DATA CLUSTERING WITH PAIRWISE CONSTRAINTS

By

Jinfeng Yi

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science - Doctor of Philosophy

2014

ABSTRACT

DATA CLUSTERING WITH PAIRWISE CONSTRAINTS

By

Jinfeng Yi

The classical unsupervised clustering is an ill-posed problem due to the absence of a unique clustering criteria. This issue can be addressed by introducing additional supervised information, usually casts in the form of pairwise constraints, to the clustering procedure. Depending on the sources, most pairwise constraints can be classified into two categories: (i) pairwise constraints collected from a set of non-expert crowd workers, which leads to the problem of crowdclustering, and (ii) pairwise constraints collected from oracle or experts, which leads to the problem of semi-supervised clustering. In both cases, the costs of collecting pairwise constraints can be expensive, thus it is important to identify the minimal number of pairwise constraints needed to accurately recover the underlying true data partition, also known as a sample complexity problem.

In this thesis, we first analyze the sample complexity of crowdclustering. At first, we propose a novel crowdclustering approach based on the theory of matrix completion. Unlike the existing crowdclustering algorithm that is based on a Bayesian generative model, the proposed approach is more desirable since it only needs a much less number of crowdsourced pairwise annotations to accurately cluster all the objects. Our theoretical analysis shows that in order to accurately cluster N objects, only $O(N \log^2 N)$ randomly sampled pairs should be annotated by crowd workers. To further reduce the sample complexity, we then introduce a semi-crowdsourced clustering framework that is able to effectively incorporate the low-level

features of the objects to be clustered. In this framework, we only need to sample a subset of $n \ll N$ objects and generate their pairwise constraints via crowdsourcing. After completing a $n \times n$ similarity matrix using the proposed crowdclustering algorithm, we can further recover a $N \times N$ similarity matrix by applying a regression-based distance metric learning algorithm to the completed smaller size similarity matrix. This enables us to reliably cluster N objects with only $O(n \log^2 n)$ crowdsourced pairwise constraints.

Next, we study the problem of sample complexity in semi-supervised clustering. To this end, we propose a novel convex semi-supervised clustering approach based on the theory of matrix completion. In order to reduce the number of pairwise constraints needed we apply a nature assumption that the feature representations of the objects are able to reflect the similarities between objects. This enables us to only utilize $O(\log N)$ pairwise constraints to perfectly recover the data partition of N objects.

Lastly, in addition to sample complexity that relates to labeling costs, we also consider the computational costs of semi-supervised clustering. Specifically, we study the problem of efficiently updating clustering results when the pairwise constraints are generated sequentially, a common case in various real-world applications such as social networks. To address this issue, we develop a dynamic semi-supervised clustering algorithm that casts the clustering problem into a searching problem in a feasible convex space, i.e., a convex hull with its extreme points being an ensemble of multiple data partitions. Unlike classical semi-supervised clustering algorithms that need to re-optimize their objective functions when new pairwise constraints are generated, the proposed method only needs to update a low-dimensional vector and its time complexity is irrelevant to the number of data points to be clustered. This enables us to update large-scale clustering results in an extremely efficient way.

To my family.

ACKNOWLEDGMENTS

First of all, I want to express my sincere gratitude to my thesis advisor Prof. Rong Jin, for his continuous support, encouragement, and help during my Ph.D. study. Back to five years ago, I was merely an undergraduate student knowing almost nothing about research. It is his help and advice opened me the door to the world of scientific research. Throughout the years, he always inspired me to tackle interesting research problems and also encourages me to work on any problems that interest me. From him, I have learned a lot not only about modeling techniques, but also how to formalize and solve a problem by myself. More importantly, his passion in research always motivates me and let me understand how enjoyable it is to learn and to do research. Without Prof. Jin, this thesis would not be possible.

I also want to express my gratitude to Prof. Anil K. Jain. We worked together on multiple interesting research problems in the areas of crowdsourcing and data clustering, which counts for a significant portion of this thesis. I really appreciate him for pointing me to right directions, asking me good questions and encouraging me. I have learned a great deal from him about how to find a valuable research problem, and how to write a convincing technical paper. In addition, I have greatly benefited from his expertise on his insightful career advice. I would also like to thank my committee members Dr. Joyce Chai, Dr. Pang-ning Tan and Dr. Selin Aviyente, who offered lots of useful advices and suggestions to this thesis.

I am also fortunate enough to be mentored by many good researchers during internships. I did my first summer intern in Stroz Friedberg LLC and this is also my first industrial experience. I am grateful to my manager, Dr. Michael Sperling, for guiding me to conduct project-oriented research. This was a great learning experience for me. I also spent a wonderful summer at

IBM Thomas J. Watson Research Center, mentored by Dr. Jun Wang and Dr. Aleksandra Mojsilovic. I enjoyed the discussions with them on research problems and I want to thank both of them for making my last summer memorable and productive.

In the last several years, I greatly enjoyed the collaborations and friendships with my labmates: Tianbao Yang, Wei Tong, Lijun Zhang, Mehrdad Mahdavi, Yang Zhou, Qi Qian and many others. I wish you all the best in your future careers. I also appreciate the help provided from the stuffs at CSE department of MSU, especially Linda Moore and Norma Teague.

Finally and most importantly, I would like to thank my family for their unconditional love and constant encouragement. To them I owe a lot and words cannot describe my love for them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Data clustering is a method of grouping similar objects together. Given a representation of N objects, the goal of data clustering is to find r groups based on a measure of similarity such that objects within the same group are alike but the objects in different groups are not alike [71]. Data clustering is an important problem that has found numerous applications in different domains, including computer vision [55], information retrieval [11, 91], bioinformatics [85, 140], recommender systems [83], etc.

Generally speaking, data clustering is considered as an unsupervised learning technique in which the input data only contains the data points themselves without any additional information. The unsupervised nature makes data clustering an ill-posed problem since data can be usually partitioned in many equally valid ways depending on users' intent and goal. Consider a simple clustering task that groups 4 face images into 2 clusters, as shown in Figure 1.1. The 4 face images are taken from a young man, a young woman, an old man, and an old woman, respectively. These images can be clustered either based on the gender of the people, as shown in Figure 1.1(b), or based on the age of the people, as shown in Figure 1.1(c). Both these partitions are equally valid, implying that it is not possible to determine the correct partition without further information. Such information, usually cast

(a) 4 face images need to be grouped into 2 clusters

(b) partition the face images based on gender

(c) partition the face images based on age

Figure 1.1 A clustering of 4 face images into 2 clusters. It is possible to cluster these face images in two ways, with both of them equally valid. They can be clustered based on gender as shown in (b), or clustered based on age as shown in (c). Without further information from the user, it is not possible to determine the correct partition.

in the form of pairwise constraints, can be used to tune the clustering algorithm towards finding the data partition sought by the user.

Pairwise constraints are introduced by Wagstaff et al. [123] to specify the relationship between class assignments of two objects, which are also known as must-link and cannot-link information. A must-link constraint between two data points implies that these two data points tend to be assigned to the same cluster, while a cannot-link constraint between two data points implies that they tend to be assigned to different clusters. These link information thus provide constraints on clustering results. Must-link and cannot-link have some interesting properties. Must-link constraint is an equivalence relation and hence are symmetrical, reflexive and transitive. For example, if object x and object y are connected by a must-link, and object y and object z are connected by a must-link, then object x and object z should also be connected by a must-link. Similarly, must-link constraints can give rise to cannot-link constraints. For example, if x and y are connected by a must-link, y and z are connected

by a cannot-link, then x and z can also be connected by a cannot-link. Despite the simple definition, must-and-cannot links are powerful and widely used for facilitating data clustering [32, 124]. In this thesis, we address two main issues regarding incorporating pairwise connections in the data clustering problem: (i) how is the pairwise constraints obtained and specified?, and (ii) how the clustering performance is improved with increasing number of pairwise constraints, an issue that is usually referred to as *sample complexity* in supervised learning [7]. In the sequel, we discuss both of these questions.

Depending on the sources, most pairwise constraints can be classified into two categories. In the first category, the pairwise constraints are collected from a set of non-expert crowd workers through crowdsourcing tools such as the Amazon's Mechanical Turk [14]. Then the problem of incorporating all the crowdsourced pairwise constraints into a data partitioning is denoted as crowdsourced clustering, or **crowdclustering** [60, 141] for short. Since the pairwise constraints collected via crowdsourcing usually have a high noise level, crowdclustering can be considered as one problem of *clustering with noisy pairwise constraints*. In the second category, a small amount of pairwise constraints are collected from oracle or experts. Then the problem of searching for the optimal data partition that is consistent with both the given pairwise constraints and the input data points to be clustered leads to the problem of **semi-supervised clustering**. Since most semi-supervised clustering frameworks assume that the provided pairwise constraints are perfect, it can be considered as a problem of *clustering with noiseless pairwise constraints*. Below, we describe each of them in detail.

1.1 Crowdclustering

Crowdclustering is a technique of utilizing human power in acquiring similarities between objects need to be clustered. The main focus of crowdsourced clustering is to combine a set of crowdsourced pairwise constraints to form a data partitioning of the entire data set. In more detail, given a collection of objects to be clustered, a subset of objects is first sampled in each *Human Intelligence Task* (HIT), and a crowd worker is asked to annotate the subset of objects in the HIT based on their own opinion. The annotation task can either be grouping objects based on their similarities or describing individual objects by multiple keywords; the annotation results then can be summarized in the form of pairwise constraints. The keyword annotation is transformed into binary pairwise constraints by checking if two objects share common annotated keywords. The results of each HIT, which can be considered as a partial local clustering of the objects in that HIT, are then combined to form a data partitioning of the entire data set. Note that the question of crowdclustering is motivated by practical considerations: if we have a large number of objects, it may not be realistic to expect a single person to look at all the objects and form an opinion as to how to group them. Crowdclustering solves one important issue in data clustering, namely the inconsistency between the similarity computed by data points' attributes and the similarity in human perception. This is due to the reason that crowdclustering obtains similarity measures between objects based on manual annotations, which capture the human perception of similarity among objects.

Two major problems play an important role in crowdclustering. First, since crowd workers are paid by the number of crowdsourcing tasks they worked on, it is important to analyze the

minimal number of crowdsourced pairwise constraints needed to recover an accurate enough data partition. The second major problem of crowdclustering lies in the issue of high noise levels in crowdsourced pairwise constraints. This is because that most crowd workers are untrained non-experts. They sometimes assign low-quality labels due to the reasons such as they do not understand the labeling criteria, or they do not look at the instances carefully when labeling. As pointed out by [141], more than 80% of crowdsourced pairwise labels can be inconsistent with the true cluster assignment. Thus how to identify the true data partitions based on the noisy crowdsourced pairwise constraints becomes one main challenge in crowdclustering.

1.2 Semi-supervised Clustering

In semi-supervised clustering, the pairwise constraints are collected from oracle or experts. In contrast to crowdclustering, most semi-supervised clustering frameworks assume that the provided pairwise constraints are perfect. These pairwise constraints, also known as side information, can be incorporated in the clustering process to attain better clustering performance. Several mechanisms have been developed to exploit the side-information to improve the clustering performance. Most semi-supervised clustering algorithms can be classified into two categories [12]: constrained clustering and distance metric based semi-supervised clustering. The constrained clustering employs the side information to restrict the solution space, then only find the feasible data partitions that are consistent with the pairwise constraints. The distance metric based semi-supervised clustering attempts to find and apply a transformation to the data such that (a) the data points in must-link constraints are separated

by small distances, and (b) data point in cannot-link constraints are separated by larger distances.

Generally speaking, the larger the number of pairwise constraints been provided, the better the clustering performance can be achieved by semi-supervised clustering. However, one major issue regarding incorporating pairwise constraints in the data clustering problem is how the clustering performance is improved with increasing number of pairwise constraints, also known as the *sample complexity* problem in supervised learning [7]. Since collecting pairwise constraints from oracle or experts are usually very expensive, it becomes crucial to discover the minimal number of pairwise constraints required to perfectly recover the underlying true data partition in semi-supervised clustering.

In addition to sample complexity that relates to the labeling costs, computational cost is also an important issue of semi-supervised clustering. In this thesis, we also study the problem of efficiently updating the clustering results when the pairwise constraints are generated sequentially, a common case in various real-world applications. For example, in social networks, we can treat the user attributes like gender, educational background, nationality, interests and so on as features, and the social connections like friendship and common community membership as the pairwise constraints. Hence, the task of grouping users in social networks is essentially a semi-supervised clustering problem. Note that the pairwise constraints in social networks are changing all over the time, a clustering algorithm that is able to cope with dynamic pairwise information is needed to solve this dynamic semi-supervised clustering problem. However, given a set of new pairwise constraints, classical semi-supervised clustering algorithms need to re-optimize the objective function over all of the data points subject to all the received connections, making them computationally infeasible for updating

large-scale data clustering results. Thus how to develop an efficient dynamic semi-supervised clustering framework becomes an interesting and important problem.

1.3 Main Contributions

In this thesis, we focus on the problem of data clustering with pairwise constraints. In particular, this thesis addresses a number of important issues in both the crowdclustering problem and the problem of semi-supervised clustering. Below, we briefly summarize each of our contributions.

- **Crowdclustering by Matrix Completion**

  The first problem we consider in this thesis is to reduce the number of pairwise annotations needed in crowdclustering. In particular, we propose a novel crowdclustering approach based on the theory of matrix completion [20]. The basic idea of the proposed algorithm is to construct a partially observed similarity matrix based on a subset of pairwise annotation labels that are agreed upon by most crowd workers. It then deploys the matrix completion algorithm to complete the similarity matrix and obtains the final data partition by applying a spectral clustering algorithm to the completed similarity matrix. Unlike the existing work of crowdclustering [60] that based on a Bayesian generative model, the main advantage of the proposed approach is that much less crowdsourced pairwise annotations are needed to accurately cluster all the objects. This is due to the reason that the Bayesian approach requires a sufficiently large number of manual annotations to discover the hidden factors for clustering decision. This results in high cost, both in computation and annotation, which limits the scalability

to clustering large data sets. In contrast, the proposed algorithm can significantly reduce the demanding of manual annotations due to a key observation, i.e. the complete similarity matrix for all the objects should be of low rank [73]. According to the matrix completion theory [20], when an $N \times N$ matrix is of low rank, it can be perfectly recovered given only a very small portion of entries (i.e. $O(\log^2 N/N)$). This guarantees that the proposed crowdclustering algorithm can accurately discover the underlying data partitions with only small number of crowdsourced annotations. Another advantage of the proposed crowclustering algorithm is that by filtering out the uncertain data pairs, the proposed algorithm is less sensitive to the noisy labels, leading to a more robust clustering.

- **Semi-Crowdsourced Clustering by Distance Metric Learning**

  As our second problem, we study the crowdclustering framework when most of the objects are not manually annotated by crowd workers. This question is motivated by a practical consideration of clustering large-scale data sets. Since it is very expensive to hire a large amount of crowd workers for annotation, it is not feasible to have each object manually annotated by crowd workers. To address this limitation, we propose a new approach for clustering, called *semi-crowdsourced clustering* that effectively combines the low-level features of objects with the manual annotations of a subset of the objects obtained via crowdsourcing. The key idea is to learn an appropriate similarity measure, based on the low-level features of objects and from the manual annotations of only a small portion of the data to be clustered. In more detail, the proposed algorithm for clustering N objects consists of three steps: (i) in the first step, we randomly sample a subset of $n \ll N$ objects and obtain their manual annotations by crowdsourcing.

Then we filter noisy pairwise similarities for n objects by only keeping object pairs whose pairwise similarities are agreed by many workers. The result of this step is a partially observed $n \times n$ similarity matrix with most of its entries removed or unobserved; (ii) we then recover the $n \times n$ similarity matrix from the partially observed entries by exploiting the matrix completion algorithm; and (iii) in the third step, we apply a regression algorithm to learn a distance metric from the recovered similarity matrix, then clustering the $N \times N$ pairwise similarities based on the learned distance metric. The main advantage of the proposed approach is that, in order to cluster N objects, we only need a small subset to be annotated by crowdsourcing.

- **Semi-supervised Clustering by Pattern Assisted Matrix Completion**

  We then turn to the problem of semi-supervised clustering. In particular, we address two main shortcomings with the existing semi-supervised clustering algorithms. First, most semi-supervised clustering algorithms have to deal with non-convex optimization problems, leading to clustering results that are only locally optimal and sensitive to the initialization. Second, although many computational algorithms have been proposed for semi-supervised learning, none of them is equipped with a theoretical guarantee on clustering performance. We address these limitations by developing a novel convex semi-supervised clustering approach based on the theory of matrix completion. Instead of penalizing the violations of pairwise constraints, which usually leads to non-convexity, we treat them as the subset of *observed entries* in the similarity matrix of the N objects need to be clustered. Namely, we consider the must-link as similarity 1 and cannot-link as similarity 0. We then deploy the matrix completion method to complete the partially observed similarity matrix under the key observation that the complete

similarity matrix should be of low rank. To reduce the number of pairwise constraints needed, we apply a nature assumption that the feature representations of the objects are good enough, i.e., they are able to reflect the similarities between objects. In more detail, we assume that the singular vectors of the similarity matrix should lie in the subspace spanned by the first k eigenvectors of feature representations of the objects. Our analysis shows that under this assumption, only $O(\log N)$ pairwise constraints are needed to accurately recover the true cluster partition. This logarithmic dependence on the number of objects been clustered makes the proposed algorithm particularly suitable for clustering large-scale data sets.

- **A Constant-Time Algorithm for Dynamic Semi-Supervised Clustering**

  Another problem we address in this thesis is how to efficiently update clustering results when the pairwise constraints are dynamic. Our motivation stems from the observation that in numerous real-world applications, the pairwise constraints are not fixed, which is different from the assumptions made by most classical semi-supervised clustering algorithms. A typical example is social network analysis. If we treat user profiles as features, and connections between users as pairwise constraints, then the task of grouping user communities is essentially a dynamic semi-supervised clustering problem. Given a set of newly generated pairwise constraints, classical semi-supervised clustering algorithms need to re-optimize their objective functions over all the data points and constraints, prohibiting them to efficiently update the data partitions. To address this issue, we propose an efficient dynamic semi-supervised clustering framework that casts the clustering problem into a searching problem in a feasible convex space, i.e., a convex hull with its extreme points being an ensemble of multiple data partitions.

According to the principle of ensemble clustering, the optimal partition lies in that convex hull and it can be uniquely represented by a low-dimensional *simplex* vector. This enables us to carry out the dynamic semi-supervised clustering problem as an updating procedure of the simplex vector based on the newly received pairwise constraints. Using this idea, we derive a *constant* time algorithm for updating the simplex vector (clustering result) and this enables us to update large-scale clustering results in an extremely efficient way.

The remainder of this thesis is organized as follows. In Chapter 2, we provide a survey of some background materials from crowdclustering, ensemble clustering, semi-supervised clustering and matrix completion. In Chapter 3, we present the framework of crowdclustering by matrix completion. This is based on our work published in the 4th Human Computation Workshop in junction with AAAI (HCOMP) [141]. Chapter 4 introduces the proposed semi-crowdsourced clustering framework. The contents in Chapter 4 follows our paper appeared in Advances in Neural Information Processing Systems (NIPS) [142]. We then present the proposed semi-supervised clustering method in Chapter 5. The material in Chapter 5 come from our work published in the International Conference on Machine Learning (ICML) [143]. In Chapter 6, we focus on the problem of efficiently updating clustering results when the pairwise constraints are dynamic. Finally, we conclude the thesis and discuss some future directions that can be explored in Chapter 7.

Chapter 2

Literature Survey

The goal of this chapter is to give an overview of the material related to the work has been done in this thesis. In particular, we will survey the problems of crowdclustering, ensemble clustering, semi-supervised clustering, the evaluation metrics of data clustering and the technique of matrix completion.

2.1 Crowdclustering

The idea of crowdclustering was first proposed in [60]. With the advent of crowdsourcing services such as Amazon's Mechanical Turk, it becomes much more convenient to purchase Human Intelligence Tasks from large groups of anonymous crowd workers. Note that the similarity between two objects computed using their attributes may not reflect human perception of inter-object similarity, crowdsourcing provides an easy way to address this issue by utilizing human power in acquiring pairwise similarities between objects. Generally speaking, crowdclustering is a divide and conquer procedure with two steps. In the first step, the problem of clustering N objects is reduced to a number of small problems. Each small problem, denoted as a human intelligence task (HIT), contains a set of objects with reasonable

size. Then the human intelligence tasks are assigned to a large pool of human workers for annotations. The annotation tasks can be either grouping objects based on their similarities, or describing individual objects by multiple keywords. Then the annotation results can be summarized in the form of pairwise constraints. In the second step, a model need to be developed to aggregate the human annotations automatically that yield a partition of N objects into clusters. Crowdclustering is a challenge problem due to the following reasons: (i) each worker has only a partial view of the data, (ii) different workers may have different clustering criteria and may also produce different numbers of clusters, and (iii) the annotation quality varies significantly among different workers since some workers are spammers or malicious users.

To address such issues, the authors in [60] proposed a Bayesian generative model for crowdclustering. In more detail, the objects to be clustered are represented in a Euclidean space and workers are modeled as pairwise binary classifiers in this space. Then the clusters are obtained by clustering these inferred points using a Dirichlet process mixture model [99]. However, one limitation of the Bayesian approach for crowdclustering is that in order to discover the hidden factors for clustering decision, it requires a sufficiently large number of manual annotations. This leads to high annotation cost for clustering large-scale data sets.

2.2 Ensemble Clustering

The main idea behind ensemble clustering [54, 116] is to combine multiple partitions of a dataset into a single data partition, termed as the consensus partition, hoping to exploit the strength of different clustering algorithms and at the same time, compensate for their limitations. This problem is related to crowdclustering problem since both of them aim to combine multiple partitions of data into a single clustering result.

According to [121], ensemble clustering can be classified into two categories: median partition based approaches and object co-occurrence based approaches. In the median partition based approaches, ensemble clustering is cast into an optimization problem that finds the partition by maximizing the within-cluster similarity, where various similarity measures have been proposed, such as Mirkin distance [126], Jaccard coefficient [10], utility function [117] and normalized mutual information [113]. Among the approaches based on object co-occurrence, one major category is the relabeling/voting based method [5, 41, 44, 50, 119, 131]. The basic idea is to first find the corresponding cluster labels between different partitions, and then obtain the consensus partition through a voting process. The second group of approaches in this category is based on co-association/similarity matrix [54, 69, 87, 120, 128]. They use the similarity measure to combine multiple partitions, thus avoiding the label correspondence problem. The third group of approaches in this category is the graph based methods [49,113]. These methods construct a weighted graph to represent multiple partitions from the ensemble and find the optimal partition of data by minimizing the cut of the graph.

While some of ensemble clustering methods can work with partial input clusterings, most have not been demonstrated in situations where the input clusterings involve only a small

subset of the objects to be clustered, which is the case in the problem of crowdclustering. In addition, since different human workers in crowdclustering may have different clustering criterion, they may produce various partial clustering results. This usually makes a significant amount of noise and inter-worker variations in their clustering results. As a consequence, there may exist a large number of uncertain data pairs for which about half of the human workers put them into the same cluster while the other half do the opposite. These uncertain data pairs can mislead the ensemble clustering algorithms to create inappropriate data partitions.

2.3 Semi-supervised Clustering

There are two major approaches for clustering with semi-supervised information: the constrained clustering and the approach based on distance metric learning. In this section, we review the existing work for both of them, followed by two categories of dynamic clustering: clustering based on user feedbacks, as well as dynamic network clustering.

2.3.1 Constrained Clustering

The constrained clustering employs the side information to restrict the solution space, and only find the feasible data partitions that are consistent with the pairwise constraints. Among them, hard constraints-based approaches [2,30,31,33,34,108,125,125] only consider the cluster assignments that all the constraints are strictly satisfied. In [125], the cluster centers are first initialized randomly. Each data point is then assigned to the nearest cluster center en-

suring that no constraints are violated. The cluster centers are updated by finding the mean of the points assigned to the cluster, like in the K-means algorithm. In [2], the authors modified the k-means and the self organizing map algorithms respectively, to adjust the cluster memberships to be consistent with the given pairwise constraints. In [108], the generalized Expectation Maximization (EM) algorithm was modified to ensure that only the mixture models that are consistent with the constraints are considered. One problem with treating the side information as hard constraints is that such methods may lead to counter-intuitive clustering solutions, or even render the clustering problem infeasible [31]. To overcome this problem, a number of studies view the side information as soft constraints [8, 9, 30, 80, 93]. Instead of trying to satisfy all the constraints, the key idea behind such methods is to satisfy as many constraints as possible, and introduce a penalty term to account for constraints that cannot be satisfied. In [8, 9, 93], probabilistic models are applied to semi-supervised clustering problem and they consider pairwise constraints as Bayesian priors to learn the model. In [74], the pairwise constraints are added as an additional penalty term in the objective in spectral learning. In [80], the authors proposed a graphical model which considered the pairwise constraints as random variables. Then an EM-based algorithm was developed to model the uncertainty of constraints. In [79], the authors enhanced the performance of the k-means algorithm by constructing a kernel matrix, which incorporates the given pairwise constraints. More discussion on constrained clustering can be found in [35], and the references therein.

2.3.2 Distance Metric Learning

Another approach to semi-supervised clustering falls into the category of distance metric learning. It aims to learn a transformation to the data from the pairwise constraints such that the data points in must-link constraints are separated by smaller distances, than the data point in cannot-link constraints [137]. Many algorithms [59, 64, 109, 129, 135, 139] have been developed for distance metric learning, such as distance metric learning by convex optimization [135], relevance component analysis [109], discriminative component analysis [64], nearest neighbor component analysis [59], local distance metric learning [139], large margin nearest neighbor classifier [129], information theoretic metric learning [37], distance function learning [115], and learning a Bregman distance function [133]. In [135], the authors formulated the problem of distance metric learning into a PSD constrained convex programming problem. The authors in [129] proposed a nearest-neighbor classifier to enforce that the examples from different classes were separated by a large margin. In [37], an information-theoretic method was proposed to learn a Mahalanobis distance function. In [6], relevant component analysis learns a distance metric by assigning large weights to relevant dimensions while low weights to irrelevant dimensions. It was further improved in [64] to effectively explore both the must-link and the cannot-link constraints simultaneously. More work on distance metric learning can be found in survey [137] and references therein. One problem with approaches based on distance metric learning is that they usually need to deal with the positive semi-definite constraint and this makes them computationally expensive.

Despite the progress, there are two main shortcomings with the exisiting algorithms for semi-supervised clustering. First, most semi-supervised clustering algorithms have to deal

with non-convex optimization problems, leading to the clustering results that are sensitive to the initializations. Second, although many computational algorithms have been proposed for semi-supervised learning, none of the exisiting semi-supervised clustering algorithms analyze the important sample complexity problem, namely how the clustering performance is improved with increasing number of pairwise constraints.

2.3.3 Dynamic Clustering

In recent years, an increasing amount of literature studied the problem of clustering evolving. Among them, clustering based on user feedbacks [13, 28] attracted considerable attentions in the last decade. As one of the earliest work, Cohn et al. [28] considered a scenario when users can iteratively provide different types of feedbacks regarding the clustering quality. Then an EM-based scheme was developed to update the distance metric for achieving better clustering results. The algorithm proposed in [13] is a variant of the complete-link hierarchical clustering. It combines the feedback pairwise constraints with spatial constraints to learn a new distance metric that only satisfies local pairwise constraints. Huang and Mitchell [67] proposed a probabilistic generative model for text clustering. This model enables users to provide four types of user feedback to further enhance the clustering performance.

