tags. By filtering infrequent words from tags and blogs, we obtained 66,418 users, 6,666 tags and 17,824 words from blogs. As suggested in [159], we used the most frequent category in the blogs of a user as the class label and obtained 147 class labels.

5.4.2.2 Evaluation Measure

With the ground truth labels, we used *Average Precision* (*AP*), *Cluster Entropy* and *Class Entropy* [150], *Purity* [156] and *Rand Index* [157] as the clustering evaluation measures. *Average Precision*, *Cluster Entropy* and *Purity* evaluate the intra-cluster compactness. *Class Entropy* evaluates how well the classes are represented by the minimum number of clusters. *Rand Index* considers both cases.

5.4.2.3 Parameter Selection Analysis

We studied the influence of parameters to the performance of GHF-ART on the BlogCatalog data set with the initial setting of $\alpha = 0.01$, $\beta = 0.6$ and $\rho = 0.2$, as shown in Fig. 5.6. We observe that, consistent with those in Fig. 5.2, the performance of GHF-ART is robust to the change in the choice and learning parameters. As expected, the performance of GHF-ART varies a lot due to the change in ρ. This curve may also be explained by the same reason for that in Fig. 5.2.

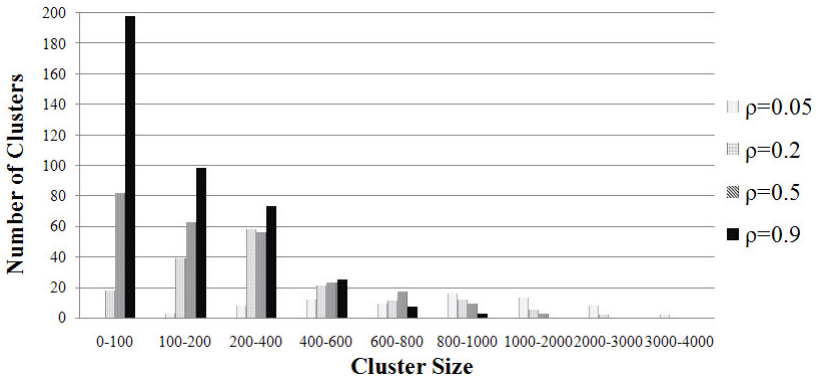


Figure 5.7: The cluster structures generated by GHF-ART on the BlogCatalog data set in terms of different values of vigilance parameter ρ.

To validate our findings to select a suitable ρ in Section 5.4.1.3, we analyzed the cluster structures corresponding to the four key points of ρ, as shown in Fig. 5.7. We observe that, at $\rho = 0.2$, nearly 20 small clusters with less than 100 patterns are generated. Interestingly, we find that the number of small clusters is also around 10% of the total number of clusters, which fits the findings that we observe on the YouTube data set. This demonstrates the feasibility of the proposed empirical way to select a suitable value of ρ.

5.4.2.4 Clustering Performance Comparison

We compared the performance of GHF-ART with the same set of algorithms compared in the YouTube data set under the same parameter settings as mentioned in Section 5.4.1.4, except the number of clusters. We varied the value of ρ from 0.1 to 0.4 with an interval of 0.05 and the number of clusters from 150-200 with an interval of 5. The best performance for each algorithm with the number of clusters is shown in Table 5.2. We observe that GHF-ART obtained much better performance (at least 4% improvement) than the other algorithms in terms of *Average Precision*, *Cluster Entropy* and *Purity*. This indicates that GHF-ART may well identify similar patterns and produce more compact clusters. Competitive performance is obtained by SRC and NMF in terms of *Class Entropy*. Considering the number of clusters under the best settings, we find that GHF-ART identifies a similar number of clusters to other algorithms, which demonstrates the effectiveness of GHF-ART.

Table 5.2: The clustering performance of GHF-ART, K-means, SRC, LMF, NMF and PMM under the best setting of pre-defined number of clusters ("k") ($\rho = 0.15$, 0.2 and 0.25 when $k = 158$, 166 and 174 respectively for GHF-ART) on the BlogCatalog data set in terms of *Average Precision* (AP), *Cluster Entropy* ($H_{cluster}$), *Class Entropy* (H_{class}), *Purity* and *Rand Index*(RI).

	AP		$H_{cluster}$		H_{class}		$Purity$		RI	
	value	k	value	k	value	k	value	k	value	k
K-means	0.6492	185	0.5892	185	0.5815	165	0.6582	185	0.5662	170
SRC	0.7062	175	0.5163	175	0.4974	160	0.7167	175	0.6481	170
LMF	0.6626	175	0.5492	175	0.5517	155	0.6682	175	0.6038	165
NMF	0.7429	175	0.4836	175	0.4883	155	0.7791	175	0.6759	165
PMM	0.6951	170	0.5247	170	0.5169	165	0.6974	170	0.6103	165
GHF-ART	**0.7884**	174	**0.4695**	174	**0.4865**	158	**0.8136**	174	**0.6867**	166

Table 5.3: The five biggest clusters identified by GHF-ART with class labels, top tags, cluster size and *Precision*.

Cluster Rank	Class Label	Top Tags	Cluster Size	*Precision*
1	Personal	music, life, art, movies, Culture	2692	0.7442
2	Blogging	news, blog, blogging, SEO, Marketing	2064	0.8166
3	Health	health, food, beauty, weight, diet	1428	0.7693
4	Personal	life, love, travel, family, friends	1253	0.6871
5	Entertainment	music, movies, news, celebrity, funny	1165	0.6528

5.4.2.5 Case Study

We further studied the identified communities by GHF-ART. First, we listed the five biggest clusters discovered, as shown in Table 5.3. We observe that those clusters are well formed to reveal the user communities since more than 1000 patterns are grouped with a reasonable level of precision. We also observe that most of the top tags discovered by the cluster weight values are semantically related to their corresponding classes. Interestingly, the clusters ranked 1 and 4 belong to the class "Personal". This may be because, according to our organized statistics, "Personal" is much larger than other classes. However, in the top 5 tags, only "life" is shared by them. To have an insight of the relation between these two clusters, we plot the tag clouds for them. As shown in Fig. 5.8, we observe that the two clusters share many key tags such as "love", "travel", "personal" and "film". Furthermore, when looking into the large number of smaller tags in the clouds, we find that such tags in Fig. 5.8(a) are more related to "music" and

(a) (b)

Figure 5.8: The tag clouds generated for the (a) 1^{st} and (b) 4^{th} biggest clusters. A larger font of tag indicates a higher weight in the cluster.

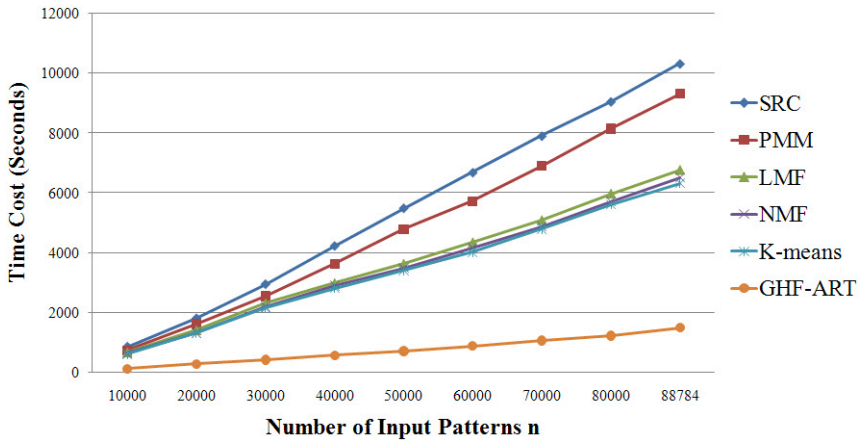


Figure 5.9: Time cost of GHF-ART, K-means, SRC, LMF, NMF and PMM on the BlogCatalog Data set with the increase in the number of input patterns.

enjoying "life", such as "game", "rap" and "sport", while those in Fig. 5.8(b) are more related to "family" life, such as "kids", "parenting" and "wedding". Therefore, although the shared key tags indicate their strong relations to the same class "Personal", they are separated into two communities due to the differences in the sub-key tags.

5.4.2.6 Time Cost Analysis

To evaluate the efficiency of GHF-ART on big data, we further analyzed the time cost of GHF-ART, K-means, SRC, LMF, NMF and PMM with the increase in the number of input patterns. To make a fair comparison, we set the number of clusters $k = 166$ for K-means, SRC, LMF, NMF and PMM and set $\rho = 0.2$ for GHF-ART so that the numbers

of the generated clusters for all the algorithms are the same. In Fig. 5.9, we observe that GHF-ART runs much faster than the other algorithms. Whereas the other algorithms incur a great increase of time cost with the increase in the number of input patterns, GHF-ART maintains a relatively small increase. This demonstrates the scalability of GHF-ART to big data.

Chapter 6

Adaptive Scaling of Cluster Boundaries for Large-Scale Social Media Data Clustering

The large-scale and complex nature of social media data raises the need to scale existing clustering techniques to big data and make them capable of automatically identifying the number of clusters instead of doing so empirically. This problem becomes more crucial for the heterogeneous data co-clustering tasks studied in Chapter 4 and Chapter 5, because the data objects usually have different distributions in different feature spaces. ART is a promising algorithm that boasts low time complexity and dependence on only a single parameter to identify clusters. In this chapter, we aim to make the vigilance parameter in ART self-adaptable in order to improve the robustness of the ART-based clustering algorithms to input parameters. To this end, we first provide a geometrical study of Fuzzy ART and define the vigilance region (VR), which is calculated from the vigilance criteria and can be used to interpret the clustering mechanism of Fuzzy ART. Subsequently, we propose three vigilance adaptation methods, namely, the Activation Maximization Rule (AMR), the Confliction Minimization Rule (CMR) and the Hybrid Integration Rule (HIR), which allow different clusters to have different vigilance levels that are adapted during the clustering process. Experiments on two social media data sets show that by incorporating AMR, CMR and HIR into Fuzzy ART, all of the resulting AM-ART, CM-ART and HI-ART are more robust than Fuzzy ART to the initial vigilance value, and they usually achieve better or comparable performance and much faster speed

than several existing clustering algorithms that do not require a pre-defined number of clusters.

6.1 Introduction

The popularity of social websites has resulted in a dramatic increase in online multimedia documents, such as images, blogs and tweets. In recent years, the clustering of web multimedia data from social websites has drawn much attention for social community discovery [1, 111], collective behavior analysis [160] and underlying topic discovery [72, 73]. However, different from traditional image and text data sets, social media data sets are usually large-scale and may cover diverse content across different topics, making it difficult to manually evaluate the number of underlying topics in the data sets. These challenging issues raise the need for existing clustering algorithms to be scalable to big data and capable of automatically, in stead of empirically, identifying the number of clusters in the data sets.

6.1.1 Related Work

Existing approaches for evaluating the number of clusters in a data set can be categorized into three types, namely, the cluster tendency analysis approach, the cluster validation approach, and the algorithms that do not require a pre-defined number of clusters.

The cluster tendency analysis approach [14, 96, 97] aims to visually identify the number of clusters in a data set before clustering by studying the reordered dissimilarity matrix of patterns so as to identify groups of patterns having similar properties. However, this approach requires heavy computation due to the involvement of eigenvalue decomposition and numerous mathematical transformation procedures, and the boundaries between dark blocks may not be distinct for complex data sets.

In contrast, the cluster validation approach [15, 98–102] determines the best clustering of patterns by quantitatively evaluating the quality of different cluster structures. Although this approach is studied widely, existing methods apply only to small data sets. Besides, recent work [96] has shown that existing validation methods may not perform consistently for real-world data sets due to their data structure assumptions.

Existing algorithms that require no pre-defined number of clusters include hierarchical clustering [103–105], genetic clustering [54, 55], density-based clustering [13, 60], Affinity Propagation [61], and the family of clustering algorithms [65–68, 154] based on Adaptive Resonance Theory [19]. Theoretically, the hierarchical clustering and genetic clustering algorithms are similar to the cluster validation approach, which generates different cluster structures of patterns and employs cluster validation methods to evaluate the quality of newly generated clusters so as to identify the best cluster structure. The density-based clustering algorithms, such as DBSCAN [13], identify clusters in a data set based on two parameters, namely, the search radius in the feature space and the minimum number of neighbors in the search area, to form the degree of density of clusters. However, the selection of the two parameters should be based on an analysis of the data set, which may be time consuming for large-scale and complex data sets. Affinity Propagation [61] is an exemplar-based clustering algorithm that iteratively identifies a set of patterns as "exemplars" that can represent other patterns in the same cluster by leveraging similarities between patterns. However, Affinity Propagation requires the tuning of four parameters, including the "preference" vector to control the number of generated clusters, the damping factor, and the maximum and minimum numbers of iterations needed to ensure convergence. Besides, both the density-based algorithms and Affinity Propagation typically require a quadratic time complexity of $O(n^2)$. The clustering algorithms based on Adaptive Resonance Theory (ART), such as Fuzzy ART [68], incrementally process patterns one at a time by performing real-time searching and matching between each input pattern and the formed clusters; a new cluster is identified when the input pattern cannot find a similar cluster within the existing clusters. The ART-based algorithms use one parameter, called the "vigilance parameter", to restrain the minimum degree of similarity for patterns in the same cluster. Moreover, the ART-based algorithms require a linear time complexity of $O(n)$.

Although the density-based clustering algorithms, Affinity Propagation and the algorithms based on Adaptive Resonance Theory do not require the number of clusters to be set, they employ other parameters to determine the properties of patterns in the same cluster. The advantages of ART-based algorithms over density-based clustering algorithms and Affinity Propagation include their low time complexity and use of a single ratio value (the vigilance parameter) to form clusters.

6.1.2 Proposed Approach

In this study, we discuss our investigation of making the vigilance parameter in Fuzzy ART self-adaptable so that, to fit clusters of different sizes in complex data, the clusters in the Fuzzy ART system will have individual vigilance levels that are able to adaptively tune their boundaries to accept similar patterns during the clustering process. Our contributions are threefold. First, we theoretically demonstrate that complement coding [68] significantly changes the clustering mechanism of Fuzzy ART. Secondly, we demonstrate that the vigilance parameter of a cluster in Fuzzy ART with complement coding actually forms a hyper-octagon region for the cluster, called a vigilance region (VR), in the high-dimensional feature space, which is centered by the weight of the cluster. By proving the properties of the VR, we show that it can provide a geometric interpretation of the clustering process of Fuzzy ART. Thirdly, in our early investigation [161], we proposed two heuristic methods, the Activation Maximization Rule (AMR) and the Confliction Minimization Rule (CMR), which allow clusters in ART to have individual vigilance parameters that self-adapt during the clustering process. In this study, using VR, we present a geometric interpretation of these methods and further propose a Hybrid Integration Rule (HIR) that considers aspects of both AMR and CMR.

AMR, CMR and HIR are applied on two social media data sets, including a subset of the NUS-WIDE image data set [2] collected from Facebook and a subset of the 20 Newsgroups data set [153] collected from netnews. Their performance is studied in terms of the robustness to the initial vigilance parameter, the convergence speed, the time-cost analysis, the clustering performance, and a comparison with Fuzzy ART, DBSCAN and Affinity Propagation in terms of purity [156], class entropy [150] and the Rand index [157]. The empirical results show that AMR, CMR and HIR are more robust than Fuzzy ART to the vigilance parameter and usually perform better than Fuzzy ART, DBSCAN and Affinity Propagation.

