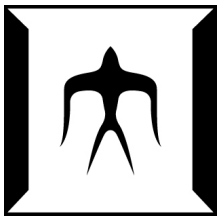
論文 / 著書情報
Article / Book Information

題目(和文)	
Title(English)	Discriminative Methods with Imperfect Supervision in Machine Learning
著者(和文)	NiuGang
Author(English)	Gang Niu
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第9330号, 授与年月日:2013年9月25日, 学位の種別:課程博士, 審査員:杉山 将,佐藤 泰介,秋山 泰,篠田 浩一,瀬々 潤,工藤 一浩
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第9330号, Conferred date:2013/9/25, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Discriminative Methods with Imperfect Supervision in Machine Learning

Gang Niu

September 2013



Department of Computer Science
Graduate School of Information Science and Engineering
Tokyo Institute of Technology

Thesis Committee:
Masashi Sugiyama, Chair
Taisuke Sato
Yutaka Akiyama
Koichi Shinoda
Jun Sese

Submitted in partial fulfillment of
the requirements for the degree of

Doctor of Engineering

Keywords: discriminative method, imperfect supervision, clustering, metric learning, semi-supervised learning

To my wife and my parents

Abstract

We have entered the age of information for more than two decades, and now we are entering the age of big data. Traditional supervised learning, which requires all labels for training a model, costs too many resources and thus becomes inapplicable nowadays. In vivid contrast, imperfectly supervised learning, which does not have such a requirement, is much less expensive and then more practical for real-world applications. To this end, this doctoral thesis is devoted to developing discriminative methods with imperfect supervision in machine learning.

We have found six major problem settings that can be viewed as learning with imperfect supervision. This thesis focuses on three of them. Three learning models each with one or two algorithms are proposed for certain learning problems, namely, clustering, metric learning, and semi-supervised classification.

Firstly, we investigate the clustering problem and propose *maximum volume clustering* (MVC). State-of-the-art clustering methods have many advantages, for example, the cluster shape can be very flexible. However, all of existing methods lack two important theoretical guarantees: Finite sample stability which analyzes when different local optima induce the same data partition, and clustering error bound which theoretically bound the clustering error from above. MVC employs the soft response vector as the hypothesis rather than the centroid or hyperplane, and is approximately solved by sophisticated optimization methods. Consequently, MVC is theoretically guaranteed with the finite sample stability and clustering error bound. Experiments demonstrate MVC is promising.

Secondly, we investigate the metric learning problem and propose *semi-supervised metric learning paradigm with hyper-sparsity* (SERAPH). State-of-the-art semi-supervised metric learning methods are all based on manifold regularization and manifold embedding. These methods can successfully extract the similarity

information of unlabeled data, but the dissimilarity information is simply ignored. Nonetheless, most unlabeled data should be dissimilar for rich enough input data domains. To this end, SERAPH learns a metric by learning a probability parameterized by that metric, while it employs entropy regularization so that it can also extract the dissimilarity information of unlabeled data. Experiments demonstrate SERAPH is promising.

Thirdly, we investigate the semi-supervised classification problem and propose *squared-loss mutual information regularization* (SMIR). Information maximization methods for semi-supervised classification, as the state-of-the-art methods, can directly deal with the multi-class out-of-sample classification problem. However, all of existing information maximization methods are non-convex, and thus they have no access to the globally optimal solution. SMIR replaces the ordinary mutual information with the squared-loss mutual information, and the optimization involved in SMIR is strongly convex under mild conditions and then has the analytic expression of the unique globally optimal solution. Experiments demonstrate SMIR is promising.

Given the encouraging experimental results of the proposed methods, we finally conclude that discriminative methods with imperfect supervision in machine learning are successful and worth a further study in the future.

Acknowledgments

First of all, I am deeply indebted to my academic supervisor, Professor Masashi Sugiyama, for his supervision in the last three years. He has provided me one of the nicest environments that I have ever seen for both research and study. Moreover, his valuable assistance and heartwarming encouragement was extraordinarily helpful, even though he was always unbelievably busy. I would surely like to express my gratitude to Professor Taisuke Sato, Professor Yutaka Akiyama, Professor Koichi Shinoda, and Professor Jun Sese for reviewing and evaluating my thesis.

Furthermore, my gratitude goes out to all the members of Sugiyama Laboratory, especially to the secretaries of our lab, Ms. Yasuyo Obana and Mrs. Ayako Tamai. They are the staff members who in fact run the lab every day. I would be lost immediately without their help whenever I go out and need to use Japanese. I am also grateful to Dr. Makoto Yamada, Dr. Hirotaka Hachiya, Dr. Ning Xie, Tingting Zhao, Song Liu, Marthinus Christoffel du Plessis, Wittawat Jitkrittum, and Akihiro Yamashita. I had fruitful discussions with them quite often in the last three years. I cannot forget to mention my old friend Bo Dai at Georgia Institute of Technology, with whom all collaborative research projects were tremendously successful.

My research projects were financially supported by the Japanese government MEXT scholarship (No. 103250) since October 2010. Without the MEXT scholarship, I could not afford my life in Japan, let alone my research. Hence, I would like to acknowledge Ministry of Education, Culture, Sports, Science and Technology for offering me the scholarship.

Last but not least, I would like to thank my wife and parents for their lasting support to me. They are the source of my power who motivate me for a Ph.D.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This doctoral thesis is devoted to developing discriminative methods with imperfect supervision in machine learning. In this chapter, we state the motivation and objective of our work.

1.1 Learning and Machine Learning

Learning is the activity of inferring certain unknown facts based on some known facts and some knowledge of the environment. When we talk about learning, we should have already implied in the context a *subject* who carries out the learning activity and an underlying *environment* where the subject of learning acts. Usually, learning is achieved by first generalizing a set of rules from given facts and knowledge and then applying these rules to infer the unknown facts. These rules obtained by learning can be further added to the knowledge of the environment, and thus the *object* of learning can be either the unknown facts or the knowledge of the environment itself.

Learning has many different internal motivations and external manifestations according to the various learning subjects. Sugiyama (2001) gives an illustrative example: Physicists who are interested in the underlying laws and principles of the world constitute a typical group of subjects, and their activity to clarify those laws and principles based on a limited amount of experimental data is a typical paradigm of learning. Hence, learning has actually been a very important issue in science.

If the subject of learning is a person, it is called *human learning*. Besides our human beings, animals such as dolphins, dogs, cats, and even bees and ants can also learn, and it is called *natural learning*. Surprisingly, besides these creatures, programs in computer systems can also learn, and it is called *machine learning*.

In the context of machine learning, the known facts are training data sampled from the environment, the unknown facts are usually unknown labels of test data, and the knowledge may be a particular regularization or a prior of parameters.

1.1.1 Three Major Issues in Learning

Learning is not an isolated research field. It is closely related to several research fields. Sugiyama (2001) has pointed out three major issues in learning:

- *Clarification of mechanism of brain*, which has been mainly studied in psychology, biology, and neuroscience;
- *Development of learning machines*, which has been mainly studied in computer science and neuro-engineering;
- *Investigation of essence of learning*, which has been mainly studied in information science.

In a special sense, machine learning refers to the third issue, while in a general sense, it covers all three issues. In this thesis, we adopt the special sense, that is, investigation of essence of learning. More specifically, the essence of learning is hidden in the answers of two questions (cf. Mitchell, 2006):

- How can computer systems *automatically* improve with experience?
- What are the fundamental laws that govern all learning methods?

Although we develop machine learning methods when we seek the answers, we do not develop learning machines, and our motivation is not to copy the way our human being thinks. We would like to argue that machine learning does not have to mimic human learning just like artificial intelligence does not have to mimic human intelligence (Russell and Norvig, 2009), as the way we think and behave is not necessarily what we want to be imitated by machines. Consider someone walks up to his "intelligent" car and says "Take me home." but it answers "Don't

you see I'm enjoying this beautiful and peaceful place? Take a cab!". Is anyone willing to buy it?

1.1.2 Machine Learning, Data Mining, and Statistics

Perhaps it is unclear what the difference of machine learning from *data mining* or *statistics* is, especially since nowadays many machine learning and data mining methods originate from the statistical learning theory (Vapnik, 1998).

Roughly speaking, machine learning shares a similar yet slightly different goal with data mining and statistics, and they diverge at what the unknown facts to be inferred are and how to evaluate the performance:

- Machine learning emphasizes *predicting the future*. It infers the unknown labels of test data and prefers the model with *higher accuracy*. In practice, the future is unknown, and thus the evaluation often relies on training data (e.g., through cross-validations). Both probabilistic methods (e.g., logistic regression) and non-probabilistic methods (e.g., support vector machines) are popular, and the classical methods are supervised.
- Data mining emphasizes *discovering novel knowledge*. Typically, it infers any unknown knowledge, while it does not have specific targets before it mines the data. Once the new knowledge has been discovered, the novelty is scored by the user. In such typical settings, the supervision is absent in mining, so the classical methods are unsupervised.
- Statistics emphasizes *understanding the past*. It infers the unknown mechanism that has generated the data and prefers the model with *better interpretability*. As a result, probabilistic methods exploring the physical meanings of features are more popular than non-probabilistic methods that look often like black boxes.

Note that, however, the clarification given above only works with the primary goals of the most representative tasks from the three fields in a traditional flavor. Nowadays, machine learning also studies clustering and anomaly detection, data mining also studies classification and regression, and besides analyses statistics also makes predictions. Consequently, in a modern flavor the difference between

machine learning, data mining and statistics is quite subtle, and the methodology to be followed mainly depends on the problem that we want to solve.

1.1.3 Machine Learning and Artificial Intelligence

Machine learning and *artificial intelligence* have been closely related since long before. The nuance between machine learning and artificial intelligence sounds even more subtle in a traditional flavor. However, nowadays they are in fact not significantly overlapped.

In artificial intelligence, the subject who acts in the environment is called the *agent* or the intelligent agent, and there are several key issues for the intelligent agent (Russell and Norvig, 2009):

- *Perceiving*, which collects the information from the environment. The collected information will be used for decision making;
- *Searching*, *planning*, and *learning*, which analyzes the output of perceiving and makes the decision as the input of acting;
- *Acting*, which executes the decision and interacts with the environment;
- *Communicating*, which exchanges the information with other agents, and thus assembles the agents into a society.

We can see that from this point of view, machine learning is a branch of artificial intelligence, and it is partially in charge of decision making. Despite that searching and planning can also be used for the same purpose, there are two intrinsic differences:

- Searching and planning rely more on the knowledge formulated by the designer, while learning relies more on the data sampled from the environment;
- Searching and planning are more logical reasoning, while learning is more statistical reasoning.

Recall the two questions about the essence of learning. The agent who can learn is able to *automatically* improve with experience, and hence it may take a better action at the same state after it explores the environment for a longer time. By

contrast, the agent who cannot learn is going to disregard the experience during the exploration of the environment, and hence it will take the same action at the same state forever.

Note that learning in artificial intelligence mainly means reinforcement learning which is overwhelmingly popular than other machine learning paradigms. In a modern flavor, machine learning includes lots of advanced data analysis topics so that it is much more general than the specific learning in artificial intelligence. Therefore, machine learning is regarded as an independent research field instead of a branch of artificial intelligence in the modern age.

1.2 Generative and Discriminative Methods

Let $\mathcal{X}$ and $\mathcal{Y}$ be the input domain and output range of our interests, and $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ be the input and output random variables respectively. Without loss of generality, assume that the Cartesian product $\mathcal{X} \times \mathcal{Y}$ has an underlying joint probability distribution $p(x, y)$ with the marginal density $p(x)$ and the marginal probability $p(y)$. In this thesis, $\mathcal{X}$ is always a continuous set and $\mathcal{Y}$ is always a discrete set.

A *generative model* is a model for either randomly generating the data with labels, i.e., drawing (x, y) according to $p(x, y)$, or randomly generating the data given labels, i.e., drawing x given y according to $p(x \mid y)$. A *generative method* in machine learning is a method that focuses on certain generative models. The goal is to estimate the hidden parameter θ, such that

$$\hat{p}(x, y; \theta) \approx p(x, y) = p(x \mid y)p(y).$$

Since our y is discrete, estimating the joint probability distribution $p(x, y)$ can be reduced to estimating the conditional probability density $p(x \mid y)$, and it can serve as an intermediate step to estimating the conditional probability $p(y \mid x)$ through Bayes' rule

$$p(y \mid x) = \frac{p(x \mid y)p(y)}{p(x)} = \frac{p(x \mid y)p(y)}{\sum_{y' \in \mathcal{Y}} p(x \mid y')p(y')}.$$

Examples of generative models are Gaussian mixture models and hidden Markov models.

A *discriminative model* is a model for analyzing the dependence of the labels on data, usually given observed data with labels yet sometimes given observed data only. A *discriminative method* in machine learning is a method that focuses on certain discriminative models. The goal is to estimate the hidden parameter θ, such that

$$\hat{p}(y \mid x; \theta) \approx p(y \mid x),$$

which can be used for predicting y after seeing x without the help of Bayes' rule. Alternatively, we can learn an unnormalized probability $q(y \mid x; \theta)$ such that

$$\hat{p}(y \mid x) = \frac{q(y \mid x; \theta)}{\sum_{y' \in \mathcal{Y}} q(y' \mid x; \theta)} \approx p(y \mid x).$$

Note that doing so in a generative method is not a good idea, since $p(y \mid x)$ is a discrete probability and the partition function

$$Z(x) = \sum_{y \in \mathcal{Y}} q(y \mid x; \theta)$$

is very easy to compute, whereas $p(x \mid y)$ is a continuous density and computing the partition function

$$Z(y) = \int_{x \in \mathcal{X}} q(x \mid y; \theta) \mathrm{d}x$$

itself can be a hard problem. Examples of discriminative models are support vector machines and logistic regression models.

Generative methods and discriminative methods have their own strong points and weak points. The differences all attribute to that generative methods use full probabilistic models of X and Y while discriminative methods use partial probabilistic models of Y conditioned on X. In other words, the randomness of generating x is not considered by discriminative methods though X is stochastic. As a result, generative methods can simulate the data generation including all forms (x, y), x, y, $x \mid y$ and $y \mid x$, while discriminative methods can only simulate the generation of $y \mid x$, and the results do not necessarily follow the same $p(x, y)$. In addition, generative models can generally express more complex dependence of X and Y than discriminative models. Nevertheless, for most complex dependence, generative models are very rough approximations to the true $p(x, y)$, and generative methods are less robust against incorrectly specified models than discriminative methods. Moreover, if the problem of interest is just predicting y for

given x and does not require $p(x, y)$, generative methods usually make more assumptions than discriminative methods, while discriminative methods only make the assumptions necessary to solve the problem at hand. In such cases, discriminative methods may yield superior performance. To sum up, the characteristics of the problem that we want to solve finally determine we should use generative methods or discriminative methods.

In this thesis, we simply focus on discriminative methods, since our problem of interest is no more than predicting y given x (Chapters 2 and 4) or predicting y given x and x' (Chapters 3). An issue of discriminative methods is that they are originally supervised methods and cannot be readily extended to the supervision other than the full supervision. Thus, we study discriminative methods with imperfect supervision. The imperfect supervision is a general name of many types of supervision less informative than the full supervision. It will be introduced in the next section.

1.3 Imperfect Supervision

The imperfect supervision can be referred to as many types of supervision which are less informative than the full supervision. In this section, we briefly explain fully supervised learning and six major imperfectly supervised learning:

- Supervised learning in Section 1.3.1;
- Unsupervised learning in Section 1.3.2;
- Semi-supervised learning in Section 1.3.3;
- Transductive learning in Section 1.3.4;
- Weak-supervised learning in Section 1.3.5;
- Active learning in Section 1.3.6;
- Reinforcement learning in Section 1.3.7.

By no means could we make a comprehensive review in a single section, so we just focus on their basic problem settings. For details, please refer to textbooks in machine learning such as Duda et al. (2001), Bishop (2006) and Hastie et al. (2009).

1.3.1 Supervised Learning

Undoubtedly, *supervised learning*, or what we call learning with full supervision here, is a best-studied type in machine learning compared with other types, particularly in statistical learning theory (Vapnik, 1998). It refers to the problem of inferring a function $y = f(x)$ based on *completely labeled training data*.

Here for the consistency of terminology, the term *training data* means the raw data to be used for training without labels, e.g.,

$$x_1, \ldots, x_n,$$

the term *labeled training data* means those raw data with the labels given by the domain expert, e.g.,

$$(x_1, y_1), \ldots, (x_n, y_n),$$

and *completely labeled training data* means that each raw datum possesses a label. Consequently, our terminology is slightly different from some commonly used ones in supervised learning where training data are our labeled training data. We distinguish here the raw and the labeled training data in order to clearly explain the data flow. We will abbreviate labeled training data to labeled data if there is no ambiguity.

The data flow of supervised learning is shown in Figure 1.1. The white boxes mean the data and labels, and the gray boxes mean certain subjects who interact with the data and labels. Moreover, the box with a red border means the primary goal of supervised learning. In Figure 1.1, the model is the key part. It can map any unseen test data to some reasonable predictions, and therefore it *generalizes* from a limited amount of training data to infinitely many test data.

More specifically, the training data flow starts at the data generator and ends at the model:

1. The data generator produces the training data $x_1, \ldots, x_n$ according to the marginal density $p(x)$;
2. The domain expert, who is the supervisor, labels all training data according to the conditional probability $p(y \mid x)$. That is, $\forall y \in \mathcal{Y}$, y is assigned to x_i with probability $p(y \mid x_i)$;

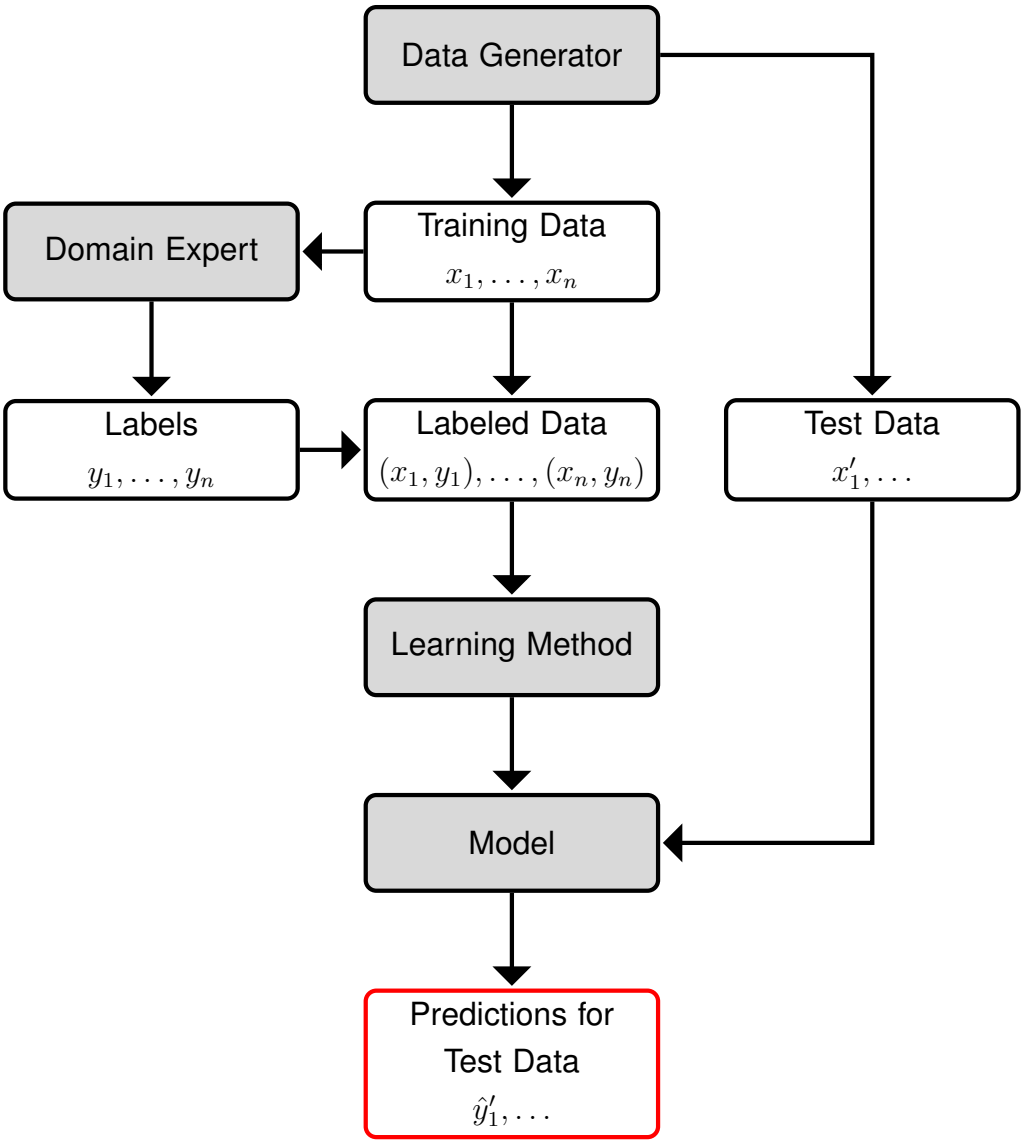


Figure 1.1: Data flow of supervised learning. The white boxes mean the data and labels, the gray boxes mean the subjects, and the box with a red border indicates the primary goal.

3. The learning method analyzes the labeled data $(x_1, y_1), \ldots, (x_n, y_n)$, and then builds the model $y = f(x)$.

After we have a model in hand, the test data flow starts at the data generator and ends at the prediction. For any test datum $x' \in \mathcal{X}$, the model predicts the label as $\hat{y}' = f(x')$, i.e., the output value of the learned function f given an input x'.

Supervised learning is the prototype and almost all other types are its variations. An understanding of supervised learning would help appreciating the imperfectly supervised problem settings.

1.3.2 Unsupervised Learning

As opposed to supervised learning reviewed in the last subsection, *unsupervised learning* refers to the problem of inferring a function $y = f(x)$ based on *completely unlabeled training data*. Since no supervision has been given, the learning method can only rely on the knowledge provided by the designer that frequently makes strong assumptions.

Unsupervised learning is often used for data analysis and unsupervised methods diverge very much according to various data analysis purposes. Some classical problems in unsupervised learning include but are not limited to:

- *Clustering*, where the output y is some discrete cluster assignment;
- *Density estimation*, where the output y is some real-valued scalar;
- *Dimensionality reduction*, where the output y is some real-valued vector.

Here, we simply use discriminative clustering where classifiers are trained in an unsupervised manner as an example to explain the data flow.

The data flow of unsupervised learning is shown in Figure 1.2. Similarly, the white boxes mean the data and labels, and the gray boxes mean certain subjects who interact with the data and labels. Unsupervised learning has two goals, i.e., making predictions for the training and test data respectively. In Figure 1.2, the primary goal is indicated by the box with a red border, while the secondary goal has a data flow composed of the dashed boxes and lines. If the learning method can achieve this secondary goal, we say that it has the *out-of-sample ability*. An *out-of-sample extension* is otherwise needed.

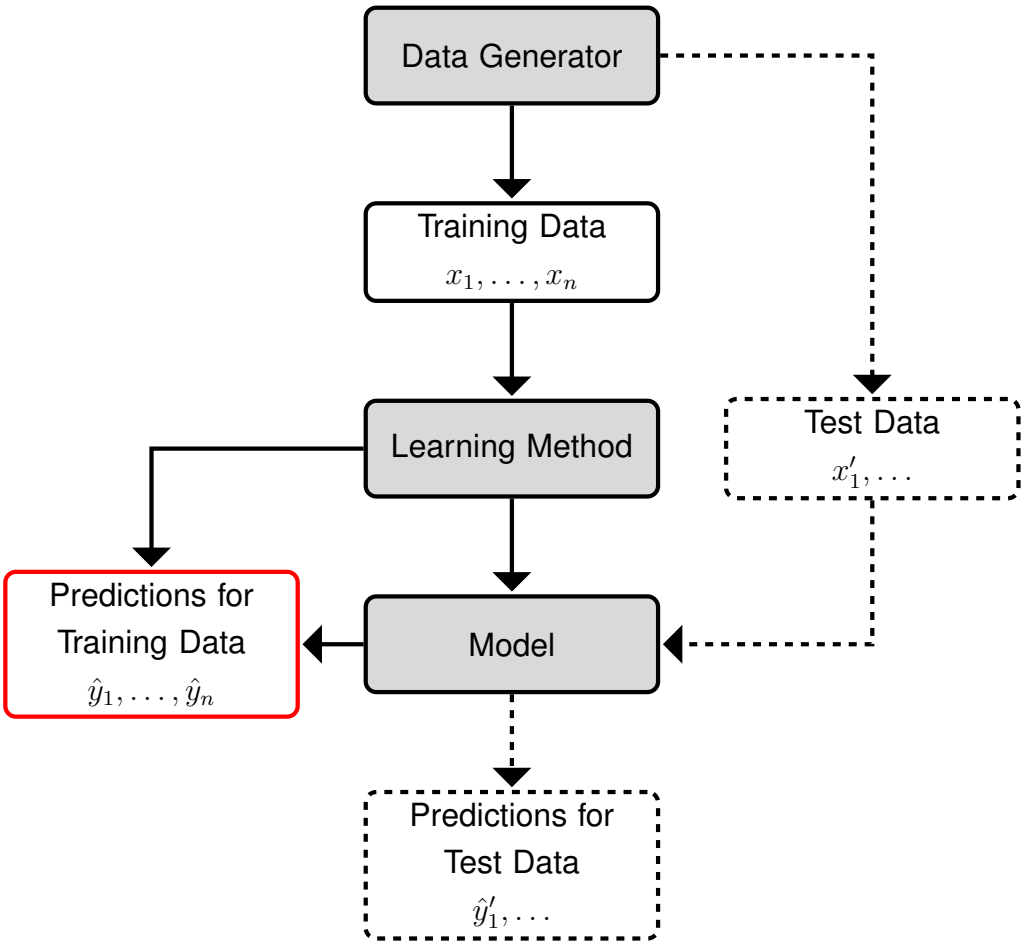


Figure 1.2: Data flow of unsupervised learning. The white boxes mean the data and labels, the gray boxes mean the subjects, and the box with a red border indicates the primary goal. The dashed boxes and lines compose the data flow for the secondary goal. In order to do so, the learning method must have the out-of-sample ability.

Compared with Figure 1.1, we can see from Figure 1.2 that there is no label for training since the domain expert is absent, and thus the training data directly go to the learning method. Then some learning methods build a model $y = f(x)$ explicitly and use it to determine the predictions $\hat{y}_1, \ldots, \hat{y}_n$, while some learning methods build a model implicitly as a by-product and determine the predictions without the model.

Unsupervised learning is an extreme in learning with imperfect supervision. Many imperfectly supervised problem settings can be obtained by cleverly combining supervised and unsupervised learning.

1.3.3 Semi-supervised Learning

Semi-supervised learning, as its name, is half supervised learning and half unsupervised learning. In semi-supervised learning, the domain expert labels a subset of training data while the learning method cannot control the data to be labeled. Then the learning method infers a function $y = f(x)$ based on *partially labeled training data*.

The data flow of semi-supervised learning is shown in Figure 1.3. It is similar to the data flow of supervised learning in Figure 1.1. Now, the training data flow starts at the data generator and ends at the prediction:

1. The data generator produces the training data $x_1, \ldots, x_n$ according to the marginal density $p(x)$;
2. The domain expert labels a subset of training data according to the conditional probability $p(y \mid x)$. That is, $\forall y \in \mathcal{Y}$, y is assigned to x_i with probability $p(y \mid x_i)$ if x_i is labeled. Without loss of generality, assume $x_1, \ldots, x_l$ are labeled, $x_{l+1}, \ldots, x_n$ are unlabeled, and l is usually much smaller than n;
3. The learning method analyzes the labeled data $(x_1, y_1), \ldots, (x_l, y_l)$ and the unlabeled data $x_{l+1}, \ldots, x_n$, and then builds the model $y = f(x)$;
4. The model sometimes predicts $\hat{y}_{l+1} = f(x_{l+1}), \ldots, \hat{y}_n = f(x_n)$, i.e., the output values of the learned function f given inputs $x_{l+1}, \ldots, x_n$.

Note that, however, the last step in the training data flow is the secondary goal of semi-supervised learning. Similarly to supervised learning, the model is the key

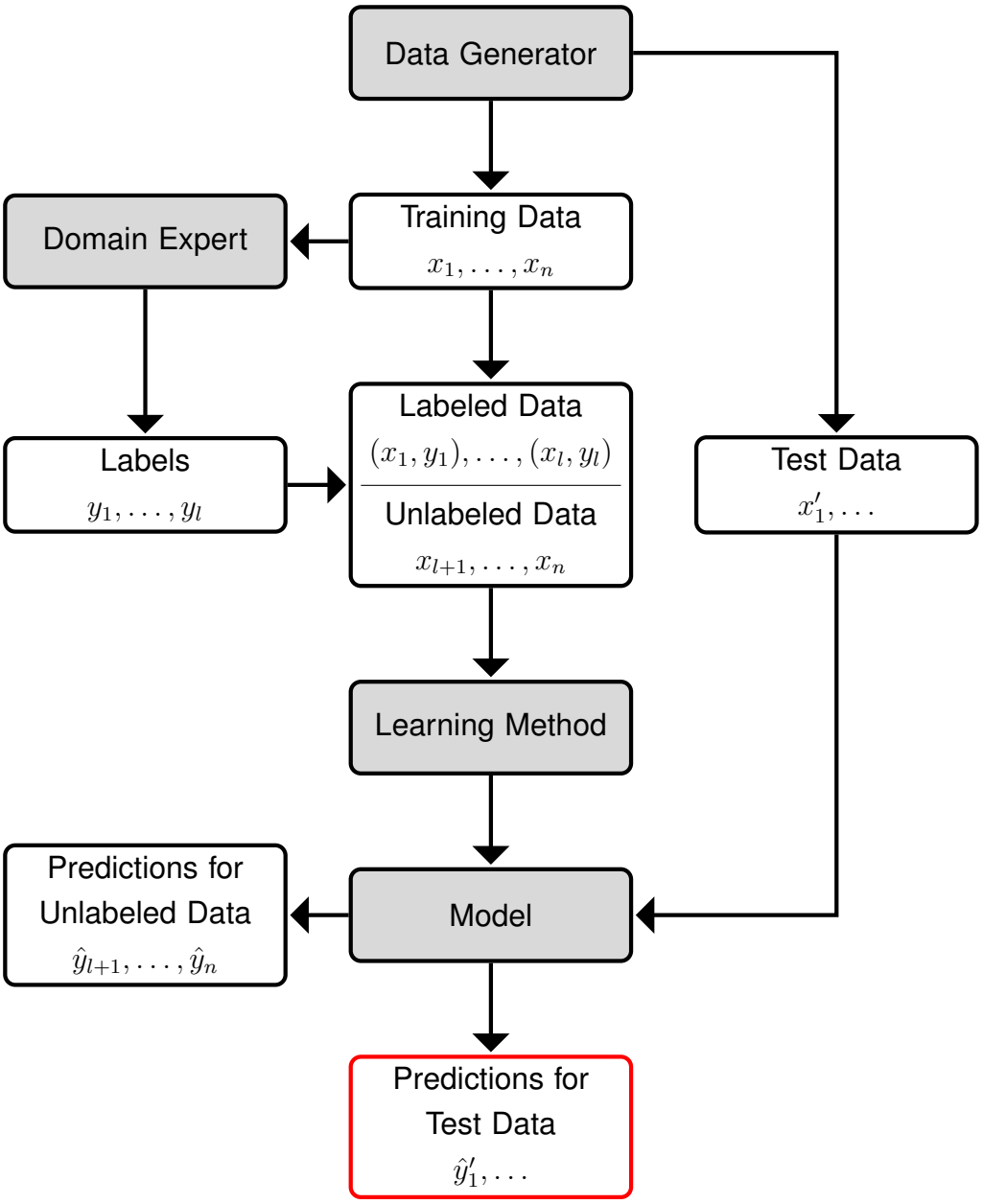


Figure 1.3: Data flow of semi-supervised learning. The white boxes mean the data and labels, the gray boxes mean the subjects, and the box with a red border indicates the primary goal.

part, which generalizes from a limited amount of training data to infinitely many test data. In Figure 1.3, the primary goal is the terminal of the test data flow and indicated by the box with a red border. After we have a model in hand, the model predicts $\hat{y}' = f(x')$ for any test datum $x' \in \mathcal{X}$.

For details of semi-supervised learning, please refer to Chapelle et al. (2006).

1.3.4 Transductive Learning

There has been two views of transductive learning, one is more similar to supervised learning and the other is more similar to semi-supervised learning. Both of them agree that *transductive learning* makes some predictions without building a model, but diverge at these predictions are for test data or for unlabeled training data.

Transductive learning is also called *transduction* or *transductive inference*. It is the opposite of *inductive learning*. It does not build any model so that it does not generalize from a limited amount of training data to infinitely many test data. The key idea of transductive learning is to infer from limited and specific training data directly to limited and specific test data. According to Vladimir Vapnik (e.g., Vapnik, 1998), transductive learning is preferable to inductive learning, since inductive learning requires solving a more general problem (inferring a function) as an intermediate step before solving a more specific problem (making predictions for test data).