As a related topic, dynamic network clustering [21, 21, 26, 26, 77, 89, 114] studies how a community evolves when a network to be clustered is changing continuously. Evolutionary clustering [21, 26] ensures that the output partition achieves high accuracies on the new data, while still consistent with the historical clustering results. Chakrabarti et al. [21] developed evolutionary versions of both k-means and agglomerative hierarchical clustering algorithms.

Chi et al. [26] then extended this idea to evolutionary spectral clustering algorithm. In [114], a parameter free algorithm called GraphScope was proposed to mine time-evolving graphs using the principle of Minimum Description Length (MDL). FacetNet [89] used probabilistic community membership models to identify dynamic communities within the graph. Kim and Han [77] further allowed a changing number of communities and proposed a particle-and-density based algorithm to form new communities or dissolve existing communities.

2.4 Evaluation Metrics

In this section, we review four widely-used metrics for evaluating the quality of clustering results. They are purity [95], normalized mutual information (NMI for short) [29], pairwise F-measure (PWF for short) [132] and Rand index [105].

Purity focuses on the frequency of the most common category in each cluster, and rewards the clustering solutions that introduce less noise in each cluster [3]. Given the ground truth partition $\mathcal{C} = \{C_1, C_2, \ldots, C_r\}$ and the partition $\mathcal{C}' = \{C'_1, C'_2, \ldots, C'_r\}$ generated by a clustering algorithm, the purity is computed by taking the weighted average of maximal precision values:

$$\text{Purity}(C, C') = \frac{1}{N} \sum_{i=1,\ldots,r} \max_{j=1,\ldots,r} |C_i \cap C'_j|,$$

where N is the total number of objects been clustered. One problem of purity is that a high purity can be easily achieved when the number of clusters is large. Thus purity is not appropriate for evaluating the clustering performance with large number of clusters.

This problem can be addressed by normalized mutual information. Given the ground truth

partition $\mathcal{C}$ and the partition $\mathcal{C}'$ generated by a clustering algorithm, the mutual information is computed as

$$\mathrm{MI}(C, C') = \sum_{C_k, C'_l} p(C_k, C'_l) \log \frac{p(C_k, C'_l)}{p(C_k)p(C'_l)},$$

where $p(C_k)$ denotes the probability that a randomly selected node belongs to the cluster C_k, and $p(C_k, C'_l)$ indicates the joint probability that a randomly selected node belongs to both of the cluster C_k and cluster C'_l. Then the normalized mutual information is given by

$$NMI(C, C') = \frac{2MI(C, C')}{H(C) + H(C')},$$

where $H(C) = \sum_k p(C_k) \log \frac{1}{p(C_k)}$ represents the Shannon entropy of partition C.

Pairwise F-measure is another commonly used measure for evaluating clustering algorithms. Let $\mathcal{A}$ be the set of data pairs that share the same class labels according to the ground truth, and let $\mathcal{B}$ be the set of data pairs that are assigned to the same cluster by a clustering algorithm. Given the pairwise precision and recall that are defined as follows

$$\text{precision} = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A}|}, \quad \text{recall} = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{B}|},$$

then the pairwise F-measure is computed as the harmonic mean of precision and recall, i.e.

$$PWF = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$

An alternative way to evaluate the clustering performance, known as Rand index, considers clustering results as $N(N-1)/2$ object pairs. Four measurements are then introduced to

compute the rand index. They are (i) true positive (TP) decision that assigns two similar objects to the same cluster, (ii) true negative (TN) decision that assigns two dissimilar objects to different clusters, (iii) false positive (FP) decision that assigns two dissimilar objects to the same cluster, and (iv) false negative (FN) decision that assigns two similar objects to different clusters. Then the *Rand index* measures the percentage of decisions that are correct, given by

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}.$$

All the four discussed measurements lie in the range $[0, 1]$ where a value of 1 indicates perfect match between the obtained partition by a clustering algorithm and the ground truth partition and 0 indicates completely mismatch.

2.5 Matrix Completion

Since all the proposed approaches utilize the technique of matrix completion [15], in this section, we review the existing work for matrix completion.

Matrix completion was originally proposed for collaborative filtering [57], where the goal is to predict the ratings of users for all the items given the ratings for a subset of randomly sampled items. Generally speaking, matrix completion aims to reconstruct a data matrix from a small subset of observed entries under the assumption that the data matrix is of low-rank. The observed entries can be either noiseless or noisy. In the noiseless setting, the observed entries are exactly sampled from the underlying true matrix while in the noisy setting, the observed entries are perturbed by some random noises from the true entries. In

the following, we discuss the matrix completion problem under both scenarios.

2.5.1 Matrix Completion with Noiseless Entries

When the sampled entries of a $n_1 \times n_2$ matrix A are revealed without any noise, we can try to recover it by solving the following optimization problem

$$\begin{aligned} \min_{A'} \quad & \text{rank}(A') \\ \text{s.t.} \quad & \mathcal{P}_\Delta(A') = \mathcal{P}_\Delta(\tilde{A}), \end{aligned} \tag{2.1}$$

where $\text{rank}(\cdot)$ is the rank of a matrix and $\mathcal{P}_\Delta : \mathbb{R}^{n_1 \times n_2} \mapsto \mathbb{R}^{n_1 \times n_2}$ is a matrix projection operator that takes a matrix A as the input and outputs a new matrix $\mathcal{P}_\Delta(A) \in \mathbb{R}^{n_1 \times n_2}$ as

$$[\mathcal{P}_\Delta(A)]_{ij} = \begin{cases} A_{ij} & (i,j) \in \Delta \\ 0 & \text{otherwise.} \end{cases} \tag{2.2}$$

This projection operator guarantees that only the observed entries in the matrix can be projected into the space where we apply matrix completion.

One problem with (2.1) is that it is non-convex because $\text{rank}(\cdot)$ is a non-convex function [20]. This makes the problem (2.1) an NP-hard problem and it is therefore computationally challenging to find the optimal solution. To address this challenge, we follow [20] and replace $\text{rank}(A)$ in (2.1) with its convex surrogate $|A|_*$, the trace norm of matrix A. This allows us

to relax (2.1) into the following convex optimization problem

$$\begin{aligned} \min_{A'} \quad & |A'|_* \\ \text{s.t.} \quad & \mathcal{P}_\Delta(A') = \mathcal{P}_\Delta(\tilde{A}). \end{aligned} \tag{2.3}$$

One important question regarding problem (2.3) is what is the minimal number of observed entries needed to perfectly reconstruct the matrix A. The authors in [19] show that, under the assumptions of incoherence property and uniform sampling, the matrix A can be correctly recovered with a high probability by solving problem (2.3) if the number of observed entries is at least $O(n^{6/5} \log n)$ with $n = \max(n_1, n_2)$. This bound was later tightened to $O(n \log^2 n)$ with contributions from [20, 62, 62, 75, 106].

2.5.2 Matrix Completion with Noisy Entries

To solve a more general setting when the sampled entries are noisy, some robust algorithms were developed to accurately recover the underlying matrix. Usually, two types of errors and corruptions are considered in the literature.

In the first case, the locations of noisy entries are assumed to be spread out, meaning that no single column or row has too many corrupted entries. This problem was first studied in [24], which assumes that the received partially observed matrix is a summation of an unknown low-rank matrix and an unknown sparse error matrix. Then several works extend the results of [24] under different settings. One line of work [65] provides worst case guarantees for arbitrary corruption in the entries. Another avenue [16, 56, 76, 86, 100] provides probabilistic

guarantees for the setting when the locations of the noisy entries are chosen uniformly at random but the values of the errors are arbitrary. In addition, approximate recovery in the presence of stochastic additive noise has also been studied in [1, 17].

In contrast, the second case allows all entries from some columns to be entirely corrupted. As a result, it is in general impossible to recover these corrupted columns, and the goal thus becomes recovering the other, uncorrupted columns. This setting is important in many applications, such as the problem of collaborative filtering [57] with malicious users. This problem has been studied in [1, 136], which use convex relaxation of minimizing rank plus column support to distinguish between authentic and corrupted columns.

Chapter 3

Crowdclustering with Sparse Pairwise Labels

In this chapter, we propose a novel approach for crowclustering that exploits the technique of matrix completion. Instead of using all the crowdsourced annotations, the proposed algorithm constructs a partially observed similarity matrix based on a subset of pairwise annotation labels that are agreed upon by most annotators. It then deploys the matrix completion algorithm to complete the similarity matrix and obtains the final data partition by applying a spectral clustering algorithm to the completed similarity matrix. The main advantage of the proposed algorithm is that only a small number of crowdsourced pairwise annotations are needed to accurately cluster all the objects. Our analysis reveals that, by exploiting the technique of matrix completion, we can perfectly recover the underlying true partition of N objects given only a very small portion of reliable pairwise constraints (i.e. $O(\log^2 N/N)$). Another advantage of the proposed crowclustering algorithm is that by filtering out the uncertain data pairs, the proposed algorithm is robust to the noisy crowdsourced labels.

The remainder of the chapter is organized as follows: In Section 3.1, we introduce the back-

ground and discuss the motivation of the proposed crowdclustering framework. Section 3.2 presents the proposed crowdclustering algorithm and the related analysis. We summarize the results of our empirical studies in Section 3.3. Section 3.4 concludes the chapter and we put the omitted proofs in the appendix.

3.1 Introduction

In data clustering problems, crowdsourcing helps to address one of the key challenges, namely how to define the similarity measure between objects. A typical clustering algorithm measures the similarity between two data points based on their attributes. However, these pairwise similarities may not reflect human perception of inter-object similarity in the unsupervised setting. In contrast, crowdclustering utilizes human power in acquiring pairwise similarities by asking each worker to perform clustering on a subset of objects, thereby defining a similarity measure between pairs of objects based on the percentage of workers who put them into the same cluster.

The core of crowdclustering is to combine the partial clustering results, generated by individual workers, into a complete data partition. One way to address this challenge is ensemble clustering [53, 113], as suggested in [60]. There are, however, two special challenges in applying ensemble clustering to the crowdclustering problem. First, since each worker deals with only a subset of the entire dataset (because the task of partitioning a large dataset is too complex for individual workers), only partial clustering results are available in the ensemble for combination. This is in contrast to most ensemble clustering studies that require a

clustering of the complete dataset from individual partitions. Second, there is a significant amount of noise and inter-worker variations in the partial clustering results generated by individual workers. As a consequence, we often observe a large number of uncertain data pairs for which about half of the human workers put them into the same cluster while the other half do the opposite. These uncertain data pairs can mislead the ensemble clustering algorithms to create inappropriate data partitions.

To address the potentially large variations in the pairwise annotation labels provided by different workers (i.e. whether or not two objects should be assigned to the same cluster), a Bayesian generative model was proposed for crowdclustering in [60]. It explicitly models the hidden factors that are deployed by individual workers to group objects into the same cluster. The empirical study in [60] shows encouraging results in comparison to the ensemble clustering methods. However, one limitation of the Bayesian approach for crowdclustering is that in order to discover the hidden factors for clustering decision, it requires a sufficiently large number of manual annotations, or HITs. This results in high cost, both in computation and annotation, which limits the scalability to clustering large data sets.

To overcome the limitation of the Bayesian approach, we propose a novel crowdclustering approach based on the theory of matrix completion [20]. The basic idea is to first compute a *partially observed* similarity matrix based only on the **reliable** pairwise annotation labels, or in other words, the labels that are in agreement with a sufficiently large percentage of the workers. It then completes the partially observed similarity matrix using a matrix completion algorithm, and obtains the final data partition by applying a spectral clustering algorithm [102] to the completed similarity matrix.

The main advantage of the matrix completion approach is that only a small number of pairwise annotations are needed to construct the partially observed similarity matrix. Therefore, we can obtain a clustering accuracy similar to the Bayesian methods, with a substantial reduction in the number of workers and/or the number of HITs performed by individual workers. The high efficiency of the proposed algorithm in exploiting manual annotations arises from a key observation, i.e. the complete similarity matrix for all the objects should be of low rank [73]. According to the theory of matrix completion [20], when an $N \times N$ matrix is of low rank, it can be perfectly recovered given only $O(N \log^2 N)$ entries. Another advantage of the proposed crowclustering algorithm is that by filtering out the uncertain data pairs, the proposed algorithm is less sensitive to the noisy labels, making the clustering results more robust.

3.2 Crowdclustering by Matrix Completion

The key idea of the proposed crowdclustering algorithm is to derive a *partially observed* similarity matrix from the partial clustering results generated by individual workers, where the entries associated with the uncertain data pairs are marked as *unobserved*. A matrix completion algorithm is applied to complete the partially observed similarity matrix by filtering out the unobserved entries. Finally, a spectral clustering algorithm [102] is applied to the completed similarity matrix to obtain the final clustering. Below, we describe in detail the two key steps of the proposed algorithm, i.e., the *filtering step* that removes the entries associated with the uncertain data pairs from the similarity matrix, and the *matrix completion step* that completes the partially observed similarity matrix.

The notations described below will be used throughout the paper. Let N be the total number of objects that need to be clustered, and m be the number of HITs. We assume that the true number of clusters in the data is known a priori[1]. Given the partial clustering result from the k-th HIT, we define a similarity matrix $W^k \in \mathbb{R}^{N\times N}$ such that $W^k_{ij} = 1$ if objects i and j are assigned to the same cluster, 0 if they are assigned to different clusters, and -1 if the pairwise label for the two objects can not be derived from the partial clustering result (i.e. neither object i nor object j is used in the HIT). Finally, given a subset of object pairs $\Delta \subset \{(i,j),\ i,j = 1,\dots N\}$, we define a matrix projection operator $\mathcal{P}_\Delta : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{N\times N}$ that takes a matrix E as the input and outputs a new matrix $\mathcal{P}_\Delta(E) \in \mathbb{R}^{N\times N}$ as

$$[\mathcal{P}_\Delta(E)]_{ij} = \begin{cases} E_{ij} & (i,j) \in \Delta \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

This projection operator guarantees that only the reliable entries in the matrix can be projected into the space where we apply matrix completion.

3.2.1 Filtering Entries with Unlabeled and Uncertain Data Pairs

The purpose of the filtering step is to remove the uncertain data pairs from the manual annotations. To this end, given the m similarity matrices $\{W^k\}_{k=1}^m$ obtained from individual

[1]We can relax this requirement by estimating the number of clusters via some heuristic, eg., considering the number of clusters as the rank of the completed matrix A.

workers, we first compute matrix $A = [A_{ij}] \in \mathbb{R}^{N \times N}$ as the average of $\{W^k\}_{k=1}^m$, i.e.,

$$A_{ij} = \begin{cases} \sum_{k=1}^m \frac{W_{ij}^k I(W_{ij}^k \geq 0)}{\sum_{l=1}^m I(W_{ij}^l \geq 0)} & \sum_{l=1}^m I(W_{ij}^l \geq 0) > 0, \\ -1 & \text{otherwise} \end{cases}$$

where $I(z)$ is an indicator function that outputs 1 when z is true and zero, otherwise. We introduce the indicator function $I(W_{ij}^k \geq 0)$ in the above equation so that only the labeled pairs of objects will be counted in computing A.

Since $A_{ij} \in [0, 1]$ for a labeled data pair (i.e. $A_{ij} \geq 0$) measures the percentage of HITs that assign objects i and j to the same cluster, it can be used as the basis for the uncertainty measure. In particular, we define the set of reliable data pairs whose labelings are agreed upon by the percentage of workers as

$$\Delta = \{(i, j) \in [N] \times [N] : A_{ij} \geq 0, A_{ij} \notin (d_0, d_1)\}$$

where $d_0 < d_1 \in [0, 1]$ are two thresholds that will be determined depending on the quality of the annotations. We then construct the partially observed similarity matrix $\tilde{A}$ as follows

$$\tilde{A}_{ij} = \begin{cases} 1 & (i, j) \in \Delta, A_{ij} \geq d_1 \\ 0 & (i, j) \in \Delta, A_{ij} \leq d_0 \\ \text{unobserved} & (i, j) \notin \Delta \end{cases} \tag{3.2}$$

3.2.2 Completing the Partially Observed Matrix

The second step of the algorithm is to reconstruct the full similarity matrix $A^* \in \mathbb{R}^{N \times N}$ based on the partially observed matrix $\tilde{A}$. To this end, we need to make several reasonable assumptions about the relationship between $\tilde{A}$ and A^*.

A simple approach is to assume $\tilde{A}_{ij} = A^*_{ij}, \forall (i,j) \in \Delta$; in other words, assume that all the observed entries in matrix $\tilde{A}$ are correct. This, however, is unrealistic because $\tilde{A}$ is constructed from the partial clustering results generated by different workers, and we expect a significant amount of noise in individual clustering results. Thus, a more realistic assumption is $\tilde{A}_{ij} = A^*_{ij}$ for *most* of the observed entries in Δ. We introduce the matrix $E \in \mathbb{R}^{N \times N}$ to capture the noise in $\tilde{A}$, i.e.,

$$\mathcal{P}_\Delta(A^* + E) = \mathcal{P}_\Delta(\tilde{A}), \tag{3.3}$$

where $\mathcal{P}_\Delta$ is a matrix projection operator defined in (3.1). Under this assumption, we expect E to be a sparse matrix with most of its entries being zero.

The assumption specified in equation (3.3) is insufficient to recover the full similarity A^* as we can fill the unobserved entries (i.e., $(i,j) \notin \Delta$) in A^* with any values. An additional assumption is needed to make it possible to recover the full matrix from a partially observed one. To this end, we follow the theory of matrix completion [20] by assuming the full similarity A^* to be of low rank. It was shown in [73] that when the similarity matrix A^* is constructed from a given clustering (i.e. $A^*_{ij} = 1$ when objects i and j are assigned to the same cluster and zero, otherwise), its rank is equal to the number of clusters. As a result, when the number

of clusters is relatively small, it is reasonable to assume A^* to be of low rank.

Combining the two assumptions together leads to the following approach, to recover the full similarity matrix A^* from the partially observed matrix $\tilde{A}$. We decompose $\tilde{A}$ into the sum of two matrices E and A^*, where E is a sparse matrix that captures the noise in $\tilde{A}$ and A^* is a low rank matrix that gives the similarity between any two objects. Based on this idea, we cast the matrix recovery problem into the following optimization problem

$$\begin{aligned} \min_{A',E} \quad & \text{rank}(A') + C\|E\|_1 \\ \text{s.t.} \quad & \mathcal{P}_\Delta(A' + E) = \mathcal{P}_\Delta(\tilde{A}) \end{aligned} \tag{3.4}$$

where $\|X\|_1 = \sum_{ij} |X_{ij}|$ is the ℓ_1 norm of matrix X that measures the sparsity of X. Parameter $C > 0$ is introduced to balance the two objectives, i.e., finding a low rank similarity matrix A' and a sparse matrix E for noise. We will discuss in Section 3.3.2 about how to automatically determine the value of C.

One problem with the objective function in (3.4) is that it is non-convex because rank$(\cdot)$ is a non-convex function [20]. It is therefore computationally challenging to find the optimal solution for (3.4). To address this challenge, we follow [20] and replace rank(A') in (3.4) with its convex surrogate $|A'|_*$, the trace norm of matrix A'. This allows us to relax (3.4) into the following convex optimization problem

$$\begin{aligned} \min_{A',E} \quad & |A'|_* + C\|E\|_1 \\ \text{s. t.} \quad & \mathcal{P}_\Delta(A' + E) = \mathcal{P}_\Delta(\tilde{A}). \end{aligned} \tag{3.5}$$

We then use the efficient first order algorithm developed in [90] to solve the optimization problem in (3.5).

Given the completed similarity matrix A^* obtained from (3.5), we apply the spectral clustering algorithm [102] to compute the final data partition, which is essentially an application of k-means algorithm [94] to the data projected into the space of the top r eigenvectors of A^*.

3.2.3 Theoretical Analysis

A theoretical question is whether the similarity matrix obtained by (3.5) is close to the true similarity matrix A^*. Our theoretical analysis gives a positive answer to this question. In the following, we show that, under appropriate conditions about the eigenvectors of A^*, A^* can be **perfectly** recovered by (3.5) if the number of noisy data pairs is significantly smaller than the number of observed data pairs.

First, we need to make a few assumptions about A^* besides its low rank. Let A^* be a low-rank matrix of rank r, with a singular value decompsition $A^* = U\Sigma V^\top$, where $U = (\mathbf{u}_1, \ldots, \mathbf{u}_r) \in \mathbb{R}^{N\times r}$ and $V = (\mathbf{v}_1, \ldots, \mathbf{v}_r) \in \mathbb{R}^{N\times r}$ are the left and right eigenvectors of A^*, satisfying the following incoherence assumptions.

- **A1:** The row and column spaces of A^* have coherence bounded above by some positive number μ_0, i.e.,

$$\max_{i\in[N]} \|P_U(\mathbf{e}_i)\|_2^2 \leq \frac{\mu_0 r}{N}, \quad \max_{i\in[N]} \|P_V(\mathbf{e}_i)\|_2^2 \leq \frac{\mu_0 r}{N}$$

where $\mathbf{e}_i$ is the standard basis vector.

- **A2:** The matrix $E = UV^\top$ has a maximum entry bounded by $\frac{\mu_1\sqrt{r}}{N}$ in absolute value for some positive μ_1, i.e. $|E_{i,j}| \leq \frac{\mu_1\sqrt{r}}{N}, \forall (i,j) \in [N] \times [N]$,

where P_U and P_V denote the orthogonal projections on the column space and row space of A^*, respectively, i.e.

$$P_U = UU^\top, \quad P_V = VV^\top$$

Remark The assumptions **A1** and **A2** essentially assume that for both the left and right eigenvectors, the values are spread over all the entries. In other words, no entry in those eigenvectors has its value dominated by the other entries of the same eigenvector. This is the key property that makes it possible to recover the full matrix from a small number of observed entries.

To state our theorem, we need to introduce a few notations. Let $\xi(A')$ and $\mu(A')$ denote the low-rank and sparsity incoherence of matrix A' defined by [22], i.e.

$$\xi(A') = \max_{E \in T(A'), \|E\| \leq 1} \|E\|_\infty \tag{3.6}$$

$$\mu(A') = \max_{E \in \Omega(A'), \|E\|_\infty \leq 1} \|E\| \tag{3.7}$$

where $T(A')$ denotes the space spanned by the elements of the form $\mathbf{u}_k\mathbf{y}^\top$ and $\mathbf{x}\mathbf{v}_k^\top$, for $1 \leq k \leq r$, $\Omega(A')$ denotes the space of matrices that have the same support to A', $\|\cdot\|$ denotes the spectral norm and $\|\cdot\|_\infty$ denotes the largest entry in magnitude. Then the following theorem states the theoretical guarantee of the proposed algorithm.

Theorem 1. *Let $A^* \in \mathbb{R}^{N\times N}$ be a similarity matrix of rank r obeying the incoherence properties (**A1**) and (**A2**), with $\mu = \max(\mu_0, \mu_1)$. Suppose we observe m_1 entries of A^* recorded in $\tilde{A}$ with locations sampled uniformly at random, denoted by $\mathcal{S}$. Under the assumption that m_0 entries randomly sampled from m_1 observed entries are corrupted, denoted by Ω, i.e. $A^*_{ij} \neq \tilde{A}_{ij}, (i,j) \in \Omega$. Given $\mathcal{P}_{\mathcal{S}}(\tilde{A}) = \mathcal{P}_{\mathcal{S}}(A^* + E^*)$, where E^* corresponds to the corrupted entries in Ω. With*

$$\mu(E^*)\xi(A^*) \leq \frac{1}{4r+5}, \quad m_1 - m_0 \geq C_1\mu^4 n(\log n)^2,$$

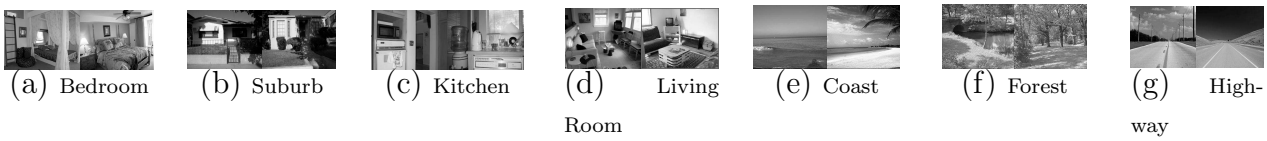
and C_1 is a constant, we have, with a probability at least $1 - N^{-3}$, the solution $(A', E) = (A^*, E^*)$ is the unique optimizer to (4.2) provided that

$$\frac{\xi(A^*) - (2r-1)\xi^2(A^*)\mu(E^*)}{1 - 2(r+1)\xi(A^*)\mu(E^*)} < \lambda < \frac{1 - (4r+5)\xi(A^*)\mu(E^*)}{(r+2)\mu(E^*)}$$

The proof can be found in the appendix. As indicated by Theorem 1, we have a good chance to recover the full similarity matrix A^* if the number of observed correct entries (i.e., m_1) is significantly larger than the number of observed noisy entries (i.e., m_0).

3.3 Experiments

In this section, we first demonstrate empirically that the proposed algorithm can achieve similar or better clustering performance as the Bayesian approach for crowdclustering [60] with significantly lower running time. We further show that, as we reduce the number of



(a) Bedroom (b) Suburb (c) Kitchen (d) Living Room (e) Coast (f) Forest (g) Highway

(h) Inside City (i) Mountain (j) Open Country (k) Street (l) Tall Building (m) Office

Figure 3.1 Some sample images from the 13 categories in the Scenes data set

(a) Human (b) Animal (c) Plant

Figure 3.2 Some sample images from the three categories in the Tattoo data set

pairwise labels, either by reducing the number of workers, or by reducing the number of HITs performed by each worker, the proposed algorithm significantly outperforms the Bayesian approach.

3.3.1 Data Sets

Four image data sets are used in our experiments. They are:

- *Scenes Data Set*: This is a subset of the larger Scenes image data set [46] which has been used in the previous study on crowdclustering [60]. It is comprised of $1,001$ images belonging to 13 categories. Figure 6.3 shows sample images of each category from this data set. To obtain the crowdsourced labels, 131 workers were employed to perform HITs. In each HIT, the worker was asked to group images into multiple clusters, where the number of clusters was determined by individual workers. Pairwise labels between

(a) Tractor (b) Horse cart (c) Bench (d) Blackberry (e) Violin (f) Saxophone (g) Hammer

Figure 3.3 Some sample images from the seven categories in the ImageNet data set

(a) Car (b) Dog (c) Chair (d) Cat (e) Bird

Figure 3.4 Some sample images from the five categories in the PASCAL 07 data set

images are derived from the partial clustering results generated in HITs. The data we used, including the subset of images and the output of HITs, were provided by the authors of [60].

- *Tattoo Data Set*: This is a subset of the Tattoo image database [72]. It contains $3,000$ images that are evenly distributed over three categories: human, animal and plant. Some sample images of each category in the Tattoo data set are shown in Figure 6.4. Unlike the Scenes data set where the objective of HIT was to group the images into clusters, the workers here were asked to annotate tatoo images with keywords of their choice. On average, each image is annotated by three different workers. Pairwise labels between images are derived by comparing the number of matched keywords between images to a threshold (which is set to 1 in our study).

- *ImageNet data set*: This is a subset of the larger ImageNet database [39]. This subset contains $6,408$ images belonging to 7 categories: *tractor*, *horse cart*, *bench*, *blackberry*, *violin*, *saxophone*, and *hammer*. Figure 3.3 shows some sample images of each category in this data set.

- *PASCAL 07 data set*: This is a subset of the PASCAL Visual Object Classes Challenge 2007 database [45]. The subset contains $2,989$ images belonging to five classes: *car*, *dog*, *chair*, *cat* and *bird*. Some sample images of each category in the *PASCAL 07 data set* are shown in Figure 3.4. Similar to the Tattoo data set, the crowd workers were asked to annotate images from both ImageNet and PASCAL 07 data sets with at most 3 keywords. On average, each image is annotated by five different workers. Pairwise labels between images are derived based on whether two images share at least one common keyword.