6.2 Fuzzy ART

The architecture of Fuzzy ART (Fig. 6.1) consists of input field F_1 for receiving the input patterns and category field F_2 for the clusters. The generic network dynamics of Fuzzy ART are described as follows.

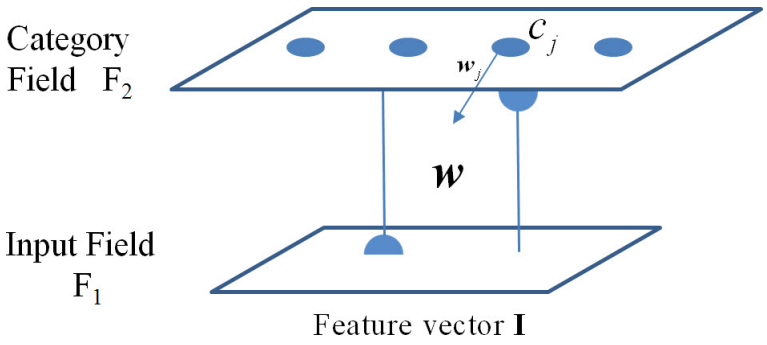


Figure 6.1: Fuzzy ART architecture.

Input vectors: Let $\mathbf{I} = \mathbf{x}$ denote the input pattern in input field F_1. With complement coding [68], $\mathbf{x}$ is further concatenated with its complement vector $\bar{\mathbf{x}}$ such that $\mathbf{I} = [\mathbf{x}, \bar{\mathbf{x}}]$.

Weight vectors: Let $\mathbf{w}_j$ denote the weight vector associated with the jth cluster c_j $(j = 1, ..., J)$ in category field F_2.

Parameters: The Fuzzy ART's dynamics are determined by choice parameter $\alpha > 0$, learning parameter $\beta \in [0, 1]$ and vigilance parameter $\rho \in [0, 1]$.

The clustering process of Fuzzy ART has three key steps:

1) Category choice: For each input pattern $\mathbf{I}$, Fuzzy ART calculates the choice function for all of the clusters in category field F_2 and selects the most suitable cluster (winner) c_{j^*}, which has the largest value. The choice function for the jth cluster c_j is defined by

$$T_j = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|}, \tag{6.1}$$

where the fuzzy AND operation $\wedge$ is defined by $(\mathbf{p} \wedge \mathbf{q})_i \equiv \min(p_i, q_i)$, and the norm $|.|$ is defined by $|\mathbf{p}| \equiv \sum_i p_i$.

2) Template matching: The similarity between input pattern $\mathbf{I}$ and winner c_{j^*} is evaluated using a match function M_{j^*}, which is defined by

$$M_{j^*} = \frac{|\mathbf{I} \wedge \mathbf{w}_{j^*}|}{|\mathbf{I}|}. \tag{6.2}$$

If the winner satisfies the vigilance criteria such that $M_{j^*} \geq \rho$, a resonance will occur, which leads to the learning step. Otherwise, a new winner will be selected among the

rest of the clusters in the category field. If no winner satisfies the vigilance criteria, a new cluster will be generated to encode the input pattern.

3) Prototype learning: If c_{j^*} satisfies the vigilance criteria, its corresponding weight vector $\mathbf{w}_{j^*}$ will be updated through a learning function, defined by

$$\mathbf{w}_{j^*}^{(new)} = \beta(\mathbf{I} \wedge \mathbf{w}_{j^*}) + (1 - \beta)\mathbf{w}_{j^*}. \tag{6.3}$$

6.3 Complement Coding and Vigilance Region in Fuzzy ART

The vigilance region (VR) of a cluster, calculated based on the vigilance criteria, is geometrically defined by a region associated to the cluster in the feature space. It provides a geometric interpretation of the vigilance criteria in Fuzzy ART that the input patterns falling into VRs are considered to be similar to the corresponding clusters.

The shapes and functional behaviors of a VR depend on the use of complement coding. As demonstrated in [68], with complement coding, the weight vector can be represented by a hyper-rectangle in the feature space. In this case, the VR is a hyper-octagon centered by the weight hyper-rectangle and it shrinks as the cluster size expands;

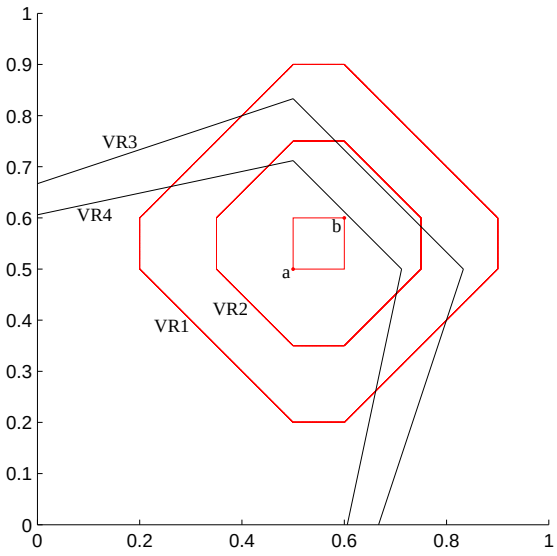


Figure 6.2: Geometric display of a cluster and its vigilance regions with or without complement coding in Fuzzy ART in 2D space.

otherwise, without complement coding, the VR is an irregular hyper-polygon with axes (details can be found in [68]). A 2D example is shown in Fig. 6.2. Without complement coding, point a denotes the weight vector of the cluster, and the corresponding VRs under vigilance parameters $\rho = 0.75$ and $\rho = 0.825$ are VR3 and VR4 respectively, as indicated by the black line. With complement coding, the weight vector is represented by the red rectangle with vertices a and b, and the corresponding VRs under vigilance parameters $\rho = 0.75$ and $\rho = 0.825$ are represented by the red octagons VR1 and VR2 respectively.

6.3.1 Complement Coding in Fuzzy ART

Complement coding [68] is employed in Fuzzy ART as a normalization method for the input patterns, which prevents cases in which the values of the weight vector of a cluster decrease to such a low level that the cluster is no longer representative of its category, and a set of new clusters must be generated to encode input patterns of this category; this is known as the problem of category proliferation.

However, complement coding significantly changes the clustering mechanism of Fuzzy ART.

6.3.1.1 Effect of Complement Coding on Category Choice

Choice function Equation 6.1 evaluates the degree to which the weight vector $\mathbf{w}_j$ of cluster c_j is a subset of input pattern $\mathbf{I}$. By employing complement coding in the ART learning system, weight vector $\mathbf{w}_j$ is concatenated by two parts, namely, the feature part that, in each dimension, has the smallest value among all of the vertices of the weight hyper-rectangle, and the complement part, whose complement vector, in contrast to the feature part, has the largest value among vertices in each dimension. For example, in a 2D feature space, as shown in Fig. 6.2, the feature part of weight vector $\mathbf{w}_j$ is point a, and the corresponding complement part is the complement vector of point b.

We can prove that, with complement coding, the choice function considers the similarity between the input pattern and the weight hyper-rectangle of the selected cluster.

Property 1 *Given the input pattern* $\mathbf{I} = (\mathbf{x}, \bar{\mathbf{x}})$*, weight vector* $\mathbf{w}_j = (\mathbf{a}, \bar{\mathbf{b}})$ *of cluster* c_j*, and* $\alpha \approx 0$*, choice function* T_j *considers the similarities between the original input pattern* $\mathbf{x}$ *and the weight hyper-rectangle of cluster* c_j*.*

Proof:

$$\begin{aligned} T_j &= \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} \\ &= \frac{|\mathbf{x} \wedge \mathbf{a}| + |\bar{\mathbf{x}} \wedge \bar{\mathbf{b}}|}{|\mathbf{w}_j|} \\ &= \frac{|\mathbf{x} \wedge \mathbf{a}| + |\overline{\mathbf{x} \vee \mathbf{b}}|}{|\mathbf{a} + \bar{\mathbf{b}}|} \\ &= \frac{|\mathbf{a}|}{|\mathbf{a} + \bar{\mathbf{b}}|} \cdot \frac{|\mathbf{x} \wedge \mathbf{a}|}{\alpha + |\mathbf{a}|} + \frac{|\bar{\mathbf{b}}|}{|\mathbf{a} + \bar{\mathbf{b}}|} \cdot \frac{|\overline{\mathbf{x} \vee \mathbf{b}}|}{\alpha + |\bar{\mathbf{b}}|}. \end{aligned} \tag{6.4}$$

As shown in Equation 6.4, the choice function evaluates both the degree to which $\mathbf{a}$ is a subset of $\mathbf{x}$ and the degree to which $\mathbf{x}$ is a subset of $\mathbf{b}$. The final choice value is obtained by their weighted summation, which is normalized by their respective norms.

Therefore, given $\mathbf{x} = (x_1, ..., x_m)$, $\mathbf{a} = (a_1, ..., a_m)$ and $\mathbf{b} = (b_1, ..., b_m)$, choice function T_j achieves its maximum for c_j when, for $\forall i \in [1, m]$, $a_i \leq x_i \leq b_i$. For example, in Fig. 6.2, the choice function for this cluster achieves its maximum when the input pattern falls into the weight rectangle.

It may be concluded that the further the input pattern is from the weight hyper-rectangle, the lower the value that the choice function achieves for the cluster. Given that Equation 6.1 and Equation 6.2 share the same numerator, for $\forall \varepsilon \in (0, 1)$, $T_j = \frac{|\mathbf{I} \wedge \mathbf{w}_j|}{\alpha + |\mathbf{w}_j|} = \varepsilon$ produces a VR-like hyper-octagon.

Therefore, with complement coding, the choice function evaluates the similarity of the input pattern to the weight hyper-rectangle of the selected cluster c_j.

6.3.1.2 Effect of Complement Coding on Template Matching

Match function Equation 6.2 evaluates the degree to which the input pattern $\mathbf{I}$ is a subset of weight vector $\mathbf{w}_{j^*}$ of cluster c_{j^*}. In template matching, input pattern $\mathbf{I}$ is considered to be similar to the winner cluster c_{j^*} if

$$M_{j^*} = \frac{|\mathbf{I} \wedge \mathbf{w}_{j^*}|}{|\mathbf{I}|} \geq \rho. \tag{6.5}$$

The VR therefore is identified to show the extent to which an input pattern can be categorized into a specific cluster. Given the weight vector $\mathbf{w}_{j^*} = (w_1, ..., w_m)$ of cluster c_{j^*}, the vigilance parameter ρ, and an arbitrary input pattern $\mathbf{I} = (x_1, ..., x_m)$ in the Fuzzy ART system. If Fuzzy ART does not employ complement coding, Equation 6.5 is equivalent to

$$\Sigma_{i=1}^{m} \min(x_i, w_i) - \rho \Sigma_{i=1}^{m} x_i \geq 0. \tag{6.6}$$

As shown in Fig. 6.2, when $m = 2$, Equation 6.6 is an irregular polygon constructed by three functions and the horizontal and vertical axes.

In contrast, if Fuzzy ART employs complement coding, the number of dimensions of the feature space will be $\frac{m}{2}$. Therefore, Equation 6.5 can be expressed as

$$\Sigma_{i=1}^{m} \min(x_i, w_i) \geq \frac{m\rho}{2}. \tag{6.7}$$

When $m = 4$, as shown in Fig. 6.2, the VR of c_{j^*} becomes a regular polygon, namely, an octagon.

6.3.2 Vigilance Region in Fuzzy ART

6.3.2.1 Properties of Weight Hyper-Rectangle and Vigilance Region

In Section 6.3.1, we demonstrated the effect of complement coding on Fuzzy ART, particularly proving that, with complement coding, the VR of a cluster becomes a hyper-octagon centered by the weight vector of the cluster, namely, the weight hyper-rectangle. In this section, we analyze the properties of the weight hyper-rectangle and VR of a cluster and subsequently use them to interpret the clustering process of Fuzzy ART.

Property 2 *Given the weight vector* $\mathbf{w}_j = (w_1, ..., w_m)$ *of cluster* c_j *in the Fuzzy ART system with complement coding, the VR of* c_j *consists of* $3^{\frac{m}{2}} - 1$ *hyper-planes.*

Proof: Similar to Equation 6.4, given $\mathbf{w}_j = (a_1, ..., a_{\frac{m}{2}}, \bar{b}_1, ..., \bar{b}_{\frac{m}{2}})$, $\mathbf{I} = (\mathbf{x}, \bar{\mathbf{x}}) = (x_1, ..., x_{\frac{m}{2}}, \bar{x}_1, ..., \bar{x}_{\frac{m}{2}})$, Equation 6.5 can by expressed as

$$\Sigma_{i=1}^{\frac{m}{2}} \min(x_i, a_i) + \Sigma_{i=1}^{\frac{m}{2}} \overline{\max(x_i, b_i)} \geq \frac{m\rho}{2}. \tag{6.8}$$

In view that the m dimensional vector $\mathbf{w}_j$ is a hyper-rectangle in the $\frac{m}{2}$ dimensional space, and for $\forall i \in [1, \frac{m}{2}]$, $x_i \in [0, a_i) \cup [a_i, b_i) \cup [b_i, 1]$. Therefore, the feature space is divided into $3^{\frac{m}{2}}$ subsections.

Considering that Equation 6.8 is an identical equation in the weight hyper-rectangle, the number of hyper-planes for constructing the VR is $3^{\frac{m}{2}} - 1$.

Property 3 *Patterns falling into the weight hyper-rectangle have the same value of match function defined by Equation 6.2.*

Proof: Given a cluster c_j and its weight vector $\mathbf{w}_j = (a_1, ..., a_{\frac{m}{2}}, \bar{b}_1, ..., \bar{b}_{\frac{m}{2}})$ and $\mathbf{I} = (x_1, ..., x_{\frac{m}{2}}, \bar{x}_1, ..., \bar{x}_{\frac{m}{2}})$ falling into the weight hyper-rectangle, we have for $\forall i \in [1, \frac{m}{2}]$, $a_i \leq x_i \leq b_i$.

In this case, according to Equation 6.8, the value of the match function depends only on weight vector $\mathbf{w}_j$ such that $|\mathbf{I} \wedge \mathbf{w}_{j^*}| = \mathbf{w}_{j^*}$. Therefore, all of the patterns in the weight hyper-rectangle have the same match value.

The situation may also be interpreted as all of those patterns having the same ℓ_1 distance to $\mathbf{a}$ and $\mathbf{b}$, as

$$\begin{aligned} |\mathbf{x} - \mathbf{a}| + |\mathbf{x} - \mathbf{b}| &= \sum_i (x_i - a_i) + \sum_i (b_i - x_i) \\ &= \sum_i (x_i - a_i + b_i - x_i) = \sum_i (b_i - a_i). \end{aligned} \tag{6.9}$$

Property 4 *Patterns falling into the weight hyper-rectangle of the winner do not result in the expansion of the weight hyper-rectangle during learning step defined by Equation 6.3.*

Proof: In *Property* 2, if $\mathbf{I}$ falls into the weight hyper-rectangle of cluster c_j, $|\mathbf{I} \wedge \mathbf{w}_{j^*}| = \mathbf{w}_{j^*}$. In this case, Equation 6.3 is equivalent to

$$\mathbf{w}_{j^*}^{(new)} = \beta \mathbf{w}_{j^*} + (1 - \beta)\mathbf{w}_{j^*} = \mathbf{w}_{j^*}. \tag{6.10}$$

Therefore, weight vector $\mathbf{w}_{j^*}$ undergoes no change after encoding input pattern $\mathbf{I}$.