Despite the philosophically advanced motivation of the above view of transductive learning, we adopt the other view which suggests that transductive learning makes predictions for unlabeled training data, since

- Transductive learning is a member of learning with imperfect supervision, when it makes predictions for unlabeled data;
- Based on the research known as the out-of-sample extension for transductive learning methods (e.g., Delalleau et al., 2005), transductive learning is able to deal with infinitely many test data, just like semi-supervised learning. The original test data now serve as unlabeled data, while the out-of-sample extension serves as the model though it is not built by the learning method;

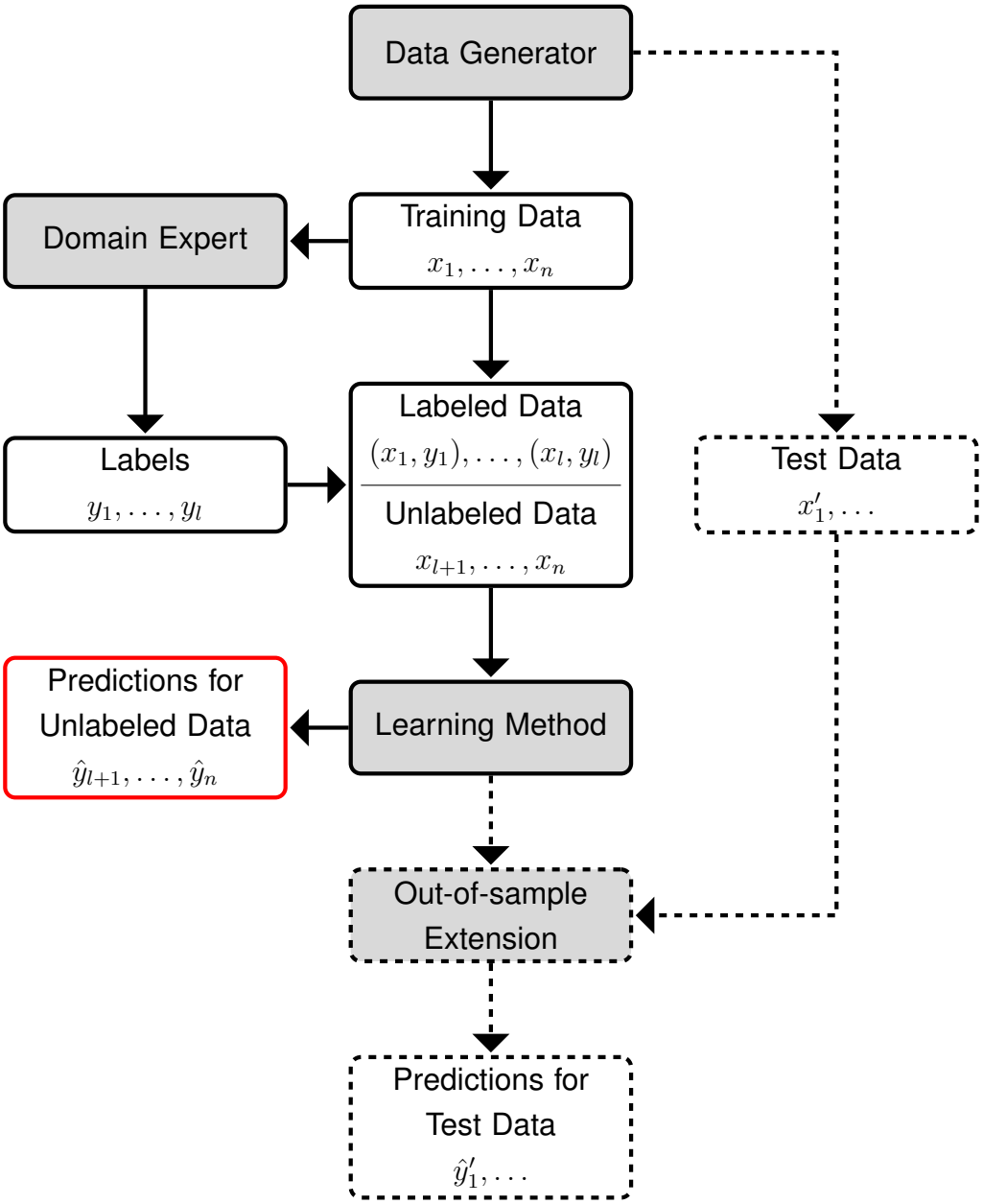


Figure 1.4: Data flow of transductive learning. The white boxes mean the data and labels, the gray boxes mean the subjects, and the box with a red border indicates the primary goal. The dashed boxes and lines compose the data flow for the secondary goal.

- As a result, a transductive learning method equipped with an out-of-sample extension need not be retrained when new test data arrive.

In recent years, transductive learning papers and semi-supervised learning papers are significantly overlapped.

The data flow of transductive learning is shown in Figure 1.4. Compared with the data flow of semi-supervised learning in Figure 1.3, we can see that the out-of-sample extension replaces the model. Now, making predictions for unlabeled data is the primary goal, and the predictions directly follow the learning method on the chain without passing the out-of-sample extension. On the other hand, the secondary goal is making predictions for test data, and its data flow is indicated by the dashed boxes and lines, since not every transductive learning method has an out-of-sample extension and can deal with test data.

1.3.5 Weak-supervised Learning

Weak-supervised learning infers a function $y = f(u, v)$, where the output y is a *weak label*, based on *completely labeled training data pairs*. A weak label only takes two values $+1$ and -1 and indicates the similarity and dissimilarity of the input data u and v, such that

- If the output $y = +1$, then the input data u and v are similar;
- If the output $y = -1$, then the input data u and v are dissimilar.

The physical meaning of weak labels implies that the function f should be symmetric with respect to its variables, i.e., $f(u, v) = f(v, u)$ for any u and v.

The data flow of weak-supervised learning is shown in Figure 1.5. Compared with the data flow of supervised learning in Figure 1.1, we can see that

- All the data points are replaced with the corresponding data pairs;
- All the class labels are replaced with the corresponding weak labels.

For real-world applications, the domain expert labels the similarity and dissimilarity of given data pairs, and it is easier for the domain expert than to label the classes of all data points. However, for laboratorial study, we still employ those benchmark data sets for classification to evaluate the learning method. All weak

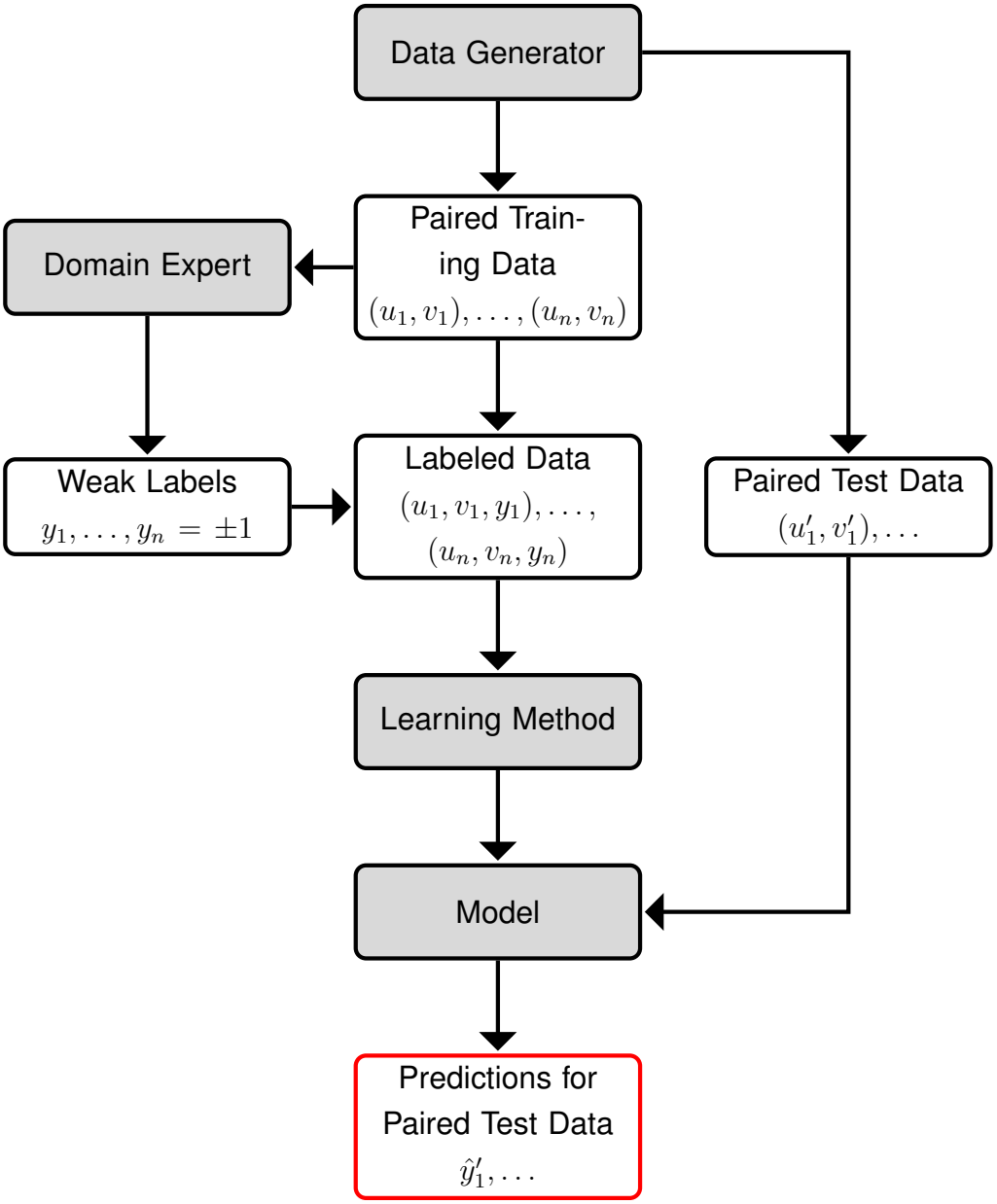


Figure 1.5: Data flow of weak-supervised learning. The white boxes mean the data and labels, the gray boxes mean the subjects, and the box with a red border indicates the primary goal. A weak label $y_i = +1$ indicates u_i and v_i are similar, and $y_i = -1$ indicates u_i and v_i are dissimilar.

labels are constructed from class labels by testing whether the class labels of u and v equal or not. Given the class-posterior probability $p(y' \mid x)$ where y' is a class label and x can be u or v, the conditional probability of a similarity weak label is then

$$p(y = +1 \mid u, v) = \sum_{y' \in \mathcal{Y}} p(y' \mid u) p(y' \mid v),$$

and the conditional probability of a dissimilarity weak label is

$$p(y = -1 \mid u, v) = 1 - p(y = +1 \mid u, v).$$

When constructing weak labels from class labels, it is clear that learning with weak labels is a member of learning with imperfect supervision:

- $O(n)$ data points with class labels can lead to $O(n^2)$ data pairs with weak labels;
- For $O(n)$ data points, $O(n)$ weal labels cannot recover the original $O(n)$ class labels, unless the set of class labels $\mathcal{Y}$ only contains two elements.

1.3.6 Active Learning

Active learning refers to the problem of inferring a function $y = f(x)$ when the learning method *actively join the data generating or labeling*. There are two approaches to active learning, *sequential active learning* and *batch active learning* with completely different stories (cf. Sugiyama, 2006):

- In sequential active learning, some training data are collected and labeled, the model is learned, then more training data are collected and labeled and the model is learned again. This iteration of more-data and better-model is repeated until the learning method decides to stop;
- In batch active learning, all training data are collected in the beginning.

The discussion below applies only to the sequential approach to active learning, since the batch approach is not learning with imperfect supervision.

The data flow of active learning is shown in Figure 1.6. We can see that there is a feedback from the model to the data generator or learning method. The training

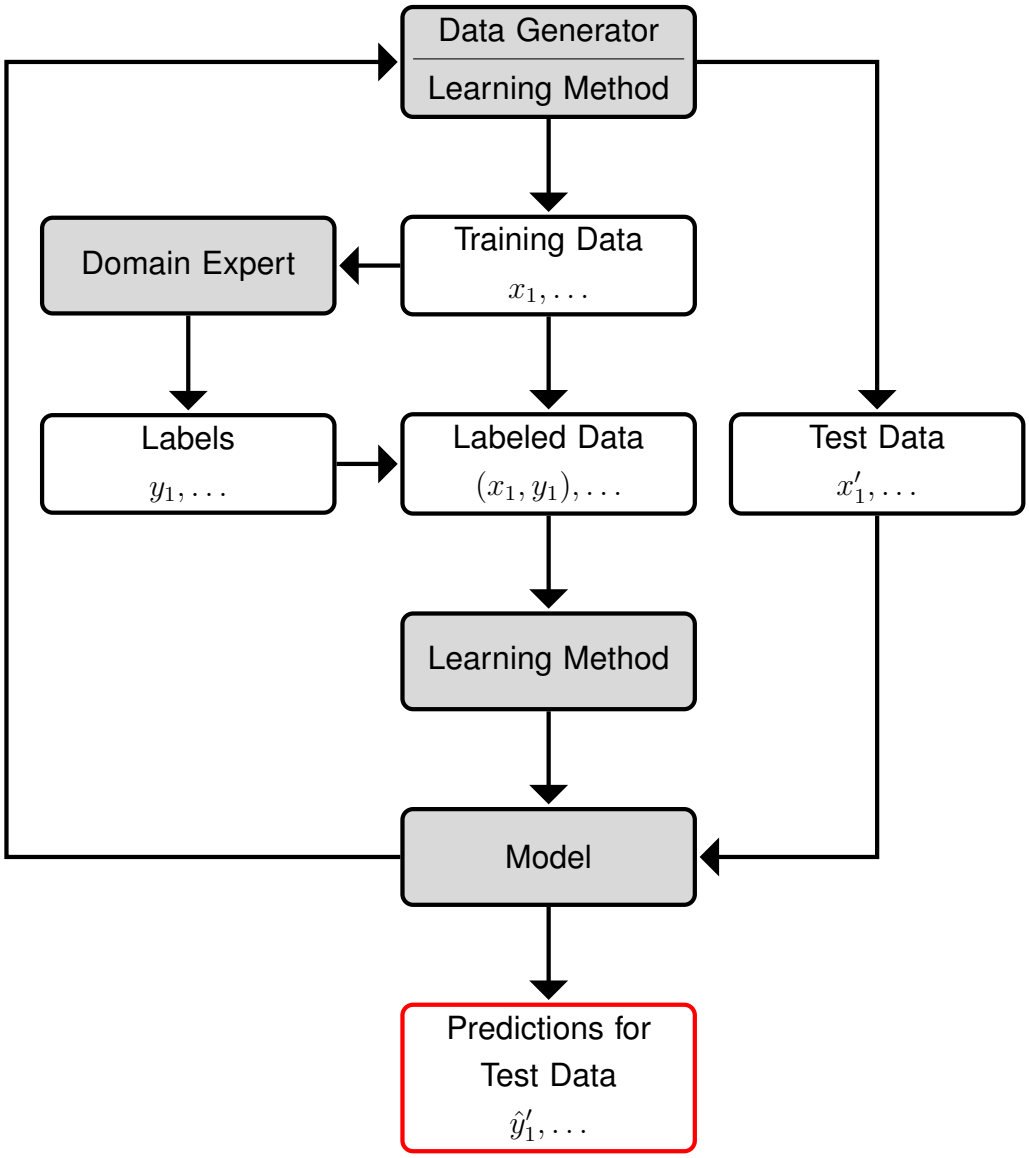


Figure 1.6: Data flow of active learning. The white boxes mean the data and labels, the gray boxes mean the subjects, and the box with a red border indicates the primary goal.

data flow is not linear as supervised learning or semi-supervised learning, but contains many circles, and each circle improves the model a bit towards the optimal model. Basically, active learning methods can either generate data by themselves or access a huge amount of data generated by the data generator and stored in the data pool:

- If a single data point is collected in each iteration of more-data and better-model, then the learning method may access a huge amount of data in the data pool;
- If a few data points are collected in each iteration of more-data and better-model, then the learning method may generate data or control the distribution according to which the data generator works.

In both cases, the training data labeled by the domain expert should have a density different from the underlying marginal density $p(x)$, and most often but not necessarily data that are more difficult to be classified deserve higher probability to be labeled.

The combination of active learning and semi-supervised learning is popular. That is, when building the model, the learning method makes use of the data in the data pool that have not been labeled by the domain expert. The difference of active and passive semi-supervised learning is the learning method or the domain expert who decides the data that are important and should be labeled. Letting the learning method decide may be better, since the domain expert usually gives the supervision independent of what the learning method needs to build a model.

1.3.7 Reinforcement Learning

Reinforcement learning refers to the problem of inferring a function $a = f(s)$ or equivalently $p(a \mid s)$ based on *returns of trajectories* where a is the action and s is the state. The problem setting of reinforcement learning is very different from other problem settings introduced before. We just illustrate it can also be viewed as learning with imperfect supervision. Please refer to Sutton and Barto (1998) for details of reinforcement learning.

The data flow of reinforcement learning is shown in Figure 1.7. We can see that the names in the boxes are different from the ones we have seen:

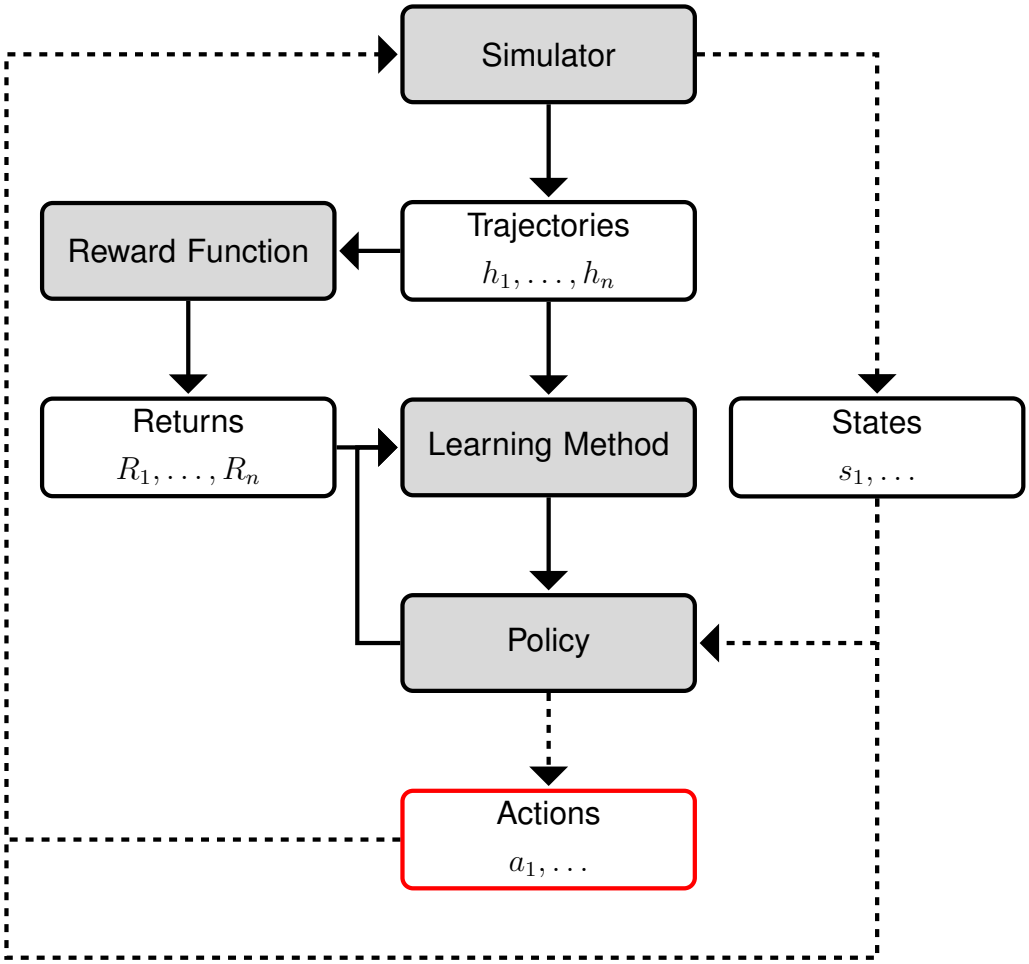


Figure 1.7: Data flow of reinforcement learning. The white boxes mean the data and labels, the gray boxes mean the subjects, and the box with a red border indicates the primary goal. The dashed boxes and lines compose the data flow for trajectory generating, while the solid ones compose the data flow for policy learning.

- Trajectories are analogous with training data. A trajectory h is a sequence of states and actions, and it can be written as $(s_1, a_1, \ldots, s_T, a_T)$ if it is of length T;
- Returns, which are discounted sums of future rewards, are analogous with labels. The higher quality a trajectory h has, the higher the corresponding return R is;
- Simulator serves as a data generator. Given the current state s_t and action a_t, it produces the next state s_{t+1};
- Policy serves as a model. Given the current state s_t, it produces the current action a_t as $f(s_t)$ or according to $p(a_t \mid s_t)$;
- Reward Function serves as a domain expert. It assigns a return R to given trajectory h.

Note that the reward function may be designed by the domain expert. Unlike the domain expert who manually labels training data, the reward function automatically computes the returns of trajectories, and thus reinforcement learning can in principle be endless.

Furthermore, it is difficult to separate the training and test data flows in Figure 1.7. We use dashed boxes and lines to represent the data flow for trajectory generating, and solid ones to represent the data flow for policy learning. In order to generate a trajectory h of length T, we need to iterate the dashed circle T times. After we have n trajectories, the learning method updates the policy. Most often but not necessarily, the update of policy involves the trajectories, the returns, and the current policy.

It is easy to see that the returns of trajectories are imperfect supervision. If it was supervised learning, there should be T desired actions given by the domain expert for each trajectory of length T, while reinforcement learning only affords a single return for each trajectory no matter how large the value of T is.

1.4 Contribution of This Thesis

Our research contributes to three machine learning problems: Clustering, metric learning, and semi-supervised classification. In this section, we explain what our

contributions are and how we have achieved them briefly.

1.4.1 An Overview

In this thesis, we study discriminative methods with imperfect supervision in machine learning. Three machine learning models each with one or two algorithms are proposed, all of which are *discriminative* and work under *imperfect supervision* for certain machine learning problems:

Firstly, *maximum volume clustering* is a discriminative approach to clustering following the *large volume principle* (El-Yaniv et al., 2008). It includes, for example, a spectral clustering and two relaxed k-means clustering as special limit cases. This research involves two types of imperfect supervision. While the discriminative clustering model itself is unsupervised (Section 1.3.2), the large volume principle comes from transductive learning (Section 1.3.4). The learning problem here is hardest among three problems with least supervision.

Secondly, *semi-supervised metric learning paradigm with hyper-sparsity* is a discriminative approach to metric learning following both the supervised *maximum entropy principle* (Berger et al., 1996) and the unsupervised *minimum entropy principle* (Grandvalet and Bengio, 2005). It improves manifold regularization for metric learning by considering dissimilarity constraints over unlabeled data. This research involves the combination of weak-supervised learning (Section 1.3.5) and semi-supervised learning (Section 1.3.3). The learning problem here is in the middle of three problems with moderate supervision.

Thirdly, *squared-loss mutual information regularization* is a discriminative approach to semi-supervised classification following the *information maximization principle* (Sugiyama et al., 2011). It is convex under mild conditions, and thus improves the non-convexity of mutual information regularization. This research involves semi-supervised learning (Section 1.3.3). The learning problem here is easiest among three problems with most supervision.

We provide solid theoretical analyses for all of these methods. Experiments demonstrate that they compare favorably with the corresponding state-of-the-art methods.

1.4.2 Clustering

Clustering aims at grouping a set of objects in such a way that objects in the same cluster are more similar to each other than to those from other clusters.

Clustering has been an important topic in machine learning and data mining communities. In recent years, a large number of clustering methods have been developed to improve classical k-means clustering, e.g., kernel k-means clustering (Girolami, 2002), spectral clustering (Shi and Malik, 2000), and *maximum margin clustering* (MMC) (Xu et al., 2005). MMC, which maximizes the margin between two opposite clusters, is the first clustering approach that is directly connected to the statistical learning theory (Vapnik, 1998). For this reason, it has been extensively investigated. The theoretical foundation of MMC is the *large margin principle* (Vapnik, 1982).

Nevertheless, in statistical learning theory, the large margin principle is not the only way to go. A useful alternative to it is the *large volume principle* proposed by Vladimir Vapnik (Vapnik, 1982), which advocates that hypotheses lying in an equivalence class with a larger volume are more preferable.

Following the large volume principle, we introduce a novel discriminative clustering model called *maximum volume clustering* (MVC), and propose two approximation schemes to solve this model:

- A soft-label MVC method using sequential quadratic programming;
- A hard-label MVC method using semi-definite programming.

Subsequently, we show theoretically MVC includes the optimization problems of a spectral clustering (von Luxburg, 2007), two k-means clustering (Ding and He, 2004) and an information-maximization clustering (Sugiyama et al., 2011) as *special limit cases*. Hence, MVC might be regarded as a natural extension of many existing clustering methods. Moreover, we establish two theoretical results in order to analyze the soft-label MVC method:

- A theory called *finite sample stability*;
- A *data-dependent error bound.*

Experiments demonstrate that MVC often outperformed

- Kernel k-means clustering (Zha et al., 2002),
- Normalized spectral clustering (Ng et al., 2002),
- Maximum margin clustering (Xu et al., 2005),
- Generalized maximum margin clustering (Valizadegan and Jin, 2007),
- Label-generation maximum margin clustering (Li et al., 2009).

1.4.3 Metric Learning

Metric learning aims at finding a Mahalanobis distance, such that under this distance metric, objects that are labeled similar are close and objects that are labeled dissimilar are far apart (Xing et al., 2003). The similarity and dissimilarity constraints (as weak labels in weak-supervised learning, Section 1.3.5) are given by the domain expert.

Semi-supervised metric learning relaxes the requirement that an object must be involved in at least one similarity or dissimilarity constraint, or otherwise invisible to supervised metric learning. *Manifold regularization* is often used as the semi-supervised extension, which explores the hidden similarity constraints but ignores the hidden dissimilarity constraints over unlabeled data (Hoi et al., 2008; Baghshah and Shouraki, 2009; Zha et al., 2009; Liu et al., 2010).

To improve it, we propose a general information-theoretic approach SERAPH *(SEmi-supervised metRic leArning Paradigm with Hyper-sparsity)* that does not rely on the manifold assumption (Belkin et al., 2006). Given the probability parameterized by a Mahalanobis distance, we follow *entropy regularization* (Grandvalet and Bengio, 2005), that is,

- We maximize the entropy of that probability on labeled data;
- We minimize the entropy of that probability on unlabeled data.

Our approach allows the supervised and unsupervised parts to be integrated in a natural and meaningful way, since now the unsupervised part considers not only similarity constraints but also dissimilarity constraints, as the supervised part. The constrained optimization problem of SERAPH can be solved efficiently and stably by an EM-like scheme:

- The E-Step has an *analytical solution*;
- The M-Step is *convex* and *Lipschitz continuous*.

Experiments demonstrate that SERAPH often outperformed

- Global distance metric learning (Xing et al., 2003),
- Neighborhood component analysis (Goldberger et al., 2005),
- Large margin nearest neighbor classification (Weinberger et al., 2006),
- Information-theoretic metric learning (Davis et al., 2007),
- Local distance metric learning (Yang et al., 2006),
- Manifold Fisher discriminant analysis (Baghshah and Shouraki, 2009),

and the learned metric possesses high discriminability even under a noisy environment.

1.4.4 Semi-supervised Classification

Semi-supervised classification aims at training classifiers with both labeled and unlabeled data. It relaxes the requirement that all objects must be labeled by the domain expert. Instead, additional assumptions about the joint distribution of the data and class labels are made under semi-supervised settings to extract helpful information from unlabeled data. Among them, the *manifold assumption* (Belkin et al., 2006), which assumes that data lie on a manifold of much lower dimensionality than the input space, is of vital importance. Its origin is the *smoothness assumption* following the *low-density separation principle*.

However, this low-density separation principle is not the only way to go. The *information maximization principle* is a useful alternative. It prefers probabilistic classifiers maximizing a certain information measure (e.g., the mutual information) between input data and output labels (Agakov and Barber, 2007; Gomes et al., 2010; Sugiyama et al., 2011).

Following the information maximization principle, we propose a regularization technique called *squared-loss mutual information regularization* (SMIR) by

specifying the squared-loss mutual information (Suzuki et al., 2009) as the information measure to be maximized. A key advantage of SMIR over mutual information regularization is the *convexity* under mild conditions such that the *unique globally optimal solution* is accessible. Furthermore, SMIR offers the following four abilities to semi-supervised algorithms:

- Analytical solution;
- Out-of-sample classification;
- Multi-class classification;
- Probabilistic output.

SMIR is the unique framework up to now which incarnates semi-supervised algorithms with all four abilities mentioned above. Again, two novel *data-dependent generalization error bounds* are derived which even incorporate the information of unlabeled data. Experiments demonstrate that SMIR often outperformed

- Plain and kernel entropy regularization (Grandvalet and Bengio, 2005),
- Plain and kernel expectation regularization (Mann and McCallum, 2007),
- Laplacian regularized least squares (Belkin et al., 2006) with a multi-class extension,
- Learning with local and global consistency (Zhou et al., 2004) with an out-of-sample extension.

1.5 Organization of This Thesis

This thesis consists of five chapters (see Figure 1.8). In this section, we explain the organization of this thesis.

Chapter 1 covers the most important concepts, including machine learning in Section 1.1, discriminative methods in Section 1.2, and the imperfect supervision in Section 1.3. Particularly, we give the basic problem settings of discriminative unsupervised, semi-supervised and weak-supervised learning in Sections 1.3.2, 1.3.3 and 1.3.5. The knowledge in these sections form the basis of this thesis, so they should be read before the main chapters.

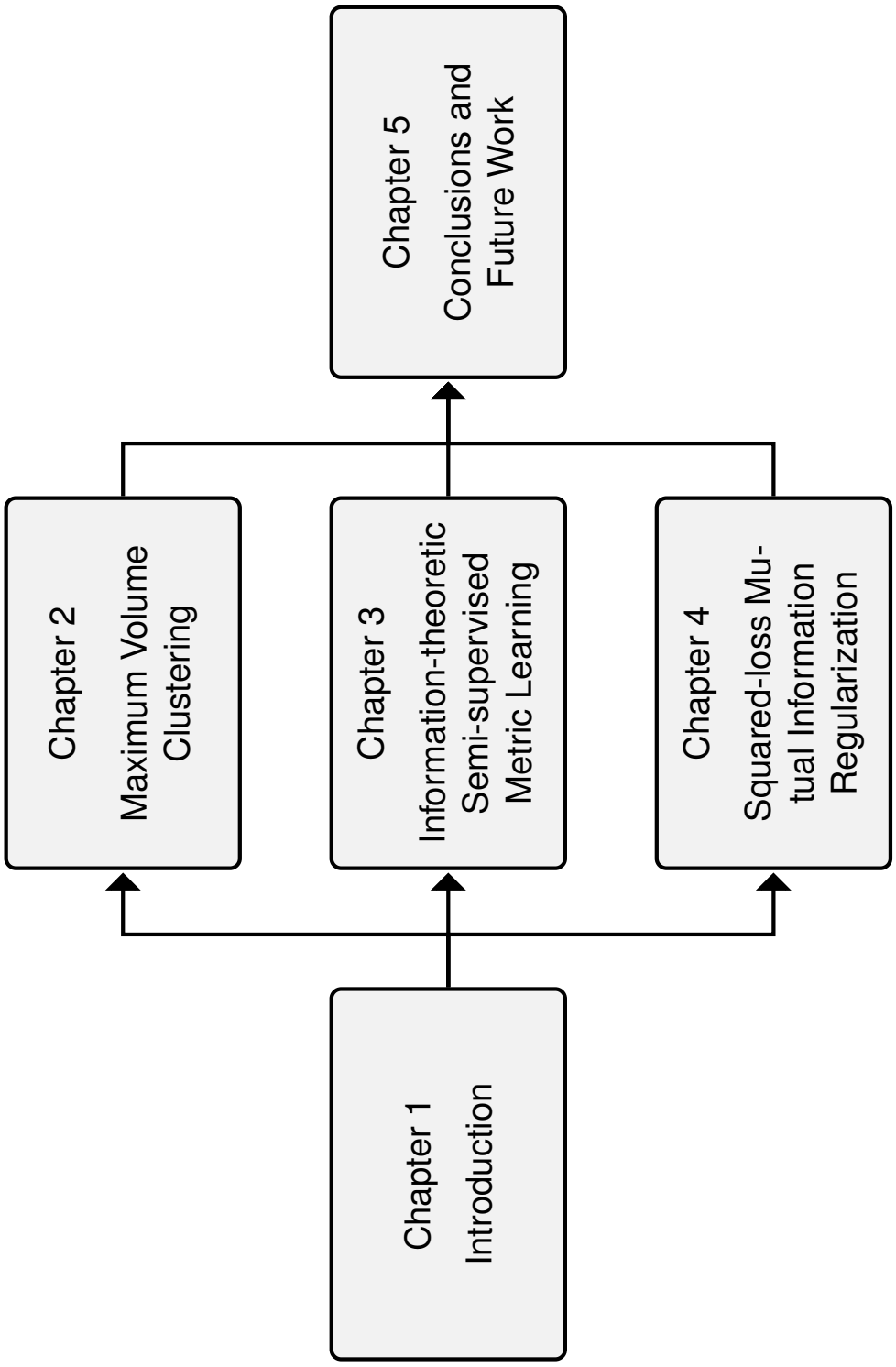


Figure 1.8: Organization of this thesis

Subsequently, Chapter 2, Chapter 3, and Chapter 4 are devoted to three discriminative methods with imperfect supervision. These chapters are independent and can be read separately. Note that among three chapters, the learning problem in Chapter 2 is hardest with least supervision, the learning problem in Chapter 4 is easiest with most supervision, and the learning problem in Chapter 3 is in the middle of them with moderate supervision.