3.3.2 Baseline and Parameter Selection

Studies in [60] have shown that the Bayesian approach performs significantly better than the ensemble clustering algorithm [113], and Non-negative Matrix Factorization (NMF) [84] in the crowdclustering setting. Hence, we use the Bayesian approach for crowdclustering as the baseline in our study.

Parameter C in (3.5) plays an important role in deciding the final similarity matrix. Since no ground truth information (true cluster labels) is available to determine C, we present a heuristic for estimating the value of C. We assume that the N objects to be clustered are roughly evenly distributed across clusters; a similar assumption was adopted in normalized cut algorithm [110]. Based on this assumption, we propose to choose a value of C that leads to the most balanced distribution of objects over different clusters. To this end, we measure the imbalance of data distribution over clusters by computing $\sum_{i,j=1}^{N} A'_{i,j} = \mathbf{1}^\top A' \mathbf{1}$, where $\mathbf{1}$ is a vector of all ones. Our heuristic is to choose a value for C that minimizes $\mathbf{1}^\top A' \mathbf{1}$. The

rationale behind the imbalance measurement $\mathbf{1}^\top A'\mathbf{1}$ is the following: Let $N_1, \cdots, N_r$ be the number of objects in the r clusters. Since $\mathbf{1}^\top A'\mathbf{1} = \sum_{k=1}^{r} N_k^2$ and $\sum_{k=1}^{r} N_k = N$, without any further constraint, the optimal solution that minimizes $\mathbf{1}^\top A'\mathbf{1}$ is $N_i = N/r, i = 1, \ldots, r$, the most balanced data distribution. Hence, $\mathbf{1}^\top A'\mathbf{1}$, to some degree, measures the imbalance of data distribution over clusters. The experimental results show that this heuristic works well. It usually helps us to find a good enough C among all the candidates.

We use normalized mutual information (NMI) and pairwise F-measure (PWF) to evaluate the clustering performance. Besides clustering accuracy, we also evaluate the efficiency of both algorithms by measuring their running time. The code of the baseline algorithm was provided by the authors of [60]. Both the baseline algorithm and the proposed algorithm were implemented in MATLAB and run on an Intel Xeon 2.40 GHz processor with 64.0 GB of main memory.

3.3.3 Experimental results with full annotations

To evaluate the clustering performance of the proposed algorithm, our first experiment is performed on the four image data sets using all the pairwise labels derived from the manual annotation process. For all data sets, we set d_0 to 0 and d_1 to 0.9. Two criteria are deployed in determining the value for d_1: d_1 should be large enough to ensure that most of the selected pairwise labels are consistent with the cluster assignments, and should be small enough to obtain enough number of entries with value 1 in the partially observed matrix $\tilde{A}$. Table 3.1 summarizes the clustering performance and running time (CPU time) of both algorithms.

Table 3.1 Clustering performance and running time of the proposed algorithm (i.e. matrix completion) and the baseline algorithm (i.e. Bayesian method) on four data sets

Datasets		Matrix Completion	Bayesian Method
Scenes Data Set	NMI	0.738	0.764
	PWF	0.584	0.618
	CPU time (s)	4.31×10^2	4.84×10^3
Tattoo Data Set	NMI	0.398	0.292
	PWF	0.595	0.524
	CPU time (s)	4.75×10^3	5.44×10^4
ImageNet data set	NMI	0.631	0.615
	PWF	0.734	0.718
	CPU time (s)	2.62×10^4	7.48×10^4
PASCAL 07 data set	NMI	0.394	0.388
	PWF	0.462	0.439
	CPU time (s)	9.23×10^3	2.36×10^4

We observed that for the Scenes data set, the proposed algorithm yields similar performance as the Bayesian crowdclustering algorithm but with much lower running time. For the Tattoo, ImageNet and PASCAL 07 data sets, the proposed algorithm outperforms the Bayesian crowdclustering algorithm in both accuracy and efficiency. The higher efficiency of the proposed algorithm is because that the proposed algorithm only needs to handle a subset of reliable pairwise labels while the Bayesian crowdclustering algorithm needs to explore all the pairwise labels derived from manual annotation. For example, for the Scenes data set, only less than 13% of image pairs satisfy the specified condition of "reliable pairs". The small percentage of reliable pairs results in a sparse matrix $\tilde{A}$, and consequently a high efficiency in solving the matrix completion problem in (3.4).

We also examine how well the conditions specified in our theoretical analysis are satisfied for the two image data sets. Besides the technical conditions that are difficult to verify, the most important condition used in our analysis is that a majority of the reliable pairwise

(a) Highway and Inside city (b) Bedroom and Kitchen (c) Mountain and Open country (d) Tall building and Street

Figure 3.5 Sample image pairs that are grouped into the same cluster by more than 50% of the workers but are assigned to different clusters according to the ground truth.

labels derived from manual annotation should be consistent with the cluster assignments (i.e. $m_1 - m_0 \geq O(N \log^2 N)$). We found that for the Scenes data set, 95% of the reliable pairwise labels identified by the proposed algorithm are consistent with the cluster assignments, and for the Tattoo data set, this percentage is 71%.

We finally evaluate the significance of the filtering step for the proposed algorithm. First, we observe that a large portion of pairwise labels derived from the manual annotation process are inconsistent with the cluster assignment. In particular, more than 80% of pairwise labels are inconsistent with the cluster assignment for the Scenes data set. Figure 3.5 shows some example image pairs that are grouped into the same cluster by more than 50% of the workers but belong to different clusters according to the ground truth.

To observe how the noisy labels affect the proposed algorithm, we fix the threshold d_0 to be 0, and vary the threshold d_1 used to determine the reliable pairwise labels from 0.1 to 0.9. Table 3.2 summarizes the clustering performance of the proposed algorithm for the Scenes data set with different thresholds and the percentage of resulting reliable pairwise labels that are consistent with the cluster assignments. Overall, we observe that the higher the percentage of consistent pairwise labels, the better the clustering performance.

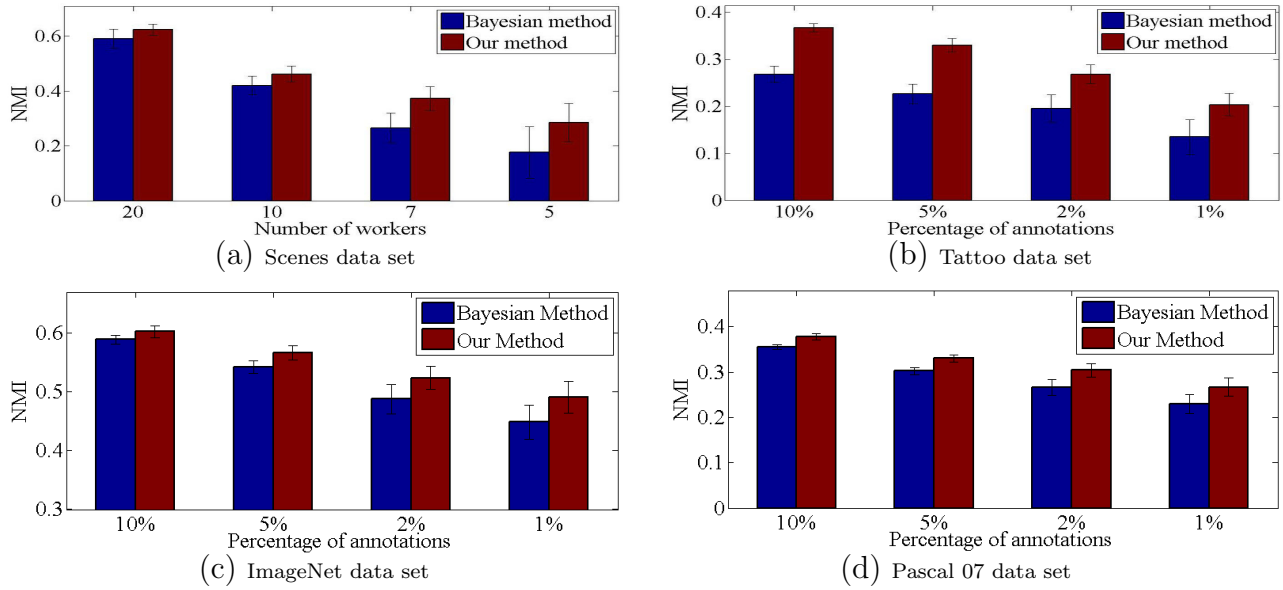
Table 3.2 Performance of the proposed clustering algorithm as a function of different threshold values and the percentage of 1 entries in the matrix $\tilde{A}$ that are consistent with the cluster assignments for the Scenes data set

Threshold d_1	0.1	0.3	0.5	0.7	0.9
Consistency percentage	18.02%	28.10%	35.53%	43.94%	61.79%
NMI	0.507	0.646	0.678	0.700	0.738
PWF	0.327	0.412	0.431	0.445	0.584

3.3.4 Experimental results with sampled annotations

The objective of the second experiment is to verify that the proposed algorithm is able to obtain an accurate clustering result even with a significantly smaller number of manual annotations. To this end, we use two different methods to sample the annotations: for the Scenes data set, we use the annotations provided by 20, 10, 7 and 5 randomly sampled workers, and for the remaining three data sets whose pairwise constraints are generated by keywords matching, we randomly sample 10%, 5%, 2% and 1% of all the annotations. Then we run both the baseline and the proposed algorithm on the sampled annotations. All the experiments in this study are repeated five times, and the performance averaged over the five trials is reported in Figure 3.6.

As expected, reducing the number of annotations deteriorates the clustering performance for both the algorithms. However, the proposed algorithm appears to be more robust and performs better than the baseline algorithm for all levels of random sampling. The robustness of the proposed algorithm can be attributed to the fact that according to our analysis, to perfectly recover the cluster assignment matrix, the proposed algorithm only requires a small number of reliable pairwise labels (i.e. $O(N \log^2 / N)$). In contrast, the Bayesian crowdclustering algorithm requires a large number of manual annotations to overcome the noisy labels



(a) Scenes data set (b) Tattoo data set (c) ImageNet data set (d) Pascal 07 data set

Figure 3.6 NMI values as a function of number of workers and percentage of annotations for four data sets

and to make reliable inference about the hidden factors used by different workers to group the images. As a consequence, we observe a significant reduction in the clustering performance of the Bayesian approach as the number of manual annotations is decreased.

3.4 Conclusion and Discussion

In this chapter, we present a matrix completion framework for crowdclustering. The key to the proposed algorithm is to identify a subset of data pairs with reliable pairwise labels. These reliable data pairs are used as the seed for a matrix completion algorithm to derive the full similarity matrix, which forms the foundation for data clustering. Currently, we identify these reliable data pairs based on the disagreement among workers, and as a result, a sufficient number of workers are needed to determine which data pairs are reliable. An

alternative approach is to improve the quality of manual annotations. Given that our matrix completion approach needs only a small number of high quality labels, we believe that combining appropriately designed incentive mechanisms with our matrix completion algorithm will lead to greatly improved performance. In [107], the authors discussed different incentive mechanisms to improve the quality of work submitted via HITs. In particular, they studied a number of incentive mechanisms and their affect on eliciting high quality work on Turk. They find that a mechanism based on accurately reporting peers' responses is the most effective in improving the performance of Turkers. As part of our future work, we plan to investigate the conjunction of appropriate incentive mechanisms with clustering algorithms for this problem.

Chapter 4

Semi-Crowdsourced Clustering by Distance Metric Learning

As discussed in Chapter 3, crowdclustering addresses the challenge of defining appropriate similarity measures between objects by using the manual annotations obtained through crowdsourcing. However, a key limitation of crowdclustering is that it can only cluster objects when their manual annotations are available. To address this limitation, in this chapter we propose a new approach for clustering, called *semi-crowdsourced clustering* that effectively combines the low-level features of objects with the manual annotations of a subset of the objects obtained via crowdsourcing. The key idea is to learn an appropriate similarity measure, based on the low-level features of objects and from the manual annotations of only a small portion of the data to be clustered. One difficulty in learning the pairwise similarity measure is that there is a large amount of noise in the manual annotations obtained via crowdsourcing. We address this difficulty by developing a metric learning algorithm based on the matrix completion method. Our empirical study with two real-world image data sets shows that the proposed algorithm outperforms state-of-the-art distance metric learning algorithms in both clustering accuracy and computational efficiency.

The remainder of the chapter is organized as follows: In Section 4.1, we describe the settings and introduce the motivation of the proposed semi-crowdsourced clustering framework. Section 4.2 presents the proposed semi-crowdsourced clustering algorithm and the related theoretical analysis. We summarize the results of our empirical studies in Section 4.3. Section 4.4 concludes the chapter.

4.1 Introduction

Despite the encouraging results obtained via crowdclustering, a main shortcoming of crowdclustering is that it can only cluster objects for which manual annotations are available, significantly limiting its application to large scale clustering problems. For instance, when clustering hundreds of thousands of objects, it is not feasible to have each object manually annotated by multiple workers. To address this limitation, we study the problem of **semi-crowdsourced clustering**, where given the annotations obtained through crowdsourcing for a small subset of the objects, the objective is to cluster the entire collection of objects. Figure 4.1 depicts the proposed framework. Given a set of N objects to be clustered, the objective of the proposed framework is to learn a pairwise similarity measure from the crowdsourced labels of n objects ($n \ll N$) and the feature representations of N objects to be clustered.

The key to semi-crowdsourced clustering is to define an appropriate similarity measure for the subset of objects that do not have manual annotations (i.e., $N-n$ objects). To this end, we propose to learn a similarity function, based on the object features, from the pairwise

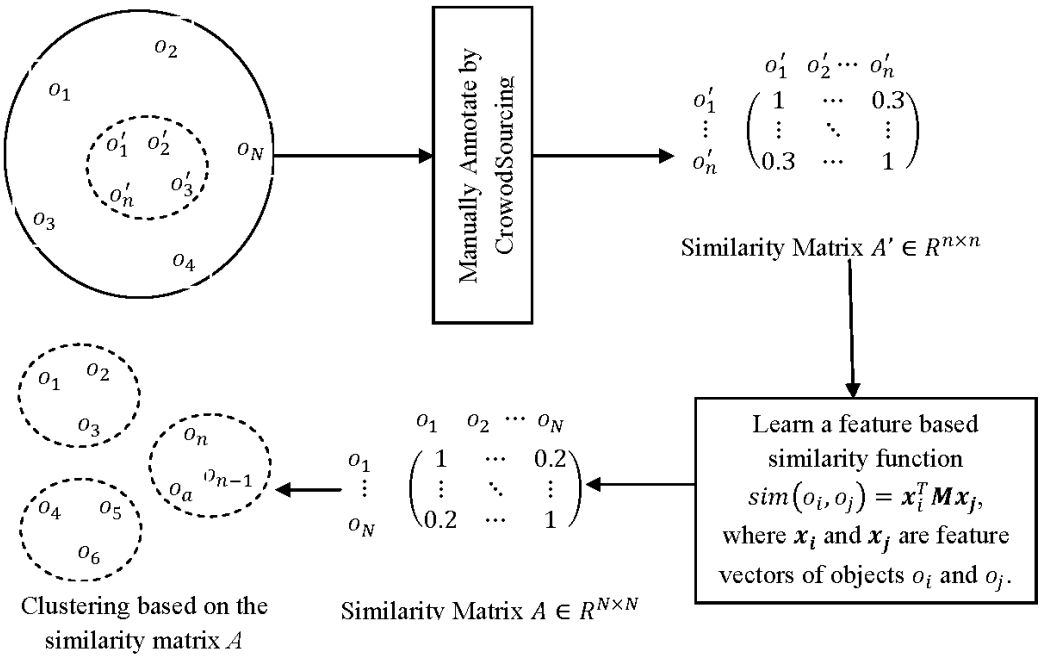


Figure 4.1 The proposed framework for semi-crowdsourced clustering. The given N objects $(o_1, o_2, \ldots, o_N)$ need to be clustered, but only a small subset of the N objects $(o'_1, o'_2, \cdots o'_n)$ have been annotated by crowdsourcing, $n \ll N$.

similarities derived from the manual annotations for the subset of n objects; we then apply the learned similarity function to compute the similarity between any two objects, and perform data clustering based on the computed similarities. In this study, for computational simplicity, we restrict ourselves to a linear similarity function, i.e. given two objects o_i and o_j and their feature representation $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, their similarity $sim(O_i, O_j)$ is given by $sim(O_i, O_j) = \mathbf{x}_i^\top M \mathbf{x}_j$, where $M \succeq 0$ is the learned distance metric.

Learning a linear similarity function from given pairwise similarities (sometimes referred to as pairwise constraints when similarities are binary) is known as distance metric learning, which has been reviewed in the Section 2.3.2. The key challenge of distance metric learning in semi-crowdsourced clustering arises due to the noise in the pairwise similarities obtained from manual annotations. According to our previous observation in Section 3.3, large dis-

agreements are often observed among human workers in specifying pairwise similarities. As a result, pairwise similarities based on the majority voting among human workers often disagree with the true cluster assignments of objects. As an example, we show in Section 3.3 that for the Scenes data set [46], more than 80% of the pairwise labels obtained from human workers are inconsistent with the true cluster assignment. This large noise in the pairwise similarities due to crowdsourcing could seriously misguide the distance metric learning and lead to a poor prediction performance, as already demonstrated in [66] as well as in our empirical study.

We propose a metric learning algorithm that explicitly addresses the presence of noise in pairwise similarities obtained via crowdsourcing. The proposed algorithm uses the matrix completion technique [20] to rectify the noisy pairwise similarities, and regression analysis to efficiently learn a distance metric from the restored pairwise similarities. More specifically, the proposed algorithm for clustering N objects consists of three components: (i) filtering noisy pairwise similarities for n objects by only keeping object pairs whose pairwise similarities are agreed by many workers (not majority of the workers). The result of the filtering step is a partially observed $n \times n$ similarity matrix ($n \ll N$) with most of its entries removed/unobserved; (ii) recovering the $n \times n$ similarity matrix from the partially observed entries by using the matrix completion algorithm; (iii) applying a regression algorithm to learn a distance metric from the recovered similarity matrix, and clustering the $N \times N$ pairwise similarities based on the learned distance metric. Figure 6.2 shows the basic steps of the proposed algorithm.

Compared to the existing approaches of distance metric learning [138], the proposed algorithm has the following three advantages: (i) by exploring the matrix completion technique,

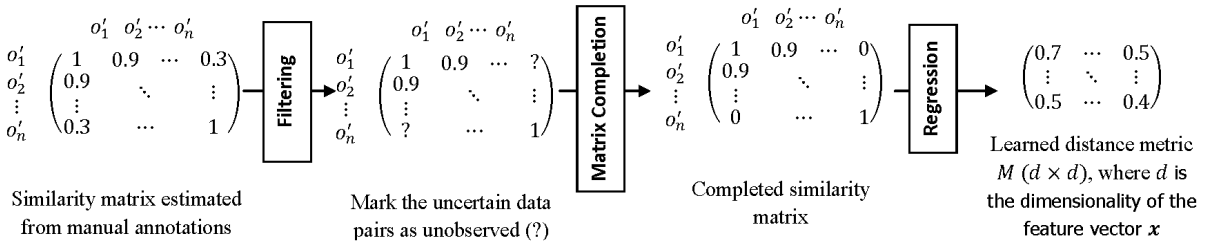


Figure 4.2 The proposed framework of learning a distance metric from noisy manual annotations

the proposed algorithm is robust to a large amount of noise in the pairwise similarities; (ii) by utilizing regression analysis, the proposed algorithm is computationally efficient and does not have to handle the positive semi-definite constraint, a key computational bottleneck for most distance metric learning algorithms; (iii) the learned distance metric, with high probability, is close to the optimal metric learned from the perfect or true similarities (i.e. similarity of 1 when two objects are in the same cluster and 0, otherwise) for arbitrarily large n.

4.2 Semi-Crowdsourced Clustering by Robust Distance Metric Learning

In this section, we first present the problem and a general framework for semi-crowdsourced clustering. We then describe the proposed algorithm for learning distance metric from a small set of noisy pairwise similarities that are derived from manual annotations.

4.2.1 Problem Definition and Framework

Let $\mathcal{D} = \{O_1, \ldots, O_N\}$ be the set of N objects to be clustered, and let $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ be their feature representation, where $\mathbf{x}_i \in \mathbb{R}^d$ is a vector of d dimensions. We randomly sample a subset of $n \ll N$ objects from the collection $\mathcal{D}$, denoted by $\widehat{\mathcal{D}} = \{\widehat{O}_1, \ldots, \widehat{O}_n\}$, and obtain their manual annotations by crowdsourcing. Let m be the number of HITs used by crowsourcing. Given the manual annotations collected from the k-th HIT, we define a similarity matrix $A^k \in \mathbb{R}^{n \times n}$ such that $A^k_{i,j} = 1$ if objects $\widehat{O}_i$ and $\widehat{O}_j$ share common annotations (i.e. share common annotated keywords or assigned to the same cluster by the worker), zero if they don't, and -1 if either of the two objects is not annotated by the kth HIT (i.e. unlabeled pair). Note that we only consider a binary similarity measure in this study because our goal is to *perfectly* reconstruct the ideal pairwise similarities based on the true cluster assignments (i.e. 1 when both objects are assigned to the same cluster and zero, otherwise). The objective of semi-crowdsourced clustering is to cluster all the N objects in $\mathcal{D}$ based on the features in X and the $m \times m$ similarity matrices $\{A^k\}_{k=1}^m$ for the objects in $\widehat{\mathcal{D}}$. Throughout this paper, we assume that the number of clusters, denoted by r, is given a priori[1].

To generalize the pairwise similarities from the subset $\widehat{\mathcal{D}}$ to the entire collection of objects $\mathcal{D}$, we propose to first learn a distance metric from the similarity matrices $\{A^k\}_{k=1}^m$, and then compute the pairwise similarity for all the N objects in $\mathcal{D}$ using the learned distance metric. The challenge is how to learn an appropriate distance metric from a set of similarity

[1]Similar to the Chapter 3, we may relax this requirement by estimating the number of clusters via some heuristic, e.g. considering the number of clusters as the rank of the completed matrix A.

matrices $\{A^k\}_{k=1}^m$. A straightforward approach is to combine multiple similarity matrices into a single similarity matrix by computing their average. More specifically, let $\widetilde{A} \in \mathbb{R}^{n \times n}$ be the average similarity matrix. We have

$$\widetilde{A}_{i,j} = \frac{1}{\sum_{k=1}^m I(A_{i,j}^k \geq 0)} \sum_{k=1}^m I(A_{i,j}^k \geq 0) A_{i,j}$$

where $A_{i,j}^k < 0$ indicates that the pair $(\widehat{O}_i, \widehat{O}_j)$ is not labeled by the kth HIT (i.e. either object $\widehat{O}_i$ or $\widehat{O}_j$ is not annotated by the kth worker) and $I(z)$ is an indicator function that outputs 1 when z is true and zero, otherwise. We then learn a distance metric M from $\widetilde{A}$. The main problem with this simple strategy is that due to the large disagreements among workers in determining the pairwise similarities, the average similarities do not correlate well with the true cluster assignments. In the next subsection, we develop an efficient and robust algorithm that learns a distance metric from a set of noisy similarity matrices.

4.2.2 Learning a Distance Metric from a Set of Noisy Similarity Matrices

As illustrated in Figure 6.2, the proposed algorithm consists of three steps, i.e. filtering step, matrix completion step and distance metric learning step. For the first two steps, namely the data preprocessing steps, we follow the idea proposed in Chapter 3.

Filtering step. To filter out the uncertain object pairs, we introduce two thresholds d_0 and $d_1 (1 \geq d_1 > d_0 \geq 0)$ into the average similarity matrix $\tilde{A}$. Since any similarity measure smaller than d_0 indicates that most workers put the corresponding object pair into different

clusters, we simply set it as 0. Similarly, we set the similarity measure larger than d_1 as 1. For object pairs with similarity measure in the range between d_0 and and d_1, they are treated as uncertain object pairs and are discarded (i.e. marked as unobserved) from the similarity matrix. The resulting partially observed similarity matrix A is given by

$$A_{i,j} = \begin{cases} 1 & \tilde{A}_{i,j} \in [d_1, 1] \\ 0 & \tilde{A}_{i,j} \in [0, d_0] \\ \text{unobserved} & \text{Otherwise} \end{cases} \tag{4.1}$$

We also define Δ as the set of observed entries in $A_{i,j}$

$$\Delta = \{(i,j) \in [N] \times [N] : \tilde{A}_{ij} \geq 0, \tilde{A}_{ij} \notin (d_0, d_1)\}$$

Matrix completion step. Since A is constructed from the partial clustering results generated by different workers, we expect some of the binary similarity measures in A to be incorrect. We introduce the matrix $E \in \mathbb{R}^{n \times n}$ to capture the incorrect entries in A. If A^* is the perfect similarity matrix, we have $\mathcal{P}_\Delta(A^* + E) = \mathcal{P}_\Delta(A)$, where $\mathcal{P}_\Delta$ outputs a matrix with $[\mathcal{P}_\Delta(B)]_{i,j} = B_{i,j}$ if $(i,j) \in \Delta$ and zero, otherwise. With appropriately chosen thresholds d_0 and d_1, we expect most of the observed entries in A to be correct and as a result, E to be a sparse matrix. To reconstruct the perfect similarity matrix A^* from A, following the matrix completion theory [20], we solve the following optimization problem

$$\begin{aligned} \min_{\widehat{A}, E} \quad & |\widehat{A}|_* + C|E|_1 \\ \text{s. t.} \quad & \mathcal{P}_\Delta(\widehat{A} + E) = \mathcal{P}_\Delta(A), \end{aligned} \tag{4.2}$$

where $|A|_*$ is the nuclear norm of matrix A and $|E|_1 = \sum_{i,j} |E_{i,j}|$ is the ℓ_1 norm of E. Using the facts that E is a sparse matrix and $\widehat{A}$ is of low rank [73], under the two assumptions made in Section 3.3, with a high probability, we have $A^* = \widehat{A}$, where $\widehat{A}$ is the optimal solution for (4.2).

Distance metric learning step. This step learns a distance metric from the completed similarity matrix $\widehat{A}$. A common problem shared by most distance metric learning algorithms is their high computational cost due to the constraint that a distance metric has to be positive semi-definite. In this study, we develop an efficient algorithm for distance metric learning that does not have to deal with the positive semi-definite constraint. Our algorithm is based on the key observation that with a high probability, the completed similarity matrix $\widehat{A}$ is positive semi-definite. This is because according to the Theorem 1, with a probability at least $1 - N^{-3}$, $\widehat{A} = YY^\top$, where $Y \in \{0,1\}^{N \times r}$ is the true cluster assignment. This property guarantees the resulting distance metric to be positive semi-definite.

The proposed distance metric learning algorithm is based on a standard regression algorithm [97]. Given the similarity matrix $\widehat{A}$, the optimal distance metric M is given by a regression problem

$$\min_{M \in \mathbb{R}^{d \times d}} \quad \widehat{\mathcal{L}}(M) = \sum_{i,j=1}^{n} (\widehat{\mathbf{x}}_i^\top M \widehat{\mathbf{x}}_j - \widehat{A}_{i,j})^2 = |\widehat{X}^\top M \widehat{X} - \widehat{A}|_F^2 \tag{4.3}$$

where $\widehat{\mathbf{x}}_i$ is the feature vector for the sampled object $\widehat{O}_i$ and $\widehat{X} = (\widehat{\mathbf{x}}_1, \ldots, \widehat{\mathbf{x}}_n)$. The optimal solution to (4.3), denoted by $\widehat{M}$, is given by

$$\widehat{M} = (\widehat{X}\widehat{X}^\top)^{-1} \widehat{X} \widehat{A} \widehat{X}^\top (\widehat{X}\widehat{X}^\top)^{-1} \tag{4.4}$$

where Z^{-1} is pseudo inverse of Z. It is straightforward to verify that $\widehat{M} \succeq 0$ if $\widehat{A} \succeq 0$.