Property 5 *The weight hyper-rectangle of a cluster reflects the cluster size, which is controlled by the learning rate β.*

Proof: Given input pattern $\mathbf{I} = (\mathbf{x}, \bar{\mathbf{x}})$, winner c_{j^*} and its corresponding weight vector $\mathbf{w}_{j^*} = (\mathbf{a}, \bar{\mathbf{b}})$, if $\mathbf{I}$ is categorized into c_{j^*}, $\mathbf{w}_{j^*}$ is updated according to Equation 6.3 such that

$$\begin{aligned}\mathbf{w}_{j^*}^{(new)} &= (\mathbf{a}^{(new)}, \bar{\mathbf{b}}^{(new)}) = \beta(\mathbf{I} \wedge \mathbf{w}_{j^*}) + (1-\beta)\mathbf{w}_{j^*} \\ &= \beta((\mathbf{x}, \bar{\mathbf{x}}) \wedge (\mathbf{a}, \bar{\mathbf{b}})) + (1-\beta)(\mathbf{a}, \bar{\mathbf{b}}) \\ &= \beta(\mathbf{x} \wedge \mathbf{a}, \overline{\mathbf{x} \vee \mathbf{b}}) + (1-\beta)(\mathbf{a}, \bar{\mathbf{b}}) \\ &= (\beta(\mathbf{x} \wedge \mathbf{a}) + (1-\beta)\mathbf{a}, \beta(\overline{\mathbf{x} \vee \mathbf{b}}) + (1-\beta)\bar{\mathbf{b}})). \end{aligned} \tag{6.11}$$

From Equation 6.11, we observe that the update of weight vector $\mathbf{w}_{j^*}$ is essentially the movement of $\mathbf{a}$ and $\mathbf{b}$ towards the input pattern $\mathbf{I}$. Specifically, $\mathbf{a}$ moves towards $\mathbf{I}$ in the dimensions $\{i|x_i < a_i\}$, while $\mathbf{b}$ moves towards $\mathbf{I}$ in the dimensions $\{i|x_i > b_i\}$.

Therefore, when learning parameter β equals 1, the weight hyper-rectangle of c_{j^*} covers all of the patterns in c_{j^*}, which indicates the boundaries of c_{j^*}. When $\beta < 1$, the weight hyper-rectangle expands towards the new patterns to some extent, making it unable to cover all the patterns. However, the weight hyper-rectangle may reflect the cluster size in a smaller scale.

Property 6 *The VR shrinks as the weight hyper-rectangle expands in order to control the minimum intra-cluster similarity.*

Proof: As demonstrated in *Property* 2, a VR in the $\frac{m}{2}$ dimensional space is constructed by $3^{\frac{m}{2}} - 1$ functions, each of which is calculated using Equation 6.8. Given the nature of the learning function defined by Equation 6.3, which suppresses the values of features, following the definitions in *Property* 2, we have, for $\forall i \in [1, \frac{m}{2}], a_i^{(new)} \leq a_i$ and $b_i^{(new)} \geq b_i$. Therefore, after the weight hyper-rectangle expands, the constant in the left part of Equation 6.8 decreases. In this situation, functions in the subsections either remain the same or move towards the weight hyper-rectangle.

Interestingly, if an input pattern $\mathbf{I}$ causes the weight hyper-rectangle of a cluster c_j to expand, the function of the VR in the subsection to which $\mathbf{I}$ belongs will remain the

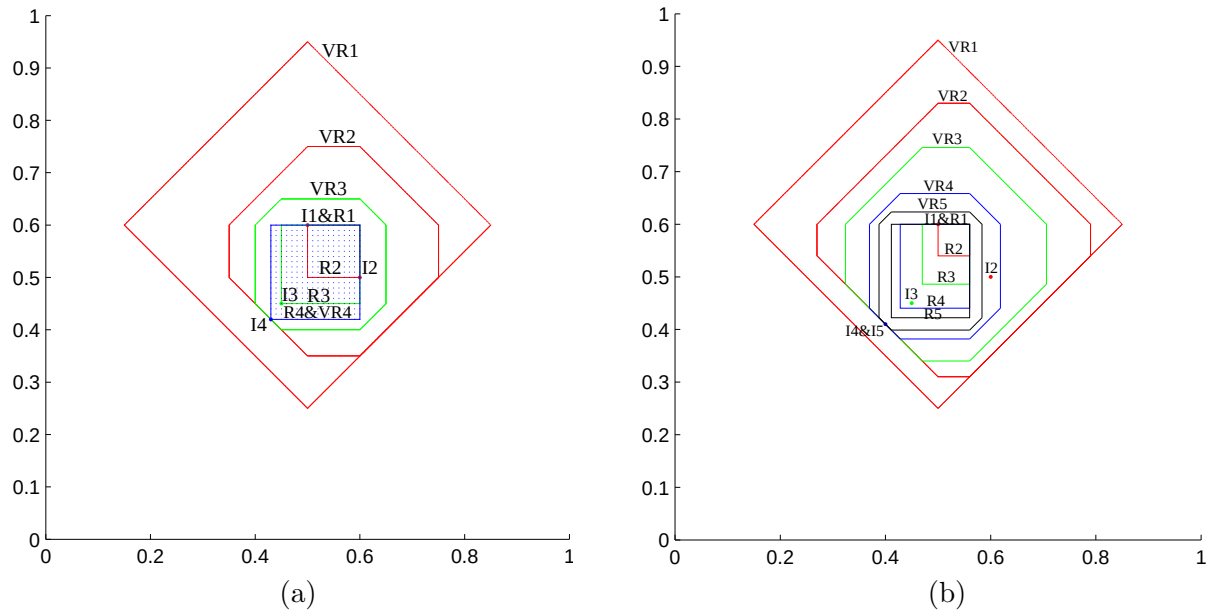


Figure 6.3: 2D example on the evolution of a cluster in Fuzzy ART under different learning parameter values (a) $\beta = 1$ and (b) $\beta = 0.6$. R1-R4 indicate the expansion of the cluster's weight rectangle and VR1-VR4 indicate the corresponding VRs.

same. Following the definitions in *Property* 2, if **I** causes the movement of **a** in the ith dimension, we have $x_i \leq a_i^{(new)} < a_i$. However, for this dimension, $min(x_i, a_i^{(new)}) = x_i$. Therefore, the function of the VR in that subsection is not related to the value of a_i. A similar conclusion can be drawn regarding the movement of **b**.

The shrinking of the VR can also be understood from another perspective. As the VR indicates the boundaries that the weight hyper-rectangle expands towards in all directions, when the weight hyper-rectangle expands in one direction, its distance from the VR is determined by the function in that direction, and the functions in other directions should shrink to meet this distance so that the updated VR remains a regular hyper-octagon centered by the weight hyper-rectangle.

6.3.2.2 Interpretation of Clustering Process of Fuzzy ART using Vigilance Region

Given the above properties of the weight hyper-rectangle and the VR, the clustering process of Fuzzy ART can be interpreted using a 2D example, as shown in Fig. 6.3. Fig. 6.3(a) depicts the evolution of a cluster in Fuzzy ART with the sequential presentation of I1(0.5,0.6), I2(0.6,0.5), I3(0.45,0.45), and I4(0.43,0.42), under the learning

parameter $\beta = 1$ and the vigilance parameter $\rho = 0.825$. When the cluster has only one pattern I1, the weight rectangle R1 is situated exactly at point I1. In this case, the corresponding VR1 is a square diamond centered by I1. After the encoding of I2, R2 becomes a rectangle, and the corresponding VR becomes an octagon, which satisfies *Property* 2. During the presentation of the subsequent patterns, the weight rectangle expands to cover all of the patterns, which satisfies *Property* 5. It is notable that I4 lies directly on the edge of VR3 and, after learning from I4, VR4 overlaps with R4. Based on *Property* 3 and *Property* 4, patterns falling into R4 have the same match function value, and this cluster will no longer expand. Also, the bottom-left edge of VR2-VR4, where I4 lies, never shrinks. This is because the weight rectangle always expands in this direction, which can be interpreted by the conclusion in *Property* 6.

Similarly, Fig. 6.3(b) shows the evolution of a cluster with the sequential presentation of I1(0.5,0.6), I2(0.6,0.5), I3(0.45,0.45), I4(0.4,0.41), and I5(0.4,0.41) under $\beta = 0.6$ and $\rho = 0.825$. We observe that, with a smaller learning parameter $\beta = 0.6$, R1 expands towards I2, but it cannot cover both I1 and I2 as shown in Fig. 6.3(a). However, with a smaller size of R2, the corresponding VR2 covers a larger region than that depicted in Fig. 6.3(a). Different from the behavior illustrated in Fig. 6.3(a), a repeated presentation I5 of input pattern I4, as shown in Fig. 6.3(b), still cause the cluster to learn. Therefore, when $\beta < 1$, the continuous presentation of the same pattern to the same cluster results in the gradual expansion of the weight rectangle of the cluster towards the input pattern. However, the cluster rectangle cannot cover that pattern due to the learning function of Fuzzy ART.

6.3.2.3 Discussion

The VR provides a geometric understanding of how Fuzzy ART works. As shown in Fig. 6.2, without complement coding, the VR of Fuzzy ART in a 2D space is an open region, so the weight vector of the cluster denoted by point a may gradually move to the origin, which causes category proliferation. With complement coding, as shown in Fig. 6.3, the VR of a cluster in Fuzzy ART is a regular polygon, which shrinks as the cluster size expands. Therefore, Fuzzy ART with complement coding tends to partition the high-dimensional feature space into regions of hyper-rectangles.

The geometric interpretation of Fuzzy ART is also helpful for deducing and improving its limitations. First, given that the VR of a new cluster is usually much larger than the weight rectangle of the cluster and shrinks quickly after the encoding of the subsequent patterns, it may be difficult to cover a group of patterns using a single cluster, even if the VR covers all of the patterns. Secondly, a small VR may result in the generation of multiple clusters to cover a group of patterns. Thirdly, a large VR may incur an incorrect categorization of patterns, as the sequence of input patterns is unknown. Therefore, the performance of Fuzzy ART depends greatly on the value of vigilance parameter ρ, and the clustering results may differ with different sequences of input patterns.

6.4 Rules for Adapting Vigilance Parameter in Fuzzy ART

As discussed in 6.3.2.3, the performance of Fuzzy ART depends greatly on the value of vigilance parameter ρ, which both determines the VRs of clusters for accepting patterns and limits the size of all of the clusters. However, because large-scale web multimedia data usually contain a large number of groups of patterns in arbitrary shapes in the feature space, it is not advisable to use a single value of the vigilance parameter in ART to scale the size of all clusters.

Based on the above consideration, in a recent study [161], we proposed two heuristic methods, the Activation Maximization Rule (AMR) and the Confliction Minimization Rule (CMR), which allow different clusters in ART to have individual vigilance parameters and make the vigilance parameters self-adaptable during the clustering process. In the following sections, we offer a geometric interpretation of AMR and CMR using VR and further propose a hybrid method to integrate AMR and CMR.

6.4.1 Activation Maximization Rule

The Activation Maximization Rule (AMR) comes from the observation that, on one hand, with a small vigilance value, input patterns are likely to incur resonances for the same cluster, while on the other hand, a large vigilance value may lead to the reset of input patterns for all clusters in the category field, requiring the creation of a new cluster.

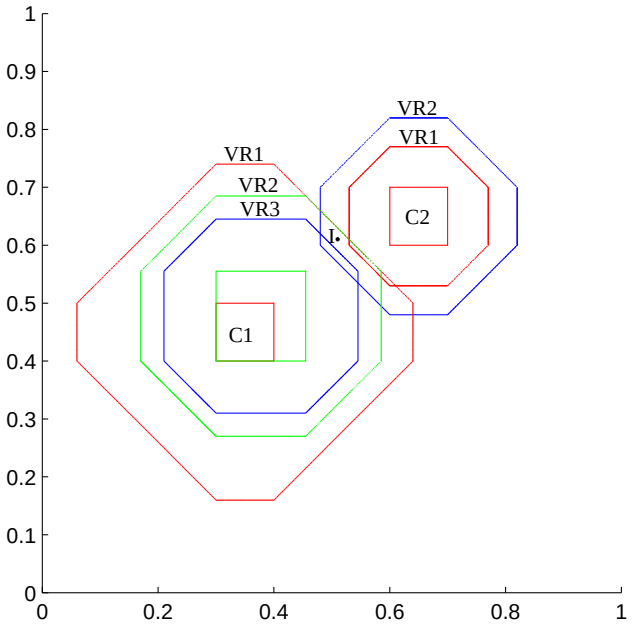


Figure 6.4: A 2D example on how AMR adapts vigilance parameters of two cluster in Fuzzy ART with complement coding.

Therefore, AMR is proposed to restrain the continuous activation of the same cluster and promote the activation of clusters that usually incur resets.

Specifically, AMR adapts the vigilance parameter ρ_{j^*} of the winner c_{j^*} when

1) **Resonance occurs**: $\rho_{j^*}^{(new)} = (1+\sigma)\rho_{j^*}$;

2) **Reset occurs**: $\rho_{j^*}^{(new)} = (1-\sigma)\rho_{j^*}$.

The restraint parameter $\sigma \in [0, 1]$ controls the degree to which the vigilance parameter increases or decreases. With a small σ, AMR incurs small changes in the vigilance values of clusters, so the performance of ART may still depend on the initial value of the vigilance parameter. In contrast, a large σ may help to make AM-ART more robust to the initial vigilance value but may result in unstable vigilance values of clusters, which could increase the risk of pattern mis-categorization.

Fig. 6.4 illustrates how AMR works. C1 and C2 are two clusters with different values of vigilance parameter. When the input pattern I is presented, C2 is the first winner. However, C2 incurs a reset due to its small VR. Subsequently, the next winner C1 encodes input pattern I. Without AMR, the VR of C2 remains the same, and that of C1 shrinks from VR1 to VR2. Therefore, if another input pattern close to I is presented, it will be

mis-categorized to C1 again. However, with AMR, the VR of C2 expands from VR1 to VR2, and that of C1 shrinks from VR1 to VR3. In this case, C2 can successfully encode input pattern I. If the initial vigilance value is large, we may easily find that AMR may increase the VRs of clusters to alleviate the over-generation of clusters. Therefore, AMR may help to improve the clustering performance of Fuzzy ART when the initial vigilance value is not suitable.

Notably, AMR may also help to even out the sizes of two very close clusters by quickly shrinking the VR of the cluster that encodes more patterns, which may help to prevent the generation of small clusters and the over-generalization of cluster weights.

6.4.2 Confliction Minimization Rule

The Confliction Minimization Rule (CMR) minimizes the overlap between VRs of close clusters to produce better cluster boundaries. CMR is based on the idea that, in Fuzzy ART, the incorrect recognition of patterns usually is caused by a small vigilance value, so the VR of a cluster may cover patterns from other classes. Therefore, well-partitioned boundaries between clusters can minimize the risk of mis-categorization.

Specifically, CMR in Fuzzy ART has three key steps:

1) **Candidate Selection**: Select all winner candidates $\mathcal{W}inc = \{c_j | M_j \geq \rho\}$ in category field F_2 through the match function defined by Equation 6.2. If no candidates are selected, CMR stops;

2) **Winner Identification**: Identify the winner c_{j^*} from all candidates through the choice function defined by Equation 6.1 such that $j^* = \arg\max_j T_j$;

3) **Confliction Minimization**: Update the vigilance parameters of all winner candidates except the winner $\{c_j | c_j \in \mathcal{W}inc \wedge j \neq j^*\}$ using $\rho_j^{(new)} = M_j + \Delta$ ($\Delta \approx 0$ is a positive value).