In Chapter 2, we present maximum volume clustering (MVC). Section 2.1 describes the motivation and the background knowledge. In Section 2.2, we briefly review the large volume approximation (El-Yaniv et al., 2008). Then in Section 2.3, we propose MVC. More specifically, the basic model of MVC is in Section 2.3.1, the soft-label MVC method is in Section 2.3.2, and the hard-label MVC method is in Section 2.3.3. In Section 2.4, we show that MVC includes the optimizations of a spectral clustering (von Luxburg, 2007), two relaxed k-means clustering (Ding and He, 2004) and an information-theoretic clustering (Sugiyama et al., 2011) as special limit cases. The theoretical results, the finite sample stability theory and the data-dependent error bound, are derived in Sections 2.5 and 2.6 respectively. Related works are compared in Section 2.7. Experimental results are reported in Section 2.8. Finally, proofs of all the theoretical results given in this chapter are provided in Section 2.9.

In Chapter 3, we present information-theoretic semi-supervised metric learning. Section 3.1 describes the motivation and the background knowledge. In Section 3.2, the model of SERAPH is proposed including the basic model in Section 3.2.2 and the regularization in Section 3.2.3. Then in Section 3.3, a practical EM-like algorithm is developed to solve the optimization problem resulted from the proposed model, where the implementation details are also included in Section 3.3.4 since the optimization is non-convex. In Section 3.4, we discuss the posterior sparsity, the projection sparsity, and the hyper-sparsity in the sense of metric learning, and present two additional justifications of the proposed model. Related works are compared in Section 3.5. Experimental results are reported in Section 3.6. Finally, proofs of all the theoretical results given in this chapter are provided in Section 3.7.

In Chapter 4, we present a regularization technique for semi-supervised classification called squared-loss mutual information regularization (SMIR). Section

4.1 describes the motivation and the background knowledge. In Section 4.2, the unsupervised SMI approximator (Sugiyama et al., 2011) is reviewed. In Section 4.3, we propose SMIR. More specifically, an alternative kernel model and an alternative SMI approximator are in Section 4.3.1, the basic model is in Section 4.3.2, the algorithm using the squared difference of two probabilities as the loss function is in Section 4.3.3, and the post-processing is in Section 4.3.4. Then in Section 4.4, we derive and discuss the data-dependent generalization error bounds for SMIR. Related works are compared in Section 4.5. Experimental results are reported in Section 4.6. Finally, the proof of the generalization error bounds is provided in Section 4.7.

In the end, concluding remarks and future prospects are delivered in Chapter 5.

Chapter 2

Maximum Volume Clustering

In this chapter, we present maximum volume clustering (MVC) which is a novel discriminative clustering approach. Our contributions can be summarized as four folds.

- We apply the large volume principle for transduction to clustering;
- We demonstrate that MVC includes well-known clustering methods as special limit cases;
- We establish a theory called finite sample stability;
- Novel data-dependent error bound is derived.

This chapter is organized as follows. Sections 2.1 and 2.2 include the background and preliminaries. Then we propose the model and algorithms of MVC in Section 2.3, and show the generality in Section 2.4. In Sections 2.5 and 2.6, we present our theoretical results. Related works are compared in Section 2.7. Experimental results are reported in Section 2.8.

2.1 Introduction

Clustering has been an important topic in machine learning and data mining communities. Over the past decades, a large number of clustering algorithms have been developed. For instance, *k-means clustering* (MacQueen, 1967; Hartigan and Wong, 1979; Girolami, 2002), *spectral clustering* (Shi and Malik, 2000; Meila

and Shi, 2001; Ng et al., 2002), *maximum margin clustering* (MMC) (Xu et al., 2005; Xu and Schuurmans, 2005), *dependence-maximization clustering* (Song et al., 2007; Faivishevsky and Goldberger, 2010) and *information-maximization clustering* (Agakov and Barber, 2006; Gomes et al., 2010; Sugiyama et al., 2011). These algorithms have been successfully applied to diverse real-world data sets for exploratory data analysis.

To the best of our knowledge, MMC, which maximizes the margin between two opposite clusters, is the first clustering approach that is directly connected to the *statistical learning theory* (Vapnik, 1998). For this reason, it has been extensively investigated recently, for example, generalized MMC (Valizadegan and Jin, 2007) and lots of approximation algorithms for speedup (Zhang et al., 2007; Zhao et al., 2008b,a; Li et al., 2009; Wang et al., 2010).

However, the *large margin principle* (LMP) is not the only way to go. There is a *large volume principle* (LVP) which was introduced by Vapnik (1982) for *hyperplanes* and extended by El-Yaniv et al. (2008) for *soft response vectors*. Roughly speaking, learning algorithms based on LVP should prefer hypotheses in some large-volume equivalence classes. See Figure 2.1 as an illustrative comparison of two principles. Here, C_1, C_2 and C_3 represent three data clouds, and our goal is to choose a better hypothesis from two candidates h_1 and h_2. A hypothesis is a line, and an equivalence class is a set of lines which equivalently separate data samples. Therefore, we have two equivalence classes H_1 and H_2. Given an equivalence class H_1 (or H_2), its margin is measured by the distance between two red (or blue) lines, and its volume is measured by the area of the red (or blue) region.[1] Though LMP prefers h_1 due to the larger margin of H_1 than H_2, we should choose h_2 when considering LVP, since H_2 has a larger volume than H_1.

In this chapter, we introduce a novel discriminative clustering approach called *maximum volume clustering* (MVC), which serves as a prototype to partition the data samples into two clusters based on LVP. We motivate our MVC as follows. Given the samples X_n, we construct an X_n-dependent hypothesis space $\mathcal{H}(X_n)$. If $\mathcal{H}(X_n)$ has a measure on it, namely the *power*, then we can talk about the *likelihood* or *confidence* of each equivalence class (Vapnik, 1998). Similarly to

[1] In Figure 2.1, we integrate all unit line segments in an equivalence class and treat the resulting area as its volume.

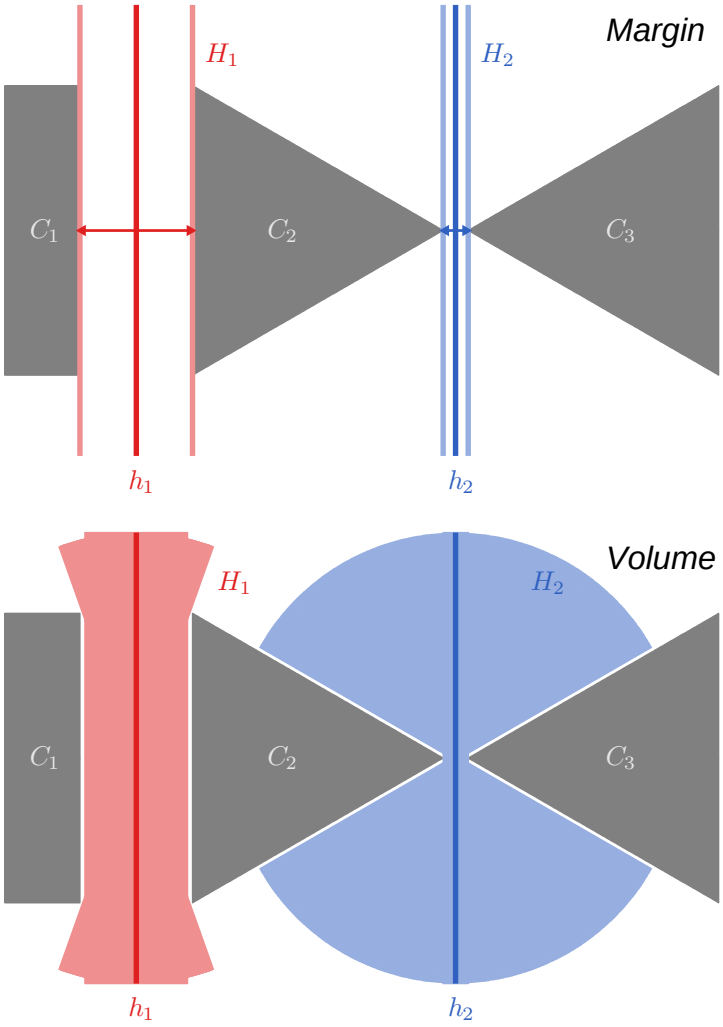


Figure 2.1: Large margin vs. large volume separation of three data clouds C_1, C_2 and C_3 into two clusters. In this figure, a hypothesis is a line (e.g., h_1), and an equivalence class is a set of lines which equivalently separate data samples (e.g., H_1). Given H_1 (or H_2), its margin is measured by the distance between two red (or blue) lines, and its volume is measured by the area of the red (or blue) region. The large margin principle prefers h_1 and the large volume principle prefers h_2, since they consider different complexity measures.

the *margin* used in MMC, the notion of *volume* (El-Yaniv et al., 2008) can also be regarded as an estimation of the power. Therefore, the larger the volume is, the more confident we are of the data partition, and we consider the partition lying in the equivalence class with the maximum volume as the best partition.

Similarly to the majority of clustering algorithms, the optimization problem involved in MVC is combinatorial and thus NP-hard, so we propose two approximation schemes:

- A soft-label MVC method that can be solved by *sequential quadratic programming* (Boggs and Tolle, 1995) in $O(n^3)$ time;
- A hard-label MVC method as a *semi-definite programming* problem (De Bie and Cristianini, 2004; Lanckriet et al., 2004) that can be solved in $O(n^{6.5})$ time.

Subsequently, we show that the primal problem of soft-label MVC can be reduced to the optimization problems of *unnormalized spectral clustering* (von Luxburg, 2007), plain and kernel k*-means clustering* after relaxations (Ding and He, 2004), and *squared-loss mutual information based clustering* (Sugiyama et al., 2011), as the regularization parameter of MVC approaches infinity. Hence, MVC might be regarded as a natural extension of many existing clustering methods. Moreover, we establish two theoretical results:

- A theory called *finite sample stability* for analyzing the soft-label MVC method. It suggests that under mild conditions, different locally optimal solutions to soft-label MVC would induce the same data partition, and thus the non-convex optimization of soft-label MVC seems like a convex one;
- A *data-dependent error bound* for the soft-label MVC method. It upper bounds the distance between the partition returned by soft-label MVC and any partially observed partition based on *transductive Rademacher complexity* (El-Yaniv and Pechyony, 2009).

Experiments on three artificial and fourteen benchmark data sets (i.e., ten IDA benchmarks, USPS, MNIST, 20Newsgroups and Isolet) demonstrate that the proposed MVC approach is promising.

2.2 Large Volume Approximation

Suppose that we are given a set of objects $X_n = \{x_1, \ldots, x_n\}$, where $x_i \in \mathcal{X}$ for $i = 1, \ldots, n$, and most often but not necessarily, $\mathcal{X} \subset \mathbb{R}^d$ for some natural number d. We will construct a *hypothesis space* $\mathcal{H}(X_n)$ that depends on X_n, such that for any hypothesis $\boldsymbol{h} \in \mathcal{H}(X_n) \subset \mathbb{R}^n$, $[\boldsymbol{h}]_i$ stands for a *soft response* or *confidence-rated label* of x_i, where $[\cdot]_i$ means the i-th component of a vector. We will then pick a *soft response vector* $\boldsymbol{h}^*$ following the large volume principle and partition X_n into two clusters $\{x_i \mid [\boldsymbol{h}^*]_i > 0\}$ and $\{x_i \mid [\boldsymbol{h}^*]_i < 0\}$.[2]

As El-Yaniv et al. (2008), assume we have a symmetric positive-definite matrix $Q \in \mathbb{R}^{n \times n}$ that contains the pairwise information about X_n. Consider the hypothesis space

$$\mathcal{H}_Q := \{\boldsymbol{h} \mid \boldsymbol{h}^\top Q \boldsymbol{h} \leq 1\},$$

which is geometrically an origin-centered ellipsoid $\mathcal{E}(\mathcal{H}_Q)$ in $\mathbb{R}^n$. The set of sign vectors

$$\{\mathrm{sign}(\boldsymbol{h}) \mid \boldsymbol{h} \in \mathcal{H}_Q\}$$

contains all 2^n possible dichotomies of X_n. In other words, $\mathcal{H}_Q$ is now partitioned into a finite number of *equivalence classes* $H_1, \ldots, H_{2^n}$, such that for fixed $k \in \{1, 2, 3, \ldots, 2^n\}$, all hypotheses in H_k will generate the same dichotomy of X_n. The *power* of an equivalence class H_k is defined as a probability mass

$$\mathcal{P}(H_k) := \int_{H_k} p(\boldsymbol{h}) \mathrm{d}\boldsymbol{h}, \quad k = 1, \ldots, 2^n,$$

where $p(\boldsymbol{h})$ is the underlying probability density of $\boldsymbol{h}$ over $\mathcal{H}_Q$. The hypotheses in H_k with a large power $\mathcal{P}(H_k)$ are preferred according to the statistical learning theory (Vapnik, 1998).

When no specific domain knowledge is available (i.e., $p(\boldsymbol{h})$ is unknown), it would be natural to assume the continuous uniform distribution

$$p(\boldsymbol{h}) = \frac{1}{\sum_{k=1}^{2^n} \mathcal{V}(H_k)},$$

where

$$\mathcal{V}(H_k) := \int_{H_k} \mathrm{d}\boldsymbol{h}, \quad k = 1, \ldots, 2^n,$$

[2] Due to our clustering model that will be defined as optimization (2.2) in page 37, $[\boldsymbol{h}^*]_i = 0$ hardly happens in practice, and we simply assume $[\boldsymbol{h}^*]_i \neq 0$ in our problem setting.

is the *volume* of H_k as well as the geometric volume of the k-th quadrant of $\mathcal{E}(\mathcal{H}_Q)$. Consequently, $\mathcal{P}(H_k)$ is proportional to $\mathcal{V}(H_k)$, and the larger the value of $\mathcal{V}(H_k)$ is, the more confident we are of the data partition $\mathrm{sign}(\boldsymbol{h}^*)$ where $\boldsymbol{h}^*$ is chosen from H_k.

However, it is very hard to accurately compute the geometric volume of a single n-dimensional convex body let alone for all 2^n convex bodies, so we employ an efficient approximation introduced by El-Yaniv et al. (2008) as follows. Let $\lambda_1 \leq \cdots \leq \lambda_n$ be the eigenvalues of Q, and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be the associated normalized eigenvectors. Then, $\boldsymbol{v}_i$ and $1/\sqrt{\lambda_i}$ are the direction and length of the i-th principal axis of $\mathcal{E}(\mathcal{H}_Q)$. Note that a small angle from $\boldsymbol{h} \in H_k$ to $\boldsymbol{v}_i$ with a small/large index i (i.e., a long/short principal axis) implies that $\mathcal{V}(H_k)$ is large/small. Based on this key observation, we define

$$V(\boldsymbol{h}) := \sum_{i=1}^{n} \lambda_i \left(\frac{\boldsymbol{h}^\top \boldsymbol{v}_i}{\|\boldsymbol{h}\|_2} \right)^2 = \frac{\boldsymbol{h}^\top Q \boldsymbol{h}}{\|\boldsymbol{h}\|_2^2}, \tag{2.1}$$

where $\boldsymbol{h}^\top \boldsymbol{v}_i / \|\boldsymbol{h}\|_2$ means the cosine of the angle between $\boldsymbol{h}$ and $\boldsymbol{v}_i$. We subsequently expect $V(\boldsymbol{h})$ to be small when $\boldsymbol{h}$ lies in a large-volume equivalence class, and conversely to be large in a small-volume equivalence class.

2.3 Maximum Volume Clustering

In this section, we define our clustering model and propose two approximation algorithms.

2.3.1 Basic Formulation

Motivated by Xu et al. (2005), we formulate the binary clustering from a regularization viewpoint. If we have labels $Y_n = \{y_1, \ldots, y_n\}$ at hand where $y_i \in \{-1, +1\}$, we can find a base algorithm to compute

$$\vartheta(X_n, Y_n) := \min_{\boldsymbol{h} \in \mathcal{H}(X_n, Y_n)} \Delta(Y_n, \boldsymbol{h}) + \gamma W(X_n, \boldsymbol{h}),$$

where $\mathcal{H}(X_n, Y_n)$ is a hypothesis space that depends upon X_n and Y_n, $\Delta(Y_n, \boldsymbol{h})$ is the overall loss function, $W(X_n, \boldsymbol{h})$ is a regularization function and $\gamma > 0$ is

a regularization parameter. The value of $\vartheta(X_n, Y_n)$ is a measure of *classification quality*.

When the labels are absent, a clustering algorithm tries to minimize $\vartheta(X_n, \boldsymbol{y})$ over all possible assignments $\boldsymbol{y} \in \{-1, +1\}^n$ for given X_n, that is, to solve the problem

$$\boldsymbol{y}^* = \operatorname*{arg\,min}_{\boldsymbol{y}\in\{-1,+1\}^n} \vartheta(X_n, \boldsymbol{y}).$$

Generally speaking, $\vartheta(X_n, \boldsymbol{y}^*)$ can be regarded as a measure of *clustering quality*. The smaller the value of $\vartheta(X_n, \boldsymbol{y}^*)$ is, the more satisfied we are with the resulting data partition $\boldsymbol{y}^*$.

In our discriminative clustering model, we hope to utilize $V(\boldsymbol{h})$ in Eq. (2.1) as our regularization function. Formally speaking, given the matrix Q, by instantiating $\Delta(\boldsymbol{y}, \boldsymbol{h}) = -2\boldsymbol{h}^\top \boldsymbol{y}$, we define the basic model of *maximum volume clustering* (MVC) as

$$\min_{\boldsymbol{y}\in\{-1,+1\}^n} \min_{\boldsymbol{h}\in\mathcal{H}_Q} -2\boldsymbol{h}^\top \boldsymbol{y} + \gamma \cdot \frac{\boldsymbol{h}^\top Q \boldsymbol{h}}{\|\boldsymbol{h}\|_2^2}, \tag{2.2}$$

where $\mathcal{H}_Q = \{\boldsymbol{h} \mid \boldsymbol{h}^\top Q \boldsymbol{h} \leq 1\}$ is the hypothesis space mentioned in Section 2.2, and $\gamma > 0$ is the regularization parameter. Optimization problem (2.2) is computationally intractable, due to not only the non-convexity of $V(\boldsymbol{h})$, but also the integer feasible region of $\boldsymbol{y}$ that makes (2.2) combinatorial. In the next two subsections, we will discuss two approximation schemes of (2.2) in detail.

2.3.2 Soft-label Approximation

We now try to optimize $\boldsymbol{h}$ alone by removing $\boldsymbol{y}$. After exchanging the order of the minimizations of $\boldsymbol{y}$ and $\boldsymbol{h}$ in optimization (2.2), it is easy to see that the optimal $\boldsymbol{y}$ should be $\operatorname{sign}(\boldsymbol{h})$, since the second term is independent of $\boldsymbol{y}$ and the first term is minimized when $\boldsymbol{y} = \operatorname{sign}(\boldsymbol{h})$ for fixed $\boldsymbol{h}$. Therefore, (2.2) becomes

$$\min_{\boldsymbol{h}\in\mathcal{H}_Q} -2\|\boldsymbol{h}\|_1 + \gamma \cdot \frac{\boldsymbol{h}^\top Q \boldsymbol{h}}{\|\boldsymbol{h}\|_2^2}. \tag{2.3}$$

Similarly to El-Yaniv et al. (2008), we replace the feasible region $\mathcal{H}_Q$ with $\mathbb{R}^n$, and relax (2.3) into

$$\min_{\boldsymbol{h}\in\mathbb{R}^n} -2\|\boldsymbol{h}\|_1 + \gamma \boldsymbol{h}^\top Q \boldsymbol{h} \qquad \text{s.t. } \|\boldsymbol{h}\|_2 = 1. \tag{2.4}$$

Although the optimization is done in $\mathbb{R}^n$ now, the regularization is done relative to $\mathcal{H}_Q$. Optimization (2.4) is the primal problem of *soft-label MVC* (MVC-SL).

Optimization (2.4) is non-convex mainly attributed to the minimization of negative ℓ_1-norm rather than the equality constraint of ℓ_2-norm. In order to solve this optimization, we resort to *sequential quadratic programming* (SQP) (Boggs and Tolle, 1995). The basic idea of SQP is modeling a non-convex problem by a sequence of convex subproblems: At each step, it uses a quadratic model for the objective function and linear models for the constraints. A nonlinear optimization problem with a quadratic objective function and linear constraints is known as *quadratic programming* (QP). An SQP constructs and solves a local QP at each iteration, yielding a step toward the optimum.

More specifically, let us include a class balance constraint $-b \leq \boldsymbol{h}^\top \mathbf{1}_n \leq b$ with a user-specified class balance parameter $b > 0$ to prevent skewed clustering sizes. Denote the objective function of optimization (2.4) by

$$f(\boldsymbol{h}) := -2\boldsymbol{h}^\top \operatorname{sign}(\boldsymbol{h}) + \gamma \boldsymbol{h}^\top Q \boldsymbol{h},$$

and the auxiliary functions by

$$\begin{aligned} f_1(\boldsymbol{h}) &:= \boldsymbol{h}^\top \boldsymbol{h} - 1, \\ f_2(\boldsymbol{h}) &:= \boldsymbol{h}^\top \mathbf{1}_n, \end{aligned}$$

where $\mathbf{1}_n$ means the all-one vector in $\mathbb{R}^n$. Subsequently, let λ_1 be the smallest eigenvalue of Q, the corresponding Lagrange function should be[3]

$$L(\boldsymbol{h}, \eta, \mu, \nu) = f(\boldsymbol{h}) - \eta f_1(\boldsymbol{h}) - \mu(f_2(\boldsymbol{h}) - b) + \nu(f_2(\boldsymbol{h}) + b),$$

where $\eta < \gamma\lambda_1$ is the Lagrangian multiplier for the constraint $f_1(\boldsymbol{h}) = 0$, and $\mu \geq 0, \nu \geq 0$ are the Lagrangian multipliers for the constraint $-b \leq f_2(\boldsymbol{h}) \leq b$. Then, given constant $\boldsymbol{h}$ and variable $\boldsymbol{p}$ with a tiny norm, the auxiliary functions can be approximated by

$$\begin{aligned} f_1(\boldsymbol{h} + \boldsymbol{p}) &\approx \boldsymbol{p}^\top \nabla f_1(\boldsymbol{h}) + f_1(\boldsymbol{h}), \\ f_2(\boldsymbol{h} + \boldsymbol{p}) &= \boldsymbol{p}^\top \nabla f_2(\boldsymbol{h}) + f_2(\boldsymbol{h}), \end{aligned}$$

[3]We will ignore variables μ and ν later, since first-order terms of $L(\boldsymbol{h}, \eta, \mu, \nu)$ would disappear in the second-order derivative $\nabla^2 L(\boldsymbol{h}, \eta, \mu, \nu)$. The Lagrange function $L(\boldsymbol{h}, \eta)$ itself has no constraint on $\boldsymbol{h}$, so we impose $\eta < \gamma\lambda_1$ to make sure that $L(\boldsymbol{h}, \eta)$ is bounded from below. Otherwise, the subproblem may be ill-defined.

so the constraints are replaced with

$$\begin{aligned} \boldsymbol{p}^\top \nabla f_1(\boldsymbol{h}) + f_1(\boldsymbol{h}) &= 0, \\ -b \leq \boldsymbol{p}^\top \nabla f_2(\boldsymbol{h}) + f_2(\boldsymbol{h}) &\leq b. \end{aligned}$$

Nevertheless, it would be incorrect to adopt the second-order Taylor expansion of $f(\boldsymbol{h}+\boldsymbol{p})$ as our new objective function, since we need to capture the curvature of $f_1(\boldsymbol{h}+\boldsymbol{p})$. The correct way is to use the quadratic model[4]

$$L(\boldsymbol{h}+\boldsymbol{p},\eta) \approx \frac{1}{2}\boldsymbol{p}^\top \nabla^2 L(\boldsymbol{h},\eta)\boldsymbol{p} + \boldsymbol{p}^\top \nabla L(\boldsymbol{h},\eta) + L(\boldsymbol{h},\eta)$$

and form our objective at any fixed $(\boldsymbol{h},\eta)$ into

$$\min_{\boldsymbol{p}\in\mathbb{R}^n} \ \frac{1}{2}\boldsymbol{p}^\top \nabla^2 L(\boldsymbol{h},\eta)\boldsymbol{p} + \boldsymbol{p}^\top \nabla f(\boldsymbol{h}),$$

according to Boggs and Tolle (1995, p. 9). As a consequence, the subproblem of the t-th iteration is a simple QP at the current estimate $(\boldsymbol{h}_t,\eta_t)$:

$$\begin{aligned} \min_{\boldsymbol{p}_t\in\mathbb{R}^n} \ & \boldsymbol{p}_t^\top(\gamma Q - \eta_t I_n)\boldsymbol{p}_t + 2\boldsymbol{p}_t^\top\big(\gamma Q\boldsymbol{h}_t - \operatorname{sign}(\boldsymbol{h}_t)\big) \\ \text{s.t.} \ & 2\boldsymbol{p}_t^\top\boldsymbol{h}_t + \boldsymbol{h}_t^\top\boldsymbol{h}_t = 1 \\ & -b \leq \boldsymbol{p}_t^\top\mathbf{1}_n + \boldsymbol{h}_t^\top\mathbf{1}_n \leq b, \end{aligned} \tag{2.5}$$

where I_n is the identity matrix of size n. The new estimate $(\boldsymbol{h}_{t+1},\eta_{t+1})$ is given by

$$\boldsymbol{h}_{t+1} = \boldsymbol{h}_t + \boldsymbol{p}_t^*, \tag{2.6}$$

$$\eta_{t+1} = \frac{\boldsymbol{h}_t^\top\big(\gamma Q\boldsymbol{h}_{t+1} - \eta_t\boldsymbol{p}_t^* - \operatorname{sign}(\boldsymbol{h}_t)\big)}{\boldsymbol{h}_t^\top\boldsymbol{h}_t}, \tag{2.7}$$

where $\boldsymbol{p}_t^*$ is the optimal solution to (2.5). Notice that we cannot obtain η_{t+1} directly from (2.5) and in fact Eq. (2.7) comes from the best fit in the least-square sense of the following equation

$$\nabla^2 L(\boldsymbol{h}_t,\eta_t)\boldsymbol{p}_t^* + \nabla f(\boldsymbol{h}_t) - \eta_{t+1}\nabla f_1(\boldsymbol{h}_t) = 0. \tag{2.8}$$

[4]Note that minimizing $-\boldsymbol{h}^\top\boldsymbol{y}$ in optimization (2.2) or $-\|\boldsymbol{h}\|_1$ in optimization (2.4) has an effect to push $\boldsymbol{h}$ away from the coordinate axes of $\mathbb{R}^n$. Thus, $[\boldsymbol{h}]_i = 0$ hardly happens in practice and we assume that $\|\boldsymbol{h}\|_1$ is always differentiable.

Algorithm 1 MVC-SL

Input: stopping criterion ϵ,
symmetric positive-definite matrix Q,
regularization parameter γ,
class balance parameter b

Output: soft response vector $\boldsymbol{h}^*$

Initialize $(\boldsymbol{h}_0, \eta_0)$, recommended but not required, from Eq. (2.9)
$t \leftarrow 0$
repeat
  Obtain $\boldsymbol{p}_t^*$ through optimization (2.5)
  Update $\boldsymbol{h}_{t+1}$ through Eq. (2.6)
  Update η_{t+1} through Eq. (2.7)
  if $\eta_{t+1} \geq \gamma\lambda_1$ **then break**
  $t \leftarrow t + 1$
until $\|\boldsymbol{h}_t - \boldsymbol{h}_{t-1}\|_2 + |\eta_t - \eta_{t-1}| \leq \epsilon$
return $\boldsymbol{h}^* = \boldsymbol{h}_t$

The MVC-SL algorithm based on SQP is summarized in Algorithm 1. In our experiments, we use an initial solution $(\boldsymbol{h}_0, \eta_0)$ defined as

$$\boldsymbol{h}_0 = \frac{1}{\sqrt{n}} \operatorname{sign}\left(\boldsymbol{v}_2 - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{v}_2\right), \quad \eta_0 = 0, \tag{2.9}$$

where $\boldsymbol{v}_2$ is the eigenvector associated with the second smallest eigenvalue of Q. The construction of $\boldsymbol{h}_0$ is explained as follows. The term $(\boldsymbol{v}_2 - \mathbf{1}_n\mathbf{1}_n^\top\boldsymbol{v}_2/n)$ equals $C_n\boldsymbol{v}_2$ where $C_n = I_n - \mathbf{1}_n\mathbf{1}_n^\top/n$ is the centering matrix, and it means that the center of $\boldsymbol{v}_2$ is subtracted from its components. Then, an initial data partition is extracted from the sign vector of $C_n\boldsymbol{v}_2$ and normalized into a unit vector as $\boldsymbol{h}_0$. The asymptotic time complexity of each subproblem is at most $O(n^3)$, and the convergence rate of SQP iterations is independent of n (Boggs and Tolle, 1995). Moreover, it takes $O(n^2)$ time to compute $\boldsymbol{h}_0$. Hence, the overall computational complexity of Algorithm 1 is no more than $O(n^3)$.

2.3.3 Hard-label Approximation

As opposed to the soft-label approximation, we can also optimize $\boldsymbol{y}$ alone. Let $\boldsymbol{h} = \boldsymbol{\alpha} \circ \boldsymbol{y}$, where $\boldsymbol{\alpha} = |\boldsymbol{h}|$ is a vector of element-wise absolute values, $\boldsymbol{y} = \operatorname{sign}(\boldsymbol{h})$ is a vector of the corresponding signs, and $\circ$ means the element-wise product. We would like to further introduce a hyperparameter C to bound each component of $\boldsymbol{\alpha}$, which might be helpful for dealing with outliers. Subsequently, the primal problem of *hard-label MVC* (MVC-HL) is written as

$$\begin{aligned} \min_{\boldsymbol{y}\in\{-1,+1\}^n} \min_{\boldsymbol{\alpha}\in\mathbb{R}^n} \; & -2\boldsymbol{\alpha}^\top \mathbf{1}_n + \gamma \boldsymbol{\alpha}^\top \left(Q \circ \boldsymbol{y}\boldsymbol{y}^\top\right) \boldsymbol{\alpha} \\ \text{s.t.} \; & \boldsymbol{\alpha}^\top \boldsymbol{\alpha} = 1 \\ & \mathbf{0}_n \leq \boldsymbol{\alpha} \leq C\mathbf{1}_n, \end{aligned} \tag{2.10}$$

where $\mathbf{0}_n$ means the all-zero vector in $\mathbb{R}^n$.

By employing the technique described in Lanckriet et al. (2004), let $M = \boldsymbol{y}\boldsymbol{y}^\top$ and then optimization (2.10) can be relaxed to

$$\begin{aligned} \min_{M\in\mathbb{R}^{n\times n}} \min_{\eta\in\mathbb{R}} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \; & 2\boldsymbol{\alpha}^\top \mathbf{1}_n - \gamma \boldsymbol{\alpha}^\top (Q \circ M)\boldsymbol{\alpha} + \eta(\boldsymbol{\alpha}^\top \boldsymbol{\alpha} - 1) \\ \text{s.t.} \; & M \succeq \mathbf{0} \\ & \operatorname{diag}(M) = \mathbf{1}_n \\ & \mathbf{0}_n \leq \boldsymbol{\alpha} \leq C\mathbf{1}_n, \end{aligned} \tag{2.11}$$

where the function $\operatorname{diag}(\cdot)$ forms the diagonal entries of a square matrix into a column vector, and $\succeq \mathbf{0}$ indicates the positive semi-definiteness of a symmetric matrix.[5] The relaxation of (2.10) to (2.11) is mainly achieved by replacing $M \in \{-1,+1\}^{n\times n}$ and $\operatorname{rank}(M) = 1$ with $M \in \mathbb{R}^{n\times n}$, $M \succeq \mathbf{0}$ and $\operatorname{diag}(M) = \mathbf{1}_n$. As a result, optimization (2.11) is a *semi-definite programming* (SDP) provided $(\gamma Q \circ M - \eta I_n) \succeq \mathbf{0}$. Let $\boldsymbol{\nu} \geq \mathbf{0}_n$ and $\boldsymbol{\mu} \geq \mathbf{0}_n$ be the Lagrangian multipliers for the constraint $\mathbf{0}_n \leq \boldsymbol{\alpha}$ and $\boldsymbol{\alpha} \leq C\mathbf{1}_n$, then (2.11) is equivalent to

$$\begin{aligned} \min_{M,\boldsymbol{\mu},\boldsymbol{\nu},\eta} \max_{\boldsymbol{\alpha}} \; & 2\boldsymbol{\alpha}^\top(\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu}) - \boldsymbol{\alpha}^\top(\gamma Q \circ M - \eta I_n)\boldsymbol{\alpha} + 2C\boldsymbol{\mu}^\top \mathbf{1}_n - \eta \\ \text{s.t.} \; & M \succeq \mathbf{0} \\ & \operatorname{diag}(M) = \mathbf{1}_n \\ & \boldsymbol{\mu} \geq \mathbf{0}_n, \boldsymbol{\nu} \geq \mathbf{0}_n. \end{aligned} \tag{2.12}$$

[5] We imply by $M \succeq \mathbf{0}$ that M is symmetric and will not explicitly write $M^\top = M$ for convenience.