Directly using the solution in (4.4) could result in the overfitting of similarity matrix $\widehat{A}$ because of the potential singularity of $\widehat{X}\widehat{X}^\top$. We address this challenge by a smoothing technique, i.e.

$$\widehat{M}_s = (\widehat{X}\widehat{X}^\top + \lambda m I)^{-1}\widehat{X}\widehat{A}\widehat{X}^\top(\widehat{X}\widehat{X}^\top + \lambda m I)^{-1} \tag{4.5}$$

where I is the identity matrix of size $d \times d$ and $\lambda > 0$ is a smoothing parameter used to address the overfitting and the curse of dimensionality.

4.2.3 Theoretical Analysis

We now state the theoretical property of $\widehat{M}_s$. Let $A(O_i, O_j)$ be the perfect similarity that outputs 1 when O_i and O_j belong to the same cluster and zero, otherwise. It is straightforward to see that $A(O_i, O_j) = \mathbf{y}_i^\top \mathbf{y}_j$, where $\mathbf{y}_i \in \{0, 1\}^r$ is the cluster assignment for object O_i. To learn an ideal distance metric from the perfect similarity measure $A(O_i, O_j)$, we generalize the regression problem in (4.3) as follows

$$\min_{M \in \mathbb{R}^{d \times d}} \quad \mathcal{L}(M) = \mathrm{E}_{\mathbf{x}_i, \mathbf{x}_j}\left[(\mathbf{x}_i^\top M \mathbf{x}_j - A(O_i, O_j))^2\right] \tag{4.6}$$

The solution to (4.6) is given by $M = C_X^{-1} B B^\top C_X^{-1}$, where $C_X = \mathrm{E}_{\mathbf{x}_i}[\mathbf{x}_i \mathbf{x}_i^\top]$ and $B = \mathrm{E}_{\mathbf{x}_i}[\mathbf{x}_i \mathbf{y}_i^\top]$. Let M_s be the smoothed version of the ideal distance metric M, i.e. $M = (C_X + \lambda I)^{-1} B B^\top (C_X + \lambda I)^{-1}$. The following theorem shows that with a high probability, the difference between $\widehat{M}_s$ and M_s is small if both λ and n are not too small.

Theorem 2. *Assume* $|\mathbf{x}|_2 \leq 1$ *for the feature representation of any object. Then, with a probability* $1 - 3n^{-3}$*, we have*

$$|M_s - \widehat{M}_s|_2 = O\left(\frac{\ln n}{\lambda^2\sqrt{n}}\right)$$

where $|Z|_2$ *stands for the spectral norm of matrix* Z*.*

Proof. To prove Theorem 2, we need the following theorem for matrix concentration.

Lemma 1. *(Lemma 2 from [111]) Let* $\mathcal{H}$ *be a Hilbert space and* ξ *be a random variable on* (Z, ρ) *with values in* $\mathcal{H}$*. Assume* $\|\xi\| \leq M < \infty$ *almost surely. Denote* $\sigma^2(\xi) = \mathrm{E}(\|\xi\|^2)$*. Let* $\{z_i\}_{i=1}^m$ *be independent random drawers of* ρ*. For any* $0 < \delta < 1$*, with confidence* $1 - \delta$*,*

$$\left\| \frac{1}{m} \sum_{i=1}^{m} (\xi_i - \mathrm{E}[\xi_i]) \right\| \leq \frac{4M \ln(2/\delta)}{\sqrt{m}}$$

Using the assumption that $|\mathbf{x}|_2 \leq 1$ and Lemma 1, we have, with a probability $1 - N^{-3}$,

$$\left| \frac{1}{m} \widehat{X}\widehat{X}^\top - C_X \right|_2 \leq \frac{12 \ln n}{\sqrt{n}}$$

and therefore

$$\left| \left(\frac{1}{m} \widehat{X}\widehat{X}^\top + \lambda I \right)^{-1} - (C_X + \lambda I)^{-1} \right|_2 \leq \frac{12 \ln n}{\lambda\sqrt{n}}$$

Second, according to Theorem 1, with a probability $1 - N^{-3}$, we have $\widehat{A} = YY^\top$ and therefore $\widehat{X}\widehat{A}\widehat{X}^\top = \widehat{X}YY^\top\widehat{X}^\top$. Again, using the matrix concentration theory, we have,

with a probability $1 - N^{-3}$,

$$\left|\frac{1}{m}\widehat{X}Y - B\right|_2 \leq \frac{12\ln n}{\sqrt{n}}$$

Finally, we rewrite $|M_s - \widehat{M}_s|_2$ as

$$\begin{aligned}
& \|M_s - \widehat{M}_s\|_2 \\
\leq\ & \left|M_s - \left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1} BB^\top C_X\right|_2 + \\
& \left|\left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1} BB^\top C_X - \left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right) BB^\top \left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1}\right|_2 + \\
& \left|\left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1} BB^\top \left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1} - \left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1} \frac{\widehat{X}Y}{m} B^\top \left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1}\right| + \\
& \left|\left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1} \frac{\widehat{X}Y}{m} B^\top \left(\frac{1}{m}\widehat{X}\widehat{X}^\top + \lambda I\right)^{-1} - \widehat{M}_s\right|
\end{aligned}$$

It is easy to see that with a probability $1 - 3n^{-3}$, each term on the right hand side of the above inequality is bounded by $\frac{12\ln n}{\lambda^2\sqrt{n}}$, leading to the result of the theorem. □

Given the learned distance metric $\widehat{M}_s$, we construct a similarity matrix $S = X^\top \widehat{M}_s X$ and then apply a spectral clustering algorithm [110] to compute the final data partition for N objects.

4.3 Experiments

In this section, we demonstrate empirically that the proposed semi-crowdsourced clustering algorithm is both effective and efficient.

4.3.1 Data Sets

Two real-world image data sets are used in our experiments: (i) *ImageNet data set* is a subset of the larger ImageNet database [39]. The subset contains $6,408$ images belonging to 7 categories: *tractor*, *horse cart*, *bench*, *blackberry*, *violin*, *saxophone*, and *hammer*. (ii) *PASCAL 07 data set* is a subset of the PASCAL Visual Object Classes Challenge 2007 database [45]. The subset contains $2,989$ images belonging to five classes: *car*, *dog*, *chair*, *cat* and *bird*. We choose these specific image categories because they yield relatively low classification performance in ImageNet competition and PASCAL VOC Challenge, indicating that it could be difficult to cluster these images using low level features without side information. The image features for these datasets were downloaded from the homepages of the ImageNet database[1] and the research group of Learning and Recognition in Vision (LEAR)[2], respectively.

To perform crowdlabeling, we follow [141], and ask human workers to annotate images with keywords of their choice in each HIT. A total of 249 and 332 workers were employed using the Amazon's Mechanical Turk [70] to annotate images from ImageNet and PASCAL datasets, respectively. On average, each image is annotated by five different workers, with three keywords from each individual worker. For every HIT, the pairwise similarity between two images (i.e. $A^k_{i,j}$ used in Section 4.2.1) is set to 1 if the two images share at least one common annotated keyword and zero, otherwise[3]. We note that the crowdsouced annotations of these two data sets are also used to evaluate our crowdclustering algorithm proposed in Chapter 3.

[1] `http://www.image-net.org/download-features`

[2] `http://lear.inrialpes.fr/people/guillaumin/data.php`

[3] We tried several other similarity measures (e.g. cosine similarity measure and tf.idf weighting) and found that none of them yielded better performance than the simple similarity measure used in this work

4.3.2 Baselines

Two baseline methods are used as reference points in our study: (a) the **Base** method that clusters images directly using image features without distance metric learning, and (b) the **Raw** method that runs the proposed algorithm against the average similarity matrix $\widetilde{A}$ without filtering and matrix completion steps. The comparison to the **Base** method allows us to examine the effect of distance metric learning in semi-crowdsourced clustering, and the comparison to the **Raw** method reveals the effect of filtering and matrix completion steps in distance metric learning.

We compare the proposed algorithm for distance metric learning to the following five state-of-the-art distance metric learning algorithms: (a) **GDM**, the global distance metric learning algorithm [134], (b) **RCA**, the relevant component analysis [6], (c) **DCA**, the discriminative component analysis [63], (d) **ITML**, the information theoretic metric learning algorithm [38], and (e) **LMNN**, the large margin nearest neighbor classifier [130]. Some of the other state-of-the-art distance metric learning algorithms (e.g. the neighborhood components analysis (NCA) [58]) were excluded from the comparison because they can only work with class assignments, instead of pairwise similarities, and therefore are not applicable in our case. The code for the baseline algorithms was provided by their respective authors (In LMNN, Principal Component Analysis (PCA) is used at first to reduce the data to lower dimensions). For a fair comparison, all distance metric learning algorithms are applied to the pairwise constraints derived from $\widehat{A}$, the $n \times n$ pairwise similarity matrix reconstructed by the matrix completion algorithm. We refer to the proposed distance metric learning algorithm as **Regression based Distance Metric Learning**, or **RDML** for short, and the proposed

semi-crowdsourced clustering algorithm as **Semi-Crowd**.

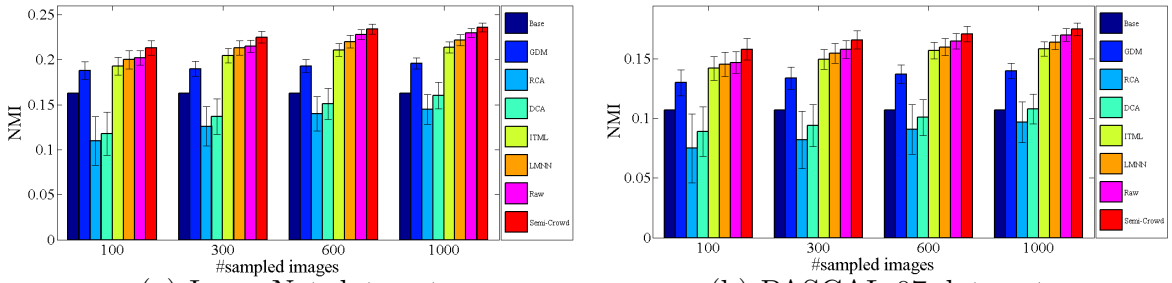
4.3.3 Parameter Selection and Evaluation

Similar to the Section 3.4, two criteria are used in determining the values for d_0 and d_1 in (4.1). First, d_0 (d_1) should be small (large) enough to ensure that most of the retained pairwise similarities are consistent with the cluster assignments. Second, d_0 (d_1) should be large (small) enough to obtain a sufficient number of observed entries in the partially observed matrix A. For both data sets, we set d_0 to 0 and d_1 to 0.8. Besides, we follow the heuristic proposed in Section 3.4 to determine the parameter C in (4.2), which is selected to generate balanced clustering results. Parameter λ in (4.5) is set to 1. We varied λ from 0.5 to 5 and found that the clustering results essentially remain unchanged.

Normalized mutual information (NMI for short) is used to measure the coherence between the inferred clustering and the ground truth categorization. The number of sampled images is varied from 100, 300, 600 to $1,000$. All the experiments are performed on a PC with Intel Xeon 2.40 GHz processor and 16.0 GB of main memory. Each experiment is repeated five times, and the performance averaged over the five trials is reported.

4.3.4 Experimental Results

First, we examine the effect of distance metric learning algorithm on semi-crowdsourced clustering. Figure 6.4 compares the clustering performance with six different metric learning algorithms with that of the Base method that does not learn a distance metric. We observed

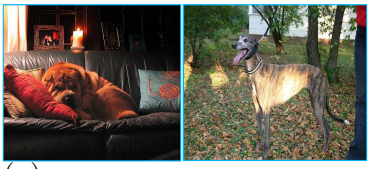


(a) ImageNet data set (b) PASCAL 07 data set

Figure 4.3 NMI vs. no. of sampled images (n) used in crowdlabeling.

that four of the distance metric learning algorithms (i.e. GDM, ITML, LMNN and the proposed RDML) outperform the Base method, while RCA and DCA fail to improve the clustering performance of Base. We conjecture that the failure of RCA and DCA methods is due to their sensitivity to the noisy pairwise similarities. In fact, RCA and DCA can yield better performance than the Base method if all the pairwise similarities are consistent with the cluster assignments. Compared to all the baseline distance metric learning algorithms, **RDML**, the proposed distance metric learning algorithm, yields the best clustering results for both the data sets and for all values of n (i.e. the number of annotated images) considered here. Furthermore, the performance of **RDML** gradually stabilizes as the number of sampled images increases. This is consistent with our theoretical analysis in Theorem 2, and implies that only a modest number of annotated images is needed by the proposed algorithm to learn an appropriate distance metric. This observation is particularly useful for crowdclustering as it is expensive to reliably label a very large number of images. Figure 4.4 shows some example image pairs for which the Base method fails to make correct cluster assignments, but the proposed **RDML** method successfully corrects these mistakes with the learned distance metric.

Our next experiment evaluates the impact of filtering and matrix completion steps. In Fig-



(a) Two images incorrectly placed in different clusters by the Base method (similarity 0.16) but correctly grouped into the same cluster by the proposed method (similarity 0.66).

(b) Two images incorrectly placed in different clusters by the Base method (similarity 0.31) but correctly grouped into the same cluster by the proposed method (similarity 0.85)

(c) Two images incorrectly grouped into the same cluster by the Base method (similarity 0.72) but correctly clustered to different clusters by the proposed method (similarity 0.22)

Figure 4.4 Sample image pairs that are incorrectly clustered by the Base method but correctly clustered by the proposed method (the similarity of our method is based on the normalized distance metric $\widehat{M}_s$).

ure 6.4, we compare the clustering results of the proposed algorithm for semi-crowdsourced clustering (i.e. Filtering+Matrix-Completion+RDML) to the Raw method that runs the proposed distance metric algorithm RDML without the filtering and matrix completion steps. Based on these experiments, we can make the following observations: (i) the proposed distance metric learning algorithms performs better than the Raw method, particularly when the number of annotated images is small; (ii) the gap between the proposed semi-crowdsourced clustering method and the Raw method decreases as the sample size increases. These results indicate the importance of filtering and matrix completion steps for the crowdsourced data in semi-crowdsourced clustering. Finally, it is interesting to observe that the Raw method still outperforms all the baseline methods, which further verifies the effectiveness of the proposed algorithm for distance metric learning.

Finally, we evaluate the computational efficiency of the proposed distance metric learning algorithm. Table 4.1 shows that the proposed distance metric learning algorithm is significantly more efficient than the baseline approaches evaluated here. The last row of Table 4.1 indicates the run time for the matrix completion step. Since all the distance metric learning algorithms

Table 4.1 CPU time (in s) for learning the distance metrics.

CPU time (s)	ImageNet Data Set				PASCAL 07 Data Set			
Sample sizes (n)	100	300	600	1,000	100	300	600	1,000
RDML (proposed)	4.2	6.3	8.0	11.2	27.4	34.2	41.7	47.3
GDM [134]	11384	14706	18140	25155	26346	36795	44237	53468
LMNN [130]	59.8	157	330	629	55.1	124	277	527
ITML [38]	2128	2376	2692	3081	5311	5721	6104	6653
DCA [63]	8.5	9.2	14.5	20.7	51.2	64.1	72.7	82.3
RCA [6]	9.7	13.5	18.6	23.6	71.4	92.7	103	122
Matrix Completion	12.4	74.2	536	1916	12.8	86.6	615	1873

are applied to the similarity matrix recovered by the matrix completion algorithm, the computational cost of matrix completion is shared by all distance metric learning algorithms used in our evaluation. One interesting observation comes from a comparison of Table 4.1 and Table 3.1, which summarizes the running time of doing standard crowdclustering. We observe that, to cluster the same ImageNet Data Set, the proposed semi-crowdsourced clustering algorithm is more than $1,000$ times more efficient than the crowdclustering algorithm proposed in Chapter 3.

4.4 Conclusions

In this chapter, we present a semi-crowdsourced clustering framework that effectively combines the low-level features of objects with the manual annotations of a subset of the objects obtained via crowdsourcing. The proposed framework overcomes the limitation of the classical crowdclustering problem, which can only cluster objects when their manual annotations are available. In addition, the proposed semi-crowdsourced clustering method provides a much more efficient way to solve the crowdclustering problem, comparing to the approach

proposed in chapter 3. Furthermore, our work also addresses the challenge of learning a reliable distance metric from noisy pairwise constraints. Although many studies on distance metric learning have been reported, one limitation of these earlier studies is that they can only work with a relatively small number (typically less than 30%) of noisy pairwise constraints. In contrast, the proposed distance metric learning approach can handle a significantly larger percentage of pairwise similarities (as many as 80%) that are inconsistent with the true cluster assignments.

Chapter 5

Semi-supervised Clustering by Input Pattern Assisted Pairwise Similarity Matrix Completion

In this chapter, we focus on the semi-supervised clustering problem. Many semi-supervised clustering algorithms have been proposed to improve the clustering accuracy by effectively exploring the available side information that is usually in the form of pairwise constraints. However, there are two main shortcomings of the existing semi-supervised clustering algorithms. First, they have to deal with non-convex optimization problems, leading to clustering results that are sensitive to the initialization. Second, none of these algorithms is equipped with theoretical guarantee regarding the clustering performance. We address these limitations by developing a framework for semi-supervised clustering based on *input pattern assisted matrix completion*. The key idea is to cast clustering into a matrix completion problem, and solve it efficiently by exploiting the correlation between input patterns and cluster assignments. Our analysis shows that under appropriate conditions, only $O(\log N)$ pairwise constraints are needed to accurately recover the true cluster partition of N objects.

The remainder of the chapter is organized as follows: In Section 5.1, we describe the background and introduce the motivation of the proposed semi-supervised clustering framework. Section 5.2 presents the proposed framework for semi-supervised clustering and efficient computational algorithm. Theoretical analysis for the proposed algorithm for semi-supervised learning is presented in Section 5.3. We summarize the results of our empirical studies in Section 5.4 and Section 5.5 concludes with the discussion about the relationship between the proposed semi-supervised clustering approach and the semi-crowdsourced clustering approach proposed in Chapter 4.

5.1 Introduction

The objective of semi-supervised clustering algorithms is to search for the optimal data partition that is consistent with both the given pairwise constraints and the input data points to be clustered. Despite the progress, there are two main shortcomings with the available semi-supervised clustering algorithms. First, most semi-supervised clustering algorithms have to deal with non-convex optimization problems, leading to clustering results that are only locally optimal and sensitive to the initialization. Second, although many computational algorithms have been proposed for semi-supervised learning, none of them is equipped with a theoretical guarantee on clustering performance. In particular, it is unknown how the clustering performance is improved with increasing number of pairwise constraints, an issue that is usually referred to as *sample complexity* in supervised learning [7].

In this chapter, we aim to address these limitations by developing a new framework for semi-

supervised learning based on the theory of matrix completion [15]. The proposed framework aims to reconstruct the pairwise similarity matrix, that gives 1 for any two data points in the same cluster and 0 otherwise, based on the given constraints and the input patterns of the objects to be clustered. The proposed framework results in a convex optimization problem and, consequentially, globally optimal solutions. More importantly, the proposed work is equipped with a strong theoretical guarantee: with a high probability, the proposed algorithm can accurately recover the true data partition provided (i) the cluster membership vectors can be well approximated by the top singular vectors of the data matrix, and (ii) the number of pairwise constraints is sufficiently large. In particular, we show that under appropriate conditions, the true data partition can be *perfectly* recovered by the proposed algorithm with $O(rk \log N)$ pairwise constraints, where N is the number of data points to be clustered, r is the number of clusters, and k is the number of singular vectors used to approximate the cluster memberships. The logarithmic dependence on N makes the proposed algorithm particularly suitable for large-scale data clustering problem.

5.2 Semi-supervised Clustering by Input Pattern Assisted Matrix Completion

In this section, we first present a matrix completion based framework for semi-supervised clustering. We then present the proposed algorithm for semi-supervised clustering.

5.2.1 A Matrix Completion Framework for Semi-supervised Clustering

Let $\mathcal{D} = \{O_1, \ldots, O_N\}$ be the set of N objects to be clustered, and let $X = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ be their feature representation, where $\mathbf{x}_i \in \mathbb{R}^d$ is a vector of d dimensions. Let $\mathcal{M}$ denote the set of must-link constraints where $(i, j) \in \mathcal{M}$ implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in the same cluster, and $\mathcal{C}$ denote the set of cannot-link constraints, where $(i, j) \in \mathcal{C}$ implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to different clusters. For the convenience of presentation, we also define set $\Omega = \mathcal{M} \cup \mathcal{C}$ to include all the pairwise constraints. Let r be the number of clusters, and $N_{\min}$ be the size of the smallest cluster. The objective of semi-supervised clustering is to partition N data points into r clusters that are consistent with (i) the pairwise constraints in $\mathcal{M}$ and $\mathcal{C}$, and (ii) the data matrix X such that data points with similar input patterns are put into the same cluster.

Let $\mathbf{u}_i \in \{0, 1\}^N$ be the membership vector of the i-th cluster, where $u_{i,j} = 1$ if $\mathbf{x}_j$ is assigned to the i-th cluster and zero, otherwise. Define the pairwise similarity matrix $S \in \{0, +1\}$ as

$$S = \sum_{i=1}^{r} \mathbf{u}_i \mathbf{u}_i^\top$$

Evidently, $S_{i,j} = 1$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned to the same cluster, and zero, otherwise. It is easy to verify that the rank of matrix S is r. The given must-links in $\mathcal{M}$ and cannot-links in $\mathcal{C}$ provide partial observations for M, i.e. $S_{i,j} = 1$ if $(i, j) \in \mathcal{M}$ and $S_{i,j} = 0$ if $(i, j) \in \mathcal{C}$. Since finding the best data partition is equivalent to recovering the binary similarity matrix S, following [73, 141], we cast the semi-supervised clustering problem into a matrix completion problem, i.e. filling out the missing entries in binary similarity matrix S based on the pairwise

constraints in $\mathcal{M}$ and $\mathcal{C}$ (i.e. the partial observations of S) and the data matrix X.

Similar to the standard theory for matrix completion [15], we can accurately recover the binary similarity matrix S because S is of low rank. We, however, note that the matrix completion problem discussed in this work is different from the previous studies of using matrix completion for clustering [73, 141] in that we aim to complete the binary similarity matrix S by utilizing both the observed entries in S and the input patterns in X. It will be shown later, both theoretically and empirically, that by effectively exploring the input patterns in X, the proposed algorithm is able to reduce the sample complexity for matrix completion from $O(N[\log N]^2)$ to $O(\log N)$, making it possible to apply the proposed algorithm to cluster very large data sets.

5.2.2 Input Pattern Assisted Matrix Completion

In this subsection, we first present input pattern assisted matrix completion for semi-supervised clustering. We then describe an efficient algorithm for solving the related convex optimization problem.

In the standard matrix completion theory [15], to reconstruct a matrix Q of size $N \times N$ from a subset of observed entries in $\Delta \subseteq [N] \times [N]$, we solve the following optimization problem

$$\min_{Q \in \mathbb{R}^{N \times N}} |Q|_{tr} \text{ s. t. } \mathcal{P}_{\Delta}(P) = \mathcal{P}_{\Delta}(S) \tag{5.1}$$

where $|\cdot|_{tr}$ is the trace norm, and $\mathcal{R}_{\Delta}(S) : \mathbb{R}^{N \times N} \mapsto \mathbb{R}^{N \times N}$ is a linear operator that maps

a matrix S to a new matrix $\mathcal{P}_\Delta(S)$ given by

$$[\mathcal{P}_\Delta(S)]_{i,j} = \begin{cases} S_{i,j} & (i,j) \in \Delta \\ 0 & (i,j) \notin \Delta \end{cases}$$

According to [15], with a high probability, matrix Q can be perfectly recovered by solving the optimization problem in (6.6) if the number of observed entries in Δ is $O(\mu(Q)^2 r(Q) N[\log N]^2)$, where $r(Q)$ is the rank of Q and $\mu(Q)$ is the coherence measure of Q. In the case of binary similarity matrix S, it is easy to verify that the coherence measure $\mu(S)$ is bounded by $\sqrt{N/[N_{\min} r]}$ and the rank of S equals to the number of clusters r. As a result, the number of pairwise constraints required for perfectly recovering the binary similarity matrix S is $O(\kappa N[\log N]^2)$, where $\kappa = N/N_{\min}$. When data points are evenly distributed over clusters, we observe that the number of pairwise constraints required by matrix completion increases at least linearly in the number of data points to be clustered, making it unscalable to large data sets.

We address this limitation by developing a matrix completion approach that explicitly incorporates the data matrix X into the matrix completion process. Let $Z = (\mathbf{z}_1, \ldots, \mathbf{z}_k)$ include the first k left singular vectors of X, where $k \geq r$. We make the following crucial assumption about the relationship between X and S:

A3 $\quad \{\mathbf{u}_i\}_{i=1}^r$ lie in the subspace spanned by $\{\mathbf{z}_i\}_{i=1}^k$,

a similar assumption used by the spectral clustering algorithm [101]. Using assumption **A3**, we can write S as $S = ZMZ^\top$, where $M \in \mathbb{R}^{k \times k}$. Following the theory of matrix completion,

we obtain the optimal M by solving the following optimization problem:

$$\begin{aligned} \min_{M \in \mathbb{R}^{k \times k}} \quad & |M|_{tr} \\ \text{s. t.} \quad & \mathcal{P}_\Omega(ZMZ^\top) = \mathcal{P}_\Omega(S) \end{aligned} \tag{5.2}$$

where $\Omega \subseteq [N] \times [N]$ includes all the observed entries in S derived from the pairwise constraints in $\mathcal{M}$ and $\mathcal{C}$.

The following theorem shows the perfect recovery result for (6.7).

Theorem 3. *Let $\mu(Z)$ be the coherence measure for matrix Z given by*

$$\mu(Z) = \max_{1 \le i \le N} \frac{n}{k} |[ZZ^\top]_{i,i}|^2 \tag{5.3}$$

Define

$$\mu_0 = \max\left(\mu(Z), \sqrt{\frac{N}{rN_{\min}}}\right). \tag{5.4}$$

For fixed $\beta > 2$, define a and B as

$$a = \frac{1}{2}\left(1 + \log_2 k - \log_2 r\right) \tag{5.5}$$

$$B = \frac{512\beta}{3}\mu_0 rk \ln N \tag{5.6}$$

Then, under assumption **A3** *with a probability $1 - 4(a+1)N^{-\beta+1} - 2aN^{-\beta+2}$, $M_* = Z^\top SZ$ is the unique optimizer to (6.7) provided $|\Omega| \ge aB$.*

Remark: Compared to the standard matrix completion theory, the sample complexity of

input pattern assisted matrix completion is reduced from $O(rN[\log N]^2)$ to $O(k \log N \log N)$ if $\mu_0 = O(1)$. Thus, if $k = \Omega(r)$ and the number of clusters r is small, Theorem 3 implies that $O(\log N)$ pairwise constraints are needed in order to obtain the perfect clustering result, provided assumption **A3** holds and the coherence measure μ_0 is small.