CMR requires Fuzzy ART to first identify all winner candidates to the input pattern through the match function. After the winner is identified in the second step, the vigilance values of all other candidates are increased to slightly higher than their respective match values. In this way, the winner is more likely to encode the subsequent input patterns that are close to the current input pattern, and the overlap between the VRs of those

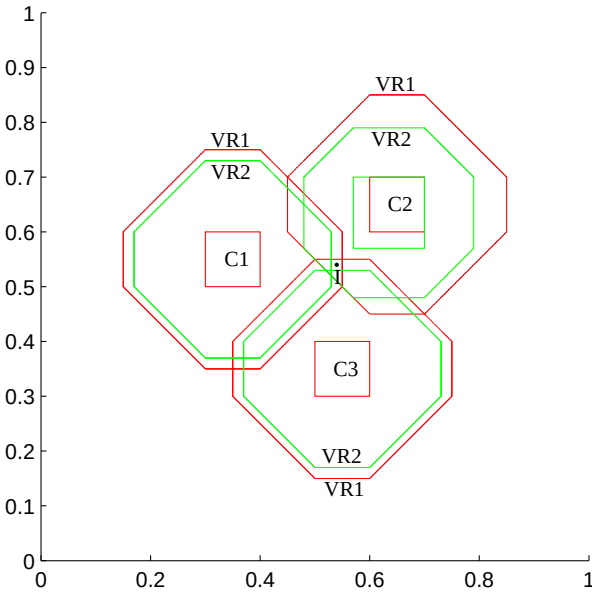


Figure 6.5: 2D example on how CMR adapts vigilance values of clusters in order to reduce overlap between their VRs.

candidates will decrease. However, when the initial vigilance value is high, unlik AMR, CMR cannot alleviate the over-generation of clusters.

Different from AMR which requires no changes to the clustering procedures of Fuzzy ART, CMR requires a change in the sequence of category choice and template matching steps. Therefore, to employ CMR in Fuzzy ART, the category choice and template matching steps should be replaced using the CMR procedures.

Fig. 6.5 illustrates the ability of CMR to reduce the overlap between the VRs of clusters. C1-C3 are three clusters with corresponding VRs denoted by VR1, and I is an input pattern falling at the overlap between VRs of all clusters. C2 encodes the input pattern I and its VR shrinks from VR1 to VR2. Without CMR, the overlap between all three clusters does not decrease. While with CMR, the VRs of C1 and C3 shrinks from VR1 to VR2 accordingly. Therefore, the overlap undergoes significant reduction. However, the improved VRs still cannot scale the boundaries between the clusters well because Fuzzy ART cannot decide which cluster best fits the patterns falling within the overlapping areas based on existing knowledge learned from the patterns.

6.4.3 Hybrid Integration of AMR and CMR

AMR and CMR are inspired by different considerations for ART and have different mechanisms when embedding in ART, so we cannot simply combine them into a single framework. However, we may simultaneously integrate the ideas of AMR and CMR. Specifically, AMR essentially rewards the clusters that have larger choice values than the winner but incur resets due to a large vigilance value, while penalizing the clusters that incur resonances to avoid a potentially low vigilance value. In contrast, CMR minimizes the overlap between the VRs of clusters. Therefore, the objectives of both AMR and CMR could be achieved using a single framework.

The implementation of the hybrid method, called the Hybrid Integration Rule (HIR), may follow the procedures of either AMR or CMR. Following AMR, after the winner is identified, HIR will subsequently discover all of the winner candidates and apply CMR to minimize the overlap of their VRs. Following CMR, after the winner is identified, HIR will subsequently search for all of the clusters having choice values equal or greater than the winner and decrease their vigilance values according to AMR.

For the purpose of time efficiency, HIR is implemented according to the procedures of CMR, as listed below:

1) **Candidate Selection**: Select all winner candidates $\mathcal{W}inc = \{c_j | M_j \geq \rho\}$ in category field F_2 through the match function Equation 6.2. If no candidates are selected, for $\forall c_j \in F_2$, set $\rho_j^{(new)} = (1-\sigma)\rho_j$, and HIR stops;

2) **Winner Identification**: Identify the winner c_{j^*} from all candidates through the choice function Equation 6.1 such that $j^* = \arg\max_j T_j$. Set $\rho_{j^*}^{(new)} = (1+\sigma)\rho_{j^*}$;

3) **Confliction Minimization**: Update the vigilance parameters of all winner candidates $\{c_j | c_j \in \mathcal{W}inc \wedge j \neq j^*\}$, except the winner, through $\rho_j^{(new)} = M_j + \Delta$;

4) **Activation Maximization**: Search in the remaining clusters to identify the set of clusters $\mathcal{R}c = \{c_j | c_j \in F_2 \wedge c_j \notin \mathcal{W}inc \wedge T_j \geq T_{j^*}\}$ and for $\forall c_j \in \mathcal{R}c$, set $\rho_j^{(new)} = (1-\sigma)\rho_j$.

HIR identifies all of the neighboring clusters of the input pattern to minimize the overlap of their VRs while simultaneously increasing the vigilance value of the winner and decreasing those values of the clusters that have equal or larger choice values but incur resets. Therefore, HIR takes advantage of both AMR and CMR.

6.5 Experiments

6.5.1 NUS-WIDE Data Set

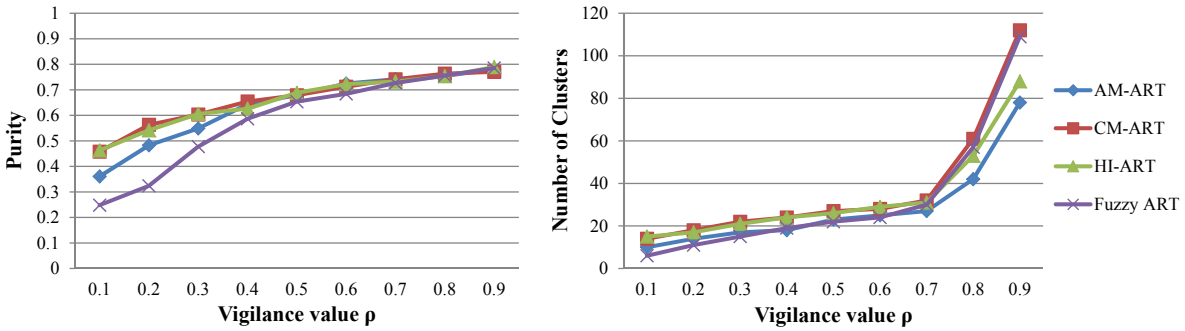
The NUS-WIDE data set [2] is a large web image set crawled from the famous photo sharing website *Flickr.com*. This data set consists of 269,648 images with respective surrounding text and ground-truth labels from 81 concepts.

To study the clustering performance of our methods on large-scale data sets, we collected a total of 10,800 images belonging to nine classes from the NUS-WIDE data set, including dog, bear, cat, bird, flower, lake, sky, sunset and wedding, each of which contains 1,200 images. We utilized a concatenation of three types of visual features to represent each image, including Grid Color Moment (225 features), Edge Direction Histogram (73 features) and Wavelet Texture (128 features). Therefore, each image was represented as a vector of 426 features.

6.5.1.1 Robustness to Vigilance Parameter

We first evaluated the performance of the proposed vigilance adaptation rules applied to Fuzzy ART, namely, AMR, CMR and HIR. We refer the resulting three Fuzzy ART models as AM-ART, CM-ART and HI-ART respectively.

In the experiments, we consistently used choice parameter $\alpha = 0.01$ and learning parameter $\beta = 0.6$ for Fuzzy ART, AM-ART, CM-ART and HI-ART, as the performance



(a) (b)

Figure 6.6: Clustering performance of AM-ART, CM-ART, HI-ART and Fuzzy ART under different vigilance values in terms of (a) cluster quality measured by purity and (b) number of generated clusters on NUS-WIDE data set.

of ART has generally demonstrated robustness to those two parameters with modest values [1, 155]. Additionally, we set restraint parameter $\sigma = 0.1$ for AM-ART and HI-ART, which, similar to the analysis in [161], usually yields satisfactory performance.

To study the effectiveness of the proposed methods, we evaluated their clustering performance in terms of cluster quality, as measured by purity [156], and the number of clusters generated under different vigilance values, as shown in Fig. 6.6. Fig. 6.6(a) illustrates that AM-ART, CM-ART and HI-ART achieved much better performance than Fuzzy ART when $\rho < 0.4$. Particularly, AM-ART performed 10 percent better than Fuzzy ART. CM-ART and HI-ART both performed better than AM-ART and nearly twice as well as Fuzzy ART. When $\rho > 0.7$, all four algorithms achieved comparable performance. As Fig. 6.6(b) illustrates, all of the algorithms produced a stable increase in the number of clusters when $\rho < 0.7$, and a significant increase when $\rho > 0.7$. This phenomenon could indicate that the patterns belonging to the same class were not well-covered by the VRs of the clusters. Comparing AM-ART with Fuzzy ART reveals that the former identified more clusters than the latter when $\rho < 0.3$ and generated significantly fewer clusters when $\rho > 0.8$. In contrast, CM-ART identified more clusters than AM-ART and Fuzzy ART when $\rho < 0.7$ and generated a similar number of clusters as Fuzzy ART. Interestingly, HI-ART generated similar number of clusters as CM-ART when $\rho < 0.7$ but significantly fewer when $\rho > 0.8$.

Based on the above observation, AM-ART may improve the clustering performance of Fuzzy ART by identifying more clusters when the initial vigilance value is low and by greatly simplifying the generated cluster network when the initial vigilance value is high. CM-ART may achieve better performance than AM-ART in terms of improving the robustness of Fuzzy ART to the initial vigilance value. However, it does not help to simplify the network when the initial vigilance value is high. More importantly, HI-ART may achieve performance comparable to CM-ART in terms of cluster quality and may simplify the cluster network more so than CM-ART. These findings indicate that, by taking into account the considerations of AMR and CMR, HI-ART takes advantage of both AM-ART and CM-ART.

6.5.1.2 Case Study on Cluster Analysis

This section provides a further analysis of the cluster structures generated by AM-ART, CM-ART, HI-ART and Fuzzy ART under different vigilance values in terms of the distribution of clusters of different sizes and of the different densities measured by the average pattern-centroid Euclidean distance. In the experiments, we followed the parameter settings as used in the previous section and reported the results under $\rho = 0.2$ and $\rho = 0.9$ due to page limitations, as shown in Fig. 6.7.

Fig. 6.7(a) illustrates that, under $\rho = 0.2$, Fuzzy ART identified fewer clusters, including just one cluster with 400 patterns and 6 clusters with more than 1000 patterns. Compared with Fuzzy ART, AM-ART identified more smaller clusters. In contrast, CM-ART and HI-ART identified relatively more smaller clusters, typically with 400-800 patterns. Fig. 6.7(b) illustrates the distribution of the average pattern-centroid distance of clusters. Many clusters identified by Fuzzy ART had distances larger than 5. In contrast, most of the clusters generated by AM-ART, CM-ART and HI-ART had distances

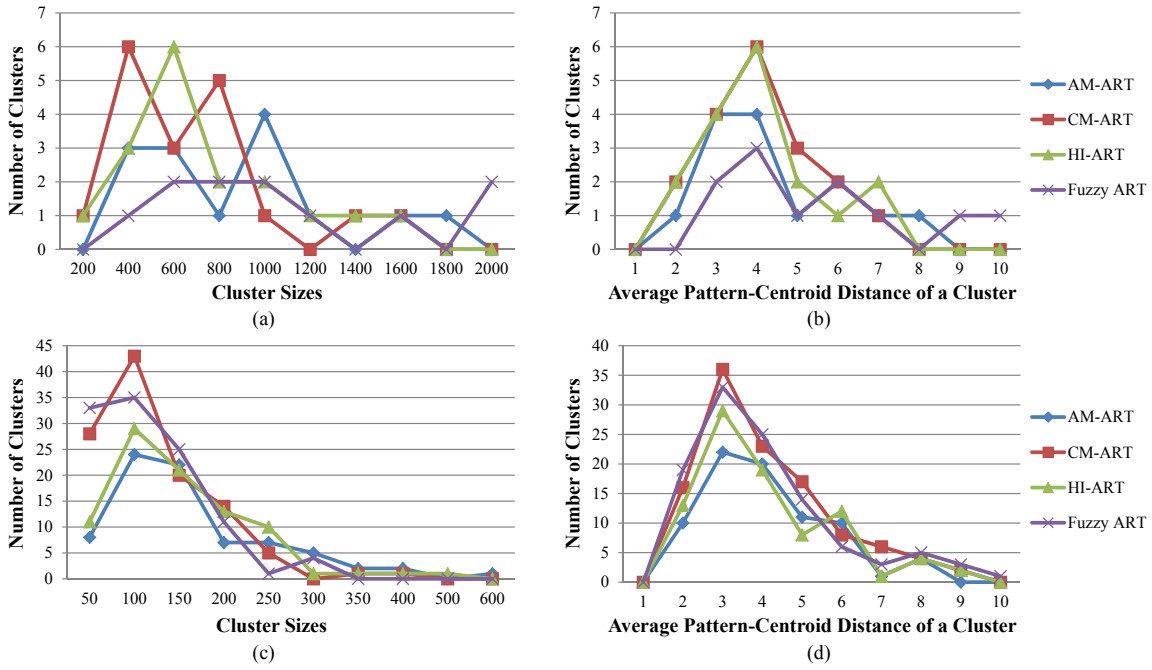


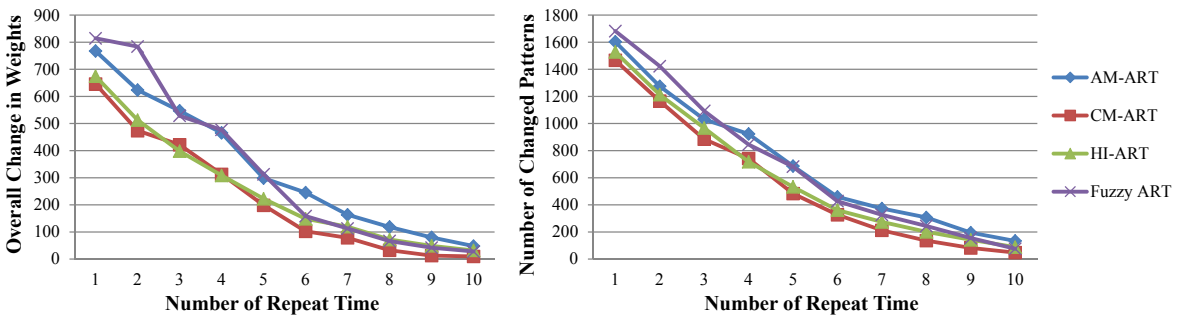
Figure 6.7: Distribution of clusters generated by AM-ART, CM-ART, HI-ART and Fuzzy ART on NUS-WIDE data set in terms of cluster sizes and average pattern-centroid distance under $\rho = 0.2$ (shown in (a) and (b)) and $\rho = 0.9$ (shown in (c) and (d)).

ranging from 3 to 4. The results in Fig. 6.7(a) indicated that AM-ART, CM-ART and HI-ART may generate more clusters with smaller intra-cluster scatters than Fuzzy ART with a small vigilance value; these results are consistent with those shown in Fig. 6.6a.