When considering the variable $\boldsymbol{\alpha}$ in (2.12), the optimal $\boldsymbol{\alpha}$ should be

$$\boldsymbol{\alpha} = (\gamma Q \circ M - \eta I_n)^{\dagger}(\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu}),$$

where † is the operator of the pseudo inverse, and we can form (2.12) into

$$\begin{aligned} \min_{M,\boldsymbol{\mu},\boldsymbol{\nu},\eta} \quad & (\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu})^{\top}(\gamma Q \circ M - \eta I_n)^{\dagger}(\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu}) + 2C\boldsymbol{\mu}^{\top}\mathbf{1}_n - \eta \\ \text{s.t.} \quad & M \succeq \mathbf{0} \\ & \operatorname{diag}(M) = \mathbf{1}_n \\ & \boldsymbol{\mu} \geq \mathbf{0}_n, \boldsymbol{\nu} \geq \mathbf{0}_n \end{aligned} \tag{2.13}$$

under an additional condition that $(\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu})$ is orthogonal to the null space of $(\gamma Q \circ M - \eta I_n)$. Eventually, by the *extended Schur complement lemma* (De Bie and Cristianini, 2004), we arrive at a standard SDP formulation:

$$\begin{aligned} \min_{M,\boldsymbol{\mu},\boldsymbol{\nu},\eta,t} \quad & t \\ \text{s.t.} \quad & M \succeq \mathbf{0} \\ & \operatorname{diag}(M) = \mathbf{1}_n \\ & \boldsymbol{\mu} \geq \mathbf{0}_n, \boldsymbol{\nu} \geq \mathbf{0}_n \\ & \begin{pmatrix} \gamma Q \circ M - \eta I_n & (\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu}) \\ (\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu})^{\top} & t + \eta - 2C\boldsymbol{\mu}^{\top}\mathbf{1}_n \end{pmatrix} \succeq \mathbf{0}. \end{aligned} \tag{2.14}$$

The asymptotic time complexity of optimization (2.14) is $O(n^{6.5})$ if directly solved by any standard SDP solver (De Bie and Cristianini, 2004). It could be reduced to $O(n^{4.5})$ with the *subspace tricks* (De Bie and Cristianini, 2006), which essentially make use of the spectral properties of Q to control the trade-off between the computational cost and the accuracy.

After we obtain M^*, $\boldsymbol{y}^*$ could be recovered from the rank one approximation of M^* by either *thresholding* (De Bie and Cristianini, 2004) or *randomized rounding* (Raghavan and Thompson, 1985; De Bie and Cristianini, 2006). In our experiments, we use the former technique: The eigenvector $\boldsymbol{v}^*$ associated with the largest eigenvalue of M^* is extracted, and then $\boldsymbol{y}^*$ is recovered as

$$\boldsymbol{y}^* = \operatorname{sign}\left(\boldsymbol{v}^* - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^{\top}\boldsymbol{v}^*\right),$$

where the threshold is the center of $\boldsymbol{v}^*$ (cf., the construction of $\boldsymbol{h}_0$ in MVC-SL).

2.4 Generality

MVC is a general framework and closely related to several existing clustering methods. The primal problem of MVC-SL can in fact be reduced to the optimization problems of *unnormalized spectral clustering* (USC) (von Luxburg, 2007, p. 6), relaxed plain and kernel k*-means clustering* (Ding and He, 2004), and *squared-loss mutual information based clustering* (SMIC) (Sugiyama et al., 2011) as special limit cases. We demonstrate these claims in this section.

First of all, consider USC. The relaxed *RatioCut* problem can formulate USC from a graph cut point of view as

$$\min_{f \in \mathbb{R}^n} \ f^\top L_{\mathrm{un}} f \qquad \text{s.t. } f \perp \mathbf{1}_n, \|f\|_2 = \sqrt{n} \tag{2.15}$$

when the number of clusters is two, where L_{un} is the *unnormalized graph Laplacian* (von Luxburg, 2007, pp. 10–11). Note that we can rewrite the primal problem of MVC-SL defined in (2.4) as

$$\min_{\boldsymbol{h} \in \mathbb{R}^n} \ -2\|\boldsymbol{h}\|_1/\gamma + \boldsymbol{h}^\top Q \boldsymbol{h} \qquad \text{s.t. } \|\boldsymbol{h}\|_2 = 1. \tag{2.16}$$

Optimizations (2.16) and (2.4) share exactly the same optimal solution with/without the class balance constraint $-b \leq \boldsymbol{h}^\top \mathbf{1}_n \leq b$, though (2.16) has an optimal objective value γ times smaller than (2.4)'s. Now, let $Q = L_{\mathrm{un}} + \varepsilon I_n$ with arbitrarily chosen $\varepsilon > 0$ to make sure the positive definiteness of Q. Assume that f^* is the solution to (2.15), and $\boldsymbol{h}_m^*$ is the solution to (2.16) with Q specified as above, a class balance parameter $b = 0$, and a regularization parameter $\gamma_m = m$ given a natural number m. Subsequently, it is obvious that

$$\lim_{m \to \infty} \boldsymbol{h}_m^* = f^*/\sqrt{n},$$

and

$$\lim_{m \to \infty} -2\|\boldsymbol{h}_m^*\|_1/\gamma_m + \boldsymbol{h}_m^{*\top} Q \boldsymbol{h}_m^* = f^{*\top} L_{\mathrm{un}} f^*/n + \varepsilon,$$

since $\|\boldsymbol{h}_m^*\|_1 \leq \sqrt{n}\|\boldsymbol{h}_m^*\|_2 = \sqrt{n}$ and then $\lim_{m \to \infty} \|\boldsymbol{h}_m^*\|_1/\gamma_m = 0$. Therefore, USC is a special limit case of MVC-SL, that is, a special case with the specification $Q = L_{\mathrm{un}} + \varepsilon I_n$ of a limit case as $\gamma \to \infty$.

Remark 2.1. The motivation of $f \perp \mathbf{1}_n$ in USC is very different from $\boldsymbol{h}^\top \mathbf{1}_n = 0$ in MVC-SL for class balancing. When L_{un} is constructed from a fully connected similarity graph, the constraint $f \perp \mathbf{1}_n$ means that the feasible region of optimization (2.15) is in a space spanned by all eigenvectors of L_{un} except the trivial eigenvector $\mathbf{1}_n$. Note that $\boldsymbol{h}^\top \mathbf{1}_n = 0$ just asks for strictly balanced soft responses and is not equivalent to $\mathrm{sign}(\boldsymbol{h})^\top \mathbf{1}_n = 0$ that demands strictly balanced cluster assignments.

On the other hand, continuous solutions to the relaxations of k-means clustering (MacQueen, 1967; Hartigan and Wong, 1979) and kernel k-means clustering (Girolami, 2002) can be obtained by principle component analysis (PCA) and kernel PCA respectively (Zha et al., 2002; Ding and He, 2004). Now, let $Q = \varepsilon I_n - C_n K C_n$ with arbitrarily chosen $\varepsilon > \|C_n K C_n\|_2$, where K is the *kernel matrix*, $C_n = I_n - \mathbf{1}_n \mathbf{1}_n^\top / n$ is the centering matrix, and $\|\cdot\|_2$ here means the spectral norm (also known as the operator norm induced by the ℓ_2-norm) of a matrix. As a result,

$$\lim_{m \to \infty} \boldsymbol{h}_m^* = \boldsymbol{v}^*,$$

and

$$\lim_{m \to \infty} -2\|\boldsymbol{h}_m^*\|_1 / \gamma_m + \boldsymbol{h}_m^{*\top} Q \boldsymbol{h}_m^* = \varepsilon - \boldsymbol{v}^* C_n K C_n \boldsymbol{v}^*,$$

where $\boldsymbol{h}_m^*$ is the solution to (2.16) with Q specified as above and $\gamma_m = m$, and $\boldsymbol{v}^*$ is the solution to the relaxed kernel k-means clustering (Ding and He, 2004, Theorem 3.5). In addition, if $\mathcal{X} \subset \mathbb{R}^d$ and $X \in \mathbb{R}^{n \times d}$ is the design matrix, we will have

$$\lim_{m \to \infty} \boldsymbol{h}_m'^* = \boldsymbol{v}'^*,$$

and

$$\lim_{m \to \infty} -2\|\boldsymbol{h}_m'^*\|_1 / \gamma_m + \boldsymbol{h}_m'^{*\top} Q \boldsymbol{h}_m'^* = \varepsilon - \boldsymbol{v}'^* C_n X X^\top C_n \boldsymbol{v}'^*,$$

where $\boldsymbol{h}_m'^*$ is the solution to (2.16) with the specification $Q = \varepsilon I_n - C_n X X^\top C_n, \varepsilon > \|C_n X X^\top C_n\|_2$ and $\gamma_m = m$, and $\boldsymbol{v}'^*$ is the solution to the relaxed k-means clustering (Ding and He, 2004, Theorem 2.2). In other words, k-means clustering and kernel k-means clustering are special limit cases of MVC-SL after relaxations.[6]

[6] When considering k-means algorithms that are referred to as certain iterative clustering algorithms rather than clustering models, by no means they can be closely related to MVC-SL.

Similarly to USC and two k-means clustering, the optimization problem of the binary SMIC is also a special limit case of MVC-SL. It involves maximizing an unsupervised *squared-loss mutual information* approximator, that is,

$$\max_{\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2\in\mathbb{R}^n} \frac{1}{n}\sum_{y=1}^{2}\boldsymbol{\alpha}_y^\top K^2\boldsymbol{\alpha}_y - \frac{1}{2} \tag{2.17}$$

under an orthonormal constraint of $\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2\}$, where $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ are model parameters of posterior probabilities and K is the kernel matrix. The optimal solutions to (2.17) can be obtained through

$$\boldsymbol{\alpha}_1^* = \operatorname*{arg\,max}_{\boldsymbol{\alpha}\in\mathbb{R}^n} \boldsymbol{\alpha}^\top K^2\boldsymbol{\alpha} \qquad \text{s.t. } \|\boldsymbol{\alpha}\|_2 = 1, \tag{2.18}$$

$$\boldsymbol{\alpha}_2^* = \operatorname*{arg\,max}_{\boldsymbol{\alpha}\in\mathbb{R}^n} \boldsymbol{\alpha}^\top K^2\boldsymbol{\alpha} \qquad \text{s.t. } \boldsymbol{\alpha}\perp\boldsymbol{\alpha}_1^*, \|\boldsymbol{\alpha}\|_2 = 1. \tag{2.19}$$

Now, let $Q = \varepsilon I_n - K^2$ with arbitrarily chosen $\varepsilon > \|K\|_2^2$. We could then know

$$\lim_{m\to\infty} \boldsymbol{h}_{1,m}^* = \boldsymbol{\alpha}_1^*,$$

and

$$\lim_{m\to\infty} -2\|\boldsymbol{h}_{1,m}^*\|_1/\gamma_m + \boldsymbol{h}_{1,m}^{*\top} Q\boldsymbol{h}_{1,m}^* = \varepsilon - \boldsymbol{\alpha}_1^* K^2\boldsymbol{\alpha}_1^*,$$

where $\boldsymbol{h}_{1,m}^*$ is the solution to (2.16) with Q specified as above and $\gamma_m = m$. Likewise,

$$\lim_{m\to\infty} \boldsymbol{h}_{2,m}^* = \boldsymbol{\alpha}_2^*,$$

and

$$\lim_{m\to\infty} -2\|\boldsymbol{h}_{2,m}^*\|_1/\gamma_m + \boldsymbol{h}_{2,m}^{*\top} Q\boldsymbol{h}_{2,m}^* = \varepsilon - \boldsymbol{\alpha}_2^* K^2\boldsymbol{\alpha}_2^*,$$

where $\boldsymbol{h}_{2,m}^*$ is the solution to (2.16) with Q specified as above, $\gamma_m = m$ and a variant constraint for class balancing as $\boldsymbol{h}^\top\boldsymbol{h}_{1,m}^* = 0$.

Remark 2.2. After optimizing (2.18) and (2.19), SMIC adopts the post-processing that encloses $\boldsymbol{\alpha}_1^*$ and $\boldsymbol{\alpha}_2^*$ into posterior probabilities and enables the out-of-sample ability to cluster any $x \in \mathcal{X}$ even for $x \notin X_n$ (Sugiyama et al., 2011), while MVC-SL can use

$$\boldsymbol{h}^* = \boldsymbol{\alpha}_1^* \operatorname{sign}(\mathbf{1}_n^\top\boldsymbol{\alpha}_1^*) - \boldsymbol{\alpha}_2^* \operatorname{sign}(\mathbf{1}_n^\top\boldsymbol{\alpha}_2^*)$$

as the optimal soft response vector since there are just two clusters.

2.5 Finite Sample Stability

The stability of the resulting clusters is important for those solved by randomized algorithms (e.g., MVC-SL and k-means clustering) rather than by casting themselves to convex dual problems (e.g., MVC-HL and MMC). In this section, we investigate the stability of the primal problem of MVC-SL in the finite sample scenario.

In the following, we presume that we are always able to find a locally optimal solution to optimization (2.4) accurately. Under this presumption, we prove that the instability is resulted from the symmetry of data samples: As long as the input matrix Q satisfies some asymmetry condition, we could obtain the same data partition based on different locally optimal solutions, and consequently the non-convex optimization of MVC-SL seems convex.

2.5.1 Definitions

Definition 2.3. *The Hamming clustering distance for two n-dimensional soft response vectors $\boldsymbol{h}$ and $\boldsymbol{h}'$ is defined as*

$$d_{\mathcal{H}}(\boldsymbol{h}, \boldsymbol{h}') := \frac{1}{2} \min(\| \operatorname{sign}(\boldsymbol{h}) + \operatorname{sign}(\boldsymbol{h}')\|_1, \| \operatorname{sign}(\boldsymbol{h}) - \operatorname{sign}(\boldsymbol{h}')\|_1).$$

When measuring the difference of two binary clusterings, $d_{\mathcal{H}}(\boldsymbol{h}, \boldsymbol{h}')$ is always a natural number smaller than $n/2$, since $\| \operatorname{sign}(\boldsymbol{h}) + \operatorname{sign}(\boldsymbol{h}')\|_1 + \| \operatorname{sign}(\boldsymbol{h}) - \operatorname{sign}(\boldsymbol{h}')\|_1 = 2n$.

Definition 2.4 (Irreducibility)**.** *A sample x_i is isolated in X_n, if $Q_{i,i} > 0$ and $\forall j \neq i, Q_{i,j} = 0$. A set of samples X_n is irreducible, if no sample is isolated in X_n; otherwise X_n is reducible.*

The idea behind the irreducibility of X_n is simple: If x_i is isolated, we cannot decide its cluster based on the information contained in Q no matter what binary clustering algorithm is used, unless we assign x_i to one cluster and $X_n \setminus x_i$ to the other cluster. We would like to remove such isolated samples and reduce the clustering of X_n to a better-defined problem.

Next we define two symmetry concepts, the submatrix-information- (SI- for short) symmetry in Definition 2.5 and the axisymmetry in Definition 2.7. SI-

asymmetry is a part of the sufficient condition for finite sample stability, and axisymmetry is a part of the sufficient condition for instability. The relationship of irreducibility, axisymmetry and SI-symmetry will be proved in Theorem 2.10.

Definition 2.5 (Submatrix-Information-Symmetry)**.** *A set of samples X_n is submatrix-information-symmetric, if there exist $\{\delta_1, \ldots, \delta_n\} \in \{-1, +1\}^n$ and nonempty $\mathcal{K} \subsetneq \{1, \ldots, n\}$ such that*

$$\sum\nolimits_{i\in\mathcal{K}, j\notin\mathcal{K}, \delta_i=\delta_j} Q_{i,j} = \sum\nolimits_{i\in\mathcal{K}, j\notin\mathcal{K}, \delta_i\neq\delta_j} Q_{i,j}. \tag{2.20}$$

Otherwise, X_n is submatrix-information-asymmetric.[7]

Remark 2.6. It is clear that

$$\sum\nolimits_{i\in\mathcal{K}, j\notin\mathcal{K}, \delta_i=\delta_j} Q_{i,j} = \sum\nolimits_{i\in\mathcal{K}, j\notin\mathcal{K}} \delta_i\delta_j Q_{i,j},$$
$$\sum\nolimits_{i\in\mathcal{K}, j\notin\mathcal{K}, \delta_i\neq\delta_j} Q_{i,j} = -\sum\nolimits_{i\in\mathcal{K}, j\notin\mathcal{K}} \delta_i\delta_j Q_{i,j},$$

and thus Eq. (2.20) is equivalent to

$$\left(\sum\nolimits_{k\in\mathcal{K}} \delta_k \boldsymbol{e}_k\right)^\top Q \left(\sum\nolimits_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k\right) = 0, \tag{2.21}$$

where $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ is a standard basis for $\mathbb{R}^n$. From now on, we may use Eq. (2.21) as the condition to check SI-symmetry or SI-asymmetry for convenience.

Intuitively, the SI-symmetry of X_n says that Q has a submatrix containing the same amount of similarity and dissimilarity information. More specifically, both $\{\delta_1, \ldots, \delta_n\}$ and $\mathcal{K}$ are valid partitions of X_n, though they have different representations and functions. The partition $\{\delta_1, \ldots, \delta_n\}$ is a reference for similarity and dissimilarity, and based on this partition, we categorize the information $Q_{i,j}$ between x_i and x_j into similarity information if $\delta_i = \delta_j$ or dissimilarity information if $\delta_i \neq \delta_j$. On the other hand, we divide Q into four parts $Q[i \in \mathcal{K}; j \in \mathcal{K}]$,

[7] Strictly speaking, saying that X_n is SI-symmetric is a bit abuse of terminology. In formal mathematical terminology, an object is symmetric with respect to some operation, if this operation, when applied to the object, preserves certain property. For example, in the axisymmetry, the object is X_n, the operation is ϕ and the property is Q. However, in the SI-symmetry, the object is a set of two vectors $\{\sum_{k\in\mathcal{K}} \delta_k \boldsymbol{e}_k, \sum_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k\}$, the operation is replacing I_n with Q, and the property is the orthogonality (preserved from the standard orthogonality to the Q-orthogonality).

$Q[i \in \mathcal{K}; j \notin \mathcal{K}]$, $Q[i \notin \mathcal{K}; j \in \mathcal{K}]$ and $Q[i \notin \mathcal{K}; j \notin \mathcal{K}]$ according to the partition $\mathcal{K}$. The SI-symmetry of X_n shown in Eq. (2.20) indicates that the submatrix $Q[i \in \mathcal{K}; j \notin \mathcal{K}]$ (likewise $Q[i \notin \mathcal{K}; j \in \mathcal{K}]$) contains the same amount of similarity information (i.e., the left-hand side) and dissimilarity information (i.e., the right-hand side).

When X_n is SI-symmetric, we can easily find two feasible solutions to optimization (2.4), such that they result in different partitions of X_n with the same value of the objective function. To see this, let

$$\boldsymbol{h}_+ = \frac{\sum_{k\in\mathcal{K}} \delta_k \boldsymbol{e}_k + \sum_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k}{\sqrt{n}},$$
$$\boldsymbol{h}_- = \frac{\sum_{k\in\mathcal{K}} \delta_k \boldsymbol{e}_k - \sum_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k}{\sqrt{n}}.$$

It is easy to verify that $\|\boldsymbol{h}_\pm\|_2 = 1$, $\|\boldsymbol{h}_\pm\|_1 = \sqrt{n}$ and $d_{\mathcal{H}}(\boldsymbol{h}_+, \boldsymbol{h}_-) \geq 1$. Moreover,

$$\boldsymbol{h}_+^\top Q \boldsymbol{h}_+ = \boldsymbol{h}_+^\top Q \boldsymbol{h}_+ - (\boldsymbol{h}_+ + \boldsymbol{h}_-)^\top Q (\boldsymbol{h}_+ - \boldsymbol{h}_-) = \boldsymbol{h}_-^\top Q \boldsymbol{h}_-,$$

where we used $(\boldsymbol{h}_+ + \boldsymbol{h}_-)^\top Q (\boldsymbol{h}_+ - \boldsymbol{h}_-) = 0$ by the condition Eq. (2.21) of SI-symmetry. Note that, however, $\boldsymbol{h}_\pm$ are not necessarily locally optimal solutions to (2.4) and there may be no solution that can result in the same partition with $\boldsymbol{h}_+$ or $\boldsymbol{h}_-$. The real reason for finite sample instability is the axisymmetry of data samples defined below.

Definition 2.7 (Axisymmetry)**.** *A set of samples X_n is axisymmetric, if there exists a permutation $\phi : \{1, \ldots, n\} \mapsto \{1, \ldots, n\}$ such that*

1. $\exists i \in \{1, \ldots, n\}, \phi(i) \neq i$;
2. $\forall i \in \{1, \ldots, n\}, \phi^{-1}(i) = \phi(i)$;
3. $\forall 1 \leq i, j \leq n, Q_{i,j} = Q_{\phi(i),\phi(j)}$.

The first property says that the permutation ϕ cannot be the identical mapping: It allows some, but not all, $x \in X_n$ mapped to x itself. The second property requires that those samples $x \in X_n$ mapped to others are all paired. In other words, X_n is separated into two types of disjoint subsets according to ϕ, either cardinality one (i.e., $\{x_i \mid \phi(i) = i\}$) or two (i.e., $\{x_i, x_{\phi(i)} \mid \phi(i) \neq i\}$), but no greater cardinality. The third property guarantees that Q is ϕ-invariant, or

equivalently the pair $\{x_i, x_{\phi(i)} \mid \phi(i) \neq i\}$ cannot be distinguished by all other single points $\{x_j \mid \phi(j) = j, j \neq i\}$ or pairs $\{x_j, x_{\phi(j)} \mid \phi(j) \neq j, j \neq i, j \neq \phi(i)\}$ based on the information contained in Q, so we can exchange x_i and $x_{\phi(i)}$ freely without modifying Q.

The axisymmetry of X_n in terms of Q is equivalent to the axisymmetry of X_n in $\mathcal{X}$, if $\mathcal{X} \subset \mathbb{R}^d$ and Q is induced from the Euclidean distance such as Gaussian kernel matrices. For example, as shown in Figure 2.2,

$$X_4 = \{(0,0), (1,0), (1,1), (0,1)\},$$
$$X_4' = \{(0,0), (1,0), (1,0.5), (0,0.5)\}$$

are axisymmetric both in $\mathbb{R}^2$ and in terms of Q if Q is a Gaussian kernel matrix, regardless of the kernel width. The permutation ϕ for X_4' could be

$$\{(1,2),(3,4)\}, \{(1,3),(2,4)\}, \{(1,4),(2,3)\},$$

and besides them, ϕ for X_4 could also be

$$\{(1),(3),(2,4)\}, \{(1,3),(2),(4)\}.$$

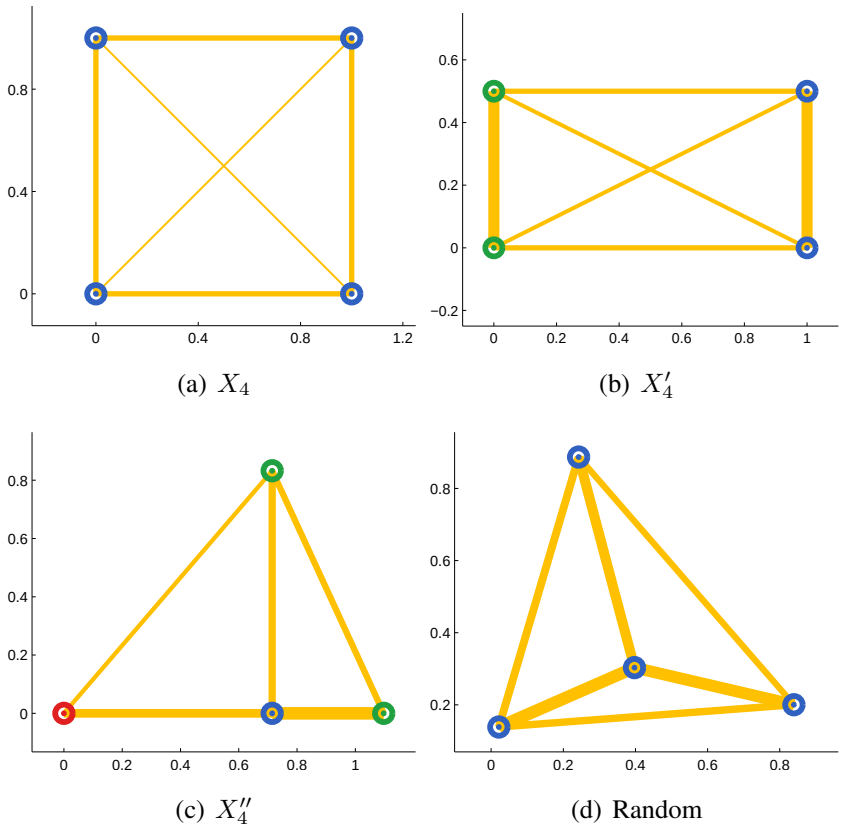
We can identify an axis of symmetry geometrically in $\mathbb{R}^d$: It passes through either x_i if $\phi(i) = i$ or $(x_i + x_{\phi(i)})/2$ if $\phi(i) \neq i$ for $i = 1, \ldots, n$. This is why we call such a property axisymmetry.

Generally speaking, the concepts of axisymmetry and SI-symmetry almost coincide, if Q is a Gaussian kernel matrix or the corresponding graph Laplacian matrix. While it is possible to deliberately construct counter-examples that are SI-symmetric but not axisymmetric, it is improbable to meet a counter-example in practice. For instance, as illustrated in panel (c) of Figure 2.2,

$$\begin{aligned} X_4'' &= \{(0,0), (\sqrt{\ln(5/3)}, 0), (\sqrt{\ln(10/3)}, 0), (\sqrt{\ln(5/3)}, \sqrt{\ln 2})\} \\ &\approx \{(0,0), (0.7147, 0), (1.0973, 0), (0.7147, 0.8326)\} \end{aligned}$$

is SI-symmetric but not axisymmetric in terms of Gaussian kernel matrix Q when $\sigma = 1/\sqrt{2}$, yet X_4'' is SI-asymmetric whenever $\sigma \neq 1/\sqrt{2}$.

Definition 2.8 (Anisotropy)**.** *A set of samples X_n is anisotropic, if Q has n distinct eigenvalues.*



(a) X_4 (b) X_4'

(c) X_4'' (d) Random

Figure 2.2: Four-point sets that are typical in the theory of finite sample stability. Gaussian similarities ($\sigma = 1/\sqrt{2}$) between nodes are visualized by the line thickness of edges. All sets in this figure are irreducible. X_4 in panel (a) is axisymmetric and SI-symmetric. X_4' in (b) is axisymmetric, SI-symmetric, anisotropic, and has a unique best partition. X_4'' in (c) is very special: It is anisotropic, *SI-symmetric but not axisymmetric*, since the similarity of the red and blue points equals the sum of the similarities of the red and green points. A random set would be anisotropic and SI-asymmetric with high probability.

The anisotropy of X_n is the other part of the sufficient condition for finite sample stability. The name comes from a geometric interpretation of the ellipsoid $\mathcal{E}(\mathcal{H}_Q)$: All its principal axes achieve distinct lengths when Q has distinct eigenvalues, and thus $\mathcal{E}(\mathcal{H}_Q)$ is anisotropic and not rotatable. The concepts of anisotropy and axisymmetry are not complementary, since they concern different aspects of different objects, that is, the rotation of $\mathcal{E}(\mathcal{H}_Q)$ vs. the reflection of X_n. In Figure 2.2, X_4 is axisymmetric, X_4' is axisymmetric and anisotropic, and most random sets are just anisotropic. There might be X_n neither axisymmetric nor anisotropic. Nonetheless, when considering the more general SI-symmetry and certain families of Q such as Gaussian kernel matrices, X_n is anisotropic as long as it is SI-asymmetric.

All definitions have been discussed. The theoretical results will be presented next.

2.5.2 Theoretical Results

The following lemma will be used in Theorems 2.11 and 2.13. All proofs are provided in Section 2.9.

Lemma 2.9. *Let X_n be an irreducible set of samples, $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be the normalized eigenvectors of Q, and $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ be a standard basis for $\mathbb{R}^n$. Then, $\forall i, j \in \{1, \ldots, n\}$, $\boldsymbol{v}_i \neq \pm \boldsymbol{e}_j$.*

The following two theorems describe the relationship between the properties defined above.

Theorem 2.10. *A set of samples X_n is SI-symmetric, if it is reducible or axisymmetric.*

Theorem 2.11. *If X_n is an SI-asymmetric set of samples, and there exists $\kappa > 0$ such that $Q_{1,1} = Q_{2,2} = \cdots = Q_{n,n} = \kappa$, then X_n is anisotropic.*

We are ready to deliver our main theorems. To begin with, given a constant η, we define (recall the assumption that $\|\boldsymbol{h}\|_1$ is differentiable thanks to the non-

sparsity of $\boldsymbol{h}$)

$$G(\boldsymbol{h}) := \gamma \boldsymbol{h}^\top Q \boldsymbol{h} - \eta \|\boldsymbol{h}\|_2^2 - 2\|\boldsymbol{h}\|_1,$$
$$g(\boldsymbol{h}) := \frac{1}{2}\nabla G(\boldsymbol{h}) = \gamma Q \boldsymbol{h} - \eta \boldsymbol{h} - \operatorname{sign}(\boldsymbol{h}).$$

Theorem 2.12 (**Twin Minimum Theorem**). *Assume that $n > 2$, X_n is an axisymmetric set of samples, ϕ is the corresponding permutation, and $\mathcal{I} = \{\{i, \phi(i)\} \mid \phi(i) \neq i\}$ is the index set of those paired samples given ϕ. For every minimum $\boldsymbol{h}^*$ of optimization* (2.4)*, if*

1. $\forall i, [\boldsymbol{h}^*]_i \neq 0$*, and*
2. $\exists \{i, \phi(i)\} \in \mathcal{I}, [\boldsymbol{h}^*]_{\phi(i)}[\boldsymbol{h}^*]_i < 0$*,*

then $\boldsymbol{h}^*$ has a twin minimum $\boldsymbol{h}^\star$ satisfying $G(\boldsymbol{h}^\star) = G(\boldsymbol{h}^*)$ and $d_{\mathcal{H}}(\boldsymbol{h}^*, \boldsymbol{h}^\star) \geq 1$. The only exception is

$$\forall i \in \{1, \ldots, n\}, [\boldsymbol{h}^*]_{\phi(i)}[\boldsymbol{h}^*]_i < 0.$$

In order to explain the implication of Theorem 2.12, let us recall X_4 and X_4' in Figure 2.2. There are many twin minima when considering the perfectly symmetric X_4, while it is also an unstable input even for those convex relaxations of MMC due to the post-processing. On the other hand, X_4' illustrates an exception: While X_4 allows $\phi(i) = i$, it is impossible for X_4'. More specifically, any minimum $\boldsymbol{h}^*$ corresponding to partition $(+1, -1, -1, +1)$ has no twin minimum, since ϕ could be $\{(1,2),(3,4)\}$, $\{(1,3),(2,4)\}$ or $\{(1,4),(2,3)\}$ for X_4', and

$$\forall \phi, (\exists i, [\boldsymbol{h}^*]_{\phi(i)}[\boldsymbol{h}^*]_i < 0) \rightarrow (\forall i, [\boldsymbol{h}^*]_{\phi(i)}[\boldsymbol{h}^*]_i < 0).$$

It suggests that if we permute $\boldsymbol{h}^*$ according to ϕ, the resultant $\operatorname{sign}(\boldsymbol{h}^\star) = \pm \boldsymbol{y}$ is the same partition and thus $d_{\mathcal{H}}(\boldsymbol{h}^*, \boldsymbol{h}^\star) = 0$. Another minimum $\boldsymbol{h}$ that corresponds to $(+1, +1, -1, -1)$ and satisfies $d_{\mathcal{H}}(\boldsymbol{h}^*, \boldsymbol{h}) \geq 1$ should have $G(\boldsymbol{h}) > G(\boldsymbol{h}^*)$. In a word, local minima corresponding to different partitions for X_4' are not equally good and the best partition is unique, as illustrated in panel (b) of Figure 2.2. The genuine instability emerges only when the best partition is not unique, like X_4.

Theorem 2.13 (**Equivalent Minima Theorem**). *All minima of optimization* (2.4) *are equivalent with respect to $d_{\mathcal{H}}$, provided that*

1. *X_n is SI-asymmetric;*
2. *X_n is anisotropic.*

By combining Theorem 2.11 and Theorem 2.13, we have a corollary immediately.

Corollary 2.14. *All minima of optimization* (2.4) *are equivalent with respect to $d_{\mathcal{H}}$, provided that*

1. *X_n is SI-asymmetric;*
2. *There exists $\kappa > 0$ such that $Q_{1,1} = Q_{2,2} = \cdots = Q_{n,n} = \kappa$.*

To sum up, as long as Q has the two properties listed above, different locally optimal solutions to optimization (2.4) would induce the same data partition. Nevertheless, the output of the algorithm is not in the same form as the solution to optimization (2.4), since the variable η has been introduced, and we cannot foresee its optimal value when we analyze the original model. Spectral clustering is consistent (von Luxburg et al., 2008), but it has a similar problem in finite sample stability, that is, when the graph Laplacian has distinct eigenvalues and the unique spectral decomposition leads to a stable spectral embedding, the following k-means clustering can still introduce high instability due to the non-convex nature of the distortion function.