Evidently, **A3** is a strong assumption that usually does not hold in real world applications. We thus relax this assumption by assuming that the cluster membership vectors $\{\mathbf{u}_i\}_{i=1}^r$ can be well approximated by the top k singular vectors of X. More specifically, we define the projection operator P_k as $P_k = ZZ^\top$, and the projection errors for the cluster membership vectors as

$$\mathcal{E}^2 = \max_{1 \le i \le r} \frac{1}{N^2} \|\mathbf{u}_i - P_k \mathbf{u}_i\|_F^2 \tag{5.7}$$

Instead of assuming $\mathcal{E} = 0$ as assumption **A3**, we assume that $\mathcal{E}$ is small enough to allow for an accurate recovery of the binary similarity matrix S. Under this assumption, we modify the optimization problem in (6.7) as follows

$$\min_{M \in \mathbb{R}^{k \times k}} \ |M|_{tr} + \frac{C}{2} \left\| \mathcal{P}_\Omega(ZMZ^\top) - \mathcal{P}_\Omega(S) \right\|_F^2 \tag{5.8}$$

where parameter $C > 0$ is introduced to balance the tradeoff between finding the low rank matrix M and fitting the observed pairwise constraints. The following theorem shows that the binary similarity matrix S can be accurately recovered by (6.8) if (i) the approximation error $\mathcal{E}$ is small and (ii) $|\Omega|$, the number of pairwise constraints, is sufficiently large.

Theorem 4. *Let $\widehat{M}$ be the optimal solution to (6.8) and $\widehat{S} = Z\widehat{M}Z$ be the reconstructed similarity matrix. For a fixed $\beta > 2$, with a probability $1 - 4(a+1)N^{-\beta+1} - 2aN^{-\beta+2}$, we*

have

$$\|\widehat{S} - S\|_F \leq \nu(k, r)\mathcal{E}$$

where

$$\nu(k, r) = 6\left(\sqrt{2k} + 4\sqrt{r}\right)\left(3 + \sqrt{r}\right)$$

provided $|\Omega| \geq aB$ *and* $C \geq 1/[\sqrt{r}\mathcal{E}]$

As indicated by the above theorem, with a sufficiently large number of pairwise constraints, we have $\|\widehat{S} - \mathcal{S}\|_F \propto \mathcal{E}$, implying a small difference between $\widehat{S}$ and S when the cluster membership vectors can be well approximated by the top k singular vectors of X.

Let $\widehat{M}$ be the optimal solution for (6.8). The estimated binary similarity matrix is given by $\widehat{S} = Z\widehat{M}Z^\top$. Since $|\widehat{S} - S|_F$ is small and the eigenvectors of S correspond to the cluster membership vectors, we expect the first r eigenvectors of $\widehat{S}$ reveal the clustering structure of the data. As a result, we apply the spectral clustering algorithm to find the best data partition, i.e. we first compute the top r eigenvectors of $\widehat{S}$, and then run the k-means algorithm over the computed eigenvectors. To improve the computational efficiency, we apply the spectral clustering algorithm proposed in [25] that reduces computational cost by the matrix sparsification technique [122] and the Nystrom approximation [51].

We finally discuss how to efficiently solve the optimization problem in (6.8). We exploit the fast stochastic subgradient descent (FSGD) method developed in [4]. Define

$$\mathcal{L}(M) = \frac{C}{2}\left\|\mathcal{P}_\Omega(ZMZ^\top) - \mathcal{P}_\Omega(S)\right\|_F^2.$$

Algorithm 1 Efficient Stochastic Subgradient Descent for Solving the Optimization Problem (6.8)

Input:
- $Z \in \mathbb{R}^{N\times k}$: first k left singular vectors of X
- $C > 0$: loss function parameter
- r: number of clusters
- T: number of iterations
- η_t: step size

Initialization: $U_0 = \mathbf{0}_{k\times r},\ \Sigma_0 = \mathbf{0}_{r\times r},\ V_0 = \mathbf{0}_{k\times r}$
for $t = 0, \ldots, T-1$ **do**
  Generate a $k \times r$ probing matrix H
  Set $\hat{U}_{t+1} = [U_t\Sigma_t, B_t]$, where $B_t = (U_t V_t^\top + C \cdot Z^\top(\mathcal{R}_\Omega(ZMZ^\top - S))Z)H$.
  Set $\hat{V}_{t+1} = [V_t \quad -\eta_t H]$
  QR factorization of $\hat{U}_{t+1}$: $\hat{U}_{t+1} = Q_U R_U$
  QR factorization of $\hat{V}_{t+1}$: $\hat{V}_{t+1} = Q_V R_V$
  Compute $K = R_U R_V^\top$
  SVD decomposition of K: $K = \tilde{M}\bar{\Sigma}_{t+1}\tilde{N}^\top$
  Set $\bar{U}_{t+1} = Q_U\tilde{M}$ and $\bar{V}_{t+1} = Q_V\tilde{N}$
  $U_{t+1} = \bar{U}_{t+1}(1{:}k, 1{:}r)$
  $\Sigma_{t+1} = \bar{\Sigma}_{t+1}(1{:}r, 1{:}r)$
  $V_{t+1} = \bar{V}_{t+1}(1{:}k, 1{:}r)$
  $M^{(t+1)} = \Pi(U_{t+1}\Sigma_{t+1}V_{t+1}^\top)$
end for

At each iteration, the proposed algorithm samples a subset of rows from the binary similarity matrix S by introducing a probing matrix H. It then computes an unbiased estimate of the gradient $\nabla\mathcal{L}(M_t)$, denoted by $\tilde{\nabla}\mathcal{L}(M_t)$, based on the sampled rows. Given the unbiased estimate of gradient, solution M_t is updated by $M_{t+1} = \Pi\left(M'_{t+1} = M_t - \eta\tilde{\nabla}\mathcal{L}(M_t)\right)$. Here, $\Pi(A)$ is a soft thresholding function and is defined as $\Pi(A) = \sum_{i=1}^{r}\max(\lambda_i - 1, 0)\mathbf{a}_i\mathbf{a}_i^\top$, where $(\mathbf{a}_i, \lambda_i), i = 1, \ldots, r$ are the top r eigenvectors and eigenvalues of A. Algorithm 1 shows the detailed steps of the proposed algorithm, where the notation $U(1{:}k, 1{:}r)$ represents the sub-matrix of U that includes the first k rows and the first r columns of U.

5.3 Theoretical Analysis

In this analysis, we will focus on the result for the noisy case, namely where the cluster membership vectors can be well approximated by the top k singular vectors of X although they do not lie in the subspace spanned by the top k singular vectors. This is a more general case and the perfect recovery result in Theorem 3 follows immediately from Theorem 4 by setting $\mathcal{E} = 0$.

We need to define a few notations before presenting our analysis. We define two linear operators $P_T : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{N\times N}$ and $P_{T^\perp} : \mathbb{R}^{N\times N} \mapsto \mathbb{R}^{N\times N}$ as follows:

$$P_T(A) = P_U A + A P_U - P_U A P_U \tag{5.9}$$

$$P_{T^\perp}(A) = (P_k - P_U) A (P_k - P_U) \tag{5.10}$$

where $P_U = UU^\top$ and $P_k = ZZ^\top$. The coherence measurement μ for binary similarity matrix S is given by

$$\mu(S) = \frac{N}{r} \max_{1\le i\le N} |P_U \mathbf{e}_i|^2 \le \frac{N}{rN_{\min}} \tag{5.11}$$

As a result, we have the following inequality for μ_0 defined in (5.4)

$$\mu_0 = \max\left(\mu(Z), \frac{N}{rN_{\min}}\right) \ge \max(\mu(Z), \mu(S))$$

Our strategy is to first identify the deterministic conditions for the optimal solution $M_* = Z^\top S Z$ to be close to $\widehat{M}$, and then confirm that these deterministic conditions will hold with

high probability.

Theorem 5. *Under the assumptions*

1. *the number of pairwise constraints is sufficiently large, i.e.*

$$|\Omega| > \frac{512\mu_0^2 r(k-r)\ln N}{3} \tag{5.12}$$

2. *there exists a dual matrix $Y \in \mathbb{R}^{N\times m}$ satisfied the following condition*

$$\begin{aligned} &\mathcal{P}_\Omega(Y) = Y, \\ &\|P_T(Y) - UU^\top\| \le \sqrt{\frac{r}{2k}}, \\ &\|P_{T^\perp}(Y)\| \le \frac{1}{2} \end{aligned} \tag{5.13}$$

3. *for any nonzero $F \in \mathbb{R}^{N\times N}$ satisfying $F = P_k F P_k$, we have*

$$\|P_T(F)\|_F \le \gamma\|P_{T^\perp}(F)\|_F + 2\|\mathcal{P}_\Omega(F)\|_F, \tag{5.14}$$

where γ is given by

$$\gamma = 4\mu_0(k-r)\sqrt{\frac{2\log N}{3|\Omega|}} \tag{5.15}$$

Then, by setting $C = \frac{1}{\sqrt{r}\mathcal{E}}$,, we have

$$|S - \widehat{S}|_F \le \left[6\left(\sqrt{2k} + 4\sqrt{r}\right)\left(3 + \sqrt{r}\right)\right]\mathcal{E}$$

Proof. Define $S_* = Z^\top M_* Z$ and $F = Z\widehat{M}Z^\top - S_*$. Evidently, we have $F = P_Z F P_Z$. Using

the condition in (5.14), we have

$$\|P_T(F)\|_F \leq \gamma \|P_{T^\perp}(F)\|_F + 2\|\mathcal{P}_\Omega(F)\|_F$$

Let $U_\perp$ be the eigenvectors of $P_{T^\perp}(Z)$. Evidently, column vectors in $U_\perp$ are orthogonal to the column vectors in U. We have

$$\begin{aligned}
|\widehat{M}|_{tr} = |\widehat{S}|_{tr} &\geq \langle S_* + Z, UU^\top + U_\perp U_\perp^\top \rangle \\
&\geq |S|_{tr} - |S_* - S|_{tr} + \langle Z, -Y + UU^\top + U_\perp U_\perp^\top \rangle \\
&\geq |S_*|_{tr} - 2\sqrt{2r}\|S_* - S\|_F \\
&\quad + \langle F, UU^\top - P_T(Y) + U_\perp U_\perp^\top - P_{T^\perp}(Y) \rangle \\
&\geq |M_*|_{tr} - 2\sqrt{2}r\mathcal{E} + \|P_T(F)\|_F \|UU^\top - P_T(Y)\|_F \\
&\quad + (1 - \|P_T(Y)\|) \|P_{T^\perp}(F)\|_F \\
&\geq |M_*|_{tr} + \|P_{T^\perp}(F)\|_F \left(\frac{1}{2} - \gamma \sqrt{\frac{r}{2k}} \right) \\
&\quad - 2\sqrt{\frac{r}{2k}} \|\mathcal{P}_\Omega(F)\|_F - 2\sqrt{2}r\mathcal{E}
\end{aligned}$$

When

$$|\Omega| > \frac{512\mu_0^2 r(k-r)\log N}{3}$$

we have

$$\begin{aligned}
|\widehat{M}|_{tr} &\geq |M_*|_{tr} + \frac{\|P_{T^\perp}(F)\|_F}{4} \\
&\quad - 2\sqrt{\frac{r}{2k}} \|\mathcal{P}_\Omega(F)\|_F - 2\sqrt{2}r\mathcal{E}
\end{aligned}$$

Since

$$\begin{aligned}\mathcal{L}(\widehat{M}) &= \frac{C}{2}\|\mathcal{P}_\Omega(Z\widehat{M}Z^\top - S)\|_F^2 \\ &\geq \frac{C}{2}\left(\|\mathcal{P}_\Omega(S_* - S)\|_F - \|\mathcal{P}_\Omega(Z)\|_F\right)^2\end{aligned}$$

and

$$C \geq \frac{1}{\sqrt{r}\mathcal{E}},$$

it is easy to verify that

$$\|\mathcal{P}_\Omega(Z)\|_F \leq (12 + 2\sqrt{r})\sqrt{r}\mathcal{E}$$

and therefore

$$\begin{aligned}\|P_{T^\perp}(Z)\| &\leq 4\left(2\sqrt{\frac{r}{2k}}\|\mathcal{P}_\Omega(Z)\|_F + 2\sqrt{2}r\mathcal{E} + \frac{C}{2}r\mathcal{E}^2\right) \\ &\leq 24\sqrt{r}\mathcal{E}\left(3 + \sqrt{r}\right)\end{aligned}$$

As a result, we have

$$\begin{aligned}\|Z\|_F &\leq \|P_T(Z)\|_F + \|P_{T^\perp}(Z)\|_F \\ &\leq (\gamma + 1)\|P_{T^\perp}(Z)\|_F + 2\|\mathcal{P}_\Omega(Z)\|_F \\ &\leq \left[6\left(\sqrt{2k} + 4\sqrt{r}\right)\left(3 + \sqrt{r}\right)\right]\mathcal{E}\end{aligned}$$

Thus finishes the proof. □

The following two theorems are developed to confirm that the conditions specified in Theorem 5 hold with a high probability.

Theorem 6. *With a probability* $1-4N^{-\beta+1}$*,for any* $Z \neq 0$ *satisfying* $Z = P_U Z P_U$*, we have*

$$\|P_T(Z)\|_F \leq \gamma \|P_{T^\perp}(Z)\|_F + 2\|\mathcal{P}_\Omega(Z)\|_F$$

where γ *is given in (5.15), provided* $|\Omega| \geq \Omega_0$ *and* $|\Omega_1| \leq \Omega_0$.

To verify if there exists a matrix Y that satisfies the condition in (5.14), we follow [18] and construct Y as follows. We randomly select $q\Omega_0$ entries from Ω, and divide the selected entries into q subsets of equal size, denoted by $\Omega_1, \ldots, \Omega_q$, with

$$|\Omega_i| = \Omega_0, \quad i = 1, \ldots, q.$$

We generate a sequence of $Y_t, t = 1, \ldots, q$ as follows

$$Y_t = \frac{N^2}{\Omega_0} \sum_{i=1}^{t} \mathcal{P}_{\Omega_i}(W_i)$$

where $W_1 = UU^\top$ and W_{t+1} is defined inductively as

$$\begin{aligned} W_{t+1} &= P_T(UU^\top - Y_t) \\ &= W_t - \frac{N^2}{\Omega_0} P_T \mathcal{P}_{\Omega_t}(W_t) \\ &= \left(P_T - \frac{N^2}{\Omega_0} P_T \mathcal{P}_{\Omega_t} P_T\right) W_t \end{aligned}$$

We construct Y as the last element of the sequence, i.e. $Y = Y_q$. Evidently, we have $Y = \mathcal{P}_\Omega(Y)$. The following theorems show that Y satisfies the other properties specified in (5.14)

Theorem 7. *With a probability* $1 - 2qN^{-\beta+1}$*, we have*

$$\|P_T(Y) - UU^\top\| \leq \sqrt{\frac{r}{2k}}$$

if $q \geq a$*.*

Theorem 6 and Theorem 7 follows directly from the analysis from [106]. Thus we omit the proofs in this thesis.

5.4 Experiments

In this section, we first conduct a simulated study to verify our theoretical claim, i.e. the sample complexity of the proposed semi-supervised clustering algorithm is only logarithmic dependence on N. We then compare the proposed algorithm to the state-of-the-art algorithms for semi-supervised clustering on several benchmark datasets.

5.4.1 Baselines, and Parameter Settings

Baselines. We compare the proposed semi-supervised clustering algorithm to the following six state-of-the-art algorithms for semi-supervised clustering, including three constrained clustering algorithms and three distance metric learning algorithms. The three constrained clustering algorithms are (a) **MPCK-means**, the metric pairwise constrained k-means algorithm [12], (b) **CCSKL**, constrained clustering by spectral kernel learning [88], and (c) **PMMC**, pairwise constrained maximum margin clustering [144]. The three state-of-the-art

distance metric learning algorithms are (d) **DCA**, the discriminative component analysis [63], (e) **LMNN**, the large margin nearest neighbor classifier [130], and (f) **ITML**, the information theoretic metric learning algorithm [36]. In order to examine the effectiveness of pairwise constraints for clustering, we also include the baseline method, referred to as **Base**, that directly applies the spectral clustering algorithm to cluster data points without any constraints. We refer to the proposed semi-supervised clustering algorithm as Matrix Completion based Constraint Clustering, or **MCCC** for short.

Evaluation and Parameter Settings. Normalized mutual information (NMI for short) [29] is used to measure the coherence between the inferred clustering and the ground truth categorization. To determine the parameter C in (6.8), we follow the heuristic used in Sections 3.3 and 4.3 that chooses the best C that results in a balanced cluster distribution. Two criteria are used in determining the values for k. First, k should be small enough to make the Algorithm 1 efficient. Second, k should be reasonably large to make the projection errors relatively small. In our experiments, we set $k = \min(100, d)$, where d is the dimensionality of the datasets.

5.4.2 Experiment with Synthesized Data

We first conduct experiments with simulated data to verify that under the assumption **A3**, the proposed semi-supervised clustering algorithm can perfectly recover the true data partition with only $O(\log N)$ sampled pairwise constraints. To this end, for a fixed N, the number of data points to be clustered, we create a partition of five clusters of equal size. Let $\mathbf{u}_i \in \{0,1\}^N, i = 1, \ldots, 5$ represent the cluster membership vectors. The target matrix

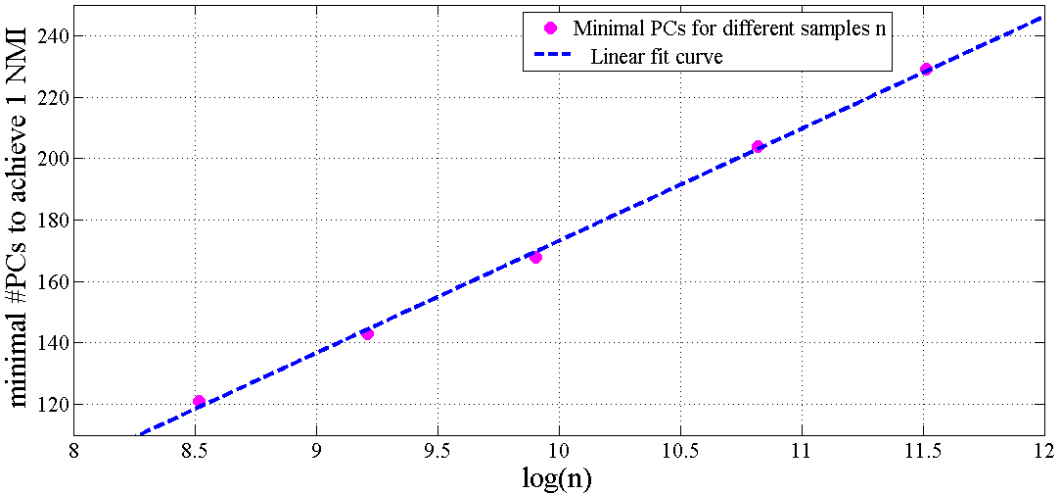


Figure 5.1 The plot of the smallest number of pairwise constraints (PCs) needed for perfect recovery. The correlation coefficient computed by the linear fit is 0.992, indicating a linear dependence of sample complexity in $\log n$.

to be recovered is $S = \sum_{i=1}^{5} \mathbf{u}_i \mathbf{u}_i^\top$. We construct the input pattern matrix X^{syn} by first generating a Gaussian random matrix $G \in \mathbb{R}^{5\times 15}$, with $G_{i,j}$ drawn independently from a Gaussian distribution $\mathcal{N}(0,1)$, and setting $X^{\text{syn}} = UG$, where $U = (\mathbf{u}_1, \ldots, \mathbf{u}_5)$. We vary N in range $\{5,000, 10,000, 20,000, 50,000, 10,0000\}$. For each N, we search for the smallest number of pairwise constraints that results in the perfect partition (i.e. NMI = 1). Figure 5.1 shows that the number of required constraints increases linearly in $\log N$, thus verifying that the sample complexity is logarithmic in the number of data points to be clustered.

Another advantage of the proposed algorithm is its scalability to large datasets since it only requires solving an optimization problem involving a small $(k\times k,\ k = \Omega(r))$ matrix. Table 5.1 summarizes the running time of recovering the synthetic data X^{syn} of different sizes, with the number of observed pairwise entries set to be the minimum required for perfect recovery. We observe that even for $N = 100,000$, it takes the proposed semi-supervised clustering algorithm less than an hour.

Table 5.1 Running time (in s) for recovering synthetic data of different size

N	5K	10K	20K	50K	100K
CPU time	24.0	77.1	217	1,086	3,429

5.4.3 Experiment with Benchmark Datasets

We then evaluate the proposed semi-supervised clustering algorithm on multiple benchmark datasets. They are (i) *Mushrooms database*[1] that contains 8,124 mushrooms belonging to 2 classes: poisonous or edible; (ii) *RCV1 M2 database*, a subset of the RCV1 corpus [82], that is comprised of 4,923 documents belonging to the categories "C15" and "GCAT".; (iii) *COIL5 database*, a subset of the larger COIL100 database [98], that is comprised of 360 images belonging to 5 objects; and (iv) *Segment database*[2] that contains 2,310 random segmentations of 7 outdoor images, (v) *USPS M5* and *L5 databases*, that are two subsets of images from the USPS handwritten dataset [68]. Among them, "USPS M5" consists of the first five categories of USPS dataset and has a total of 5,427 images. "USPS L5" consists of the last five categories and includes 3,871 images in total, (vi) *MNIST4k database* that is a subset of the MNIST handwritten digits data set [81]. The subset contains the widely used first 4,000 images which belong to 10 classes, (vii) *20 Newsgroups* database[3] which contains 18,774 documents belonging to 20 news categories, and (viii) *ImageNet data set*, a subset of the larger ImageNet database [39], that is comprised of 6,408 images belonging to 7 categories (i.e. *tractor*, *horse cart*, *bench*, *blackberry*, *violin*, *saxophone*, and *hammer*). Details of these nine datasets are given in Table 5.2.

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`
[2] `http://archive.ics.uci.edu/ml/datasets/Image+Segmentation/`
[3] `http://qwone.com/~jason/20Newsgroups/`

Table 5.2 Description of Datasets

Name	#Instances	#Features	#Clusters
Mushrooms	8,124	112	2
RCV1 M2	4,923	29,992	2
COIL5	360	1,024	5
Segment	2,310	19	7
USPS M5	5,427	256	5
USPS L5	3,871	256	5
MNIST4k	4,000	784	10
20 Newsgroups	18,774	61,188	20
ImageNet	6,408	1,000	7

We vary the number of randomly sampled pairwise constraints from 2,000, 4,000 to 6,000 for each data sets. We note that we did not run experiments with smaller numbers of pairwise constraints because our theoretical analysis shows that the proposed algorithm is effective only when the number of constraints is sufficiently large. All the experiments are performed on a PC with Xeon 2.40 GHz processor and 64.0 GB memory. Each experiment is repeated five times, and the performance averaged over five trials is reported.

Table 6.1 summarizes the performance of the proposed semi-supervised clustering algorithm and the baseline algorithms. We first observed that although all the semi-supervised clustering algorithms significantly outperform the Base method with sufficiently large numbers of pairwise constraints, generally speaking, the distance metric based algorithms outperform the constrained clustering algorithms. We conjecture that this may be due to the fact that the number of pairwise constraints is large enough to learn a good distance metric such that data points of the same class will be separated by a small distance and data points from different classes are separated by a large distance. For the Mushrooms, RCV1 M2, COIL5 and USPS M5 databases, we observe that MCCC, the proposed semi-supervised clustering algorithm, achieves very high NMI values (> 0.9) and outperforms all the baseline methods

when the number of constraints is relatively large (i.e. $4,000$ and $6,000$). Since the Base method can achieve high NMI scores (> 0.5) on these datasets without utilizing any pairwise constraints, we conjecture that these datasets satisfy our assumption, i.e., the cluster membership vectors can be well approximated by the top eigenvectors of the data matrix. Among such data sets, the experimental results for Mushrooms dataset is very encouraging since only $4,000$ pairwise constraints are needed to achieve more than 0.99 NMI. This only accounts for about 0.012% of all possible pairwise constraints. In contrast, for the datasets 20 Newsgroups and ImageNet, the Base method yields very low NMI scores (i.e. 0.221 for 20 Newsgroups and only 0.148 for ImageNet). Thus it is very likely that the cluster membership vectors of them do not lie in the space spanned by the top singular vectors of their features. For these two challenging datasets, the proposed algorithm yields similar clustering performance as the baseline methods when the number of constraints is relatively small (i.e. $2,000$ and $4,000$) and starts to significantly outperform the baseline methods when the number of constraints reaches $6,000$. This result demonstrates that even when the assumption is violated, the proposed algorithm is still able to yield good clustering performance with sufficiently large numbers of constraints.

5.5 Conclusion and Discussion

In this chapter, we propose a framework for semi-supervised clustering based on input pattern assisted matrix completion. The key idea is to cast clustering into a matrix completion problem, and solve it efficiently by exploiting the correlation between input patterns and class assignments. A stochastic subgradient descend method is employed to optimize the convex

optimization problem. Since k is usually very small (≤ 100), directly computing the SVD of $k \times k$ matrix M is also very fast. This meets our observation that a standard subgradient descend method can also be very efficient. Under the assumption that cluster membership vectors can be well approximated by the top few singular vectors of the data matrix, we show that with an overwhelming probability, the proposed algorithm can accurately recover the true data partition with only $O(\log N)$ randomly sampled pairwise constraints. Our empirical study verifies the effectiveness of the proposed algorithm.

We note that the proposed semi-supervised clustering framework is related to the semi-crowdsourced clustering framework proposed in Chapter 4. Both of them address the problem of combining features of objects with the pairwise constraints to enhance the clustering performance. The major difference between them is that the semi-crowdsourced clustering uses human annotations that are collected from a small subset of n ($n \ll N$) objects. As a result, it only needs $O(n \log^2 n)$ reliable pairwise constraints to accurately recover the true data partitions. In contrast, the proposed semi-supervised clustering approach exploits a small amount of noiseless pairwise constraints sampled from all of N objects. Then the true data partition can be almost perfectly recovered with only $O(\log N)$ randomly sampled pairwise constraints. The proposed semi-supervised clustering can also be applied to the setting of semi-crowdsourced clustering. By randomly sample a set of object pairs and ask crowd workers for annotation, we can generate a $N \times N$ partially-observed pairwise similarity matrix with all the observed entries to be reliable pairwise constraints that are agreed upon by many crowd workers. Then the proposed semi-supervised clustering framework can be used to solve the problem of semi-crowdsourced clustering.