Under $\rho = 0.9$, as shown in Fig. 6.7(c), a large number of small clusters were generated. Different from Fig. 6.7(a), most of the generated clusters had around 100 patterns. The key difference between the four algorithms was the number of generated clusters with less than 100 patterns. Fuzzy ART and CM-ART identified a large number of small clusters with less than 100 patterns, while AM-ART and HI-ART identified significantly fewer small clusters. This result demonstrates the effectiveness of AMR in simplifying the cluster network when the vigilance value is high. Fig. 6.7(d) illustrates that the four algorithms had similar distributions of pattern-centroid distances of clusters. Note that AM-ART identified fewer clusters with distances around 3 than the other three algorithms, which occurred because AM-ART identified fewer clusters than the other algorithms.

6.5.1.3 Convergence Analysis

This section presents an evaluation of the convergence property of AM-ART, CM-ART, HI-ART and Fuzzy ART on the NUS-WIDE data set in terms of the overall change in weights and the number of patterns jumping across clusters with respect to the repeat presentation of patterns. In the experiments, we followed the parameter settings as used in the previous section and set the vigilance value to $\rho = 0.7$, at which setting all of



(a) (b)

Figure 6.8: Convergence analysis of AM-ART, CM-ART, HI-ART and Fuzzy ART on NUS-WIDE data set in terms of (a) the overall change in weight values and (b) the number of changed patterns through the repeat presentation of patterns.

the algorithms achieved satisfactory performance. Fig. 6.8(a) illustrates that, during the first five rounds of pattern presentations, all of four algorithms experienced large weight changes. This circumstance likely is due to the generation of new clusters. After the sixth round of presentations, only minimal changes occurred in the cluster structure caused by the subsequent repeat presentation of patterns. Comparing the results between the algorithms reveals that, despite the fluctuation at the beginning, AM-ART achieved a convergence speed similar to Fuzzy ART before the fifth round of presentations but slowed after that. In contrast, CM-ART demonstrated the fastest convergence speed among all the algorithms. HI-ART achieved a convergence speed close to that of CM-ART before the sixth round of presentations. However, during the subsequent rounds, its convergence speed became close to that of Fuzzy ART. Fig. 6.8(b) presents a similar situation.

The above discussion indicates that AM-ART usually converges slower than Fuzzy ART. This may be due to the fact that AMR increases the vigilance value of competitive winner candidates and decreases that of the winner so that patterns may jump across those winner candidates when they are presented multiple times. In contrast, CMR may promote the shrinking of the VRs of neighboring clusters by reducing their overlap. Therefore, CM-ART converges faster than Fuzzy ART. Additionally, because HIR incorporates both AMR and CMR, HI-ART converges faster than Fuzzy ART during the first five rounds of presentations due to CMR but achieves a convergence speed similar to that of Fuzzy ART after the network becomes stable due to AMR.

6.5.1.4 Clustering Performance Comparison

This section presents an evaluation of the clustering performance of AM-ART, CM-ART and HI-ART, and compares them to existing clustering approaches that may automatically identify the number of clusters in data, including the base model Fuzzy ART, Affinity Propagation and DBSCAN. All of the algorithms were implemented in Matlab. DBSCAN was an in-house implementation using Euclidean Distance and Affinity Propagation was downloaded from the website of Frey Lab[1]. Hierarchical and genetic clustering approaches were not considered here because they require heavy computation and are not scalable to large-scale data sets.

[1]http://genes.toronto.edu/index.php?q=affinity%20propagation

In the experiments, we applied min-max normalization to the data set because ART-based algorithms require the input values to be in [0,1]. Experimental results indicated that the normalization of data has a minor effect on the performance of DBSCAN and Affinity Propagation. Therefore, to make a fair comparison, we used the normalized data in our experiments for all of the algorithms. Affinity Propagation requires a pairwise similarity matrix as input, so we calculated the distance between data patterns using Euclidean Distance.

Regarding parameter selection, DBSCAN requires empirical setting of the search radius ε and the minimum number of neighbors $minPts$ to determine a region's degree of density. Currently, no accepted method exists for determining a suitable value of $minPts$. However, given that $minPts$ indicates the minimum cluster size, a suitable value of $minPts$ may be determined by analyzing the sizes of compact clusters. Considering that Fuzzy ART may restrict the intra-cluster similarity using vigilance parameter ρ, we computed the average Sum-of-Squared Error (SSE) of clusters generated under different vigilance values $\rho \in [0.8, 0.9]$, and identified the vigilance value that causes a "bend" in the plot. We then arrived at a range of $minPts$ by evaluating the sizes of the small clusters. After determining $minPts$, ε can be identified, as suggested by the authors in [13], by plotting the k-distance graph (k is the value of $minPts$) and choosing the "bend" value. Therefore, a suitable value of $minPts$ should be able to produce a k-distance graph with a clear "bend" and few patterns as noise. Affinity Propagation requires the settings of a preference vector p as the priori suitability of patterns to serve as exemplars, a dumping parameter $dampfact$ to prevent oscillations, and two parameters, $convits$ and $maxits$, to control the convergence speed. The range of the preference value p can be calculated using the Matlab function "preferenceRange.m" which may also be downloaded from the Frey Lab website. As suggested by the authors, all of the patterns share the same preference value. The values of $dampfact$, $convits$ and $maxits$ were first set to the suggested values and then changed with respect to the preference value p to ensure convergence.

We used three external clustering performance measures, including purity [156], class entropy [150] and the Rand index [157]. Purity evaluates the precision, and a higher value indicates better performance in preventing the mis-categorization of patterns. Class

Table 6.1: Clustering results of Affinity Propagation (AP), DBSCAN, Fuzzy ART, AM-ART, CM-ART and HI-ART on NUS-WIDE data set at their best parameter settings in terms of number of clusters, purity, class entropy and Rand index.

	AP	DBSCAN	Fuzzy ART	AM-ART	CM-ART	HI-ART
Number of Clusters	32	28-29	28-31	27-29	30-31	30-32
Purity	0.6827	0.6598 ± 0.015	0.7264 ± 0.026	0.7313 ± 0.023	0.7436 ± 0.023	0.7348 ± 0.025
Class Entropy	0.7163	0.7188 ± 0.011	0.7287 ± 0.024	0.7248 ± 0.022	0.7266 ± 0.026	0.7259 ± 0.021
Rand Index	0.8084	0.7970 ± 0.012	0.8305 ± 0.027	0.8244 ± 0.019	0.8461 ± 0.026	0.8419 ± 0.023

entropy evaluates the recall, and a lower value indicates better performance in grouping patterns of the same class into a smaller number of clusters. The Rand index considers both aspects by rewarding the correct recognition of patterns of the same class and penalizing the partitioning of patterns of the same class to different clusters. The internal performance measures, such as the SSE, were not used because they make strong assumptions based on the shape of clusters. As DBSCAN identifies clusters with arbitrary shapes, such performance measures are not qualified to evaluate the cluster quality of DBSCAN.

The performance of Affinity Propagation, DBSCAN, Fuzzy ART, AM-ART, CM-ART and HI-ART are reported at their best parameter settings obtained in the experiments, namely, $p = 3.5$, $dampfact = 0.9$, $convits = 100$ and $maxits = 500$ for Affinity Propagation; $minPts = 38$ and $\varepsilon = 3.2$ for DBSCAN; $\alpha = 0.01$, $\beta = 0.6$ and $\rho = 0.7$ for Fuzzy ART, AM-ART, CM-ART and HI-ART; and $\sigma = 0.1$ for AM-ART and HI-ART. Additionally, for DBSCAN and the four ART-based algorithms whose results may be affected by the input data sequence, we repeated the experiments ten times and reported the means and standard derivations. We report the performance of Affinity Propagation on a single run.

As shown in Table 6.1, under the best parameter settings, all of the algorithms identify near 30 clusters. Regarding purity, the four ART-based algorithms achieved significantly better performance than Affinity Propagation and DBSCAN at the significant level $p = 0.001$ by t-test. However, their performances were not significantly different at the significant level $p = 0.1$. Regarding class entropy, Affinity Propagation performed best. The performance of DBSCAN did not differ significantly from that of Affinity Propagation at the significant level $p = 0.1$. AM-ART, CM-ART and HI-ART achieved comparable performances, which were better than Fuzzy ART but worse than

DBSCAN. However, the ART-based algorithms did not perform significantly differently than DBSCAN or Affinity Propagation at the significant level $p = 0.1$. Regarding the Rand index, the ART-based algorithms outperformed Affinity Propagation and DBSCAN. Specifically, AM-ART performed worse than Fuzzy ART, CM-ART and HI-ART, but better than Affinity Propagation and DBSCAN at the significant levels $p = 0.05$ and $p = 0.01$ respectively. The performance of Fuzzy ART was not significantly different from that of AM-ART at the significant level $p = 0.1$, and CM-ART and HI-ART performed significantly better than CM-ART at the significant levels $p = 0.05$ and $p = 0.1$ respectively.

These results indicate that, compared with Affinity Propagation and DBSCAN, the ART-based algorithms have the advantage of producing clusters with higher precision. Although they performed slightly worse in terms of recall, the results of the Rand index demonstrate their effectiveness in terms of overall evaluation.

6.5.1.5 Time Cost Analysis

This section presents an evaluation of the time cost of AM-ART, CM-ART, HI-ART, Fuzzy ART, Affinity Propagation and DBSCAN. We first divided the patterns of the nine classes into 10 subsets, each of which contained 120 patterns. Subsequently, we constructed 10 data sets, each of which contained one subset from each class and had 1080 patterns. Therefore, bias was avoided because each data set contained an equal

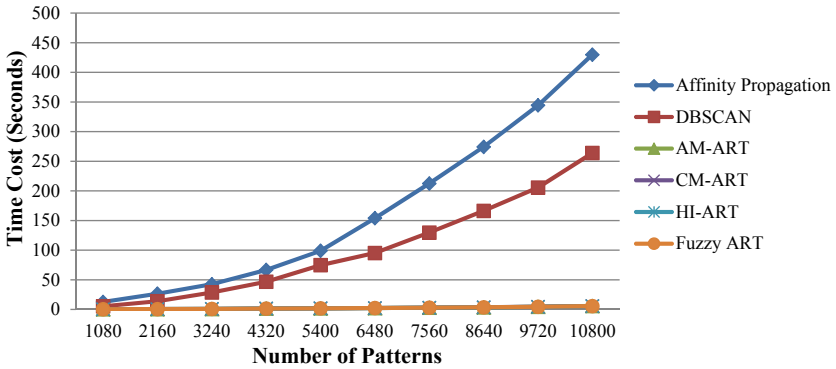


Figure 6.9: Time cost of AM-ART, CM-ART, HI-ART and Fuzzy ART on NUS-WIDE data set in terms of (a) the overall change in weight values and (b) the number of changed patterns through the repeat presentation of patterns.

number of patterns from all of the classes. Finally, we tested the time cost of each clustering algorithm on the data set, which contained one data set at the first time point and 10 at the last time point. To ensure a fair comparison, we followed the parameter settings of each algorithm used in the previous section but tuned them slightly to force them to generate the same number of clusters. All of the algorithms were run on a 3.40GHz Intel(R) Core(TM) i7-4770 CPU with 16GB RAM.

Fig. 6.9 illustrates that, compared with Affinity Propagation and DBSCAN, the time cost of the four ART-based algorithms increased slightly as the number of input patterns increased. AM-ART, CM-ART, HI-ART and Fuzzy ART were able to cluster 10800 patterns in 6 seconds. Moreover, the largest difference in their time cost was only less than 0.2 seconds, which demonstrates that the incorporation of AMR, CMR and HIR into Fuzzy ART incurs little computation. As analyzed in [75], the time complexity of Fuzzy ART is $O(n)$ due to the real-time searching and matching functions for pattern recognition and the one-pass learning mechanism.

6.5.2 20 Newsgroups Data Set

The 20 Newsgroups data set [153] is a public data set consisting of approximately 20,000 messages from 20 different netnews newsgroups, each of which contains nearly 1,000 documents. It has been used widely for experiments on text clustering techniques.

We directly collected 10 classes from the processed Matlab version of the 20newsbydate data set[2], including alt.atheism, comp.graphics, comp.windows.x, rec.sport.baseball, rec.sport.hockey, sci.med, sci.space, misc.forsale, talk.politics.guns and talk.politics.misc. The original data set is divided into training and testing subsets. In our experiments, we combined the patterns belonging to the above 10 classes in both subsets, for a total number of 9,357 patterns. Besides, we filtered any words that occurred less than 30 times. Therefore, each pattern was represented by a bag-of-words vector of 6823 features, weighted by the term frequency-inverse document frequency (tf-idf) algorithm.

[2] `http://qwone.com/~jason/20Newsgroups/`

6.5.2.1 Robustness to Vigilance Parameter

Similar to the experiments on the NUS-WIDE data set, we first evaluated the robustness of AM-ART, CM-ART, HI-ART and Fuzzy ART to the vigilance parameter. The results generated using the parameter settings and evaluation measures from Section 6.5.1.1 appear in Fig. 6.10.

In terms of cluster quality, Fig. 6.10(a) illustrates that AM-ART, CM-ART and HI-ART usually performed better than Fuzzy ART, especially when $\rho < 0.4$. When $\rho > 0.7$, all four algorithms performed comparably. Fig. 6.10(b) illustrates that AM-ART, CM-ART and HI-ART identified more clusters than Fuzzy ART when $\rho < 0.4$. Additionally, AM-ART and HI-ART generate significantly fewer clusters than CM-ART and Fuzzy ART when $\rho > 0.8$. Compared with the results on the NUS-WIDE data set presented in Fig. 6.6, all of the algorithms performed better in terms of purity on the 20 Newsgroups data set. This may have occurred because they identified more clusters on the latter data set. Moreover, the number of clusters identified in the 20 Newsgroups data set increased greatly when $\rho > 0.6$, while in the NUS-WIDE data set, such an increase did not occur untill $\rho > 0.7$. The may have been due to the high dimensionality of the data, which caused the patterns to be less compact in the feature space. This may also have been why all of the algorithms identified more clusters on this data set.

The results from the NUS-WIDE and 20 Newsgroups data sets (Fig. 6.6 and Fig. 6.10 respectively) reveal that AM-ART, CM-ART and HI-ART consistently performed better than or comparable to Fuzzy ART, and identified more clusters when the vigilance value was small. Besides, AM-ART and HI-ART effectively alleviate the over-generation of clusters when the vigilance value was large. Those findings confirm our hypothesis presented in Section 6.5.1.1 and indicate the feasibility of combining multiple rules to improve the performance of Fuzzy ART.