Remark 2.15. We rely on a Karush-Kuhn-Tucker stationarity condition $g(\boldsymbol{h}^*) = \mathbf{0}_n$ in the proofs of Theorems 2.12 and 2.13, where $\boldsymbol{h}^*$ is the optimal solution to (2.4). Actually, the objective of (2.4) usually has a non-zero derivative, and the objective of (2.3) always has a non-zero derivative in the feasible regions. Therefore, we introduce the function $g(\boldsymbol{h})$ to analyze MVC-SL from a theoretical point of view. In MVC-SL, Eq. (2.7) comes from the least-square fit of Eq. (2.8), and if $t \to \infty$, we will have $\boldsymbol{p}_t^* \to \mathbf{0}_n$ and then Eq. (2.8) will turn into $g(\boldsymbol{h}^*) = \lim_{t\to\infty} g(\boldsymbol{h}_t) = \mathbf{0}_n$.

2.6 Data-dependent Error Bound

In this section, we derive a data-dependent error bound for MVC-SL based on the theory of *transductive Rademacher complexity* (El-Yaniv and Pechyony, 2009).

It is extremely difficult, if possible, to evaluate clustering methods in an objective and domain-independent manner (von Luxburg et al., 2012). However, when the goals and interests are clear, it makes sense to evaluate clustering results using classification benchmark data sets, where the class structure coincides with the desired cluster structure according to the goals and interests.

In real-world applications, we often find some experts to label a small portion $X_{n'}$ of X_n where $n' < n$ according to their professional knowledge, run a pool of candidate clustering algorithms, see their agreement with the labels and eliminate those low agreement algorithms. This procedure may be viewed as propagating the knowledge of domain experts from $X_{n'}$ to X_n.

Here, we present a data-dependent error bound to ensure the quality of the knowledge propagation. The key technique is called transductive Rademacher complexity for deriving data-dependent transductive error bounds. To begin with, we define transductive Rademacher complexity based on El-Yaniv and Pechyony (2009) as follows.

Definition 2.16. *Fix positive integers m and u. Let $\mathcal{H} \subseteq \mathbb{R}^{m+u}$ be a hypothesis space, $p \in [0, 1/2]$ be a parameter, and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_{m+u})^\top$ be a vector of independent and identically distributed random variables, such that*

$$\sigma_i := \begin{cases} +1 & \text{with probability} \quad p, \\ -1 & \text{with probability} \quad p, \\ 0 & \text{with probability} \quad 1 - 2p. \end{cases}$$

Then, the transductive Rademacher complexity of $\mathcal{H}$ with parameter p is defined as

$$\mathcal{R}_{m+u}(\mathcal{H}, p) := \left(\frac{1}{m} + \frac{1}{u} \right) \mathbb{E}_{\boldsymbol{\sigma}} \left\{ \sup_{\boldsymbol{h} \in \mathcal{H}} \boldsymbol{\sigma}^\top \boldsymbol{h} \right\}.$$

For the sake of comparison, we give a definition of inductive Rademacher complexity following El-Yaniv and Pechyony (2009).[8]

Definition 2.17. *Let $p(x)$ be a probability density on $\mathcal{X}$, and suppose that $X_n = \{x_1, \ldots, x_n\}$ are independent observations drawn from $p(x)$. Let $\mathcal{H}$ be a class of*

[8] Albeit there are many definitions of Rademacher complexity, for example, Koltchinskii (2001), Bartlett and Mendelson (2002), Meir and Zhang (2003) and Bousquet et al. (2004), they are similar and conceptually equivalent.

functions from $\mathcal{X}$ to $\mathbb{R}$, and $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_n)^\top$ be a vector of independent and identically distributed random variables, such that

$$\sigma_i := \begin{cases} +1 & \text{with probability} \quad 1/2, \\ -1 & \text{with probability} \quad 1/2. \end{cases}$$

The empirical Rademacher complexity of $\mathcal{H}$ conditioned on X_n is

$$\widehat{\mathcal{R}}_n^{(ind)}(\mathcal{H}) := \frac{2}{n}\mathbb{E}_{\boldsymbol{\sigma}}\left\{\sup_{h\in\mathcal{H}} \boldsymbol{\sigma}^\top \boldsymbol{h} \mid X_n\right\},$$

where $\boldsymbol{h} = (h(x_1), \ldots, h(x_n))^\top$, and the inductive Rademacher complexity of $\mathcal{H}$ is

$$\mathcal{R}_n^{(ind)}(\mathcal{H}) := \mathbb{E}_{X_n}\left\{\widehat{\mathcal{R}}_n^{(ind)}(\mathcal{H})\right\}.$$

The transductive Rademacher complexity of $\mathcal{H}$ is an empirical quantity that depends only on p. Given the data X_n, we have that $\mathcal{R}_{m+u}(\mathcal{H}) = 2\widehat{\mathcal{R}}_{m+u}^{(ind)}(\mathcal{H})$, if $p = 1/2$ and $m = u$.[9] Whenever $p < 1/2$, some Rademacher variables will attain zero values and reduce the complexity. We simply consider $p_0 = mu/(m+u)^2$ and abbreviate $\mathcal{R}_{m+u}(\mathcal{H}, p_0)$ to $\mathcal{R}_{m+u}(\mathcal{H})$ as El-Yaniv and Pechyony (2009) in Lemma 2.18 and Theorem 2.19, though these theoretical results hold for all $p > p_0$, since $\mathcal{R}_{m+u}(\mathcal{H}, p)$ is monotonically increasing with p. Please see El-Yaniv and Pechyony (2009) for the detailed discussions about transductive Rademacher complexity.

Lemma 2.18. *Let $\widetilde{\mathcal{H}}_Q$ be the set of all possible $\boldsymbol{h}$ returned by Algorithm 1 for the given Q, η^* be the optimal η when Algorithm 1 stops,*

$$\mu = \sup_{\boldsymbol{h}\in\widetilde{\mathcal{H}}_Q} \operatorname{sign}(\boldsymbol{h})^\top(\gamma Q - \eta^* I_n)^{-1}\operatorname{sign}(\boldsymbol{h}),$$

and $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of Q. Then, for the transductive Rademacher complexity of $\widetilde{\mathcal{H}}_Q$, the following upper bound holds for any integer $n' = 1, 2, \ldots, n-1$,

$$\mathcal{R}_n(\widetilde{\mathcal{H}}_Q) \le \sqrt{\frac{2}{n'(n-n')}} \min\left\{\sqrt{n}, \left(\sum_{i=1}^{n} \frac{n}{(\gamma\lambda_i - \eta^*)^2}\right)^{1/2}, \left(\sum_{i=1}^{n} \frac{\mu}{\gamma\lambda_i - \eta^*}\right)^{1/2}\right\}.$$

[9] A class of functions conditioned on fixed data is equivalent to a hypothesis space of soft response vectors.

The proof of Lemma 2.18 can be found in Section 2.9. By Lemma 2.18 together with Theorem 2 of El-Yaniv and Pechyony (2009), we can derive immediately the data-dependent error bound:

Theorem 2.19. *Assume that the ground truth partition on X_n is $\boldsymbol{y}^*$, and $\mathcal{L}$ is chosen uniformly over $\{\mathcal{L} \mid \mathcal{L} \subset \{1, \ldots, n\}, |\mathcal{L}| = n'\}$. Let $\ell(z) = \min(1, \max(0, 1 - z))$ for $z \in \mathbb{R}$ be the ramp loss, $\widetilde{\mathcal{H}}_Q$ be the set of all possible $\boldsymbol{h}$ returned by Algorithm 1 for the given Q, η^* be the optimal η when Algorithm 1 stops,*

$$\mu = \sup_{\boldsymbol{h} \in \widetilde{\mathcal{H}}_Q} \operatorname{sign}(\boldsymbol{h})^\top (\gamma Q - \eta^* I_n)^{-1} \operatorname{sign}(\boldsymbol{h}),$$

$\lambda_1, \ldots, \lambda_n$ be the eigenvalues of Q, and $c_0 = \sqrt{32(1 + \ln 4)/3}$. For any $\boldsymbol{h} \in \widetilde{\mathcal{H}}_Q$, with probability at least $1 - \delta$ over the choice of $\mathcal{L}$, we have

$$\begin{aligned} d_{\mathcal{H}}(\operatorname{sign}(\boldsymbol{h}), \boldsymbol{y}^*) \leq{}& \frac{n}{n'} \min \left\{ \sum_{i \in \mathcal{L}} \ell([\boldsymbol{h}]_i [\boldsymbol{y}^*]_i), \sum_{i \in \mathcal{L}} \ell(-[\boldsymbol{h}]_i [\boldsymbol{y}^*]_i) \right\} \\ &+ \frac{c_0 n}{\sqrt{n'}} + \sqrt{\frac{2n^2(n - n')^2}{n'(2n - 1)(2n - 2n' - 1)} \ln(1/\delta)} \\ &+ \sqrt{\frac{2(n - n')}{n'}} \min \left\{ \sqrt{n}, \left(\sum_{i=1}^{n} \frac{n}{(\gamma \lambda_i - \eta^*)^2} \right)^{1/2}, \left(\sum_{i=1}^{n} \frac{\mu}{\gamma \lambda_i - \eta^*} \right)^{1/2} \right\}. \end{aligned} \tag{2.22}$$

There are four terms in the right-hand side of inequality (2.22). The first term is a measure of the clustering error on $X_{n'} = \{x_i \mid i \in \mathcal{L}\}$ by the ramp loss times the ratio n/n'. More specifically, we would like to select a proper similarity measure via the given labels $\{[\boldsymbol{y}^*]_i \mid i \in \mathcal{L}\}$ to make the error on $X_{n'}$ as small as possible, under the assumption that the error rates on $X_{n'}$ and X_n should be close for a fixed similarity measure (the given labels are not used for training). The second term depends only on n and n', i.e., the sizes of the whole set and the labeled subset. Besides n and n', the third term further depends on the significance level δ, as in common error bounds. The last term is the upper bound of $(n - n')\mathcal{R}_n(\widetilde{\mathcal{H}}_Q)$, which carries out the complexity control of $\widetilde{\mathcal{H}}_Q$ implicitly: The smaller the value of $\mathcal{R}_n(\widetilde{\mathcal{H}}_Q)$ is, the more confident we are that $d_{\mathcal{H}}(\operatorname{sign}(\boldsymbol{h}), \boldsymbol{y}^*)$ will be small if the error on $X_{n'}$ is small. The order of the bound is roughly $O(n/\sqrt{n'})$ if the clustering error is measured by $d_{\mathcal{H}}$, and it is roughly $O(1/\sqrt{n'})$ if the clustering error is measured by $d_{\mathcal{H}}/n$.

Remark 2.20. Our problem setting is equivalent to neither semi-supervised clustering nor transductive classification: We do not reveal any labels to the clustering algorithm in Theorem 2.19; instead, a set of randomly chosen labels are revealed to an evaluator who then returns an evaluation of the quality of any possible partition generated by the algorithm. We can use the theory of transductive Rademacher complexity to derive a data-dependent error bound for our clustering algorithm, since it can be viewed as a transductive algorithm that ignores all revealed labels.

2.7 Related Works

In this section, we review related works and qualitatively compare the proposed MVC with them.

2.7.1 Maximum Margin Clustering

Among existing clustering methods, *maximum margin clustering* (MMC) is closest to MVC. Both of them come from the statistical learning theory, but their geneses and underlying criteria are still different: The primal problems of various MMC models adopt a regularizer $\|\boldsymbol{w}\|_2^2$ originated from the margin, while MVC relies on the regularizer $V(\boldsymbol{h})$ in Eq. (2.1) originated from the volume. The hypothesis shared by all MMC is the hyperplane for induction, whereas the hypothesis in MVC is the soft response vector for transduction.

The family of MMC algorithms was initiated by Xu et al. (2005). It follows the support vector machine (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) and its hard-margin version can be formulated as

$$\min_{\boldsymbol{y}\in\{-1,+1\}^n} \min_{\boldsymbol{w},\boldsymbol{\xi}} \ \|\boldsymbol{w}\|_2^2$$
$$\text{s.t.} \ \ y_i\boldsymbol{w}^\top x_i \geq 1, \quad i = 1,\ldots,n.$$

The value of $y_i\boldsymbol{w}^\top x_i$ is called the functional margin of (x_i, y_i), and the value of $y_i\boldsymbol{w}^\top x_i/\|\boldsymbol{w}\|_2$ is called the geometric margin of (x_i, y_i). MMC can maximize the geometric margin of all $x_i \in X_n$ over $\boldsymbol{y} \in \{-1,+1\}^n$ by minimizing $\|\boldsymbol{w}\|_2$ and requiring the minimal functional margin to be one simultaneously. Likewise, the

primal problem of the soft-margin MMC is

$$\min_{\boldsymbol{y}\in\{-1,+1\}^n} \min_{\boldsymbol{w},\boldsymbol{\xi}} \ \|\boldsymbol{w}\|_2^2 + C\sum_{i=1}^{n}\xi_i$$
$$\text{s.t. } y_i\boldsymbol{w}^\top x_i \geq 1-\xi_i, \xi_i \geq 0, \quad i=1,\ldots,n,$$

where $C > 0$ is a regularization parameter, and $\boldsymbol{\xi} \geq \mathbf{0}_n$ is a vector of slack variables. Then, it can be relaxed into a standard SDP dual

$$\begin{aligned} \min_{M,\boldsymbol{\mu},\boldsymbol{\nu},t} \ & t \\ \text{s.t.} \ & M \succeq \mathbf{0} \\ & \operatorname{diag}(M) = \mathbf{1}_n \\ & \boldsymbol{\mu} \geq \mathbf{0}_n, \boldsymbol{\nu} \geq \mathbf{0}_n \\ & \begin{pmatrix} M \circ K & (\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu}) \\ (\mathbf{1}_n - \boldsymbol{\mu} + \boldsymbol{\nu})^\top & t - 2C\boldsymbol{\mu}^\top\mathbf{1}_n \end{pmatrix} \succeq \mathbf{0}, \end{aligned} \tag{2.23}$$

and solved by any standard SDP solver in $O(n^{6.5})$ time.

Remark 2.21. Xu et al. (2005) initially imposed three groups of linear constraints on the entries of M in MMC:

1. $\forall ijk$, $M_{i,k} \geq M_{i,j} + M_{j,k} - 1$;
2. $\forall ijk$, $M_{i,k} \geq -M_{i,j} - M_{j,k} - 1$;
3. $\forall i$, $-b \leq \sum_j M_{i,j} \leq b$.

However, Xu and Schuurmans (2005) and Valizadegan and Jin (2007) considered (2.23) as the dual problem of MMC, sometimes with an additional class balance constraint $-b\mathbf{1}_n \leq M\mathbf{1}_n \leq b\mathbf{1}_n$. In other words, the first and second groups of constraints were ignored.

Subsequently, *generalized maximum margin clustering* (GMMC) (Valizadegan and Jin, 2007) relaxes the restriction that the original MMC only considers homogeneous hyperplanes and hence demands every possible clustering boundary to pass through the origin. Furthermore, GMMC is a convex relaxation of MMC, and its computational complexity is $O(n^{4.5})$ that is remarkably faster than

MMC. In fact, GMMC optimizes an n-dimensional vector rather than an $n \times n$ matrix. More specifically, the hard-margin GMMC converts the original MMC following Lanckriet et al. (2004) into a dual problem as

$$\min_{\boldsymbol{y}\in\{-1,+1\}^n} \min_{\boldsymbol{\nu},\lambda} \frac{1}{2}(\mathbf{1}_n + \boldsymbol{\nu} + \lambda\boldsymbol{y})^\top \operatorname{diag}(\boldsymbol{y}) K^{-1} \operatorname{diag}(\boldsymbol{y})(\mathbf{1}_n + \boldsymbol{\nu} + \lambda\boldsymbol{y}) \quad \text{s.t. } \boldsymbol{\nu} \geq \mathbf{0}_n,$$

where the function $\operatorname{diag}(\cdot)$ here converts a column vector into a diagonal matrix. The trick here is

$$\left(K \circ \boldsymbol{y}\boldsymbol{y}^\top\right)^{-1} = (\operatorname{diag}(\boldsymbol{y}) K \operatorname{diag}(\boldsymbol{y}))^{-1} = \operatorname{diag}(\boldsymbol{y}) K^{-1} \operatorname{diag}(\boldsymbol{y}),$$

since $\boldsymbol{y} \in \{-1, +1\}^n$. By a tricky substitution $\boldsymbol{w} = (\operatorname{diag}(\boldsymbol{y})(\mathbf{1}_n + \boldsymbol{\nu}); \lambda) \in \mathbb{R}^{n+1}$ where we use the semicolon to separate the rows of a vector or matrix (i.e., $(A; B) = (A^\top, B^\top)^\top$), it becomes

$$\begin{aligned} \min_{\boldsymbol{w}\in\mathbb{R}^{n+1}} \quad & \boldsymbol{w}^\top (I_n; \mathbf{1}_n^\top) K^{-1} (I_n, \mathbf{1}_n) \boldsymbol{w} + C_e \left((\mathbf{1}_n^\top, 0)\boldsymbol{w}\right)^2 \\ \text{s.t.} \quad & [\boldsymbol{w}]_i^2 \geq 1, i = 1, \ldots, n, \end{aligned} \tag{2.24}$$

where $((\mathbf{1}_n^\top, 0)\boldsymbol{w})^2$ is another regularization to remove the translation invariance from the objective function and C_e is the corresponding regularization parameter. Let

$$W = (I_n; \mathbf{1}_n^\top) K^{-1} (I_n, \mathbf{1}_n) + C_e (\mathbf{1}_n; 0)(\mathbf{1}_n^\top, 0) - \operatorname{diag}((\boldsymbol{\gamma}; 0)).$$

The SDP dual of optimization (2.24) is then

$$\max_{\boldsymbol{\gamma}\in\mathbb{R}^n} \boldsymbol{\gamma}^\top \mathbf{1}_n \qquad \text{s.t. } W \succeq \mathbf{0}, \boldsymbol{\gamma} \geq \mathbf{0}_n.$$

This is the dual problem of the hard-margin GMMC. The dual problem of the soft-margin GMMC is slightly different such that $\boldsymbol{\gamma}$ is upper bounded:

$$\max_{\boldsymbol{\gamma}\in\mathbb{R}^n} \boldsymbol{\gamma}^\top \mathbf{1}_n \qquad \text{s.t. } W \succeq \mathbf{0}, \mathbf{0}_n \leq \boldsymbol{\gamma} \leq C_\delta \mathbf{1}_n, \tag{2.25}$$

where C_δ is a regularization parameter to control the trade-off between the clustering error and the margin. After obtaining the optimal $\boldsymbol{\gamma}$, the partition can be inferred from the sign of the eigenvector of W associated with the zero eigenvalue, since the Karush-Kuhn-Tucker complementary condition is $W\boldsymbol{w} = \mathbf{0}_{n+1}$, and $\operatorname{sign}([\boldsymbol{w}]_i) = \operatorname{sign}([\boldsymbol{y}]_i)$ for $i = 1, \ldots, n$.

There exist a few faster MMC algorithms. *Iterative support vector regression* (IterSVR) (Zhang et al., 2007) replaces SVM with the hinge loss in the inner optimization subproblem with SVR with the Laplacian loss, while for each inner SVR the time complexity is at most $O(n^3)$ and the empirical time complexity is usually between $O(n)$ and $O(n^{2.3})$. *Cutting-plane maximum margin clustering* (CPMMC) (Zhao et al., 2008b) can be solved by a series of constrained concave-convex procedures within a linear time complexity $O(sn)$ where s is the average number of non-zero features. Unlike MMC and GMMC that rely on SDP or IterSVR and CPMMC that are non-convex, *label-generation maximum margin clustering* (LGMMC) (Li et al., 2009) is scalable yet convex so that it can achieve its globally optimal solution. Roughly speaking, LGMMC replaces the hinge loss in SVM with the squared hinge loss to get an alternative MMC:

$$\begin{aligned}\min_{\boldsymbol{y}\in\{-1,+1\}^n} \min_{\boldsymbol{w},\boldsymbol{\xi}} \quad & \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \rho + \frac{C}{2}\sum_{i=1}^{n}\xi_i^2 \\ \text{s.t.} \quad & y_i\boldsymbol{w}^\top x_i \geq \rho - \xi_i, \quad i = 1,\ldots,n \\ & -b \leq \boldsymbol{y}^\top \mathbf{1}_n \leq b.\end{aligned}$$

After a long derivation, LGMMC can be expressed as a *multiple kernel learning* problem:

$$\begin{aligned}\min_{\boldsymbol{\mu}\in\mathbb{R}^{2^n}} \max_{\boldsymbol{\alpha}} \quad & -\frac{1}{2}\boldsymbol{\alpha}^\top\left(\sum\nolimits_{t:-b\leq\boldsymbol{y}_t^\top\mathbf{1}_n\leq b}\mu_t K\circ\boldsymbol{y}_t\boldsymbol{y}_t^\top + \frac{1}{C}I_n\right)\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{\mu}^\top\mathbf{1}_{2^n} = 1, \boldsymbol{\mu}\geq\mathbf{0}_{2^n} \\ & \boldsymbol{\alpha}^\top\mathbf{1}_n = 1, \boldsymbol{\alpha}\geq\mathbf{0}_n.\end{aligned}$$

This optimization is again solved by the cutting plane method, that is, finding the most violated $\boldsymbol{y}_t$ iteratively, and the empirical time complexity of multiple kernel learning has the same order as the complexity of SVM which usually scales between $O(n)$ and $O(n^{2.3})$.

On the other hand, the stability of MVC is by no means inferior to those non-convex MMC in terms of the resulting clusters. The optimization involved in MVC-HL is a convex SDP problem; the optimization involved in MVC-SL is a non-convex SQP problem, while under mild conditions, it seems a convex one if we only care the resulting clusters. Moreover, MVC-SL possesses a data-

dependent error bound, and to the best of our knowledge no MMC has such a result. Although the time complexity of MVC-SL is $O(n^3)$, its computation time has exhibited less potential of growth in our experiments than the computationally-efficient LGMMC (see Figure 2.5 in page 71).

2.7.2 Spectral Clustering

Spectral clustering (SC) (Shi and Malik, 2000; Meila and Shi, 2001; Ng et al., 2002) is also closely related to MVC. SC algorithms include two steps, a spectral embedding step to unfold the manifold structure and embed the input data into a low-dimensional space in a geodesic manner, and then a k-means step to carry out the clustering using the embedded data.

Given a similarity matrix $W \in \mathbb{R}^{n\times n}$ and the degree matrix $D = \mathrm{diag}(W\mathbf{1}_n)$, we have three popular graph Laplacian matrices: The unnormalized graph Laplacian is defined as

$$L_{\mathrm{un}} := D - W,$$

and two normalized graph Laplacian are

$$L_{\mathrm{sym}} := D^{-1/2} L_{\mathrm{un}} D^{-1/2} = I_n - D^{-1/2} W D^{-1/2}$$
$$L_{\mathrm{rw}} := D^{-1} L_{\mathrm{un}} = I_n - D^{-1} W.$$

The first matrix is denoted by L_{sym} since it is a symmetric matrix and the second one by L_{rw} since it is closely related to a random walk. Each popular graph Laplacian corresponds to a popular SC algorithm according to von Luxburg (2007). Unnormalized SC computes the first k eigenvectors of L_{un} where the eigenvalues are all positive and listed in an increasing order. Shi and Malik (2000) computes the first k generalized eigenvectors of the generalized eigenvalue problem $L_{\mathrm{un}}\boldsymbol{u} = \lambda D\boldsymbol{u}$ that are also the eigenvectors of L_{rw}, and hence it is called normalized SC.[10] The other normalized SC algorithm, namely Ng et al. (2002), computes the first k eigenvectors of L_{sym}, puts them into an $n \times k$ matrix, and normalizes all rows of that matrix to the unit norm, that is, projects the embedded data further to the k-dimensional unit sphere. Anyway, the main idea is to change the representation from $\mathbb{R}^d$ to $\mathbb{R}^k$ and then run k-means clustering.

[10] Actually, two algorithms were proposed in Shi and Malik (2000): The two-way cut algorithm only makes use of the second eigenvector and the k-way cut algorithm uses all first k eigenvectors.

MVC-SL is able to integrate the two steps of unnormalized SC into a single optimization when the number of clusters is two and the highly non-convex k-means step is unnecessary. Furthermore, a vital difference between MVC and SC is that the basic model of MVC has a loss function which pushes hypotheses away from the coordinate axes and always leads to non-sparse optimal solutions. When considering the finite sample stability, the spectral embedding step of SC is stable if MVC-SL is stable but not vice versa, since SC only requires that the graph Laplacian has distinct eigenvalues; the k-means step is always unstable for fixed data due to the non-convex distortion function which is essentially an integer programming, but it is stable for different random samplings from the same underlying distribution, if the globally optimal solution is unique (Rakhlin and Caponnetto, 2007). In addition, there are a few theoretical results about the infinite sample stability or the consistency of SC. Globally optimal solutions to k-means clustering converge to a limit partition of the whole data space $\mathcal{X}$, if the underlying distribution has a finite support, and the globally optimal solution to the expectation of the distortion function with respect to the underlying distribution is unique (Ben-David et al., 2007). Eigenvectors of graph Laplacian also converge to eigenvectors of certain limit operators, while the conditions for convergence are very general for L_{sym}, but very special for L_{un} so that they are not easily satisfied (von Luxburg et al., 2005, 2008). In contrast, the infinite sample stability of MVC is currently an open problem.

Remark 2.22. Certain SC algorithms such as Belkin and Niyogi (2002) ignore the first eigenvector by extracting the second to k-th eigenvectors of some graph Laplacian, and thus change the representation to $\mathbb{R}^{k-1}$ rather than $\mathbb{R}^k$. Nevertheless, the multiplicity of the eigenvalue zero of the graph Laplacian equals the number of connected components of the similarity graph, and the eigenspace of eigenvalue zero is spanned by the indicator vectors of the connected components (von Luxburg, 2007, Propositions 2 and 4). As a consequence, all three aforementioned SC algorithms keep the first eigenvector in order to deal with disconnected similarity graphs.

2.7.3 Approximate Volume Regularization

The connection of *approximate volume regularization* (AVR) (El-Yaniv et al., 2008) and MVC is analogous with the connection of SVM and MMC.

Compared with MVC, AVR is a transductive method for classification so that the label vector $\boldsymbol{y}$ is constant and only the soft response vector $\boldsymbol{h}$ needs to be optimized. More specifically, given m labeled data $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ and u unlabeled data $\{x_{m+1}, \ldots, x_{m+u}\}$, the label vector is denoted by $\boldsymbol{y} = (y_1, \ldots, y_m, 0, \ldots, 0)^\top \in \mathbb{R}^{m+u}$, and the primal problem of AVR is defined as

$$\min_{\boldsymbol{h} \in \mathbb{R}^{m+u}} -\frac{1}{m}\boldsymbol{h}^\top \boldsymbol{y} + \gamma \boldsymbol{h}^\top Q \boldsymbol{h} \qquad \text{s.t. } \|\boldsymbol{h}\|_2 = t, \tag{2.26}$$

where t is a hyperparameter to control the scale of $\boldsymbol{h}$. Since $\boldsymbol{y}$ is constant, optimization (2.26) can be directly solved using Lagrangian multipliers and the Karush-Kuhn-Tucker conditions

$$\begin{aligned} -\boldsymbol{y}/m + 2\gamma Q\boldsymbol{h} - 2\eta\boldsymbol{h} &= 0, \\ \boldsymbol{h}^\top \boldsymbol{h} - t^2 &= 0. \end{aligned}$$

Let the eigen-decomposition of Q be $Q = V\Lambda V^\top$ and $d_i = [V^\top \boldsymbol{y}]_i$, then we get an equation about the optimal η:

$$\frac{1}{4m^2}\sum_{i=1}^{m+u} \frac{d_i^2}{(\gamma\lambda_i - \eta)^2} - t^2 = 0. \tag{2.27}$$

Thanks to the special structure of (2.27), a binary search procedure is enough for finding its smallest root η^*, and the optimal $\boldsymbol{h}$ is recovered by

$$\boldsymbol{h}^* = \frac{1}{2m}(\gamma Q - \eta^* I_{m+u})^{-1}\boldsymbol{y}.$$

On the other hand, MVC involves a combinatorial optimization similarly to the most clustering models and several semi-supervised learning models such as MMC. This difficulty caused by the integer feasible region is intrinsically owing to the clustering problem and has no business with the large volume approximation $V(\boldsymbol{h})$. In order to solve the basic model, we proposed two approximation schemes based on SQP and SDP that are more complicated than finding the smallest root of Eq. (2.27) as in AVR.

2.8 Experiments

In this section, we numerically evaluate the performance of the proposed MVC algorithms.

2.8.1 Setup

Seven clustering algorithms were included in our experiments:

- Kernel k-means clustering (KM; Zha et al., 2002),
- Normalized spectral clustering (NSC; Ng et al., 2002),
- Maximum margin clustering (MMC; Xu et al., 2005),
- Generalized MMC (GMMC; Valizadegan and Jin, 2007),
- Label-generation MMC (LGMMC; Li et al., 2009),
- Soft-label maximum volume clustering (MVC-SL),
- Hard-label maximum volume clustering (MVC-HL).

The *CVX* package (Grant and Boyd, 2011), which is a Matlab-based modeling system for disciplined convex programming, was utilized to solve the QP problem (2.5) for MVC-SL and the SDP problems (2.14), (2.23) and (2.25) for MVC-HL, MMC and GMMC.

Table 2.1 summarizes the specification of data sets in our experiments. We first evaluated all seven algorithms on three artificial data sets. MVC-HL and MMC were excluded from the middle-scale experiments since they were very time-consuming when $n > 100$. The *IDA benchmark repository*[11] contains thirteen benchmark data sets for binary classification, and ten of them that have no intrinsic within-class multi-modality were included. Additionally, we made intensive comparisons based on four well-known benchmark data sets for classification: *USPS* and *MNIST*[12] contain 8-bit gray-scale images of handwritten digits '0' through '9' with the resolution 16×16 and 28×28, *20Newsgroups sorted*

[11] http://ida.first.gmd.de/~raetsch/data/benchmarks.htm.

[12] http://cs.nyu.edu/~roweis/data.html.

	# Classes	# Features	# Data	# Samplings
		Artificial Data		
2gaussians	2	3	-	12×10
2moons	2	2	400	12×10
2circles	2	2	315	12×10
		IDA Benchmarks		
Breast-cancer	2	9	200	100
Diabetes	2	8	468	100
Flare-solar	2	9	666	100
German	2	20	700	100
Heart	2	13	170	100
Image	2	18	1300	20
Ringnorm	2	20	400	100
Splice	2	60	1000	20
Titanic	2	3	150	100
Twonorm	2	20	400	100
		Other Benchmarks		
USPS	10	256	11000	8×10
MNIST	10	784	70000	8×10
20Newsgroups	7	26214	18846	8×10
Isolet	26	617	7797	8×10

Table 2.1: Specification of artificial and benchmark data sets

by date[13] contains term-frequency vectors of documents that come from twenty newsgroups, and *Isolet*[14] contains acoustic features of isolated spoken letters from 'A' to 'Z'.

In our experiments, the performance was measured by the clustering error rate

$$\frac{1}{n} d_{\mathcal{H}}(\boldsymbol{y}, \boldsymbol{y}^*) = \frac{1}{2n} \min(\|\boldsymbol{y} + \boldsymbol{y}^*\|_1, \|\boldsymbol{y} - \boldsymbol{y}^*\|_1),$$

where $\boldsymbol{y}$ is the label vector returned by clustering algorithms and $\boldsymbol{y}^*$ is the ground truth label vector. The similarity measure was either the Gaussian similarity

$$W_{i,j} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\right)$$

with a hyperparameter σ, the cosine similarity

$$W_{i,j} = \begin{cases} \dfrac{\langle x_i, x_j \rangle}{\|x_i\|_2 \|x_j\|_2} & \text{if } x_i \sim_k x_j, \\ 0 & \text{otherwise,} \end{cases}$$

with a hyperparameter k, where $x_i \sim_k x_j$ means that x_i and x_j are among the k-nearest neighbors of each other, or the locally-scaled Gaussian-like similarity (Zelnik-Manor and Perona, 2005)

$$W_{i,j} = \exp\left(-\frac{\|x_i - x_j\|_2^2}{2\sigma_i \sigma_j}\right)$$

with a hyperparameter k, where $\sigma_i = \|x_i - x_i^{(k)}\|_2$ is the local scaling factor of x_i and $x_i^{(k)}$ is the k-th nearest neighbor of x_i in X_n. The kernel matrix was $K = W$ for KM, MMC and LGMMC, and $K = W + I_n/n$ for GMMC since it would be very unstable without this small eigenvalue shift. NSC relied on the graph Laplacian L_{sym} constructed from W. Due to the requirement of positive definiteness of Q for MVC, we also slightly shifted the eigenvalues of certain positive semi-definite matrices and adopted $Q = L_{\text{sym}} + I_n/n$ for MVC-SL and $Q = W + I_n/n$ for MVC-HL.

Numerical issues always exist and there may be more than one candidate $\boldsymbol{h}_0$ for MVC-SL. Let $\lambda_1 \leq \cdots \leq \lambda_n$ be the eigenvalues of Q, and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be the

[13] http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

[14] http://archive.ics.uci.edu/ml/datasets/isolet.

associated normalized eigenvectors. In our implementation, we initialize MVC-SL by a few eigenvectors whose eigenvalues are close to λ_2. Specifically, we construct a set of candidate eigenvectors $\mathcal{V} = \{\boldsymbol{v}_i \mid |\lambda_i - \lambda_2| < 10^{-4}\}$, and if $\#\mathcal{V} > 10$, we say that Q is ill-defined and only keep ten such $\boldsymbol{v}_i$ in $\mathcal{V}$. Next we obtain one $\boldsymbol{h}_0$ from each $\boldsymbol{v}_i \in \mathcal{V}$ and solve the SQP problem based on each $\boldsymbol{h}_0$. At last, the solution $\boldsymbol{h}^*$ resulting in the smallest objective value $-2\|\boldsymbol{h}^*\|_1 + \gamma \boldsymbol{h}^{*\top} Q \boldsymbol{h}^*$ would be selected as the final solution to MVC-SL. This trick can sometimes improve the performance significantly, while the cost is the increase of the computation time by no more than ten times.