Table 5.3 Average Clustering performance of the proposed semi-supervised clustering algorithm (MCCC) and the baseline algorithms (Base, MPCKmeans (MPCK) [12], CCSKL [88], PMMC [144], DCA [63], LMNN [130], and ITML [36]) on nine datasets with $2,000$, $4,000$ and $6,000$ randomly sampled pairwise constraints (PCs)

Datasets	#PCs	**MCCC**	Base	MPCK	CCSKL	PMMC	DCA	LMNN	ITML
Mushrooms	$2,000$	**0.982**	0.540	0.645	0.652	0.876	0.873	0.980	0.971
	$4,000$	**0.991**	0.540	0.684	0.786	0.898	0.977	0.982	0.981
	$6,000$	**0.998**	0.540	0.713	0.754	0.923	0.988	0.983	0.984
RCV1 M2	$2,000$	**0.844**	0.702	0.792	0.814	0.821	0.811	0.802	0.815
	$4,000$	**0.897**	0.702	0.846	0.884	0.895	0.867	0.874	0.883
	$6,000$	**0.932**	0.702	0.903	0.894	0.886	0.917	0.928	0.924
COIL5	$2,000$	0.914	0.582	0.909	0.624	0.897	0.818	0.925	**0.931**
	$4,000$	**0.973**	0.582	0.970	0.668	0.955	0.824	0.970	0.968
	$6,000$	**1.000**	0.582	**1.000**	0.737	0.992	0.846	0.998	**1.000**
Segment	$2,000$	**0.750**	0.651	0.693	0.721	0.718	0.723	0.714	0.706
	$4,000$	**0.755**	0.651	0.701	0.695	0.734	0.741	0.744	0.740
	$6,000$	**0.774**	0.651	0.718	0.684	0.748	0.760	0.751	0.743
USPS M5	$2,000$	**0.901**	0.681	0.864	0.869	0.793	0.872	0.890	0.884
	$4,000$	**0.923**	0.681	0.883	0.886	0.831	0.896	0.919	0.914
	$6,000$	**0.944**	0.681	0.901	0.910	0.887	0.916	0.928	0.931
USPS L5	$2,000$	**0.811**	0.521	0.792	0.798	0.789	0.793	0.802	0.808
	$4,000$	**0.830**	0.521	0.809	0.817	0.804	0.821	0.819	0.825
	$6,000$	**0.862**	0.521	0.833	0.838	0.820	0.848	0.860	0.858
MNIST4k	$2,000$	0.784	0.435	0.769	0.773	0.760	0.783	**0.789**	0.785
	$4,000$	**0.817**	0.435	0.794	0.802	0.785	0.803	0.809	0.811
	$6,000$	**0.841**	0.435	0.823	0.821	0.805	0.829	0.831	0.832
20 Newsgroups	$2,000$	0.244	0.221	0.243	0.235	**0.254**	0.164	0.213	0.225
	$4,000$	**0.293**	0.221	0.246	0.280	0.291	0.252	0.267	0.270
	$6,000$	**0.323**	0.221	0.301	0.313	0.311	0.289	0.302	0.299
ImageNet	$2,000$	0.196	0.148	0.192	0.099	0.171	0.213	0.194	**0.218**
	$4,000$	**0.255**	0.148	0.172	0.191	0.214	0.258	0.249	**0.255**
	$6,000$	**0.279**	0.148	0.202	0.213	0.256	0.271	0.269	0.265

Chapter 6

A Constant-Time Algorithm for Dynamic Semi-Supervised Clustering

One condition that is often overlooked by existing semi-supervised clustering is that side information can be generated sequentially and dynamically, a practical setting in numerous real-world applications such as social network and E-commerce system analysis. Given a set of new pairwise constraints, classical semi-supervised clustering algorithms need to re-optimize their objective functions over all the data points and constraints, prohibiting them to efficiently update the data partitions. In this chapter, we propose an efficient dynamic semi-supervised clustering framework that casts the clustering problem into a searching problem in a feasible convex space, i.e., a convex hull with its extreme points being an ensemble of multiple data partitions. According to the principle of ensemble clustering, the optimal partition lies in that convex hull and it can be uniquely represented by a low-dimensional *simplex* vector. This enables us to carry out the dynamic semi-supervised clustering problem as an updating procedure of the simplex vector based on the newly received pairwise constraints. We then derive an efficient algorithm that is able to update the simplex vec-

tor (clustering result) in a *constant* time. Our empirical studies with multiple real-world benchmark datasets show that the proposed algorithm outperforms several state-of-the-art semi-supervised clustering approaches with visible performance gain and significantly reduced running time.

The remainder of the paper is organized as follows. In Section 6.1, we describe the settings and introduce the motivation of the proposed dynamic semi-supervised clustering framework. Section 6.2 presents the proposed framework for dynamic semi-supervised clustering followed by an efficient solution using convex searching. We summarize the results of our empirical studies in Section 6.3. Section 6.4 concludes with the future work.

6.1 Introduction

Despite the progress of semi-supervised clustering, one issue that is often overlooked is how to efficiently update the clustering results when the pairwise constraints are *dynamic*, i.e., new pairwise constraints are generated sequentially. This condition is closely related to numerous real-world applications. For example, one common application in social network analysis is to group user communities based on users' profiles as well as their social connections. If we treat user profiles as features, and connections between users as pairwise constraints, then this application is essentially a semi-supervised clustering problem. Note that user connections in social networks are changing all over the time, it requires an efficient updating of user groups given newly generated connection links. In addition, similar situation also occurs in various real-world E-commerce platforms, which usually group items or customers according to their

attributes (features) and dynamic co-purchasing histories (pairwise constraints).

We note that although the problem of clustering evolving has been extensively studied, no previous study has investigated the efficiency issue in the dynamic semi-supervised clustering setting. For example, [13, 28] studied the problem of updating clustering results based on various user feedbacks. However, these algorithms are less scalable since they often need to learn distance metrics iteratively. In addition, dynamic network clustering [21, 21, 26, 26, 77, 89, 114] studied the problem of community evolving when a network to be clustered is changing continuously. However, they only use link information to guide clustering procedure and ignore the important attributes of the data points.

To address the issue of clustering updating, we propose an efficient dynamic semi-supervised clustering framework for large scale applications. The key idea is to cast the large-scale semi-supervised clustering problem into a problem of searching in a low-dimensional convex space. More specifically, the proposed algorithm consists of two components: (i) an offline step for constructing a low-dimensional convex space, and (ii) an online step for efficiently updating clustering results when new pairwise constraints are generated. In the first component, we employ the ensemble clustering technique [113] to generate m ensemble partitions of all the data points to be clustered. Note that the m ensemble partitions can form a convex hull [61] with m extreme points. According to the principle of ensemble clustering [52, 113, 117], the optimal data partition can be approximated by a linear combination of the m ensemble partitions, indicating that the optimal partition should lie in the inner space of that convex hull. Since the inner space of a convex hull can be spanned by the linear combinations of the extreme points, the problem of finding the optimal data partition is equivalent to deriving the

combination weights, denoted as a m-dimensional simplex vector $\boldsymbol{\gamma}$ [1]. Given new pairwise constraints generated at time t, this enables us to efficiently update data partitions by computing an updated simplex vector $\boldsymbol{\gamma}^t$ from $\boldsymbol{\gamma}^{t-1}$. Therefore, in the second component of the proposed framework, we design an efficient updating scheme that is able to update the simplex vector in a constant time. Note that the new data partition should be somewhat similar to the old data partition. We cast the simplex updating problem into a learning problem that aims to learn a vector $\boldsymbol{\gamma}^t$ that is close to $\boldsymbol{\gamma}^{t-1}$, and also consistent with the new pairwise constraints. We then present an efficient solver for our learning problem.

Compared to the existing approaches of semi-supervised clustering, the proposed algorithm has the following two advantages: (i) by exploring the ensemble clustering technique, the proposed algorithm is able to exploit the strength of different ensemble partitions and at the same time, compensate for their limitations; (ii) by casting the problem of clustering n data points into the problem of learning a m-dimensional vector $\boldsymbol{\gamma}$, the proposed algorithm is computationally efficient and the time complexity of updating $\boldsymbol{\gamma}$ is irrelevant to the number of data points to be clustered. This enables us to update large-scale clustering results in an extremely efficient way. To evaluate the performance of the proposed algorithm, we conduct empirical studies on several large-scale real-world datasets. The experimental results and the comparison with peer methods verify both the efficiency and effectiveness of the proposed algorithm.

[1] A simplex vector is a vector whose elements are non-negative and the summation of all the elements equals to 1

6.2 Semi-supervised Clustering with Dynamic Constraints

In this section, we first present a more general framework for semi-supervised clustering, then discuss the proposed efficient dynamic semi-supervised clustering algorithm.

6.2.1 Semi-supervised clustering

Let $\mathcal{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ be a set of N data points to be clustered, where each data point $\mathbf{x}_i \in \mathbb{R}^d,\ i \in [N]$ is a vector of d dimensions. Let $\mathcal{M}_t$ be the set of must-link constraints generated till time t, where each must-link pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}_t$ implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ should be in the same cluster. Similarly, let $\mathcal{C}_t$ be the set of cannot-link constraints generated till time t, where each cannot-link pair $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}_t$ implies that $\mathbf{x}_i$ and $\mathbf{x}_j$ should belong to different clusters. For the ease of presentation, we also define $\Omega_t = \mathcal{M}_t \cup \mathcal{C}_t$ to include all pairwise constraints generated till time t. Similar to most studies on data clustering, we assume that the number of clusters r is given as a priori. Throughout this paper, we use a binary matrix $F \in \{0,1\}^{N \times r}$ to represent the result of partitioning N data points into r clusters, where $F_{ij} = 1$ indicates that $\mathbf{x}_i$ is associated with the j-th cluster. We further denote $\mathcal{F}$ as the set of all possible clustering results

$$\mathcal{F} = \{F \in \{0,1\}^{N \times r} :\ F_{*i}^{\top} F_{*j} = 0\ \ \forall i \neq j, \sum F_{k*} = 1\ \forall k\}.$$

Let $\kappa(\mathbf{x}, \mathbf{x}')$ be a kernel function used to measure the similarity between two data points $\mathbf{x}$ and $\mathbf{x}'$. Let $K = [\kappa(\mathbf{x}_i, \mathbf{x}_j)] \in \mathbb{R}_{+}^{N \times N}$ be the kernel matrix, and let $K_{i,*}$ be the i-th row vector of K. The goal of semi-supervised learning is to find the clustering result that is

consistent with both the kernel similarities in K and pairwise constraints in Ω_t. To measure the discrepancy between the kernel similarity K and a clustering result F, we define the distance between K and F as

$$d(K, F) = \sum_{i=1}^{N} F_{i,*}^{\top} K^{-1} F_{j,*} = \mathrm{tr}\left(F^{\top} K^{-1} F\right) \tag{6.1}$$

As indicated by the above measure, the smaller the distance $d(K, F)$, the better the consistency between the clustering result F and the similarity matrix K. We note that an alternative approach is to measure the distance by $\mathrm{tr}(F^{\top} L F)$ where $L = \mathrm{diag}(K\mathbf{1}) - K$ is the graph Laplacian.

To measure the inconsistency between the clustering result F and pairwise constraints, we introduce two loss functions, one for must-links and one for cannot-links. More specifically, given a must-link $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}_t$, we define the loss function $\ell_-(F_{i,*}, F_{j,*})$ as

$$\ell_-(F_{i,*}, F_{j,*}) = \|F_{i,*} - F_{j,*}\|_2^2 \tag{6.2}$$

Similarly, given a cannot-link $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}_t$, we define the loss function $\ell_+(F_{i,*}, F_{j,*})$ as

$$\ell_+(F_{i,*}, F_{j,*}) = \|F_{i,*} + F_{j,*}\|_2^2 \tag{6.3}$$

We note that a loss function similar to (6.3) has been used in label propagation with cannot-links [92]. The justification of using loss function $\ell_+(\cdot, \cdot)$ for cannot-links is provided in the following proposition.

Proposition 1. *Let* $\mathbf{a} \in \mathbb{R}_+^d$ *be a fixed vector. Let* $\mathbf{b}_*$ *be the minimizer to the following*

optimization problem

$$\min_{\mathbf{b}\in\mathbb{R}^d_+, \|\mathbf{b}\|_q \le R} \ell_+(\mathbf{a}, \mathbf{b})$$

where $q \geq 1$. *Then we have* $\mathbf{b}_* \perp \mathbf{a}$.

As indicated by Proposition 1, by minimizing the loss function $\ell_+(F_{i,*}, F_{j,*})$, the resulted solution, under no other constraints, will satisfy $F_{j,*} \perp F_{i,*}$, implying that $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned to different clusters.

Using the distance measure $d(K, F)$ and the loss functions, we can cast semi-supervised clustering into the following optimization problem

$$\begin{aligned} \min_{F\in\mathcal{F}} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t} \ell_+(F_{i,*}, F_{j,*}) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t} \ell_-(F_{i,*}, F_{j,*}) \\ \text{s. t.} \quad & d(F, K) \leq \varepsilon, \end{aligned} \tag{6.4}$$

where the threshold ϵ decides the level of consistency between the clustering results and the kernel similarity.

The main challenge of dynamic semi-supervised clustering arises from the fact that pairwise constraints are dynamically updated over time. A naive approach is to solve the optimization problem in (6.4) from the scratch whenever pairwise constraints are updated. A more efficient approach is to explore the fact that only a small portion of pairwise constraints are updated at each time, leading to a small change in the clustering result. Based on this intuition, we can use the existing solution as an initial solution to the optimization problem with updated constraints. Despite the simplicity, this approach can significantly reduce the running time, and has been widely used in clustering social networks with dynamic up-

dates [21, 26]. Although this simple approach reduces the number of iterations due to the appropriate initialization, it still needs to solve the optimization problem in (6.4), which is computationally expensive when the number of data points N is very large. To address this challenging issue, we propose an efficient dynamic semi-supervised clustering algorithm that it highly attractive for clustering large-scale data sets.

6.2.2 A Constant Time Algorithm for Dynamic Semi-supervised Clustering

The proposed algorithm is based on the key observation that the number of different clustering results F in the subspace $\Delta = \{F \in \mathcal{F} : d(K, F) \leq \varepsilon\}$ is not large when ε is sufficiently small and the eigenvalues of K follow a skewed distribution, where $d(K, F) = tr(F^\top K^{-1} F)$.

To this end, we denote by $\lambda_1, \ldots, \lambda_N$ the eigenvalues of K ranked in the descending order, and by $\mathbf{v}_1, \ldots, \mathbf{v}_N$ the corresponding eigenvectors. $\{\lambda_i\}$ follows a q-power law if there exists constant c such that $\lambda_k \leq ck^{-q}$, where $q > 2$. The following lemma summarizes an important property of K when its eigenvalues follow a q-power law.

Lemma 2. *Define* $V = (\mathbf{v}_1, \ldots, \mathbf{v}_N)$ *and* $\xi = V^\top \mathbf{x}$ *for a unit vector* $|\mathbf{x}| = 1$. *If* $\mathbf{x}^\top K^{-1} \mathbf{x} \leq \varepsilon$, *we have*

$$\frac{|\xi|_1}{|\xi|_2} \leq \sqrt{\varepsilon N} \left(1 + \frac{2}{q-2}\right)$$

provides that eigenvalues of K *follow a* q*-power law.*

Proof. Since

$$\mathbf{x}^\top K^{-1}\mathbf{x} = \sum_{i=1}^{N} \lambda_i^{-1}\xi_i^2 \leq \varepsilon,$$

we have

$$\xi_k \leq \sqrt{c\varepsilon}k^{-q/2},\ k = 1, \ldots, N$$

and therefore

$$|\xi|_1 \leq \sqrt{c\varepsilon}\sum_{k=1}^{N} k^{-q/2} \leq \sqrt{\varepsilon c}\left(1 + \frac{2}{q-2}\right)$$

We complete the proof by using the fact $|\mathbf{x}|_2 = 1$ and $c \leq n$. □

The above lemma shows that when the eigenvalues of K follow a power law, $V\mathbf{x}$ is a ℓ_1 sparse vector if $\mathbf{x}^\top K^{-1}\mathbf{x} \leq \varepsilon$. The observation provides the key foundation for our analysis.

In order to show that the number of significantly different partitions in Δ is small, we first consider the simple case where the number of classes is 2. In this case, we can simplify the domain Δ as

$$\Delta_2 = \left\{\mathbf{v} \in \{-1, +1\}^N : \mathbf{v}^\top K^{-1}\mathbf{v} \leq N\varepsilon\right\}$$

We define $\theta_N(\rho)$ the maximum number of partitions in Δ_2 such that the difference between any two partition vectors $\mathbf{v}_1$ and $\mathbf{v}_2$ is at least ρN. The theorem below bound $\theta_N(\rho)$.

Theorem 1.

$$\theta_N(\rho) \leq \left(\frac{2d}{s}\right)^{Cs/[2\rho]}$$

where C is an universal constant and

$$s = \sqrt{\varepsilon N}\left(1 + \frac{2}{q-2}\right) \tag{6.5}$$

Proof. We first notice that $|\mathbf{v}|_2 = \sqrt{N}$. As the first step, we relax Δ_2 into Δ_2' as follows

$$\Delta_2' = \left\{ \mathbf{x} \in \mathbb{R}^d : |\mathbf{x}|_2 \leq 1, \mathbf{x}^\top K^{-1} \mathbf{x} \leq \varepsilon \right\}$$

Define $\gamma_N(\mathbf{K}, \delta)$ the maximum number of vectors such that the distance between any two vectors is at least δ. Since the distance between any two partitions that differs by at least ρN entries is at least $\sqrt{2\rho}$

$$\theta_N \leq \gamma_N \left(\Delta_2', \sqrt{2\rho} \right)$$

Using Lemma 3.4 from [104] to bound, we have

$$\gamma_n \left(\Delta_2', \frac{2}{\sqrt{N}} \right) \leq \exp \left(\frac{Cs}{2\rho} \log \frac{2d}{s} \right) = \left(\frac{2d}{s} \right)^{Cs/[2\rho]}$$

where s is defined in (6.5). □

It is straightforward to extend the above result for two-way clustering into multiple-way clustering. Define $\theta_N(\rho; r)$ be the maximum number of partitions in Δ for r-way clustering such that the difference between any two partitions is at least ρN. Then we have

$$\theta_N(\rho; r) \leq \left(\frac{2d}{(r-1)s} \right)^{Cs(r-1)/[2\rho]}$$

Based on the above results, there is a relatively small number of significantly different clustering results in the subspace Δ. Hence, to improve the computational efficiency of dynamic semi-supervised clustering, a natural thought is to pre-compute all the possible clustering results in Δ, and find the best clustering result in Δ that is consistent with most of the

dynamically updated pairwise constraints. The main shortcoming of this approach is that it is computationally infeasible to identify all the different clustering results in Δ.

To address this problem, we propose to construct a convex hull $\widetilde{\Delta} \subset \mathcal{F}$ to approximate different clustering results in Δ. The key advantage of using convex hull approximation is that all the solutions in $\widetilde{\Delta}$ can be represented by a linear combination of the extreme points of the convex hull. As a result, instead of searching for the best clustering result, we only need to compute the linear combination weights. Since the number of combination weights to be determined equals to the number of extreme points, which is much smaller than the number of data points to be clustered, it is significantly more efficient to compute combination weights than to directly estimate the best clustering result. More specifically, the proposed learning process is comprised of an offline step and an online step. In the offline step, we compute multiple partitions for the data points in $\mathcal{X}$ using the ensemble clustering technique. These clustering results will be used to construct the convex hull $\widetilde{\Delta}$ that approximates the solutions in Δ. In the online step, an efficient learning algorithm is developed to update the combination weights based on the newly received pairwise constraints. Below, we describe in detail the two key steps of the proposed algorithm for dynamic semi-supervised clustering.

6.2.3 Offline Step: Ensemble Clustering

In this step, we generate a convex hull $\widetilde{\Delta} \subset \mathcal{F}$ using the technique of ensemble clustering [113]. Ensemble clustering is motivated by the fact that different clustering algorithms have their own merits, as well as their own limitations. Thus no single clustering algorithm

is universally better than others for all types of data. Using the idea of ensemble clustering, we will create m different partitioning results of the data, and construct a convex hull based on the computed data partitions to approximate the feasible clustering results in Δ.

According to [54], multiple partitions of a same dataset can be generated by (i) running different clustering algorithms [48], (ii) running the same algorithm with different initializations and parameters [47, 116], (iii) clustering via sub-sampling the data repeatedly [96, 118], and (iv) clustering via projecting the data onto different subspaces [42, 48]. In order to efficiently generate m ensemble partitions for a large-scale data set, we follow the last approach by first randomly selecting m different subsets of attributes in $\mathcal{X}$ and then applying the approximate kernel k-means algorithm [27] to each subset of selected features to create m data partitions. We denote by $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ the m realigned ensemble partitions processed by the Hungarian algorithm [78], where each partition $P_l \in \mathcal{F},\ l \in [m]$ divides $\mathcal{X}$ into r disjoint subsets. Using m partitions of dataset $\mathcal{X}$, we then construct a convex hull

$$\widetilde{\Delta} = \left\{ F |\ F = \sum_{i=1}^{m} \gamma_i P_i, \boldsymbol{\gamma} \succeq 0, 1^\top \boldsymbol{\gamma} = 1 \right\},$$

where each clustering solution F is represented by a simplex vector $\boldsymbol{\gamma} \in \mathbb{R}^m$. Two key observations are made here: (i) by generating multiple partitions using a kernel k-means algorithm, which is shown to be closely related to spectral clustering [40], all the ensemble partitions should satisfy the condition

$$d(K, P_i) \leq \varepsilon, \quad \forall i \in [m].$$

This indicates that the constructed convex hull $\widetilde{\Delta}$ is a subspace of Δ; and (ii) according to the

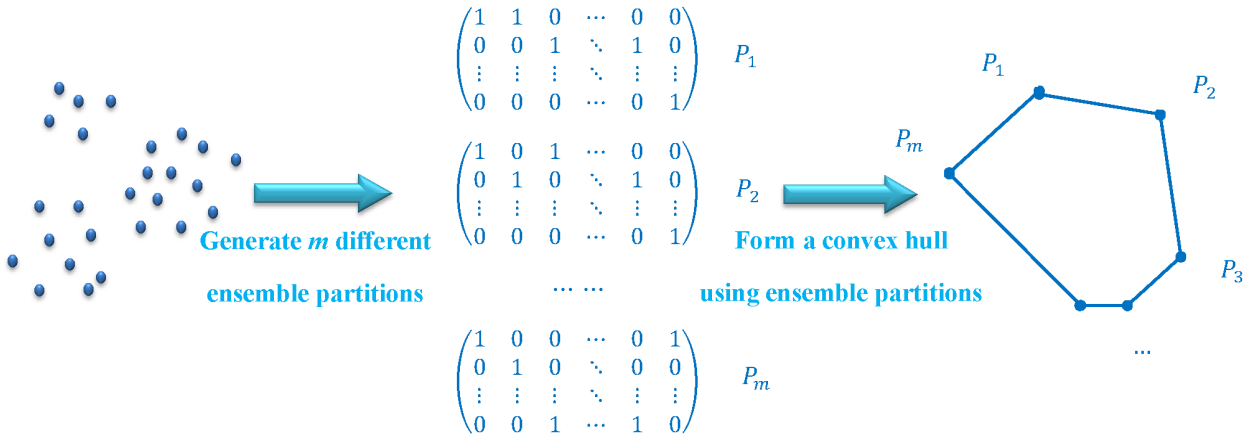


Figure 6.1 The offline step: generate a convex hull using the technique of ensemble clustering

widely used median partition based ensemble clustering approaches [52,113,117], the optimal data partition should be similar to all the ensemble partitions $P_1, \ldots, P_m$, indicating that the optimal partition should lie in the inner space of $\widetilde{\Delta}$ and it can be uniquely represented by a m-dimensional simplex vector $\boldsymbol{\gamma}$.

Using the convex hull representation, we cast a problem of clustering N data points into the problem of finding a m-dimensional simplex vector $\boldsymbol{\gamma}$. More specifically, instead of finding the best F, we will solve the following optimization problem to find the optimal $\boldsymbol{\gamma}$

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\mathbb{R}_+^m} \quad & \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t} \ell_+(F_{i,*},F_{j,*}) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t} \ell_-(F_{i,*},F_{j,*}) \\ \text{s. t.} \quad & \boldsymbol{\gamma}^\top \mathbf{1} = 1, \quad F = \sum_{i=1}^m \gamma_i P_i \end{aligned} \tag{6.6}$$

Since $m = O(1)$ is a small number independent of the data size N, the optimization problem in (6.6) can be solved efficiently with a constant running time [1]. When no pairwise constraints

[1]We note that the ensemble partitions are computed offline and therefore do not affect the efficiency of online computation, which is the main concern of this work

is generated, the clustering problem reduces to a standard ensemble clustering problem. The clustering result in terms of simplex $\boldsymbol{\gamma}$, denoted as $\boldsymbol{\gamma}^0$, can be simply set to $(\frac{1}{m}, \ldots, \frac{1}{m})$ or computed by some existing ensemble clustering algorithms such as [113, 117]. Figure 6.1 shows the conceptual framework of the offline step. In the next subsection, we introduce the online step for efficiently updating the data partition (i.e. the simplex vector) when pairwise constraints are dynamically updated over time.

6.2.4 Online Step: Efficient Updating Simplex Vector

Although the formulation in (6.6) already significantly reduces the overall computational cost, it is still undesirable to solve the optimization problem in (6.6) from the scratch whenever new pairwise constraints are introduced. This is particularly true if for most of the time, only a few constraints are added to the system. In this subsection, we develop an *online* step to efficiently update the simplex vector when the number of newly added constraints is small. To simplify the presentation, we divide $\mathcal{M}_t$, the set of must-link constraints received till time t, into two subsets: $\mathcal{M}_t^a$ that includes all the must-link constraints received before time t and $\mathcal{M}_t^b$ that includes the new must-link constraints added at time t. Similarly, we divide the cannot-link set $\mathcal{C}_t$ into $\mathcal{C}_t^a$ and $\mathcal{C}_t^b$. We also denote by $\boldsymbol{\gamma}^1, \ldots, \boldsymbol{\gamma}^T$ the sequence of simplex vectors computed based on the updated constraints.

Using the above notation, we rewrite the optimization problem in (6.6) as

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\mathbb{R}_+^m} \quad & \mathcal{L}_t^a(F(\boldsymbol{\gamma})) + \mathcal{L}_t^b(F(\boldsymbol{\gamma})) \\ \text{s. t.} \quad & \boldsymbol{\gamma}^\top \mathbf{1} = 1, \quad F = \sum_{i=1}^m \gamma_i P_i \end{aligned} \tag{6.7}$$

where

$$\begin{aligned} \mathcal{L}_t^a(F) &= \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t^a} \ell_+(F_{i,*}, F_{j,*}) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t^a} \ell_-(F_{i,*}, F_{j,*}) \\ \mathcal{L}_t^b(F) &= \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t^b} \ell_+(F_{i,*}, F_{j,*}) + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t^b} \ell_-(F_{i,*}, F_{j,*}) \end{aligned}$$

Since $\boldsymbol{\gamma}^{t-1}$ approximately minimizes the objective $\mathcal{L}_t^a(F(\boldsymbol{\gamma}))$, we can approximate $\mathcal{L}_t^a(F(\boldsymbol{\gamma}))$ as

$$\mathcal{L}_t^a(F(\boldsymbol{\gamma})) \approx \mathcal{L}_t^a(F(\boldsymbol{\gamma}^{t-1})) + \lambda\|\boldsymbol{\gamma} - \boldsymbol{\gamma}^{t-1}\|^2$$

and as a result, the optimization problem in (6.7) is further simplified as

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\mathbb{R}_+^m} \quad & \mathcal{L}_t^b(F(\boldsymbol{\gamma})) + \lambda\|\boldsymbol{\gamma} - \boldsymbol{\gamma}^{t-1}\|^2 \\ \text{s. t.} \quad & \boldsymbol{\gamma}^\top \mathbf{1} = 1, \quad F = \sum_{i=1}^m \boldsymbol{\gamma}_i P_i, \end{aligned} \tag{6.8}$$

where parameter λ is introduced to balance between two objectives, i.e., ensuring that the learned $\boldsymbol{\gamma}$ is close to $\boldsymbol{\gamma}^{t-1}$, and also consistent with the new pairwise constraints. Compared to (6.7), the main advantage of using (6.8) is that it only involves the new constraints that are added to the system at time t and does not need to store and work with the constraints

received before time t.

In the following, we discuss how to efficiently update the simplex vector with the new pairwise constraints. By representing F as a linear combination of $\{P_i\}_{i=1}^m$, we rewrite the optimization problem (6.8) as

$$\begin{aligned} \min_{\boldsymbol{\gamma}\in\mathbb{R}_+^m} \quad & f(\boldsymbol{\gamma}) = \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t^b} || \sum_{k=1}^m \boldsymbol{\gamma}_k [P_k(i,:) - P_k(j,:)]||^2 \\ & + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t^b} || \sum_{k=1}^m \boldsymbol{\gamma}_k [P_k(i,:) + P_k(j,:)]||^2 \\ & + \lambda ||\boldsymbol{\gamma} - \boldsymbol{\gamma}^{t-1}||^2 \qquad (6.9) \\ \text{s. t.} \quad & 1^\top \boldsymbol{\gamma} = 1, \end{aligned}$$

where $P_k(i,:)$ and $P_k(j,:)$ represent the i-th and the j-th row of the matrix P_k. The optimization problem (6.9) can be efficiently solved by a gradient descend method. Specifically, we update the $\boldsymbol{\gamma}$ by

$$\boldsymbol{\gamma} = P_\Delta(\boldsymbol{\gamma}^{t-1} - \eta \nabla f(\boldsymbol{\gamma}^{t-1})),$$

where η is a step size and $P_\Delta : \mathbb{R}^m \to \mathbb{R}^m$ is a projection operator that takes a m-dimensional vector as input and outputs its projected simplex vector, as described in [43].