6.5.2.2 Convergence Analysis

This section presents the evaluation of the convergence property of AM-ART, CM-ART, HI-ART and Fuzzy ART on the 20 Newsgroups data set in terms of the overall change in weights and the number of patterns jumping across clusters with respect to the repeat presentation of patterns. Similar to the experiments on the NUS-WIDE data set, we

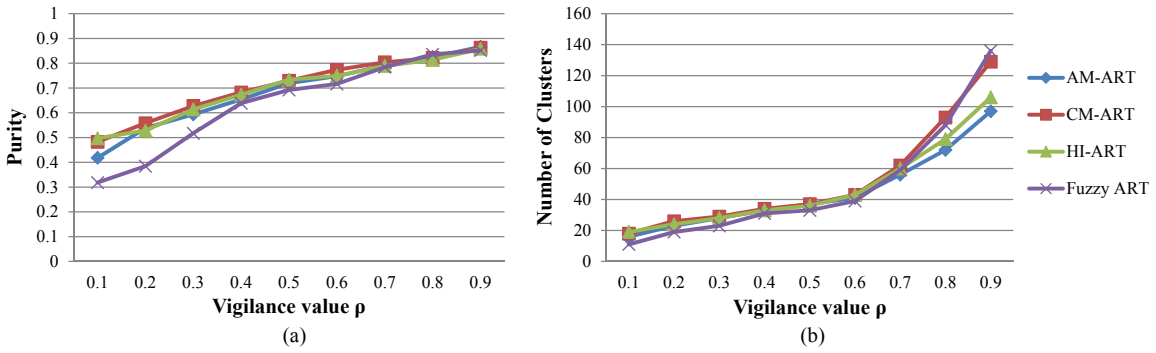


Figure 6.10: Clustering performance of AM-ART, CM-ART, HI-ART and Fuzzy ART under different vigilance values in terms of (a) cluster quality measured by Purity and (b) number of generated clusters on 20 Newsgroups data set.

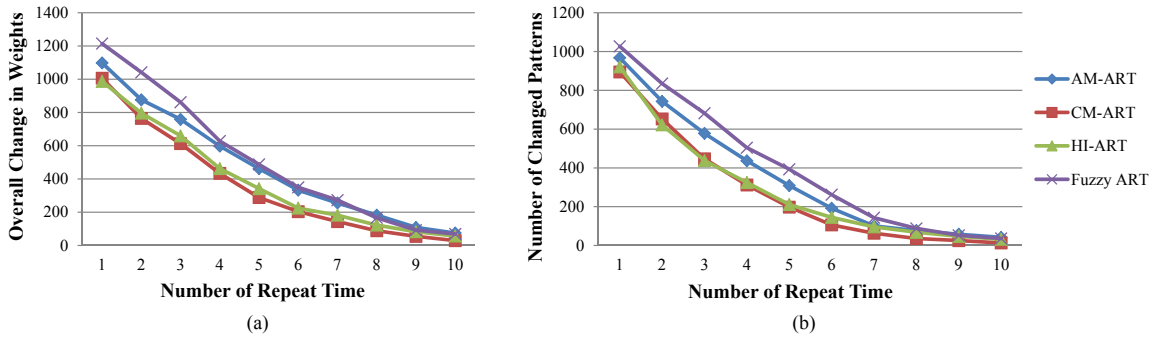


Figure 6.11: The convergence analysis of AM-ART, CM-ART, HI-ART and Fuzzy ART on 20 Newsgroups data set in terms of (a) the overall change in weight values and (b) the number of changed patterns through the repeat presentation of patterns.

followed the parameter settings used in Section 6.5.1.3 and set the vigilance value to $\rho = 0.6$.

Fig. 6.11 illustrates that all of the algorithms changed slightly after 10 rounds of repeat pattern presentations in terms of both the overall change in the weight values and the number of patterns jumping across clusters. Particularly, CM-ART and HI-ART, similar to their performance on the NUS-WIDE data set, obtained comparable convergence speeds, which were faster than AM-ART and Fuzzy ART. In contrast to its performance on the NUS-WIDE data set, AM-ART converged faster than Fuzzy ART at the beginning but then slowed, becoming the same as or even a bit slower than Fuzzy ART after several rounds of presentations. This may have been due to the larger dispersion of patterns in the feature space, which caused the increase in the size of the VRs to have

Table 6.2: Clustering results of Affinity Propagation (AP), DBSCAN, Fuzzy ART, AM-ART, CM-ART and HI-ART on 20 Newsgroups data set at their best parameter settings in terms of number of clusters, purity, class entropy and Rand index.

	AP	DBSCAN	Fuzzzy ART	AM-ART	CM-ART	HI-ART
Number of Clusters	41	39-40	37-40	40-42	42-44	40-43
Purity	0.7225	0.7084 ± 0.017	0.7165 ± 0.027	0.7476 ± 0.022	0.7735 ± 0.019	0.7491 ± 0.024
Class Entropy	0.5779	0.5604 ± 0.016	0.5679 ± 0.027	0.5873 ± 0.021	0.6081 ± 0.024	0.5936 ± 0.026
Rand Index	0.8522	0.8303 ± 0.013	0.8527 ± 0.021	0.8745 ± 0.023	0.8918 ± 0.018	0.8794 ± 0.024

less of an effect on the class assignments of patterns.

6.5.2.3 Clustering Performance Comparison

Similar to the experiments on the NUS-WIDE data set in Section 6.5.1.4, we evaluated the cluster performance of AM-ART, CM-ART, HI-ART, Fuzzy ART, Affinity Propagation and DBSCAN on the 20 Newsgroups data set in terms of purity, class entropy and the Rand index.

Using the same parameter selection methods, we studied the performances of the six clustering algorithms and identified their best parameter settings, which were $p = 8$, $dampfact = 0.9$, $convits = 100$ and $maxits = 1000$ for Affinity Propagation; $minPts = 40$ and $\varepsilon = 5$ for DBSCAN; $\alpha = 0.01$, $\beta = 0.6$ and $\rho = 0.6$ for Fuzzy ART, AM-ART, CM-ART and HI-ART; and $\sigma = 0.1$ for AM-ART and HI-ART.

As shown in Table 6.2, under the best parameter settings, all six algorithms identifed around 40 clusters each. CM-ART obtained the best performance in terms of purity, significantly better than the second and third best performance achieved by HI-ART and AM-ART respectively at the significant level $p = 0.05$. Additionally, Affinity Propagation performed comparably to Fuzzy ART, which was, however, significantly better than DBSCAN at $p = 0.05$. Regarding class entropy, Fuzzy ART and DBSCAN performed best and were not significantly different from each other at $p = 0.1$. Affinity Propagation obtained the second best performance, which was not significantly better than that of AM-ART at $p = 0.1$. Besides, both AM-ART and CM-ART obtained comparable performance to that of HI-ART. However, AM-ART performed significantly better than CM-ART at $p = 0.1$. Regarding the Rand index, CM-ART obtained the best performance, which was, however, not significantly different from that of HI-ART. AM-ART

and HI-ART obtained comparable performances, which were significantly better than that of Fuzzy ART at $p = 0.05$.

The performance comparison presented in both Table 6.1 and Table 6.2 reveals that AM-ART, CM-ART and HI-ART usually performed better than or comparable to Fuzzy ART, Affinity Propagation and DBSCAN in terms of purity and the Rand index, and also performed reasonable in terms of class entropy. This demonstrates the effectiveness of the proposed vigilance adaptation rules in improving the performance of Fuzzy ART and also the feasibility of using ART-based algorithms to cluster large-scale data sets.

Chapter 7

Conclusion and Future Work

7.1 Summary of Contributions

In this thesis, we aim to investigate and develop novel clustering algorithms for identifying natural groups of social media data at semantic level by exploiting their rich and interrelated but noisy and heterogenous meta-information as well as user preferences. Working towards this goal, we have completed four research tasks, summarized as follows:

(i) As a starting point, we focus on the task of web image organization using the textual features extracted from the surrounding text of images. To achieve this goal, we develop a two-step semi-supervised hierarchical clustering algorithm, Personalized Hierarchical Theme Based Clustering (PHTC), for generating the systematical hierarchical view of the web images.

The contributions of this work are two-fold. First, we propose a semi-supervised clustering algorithm, Probabilistic Fusion ART (PF-ART), to group the web images having similar semantics together and identify the key tags in each group at the same time. Taking into consideration of the feature weighting problem of short text, PF-ART does not directly learn the semantic of clusters from the feature distribution of images, but models the semantics of clusters using the probabilistic distribution of tag occurrences. Additionally, PF-ART is able to receive user-provided group label constraint and annotations as user preferences to guide the clustering process.

Secondly, we propose an agglomerative algorithm to associate the clusters identified by PF-ART, by utilizing the key tags and user annotations in the clusters. Different from traditional merging strategies, the proposed algorithm can identify whether the selected pairs of clusters should be merged into one cluster or one of them can be a child of the other one, by measuring the scatters of their children nodes.

In Our experiments, we observe that PF-ART usually outperforms several state-of-the-art clustering algorithms in terms of average precision, F-score, cluster entropy and class entropy. The proposed two-step hierarchical clustering algorithm PHTC generates the hierarchy of data with a higher quality and more systematical structure than existing hierarchical clustering algorithms. It is worth to mention that, as PHTC builds the hierarchy based on clusters, PHTC requires much lower time cost than traditional clustering algorithms that treat individual data objects as hierarchy leaves.

(ii) Subsequently, we investigate the problem of clustering several types of interconnected multimedia data simultaneously, called heterogeneous data co-clustering. Those multimedia data should be the different descriptions, called meta-information, of the same data object, such as the visual content and surrounding text of web images, the tags and text content of web articles, and the different types of social links among social users.

Existing heterogeneous data co-clustering approaches usually suffer from the problems of heavy computation, sensitivity to noisy information and employing equal or empirical weights for different types of features. To address these issues, we propose a Generalized Heterogeneous Fusion ART (GHF-ART) for handling composite multimedia data objects with an arbitrarily rich level of meta-information. GHF-ART has multiple input feature channels for independently receiving and processing each type of features, and employs different representation and learning methods for commonly used multimedia data, including images, articles, and short text. More importantly, to achieve an effective fusion of the similarities obtained

from each feature channel, GHF-ART incorporates a weighting function for adaptively evaluating the importance of different types of features on representing the patterns in the same cluster during the clustering process.

Experiments on two web image sets and one netnews data set have shown that GHF-ART achieves superior performance than many state-of-the-art approaches. Through a case study, we observe that, without the weighting algorithm, the performance of GHF-ART has an obvious decrease, but it still obtains the best results in most cases. If we further apply the learning function of ART for modelling the features of meta-information, GHF-ART degenerates to Fusion ART. Under this circumstance, GHF-ART achieves the performance similar to other algorithms. These findings demonstrated the effectiveness of the proposed weighting algorithm and learning function for tackling short and noisy meta-information.

(iii) We further explore the feasibility of GHF-ART for discovering user communities in the heterogeneous social networks, in which users are connected by multiple and different types of links. Compared with the web multimedia data, the social network data usually appear in a much larger size and involve more types of information, especially the links of user interactions. Therefore, beside the problems faced in multimedia data co-clustering, existing approaches also meet the problem of requiring the number of cluster a priori.

Based on the above consideration, GHF-ART has several advantages over existing approaches. First, GHF-ART performs online and one-pass learning which guarantees its low computational cost. Secondly, GHF-ART does not need the number of clusters a priori. Thirdly, GHF-ART allows different feature representation and learning functions for different types of links, of which the output similarities are effectively fused by the weighting function. Fourthly, GHF-ART considers both the overall similarity and individual similarity in each feature channel. Therefore, users in the same cluster should have consistently similar behavior in all types of links. For clustering data patterns of social networks, we develop a set of specific feature representation and learning rules for GHF-ART to handle various heterogeneous types of social links, including relational links, textual links in articles and textual links in short text.

We analyze the performance of GHF-ART on two social network data sets in terms of parameter selection, clustering performance comparison, effectiveness of the weighting function and time cost. From the experiment results, we find that the performance of GHF-ART is only sensitive to the vigilance parameter, which controls the intra-cluster similarity. In addition, we experimentally show that a suitable value of the vigilance parameter could be selected by tuning the vigilance parameter until a small number of small clusters are generated. We also demonstrate the effectiveness of GHF-ART on clustering heterogeneous social network data by comparing the performance of GHF-ART with existing algorithms, and evaluate its capability of discovering the key features of clusters and correlation analysis across heterogeneous links using case studies.

(iv) The social media data are typically big, complex and noisy. As such, the parameter selection problem is a challenge for existing clustering algorithms. Compared with other clustering algorithms, ART has the advantage of dependence on a single parameter, the vigilance parameter, to identify clusters. Therefore, we study the task of improving the robustness of ART to the vigilance parameter.

To this end, we first theoretically analyzed the effect of complement coding on the Fuzzy ART clustering mechanism. Secondly, we offered a geometric interpretation of the cluster mechanism of Fuzzy ART using the identified vigilance region (VR), as calculated from the vigilance criteria. Thirdly, we proposed three adaptation rules, namely, the Activation Maximization Rule (AMR), the Confliction Minimization Rule (CMR) and the Hybrid Integration Rule (HIR), for the vigilance parameter so that the clusters in the Fuzzy ART system would have individual vigilance levels and be able to adaptively tune their VR boundaries during the clustering process.

Experiments on two social media data sets have demonstrated that by incorporating AMR, CMR, and HIR into Fuzzy ART, the resulting AM-ART, CM-ART and HI-ART usually performed better than or similar as Fuzzy ART and several existing clustering algorithms that do not require a pre-defined number of clusters. Through the experimental results, we find that AM-ART, CM-ART and HI-ART are more robust to the initial vigilance value than Fuzzy ART. Particularly, AM-ART significantly reduces the number of small clusters when the vigilance value

is large. In contrast, CM-ART performs much better than Fuzzy ART when the vigilance value was small. HI-ART, which incorporates the ideas from both AMR and CMR, takes advantages of both AM-ART and CM-ART.

7.2 Future Work

This thesis, so far, has completed four studies, including the tag-based hierarchical clustering for personalized web image organization, the semi-supervised heterogeneous data co-clustering for web multimedia data co-clustering, the application of the proposed heterogeneous data co-clustering algorithm to social network data for discovering user communities in heterogeneous social networks, and the adaptive tuning of vigilance parameter in Fuzzy ART for adaptive scaling of cluster boundaries in the feature space in order to make Fuzzy ART insensitive to the input parameters for clustering big and complex social media data.

However, there is still plenty of room for the improvement of the developed clustering techniques. Also, those studies provide the basis for the clustering of heterogeneous social media data with user preferences and other associative social media mining tasks. Therefore, the future work of this thesis will focus on the following aspects:

(i) **Modeling of Short and Noisy Text** Some types of social media data, such as tweets, the surrounding text of web documents, user tagging and comments, are short, noisy but informative, so extracting meaningful key tags from those types of data is important for the subsequent clustering tasks.

The key problem in this task is the insufficient statistical information for distinguishing key tags by evaluating their frequencies. In our proposed GHF-ART (Chapter 4), we address this problem by building the features of the meta-information using the presence of tags, and modeling the cluster prototype using probability distribution of tag occurrences. However, one problem of this method is that the noisy tags have equal weights as the key tags. Consequently, a large number of matching incurred by noisy tags in a cluster may lead to the ill-represented prototype of the cluster.

Therefore, tag ranking methods can be employed in the textual feature construction stage to filter noisy tags or give higher weights to key tags so as to further depress the effect of noisy tags.

(ii) **Automated Selection of Vigilance Parameter in Fuzzy ART:** In Chapter 6, we propose three methods, the AMR, the CMR and the HIR, for adapting the value of the vigilance parameter in Fuzzy ART. However, the proposed rules still require an initial vigilance value. Therefore, a future direction of this study is to develop techniques for choosing a suitable initial vigilance value or making the selection of vigilance value fully automated.

(iii) **Improvement of Clustering Mechanism of ART:** In Chapter 6, we offer the geometrical interpretation of the vigilance region (VR) and Fuzzy ART. Therefore, further studies can be conducted to improve the clustering mechanism of Fuzzy ART. For example, integrating the choice and match procedures into a unified step, and improving the shape of the VR to produce better cluster boundaries.