2.8.2 Artificial Data Sets

To begin with, we compare the clustering error and the computation time of all seven algorithms based on three artificial data sets. As visualized in Figure 2.3, *2gaussians* is a three-dimensional data set generated as follows. We first randomly sampled $X^+_{n/2}$ from a Gaussian distribution with zero mean and covariance matrix $\mathrm{diag}(100, 4)$ and $X^-_{n/2}$ from the other Gaussian distribution with zero mean and covariance matrix $\mathrm{diag}(4, 100)$, set the third dimension as $+3$ for $X^+_{n/2}$ and -3 for $X^-_{n/2}$ and combined $X^+_{n/2}$ and $X^-_{n/2}$ into X_n. Subsequently, *2moons* is a two-dimensional data set with two non-Gaussian crescent-like clusters, and *2circles* is another two-dimensional data set with two non-Gaussian ring-like clusters. The Gaussian similarity was applied to all algorithms, and σ was fixed to $m_\sigma/10$, where m_σ is the mean pairwise distance given by

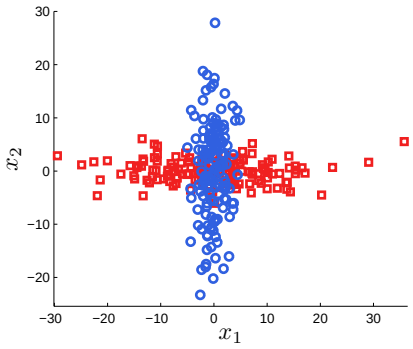
$$m_\sigma = \frac{\sum_{1 \le i < j \le n} \|x_i - x_j\|_2}{n(n-1)/2} = \frac{\sum_{i,j=1}^{n} \|x_i - x_j\|_2}{n(n-1)}, \tag{2.28}$$

since $\|x_i - x_j\|_2 = 0$ when $i = j$. The regularization parameter C of MMC was the best value among $\{10^{-3}, 1, 10^3\}$, that is, we ran MMC three times using $C = 10^{-3}, 1, 10^3$ and recorded the best performance, since there lacks a uniformly effective model selection framework for clustering algorithms. The regularization parameter C of LGMMC was also selected from $\{10^{-3}, 1, 10^3\}$ in the same way. For GMMC, the regularization parameter C_e was set to 10^4 following Valizadegan and Jin (2007) and the other regularization parameter C_δ was the best candidate in $\{10^{-3}, 1, 10^3\}$. We fixed the stopping threshold ϵ to 10^{-6}, the regularization

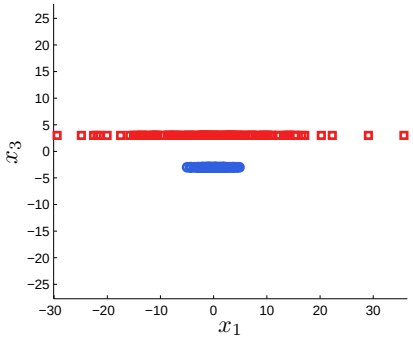
parameter γ to 10^{-2} and let the class balance parameter b adaptively be $1/n$ for MVC-SL, while for MVC-HL, we fixed C to 1 and tried $\gamma \in \{10^{-3}, 1, 10^3\}$.

The experimental results in terms of the means of the clustering error are reported in Figure 2.4. All of the results were obtained by repeatedly running an algorithm on 10 random samplings with given sample size n, and the sample sizes were $\{50, 60, \ldots, 100\}$ for the small-scale experiments and $\{50, 100, \ldots, 300\}$ for the middle-scale experiments. We can see that among the three data sets, 2gaussians is most difficult such that LGMMC still had a mean clustering error around twenty percents even when $n = 300$, and 2circles is easiest because MMC, MVC-SL and MVC-HL already got near zero errors when $n = 80$ and LGMMC, GMMC and NSC also achieved perfect partitions after $n = 150$. In contrast, KM cannot deal with these artificial data well due to the non-convex distortion function and the random initialization of cluster centers, even though it was equipped with the Gaussian similarity. Surprisingly, NSC was worse than KM on 2guassians, whereas MVC-SL based on the almost same input $Q = L_{\text{sym}} + I_n/n$ had much lower clustering errors, which implies that the highly non-convex k-means step may be a bottleneck of NSC.

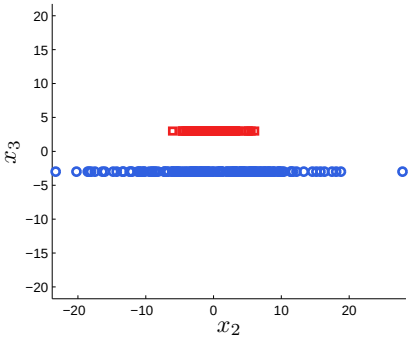
Next we report the corresponding computation time of these algorithms in Figure 2.5. All of the results were measured in average seconds per run on Xeon X5670 processors. Note that the worst case running time (i.e., the asymptotic time complexity) of KM is super-polynomial in the sample size n (Arthur and Vassilvitskii, 2006), and so is the worst case running time of NSC. On the other hand, the asymptotic time complexities of LGMMC, MVC-SL, GMMC, MMC and MVC-HL are $O(n^3)$, $O(n^3)$, $O(n^{4.5})$, $O(n^{6.5})$ and $O(n^{6.5})$ respectively. In our experiments, NSC was the most computationally-efficient algorithm and almost always faster than KM, since the k-means invoked by NSC after the spectral embedding converged in fewer iterations than KM. While LGMMC was consistently faster than GMMC, MVC-SL lay between them and was comparable with GMMC in the small-scale experiments and comparable with LGMMC in the middle-scale experiments. As a result, the computation time or empirical time complexity of MVC-SL exhibited less potential of growth than LGMMC and GMMC. The worst-case computational complexities of MVC-HL and MMC made them extremely time-consuming, poorly scalable to middle or large sample sizes,



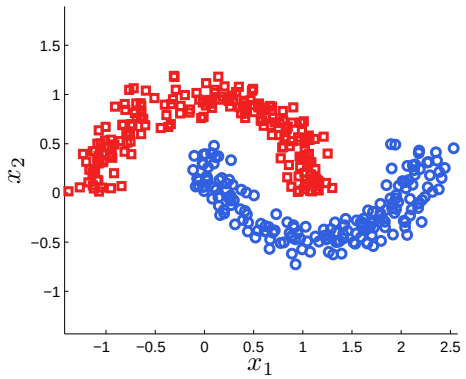
(a) 2gaussians, x_1 vs. x_2



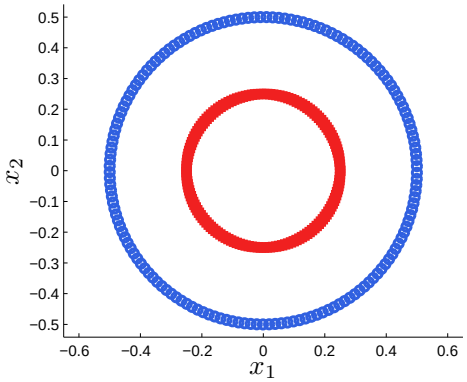
(b) 2gaussians, x_1 vs. x_3



(c) 2gaussians, x_2 vs. x_3

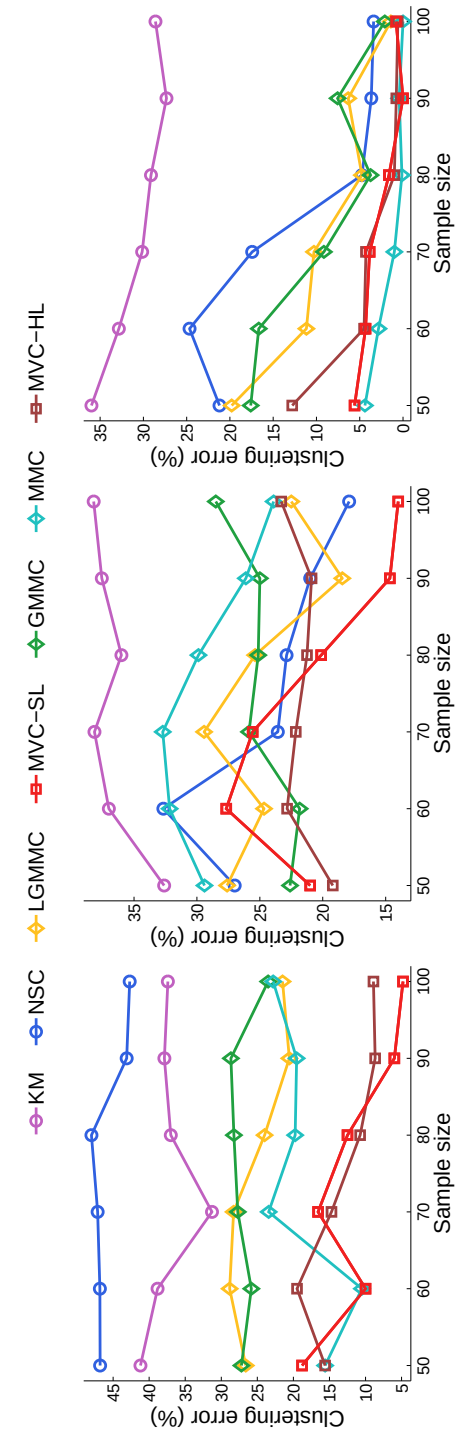


(d) 2moons



(e) 2circles

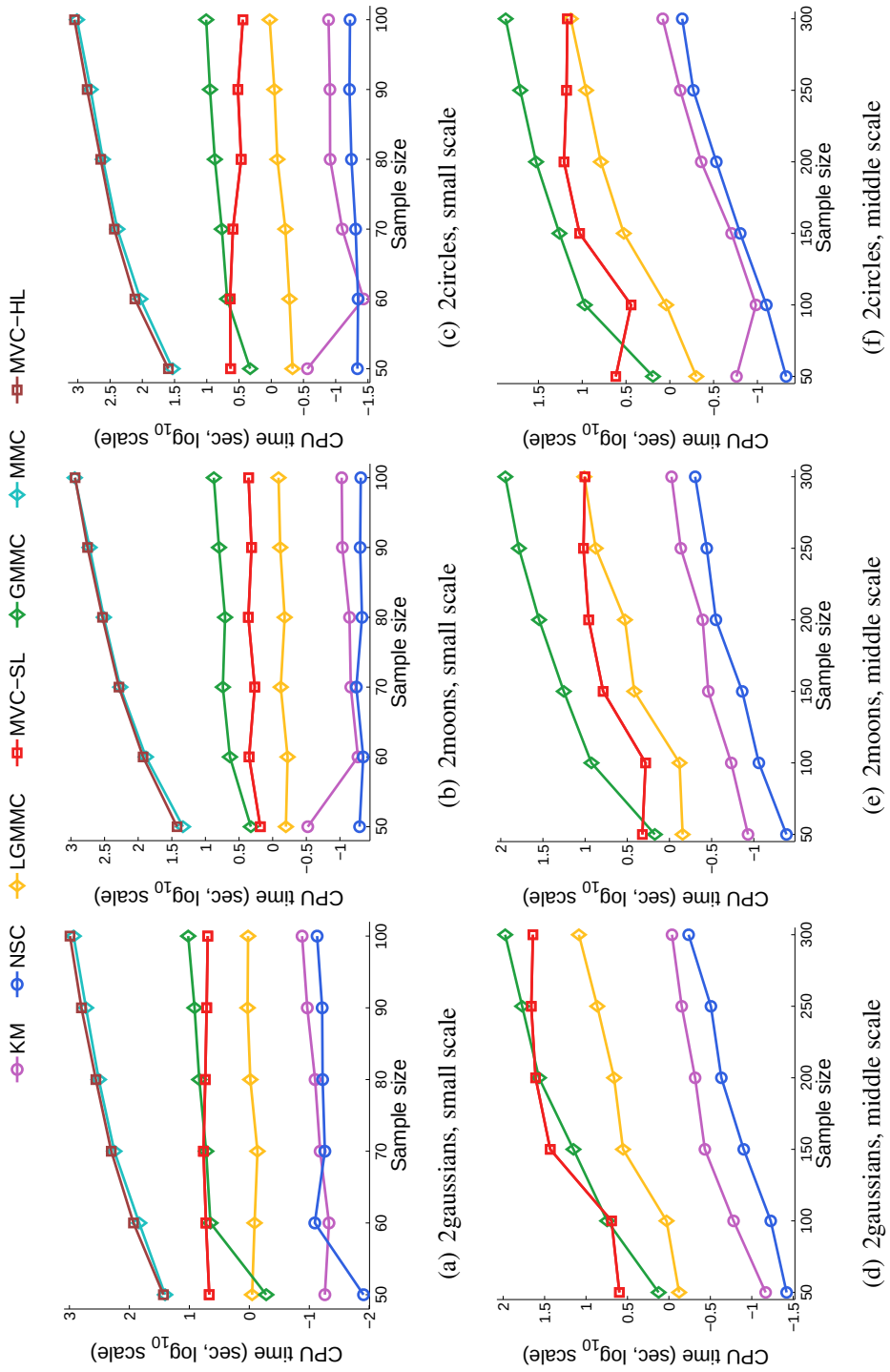
Figure 2.3: Visualization of artificial data sets



(a) 2gaussians, small scale (b) 2moons, small scale (c) 2circles, small scale

(d) 2gaussians, middle scale (e) 2moons, middle scale (f) 2circles, middle scale

Figure 2.4: Means of the clustering error (in %) on 2gaussians, 2moons and 2circles



(a) 2gaussians, small scale (b) 2moons, small scale (c) 2circles, small scale

(d) 2gaussians, middle scale (e) 2moons, middle scale (f) 2circles, middle scale

Figure 2.5: Means of the CPU time (in sec, per run) on 2gaussians, 2moons and 2circles

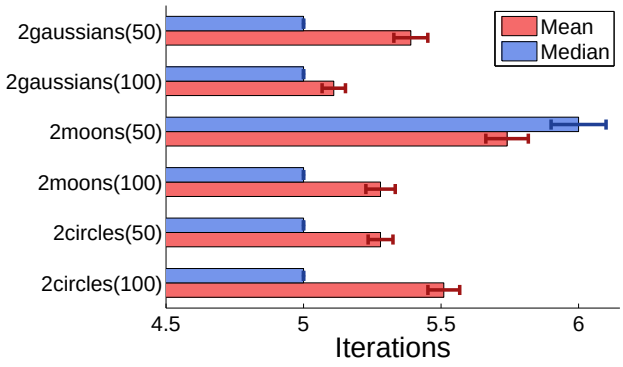
and hence impractical despite their low mean clustering errors on 2guassians and 2circles.

Furthermore, we investigate three important properties of MVC-SL, and report the results over 100 random samplings in Figure 2.6.

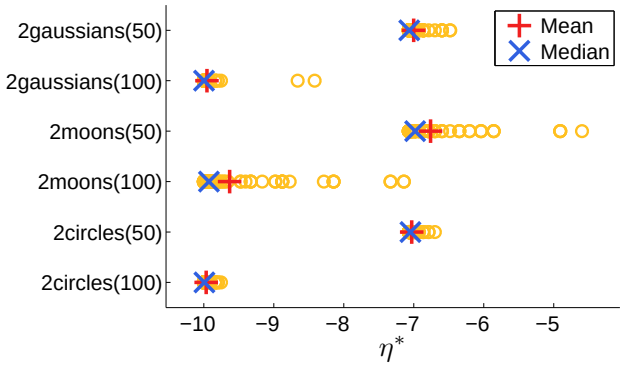
Firstly, panel (a) shows the mean and median values about the number of iterations required by MVC-SL, where each mean is shown with the *standard error*, and each median is shown with the *median absolute deviation* divided by the square root of the number of random samplings (i.e., 10). As mentioned before, the convergence rate of SQP iterations is independent of the sample size n, and we can see that MVC-SL usually stopped within just a few iterations in our experiments. This phenomenon implies that the empirical time complexity of MVC-SL is directly proportional to the internal QP solver.

Secondly, we examine the distribution of η^* which may influence the stability of the resulting clusters. Fortunately, panel (b) shows that η^* for fixed data set and fixed sample size were highly concentrated, and the mean and median values exhibited a strong correlation with the sample size as well as a weak correlation with the data set.

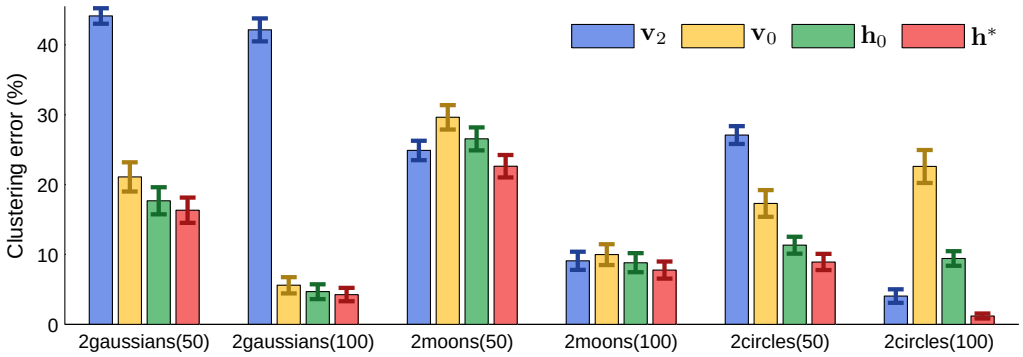
Thirdly, recall that there may be more than one candidate $\boldsymbol{h}_0$ and we initialize MVC-SL using $\mathcal{V} = \{\boldsymbol{v}_i \mid |\lambda_i - \lambda_2| < 10^{-4}\}$. Although all $\boldsymbol{v} \in \mathcal{V}$ appear nearly equally good to NSC, they could induce initial solutions of very different qualities for MVC-SL, as shown in panel (c). The vectors $\boldsymbol{v}_2$, $\boldsymbol{v}_0$, $\boldsymbol{h}_0$, and $\boldsymbol{h}^*$ are all treated as soft response vectors, and the means with standard errors of the clustering error are plotted in panel (c), where $\boldsymbol{v}_2$ is the eigenvector of Q and L_{sym} associated with λ_2, $\boldsymbol{v}_0$ is the eigenvector selected by MVC-SL, and $\boldsymbol{h}_0$ and $\boldsymbol{h}^*$ are the corresponding initial and final solutions. We can see that $\boldsymbol{h}^*$ was better than $\boldsymbol{h}_0$ and $\boldsymbol{h}_0$ was better than $\boldsymbol{v}_0$. Moreover, $\boldsymbol{v}_0$ was significantly superior to $\boldsymbol{v}_2$ on 2guassians. It is interesting and surprising that both $\boldsymbol{h}_0$ and $\boldsymbol{v}_0$ were significantly inferior to $\boldsymbol{v}_2$ on 2circles when $n = 100$, but they still resulted in $\boldsymbol{h}^*$ with the lowest mean clustering error. In a word, not only good initial solutions but also the SQP method contribute to the success of MVC-SL, and the underlying large volume principle is reasonable and useful for clustering.



(a) Number of iterations



(b) Distribution of η^*



(c) Clustering error (in %) of $\boldsymbol{v}_2$, $\boldsymbol{v}_0$, $\boldsymbol{h}_0$ and $\boldsymbol{h}^*$

Figure 2.6: Experimental results concerning three important properties of MVC-SL

2.8.3 Benchmark Data Sets

In the following, we discuss the experiments on the benchmarks listed in Table 2.1: The experiments involving ten IDA benchmarks are discussed in the first part, then USPS and MNIST in the second part, 20Newsgroups in the third part, and Isolet in the fourth part.

IDA Benchmarks

We compare KM, NSC, LGMMC, GMMC, and MVC-SL on ten data sets in the IDA benchmark repository that are designed for binary classification tasks and have one hundred fixed realizations for each data set except that the data sets Image and Splice only have twenty realizations. For each realization of each data set, we ignored the test data and tested five clustering algorithms using the training data, yet GMMC was not tested on the data sets Flare-solar, German, Image and Splice as it required a very long execution time when $n \geq 600$. The Gaussian similarity was applied and σ was the best value among $\{4m_\sigma, 2m_\sigma, m_\sigma, m_\sigma/2, m_\sigma/4\}$ for each realization and each algorithm, where the variable m_σ was the mean pairwise distance defined in Eq. (2.28). An exception is the data set Ringnorm where the locally-scaled similarity with $k = 7$ was applied, since it consists of data from two highly overlapped Gaussian distributions and can be treated as a multi-scale data set.[15] The settings for other hyperparameters of LGMMC, GMMC, and MVC-SL were exactly same as the experiments on the artificial data sets, specifically, $C \in \{10^{-3}, 1, 10^3\}$ for LGMMC, $C_e = 10^4$ and $C_\delta \in \{10^{-3}, 1, 10^3\}$ for GMMC, and $\epsilon = 10^{-6}$, $\gamma = 10^{-2}$ and $b = 1/n$ for MVC-SL.

Table 2.2 describes the means with standard errors of the clustering error rate by each algorithm on each data set. For the sake of comparison, Table 2.2 also lists the means of the classification error rate of highly-tuned SVM provided by the official web site of the IDA benchmark repository.

We could see from Table 2.2 that LGMMC and MVC-SL were either the best algorithm or comparable to the best algorithm based on the unpaired t-test at the

[15] In fact, Ringnorm violates the underlying assumption when evaluating clustering results using classification data sets, that is, the class structure and the cluster structure must coincide with each other. However, 2circles does not violate this assumption, since those ring-like clusters are neither Gaussian distributions nor overlapped clusters.

	KM	NSC	LGMMC	MVC-SL	GMMC	SVM
Breast-cancer	38.9 ± 0.65	26.4 ± 0.18	27.2 ± 0.19	$\mathbf{25.6 \pm 0.17}$	30.5 ± 0.23	26.0
Diabetes	30.3 ± 0.17	30.6 ± 0.18	$\mathbf{27.6 \pm 0.13}$	30.4 ± 0.15	28.6 ± 0.15	23.5
Flare-solar	$\mathbf{35.5 \pm 0.20}$	44.9 ± 0.11	37.6 ± 0.16	44.5 ± 0.12	N/A	32.4
German	39.4 ± 0.20	$\mathbf{30.2 \pm 0.09}$	$\mathbf{30.1 \pm 0.09}$	$\mathbf{30.2 \pm 0.09}$	N/A	23.6
Heart	$\mathbf{18.5 \pm 0.38}$	$\mathbf{18.0 \pm 0.23}$	18.7 ± 0.28	18.8 ± 0.22	18.9 ± 0.21	16.0
Image	41.0 ± 0.36	40.5 ± 0.15	$\mathbf{39.7 \pm 0.20}$	40.9 ± 0.11	N/A	2.96
Ringnorm	4.68 ± 0.11	$\mathbf{2.20 \pm 0.06}$	6.61 ± 0.11	$\mathbf{2.17 \pm 0.06}$	$\mathbf{2.07 \pm 0.06}$	1.66
Splice	29.1 ± 1.41	35.5 ± 0.44	$\mathbf{25.5 \pm 0.72}$	36.1 ± 0.44	N/A	10.9
Titanic	27.2 ± 0.59	26.8 ± 0.42	23.1 ± 0.36	$\mathbf{21.9 \pm 0.37}$	26.1 ± 0.43	22.4
Twonorm	3.61 ± 0.78	2.28 ± 0.07	$\mathbf{2.18 \pm 0.07}$	$\mathbf{2.20 \pm 0.07}$	$\mathbf{2.08 \pm 0.06}$	2.96

Table 2.2: Means with standard errors of the clustering error (in %) on IDA benchmark data sets. For each data set, the best algorithm and comparable ones based on the unpaired t-test at the significance level 5% are highlighted in boldface. Additionally, means of the classification error of highly-tuned SVM provided by IDA are also listed for comparison.

significance level 5% on five data sets. The clustering errors of five algorithms exhibited large differences on five data sets, namely, Breast-cancer, Flare-solar, German, Ringnorm and Splice, among which MVC-SL was one of the best algorithms on three data sets, and LGMMC was one of the best algorithms on two data sets. The clustering errors exhibited merely small differences on the other five data sets. Moreover, the fully supervised SVM has a mean classification error obviously smaller than the lowest mean clustering error on the data sets German, Image and Splice, and larger than the lowest mean clustering error on the data sets Breast-cancer, Titanic and Twonorm. It should not be surprising or confusing since the classification error is the out-of-sample test error on the test data whereas the clustering error is the in-sample test error on the same data to be clustered.

Images of Handwritten Digits

Secondly, we take the images of handwritten digits in USPS and MNIST. Instead of testing KM, NSC, LGMMC, GMMC and MVC-SL on all forty-five pairwise clustering tasks, a few challenging tasks were selected, namely, the pairs

$$\{1,7\},\{1,9\},\{8,9\},\{3,5\},\{3,8\},\{5,8\}$$

of USPS and

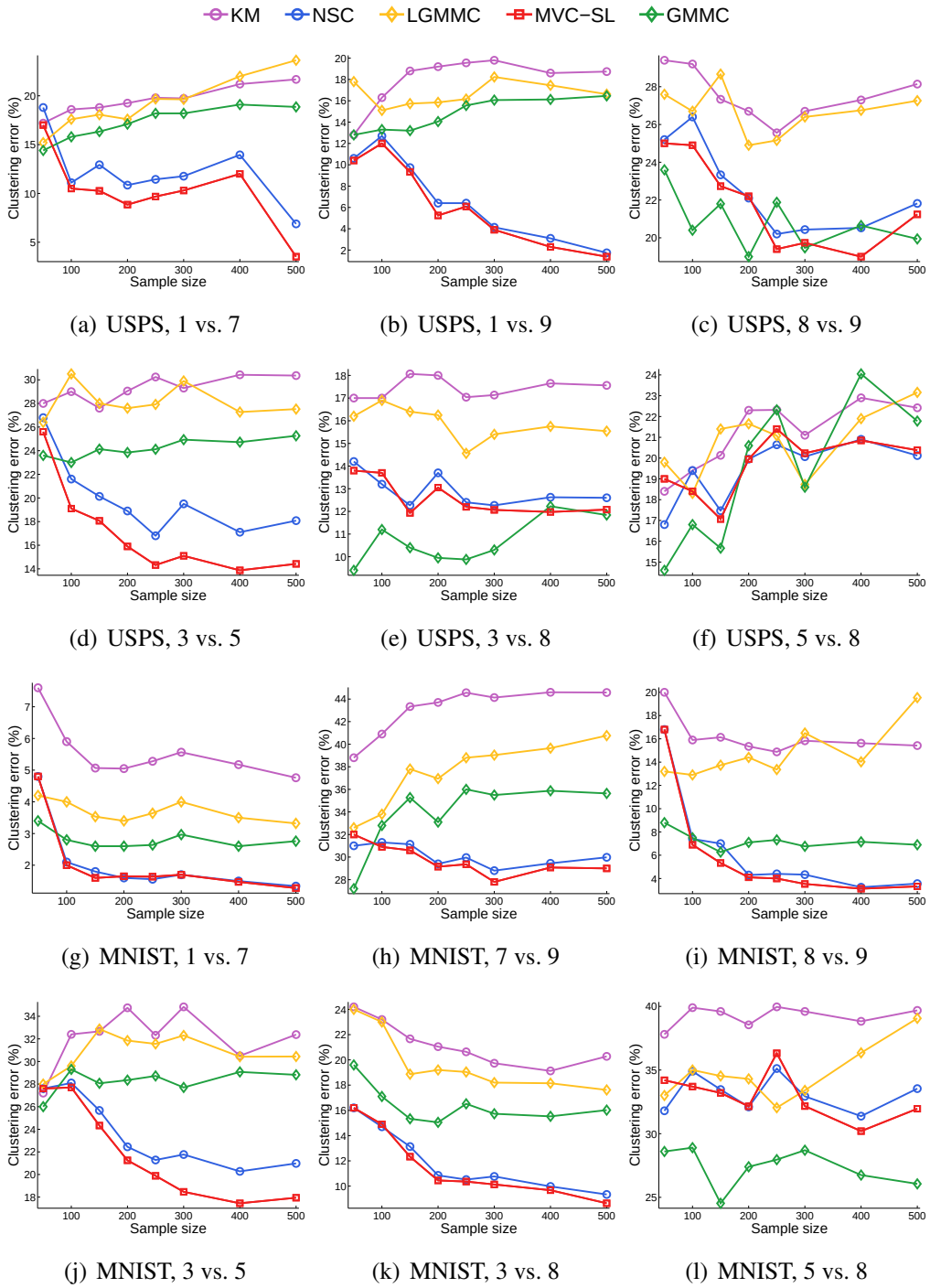
$$\{1,7\},\{7,9\},\{8,9\},\{3,5\},\{3,8\},\{5,8\}$$

of MNIST. The task digits 7 vs. 9 of USPS is too hard for all algorithms, so we selected an easier task digits 1 vs. 9. Unlike the training data in the IDA benchmark repository that are already standardized (i.e., normalized to mean zero and standard deviation one) by the provider, the 8-bit gray-scale images in USPS/MNIST are raw data represented by 256-/784-dimensional vectors of integers between 0 and 255. The popular pre-processing is to divide each integer by 255 and thus change the representation to vectors of floating-point numbers between 0 and 1. As a consequence, $\langle x_i, x_j \rangle$ is always nonnegative for any $1 \leq i, j \leq n$ and we can use the cosine similarity for NSC, where in our experiments the hyperparameter k of the k-nearest neighbors was the best value among $\{3,4,5,6,7,8\}$ for each random sampling. The same cosine similarity was also applied to MVC-SL. However, this cosine similarity did not work for the other three algorithms here, and then we still used the Gaussian similarity with σ as the best value among

$\{4m_\sigma, 2m_\sigma, m_\sigma, m_\sigma/2, m_\sigma/4\}$ for each random sampling, where m_σ was defined in Eq. (2.28). The settings for other hyperparameters of LGMMC, GMMC and MVC-SL were exactly same as the experiments on the artificial data sets.

Figure 2.7 reports the means of the clustering error by each algorithm on each task. The sample sizes were $\{50, 100, 150, 200, 250, 300, 400, 500\}$ for all tasks, and each mean value was obtained by repeatedly running an algorithm on 10 random samplings. Given a certain task with sample size n, we first merged all data of the two classes and then randomly sampled a subset of size n, so the classes in the resulting subset were not necessarily balanced when n was small. Moreover, Table 2.3 summarizes the means with standard errors of the clustering error, in which each algorithm has 80 random samplings on each task. Since the sample sizes here varied in a large range, we performed the paired t-test of the null hypothesis that the difference of the clustering error is from a Gaussian distribution with mean zero and unknown variance, against the alternative hypothesis that the mean is not zero.

We can see from Figure 2.7 that the easiest task is MNIST 1 vs. 7, such that the mean clustering errors of MVC-SL and NSC were less than two percents when $n \geq 100$, and the hardest tasks are MNIST 7 vs. 9 and 5 vs. 8, where no algorithm was better than twenty-five percents. Both Figure 2.7 and Table 2.3 show that the relatively easy tasks include the pairs $\{1, 7\}, \{1, 9\}, \{3, 8\}$ of USPS and $\{1, 7\}, \{8, 9\}, \{3, 8\}$ of MNIST, while the relatively hard tasks are the pairs $\{8, 9\}, \{3, 5\}, \{5, 8\}$ of USPS and $\{7, 9\}, \{3, 5\}, \{5, 8\}$ of MNIST. In addition, according to Figure 2.7, the mean clustering errors of MVC-SL were basically non-increasing except in panel (f) USPS 5 vs. 8, and MVC-SL, NSC and GMMC usually outperformed KM and LGMMC, as in Table 2.3. Similarly, MVC-SL was either the best algorithm or comparable to the best algorithm on ten out of twelve tasks according to Table 2.3, among which it was best on eight tasks and outperformed all others on seven tasks. The second best algorithm GMMC was best on four tasks, and then NSC was comparable on two tasks. In a word, MVC-SL was fairly promising on USPS and MNIST.