Note that $\nabla f(\boldsymbol{\gamma}^{t-1})$ has a closed-form solution as

$$\begin{aligned} \nabla f(\boldsymbol{\gamma}^{t-1}) \quad = \quad & 2(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t^b} \boldsymbol{\gamma}^{t-1} U^{(ij)} U^{(ij)\top} + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t^b} \boldsymbol{\gamma}^{t-1} V^{(ij)} V^{(ij)\top} \\ & + \lambda \boldsymbol{\gamma}^{t-1} - \lambda \boldsymbol{\gamma}^{t-2}), \qquad (6.10) \end{aligned}$$

where $U^{(ij)}$ and $V^{(ij)}$ are two $m \times r$ matrices, satisfying

$$U^{(ij)}(k,:) = P_k(i,:) - P_k(j,:), \quad V^{(ij)}(k,:) = P_k(i,:) + P_k(j,:).$$

Then the optimal solution of $\boldsymbol{\gamma}$ is given by

$$\begin{aligned} \boldsymbol{\gamma} \;=\; & P_\Delta\{\boldsymbol{\gamma}^{t-1}[I_m - 2\eta(\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t^b} U^{(ij)}U^{(ij)\top} + \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t^b} V^{(ij)}V^{(ij)\top})] \\ & -2\lambda\eta(\boldsymbol{\gamma}^{t-1} - \boldsymbol{\gamma}^{t-2})\}, \end{aligned} \tag{6.11}$$

where I_m is the $m \times m$ identity matrix. Since we can precompute $U^{(ij)}U^{(ij)\top}$ and $V^{(ij)}V^{(ij)\top}$ for each pair (i,j) offline and store the results in a server, we can efficiently update $\boldsymbol{\gamma}$ using equation (6.11).

Despite a low time complexity of the above updating scheme, it however requires a space complexity of $O(N^2m^2)$ to store all the matrices, which is expensive when the number of objects N is large. In the following, we discuss how to reduce the expensive storage cost by relaxing the optimization procedure.

Note that the k-th row of the matrix $U^{(ij)}$ should equal to either of these two cases: (i) containing all zero entries if the ensemble partition P_k put object i and object j in the same cluster, or (ii) containing one positive entry (=2) , and one negative entry (=−2) if the ensemble partition P_k put object i and object j in different clusters. Then the diagonal elements of the matrix $U^{(ij)}U^{(ij)\top}$ either equal to 0 or equal to a positive value (=8). Thus

the matrix

$$I_m - 2\eta \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t^b} U^{(ij)}U^{(ij)\top}$$

essentially suggests us to assign less weights to the ensemble partitions that mistakenly assign the object i and object j in different clusters when they share a must-link connection. Likewise, the matrix

$$I_m - 2\eta \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t^b} V^{(ij)}V^{(ij)\top}$$

essentially suggests us to assign less weights to the ensemble partitions that mistakenly assign the object i and object j in the same cluster when they share a cannot-link constraint. After updating $\boldsymbol{\gamma}$ from $\boldsymbol{\gamma}^{t-1}$, the ensemble partitions that are consistent with the new pairwise constraints are assigned with larger weights, while the ensemble partitions that are not consistent with the new pairwise constraints are assigned with smaller weights. This leads to a relaxed updating procedure

$$\boldsymbol{\gamma} = P_\Delta\{(1-2\lambda\eta)\boldsymbol{\gamma}^{t-1} + 2\lambda\eta\boldsymbol{\gamma}^{t-2} - \eta\boldsymbol{\gamma}^{t-1} \circ \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in \{\mathcal{M}_t^b\cup\mathcal{C}_t^b\}} \mathbf{a}^{(ij)}\} \tag{6.12}$$

where $\mathbf{a}^{(ij)}$ is a m-dimensional vector with the k-th element equaling to 1 if the ensemble partition P_k is not consistent with the pairwise constraints $(\mathbf{x}_i, \mathbf{x}_j)$, and 0 otherwise.

Given the learned simplex vector $\boldsymbol{\gamma}$, we can compute a soft labeling matrix as a linear combination of the ensemble partitions

$$P = \boldsymbol{\gamma}_1 P_1 + \boldsymbol{\gamma}_2 P_2 + \ldots + \boldsymbol{\gamma}_m P_m.$$

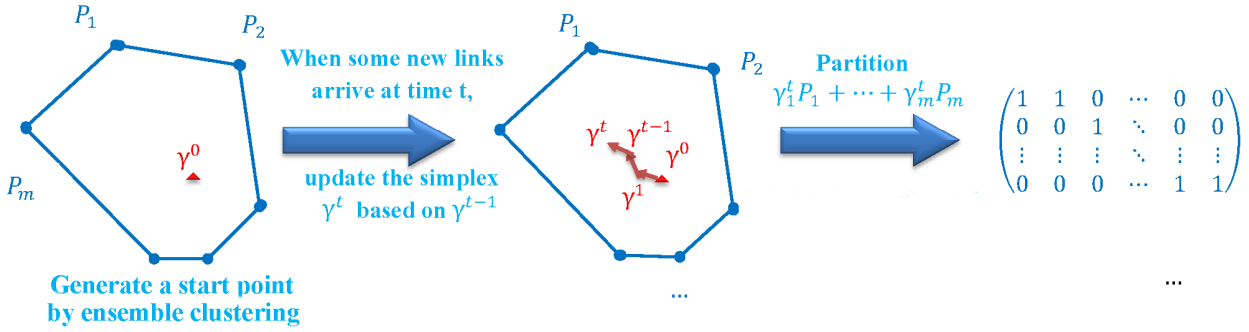


Figure 6.2 The online step: efficiently update clustering results when new pairwise constraints are generated

Then the hard partition can be easily obtained by rounding, i.e., assign the i-th data point to the k-th cluster if P_{ik} is the largest entry in the i-th row of the matrix P. The time complexity to update the simplex vector $\boldsymbol{\gamma}$ is only $O(m)$, which is a constant time irrelevant to the number of data points. Although we need to go through each row of the matrix P for final partitioning, it is still very efficient since we only need to find the largest element in each row of P. Figure 6.2 depicts the steps of online updating.

6.3 Experiments

In this section, we present extensive empirical evaluations of the proposed dynamic semi-supervised clustering algorithm, namely Constrained Clustering by Convex Searching (CCCS for short). In particular, we aim to address the following questions in our study:

1. *How does the number of extreme points m influence our performance*?

2. *Does the proposed semi-supervised clustering method outperform the state-of-the-art algorithms for semi-supervised clustering*?

3. *Is the proposed semi-supervised clustering method significantly more efficient than the state-of-the-art algorithms for semi-supervised clustering?*

6.3.1 Experimental Setup

Datasets. In order to examine the effectiveness of the proposed dynamic semi-supervised clustering algorithm, nine public benchmark datasets are used in the experiments:

- ***USPS*** [68] is a widely used handwritten digit database containing $9,298$ handwritten images. Each image is a 256 dimensional vector that belongs to one of 10 classes.

- ***RCV1*** is a subset of text documents from the RCV1 corpus [82]. It contains $193,844$ documents that belongs to one of 103 classes.

- ***20 News*** [1] is a well-known database that contains $18,774$ documents belonging to 20 news categories.

- ***BlogCatalog*** is a social blog directory dataset that was crawled from BlogCatalog[2] by Wang et. al [127]. This dataset contains a total of $19,664$ bloggers that are grouped into 60 categories. Each blogger is represented as a $5,413$-dimensional vector with each dimension denoting one semantic tag.

- ***TDT2*** [3] is a text dataset that contains $9,394$ documents belonging to one of 30 classes.

[1] `http://qwone.com/~jason/20Newsgroups/`
[2] `http://www.blogcatalog.com`
[3] `http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html`

- ***MNIST*** is also a well-known handwritten digits database[1]. It contains a total of $70,000$ handwritten digits which belong to one of 10 classes.

- ***YouTube*** [103] is a dataset retrieved using the YouTube Data API[2] from Oct. 31st, 2011 to Jan. 17th, 2012. This dataset has $2,860,264$ users and their comments on $6,407$ YouTube videos. All the users are belong to one of the two imbalanced classes: $177,542$ spam users or $2,682,722$ non-spam users.

- ***BBC*** [3] is a database consisting of BBC news during the years of 2004 and 2005. It contains $2,225$ news with $9,636$ distinct words, categorized into five topics: business, politics, tech, sport, and entertainment.

- ***Network Intrusion*** [112] is a database containing $4,897,988$ patterns representing TCP dump data of network traffic for a local-area network (LAN). Each pattern is a 50-dimensional vector that belongs to one of 10 different classes.

Parameter Selection. In order to generate m ensemble partitions, we need to randomly sample $\tilde{d}$ out of d features in each time. Two criteria are used in determining the values of $\tilde{d}$. First, $\tilde{d}$ should be small enough to make the ensemble partitions diverse. Second, $\tilde{d}$ should be reasonably large to get good enough ensemble partitions since the quality of the starting point $\boldsymbol{\gamma}^0$ is rely on the quality of m ensemble partitions. In our experiments, we set $\tilde{d} = [d/10]$. The parameter λ in problem (6.9) is introduced to balance the tradeoff between the change of the simplex vector and the fitness over the new pairwise constraints. Note that

[1] `http://yann.lecun.com/exdb/mnist/`
[2] `http://code.google.com/apis/youtube/getting_started.html`
[3] `http://mlg.ucd.ie/datasets/bbc.html`

the term $||\boldsymbol{\gamma} - \boldsymbol{\gamma}^{t-1}||^2$ is upper bounded by 2, while the term

$$\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{M}_t^b} ||\sum_{k=1}^{m} \boldsymbol{\gamma}_k[P_k(i,:) - P_k(j,:)]||^2$$

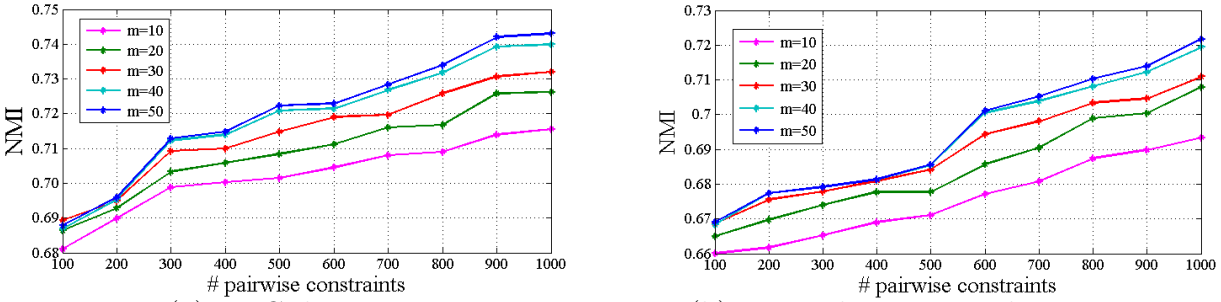
and

$$\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{C}_t^b} ||\sum_{k=1}^{m} \boldsymbol{\gamma}_k[P_k(i,:) + P_k(j,:)]||^2$$

are upper bounded by $2 \times |\mathcal{M}_t^b|$ and $2 \times |\mathcal{C}_t^b|$, respectively. In order to make two terms comparable, we set λ to be $|\mathcal{M}_t^b \cup \mathcal{C}_t^b|$, the number of pairwise constraints generated at time t.

6.3.2 Parameter Sensitivity of m

A key factor that influences the performance of the proposed algorithm is the number of ensemble partitions m, which introduces a trade-off between clustering quality and efficiency. As m increases, the clustering quality tend to improve at the cost of increased computational time. In order to analyze how the clustering performance is influenced by different m, we first conduct experiments with two benchmark datasets, BBC and Network Intrusion. We choose these two datasets since they are the smallest and the largest testbeds in our study. By this means, we can also analyze whether the selection of m is influenced by the data size N. To this end, for both of the two datasets, we begin with the unsupervised data partition $\boldsymbol{\gamma}^0$ generated from $m = \{10, 20, 30, 40, 50\}$ different ensemble partitions. Then we randomly generate 100 pairwise constraints based on the ground truth information of the data sets in each tier, and apply the proposed **CCCS** algorithm to update the partition. We repeat this



(a) BBC data set (b) Network Intrusion data set

Figure 6.3 NMI vs. the number of ensemble partitions $m = \{10, 20, 30, 40, 50\}$ with different number of pairwise constraints.

step for 10 times, until the total number of pairwise constraints achieves $1,000$. Figure 6.3 shows how the performance of the proposed dynamic semi-supervised clustering algorithm is changed with different m and different number of pairwise constraints.

From this figure, we first observe that the clustering performance keeps increasing with more and more pairwise constraints being provided. In addition, the clustering performance was improved when m increases. We conjecture that the performance gain may be due to the fact that a larger m indicates a larger searching space and also a larger overlap between the the true searching space Δ and the relaxed space $\tilde{\Delta}$. In addition, a larger number of ensemble partitions usually provide a more diverse clustering results, leading to a higher chance to find data partitions that are consistent with most of the pairwise constraints. Furthermore, the performance of the proposed semi-supervised clustering algorithm gradually stabilizes as m increases to 50. This is also consistent with the intuition since when the number of ensemble partitions are already large enough, adding more partitions cannot provide more information since it is likely that the new data partitions are coincide with some existing partitions. This provides us a guidance to appropriately choose m: on one hand, m should be reasonably large to provide a diverse and large enough convex searching space, while on

the other hand, m should be relatively small to reduce the computational cost to update the partitions. In our experiments, we set $m = 50$ for all the remaining experiments.

6.3.3 Comparison with Other Baseline Algorithms

To examine the effectiveness and efficiency of the proposed semi-supervised clustering algorithms, we compare it to the following seven state-of-the-art algorithms for semi-supervised clustering, including three constrained clustering algorithms and four distance metric learning algorithms. The three constrained clustering algorithms are (a) **MPCK-means**, the metric pairwise constrained k-means algorithm [12], (b) **CCSKL**, constrained clustering by spectral kernel learning [88], and (c) **PMMC**, pairwise constrained maximum margin clustering [144]. The four state-of-the-art distance metric learning algorithms are (d) **RCA**, the relevant component analysis [6], (e) **DCA**, the discriminative component analysis [63], (f) **LMNN**, the large margin nearest neighbor classifier [130], and (g) **ITML**, the information theoretic metric learning algorithm [36]. The code for the baseline algorithms was provided by their respective authors. For a fair comparison, all the parameters used in the baseline algorithms were their suggested values (if applicable).

In our experiments, we begin with 100 randomly generated pairwise constraints, denoted as tier t_1. In each of the following tier, another set of 100 randomly sampled pairwise constraints are generated and all the semi-supervised clustering algorithms are called to update the data partition given the newly generated pairwise constraints. Specifically, we rerun all the baseline semi-supervised clustering algorithms by combining the new pairwise constraints with the old ones. We repeat such steps from tier t_1 to tier t_{10}, finally leading to a total of

Table 6.1 Average CPU time (in s) for updating the partition in each tier. (N/A means that the clustering task cannot be accomplished by the algorithms within 5 hours.)

Datasets	CCCS	MPCK	CCSKL	PMMC	RCA	DCA	LMNN	ITML
20 News	**0.6**	9.7	N/A	N/A	489	377	N/A	N/A
RCV1	**1.3**	64	N/A	N/A	926	601	N/A	N/A
YouTube	**1.7**	49	N/A	N/A	177	132	N/A	N/A
Network Intrusion	**2.8**	31	N/A	N/A	148	99	N/A	N/A

$1,000$ randomly sampled pairwise constraints. All the experiments are performed on a server with Intel Xeon 2.4 GHz processor and 64 GB of main memory. Each experiment is repeated ten times, and the average clustering performances are reported in Figure 6.4.

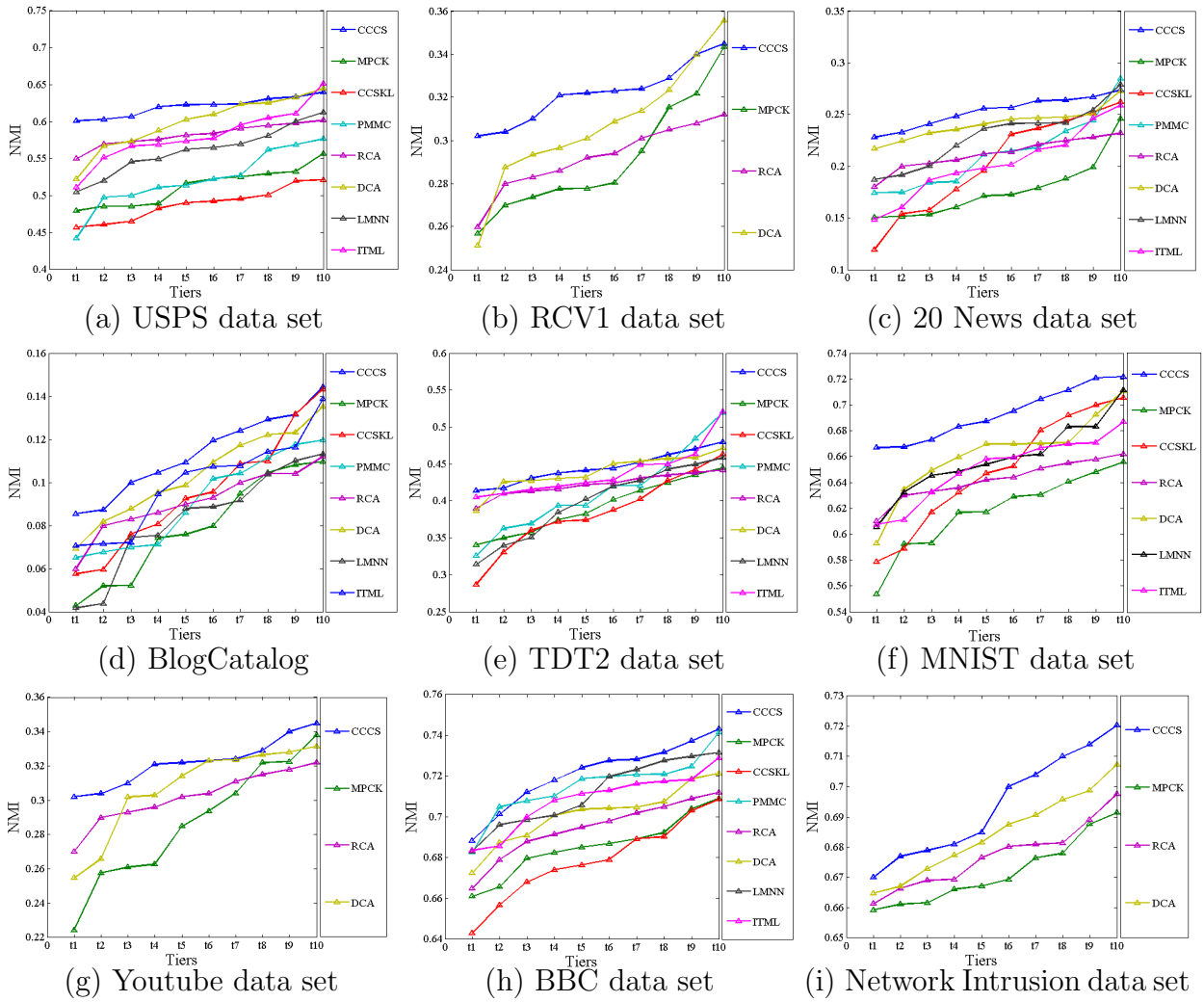
We first observe that, compared to all the baseline algorithms, the proposed dynamic semi-supervised clustering algorithm CCCS yields the best performance for most of the datasets (USPS, RCV1, 20 News, BlogCatalog, MNIST, Youtube and Network Intrusion). Specifically, when given a small number of the pariwise constraints, our method outperform the compared methods with significant performance gain. This is because that by generating a convex hull from a set of ensemble partitions, we dramatically reduce the possible searching space and it is expected that all the inner points in that convex hull can correspond to reasonably good data partitions. Also, by computing the starting point $\boldsymbol{\gamma}^0$ based on the idea of ensemble clustering, it is not surprising that $\boldsymbol{\gamma}^0$ should already be close to the optimal solution. Thus a simple locally search is good enough to recover the optimal partition.

In addition to superior performance, the proposed CCCS algorithm is also extremely efficient, requiring much less running time compared to all the baseline algorithms. Table 6.1 summarizes the average running time to update the data partitions in the four largest datasets, namely 20 News, RCV1, YouTube and Network Intrusion. We mark the results as N/A if an algorithm cannot output the results within 5 hours. By comparing the results in

Table 6.1, we observe that the running time of the proposed dynamic semi-supervised clustering algorithm to update the partition is significantly less than all the baseline algorithms. In particular, the results show that even for updating the partition of about 5 million data points, it only takes the proposed algorithm less than 3 seconds.

6.4 Conclusions

In this chapter, we propose a dynamic semi-supervised clustering algorithm that is able to efficiently update the clustering results when new pairwise constraints are generated. The key idea is to cast the clustering process into the problem of searching in a feasible clustering space, i.e., a convex hull generated from multiple ensemble partitions. Based on ensemble clustering, the optimal partition should lie in the inner space of this convex hull. Since every inner point of a convex hull can be uniquely represented by a simplex vector, the dynamic semi-supervised clustering problem can be reduced to the problem of learning a low-dimensional vector. Given a new set of pairwise constraints, we derive an efficient updating scheme that is able to learn an optimal simplex vector in a constant time. Our empirical studies conducted on multiple real-world datasets verify both the effectiveness and efficiency of the proposed algorithm.



(a) USPS data set (b) RCV1 data set (c) 20 News data set

(d) BlogCatalog (e) TDT2 data set (f) MNIST data set

(g) Youtube data set (h) BBC data set (i) Network Intrusion data set

Figure 6.4 Average clustering performance of the proposed dynamic semi-supervised clustering algorithm (CCCS) and the baseline algorithms (MPCKmeans (MPCK) [12], CCSKL [88], PMMC [144], RCA [6] DCA [63], LMNN [130], and ITML [36]) from tier t_1 to t_{10} on nine datasets.

Chapter 7

Conclusions and Future Directions

In this thesis, we study the issue of data clustering with pairwise constraints. Three important problems are considered, namely crowdclustering, semi-crowdsourced clustering, and semi-supervised clustering. In this chapter, we first review our main contributions to each problem, then point out a few possible directions for future research..

7.1 Summary of Main Results

In previous chapters, we have presented a matrix completion based approach for addressing the problem of crowdclustering. In addition, we solved the problem of semi-crowdsourced clustering based on the proposed robust distance metric learning approach. Furthermore, we developed an input pattern assisted matrix completion to address the semi-supervised clustering problem. Finally, we proposed a semi-supervised clustering algorithm that can efficiently update clustering results when new pairwise constraints are generated. In below, we summarize each of them.

- In Chapter 3, we present a matrix completion based framework to solve the problem of crowdclustering. To address the issue of high noise levels in crowdsourced annotations,

we first identify a subset of pairwise constraints that are agreed upon by most crowd workers. We then use these reliable pairwise constraints as the seed to derive the full similarity matrix by exploiting a matrix completion algorithm. Then the final data partition can be obtained by applying a spectral clustering algorithm to the completed similarity matrix. Different from the previous Bayesian crowdclustering approach that requires a sufficiently large number of manual annotations to discover the hidden factors, the proposed algorithm needs only a small number of manual annotations to obtain an accurate data partition. In addition, by filtering out the uncertain pairwise constraints collected from crowd workers, the proposed crowdclustering algorithm yield more robust clustering result.

- In Chapter 4, we focus on the problem of semi-crowdsourced clustering. This is motivated by a practical consideration that the classical crowdclustering problem can only cluster objects when their manual annotations are available. This significantly limits its application to large scale clustering problems since it is not feasible to have each object manually annotated by multiple workers when the number of objects is large. To address this issue, we at first randomly sample a small subset of objects for human annotation, and construct a partially observed similarity matrix based on the reliable annotations. We then exploit a matrix completion algorithm to recover the similarity matrix, and finally learn a linear similarity function to compute the similarities between all the objects. To the best of our knowledge, it is the first semi-crowdsourced clustering approach been proposed.

- In Chapter 5, we aim to address two main shortcomings with the existing semi-supervised clustering algorithms. First, most semi-supervised clustering algorithms

suffer from non-convexity optimization problems, leading to only locally optimal clustering results. Second, none of the existing approaches analyze the theoretical guarantee on sample complexity problem, i.e., what is the minimal number of pairwise constraints needed to accurately recover the underlying true partitions? To address such issues, we propose a framework for semi-supervised clustering based on the input pattern assisted matrix completion. Under the assumption that cluster membership vectors can be well approximated by the top few singular vectors of the data matrix, we cast the problem of semi-supervised clustering into a convex matrix completion problem. We then solve it efficiently by exploiting the stochastic subgradient descent method. Our theoretical analysis shows that only $O(\log N)$ pairwise constraints are needed to accurately recover the true cluster partition of N objects. The logarithmic dependence on N makes the proposed algorithm particularly suitable for clustering large-scale data sets.

- In Chapter 6, we focus on the problem of efficiently updating data partitions upon receiving some newly generated pairwise constraints, which is a common case in numerous real-world applications. Traditional semi-supervised clustering algorithms are not suitable to handle this problem since they need to re-optimize their objective functions over all the data points subject to all the pairwise connections, rendering high computational costs when data sets are large. To address this issue, we proposed an efficient algorithm that cast the problem of semi-supervised clustering into a searching problem in a convex hull, whose extreme points are an ensemble of multiple data partitions. We show that the time complexity of this algorithm is irrelevant to the number of data points to be clustered. This enables us to update large-scale clustering results in an extremely efficient way.

7.2 Future Directions

In addition to sample complexity and time complexity discussed in this thesis, both crowdclustering and semi-supervised clustering still remain many interesting open questions that worth to be considered in the future. We discuss three such research directions.

- Spammers and multi-objectives detection in crowdclustering

  Note that the quality of crowdsourced annotations is usually very difficult to control. As indicated by [141], more than 80% of crowdsourced pairwise labels can be inconsistent with the true cluster assignment. Very often that the crowdsourced annotations can be dominated by spammers, who tend to assign labels randomly. A large amount of spammers can significantly increase the labeling costs, and also degrade the overall quality of the crowdsourced annotations. Thus how to separate spammers from good users is a very useful and important application.

  Think along this line, even among the good users, people may not share only one view to partition the objects. Instead, they can have multiple objectives to cluster the data with all of them meaningful and valid. A typical example is the clustering of face images as illustrated in chapter 1. Both the clustering criteria that based on age and based on gender are equally valid, although their clustering results are orthogonal to each other. Compared to spammer detection, learning multiple objectives of workers is probably a even more challenging problem. This is due to the reason that spammers tend to assign random labels since they usually annotate instances without truly looking at them. This enables us to detect spammers by considering the consistency of their annotations. In contrast, the objectives of crowd workers are hidden and unknown,

and they need to be learned from the crowd annotations. However, since the number of pairwise constraints provided by each worker is usually limited, it becomes a problem of learning multiple hidden variables based on sparse labels. This is definitely a challenge but very interesting research problem.

- Active data clustering by matrix completion

  In all the aforementioned matrix completion based clustering algorithms, we assume that the pairwise constraints are generated randomly. This is consistent with the classical matrix completion theory [20] that the observed entries are randomly sampled. However, it is expected that by actively selecting the most informative pairs of objects for querying, we have chance to achieve an even lower sample complexity. The research in this field may lead to a profound development to the theory of matrix completion.