(iv) **Multimedia Data Indexing, Annotation and Retrieval:** In recent years, there has been a growing interest in exploiting the surrounding text and visual content of images to perform image annotation and retrieval [162, 163]. Given the query keywords and/or an query image, the retrieval system returns relevant images or annotations according to users options.

Our proposed Generalized Heterogeneous Fusion ART (GHF-ART) provides a basis for this task. In view that GHF-ART generates clusters by identifying the key features of data, the weights of clusters associated to multi-modal features are the natural indexes of the within-cluster images. By incorporating suitable similarity measures and a ranking process, a multi-modal image retrieval system, based on GHF-ART, can be implemented. One advantage of our method over other image indexing techniques should be the incremental learning capability that enables the image retrieval system to have a real-time response to the query image without the time-consuming re-clustering process. Besides, as GHF-ART is able to incorporate constraints for all of the feature channels rather than just an overall constraint, users are more likely to have the interesting results.

(v) **Exploiting Temporal Factor for Multimedia Data Storage and Mining:** Temporal databases, built by extracting relational facts from web and text resources with the corresponding time stamps, is an important topic in the field of data mining [164]. By incorporating temporal factor, the database is able to interpret periodic events within their respective timespans, such as performance of a soccer team and positions held by a person.

Time is also an important factor for automatic multimedia data organization and retrieval [165, 166]. This is because users often organize their videos and photos in terms of "events", such as a trip or a party. However, these multimedia data associated with the same event can have little visual similarity. For example, considering a birthday party, the photos could have widely different subjects such as the scenery of venue, the cake, the people and the dinner. It poses a great challenge to associate the visual features to the concepts, a problem known as semantic gap. On the other hand, photos from the same event are usually taken over a relatively short period. Thus, incorporating time factor will help to identify the photos belonging to the same event. Based on the above considerations, we plan to incorporate the temporal factor into our clustering algorithms, such as GHF-ART. In this way, we can investigate the feasibility of enhancing the clustering performance as well as building a temporal semantic database for image organization.

(vi) **Applications to Social Media Mining:** As illustrated in Chapter 2, there has been various emergent tasks in social media mining that employ clustering techniques as a solution. Therefore, one direction of the future work is to identify up-to-date problems in social media mining and extend the techniques proposed in this thesis to address them. For example, we have demonstrated the feasibility of GHF-ART to the problem of discovering user communities in heterogenous social networks. Therefore, a further study could be the mining of the correlation among different types of social links and their respective characteristics. Besides, in view that GHF-ART is an online clustering algorithm, it is natural to extend it to perform online learning tasks, such as the real-time detection of social events on Twitter.

List of Author's Publications

- Lei Meng and Ah-Hwee Tan, "Semi-supervised Hierarchical Clustering for Personalized Web Image Organization," *In proceedings of International Joint Conference on Neural Networks*, pp. 251-258, 2012.

- Lei Meng, Ah-Hwee Tan, and Dong Xu, "Semi-Supervised Heterogeneous Fusion for Multimedia Data Co-clustering," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 26, no. 9, pp.2293-2306, 2014.

- Lei Meng, Ah-Hwee Tan, and Donald C. Wunsch II, "Vigilance Adaptation in Adaptive Resonance Theory," *In proceedings of International Joint Conference on Neural Networks*, 2013.

- Lei Meng and Ah-Hwee Tan, "Community Discovery in Social Networks via Heterogeneous Link Association and Fusion," *In proceedings of SIAM International Conference on Data Mining (SDM)*, pp. 803-811, 2014.

- Lei Meng, Ah-Hwee Tan, and Donald C. Wunsch II, "Adaptive Scaling of Cluster Boundaries for Large-Scale Social Media Data Clustering," *In submission to IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*.

References

[1] L. Meng and A.-H. Tan, "Community discovery in social networks via heterogeneous link association and fusion," *SIAM International Conference on Data Mining (SDM)*, pp. 803–811, 2014.

[2] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, , and Y. Zheng, "NUS-WIDE: A real-world web image database from national university of singapore," *In proc. CIVR*, pp. 1–9, 2009.

[3] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," *In Proc. ACM CIKM*, pp. 919–928, 2009.

[5] D. Liu, X. Hua, L. Yang, M. Wang, and H. Zhang, "Tag ranking," *In Proc. WWW*, pp. 351–360, 2009.

[6] X. Li, C. G. M. Snoek, and M. Worring, "Learning tag relevance by neighbor voting for social image retrieval," *Proc. of ACM Multimedia*, pp. 180–187, 2008.

[7] J. J. Whang, X. Sui, Y. Sun, and I. S. Dhillon, "Scalable and memory-efficient clustering of large-scale social networks," *In ICDM*, pp. 705–714, 2012.

[8] S. Bickel and T. Scheffer, "Multi-view clustering," *In ICDM*, pp. 19–26, 2004.

[9] D. Zhou and C. J. C. Burges, "Spectral clustering and transductive learning with multiple views," *In ICML*, pp. 1159–1166, 2007.

[10] Y. Chen, L. Wang, and M. Dong, "Non-negative matrix factorization for semisupervised heterogeneous data coclustering," *In TKDE*, vol. 22, no. 10, pp. 1459–1474, 2010.

[11] G. Bisson and C. Grimal, "Co-clustering of multi-view datasets: A parallelizable approach," *In ICDM*, pp. 828–833, 2012.

[12] R. Bekkerman and J. Jeon, "Multi-modal clustering for multimedia collections," *In CVPR*, pp. 1–8, 2007.

[13] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," *In KDD*, pp. 226–231, 1996.

[14] L. Wang, C. Leckie, K. Ramamohanarao, and J. Bezdek, "Automatically determining the number of clusters in unlabeled data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 3, pp. 335–350, 2012.

[15] W. Wang and Y. Zhang, "On fuzzy cluster validity indices," *Fuzzy Sets and Systems*, vol. 158, no. 19, pp. 2095–2117, 2007.

[16] Y. Chen, M. Rege, M. Dong, and J. Hua, "Incorporating user provided constraints into document clustering," *In ICDM*, pp. 103–112, 2007.

[17] Y. Chen, M. Dong, and W. Wan, "Image co-clustering with multi-modality features and user feedbacks," *In ACM Multimedia*, pp. 689–692, 2007.

[18] X. Shi, W. Fan, and P. S. Yu, "Efficient semi-supervised spectral co-clustering with constraints," *In ICDM*, pp. 532–541, 2010.

[19] S. Grossberg, "How does a brain build a cognitive code," *Psychological Review*, vol. 87, no. 1, pp. 1–51, 1980.

[20] A.-H. Tan, G. A. Carpenter, and S. Grossberg, "Intelligence through interaction: Towards a unified theory for learning," *In LNCS*, vol. 4491, pp. 1094–1103, 2007.

[21] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] R. Nock and F. Nielsen, "On weighting clustering," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1–13, 2006.

[23] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551–556, 2004.

[24] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1027–1035, 2007.

[25] B. Bahmani, B. Moseley, A. Vattani, R. Kumar, and S. Vassilvitskii, "Scalable k-means++," *Proceedings of the VLDB Endowment*, vol. 5, no. 7, pp. 622–633, 2012.

[26] V. Faber, "Clustering and the continuous k-means algorithm," *Los Alamos Science*, vol. 22, pp. 138–144, 1994.

[27] C. Chen, J. Luo, and K. J. Parker, "Image segmentation via adaptive k-mean clustering and knowledge-based morphological operations with biomedical applications," *IEEE Transactions on Image Processing*, vol. 7, no. 12, pp. 1673–1683, 1998.

[28] D. Cai, X. He, Z. Li, W. Ma, and J. Wen, "Hierarchical clustering of www image search results using visual, textual and link information," *In Proc. ACM Multimedia*, pp. 952–959, 2004.

[29] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[30] M. Rege, M. Dong, and J. Hua, "Graph theoretical framework for simultaneously integrating visual and textual features for efficient web image clustering," *Proc. of Int'l Conference on World Wide Web*, pp. 317–326, 2008.

[31] J. Gower and G. Ross, "Minimum spanning trees and single linkage clustering analysis," *J. R. Stat. Soc. Ser. C*, pp. 595–616, 1969.

[32] H. Schtze and C. Silverstein, "Projections for efficient document clustering," *In proc. SIGIR*, pp. 74–81, 1997.

[33] O. Aichholzer and F. Aurenhammer, "Classifying hyperplanes in hypercubes," *SIAM J. Discrete Math.*, pp. 225–232, 1996.

[34] C. Hsu, J. Caverlee, and E. Khabiri, "Hierarchical comments-based clustering," *In Proc. ACM SAC*, pp. 1130–1137, 2011.

[35] N. Sahoo, J. Callan, R. Krishnan, G. Duncan, and R. Padman, "Incremental hierarchical clustering of text documents," *Proceedings of ACM international conference on Information and knowledge management*, pp. 357–366, 2006.

[36] H. Ding, J. Liu, and H. Lu, "Hierarchical clustering-based navigation of image search results," *In Proc. ACM Multimedia*, pp. 741–744, 2008.

[37] Z. Wu and R. Leahy, "An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1101–1113, 1993.

[38] M. Rege, M. Dong, and F. Fotouhi, "Co-clustering documents and words using bipartite isoperimetric graph partitioning," *In Proc. ICDM*, pp. 532–541, 2006.

[39] G. H. Golub and C. F. V. Loan, "Matrix computations," *Johns Hopkins University Press*, 1996.

[40] G. Qiu, "Clustering and the continuous k-means algorithm," *Proceedings of the International Conference on Pattern Recognition*, pp. 991–994, 2004.

[41] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science*, vol. 41, pp. 391–407, 1990.

[42] J. Steinberger and K. Jezek, "Using latent semantic analysis in text summarization and summary evaluation," *In Proc. ISIM*, pp. 93–100, 2004.

[43] M. G. Ozsoy, I. Cicekli, and F. N. Alpaslan, "Text summarization of turkish texts using latent semantic analysis," *Proceedings of International Conference on Computational Linguistics*, pp. 869–876, 2010.

[44] B. Fasel, F. Monay, and D. Gatica-Perez, "Latent semantic analysis of facial action codes for automatic facial expression recognition," *Proceedings of International Conference on Multimedia Information Retrieval*, pp. 181–188, 2004.

[45] T.-T. Pham, N. E. Maillot, J.-H. Lim, and J.-P. Chevallet, "Latent semantic fusion model for image retrieval and annotation," *Proceedings of International Conference on Information and Knowledge Management*, pp. 439–444, 2007.

[46] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[47] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," *In Proc. SIGIR conference on Research and development in informaion retrieval*, pp. 268–273, 2003.

[48] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix trifactorizations for clustering," *Proc. ACM SIGKDD Intl Conf. Knowledge Discovery and Data Mining*, pp. 126–135, 2006.

[49] Q. Gu and J. Zhou, "Co-clustering on manifolds," *In KDD*, pp. 359–367, 2009.

[50] P. D. McNicholas and T. B. Murphy, "Model-based clustering of microarray expression data via latent gaussian mixture models," *Bioinformatics*, vol. 26, no. 21, pp. 2705–2712, 2010.

[51] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[52] Y. Qin and C. E. Priebe, "Maximum lq-likelihood estimation via the expectation-maximization algorithm: A robust estimation of mixture models," *Journal of the American Statistical Association*, vol. 108, no. 503, pp. 914–928, 2013.

[53] A. Mumtaz, E. Coviello, G. R. G. Lanckriet, and A. B. Chan, "Clustering dynamic textures with the hierarchical EM algorithm for modeling video," *IEEE transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 7, pp. 1606–1621, 2013.

[54] S. Bandyopadhyay and S. Saha, "A point symmetry-based clustering technique for automatic evolution of clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1441–1457, 2008.

[55] S. Bandyopadhyay, "Genetic algorithms for clustering and fuzzy clustering," *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 6, pp. 524–531, 2011.

[56] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 169–194, 1998.

[57] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD International Conference on Management of Data*, pp. 49–60, 1999.

[58] T. Pei, A. Jasra, D. Hand, A.-X. Zhu, and C. Zhou, "DECODE: A new method for discovering clusters of different densities in spatial data," *Data Mining and Knowledge Discovery*, vol. 18, no. 3, pp. 337–369, 2009.

[59] T. N. Tran, R. Wehrens, and L. M. C. Buydens, "KNN-kernel density-based clustering for high-dimensional multivariate data," *Computational Statistics & Data Analysis*, vol. 51, no. 2, pp. 513–525, 2006.

[60] H.-P. Kriegel, P. Kroger, J. Sander, and A. Zimek, "Density-based clustering," *WIREs Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[61] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, pp. 972–976, 2007.

[62] C. Z. Y. Jia, J. Wang and X.-S. Hua, "Finding image exemplars using fast sparse affinity propagation," *In ACM MM*, pp. 639–642, 2008.

[63] Y. Fujiwara, G. Irie, and T. Kitahara, "Fast algorithm for affinity propagation," *In IJCAI*, pp. 2238–2243, 2011.

[64] R. Zhao and W. Grosky, "Narrowing the semantic gap improved text-based web document retrieval using visual features," *IEEE Transactions on Multimedia*, pp. 189–200, 2002.

[65] G. A. Carpenter and S. Grossberg, "ART 2: Self-organization of stable category recognition codes for analog input patterns," *Applied Optics*, vol. 26, no. 23, pp. 4919–4930, 1987.

[66] G. A. Carpenter, S. Grossberg, and D. Rosen, "ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition," *Neural Networks*, vol. 4, pp. 493–504, 1991.

[67] G. A. Carpenter and S. Grossberg, "ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures," *Neural Networks*, vol. 3, no. 2, pp. 129–152, 1990.

[68] G. Carpenter, S. Grossberg, and D. Rosen, "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, vol. 4, no. 6, pp. 759–771, 1991.

[69] G. A. Carpenter, S. Grossberg, and J. Reynolds, "ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network," *Neural Networks*, vol. 4, no. 5, pp. 565–588, 1991.

[70] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," *IEEE Transactions on Neural Networks*, vol. 3, no. 5, pp. 698–713, 1992.

[71] L. Massey, "Real-world text clustering with adaptive resonance theory neural networks," *In Proceedings of International Joint Conference on Neural Networks*, pp. 2748–2753, 2005.

[72] A.-H. Tan, H.-L. Ong, H. Pan, J. Ng, and Q. Li, "Towards personalised web intelligence," *Knowl. Inf. Syst.*, vol. 6, no. 5, pp. 595–616, 2004.

[73] L. Meng and A.-H. Tan, "Semi-supervised hierarchical clustering for personalized web image organization," *International Joint Conference on Neural Networks*, pp. 251–258, 2012.

[74] T. Jiang and A.-H. Tan, "Learning image-text associations," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 2, pp. 161–177, 2009.

[75] L. Meng, A.-H. Tan, and D. Xu, "Semi-supervised heterogeneous fusion for multimedia data co-clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2293–2306, 2014.

[76] N. Shental, T. Hertz, D. Weinshall, and M. Pavel, "Adjustment learning and relevant component analysis," *In ECCV*, pp. 776–792, 2002.

[77] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," *In NIPS*, pp. 505–512, 2003.

[78] T. Joachims, "Transductive learning via spectral graph partitioning," *In ICML*, pp. 290–297, 2003.

[79] W. Liu and S. Chang, "Robust multi-class transductive learning with graphs," *In CVPR*, pp. 381–388, 2009.

[80] Z. Fu, H. H. S. Ip, H. Lu, and Z. Lu, "Multi-modal constraint propogation for heterogeneous image clustering," *In MM*, pp. 143–152, 2011.