(a) USPS, 1 vs. 7 (b) USPS, 1 vs. 9 (c) USPS, 8 vs. 9

(d) USPS, 3 vs. 5 (e) USPS, 3 vs. 8 (f) USPS, 5 vs. 8

(g) MNIST, 1 vs. 7 (h) MNIST, 7 vs. 9 (i) MNIST, 8 vs. 9

(j) MNIST, 3 vs. 5 (k) MNIST, 3 vs. 8 (l) MNIST, 5 vs. 8

Figure 2.7: Means of the clustering error (in %) on USPS and MNIST

	KM	NSC	LGMMC	MVC-SL	GMMC
USPS, 1 vs. 7	19.5 ± 0.36	12.2 ± 0.91	19.2 ± 0.68	$\mathbf{10.3 \pm 0.89}$	17.3 ± 0.37
USPS, 1 vs. 9	18.0 ± 0.55	6.9 ± 0.85	16.6 ± 0.54	$\mathbf{6.3 \pm 0.77}$	14.7 ± 0.38
USPS, 8 vs. 9	27.5 ± 0.72	22.5 ± 0.89	26.7 ± 0.87	$\mathbf{21.8 \pm 0.93}$	$\mathbf{20.8 \pm 0.80}$
USPS, 3 vs. 5	29.2 ± 0.61	19.9 ± 0.97	28.1 ± 0.72	$\mathbf{17.0 \pm 0.96}$	24.2 ± 0.62
USPS, 3 vs. 8	17.4 ± 0.52	12.9 ± 0.46	15.9 ± 0.47	12.6 ± 0.47	$\mathbf{10.6 \pm 0.48}$
USPS, 5 vs. 8	21.1 ± 0.65	$\mathbf{19.4 \pm 0.59}$	20.8 ± 0.74	$\mathbf{19.7 \pm 0.67}$	$\mathbf{19.3 \pm 0.84}$
MNIST, 1 vs. 7	5.5 ± 0.36	$\mathbf{2.1 \pm 0.35}$	3.7 ± 0.21	$\mathbf{2.0 \pm 0.35}$	2.8 ± 0.19
MNIST, 7 vs. 9	43.1 ± 0.44	30.1 ± 0.59	37.4 ± 0.58	$\mathbf{29.7 \pm 0.62}$	33.9 ± 0.56
MNIST, 8 vs. 9	16.1 ± 0.96	6.4 ± 0.77	14.7 ± 0.80	$\mathbf{5.9 \pm 0.75}$	7.2 ± 0.36
MNIST, 3 vs. 5	32.1 ± 0.65	23.5 ± 0.66	30.9 ± 0.52	$\mathbf{21.8 \pm 0.72}$	28.3 ± 0.47
MNIST, 3 vs. 8	21.2 ± 0.49	11.9 ± 0.54	19.8 ± 0.58	$\mathbf{11.6 \pm 0.59}$	16.4 ± 0.54
MNIST, 5 vs. 8	39.2 ± 0.47	33.2 ± 1.17	34.7 ± 0.79	33.0 ± 1.22	$\mathbf{27.4 \pm 0.80}$

Table 2.3: Means with standard errors of the clustering error (in %) on USPS and MNIST. For each task, the best algorithm and comparable ones based on the paired t-test at the significance level 5% are highlighted in boldface.

Newsgroup Documents

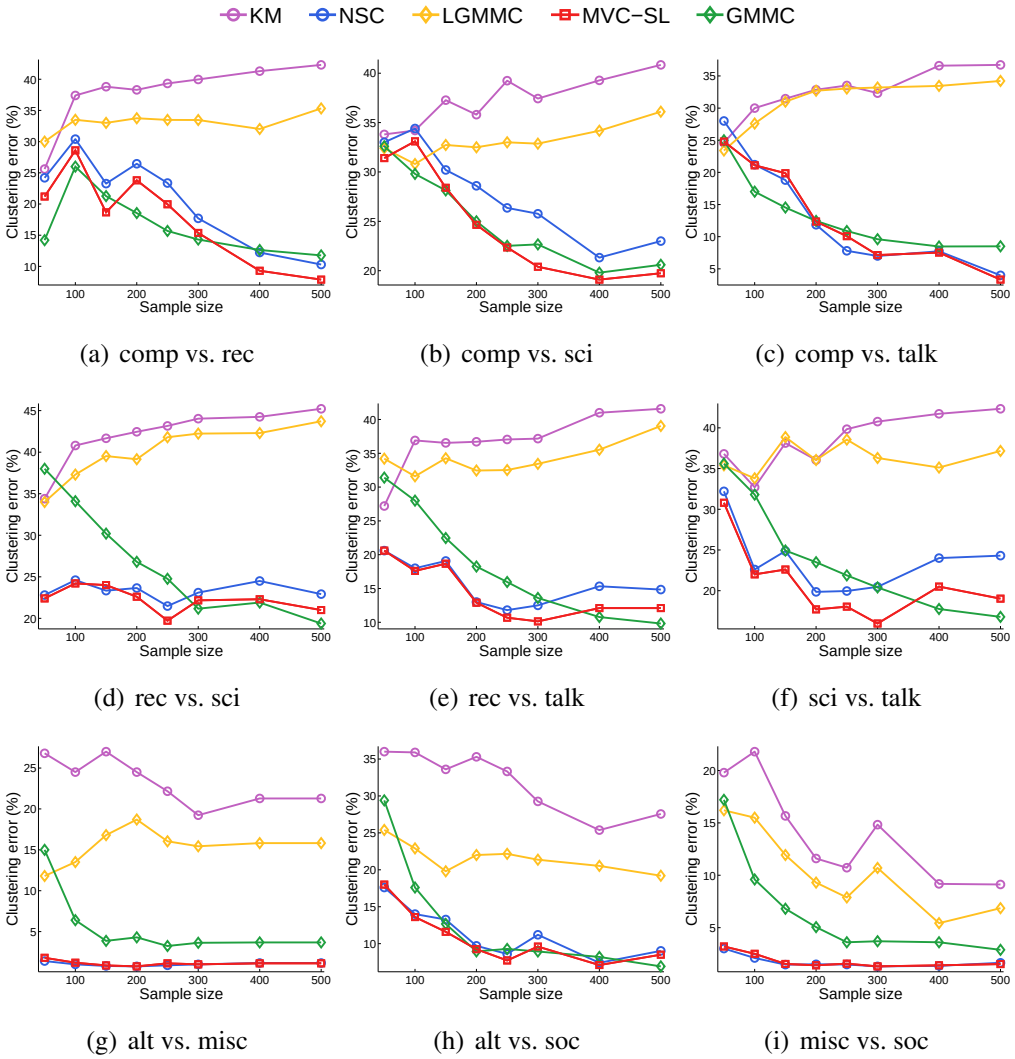
The benchmark 20Newsgroups has three versions containing 19997, 18846, and 18828 newsgroup documents, partitioned nearly evenly across twenty different newsgroups. The second version with 18846 documents is recommended by the original provider[16] and hence is used in our experiments. The documents in 20Newsgroups can be further grouped into seven topics: They are 'alt', 'comp', 'misc', 'rec', 'sci', 'soc' and 'talk', with 799, 4891, 975, 3979, 3952, 997 and 3253 documents respectively, where comp consists of five classes, each of rec, sci and talk consists of four classes, and each of alt, misc and soc consists of a single class. We prepared nine pairwise clustering tasks which included all tasks between the four multi-modal topics and all tasks between the three uni-modal topics. The term-frequency vectors were processed into term-frequency-inverse-document-frequency vectors using the script written by the provider[17] for the whole data set. We tried all of the three similarity measures, and found that for any algorithm no one was consistently better than the other two. However, the locally-scaled similarity generally fitted all five algorithms, where the hyperparameter k was the best value in $\{3, 4, 5, 6, 7, 8\}$ for each random sampling. The settings for other hyperparameters of LGMMC, GMMC and MVC-SL were exactly same as the experiments on the artificial data sets.

Figure 2.8 reports the means of the clustering error by each algorithm on each task. The sample sizes were $\{50, 100, 150, 200, 250, 300, 400, 500\}$ for all tasks, and each mean value was averaged over 10 random samplings. Similarly to the random samplings of USPS and MNIST, the classes in each random sampling here were not necessarily balanced when n was small. In addition, Table 2.4 summarizes the means with standard errors of the clustering error, in which each algorithm has 80 random samplings on each task. The paired t-test was performed due to the varied sample sizes.

We can see from Figure 2.8 and Table 2.4 that the tasks between the four multi-modal topics are more difficult than the tasks between the three uni-modal topics. Two tasks involving misc (i.e., alt vs. misc and misc vs. soc) are easiest, and three tasks involving sci (i.e., comp vs. sci, rec vs. sci, and sci vs. talk) are

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

[17] http://www.cad.zju.edu.cn/home/dengcai/Data/code/tfidf.m.



(a) comp vs. rec (b) comp vs. sci (c) comp vs. talk

(d) rec vs. sci (e) rec vs. talk (f) sci vs. talk

(g) alt vs. misc (h) alt vs. soc (i) misc vs. soc

Figure 2.8: Means of the clustering error (in %) on 20Newsgroups

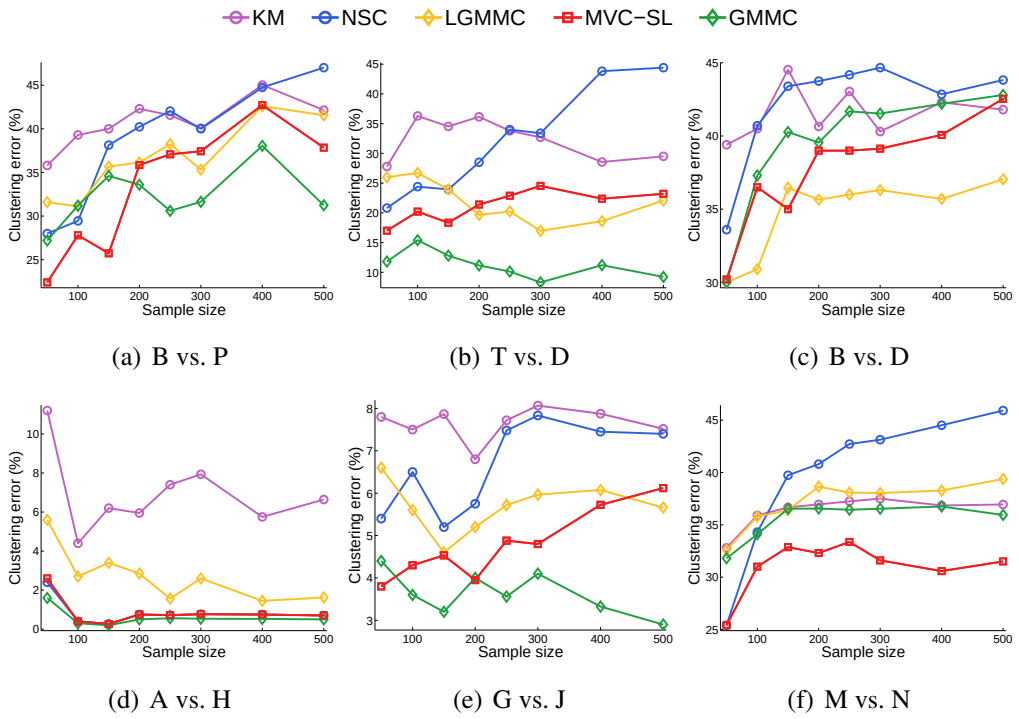
	KM	NSC	LGMMC	MVC-SL	GMMC
comp vs. rec	37.9 ± 0.77	21.0 ± 1.46	33.1 ± 0.57	$\mathbf{18.1 \pm 1.41}$	$\mathbf{16.8 \pm 0.74}$
comp vs. sci	37.2 ± 0.65	27.8 ± 1.20	33.1 ± 0.61	$\mathbf{24.9 \pm 1.17}$	$\mathbf{25.1 \pm 0.69}$
comp vs. talk	32.3 ± 0.93	$\mathbf{13.3 \pm 1.69}$	31.1 ± 0.73	$\mathbf{13.3 \pm 1.65}$	$\mathbf{13.3 \pm 0.80}$
rec vs. sci	42.0 ± 0.55	23.3 ± 0.84	40.0 ± 0.73	$\mathbf{22.3 \pm 0.85}$	27.0 ± 1.01
rec vs. talk	36.8 ± 0.76	15.6 ± 1.11	34.1 ± 1.02	$\mathbf{14.3 \pm 1.08}$	18.8 ± 1.08
sci vs. talk	38.5 ± 0.71	23.5 ± 1.01	36.4 ± 0.67	$\mathbf{20.8 \pm 0.97}$	24.1 ± 0.86
alt vs. misc	23.3 ± 1.85	$\mathbf{1.0 \pm 0.12}$	15.5 ± 1.07	1.1 ± 0.13	5.5 ± 0.60
alt vs. soc	32.0 ± 1.05	11.3 ± 1.01	21.7 ± 0.95	$\mathbf{10.7 \pm 0.85}$	12.7 ± 0.91
misc vs. soc	14.1 ± 1.32	$\mathbf{1.7 \pm 0.16}$	10.5 ± 0.67	$\mathbf{1.8 \pm 0.16}$	6.6 ± 0.60

Table 2.4: Means with standard errors of the clustering error (in %) on 20Newsgroups. For each task, the best algorithm and comparable ones based on the paired t-test at the significance level 5% are highlighted in boldface.

hardest. Moreover, MVC-SL, NSC and GMMC usually outperformed KM and LGMMC, and Figure 2.8 also illustrates that the mean clustering errors of MVC-SL were basically non-increasing. As shown in Table 2.4, MVC-SL was either the best algorithm or comparable to the best algorithm on eight out of nine tasks, among which it was best on six tasks and outperformed all others on four tasks. The second best algorithm NSC was best on three tasks, and then GMMC was best on two tasks and comparable on one task. In a word, MVC-SL was also fairly promising on 20Newsgroups.

Isolated Spoken Letters

The final benchmark is Isolet from the *UCI machine learning repository*. The data were collected by letting 150 subjects speak the name of each letter of the alphabet twice, while two 'F' and one 'M' were dropped due to difficulties in recording. Unlike the features of the previous benchmarks USPS, MNIST and 20Newsgroups, the acoustic features of Isolet are extracted by different ways and possess different physical meanings, including spectral coefficients, contour features, sonorant features, pre-sonorant features and post-sonorant features. All features are real-valued and scaled into the range -1 to $+1$. Generally speaking, all five algorithms can easily deal with the majority of pairwise clustering tasks, if we randomly choose two letters. Therefore, similarly to USPS and MNIST, a few challenging tasks that might sometimes be difficult for the mankind were selected: The letters B vs. P, T vs. D, B vs. D, A vs. H, G vs. J, and M vs. N. The hyperparameters here were slightly different from the previous experiments for better performance. The cosine similarity was applied to NSC, and the hyperparameter k was the best value in $\{1, 2, 3, 4, 5, 6\}$ for each random sampling. The Gaussian similarity was still used for KM, LGMMC and GMMC, and the hyperparameter σ was the best value in $\{2m_\sigma, m_\sigma, m_\sigma/2, m_\sigma/4, m_\sigma/8\}$ for each random sampling, where m_σ was defined in Eq. (2.28). For MVC-SL, we adopted either $Q = L_{\mathrm{sym}} + I_n/n$ where L_{sym} was constructed from the cosine similarity or $Q = nI_n - W$ with the Gaussian similarity depending on the task and the sample size n, and the hyperparameter k or σ was chosen in the same way. A key observation here was that for certain tasks such as M vs. N, the former specification was preferable for small n, whereas the latter specification was more advisable



(a) B vs. P (b) T vs. D (c) B vs. D

(d) A vs. H (e) G vs. J (f) M vs. N

Figure 2.9: Means of the clustering error (in %) on Isolet

for relatively large n. The settings for other hyperparameters were exactly same as the experiments on the artificial data sets.

Figure 2.9 reports the means of the clustering error by each algorithm on each task. The sample sizes were $\{50, 100, 150, 200, 250, 300, 400, 500\}$ for all tasks, and each mean value was averaged over 10 random samplings. Similarly to the random samplings of USPS and MNIST, the classes in each random sampling here were not necessarily balanced when n was small. In addition, Table 2.5 summarizes the means with standard errors of the clustering error, in which each algorithm has 80 random samplings on each task. The paired t-test was performed due to the varied sample sizes.

We can see from Figure 2.9 and Table 2.5 that the tasks A vs. H and G vs. J are very easy, and the tasks B vs. P, B vs. D and M vs. N are very hard. Interestingly, T vs. D is much easier than B vs. P and B vs. D, such that the lowest mean clustering errors on B vs. P and B vs. D were almost three times larger than the lowest mean clustering error on T vs. D. Unlike the curves shown in Figures 2.7 and 2.8, the mean clustering errors of MVC-SL in Figure 2.9 were

	KM	NSC	LGMMC	MVC-SL	GMMC
B vs. P	40.8 ± 0.64	38.7 ± 1.04	36.5 ± 0.86	$\mathbf{33.4 \pm 1.25}$	$\mathbf{32.3 \pm 1.30}$
T vs. D	32.4 ± 0.93	31.7 ± 1.48	21.8 ± 1.24	21.2 ± 1.02	$\mathbf{11.2 \pm 1.09}$
B vs. D	41.6 ± 0.55	42.1 ± 0.63	$\mathbf{34.8 \pm 0.73}$	37.7 ± 0.82	39.4 ± 0.73
A vs. H	6.9 ± 0.68	0.8 ± 0.19	2.7 ± 0.41	0.9 ± 0.21	$\mathbf{0.6 \pm 0.15}$
G vs. J	7.6 ± 0.32	6.6 ± 0.72	5.7 ± 0.28	4.8 ± 0.28	$\mathbf{3.6 \pm 0.22}$
M vs. N	36.4 ± 0.49	39.6 ± 0.87	37.2 ± 0.47	$\mathbf{31.1 \pm 0.64}$	35.6 ± 0.47

Table 2.5: Means with standard errors of the clustering error (in %) on Isolet. For each task, the best algorithm and comparable ones based on the paired t-test at the significance level 5% are highlighted in boldface.

basically non-increasing only in panel (d) A vs. H. Furthermore, LGMMC instead of NSC became a competitive algorithm besides GMMC and MVC-SL in Table 2.5, unlike the performance in Tables 2.3 and 2.4. According to Table 2.5, GMMC was the best algorithm on four tasks, MVC-SL was best on one task and also comparable to the best algorithm on one task, and LGMMC was best on one task. Nevertheless, MVC-SL was still satisfying on Isolet, if considering that MVC-SL consumed less than five percents of the total computation time while GMMC consumed over ninety percents, and thus GMMC was remarkably less computationally-efficient than MVC-SL.

2.9 Proofs of Theoretical Results

2.9.1 Proof of Lemma 2.9

If $\exists j \in \{1, \ldots, n\}$, $\boldsymbol{e}_j$ or $-\boldsymbol{e}_j$ is an eigenvector of Q, there should exist an eigenvalue $\lambda > 0$ such that $Q\boldsymbol{e}_j = \lambda \boldsymbol{e}_j$. This equation means that $Q_{j,j} = \lambda$ and $\forall i \neq j, Q_{i,j} = 0$. In other words, x_j is isolated and X_n is reducible. □

2.9.2 Proof of Theorem 2.10

If x_i is isolated in X_n, let $\delta_1 = \cdots = \delta_n = 1$, $\mathcal{K} = \{i\}$ and by definition X_n is SI-symmetric.

If X_n is axisymmetric under a permutation ϕ, without loss of generality, we assume $\phi(1) = 2$ and let $\delta_1 = -1, \delta_2 = \cdots = \delta_n = 1$ and $\mathcal{K} = \{1, 2\}$. Then X_n is SI-symmetric by Eq. (2.21),

$$\left(\sum\nolimits_{k\in\mathcal{K}} \delta_k \boldsymbol{e}_k\right)^\top Q \left(\sum\nolimits_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k\right) = \sum_{i=3}^{n} (Q_{2,i} - Q_{1,i}) = 0,$$

since $\forall i \in \{3, \ldots, n\}$, $\phi(i) \notin \{1, 2\}$ and $Q_{1,i} = Q_{2,\phi(i)}$. □

2.9.3 Proof of Theorem 2.11

When $n = 2$, X_2 must be axisymmetric if $Q_{1,1} = Q_{2,2}$, and we know that X_2 is SI-symmetric by Theorem 2.10.

When $n > 2$, assume that X_n is irreducible due to Theorem 2.10, and then $\mathbb{R}^n$ has two disjoint bases: the standard basis and the set of the principle axes of $\mathcal{E}(\mathcal{H}_Q)$ according to Lemma 2.9. We present an indirect proof of the theorem as follows.

Step 1. Let $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of Q and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be the associated normalized eigenvectors. Suppose that $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the directions of two principal axes of $\mathcal{E}(\mathcal{H}_Q)$ with the same length $1/\sqrt{\lambda_i} = 1/\sqrt{\lambda_j}$. There should be at least one principal axis $\boldsymbol{v}_l$ such that $l \notin \{i, j\}$, $\lambda_l \neq \lambda_j$ and $\mathcal{E}(\mathcal{H}_Q)$ is rotational about $\boldsymbol{v}_l$ along the circle

$$C(\boldsymbol{v}_i, \boldsymbol{v}_j) := \{\cos(\theta)\boldsymbol{v}_i + \sin(\theta)\boldsymbol{v}_j \mid \theta \in [0, 2\pi)\}.$$

Otherwise, all principal axes have the same length and $\mathcal{E}(\mathcal{H}_Q)$ is a perfect ball, which contradicts the fact that $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n$ are not eigenvectors of Q.

Further suppose that $\lambda_k \neq \lambda_l$ for any $k \neq l$, that is, the principal axis with the direction $\boldsymbol{v}_l$ has a unique length. As a consequence, $\boldsymbol{v}_l$ has a fixed position and cannot rotate within $C(\boldsymbol{v}_k, \boldsymbol{v}_l)$ for any $k \notin \{i, j, l\}$. Otherwise, all vectors in $C(\boldsymbol{v}_k, \boldsymbol{v}_l)$ are legal principal axes and can be considered as $\boldsymbol{v}_l$ with a fixed position.

We know that $\mathcal{E}(\mathcal{H}_Q)$ intersects the k-th coordinate axis at $\pm\boldsymbol{e}_k/\sqrt{\kappa}$ from $Q_{k,k} = \kappa$, and the intersections compose an $(n-1)$-dimensional hyperplane. Principal axes of $\mathcal{E}(\mathcal{H}_Q)$ are orthogonal and have at most $(n-1)$ distinct lengths, and $\mathcal{E}(\mathcal{H}_Q)$ also has a set of n orthogonal axes with the same length $1/\sqrt{\kappa}$, i.e., the standard basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ extended by $1/\sqrt{\kappa}$. Hence, any principal axis in a fixed position especially $\boldsymbol{v}_l$ should lie on the central direction of some quadrant with the dimensionality at least two. In other words, $\boldsymbol{v}_l$ can be written in the form of

$$\boldsymbol{v}_l = \frac{1}{\sqrt{\sum_{k=1}^n \delta_k^2}} \sum_{k=1}^n \delta_k \boldsymbol{e}_k, \quad \delta_k \in \{-1, 0, 1\},$$

where $\delta_1, \ldots, \delta_n$ cannot be all zeros.

Step 2. Let $\mathcal{K} = \{k \mid \delta_k = 0\}$ and one has $0 \leq \#\mathcal{K} < n$ where $\#$ measures the cardinality. We discuss the cases $\#\mathcal{K} > 0$ and $\#\mathcal{K} = 0$ separately.

If $\#\mathcal{K} > 0$, we reset $\delta_k = 1$ for $k \in \mathcal{K}$. Subsequently,

$$\begin{aligned}
\left(\sum\nolimits_{k\in\mathcal{K}} \delta_k \boldsymbol{e}_k\right)^\top Q \left(\sum\nolimits_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k\right) &= \left(\sum\nolimits_{k\in\mathcal{K}} \boldsymbol{e}_k\right)^\top Q \left(\sqrt{n-\#\mathcal{K}}\boldsymbol{v}_l\right) \\
&= \left(\sum\nolimits_{k\in\mathcal{K}} \boldsymbol{e}_k\right)^\top \sqrt{n-\#\mathcal{K}}(Q\boldsymbol{v}_l) \\
&= \left(\sum\nolimits_{k\in\mathcal{K}} \boldsymbol{e}_k\right)^\top \sqrt{n-\#\mathcal{K}}(\lambda_l\boldsymbol{v}_l) \\
&= \lambda_l \left(\sum\nolimits_{k\in\mathcal{K}} \boldsymbol{e}_k\right)^\top \left(\sqrt{n-\#\mathcal{K}}\boldsymbol{v}_l\right) \\
&= \lambda_l \left(\sum\nolimits_{k\in\mathcal{K}} \boldsymbol{e}_k\right)^\top \left(\sum\nolimits_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k\right) \\
&= \lambda_l \sum_{k\in\mathcal{K},k'\notin\mathcal{K}} \delta_{k'} \boldsymbol{e}_k^\top \boldsymbol{e}_{k'} \\
&= 0,
\end{aligned}$$

due to $Q\boldsymbol{v}_l = \lambda_l \boldsymbol{v}_l$ and the orthonormal condition of the basis $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$. If $\#\mathcal{K} = 0$, without loss of generality, assume that $\delta_1 = -\delta_2 = 1$ since $n > 2$ and the sign of $\boldsymbol{v}_l$ is arbitrary. The first two rows of the eigenvalue equation $Q\boldsymbol{v}_l = \lambda_l \boldsymbol{v}_l$ tell us

$$\begin{cases} \kappa - Q_{1,2} + \sum_{k=3}^{n} \delta_k Q_{1,k} = \lambda_l \\ Q_{2,1} - \kappa - \sum_{k=3}^{n} \delta_k Q_{2,k} = -\lambda_l \end{cases} \quad \Rightarrow \quad \sum_{k=3}^{n} \delta_k (Q_{1,k} - Q_{2,k}) = 0.$$

Hence by resetting $\mathcal{K} = \{1, 2\}$, we obtain

$$\left(\sum\nolimits_{k\in\mathcal{K}} \delta_k \boldsymbol{e}_k\right)^\top Q \left(\sum\nolimits_{k\notin\mathcal{K}} \delta_k \boldsymbol{e}_k\right) = \sum_{k=3}^{n} \delta_k (Q_{1,k} - Q_{2,k}) = 0.$$

Both cases lead to a contradiction since X_n is SI-asymmetric.

Therefore, all principal axes of $\mathcal{E}(\mathcal{H}_Q)$ have distinct lengths, which is exactly what we were to prove. □

2.9.4 Proof of Theorem 2.12

Let us denote $\boldsymbol{h}^* = (h_1, \ldots, h_n)^\top$ and consider $\boldsymbol{h}^\star = (h_{\phi(1)}, \ldots, h_{\phi(n)})^\top$.

Obviously, $\|\boldsymbol{h}^\star\|_1 = \|\boldsymbol{h}^*\|_1$ and $\|\boldsymbol{h}^\star\|_2 = \|\boldsymbol{h}^*\|_2$. Moreover,

$$\sum_{i,j=1}^{n} Q_{i,j} h_{\phi(i)} h_{\phi(j)} = \sum_{i,j=1}^{n} Q_{\phi(i),\phi(j)} h_{\phi(i)} h_{\phi(j)} = \sum_{k,l=1}^{n} Q_{k,l} h_k h_l,$$

because of the third property of Definition 2.7. Hence, $\boldsymbol{h}^{\star\top} Q \boldsymbol{h}^\star = \boldsymbol{h}^{*\top} Q \boldsymbol{h}^*$ and then $G(\boldsymbol{h}^\star) = G(\boldsymbol{h}^*)$.

Similarly, $\forall i \in \{1, \ldots, n\}$,

$$\sum_{j=1}^{n} Q_{i,j} h_{\phi(j)} = \sum_{j=1}^{n} Q_{\phi(i),\phi(j)} h_{\phi(j)} = \sum_{k=1}^{n} Q_{\phi(i),k} h_k,$$

where we use the third property of Definition 2.7 again. As a result,

$$\begin{aligned}
[g(\boldsymbol{h}^\star)]_i &= \gamma \sum_{j=1}^{n} Q_{i,j} h_{\phi(j)} - \eta h_{\phi(i)} - \operatorname{sign}(h_{\phi(i)}) \\
&= \gamma \sum_{k=1}^{n} Q_{\phi(i),k} h_k - \eta h_{\phi(i)} - \operatorname{sign}(h_{\phi(i)}) \\
&= [g(\boldsymbol{h}^*)]_{\phi(i)}.
\end{aligned}$$

Hence, $g(\boldsymbol{h}^\star) = \mathbf{0}_n$ according to the Karush-Kuhn-Tucker condition $g(\boldsymbol{h}^*) = \mathbf{0}_n$, which indicates that $\boldsymbol{h}^\star$ is also a minimum of optimization (2.4), since the Hessian matrix $\nabla^2 G(\boldsymbol{h}) = 2(\gamma Q - \eta I_n)$ must be symmetric and positive-definite.

Notice that $d_{\mathcal{H}}(\boldsymbol{h}^*, \boldsymbol{h}^\star) \geq 1$ due to the condition that $\exists i, h_{\phi(i)} h_i < 0$, with the only exception $d_{\mathcal{H}}(\boldsymbol{h}^*, \boldsymbol{h}^\star) = 0$ when $\operatorname{sign}(\boldsymbol{h}^\star) = -\operatorname{sign}(\boldsymbol{h}^*)$, i.e., $\forall i, h_{\phi(i)} h_i < 0$. This completes the proof. □

2.9.5 Proof of Theorem 2.13

We prove the theorem in three steps.

Step 1. Let $0 < \lambda_1 < \cdots < \lambda_n$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n$ be the eigenvalues and eigenvectors of Q. Given a minimum $\boldsymbol{h}$, the Karush-Kuhn-Tucker condition $g(\boldsymbol{h}) = \mathbf{0}$ implies that

$$\boldsymbol{h} = \hat{Q} \boldsymbol{y}, \tag{2.29}$$

where $\boldsymbol{y} = \text{sign}(\boldsymbol{h})$, $\hat{Q} = (\gamma Q - \eta I_n)^{-1}$, and the unknown η satisfies $\eta < \gamma\lambda_1$. Plug Eq. (2.29) into the constraint $\|\boldsymbol{h}\|_2 = 1$, note that $\hat{Q}$ is a symmetric matrix, and then we will have

$$\boldsymbol{y}^\top \hat{Q}^2 \boldsymbol{y} = (\hat{Q}\boldsymbol{y})^\top (\hat{Q}\boldsymbol{y}) = \boldsymbol{h}^\top \boldsymbol{h} = 1. \tag{2.30}$$

All eigenvalues of Q are different and positive since X_n is anisotropic, so are all eigenvalues of $\hat{Q}$. Consequently, $\hat{Q}^2$ has a unique spectral decomposition. It is easy to see that

$$\boldsymbol{y}^\top \hat{Q}^2 \boldsymbol{y} = \boldsymbol{y}^\top \left(\sum_{i=1}^{n} \mu_i \boldsymbol{v}_i \boldsymbol{v}_i^\top \right) \boldsymbol{y} = \sum_{i=1}^{n} \mu_i (\boldsymbol{v}_i^\top \boldsymbol{y})^2, \tag{2.31}$$

where $\mu_i = 1/(\gamma\lambda_i - \eta)^2$ is the i-th largest eigenvalue of $\hat{Q}^2$.

Step 2. Define a linear mapping

$$\begin{aligned} \psi : \mathbb{R}_\beta^n &\mapsto \mathbb{R}^n \\ \boldsymbol{\beta} &\mapsto \beta_1 \boldsymbol{v}_1 + \cdots + \beta_n \boldsymbol{v}_n, \end{aligned}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_n)^\top$, $\mathbb{R}_\beta^n = \mathbb{R}^n$ and we use the symbol $\mathbb{R}_\beta^n$ to distinguish the domain and the range. It is obvious that ψ is a vector space automorphism, and the set of vectors $\{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n\}$ and the set of images $\{\psi(\boldsymbol{e}_1), \ldots, \psi(\boldsymbol{e}_n)\} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ are completely different bases due to Theorem 2.10 and Lemma 2.9.

Let $\boldsymbol{\beta} = \psi^{-1}(\boldsymbol{y})$. Then,

$$\|\boldsymbol{y}\|_2 = \sqrt{n} \quad \Rightarrow \quad \beta_1^2 + \cdots + \beta_n^2 = n \tag{2.32}$$

$$(2.30) + (2.31) \quad \Rightarrow \quad \mu_1 \beta_1^2 + \cdots + \mu_n \beta_n^2 = 1. \tag{2.33}$$

Eq. (2.32) represents a hyper-ball in $\mathbb{R}_\beta^n$, and Eq. (2.33) represents an irrotational ellipsoid in $\mathbb{R}_\beta^n$ since $\mu_1, \ldots, \mu_n$ are distinct eigenvalues. As a result, given any other $\boldsymbol{\beta}' = (\beta_1', \ldots, \beta_n')^\top$ satisfying (2.32) and (2.33), there exist three disjoint index sets $\mathcal{J}_+, \mathcal{J}_-, \mathcal{J}_0$ such that $\mathcal{J}_+ \cup \mathcal{J}_- \cup \mathcal{J}_0 = \{1, \ldots, n\}$ and

$$\begin{aligned} &\forall j \in \mathcal{J}_+, \beta_j \neq 0, \beta_j + \beta_j' = 0 \\ &\forall j \in \mathcal{J}_-, \beta_j \neq 0, \beta_j - \beta_j' = 0 \\ &\forall j \in \mathcal{J}_0, \beta_j = \beta_j' = 0. \end{aligned}$$

Step 3. For another arbitrarily chosen minimum $\boldsymbol{h}'$ of (2.4), let $\boldsymbol{y}' = \operatorname{sign}(\boldsymbol{h}')$ and $\boldsymbol{\beta}' = \psi^{-1}(\boldsymbol{y}')$, then $\boldsymbol{\beta}'$ is also a solution to the system of Eqs. (2.32) and (2.33), and it is guaranteed the existence of aforementioned $\mathcal{J}_+, \mathcal{J}_-, \mathcal{J}_0$.