- Semi-supervised clustering with dynamic instances

  In chapter 6, we presented a dynamic semi-supervised clustering algorithm that is able to efficiently update clustering results given new pairwise constraints. However, in many real-world applications, both the instances and pairwise constraints can be dynamic. Take Facebook as an example, on average 5 new accounts and hundreds of connections are created in every second [1]. This demands a semi-supervised clustering algorithm that is able to update clustering results with both new instances and pairwise constraints.

[1] http://www.iacpsocialmedia.org/Resources/FunFacts.aspx

APPENDIX

In order to prove Theorem 1, we aim to construct Q that satisfies the following conditions: (a) $Q = \mathcal{P}_{\mathcal{S}}(Q)$, (b) $\mathcal{P}_T(Q) = UV^\top$, (c) $\|\mathcal{P}_{T^\top}(Q)\| < 1$, (d) $\mathcal{P}_\Omega(Q) = \gamma \text{sgn}(\mathcal{P}_\Omega(A))$, and (e) $|\mathcal{P}_{\Omega^c}(Q)|_\infty < \gamma$. We first provide the following proposition.

Proposition 1. *Let Ω be a set of m entries sampled uniformly from $[N]\times[N]$ according to the Bernoulli model, and $\mathcal{P}_\Omega(Z)$ projects Z onto the subset Ω. Let $\beta > 1$. Assume $m > m_0$, where $m_0 = C_R^2 \mu_0 r n \beta \log n$ and C_R is some positive constant. Then, with probability $1 - 3N^{-\beta}$, for any $Z \in T$ with $\mathcal{P}_\Omega(Z) = 0$, we have $Z = 0$.*

Let T be the space spanned by the elements of the form $\mathbf{u}_k\mathbf{y}^\top$ and $\mathbf{x}\mathbf{v}_k^\top$, for $1 \le k \le r$, where $\mathbf{x}$ and $\mathbf{y}$ are arbitrary, and let $T^\perp$ be the orthogonal complement to space T. First, according to Theorem 2 of [23], when $N > m_0$, where m_0 is defined in Proposition 1, with a probability at least $1 - 3N^{-\beta}$, mapping $\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T(Z) : T \mapsto T$ is one to one mapping and therefore its inverse mapping, denoted by $(\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T)^{-1}$ is well defined. Similar to the proof for Theorem 2 in [23], we construct the dual certificate Q as follows

$$Q = \lambda \text{sgn}(N^*) + \epsilon_\Omega + \mathcal{P}_{\mathcal{S}}\mathcal{P}_T(\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T)^{-1}(UV^\top + \epsilon_T)$$

where $\epsilon_T \in T$ and $\epsilon_\Omega = \mathcal{P}_\Omega(\epsilon_\Omega)$. We further define

$$\begin{aligned} H &= \mathcal{P}_{\mathcal{S}}\mathcal{P}_T(\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T)^{-1}(UV^\top) \\ F &= \mathcal{P}_{\mathcal{S}}\mathcal{P}_T(\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T)^{-1}(\epsilon_T) \end{aligned}$$

Evidently, we have $\mathcal{P}_{\mathcal{S}}(Q) = Q$ since $\Omega \subset \mathcal{S}$, and therefore (a) is satisfied. To satisfy (b)-(e), we need

$$\mathcal{P}_T(Q) = UV^\top \rightarrow \epsilon_T = -\mathcal{P}_T(\lambda \text{sgn}(N^*) + \epsilon_\Omega) \tag{1}$$

$$\|\mathcal{P}_{T^\perp}(Q)\| < 1 \rightarrow \mu(N^*)\left(\gamma + |\epsilon_\Omega|_\infty\right) + \|\mathcal{P}_{T^\perp}(H)\| + \|\mathcal{P}_{T^\perp}(F)\| < 1 \tag{2}$$

$$\mathcal{P}_\Omega(Q) = \lambda \text{sgn}(N^*) \rightarrow \epsilon_\Omega = -\mathcal{P}_\Omega(H + F) \tag{3}$$

$$|\mathcal{P}_{\Omega^c}(Q)|_\infty < \lambda \rightarrow \xi(M^*)(1 + \|\epsilon_T\|) < \lambda \tag{4}$$

Below, we will first show that there exist solutions to $\epsilon_T \in T$ and ϵ_Ω that satisfy conditions (1) and (3). We will then bound $|\epsilon_\Omega|_\infty$, $\|\epsilon_T\|$, $\|\mathcal{P}_{T^\perp}(H)\|$, and $\|\mathcal{P}_{T^\perp}(F)\|$ to show that with sufficiently small $\mu(N^*)$ and $\xi(M^*)$, and appropriately chosen λ, conditions (2) and (4) can be satisfied.

First, we show the existence of ϵ_Ω and ϵ_T that obey the relationships in (1) and (3). It is equivalent to show there exists ϵ_T that satisfies the following relation

$$\epsilon_T = -\mathcal{P}_T(\lambda \text{sgn}(N^*)) + \mathcal{P}_T\mathcal{P}_\Omega(H) + \mathcal{P}_T\mathcal{P}_\Omega\mathcal{P}_T(\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T)^{-1}(\epsilon_T)$$

or

$$\mathcal{P}_T\mathcal{P}_{\mathcal{S}\setminus\Omega}\mathcal{P}_T(\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T)^{-1}(\epsilon_T) = -\mathcal{P}_T(\lambda \text{sgn}(N^*)) + \mathcal{P}_T\mathcal{P}_\Omega(H)$$

Similar to the previous argument, when $|\mathcal{S}\setminus\Omega| = N - m > m_0$, with a probability $1 - 3N^{-\beta}$, $\mathcal{P}_T\mathcal{P}_{\mathcal{S}\setminus\Omega}\mathcal{P}_T(Z) : T \mapsto T$ is one to one mapping, and therefore $(\mathcal{P}_T\mathcal{P}_{\mathcal{S}\setminus\Omega}\mathcal{P}_T(Z))^{-1}$ is well

defined. Using this result, we have the following solution to the above equation

$$\epsilon_T \;=\; \mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T(\mathcal{P}_T\mathcal{P}_{\mathcal{S}\backslash\Omega}\mathcal{P}_T)^{-1}\left(-\mathcal{P}_T(\lambda \mathrm{sgn}(N^*)) + \mathcal{P}_T\mathcal{P}_\Omega(H)\right)$$

We now bound $\|\epsilon_T\|$ and $|\epsilon_\Omega|_\infty$. Since $\|\epsilon_T\| \leq |\epsilon_T|_F$, we will bound $|\epsilon|_F$. First, according to Corollary 3.5 in [15], under the assumption of this theorem, when $\beta = 4$ and with a probability $1 - N^{-3}$, for any $Z \in T$, we have

$$\left|\mathcal{P}_{T^\perp}\mathcal{P}_{\mathcal{S}}\mathcal{P}_T(\mathcal{P}_T\mathcal{P}_{\mathcal{S}}\mathcal{P}_T)^{-1}(Z)\right|_F \leq |Z|_F.$$

Using this result, we have

$$\begin{aligned}
|\epsilon_\Omega|_\infty &\leq \xi(M^*)\left(\|H\| + \|F\|\right) \\
&\leq \xi(M^*)\left(1 + |\mathcal{P}_{T^\perp}(H)|_F + \|\epsilon_T\| + |\mathcal{P}_{T^\perp}(F)|_F\right) \\
&\leq \xi(M^*)\left(2 + \|\epsilon_T\| + |\epsilon_T|_F\right) \\
&\leq \xi(M^*)\left(2 + (2r+1)\|\epsilon_T\|\right)
\end{aligned}$$

In the last step, we use the fact $\mathrm{rank}(\epsilon_T) \leq 2r$ if $\epsilon_T \in T$. We then proceed to bound $\|\epsilon_T\|$ as follows

$$\|\epsilon_T\| \;\leq\; \mu(N^*)\left(\lambda + |\epsilon_\Omega|_\infty\right)$$

Combining the above two inequalities together, we have

$$\begin{aligned} \|\epsilon_T\| &\leq \xi(M^*)\mu(N^*)(2r+1)\|\epsilon_T\| + 2\xi(M^*)\mu(N^*) + \lambda\mu(N^*) \\ |\epsilon_\Omega|_\infty &\leq \xi(B)\left(2 + (2r+1)\mu(N^*)(\lambda + |\epsilon_\Omega|_\infty)\right) \end{aligned}$$

leading to

$$\begin{aligned} \|\epsilon_T\| &\leq \frac{\lambda\mu(N^*) + 2\xi(M^*)\mu(N^*)}{1-(2r+1)\xi(M^*)\mu(N^*)} \\ |\epsilon_\Omega|_\infty &\leq \frac{2\xi(M^*) + (2r+1)\lambda\xi(M^*)\mu(N^*)}{1-(2r+1)\xi(M^*)\mu(N^*)} \end{aligned}$$

Using the bound for $|\epsilon_\Omega|_\infty$ and $\|\epsilon_T\|$, we now check condition (2)

$$1 > \mu(N^*)\left(\lambda + |\epsilon_\Omega|_\infty\right) + \frac{1}{2} + \frac{r}{2}\|\epsilon_T\|$$

or

$$\lambda < \frac{1-\xi(M^*)\mu(N^*)(4r+5)}{\mu(N^*)(r+2)}$$

For condition (4), we have

$$\lambda > \xi(M^*) + \xi(M^*)\|\epsilon_T\|$$

or

$$\lambda > \frac{\xi(M^*) - (2r-1)\xi^2(M^*)\mu(N^*)}{1-2(r+1)\xi(M^*)\mu(N^*)}$$

To ensure that there exists $\lambda \geq 0$ satisfies the above two conditions, we have

$$1 - 5(r+1)\xi(M^*)\mu(N^*) + (10r^2+21r+8)[\xi(M^*)\mu(N^*)]^2 > 0$$

and

$$1 - \xi(M^*)\mu(N^*)(4r + 5) \geq 0$$

Since the first condition is guaranteed to be satisfied for $r \geq 1$, we have

$$\mu(N^*)\xi(M^*) \leq \frac{1}{4r + 5}$$

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Alekh Agarwal, Sahand Negahban, and Martin J. Wainwright. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. In *ICML*, pages 1129–1136, 2011.

[2] Kais Allab and Khalid Benabdeslem. Constraint selection for semi-supervised topological clustering. In *ECML/PKDD*, pages 28–43, 2011.

[3] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The semeval-2007 weps evaluation: Establishing a benchmark for the web people search task. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. ACL, 2007.

[4] H Avron, S Kale, S Kasiviswanathan, and V Sindhwani. Efficient and practical stochastic subgradient descent for nuclear norm regularization. In *ICML*, 2012.

[5] Hanan Ayad and Mohamed S. Kamel. Cumulative voting consensus method for partitions with variable number of clusters. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(1):160–173, 2008.

[6] Aharon Bar-Hillel, Tomer Hertz, Noam Shental, and Daphna Weinshall. Learning a mahalanobis metric from equivalence constraints. *JMLR*, 6:937–965, 2005.

[7] P. Bartlett. The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Transactions on Information Theory*, 44(2):525–536, 1998.

[8] Sugato Basu, Mikhail Bilenko, and Raymond J. Mooney. A probabilistic framework for semi-supervised clustering. In *KDD*, pages 59–68, 2004.

[9] R. Bekkerman and M. Sahami. Semi-supervised clustering using combinatorial MRFs. In *ICML Workshop on Learning in Structured Output Spaces*, 2006.

[10] Asa Ben-Hur, André Elisseeff, and Isabelle Guyon. A stability based method for discovering structure in clustered data. In *Pacific Symposium on Biocomputing*, pages 6–17, 2002.

[11] S. K. Bhatia and J. S. Deogun. Conceptual clustering in information retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 28(3):427–436, 1998.

[12] M Bilenko, S Basu, and R J. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *ICML*, 2004.

[13] M. Bilenko and R. J. Mooney. Adaptive duplicate detection using learnable string

similarity measures. In *KDD*, pages 39–48, 2003.

[14] Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5, 2011.

[15] E J. Candès and T Tao. The power of convex relaxation: near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5):2053–2080, 2010.

[16] Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11, 2011.

[17] Emmanuel J. Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[18] Emmanuel J. Candès and Benjamin Recht. Simple bounds for low-complexity model reconstruction. *CoRR*, abs/1106.1474, 2011.

[19] Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Commun. ACM*, 55(6):111–119, 2012.

[20] Emmanuel J. Candès and Terence Tao. The power of convex relaxation: near-optimal matrix completion. volume 56, pages 2053–2080, 2010.

[21] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *KDD*, pages 554–560, 2006.

[22] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. volume 21, pages 572–596, 2011.

[23] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[24] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, S. Willsky, and Alan. Rank-sparsity incoherence for matrix decomposition. *CoRR*, 2009.

[25] Wen-Yen Chen, Yangqiu Song, Hongjie Bai, Chih-Jen Lin, and Edward Y. Chang. Parallel spectral clustering in distributed systems. *PAMI*, 33(3):568–586, 2011.

[26] Y. Chi, X. Song, D. Zhou, K. Hino, and B. L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *KDD*, pages 153–162, 2007.

[27] R. Chitta, R. Jin, T. C. Havens, and A. K. Jain. Approximate kernel k-means: solution to large scale kernel clustering. In *KDD*, pages 895–903, 2011.

[28] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user feedback. *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, 4(1):17–32, 2003.

[29] T. Cover and J. Thomas. *Elements of Information Theory (2nd ed.)*. Wiley, 2006.

[30] Ian Davidson and S. S. Ravi. Agglomerative hierarchical clustering with constraints: Theoretical and empirical results. In *PKDD*, pages 59–70, 2005.

[31] Ian Davidson and S. S. Ravi. Clustering with constraints: Feasibility issues and the k-means algorithm. In *SDM*, 2005.

[32] Ian Davidson and S. S. Ravi. Hierarchical clustering with constraints: Theory and practice. In *Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 59–70, 2005.

[33] Ian Davidson and S. S. Ravi. Identifying and generating easy sets of constraints for clustering. In *AAAI*, pages 336–341, 2006.

[34] Ian Davidson and S. S. Ravi. The complexity of non-hierarchical clustering with instance and cluster level constraints. *Data Min. Knowl. Discov.*, 14(1):25–61, 2007.

[35] Ian Davidson and Basu Sugato. A survey of clustering with instance level constraints. Technical report, 2007.

[36] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007.

[37] Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proceedings of the 24th Annual International Conference on Machine Learning*, pages 209–216, 2007.

[38] J.V. Davis, B. Kulis, P. Jain, S. Sra, and I.S. Dhillon. Information-theoretic metric learning. In *ICML*, pages 209–216, 2007.

[39] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[40] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Kernel k-means: spectral clustering and normalized cuts. In *KDD*, pages 551–556, 2004.

[41] Evgenia Dimitriadou, Andreas Weingessel, and Kurt Hornik. A combination scheme for fuzzy clustering. pages 332–338, 2002.

[42] Carlotta Domeniconi and Muna Al-Razgan. Weighted cluster ensembles: Methods and

analysis. *TKDD*, 2(4), 2009.

[43] J. C. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *ICML*, pages 272–279, 2008.

[44] Sandrine Dudoit and Jane Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

[45] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[46] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, 2005.

[47] Xiaoli Zhang Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *ICML*, pages 186–193, 2003.

[48] Xiaoli Zhang Fern and Carla E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, 2004.

[49] X.Z. Fern and C.E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *ICML*, page 36. ACM, 2004.

[50] Bernd Fischer and Joachim M. Buhmann. Bagging for path-based clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(11):1411–1415, 2003.

[51] Charless Fowlkes, Serge Belongie, Fan R. K. Chung, and Jitendra Malik. Spectral grouping using the nyström method. *PAMI*, 26(2):214–225, 2004.

[52] Lucas Franek and Xiaoyi Jiang. Ensemble clustering by means of clustering embedding in vector spaces. *Pattern Recognition*, 47(2):833–842, 2014.

[53] Ana L. N. Fred and Anil K. Jain. Data clustering using evidence accumulation. In *International Conference on Pattern Recognition*, volume 4, pages 276–280, 2002.

[54] Ana L. N. Fred and Anil K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(6):835–850, 2005.

[55] Hichem Frigui and Raghu Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *PAMI*, 21(5):450–465, 1999.

[56] Arvind Ganesh, John Wright, Xiaodong Li, Emmanuel J. Candès, and Yi Ma. Dense error correction for low-rank matrices via principal component pursuit. In *ISIT*, pages

1513–1517, 2010.

[57] D. Goldberg, D. A. Nichols, B. M. Oki, and D. B. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

[58] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *NIPS*, 2004.

[59] Jacob Goldberger, Sam Roweis, Geoff Hinton, and Ruslan Salakhutdinov. Neighbourhood components analysis. In *Proceedings of the 18nd Annual Conference on Neural Information Processing Systems*, pages 513–520, 2004.

[60] R. Gomes, P. Welinder, A. Krause, and P. Perona. Crowdclustering. In *NIPS*, 2011.

[61] Ronald L. Graham. An efficient algorith for determining the convex hull of a finite planar set. *Information processing letters*, 1(4):132–133, 1972.

[62] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

[63] S.C.H. Hoi, W. Liu, M.R. Lyu, and W.Y. Ma. Learning distance metrics with contextual constraints for image retrieval. In *CVPR*, pages 2072–2078, 2006.

[64] Steven C. H. Hoi, Wei Liu, Michael R. Lyu, and Wei-Ying Ma. Learning distance metrics with contextual constraints for image retrieval. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2072–2078, 2006.

[65] Daniel Hsu, Sham M. Kakade, and Tong Zhang. Robust matrix decomposition with sparse corruptions. *IEEE Transactions on Information Theory*, 57(11):7221–7234, 2011.

[66] Kaizhu Huang, Rong Jin, Zenglin Xu, and Cheng-Lin Liu. Robust metric learning by smooth optimization. In *UAI*, 2010.

[67] Yifen Huang and Tom M. Mitchell. Text clustering with extended user feedback. In *SIGIR*, pages 413–420, 2006.

[68] J.J. Hull. A database for handwritten text recognition research. *PAMI*, 16(5):550–554, 1994.

[69] Natthakan Iam-on, Tossapon Boongoen, and Simon Garrett. Refining pairwise similarity matrix for cluster ensemble problem with cluster relations. In *Discovery Science*, pages 222–233, 2008.

[70] Panagiotis G. Ipeirotis. Analyzing the amazon mechanical turk marketplace. *ACM Crossroads*, 17(2):16–21, 2010.

[71] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[72] Anil K. Jain, Jung-Eun Lee, and Rong Jin. Tattoo-ID: Automatic tattoo image retrieval for suspect and victim identification. In *Advances in Multimedia Information Processing–PCM*, pages 256–265, 2007.

[73] Ali Jalali, Yudong Chen, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. In *ICML*, pages 1001–1008, 2011.

[74] S D. Kamvar, D Klein, and C D. Manning. Spectral learning. In *IJCAI*, pages 561–566, 2003.

[75] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, 2010.

[76] Raghunandan H. Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010.

[77] Min-Soo Kim and Jiawei Han. A particle-and-density based evolutionary clustering method for dynamic networks. *PVLDB*, 2(1):622–633, 2009.

[78] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

[79] Brian Kulis, Sugato Basu, Inderjit S. Dhillon, and Raymond J. Mooney. Semi-supervised graph clustering: a kernel approach. *Machine Learning*, 74(1):1–22, 2009.

[80] M Law, A P. Topchy, and A K. Jain. Model-based clustering with probabilistic constraints. In *SDM*, 2005.

[81] Y. LeCun and C. Cortes. MNIST handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 1998.

[82] David D. Lewis, Yiming Yang, Tony G. Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.

[83] Q Li and B Kim. Clustering approach for hybrid recommender system. In *Web Intelligence*, pages 33–38, 2003.

[84] Tao Li, Chris H. Q. Ding, and Michael I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Seventh*

IEEE International Conference on Data Mining, pages 577–582, 2007.

[85] Weizhong Li, Lukasz Jaroszewski, and Adam Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, 17(3):282–283, 2001.

[86] Xiaodong Li. Compressed sensing and matrix completion with constant proportion of corruptions. *CoRR*, abs/1104.1041, 2011.

[87] Yan Li, Jian Yu, Pengwei Hao, and Zhulin Li. Clustering ensembles based on normalized edges. pages 664–671, 2007.

[88] Zhenguo Li and Jianzhuang Liu. Constrained clustering by spectral kernel learning. In *ICCV*, pages 421–427, 2009.

[89] Yu-Ru Lin, Y. Chi, S. Zhu, H. Sundaram, and B. L. Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *WWW*, pages 685–694, 2008.

[90] Z. Lin, M. Chen, L. Wu, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *Arxiv preprint arXiv:1009.5055*, 2010.

[91] Xiaoyong Liu and W. Bruce Croft. Cluster-based retrieval using language models. In *SIGIR*, pages 186–193, 2004.

[92] Z. Lu and M. Á. Carreira-Perpiñán. Constrained spectral clustering through affinity propagation. In *CVPR*, 2008.

[93] Z. Lu and T. K. Leen. Semi-supervised learning with penalized probabilistic clustering. In *NIPS*, 2004.

[94] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, page 14. California, USA, 1967.

[95] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[96] Behrouz Minaei-Bidgoli, Alexander P. Topchy, and William F. Punch. Ensembles of partitions via data resampling. pages 188–192, 2004.

[97] D.C. Montgomery, E.A. Peck, and G.G. Vining. *Introduction to Linear Regression Analysis*, volume 49. John Wiley & Sons, 2007.

[98] Nayar and H. Murase. Columbia object image library: COIL-100. Technical Report

CUCS-006-96, Department of Computer Science, Columbia University, February 1996.

[99] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[100] Sahand Negahban and Martin J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13:1665–1697, 2012.

[101] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[102] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, pages 849–856, 2001.

[103] D. O'Callaghan, M. Harrigan, J. Carthy, and P. Cunningham. Network analysis of recurring youtube spam campaigns. In *ICWSM*, 2012.

[104] Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *CoRR*, abs/1202.1212, 2012.

[105] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[106] Benjamin Recht. A simpler approach to matrix completion. *JMLR*, 12:3413–3430, 2011.

[107] Aaron D. Shaw, John J. Horton, and Daniel L. Chen. Designing incentives for inexpert human raters. In *ACM Conference on Computer Supported Cooperative Work* [107], pages 275–284.

[108] Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall. Computing gaussian mixture models with em using equivalence constraints. In *NIPS*, 2003.

[109] Noam Shental, Tomer Hertz, Daphna Weinshall, and Misha Pavel. Adjustment learning and relevant component analysis. In *Proceedings of the 17th European Conference on Computer Vision*, pages 776–792, 2002.

[110] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

[111] Steve Smale and Ding-Xuan Zhou. Geometry on probability spaces. *Constr Approx*, 30:311–323, 2009.

[112] S. J Stolfo, W. Fan, W. Lee, A. Prodromidis, and P. K Chan. Cost-based modeling for

fraud and intrusion detection: results from the jam project. In *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, volume 2, pages 130–144. IEEE, 2000.

[113] Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *JMLR*, 3:583–617, 2002.

[114] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *KDD*, pages 687–696, 2007.

[115] Tomer Hertz Tomboy, Aharon Bar-hillel, and Daphna Weinshall. Boosting margin based distance functions for clustering. In *Proceedings of the 21st Annual International Conference on Machine Learning*, 2004.

[116] Alexander P. Topchy, Anil K. Jain, and William F. Punch. A mixture model for clustering ensembles. In *SDM*, 2004.

[117] Alexander P. Topchy, Anil K. Jain, and William F. Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1866–1881, 2005.

[118] Alexander P. Topchy, Behrouz Minaei-Bidgoli, Anil K. Jain, and William F. Punch. Adaptive clustering ensembles. In *ICPR*, pages 272–275, 2004.

[119] Kagan Tumer and Adrian K. Agogino. Ensemble clustering with voting active clusters. *Pattern Recognition Letters*, 29(14):1947–1953, 2008.

[120] Sandro Vega-Pons and José Ruiz-Shulcloper. Clustering ensemble method for heterogeneous partitions. In *CIARP*, pages 481–488, 2009.

[121] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *IJPRAI*, 25(3):337–372, 2011.

[122] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[123] K. Wagstaff and C. Cardie. Clustering with instance-level constraints. In *Proceedings of the 17th Annual International Conference on Machine Learning*, pages 1103–1110, 2000.

[124] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means clustering with background knowledge. In *Proceedings of the 18th Annual International Conference on Machine Learning*, 2001.

[125] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. Constrained k-means

clustering with background knowledge. In *ICML*, pages 577–584, 2001.

[126] Y. Wakabayashi. *Aggregation of binary relations: algorithmic and polyhedral investigations*. PhD thesis, 1986.

[127] X. Wang, L. Tang, H. Gao, and H. Liu. Discovering overlapping groups in social media. In *ICDM*, pages 569–578, 2010.

[128] X. Wang, C. Yang, and J. Zhou. Clustering aggregation by probability accumulation. *Pattern Recognition*, 42(5):668–675, 2009.

[129] K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Proceedings of the 20nd Annual Conference on Neural Information Processing Systems*, pages 207–244, 2006.

[130] K.Q. Weinberger, J. Blitzer, and L.K. Saul. Distance metric learning for large margin nearest neighbor classification. In *NIPS*, 2006.

[131] A. Weingessel, E. Dimitriadou, and K. Hornik. An ensemble method for clustering. In *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*. Citeseer, 2003.

[132] JD Maitland Wright. Measures with values in a partially ordered vector space. *Proceedings of the London Mathematical Society*, 3(4):675–688, 1972.

[133] Lei Wu, Rong Jin, Steven C.H. Hoi, Jinfeng Zhuang, and Nenghai Yu. Simplenpkl: Simple non-parametric kernel learning. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*, 2009.

[134] E.P. Xing, A.Y. Ng, M.I. Jordan, and S. Russell. Distance metric learning, with application to clustering with side-information. volume 15, pages 505–512, 2002.

[135] Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning, with application to clustering with side-information. In *Proceedings of the 17th Annual Conference on Neural Information Processing Systems*, pages 505–512, 2003.

[136] Huan Xu, Constantine Caramanis, and Sujay Sanghavi. Robust pca via outlier pursuit. *IEEE Transactions on Information Theory*, 58(5):3047–3064, 2012.

[137] Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. Technical report, 2006.

[138] Liu Yang and Rong Jin. Distance metric learning: A comprehensive survey. Technical report, Department of Computer Science and Engineering, Michigan State University,

2006.

[139] Liu Yang, Rong Jin, Rahul Sukthankar, and Yi Liu. An efficient algorithm for local distance metric learning. In *Proceedings of the 21st National Conference on Aartifical Intelligence*, pages 543–548, 2006.

[140] Ka Yee Yeung, Chris Fraley, Alejandro Murua, Adrian E Raftery, and Walter L Ruzzo. Model-based clustering and data transformations for gene expression data. *Bioinformatics*, 17(10):977–987, 2001.

[141] J. Yi, R. Jin, A. K. Jain, and S. Jain. Crowdclustering with sparse pairwise labels: A matrix completion approach. In *AAAI Workshop on Human Computation*, 2012.

[142] J. Yi, R. Jin, A. K. Jain, S. Jain, and T. Yang. Semi-crowdsourced clustering: Generalizing crowd labeling by robust distance metric learning. In *NIPS*, pages 1781–1789, 2012.

[143] J. Yi, L. Zhang, R. Jin, Q. Qian, and A. K. Jain. Semi-supervised clustering by input pattern assisted pairwise similarity matrix completion. In *ICML (3)*, pages 1400–1408, 2013.

[144] H. Zeng and Y. Cheung. Semi-supervised maximum margin clustering with pairwise constraints. *TKDE*, 24(5):926–939, 2012.