[81] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: A kernel approach," *Proc. Intl Conf. Machine Learning*, pp. 457–464, 2005.

[82] X. Ji and W. Xu, "Document clustering with prior knowledge," *Proc. Intl ACM SIGIR Conf. Research and Development in Information Retrieval*, pp. 405–412, 2006.

[83] B. Gao, T. Liu, T. Qin, X. Zheng, Q. Cheng, and W. Ma, "Web image clustering by consistent utilization of visual features and surrounding texts," *Proc. of ACM Multimedia*, pp. 112–121, 2005.

[84] B. Gao, T.-Y. Liu, X. Zheng, Q.-S. Cheng, and W.-Y. Ma, "Consistent bipartite graph co-partitioning for star-structured high-order heterogeneous data co-clustering," *Proc. of Int'l Conference on Knowledge Discovery and Data Mining*, pp. 41–50, 2005.

[85] B. Long, X. Wu, Z. Zhang, and P. S. Yu, "Spectral clustering for multi-type relational data," *In ICML*, pp. 585–592, 2006.

[86] X. Cai, F. Nie, H. Huang, and F. Kamangar, "Heterogeneous image feature integration via multi-modal spectral clustering," *In CVPR*, pp. 1977–1984, 2011.

[87] W. Tang, Z. Lu, and I. S. Dhillon, "Clustering with multiple graphs," *In ICDM*, pp. 1016–1021, 2009.

[88] R. Bekkerman, M. Sahami, and E. Learned-Miller, "Combinatorial markov random fields," *In ECML*, pp. 30–41, 2006.

[89] R. Bekkerman and M. Sahami, "Semi-supervised clustering using combinatorial mrfs," *In ICML Workshop on Learning in Structured Output Spaces*, 2006.

[90] R. Bekkerman, M. Scholz, and K. Viswanathan, "Improving clustering stability with combinatorial mrfs," *In KDD*, pp. 99–108, 2009.

[91] I. Drost, S. Bickel, and T. Scheffer, "Discovering communities in linked data by multi-view clustering," *From Data and Information Analysis to Knowledge Engineering*, pp. 342–349, 2006.

[92] K. Chaudhuri, S. M. Kakade, K. Livescu, and K. Sridharan, "Multi-view clustering via canonical correlation analysis," *In ICML*, pp. 129–136, 2009.

[93] A. Kumar and H. D. III, "A co-training approach for multi-view spectral clustering," *In ICML*, pp. 393–400, 2011.

[94] L. Tang, X. Wang, and H. Liu, "Uncovering groups via heterogeneous interaction analysis," *In ICDM*, pp. 503–512, 2009.

[95] L. Nguyen, K. Woon, and A.-H. Tan, "A self-organizing neural model for multimedia information fusion," *International Conference on Information Fusion*, pp. 1–7, 2008.

[96] J. C. Bezdek and R. Hathaway, "VAT: A tool for visual assessment of (cluster) tendency," *In Proc. Intl Joint Conf. Neural Networks*, pp. 2225–2230, 2002.

[97] I. Sledge, J. Huband, and J. C. Bezdek, "(automatic) cluster count extraction from unlabeled datasets," *Fifth Intl Conf. Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 3–13, 2008.

[98] J. Liang, X. Zhao, D. Li, F. Cao, and C. Dang, "Determining the number of clusters using information entropy for mixed data," *Pattern Recognition*, vol. 45, no. 6, pp. 2251–2265, 2012.

[99] C. A. Sugar and G. M. James, "Finding the number of clusters in a data set: An information theoretic approach," *Journal of the American Statistical Association*, vol. 98, no. 463, pp. 750–763, 2003.

[100] R. Kothari and D. Pitts, "On finding the number of clusters," *Pattern Recognition Letters*, vol. 20, no. 4, pp. 405–416, 1999.

[101] J.-S. Lee and S. Olafsson, "A meta-learning approach for determining the number of clusters with consideration of nearest neighbors," *Information Sciences*, vol. 232, pp. 208–224, 2013.

[102] H. Sun, S. Wang, and Q. Jiang, "FCM-based model selection algorithms for determining the number of clusters," *Pattern Recognition*, vol. 37, no. 10, pp. 2027–2037, 2004.

[103] M. J. Li, M. K. Ng, Y. Cheung, and Z. X. Huang, "Agglomerative fuzzy k-means clustering algorithm with selection of number of clusters," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1519–1534, 2008.

[104] Y. Leung, J. S. Zhang, and Z. B. Xu, "Clustering by scale-space filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 12, pp. 1394–1410, 2000.

[105] H. Yan, K. K. Chen, L. Liu, and J. Bae, "Determining the best k for clustering transactional datasets: A coverage density-based approach," *Data & Knowledge Engineering*, vol. 68, no. 1, pp. 28–48, 2009.

[106] F. Jing, C. Wang, Y. Yao, L. Zhang, and W. Ma, "Igroup: web image search results clustering," *In Proc. ACM Multimedia*, pp. 377–384, 2006.

[107] L. Chen, D. Xu, I. W. Tsang, and J. Luo, "Tag-based image retrieval improved by augmented features and group-based refinement," *IEEE Transactions on Multimedia (T-MM)*, pp. 1057–1067, 2012.

[108] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas, "Short text classification in twitter to improve information filtering," *In International ACM SIGIR conference on Research and development in information retrieval*, pp. 841–842, 2010.

[109] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth," *In SDM*, pp. 745–754, 2012.

[110] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: Membership, growth, and evolution," *In KDD*, pp. 44–54, 2006.

[111] S. Papadopoulos, Y. Kompatsiaris, A. Vakali, and P. Spyridonos, "Community detection in social media," *Data Mining and Knowledge Discovery*, vol. 24, no. 3, pp. 515–554, 2012.

[112] B. Liu, "Sentiment analysis and subjectivity," *Handbook of Natural Language Processing, CRC Press*, pp. 627–666, 2010.

[113] G. Paltoglou and M. Thelwall, "Twitter, myspace, digg: Unsupervised sentiment analysis in social media," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 4, pp. 1–19, 2012.

[114] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in twitter: The million follower fallacy," *In Proceedings of International AAAI Conference on Weblogs and Social Media*, pp. 10–17, 2010.

[115] N. Agarwal, H. Liu, L. Tang, and P. S. Yu, "Modeling blogger influence in a community," *Social Network Analysis and Mining*, vol. 2, no. 2, pp. 139–162, 2012.

[116] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, and H. Cao, "Link prediction and recommendation across heterogeneous social networks," *In ICDM*, pp. 181–190, 2012.

[117] Y. Yang, N. Chawla, Y. Sun, and J. Han, "Predicting links in multi-relational and heterogeneous networks," *In ICDM*, pp. 755–764, 2012.

[118] I. Konstas, V. Stathopoulos, and J. M. Jose, "On social networks and collaborative recommendation," *In Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 195–202, 2009.

[119] A. Anderson, D. Huttenlocher, J. Kleinberg, and J. Leskovec, "Discovering value from community activity on focused question answering sites: A case study of stack overow," *In KDD*, pp. 850–858, 2012.

[120] R. Hong, M. Wang, G. Li, L. Nie, Z.-J. Zha, and T.-S. Chua, "Multimedia question answering," *IEEE Transactions on MultiMedia*, vol. 19, no. 4, pp. 72–78, 2012.

[121] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors," *In Proceedings of International Conference on World Wide Web*, pp. 851–860, 2010.

[122] R. Li, K. H. Lei, R. Khadiwala, and K. C.-C. Chang, "Tedas: A twitter-based event detection and analysis system," *International Conference on Data Engineering*, pp. 1273–1276, 2012.

[123] H. Becker, M. Naaman, and L. Gravano, "Beyond trending topics: Real-world event identication on twitter," *In Proceedings of International AAAI Conference on Weblogs and Social Media*, pp. 438–441, 2011.

[124] P. Gundecha and H. Liu, "Mining social media: A brief introduction," *Tutorials in Operations Research*, 2012.

[125] R. H. V. Leuken, L. Garcia, X. Olivares, and R. V. Zwol, "Visual diversication of image search results," *In WWW*, pp. 341–350, 2009.

[126] Q. Chen, G. Wang, and C. L. Tan, "Web image organization and object discovery by actively creating visual clusters through crowdsourcing," *In Proceedings of International Conference on Tools with Artificial Intelligence*, pp. 419–427, 2012.

[127] T. Jiang and A.-H. Tan, "Discovering image-text associations for cross-media web information fusion," *In PKDD*, pp. 561–568, 2006.

[128] S. Harabagiu and F. Lacatusu, "Using topic themes for multi-document summarization," *ACM Transactions on Information Systems*, vol. 28, no. 3, pp. 1–47, 2010.

[129] D. Wang, S. Zhu, T. Li, Y. Chi, and Y. Gong, "Integrating document clustering and multidocument summarization," *ACM Transactions on Knowledge Discovery from Data*, vol. 5, no. 3, pp. 1–26, 2011.

[130] P. Chandrika and C. Jawahar, "Multi modal semantic indexing for image retrieval," *In CIVR*, pp. 342–349, 2010.

[131] A. Messina and M. Montagnuolo, "A generalised cross-modal clustering method applied to multimedia news semantic indexing and retrieval," *In WWW*, pp. 321–330, 2009.

[132] N. Rasiwasia and J. Pereira, "A new approach to cross-modal multimedia retrieval," *In MM*, pp. 251–260, 2010.

[133] M. Li, X.-B. Xue, and Z.-H. Zhou, "Exploiting multi-modal interactions: A unified framework," *In IJCAI*, pp. 1120–1125, 2009.

[134] K. Zhang, D. Lo, E.-P. Lim, and P. K. Prasetyo, "Mining indirect antagonistic communities from social interactions," *Knowledge and Information Systems*, vol. 35, no. 3, pp. 553–583, 2013.

[135] V. Satuluri, S. Parthasarathy, and Y. Ruan, "Local graph sparsification for scalable clustering," *In SIGMOD*, pp. 721–732, 2011.

[136] K. Macropol and A. Singh, "Scalable discovery of best clusters on large graphs," *In VLDB Endowment*, pp. 693–702, 2010.

[137] L. Zhu, A. Galstyan, J. Cheng, and K. Lerman, "Tripartite graph clustering for dynamic sentiment analysis on social media," *In SIGMOD*, pp. 1531–1542, 2014.

[138] X. Hu, J. Tang, H. Gao, and H. Liu, "Unsupervised sentiment analysis with emotional signals," *In WWW*, pp. 607–618, 2013.

[139] L. Chen and A. Roy, "Event detection from flickr data through wavelet-based spatial analysis," *In CIKM*, pp. 523–532, 2009.

[140] S. Papadopoulos, C. Zigkolis, Y. Kompatsiaris, and A. Vakali, "Cluster-based landmark and event detection for tagged photo collections," *In IEEE Multimedia Magazine*, vol. 18, no. 1, pp. 52–63, 2011.

[141] G. Petkos, S. Papadopoulos, and Y. Kompatsiaris, "Social event detection using multimodal clustering and integrating supervisory signals," *In ICMR*, pp. 23:1–23:8, 2012.

[142] C. Kwok, O. Etzioni, and D. S. Weld, "Scaling question answering to the web," *ACM Transactions on Information Systems (TOIS)*, vol. 19, no. 3, pp. 242–262, 2001.

[143] M. J. Blooma, A. Y. K. Chua, and D. H.-L. Goh, "Quadripartite graph-based clustering of questions," *International Conference on Information Technology: New Generations*, pp. 591–596, 2011.

[144] J. Ko, L. Si, and E. Nyberg, "Combining evidence with a probabilistic framework for answer ranking and answer merging in question answering," *Information Processing & Management*, vol. 46, no. 5, pp. 541–554, 2010.

[145] K. W.-T. Leung, W. Ng, and D. L. Lee, "Personalized concept-based clustering of search engine queries," *IEEE Transactions on Knoledge and Data Engineering*, vol. 20, no. 11, pp. 1505–1518, 2008.

[146] Z. Zhang and O. Nasraoui, "Mining search engine query logs for query recommendation," *In WWW*, pp. 1039–1040, 2006.

[147] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. Grossman, and O. Frieder, "Hourly analysis of a very large topically categorized web query log," *In SIGIR*, pp. 321–328, 2004.

[148] T. Pedersen, S. Patwardhan, and J. Michelizzi, "Wordnet::similarity: measuring the relatedness of concepts," *Demonstration papers at HLT-NAACL*, 2004.

[149] R. Cilibrasi and P. M. B. Vitanyi, "The google similarity distance," *In TKDE*, vol. 19, no. 3, pp. 370–383, 2007.

[150] J. He, A.-H. Tan, C.-L. Tan, and S.-Y. Sung, "On quantitative evaluation of clustering systems," *Clustering and Information Retrieval, Kluwer Academic Publishers*, pp. 105–133, 2003.

[151] L. Li and Y. Liang, "A hierarchical fuzzy clustering algorithm," *In proc. ICCASM*, pp. 248–255, 2010.

[152] L. Meng and A.-H. Tan, "Heterogeneous learning of visual and textual features for social web image co-clustering," *Technical Report, School of Computer Engineering, Nanyang Technological University*, 2012.

[153] K. Lang, "Newsweeder: Learning to filter netnews," *Proc. Int'l Conf. Machine Learning*, pp. 331–339, 2005.

[154] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Computer Vision, Graphics, and Image Processing*, vol. 37, no. 1, pp. 54–115, 1987.

[155] A.-H. Tan, "Adaptive resonance associative map," *Neural Networks*, vol. 8, no. 3, pp. 437–446, 1995.

[156] Y. Zhao and G. Karypis, "Criterion functions for document clustering: experiments and analysis," *Technical Report, Department of Computer Science, University of Minnesota*, 2001.

[157] R. Xu and D. C. W. II, "BARTMAP: A viable structure for biclustering," *Neural Networks*, pp. 709–716, 2011.

[158] X. Wang, B. Qian, J. Ye, and I. Davidson, "Multi-objective multi-view spectral clustering via pareto optimization," *In SDM*, pp. 234–242, 2013.

[159] X. Wang, L. Tang, H. Gao, and H. Liu, "Discovering overlapping groups in social media," *In ICDM*, pp. 569–578, 2010.

[160] L. Tang and H. Liu, "Scalable learning of collective behavior based on sparse social dimensions," *In CIKM*, pp. 1107–1116, 2009.

[161] L. Meng, A.-H. Tan, and D. C. W. II, "Vigilance adaptation in adaptive resonance theory," *In IJCNN*, pp. 1–7, 2013.

[162] P. Chandrika and C. V. Jawahar, "Multi modal semantic indexing for image retrieval," *In proc. International Conference on Image and Video Retrieval*, pp. 342–349, 2010.

[163] F. Gonzalez and J. Caicedo, "Nmf-based multimodal image indexing for querying by visual example," *In proc. International Conference on Image and Video Retrieval*, pp. 366–373, 2010.

[164] Y. Wang, B. Yang, L. Qu, M. Spaniol, and G. Weikum, "Harvesting facts from textual web sources by constrained label propagation," *Proceedings of ACM international conference on Information and knowledge management*, pp. 837–846, 2011.

[165] M. Cooper, J. Foote, A. Girgensohn, and L. Wilcox, "Temporal event clustering for digital photo collections," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 1, no. 3, pp. 269–288, 2005.

[166] A. Graham, H. Garcia-Molina, A. Paepcke, and T. Winograd, "Time as the essence for photo browsing through personal digital libraries," *Proc. Joint Conf. on Digital Libraries*, pp. 837–846, 2002.