Notice that $\forall j \in \mathcal{J}_+$,

$$\boldsymbol{v}_j^\top(\boldsymbol{y} + \boldsymbol{y}') = \beta_j + \beta_j' = 0 \quad \Rightarrow \quad \boldsymbol{v}_j^\top \boldsymbol{y}' = -\boldsymbol{v}_j^\top \boldsymbol{y}.$$

Similarly, $\forall j \in \mathcal{J}_-, \boldsymbol{v}_j^\top \boldsymbol{y}' = \boldsymbol{v}_j^\top \boldsymbol{y}$ and $\forall j \in \mathcal{J}_0, \boldsymbol{v}_j^\top \boldsymbol{y}' = \boldsymbol{v}_j^\top \boldsymbol{y} = 0$. In a word, we have $(\boldsymbol{v}_j^\top \boldsymbol{y}')^2 = (\boldsymbol{v}_j^\top \boldsymbol{y})^2$ for all $j = 1, \ldots, n$. Hence,

$$\boldsymbol{y}^\top Q \boldsymbol{y} = \sum_{j=1}^{n} \lambda_j (\boldsymbol{v}_j^\top \boldsymbol{y})^2 = \sum_{j=1}^{n} \lambda_j (\boldsymbol{v}_j^\top \boldsymbol{y}')^2 = \boldsymbol{y}'^\top Q \boldsymbol{y}',$$

which indicates that $(\boldsymbol{y} + \boldsymbol{y}')^\top Q (\boldsymbol{y} - \boldsymbol{y}') = 0$.

Let $\delta_1 = [\boldsymbol{y}]_1, \cdots, \delta_n = [\boldsymbol{y}]_n$ and $\mathcal{K} = \{k \mid [\boldsymbol{y}]_k = [\boldsymbol{y}']_k, 1 \le k \le n\}$. Subsequently, check the condition Eq. (2.21), and we will find that

$$\left(\sum\nolimits_{k \in \mathcal{K}} \delta_k \boldsymbol{e}_k\right)^\top Q \left(\sum\nolimits_{k \notin \mathcal{K}} \delta_k \boldsymbol{e}_k\right) = \frac{1}{4} (\boldsymbol{y} + \boldsymbol{y}')^\top Q (\boldsymbol{y} - \boldsymbol{y}') = 0.$$

However, X_n is SI-asymmetric, and there must be $\#\mathcal{K} = 0$ or $\#\mathcal{K} = n$, i.e., $\boldsymbol{y}' = -\boldsymbol{y}$ or $\boldsymbol{y}' = \boldsymbol{y}$. Therefore, $d_{\mathcal{H}}(\boldsymbol{h}, \boldsymbol{h}') = 0$ and $\boldsymbol{h}'$ is equivalent to $\boldsymbol{h}$. □

2.9.6 Proof of Lemma 2.18

For any $\boldsymbol{h} \in \widetilde{\mathcal{H}}_Q$, there exists $\boldsymbol{\alpha} \in \mathbb{R}^n$ such that $\boldsymbol{h} = U\boldsymbol{\alpha}$, where U consists of n orthonormal eigenvectors of Q, and $\|\boldsymbol{\alpha}\|_2 = 1$ since $\|\boldsymbol{h}\|_2 = 1$ and $U^\top U = I$. The expression $\boldsymbol{h} = U\boldsymbol{\alpha}$ is an unlabeled-labeled representation (ULR) since U only has the information about unlabeled samples. Each column of U has a unit length, and thus $\|U\|_F^2 = n$ where $\|\cdot\|_F$ is the Frobenius norm. The first part of the upper bound, namely,

$$\mathcal{R}_n(\widetilde{\mathcal{H}}_Q) \le \sqrt{2n/n'(n - n')},$$

comes from Eqs. (20)–(22) of El-Yaniv and Pechyony (2009).

Let $\hat{Q} = (\gamma Q - \eta^* I_n)^{-1}$. Another ULR is shown in Eq. (2.29), in the proof of Theorem 2.13:

$$\boldsymbol{h} = \hat{Q} \operatorname{sign}(\boldsymbol{h}).$$

It is obvious that $\{1/(\gamma\lambda_i - \eta^*)\}_{i=1}^n$ are the eigenvalues of $\hat{Q}$. Subsequently, the second part of the upper bound, i.e.,

$$\mathcal{R}_n(\widetilde{\mathcal{H}}_Q) \leq \sqrt{\frac{2}{n'(n-n')}} \left(\sum_{i=1}^{n} \frac{n}{(\gamma\lambda_i - \eta^*)^2} \right)^{1/2},$$

can be derived from Eqs. (20)–(22) of El-Yaniv and Pechyony (2009) with $\mu_1 = \sqrt{n}$. Furthermore, Eq. (2.29) is also a kernel ULR, since $\hat{Q}$ is symmetric positive definite and can be viewed as a kernel matrix. Thereby we can obtain the third part of the upper bound

$$\mathcal{R}_n(\widetilde{\mathcal{H}}_Q) \leq \sqrt{\frac{2}{n'(n-n')}} \left(\sum_{i=1}^{n} \frac{\mu}{\gamma\lambda_i - \eta^*} \right)^{1/2}$$

based on Eqs. (23)–(25) of El-Yaniv and Pechyony (2009) with $\mu_2 = \sqrt{\mu}$. □

Chapter 3

Information-theoretic Semi-supervised Metric Learning

In this chapter, we present information-theoretic semi-supervised metric learning. Our contributions can be summarized as two folds.

- We formulate supervised metric learning as an instance of the generalized maximum entropy distribution estimation (Dudík and Schapire, 2006);
- We propose a semi-supervised extension of the above estimation following entropy regularization (Grandvalet and Bengio, 2005).

This chapter is organized as follows. Section 3.1 describes the background. The proposed model and algorithm are formulated in Sections 3.2 and 3.3. In Section 3.4, we discuss the sparsity issues and additional justifications. Related works are compared in Section 3.5. Experimental results are reported in Section 3.6.

3.1 Introduction

How to learn a good distance metric for the input data domain is a crucial issue for many distance-based learning algorithms. The majority of metric learning methods developed in the last decade fall into three types:

(a) Supervised type requiring class labels (e.g., Chiaromonte and Cook, 2002; Sugiyama, 2007; Fukumizu et al., 2009);

(b) Supervised type requiring weak labels, that is, $\{\pm 1\}$-valued labels that indicate the similarity/dissimilarity of data pairs (e.g., Xing et al., 2003; Goldberger et al., 2005; Weinberger et al., 2006; Globerson and Roweis, 2006; Torresani and Lee, 2007; Davis et al., 2007). See the illustration in Figure 3.1;

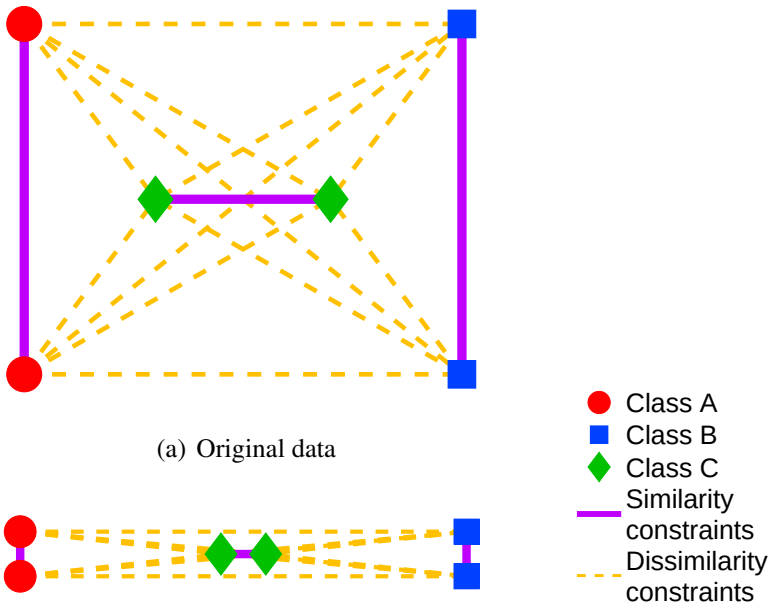
(c) Unsupervised type that requires no label information (e.g., Roweis and Saul, 2000; Tenenbaum et al., 2000; Belkin and Niyogi, 2002). See the illustration in Figure 3.2.

There are many examples of using weak labels: Asking anybody for weak labels of image pairs from Flickr, and asking anybody for weak labels of book/DVD pairs from Amazon. Compared with class labels, weak labels are fast and cheap, and crowdsourcing is allowed. Notice that classical supervised metric learning methods have a strict limitation. Algorithms in (a) need all class labels, and algorithms in (b) still need each data point be involved in at least one weak label, otherwise the algorithm cannot see this point. These requirements are sometimes problematic for real-world applications. Based on the belief that preserving the intrinsic geometric structure of all training data in an unsupervised manner can be better than strongly relying on limited labeled data, semi-supervised metric learning has emerged. To the best of our knowledge, all previous semi-supervised methods that extend types (a) and (b) employ *off-the-shelf* unsupervised techniques in type (c). For example,

- Principal component analysis (e.g., Yang et al., 2006; Sugiyama et al., 2010);
- Manifold regularization or embedding (e.g., Hoi et al., 2008; Baghshah and Shouraki, 2009; Zha et al., 2009; Liu et al., 2010).

They can be regarded as propagating labels along an assistant metric by some unsupervised techniques and learning a target metric implicitly in a supervised manner.

However, the target and assistant metrics assume different forms: The target metric is a Mahalanobis distance defined over a Euclidean space, while the assistant metric is a geodesic distance defined over a curved space or a Riemannian manifold. The target and assistant metrics also share slightly different goals:



(a) Original data

(b) Projected data

Figure 3.1: Illustration of supervised metric learning based on weak labels. In this figure, we have three classes, each with two labeled data. The goal is to find a metric so that data in the same class are close and data from different classes are far apart. Note that the class labels will not be revealed to algorithms, and we show the projected data here, since the Mahalanobis distance of the original data equals the Euclidean distance of the projected data.

- The target metric aims at a metric so that data in the same class are close and data from different classes are far apart (as illustrated in Figure 3.1), for instance, Fisher discriminant analysis (Fisher, 1936);

- The assistant metric tries to identify and preserve the intrinsic geometric structure of unlabeled data (as illustrated in Figure 3.2), for instance, Laplacian eigenmaps (Belkin and Niyogi, 2002).

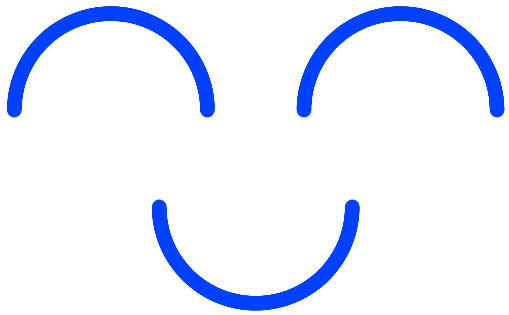
Simply putting two metrics together works in practice, but the paradigm is conceptually neither natural nor unified.

In this chapter, we propose SERAPH (SEmi-supervised metRic leArning Paradigm with Hyper-sparsity), a novel semi-supervised metric learning approach, as an *information-theoretic alternative* to those manifold-based methods. Our idea is to optimize a metric by optimizing a conditional probability parameterized by that metric. We maximize the entropy of that probability on labeled data, and minimize the entropy of that probability on unlabeled data via *entropy regularization* (Grandvalet and Bengio, 2005), which can achieve the sparsity of the posterior distribution (Graça et al., 2009; Gillenwater et al., 2011), i.e., unlabeled data can be classified with high confidence. Furthermore, we employ *mixed-norm regularization* (Argyriou et al., 2007) to encourage the sparsity of the projection matrix (Ying et al., 2009), i.e., the low-rank projection matrix induced from the metric can carry out dimensionality reduction adaptively. Unifying the posterior sparsity and the projection sparsity brings to us the *hyper-sparsity*. Thanks to the hyper-sparsity, the Mahalanobis distance learned by SERAPH possesses high discriminability even under a noisy environment. Notice that our extension is compatible with the manifold-based extension, which means that SERAPH can have an additional manifold regularization term.

3.2 SERAPH, the Model

In this section, we formulate the model of SERAPH. We first propose the supervised part, and then introduce its regularization terms.

(a) Data clouds



(b) Identified manifolds

Figure 3.2: Illustration of manifold learning, a subclass of unsupervised metric learning. Here, the goal is to identify and preserve the intrinsic geometric structure of unlabeled data. Previous semi-supervised metric learning methods share the same assumption with manifold learning, though they usually carry it out in a regularization manner.

3.2.1 Problem Setting

Suppose that we have a training set $\mathcal{X} = \{x_i \mid x_i \in \mathbb{R}^m\}_{i=1}^n$ that contains n points each with m features. Let the set of similar data pairs be

$$\mathcal{S} = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are similar}\},$$

and the set of dissimilar data pairs be

$$\mathcal{D} = \{(x_i, x_j) \mid x_i \text{ and } x_j \text{ are dissimilar}\}.$$

With some abuse of terminology, we refer to $\mathcal{S} \cup \mathcal{D}$ as the labeled data, and

$$\mathcal{U} = \{(x_i, x_j) \mid i \neq j, (x_i, x_j) \notin \mathcal{S} \cup \mathcal{D}\}$$

as the unlabeled data. A weak label $y_{i,j}$ is assigned to (x_i, x_j) such that

$$y_{i,j} = \begin{cases} +1 & \text{if } (x_i, x_j) \in \mathcal{S}, \\ -1 & \text{if } (x_i, x_j) \in \mathcal{D}, \\ \text{undefiend} & \text{if } (x_i, x_j) \in \mathcal{U}. \end{cases}$$

We abbreviate $\sum_{(x_i,x_j)\in\mathcal{S}\cup\mathcal{D}}$, $\sum_{(x_i,x_j)\in\mathcal{U}}$ and $\sum_{y\in\{+1,-1\}}$ to $\sum_{\mathcal{S}\cup\mathcal{D}}$, $\sum_{\mathcal{U}}$ and $\sum_y$ for simplicity. Consider learning a Mahalanobis distance metric for $x, x' \in \mathbb{R}^m$ of the form

$$d(x, x') = \|x - x'\|_A = \sqrt{(x - x')^\top A (x - x')}, \tag{3.1}$$

where $^\top$ is the transpose operator, and $A \in \mathbb{R}^{m\times m}$ is a symmetric positive semi-definite matrix to be learned[1]. The probability of labeling $(x, x') \in \mathbb{R}^m \times \mathbb{R}^m$ with $y = \pm 1$ is denoted by $p^A(y \mid x, x')$ that is explicitly parameterized by the matrix A. When the pair (x, x') comes from $\mathcal{S} \cup \mathcal{D} \cup \mathcal{U}$, $p^A(y \mid x_i, x_j)$ is abbreviated into $p^A_{i,j}(y)$.

3.2.2 Basic Model

To begin with, we derive a probabilistic model to investigate the conditional probability of $y = \pm 1$ given $(x, x') \in \mathbb{R}^m \times \mathbb{R}^m$. We resort to a parametric form of

[1] In the rest of this chapter, the matrix A is always assumed symmetric and positive semi-definite if it is an optimization variable, and the constraints $A = A^\top$ and $A \succeq 0$ will not be explicitly written for convenience.

$p^A(y \mid x, x')$, and we will focus on this parametric form for the out-of-sample ability.

The *maximum entropy principle* (Jaynes, 1957; Berger et al., 1996) suggests us to choose the probability distribution with the maximum entropy out of all distributions that match the data moments. Let[2]

$$H(p^A_{i,j}) = -\sum\nolimits_y p^A_{i,j}(y) \ln p^A_{i,j}(y)$$

be the entropy of the conditional probability $p^A_{i,j}(y)$, and

$$f(x, x', y; A) : \mathbb{R}^m \times \mathbb{R}^m \times \{+1, -1\} \mapsto \mathbb{R}$$

be a feature function that is convex with respect to A, then the constrained optimization problem is

$$\begin{aligned} \max_{A, p^A_{i,j}, \xi} \quad & \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} H(p^A_{i,j}) - \frac{1}{2\gamma}\xi^2 \\ \text{s.t.} \quad & \left| \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \mathbb{E}_{p^A_{i,j}}[f(x_i, x_j, y; A)] - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} f(x_i, x_j, y_{i,j}; A) \right| \leq \xi, \end{aligned} \tag{3.2}$$

where ξ is a slack variable and $\gamma > 0$ is a regularization parameter. After the introduction of ξ, distributions are allowed to match two data moments in a way that is not strictly exact. The penalty term $\xi^2/(2\gamma)$ in the objective function presumes the Gaussian prior of the expected data moment

$$\sum_{\mathcal{S}\cup\mathcal{D}} \mathbb{E}_{p^A_{i,j}}[f(x_i, x_j, y; A)]$$

from the empirical data moment

$$\sum_{\mathcal{S}\cup\mathcal{D}} f(x_i, x_j, y_{i,j}; A),$$

which is essentially consistent in spirit with the *generalized maximum entropy principle* (Dudík and Schapire, 2006). Please see Section 3.4.2 for the necessity of the introduction of the slack variable ξ and the alternative explanation of optimization problem (3.2) in the sense of the generalized maximum entropy principle.

[2]Throughout this thesis, we adopt that $0 \ln 0 = \lim_{x\to 0^+} x \ln x = 0$.

Theorem 3.1. *The primal solution p^{*A} is given in terms of the dual solution (A^*, κ^*) by*

$$p^{*A}(y \mid x, x') = \frac{\exp(\kappa^* f(x, x', y; A^*))}{Z(x, x'; A^*, \kappa^*)}, \tag{3.3}$$

where $Z(x, x'; A, \kappa) = \sum_{y'} \exp(\kappa f(x, x', y'; A))$ is the partition function, and (A^*, κ^*) can be obtained by solving the dual problem

$$\min_{A,\kappa} \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln Z(x_i, x_j; A, \kappa) - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \kappa f(x_i, x_j, y_{i,j}; A) + \frac{\gamma}{2}\kappa^2. \tag{3.4}$$

Define the regularized log-likelihood function on labeled data (i.e., on observed weak labels) as

$$\mathcal{L}_1(A, \kappa) = \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln p^A_{i,j}(y_{i,j}) - \frac{\gamma}{2}\kappa^2.$$

Then, for supervised metric learning, the regularized maximum log-likelihood estimation and the generalized maximum entropy estimation are equivalent.[3]

When considering $f(x, x', y; A)$ that should take moments about the metric information into account, we propose[4]

$$f(x, x', y; A, \eta) = -\frac{y}{2}(\|x - x'\|^2_A - \eta), \tag{3.5}$$

where $\eta > 0$ is a hyperparameter served as the threshold to separate the similar and dissimilar data pairs in $\mathcal{S}$ and $\mathcal{D}$ under the target metric $d(x, x')$. Now the probabilistic model (3.3) becomes

$$p^A(y \mid x, x') = \frac{1}{1 + \exp(\kappa y(\|x - x'\|^2_A - \eta))}. \tag{3.6}$$

For the optimal solution (p^{*A}, A^*, κ^*) and reasonable η, we hope for two properties:

(i) The feature function can indicate the correctness of the observed weak labels, i.e.,

$$f(x_i, x_j, y_{i,j}; A^*, \eta) = -y_{i,j}(\|x_i - x_j\|^2_{A^*} - \eta)/2 > 0;$$

[3]The proofs of all theorems are in Section 3.7.

[4]Note that in Niu et al. (2012) the feature function is $f(x, x', y; A, \eta) = y(\|x - x'\|^2_A - \eta)/2$ that has the opposite sign with the feature function in Eq. (3.5). However, they are equivalent feature functions, since the signs of κ^* are also opposite here and there.

(ii) The probabilistic model can correctly classify the observed weak labels, i.e.,

$$p^{*A}(y_{i,j} \mid x_i, x_j) = 1/\left(1 + \exp(\kappa^* y_{i,j}(\|x_i - x'_j\|^2_{A^*} - \eta))\right) > 1/2.$$

Therefore, there must be $\kappa^* > 0$.

Although we use Eq. (3.5) as our feature function in this chapter, other feature functions emphasizing different perspectives of the metric information are available. In fact, optimization (3.2) can even be applied to other problem settings. For instance, the local distance metric feature function is given by

$$f(x, x', y; A, \eta) = -\frac{y}{2}\left(\frac{\|x - x'\|_A}{\|x - x'\|_2} - \eta\right),$$

and the global distance metric feature function for multi-label metric learning can be defined as

$$f(x, x', y, y'; A, \eta) = \left(\frac{1}{2} - \frac{\langle y, y'\rangle}{\|y\|_2\|y'\|_2}\right)(\|x - x'\|^2_A - \eta),$$

where the labels y and y' are binary-valued vectors.

3.2.3 Regularization

In this subsection, we extend $\mathcal{L}_1(A, \kappa)$ defined above to semi-supervised metric learning by entropy regularization, and further regularize it by trace-norm regularization.

Our unsupervised part does not rely upon the manifold assumption and is not in the paradigm of smoothing the projected training data. In order to be integrated with the supervised part more naturally, our unsupervised part follows the *minimum entropy principle* (Grandvalet and Bengio, 2005), and hence $p^A_{i,j}$ should have low entropy (i.e., low uncertainty) for $(x_i, x_j) \in \mathcal{U}$. Generally speaking, the resultant discriminative models prefer peaked distributions on unlabeled data, which can carry out a probabilistic *low-density separation*. Subsequently, according to Grandvalet and Bengio (2005), our optimization becomes

$$\begin{aligned}\max_{A,\kappa}\ \mathcal{L}_2(A, \kappa) &= \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln p^A_{i,j}(y_{i,j}) \\ &\quad + \mu \sum\nolimits_{\mathcal{U}} \sum\nolimits_{y} p^A_{i,j}(y) \ln p^A_{i,j}(y) - \frac{\gamma}{2}\kappa^2,\end{aligned}$$

where $\mu \geq 0$ is a regularization parameter.

In addition, we hope for the dimensionality reduction ability by encouraging a low-rank projection matrix induced from A. It will be helpful in dealing with corrupted data or data distributed intrinsically in a low-dimensional subspace. It is known that the trace is a convex relaxation of the rank for positive semi-definite matrices, so we revise our optimization problem into

$$\begin{aligned} \max_{A,\kappa} \ & \mathcal{L}(A,\kappa) = \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln p^A_{i,j}(y_{i,j}) \\ & + \mu \sum\nolimits_{\mathcal{U}} \sum\nolimits_{y} p^A_{i,j}(y) \ln p^A_{i,j}(y) - \frac{\gamma}{2}\kappa^2 - \lambda \operatorname{tr}(A), \end{aligned} \tag{3.7}$$

where $\operatorname{tr}(A)$ is the trace of A, and $\lambda \geq 0$ is a regularization parameter.

Optimization problem (3.7) is the final model of SERAPH. We say that SERAPH is equipped with the hyper-sparsity when both μ and λ are positive and hence both regularization terms are active. The hyper-sparsity, as well as the posterior and projection sparsity, will be discussed in Section 3.4.1. Moreover, SERAPH possesses standard kernel and manifold extensions, and we will explain them later.

3.3 SERAPH, the Algorithm

In this section, we reduce the model defined in optimization (3.7) to a form that is easy to handle, and develop a practical EM-like algorithm for solving the reduced optimization problem.

3.3.1 Reduction

First, we eliminate κ from optimization (3.7) and thus reduce it to a simpler form, thanks to the fact that we use a single feature function (3.5) in optimization (3.2).

Theorem 3.2. *Define the reduced optimization problem as*[5]

$$\begin{aligned} \max_{A} \ & \hat{\mathcal{L}}(A) = \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln \hat{p}^A_{i,j}(y_{i,j}) \\ & + \mu \sum\nolimits_{\mathcal{U}} \sum\nolimits_{y} \hat{p}^A_{i,j}(y) \ln \hat{p}^A_{i,j}(y) - \hat{\lambda} \operatorname{tr}(A), \end{aligned} \tag{3.8}$$

[5]The new functions and parameters are denoted by $\hat{\cdot}$ within this theorem for the sake of clarity.

where the reduced probabilistic model is

$$\hat{p}^A(y \mid x, x') = \frac{1}{1 + \exp(y(\|x - x'\|_A^2 - \hat{\eta}))}. \tag{3.9}$$

Let $\hat{A}$ *and* (A^*, κ^*) *be the optimal solutions to optimizations* (3.8) *and* (3.7)*, respectively. Then, there exist well-defined hyperparameters* $\hat{\eta}$ *and* $\hat{\lambda}$*, such that*

(i) $d(x, x')$ *parameterized by* $\hat{A}$ *is equivalent to* $d(x, x')$ *parameterized by* A^**, i.e.,*

$$\forall x, x' \in \mathbb{R}^m, \ \frac{d(x, x'; \hat{A})}{d(x, x'; A^*)} = \text{Const.};$$

(ii) $\hat{p}^A(y \mid x, x')$ *parameterized by* $\hat{A}$ *and* $\hat{\eta}$ *is identical to the original* $p^A(y \mid x, x')$ *parameterized by* A^**,* κ^* *and* η*, i.e.,*

$$\forall x, x' \in \mathbb{R}^m, y \in \{+1, -1\}, \ \hat{p}^A(y \mid x, x'; \hat{A}, \hat{\eta}) = p^A(y \mid x, x'; A^*, \kappa^*, \eta).$$

Remark 3.3. After the reduction of Theorem 3.2, γ has been dropped, η and λ have been modified, but the regularization parameter μ remains the same, which means that the tradeoff between the supervised and unsupervised parts has not been affected.

3.3.2 EM-like Algorithm

There exist several approaches for solving optimization (3.8), for example, EM algorithms or gradient ascent algorithms (cf. Grandvalet and Bengio, 2006, pp. 155–158). We would like to pose it as an EM-like iterative scheme to access the derandomization by the initial solution, the stability for the gradient update, and the insensitivity to the step size, just to name a few nice algorithmic properties.

The EM-like iterative scheme runs as follows. In the beginning, we initialize a nonparametric probability $q(y \mid x_i, x_j)$ for each pair $(x_i, x_j) \in \mathcal{U}$. Then the M-Step and the E-Step get executed repeatedly until some stopping conditions are satisfied. The initial solution of our current implementation is

$$q(y = -1 \mid x_i, x_j) = 1,$$

which means that at the beginning we treat all unlabeled pairs as dissimilar pairs.

At the t-th E-Step, similarly to Graça et al. (2009) and Gillenwater et al. (2011), we have for each pair $(x_i, x_j) \in \mathcal{U}$ that

$$\min_q \ \mathrm{KL}(q \,||\, p^A_{i,j}) + \mu \mathbb{E}_q[-\ln p^A_{i,j}(y)], \tag{3.10}$$

where KL is the Kullback-Leibler divergence, and $p^A_{i,j}$ is parameterized by the metric $A^{(t)}$ found at the last M-Step. Optimization (3.10) can be viewed as a projection of probabilities from $p^A(y \mid x_i, x_j)$ to $q(y \mid x_i, x_j)$ restricted to $(x_i, x_j) \in \mathcal{U}$, and it can be solved analytically as shown in Theorem 3.4 below. It is interesting that the optimal solution to (3.10) looks quite similar to the E-Step of the deterministic annealing EM algorithm described in Grandvalet and Bengio (2006, pp. 155–158). From a viewpoint of minimizing the thermodynamic free energy, the temperature is $1 - \mu$ in their E-Step and $1/(1 + \mu)$ in ours.

Theorem 3.4. *The optimal solution to optimization* (3.10) *is given by*

$$q(y \mid x_i, x_j) = \frac{\left(p^A_{i,j}(y)\right)^{1+\mu}}{\sum_{y'} \left(p^A_{i,j}(y')\right)^{1+\mu}}. \tag{3.11}$$

On the other hand, at the t-th M-Step, we find new metric $A^{(t)}$ through $q(y \mid x_i, x_j)$ which is generated in the last E-Step and only defined for $(x_i, x_j) \in \mathcal{U}$:

$$\begin{aligned} \max_A \ \mathcal{F}(A) = &\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln p^A_{i,j}(y_{i,j}) \\ &+ \mu \sum\nolimits_{\mathcal{U}} \sum\nolimits_{y} q(y \mid x_i, x_j) \ln p^A_{i,j}(y) - \lambda \operatorname{tr}(A). \end{aligned} \tag{3.12}$$

Since the convexity of the objective $\mathcal{F}(A)$ has already been implied by the convexity of the feature function $f(x, x', y; A)$ with respect to A (Boyd and Vandenberghe, 2004, p. 74), optimization (3.12) could be solved without worry about the local maximum by the gradient projection method (Polyak, 1967), using the calculation of the gradient matrix $\nabla\mathcal{F}$ given by

$$\begin{aligned} \nabla\mathcal{F}(A) = &-\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} y_{i,j}(1 - p^A_{i,j}(y_{i,j}))(x_i - x_j)(x_i - x_j)^\top \\ &- \mu \sum\nolimits_{\mathcal{U}} \sum\nolimits_{y} y q(y \mid x_i, x_j)(1 - p^A_{i,j}(y))(x_i - x_j)(x_i - x_j)^\top \\ &- \lambda I_m. \end{aligned} \tag{3.13}$$

A remarkable property of $\mathcal{F}(A)$ is that its gradient is uniformly bounded, regardless of the scale of A, i.e., the magnitude of $\operatorname{tr}(A)$.

Theorem 3.5. *The objective function $\mathcal{F}(A)$ is Lipschitz continuous, and the best Lipschitz constant $\mathrm{Lip}_{\|\cdot\|_F}(\mathcal{F})$ with respect to the Frobenius norm $\|\cdot\|_F$ satisfies*

$$\mathrm{Lip}_{\|\cdot\|_F}(\mathcal{F}) \leq (\#\mathcal{S} + \#\mathcal{D} + \mu\#\mathcal{U})(\mathrm{diam}(\mathcal{X}))^2 + \lambda m, \tag{3.14}$$

where $\mathrm{diam}(\mathcal{X}) = \max_{x_i, x_j \in \mathcal{X}} \|x_i - x_j\|_2$ is the diameter of $\mathcal{X}$, and $\#$ measures the cardinality of a set.

3.3.3 Asymptotic Time Complexity

As mentioned before, we solve optimization (3.12) by the gradient projection method. Let π be the operator that projects a symmetric matrix to the cone of symmetric positive semi-definite matrices, which includes eigen-decomposing a symmetric matrix and recovering it from the positive eigenvalues and the eigenvectors associated with those positive eigenvalues. Assume s_k is the step size of the k-th iteration, and denote the gradient projection update as

$$A_{k+1} \leftarrow \pi(A_k + s_k \nabla \mathcal{F}).$$

Now consider the asymptotic time complexity of the EM-like algorithm. The computational complexity of calculating the gradient matrix $\nabla\mathcal{F}$ is $O(n^2 m)$, the complexity of projecting the matrix $(A_k + s_k \nabla \mathcal{F})$ back to the positive semi-definite cone is $O(m^3)$, and thus each inner iteration takes $O(n^2 m + m^3)$ time. Let ϵ' be a stopping criterion of the M-Step such that $\mathcal{F}(A)$ has to increase at least ϵ', then the asymptotic time complexity of each M-Step will be

$$O\left(\frac{n^2 m + m^3}{\epsilon'}\right).$$

Secondly, it is easy to see that each E-Step consumes the time of order $O(n^2)$. Thirdly, let ϵ be a stopping criterion of the whole algorithm such that $\mathcal{L}(A)$ must increase at least ϵ, the total number of outer iterations is then $O(1/\epsilon)$. Therefore, the overall asymptotic time complexity is

$$O\left(\frac{n^2 m + m^3}{\epsilon\epsilon'}\right).$$

3.3.4 Implementation

The warm start is used for M-Steps, and the initial solution of the first M-Step is

$$A_0 = \frac{m}{m+1} I_m,$$

where I_m is the identity matrix of size m. We employ a heuristic strategy for the step sizes. At the k-th iteration of the gradient projection method, we initially set the step size

$$s_k = \frac{m}{10\sqrt{k}} \frac{1}{\|\nabla \mathcal{F}\|_F}.$$

Then we try the aforementioned gradient projection update

$$A_{k+1} \leftarrow \pi(A_k + s_k \nabla \mathcal{F})$$

and keep it if A_{k+1} improves A_k, otherwise we decrease s_k by half

$$s_k \leftarrow \frac{s_k}{2}$$

and try the update again. The maximum number of these attempts is 20. We will not be trapped by local maxima, since we are maximizing a concave objective function. Furthermore, the maximum number $k_{\max}$ of inner gradient projection iterations and the maximum number $t_{\max}$ of outer EM iterations are 10. We halt the algorithm before reaching the maximum iteration numbers if either the solutions have been converged for both inner and outer iterations, or we fail to further improve the objective function $\mathcal{L}$ after the last M-Step.

In practice, the main computational bottleneck is how to compute $\nabla \mathcal{F}(A)$ in a high-level language such as Matlab without computationally-inefficient double FOR loops, as well as computing $\mathcal{L}(A)$, $\mathcal{F}(A)$ and $q(y \mid x_i, x_j)$ for all $(x_i, x_j) \in \mathcal{U}$ without double FOR loops. Fortunately, there are nice implementations. Without loss of generality, we describe the efficient method for computing $\nabla \mathcal{F}(A)$ in Algorithm 2. We observed that in our experiments Algorithm 2 was at least twenty times faster than the naive implementation of the same subroutine using double FOR loops. SERAPH was in general the second most computationally-efficient algorithm in our experiments, and manifold Fisher discriminant analysis (Baghshah and Shouraki, 2009), as the most computationally-efficient one, only involves solving a linear system in locally linear embedding (Roweis and Saul, 2000) and a generalized eigenvalue problem as in Fisher discriminant analysis (Fisher, 1936).

Algorithm 2 Efficient Computation of $\nabla\mathcal{F}(A)$

Input: the current solution A,
$X \in \mathbb{R}^{n\times m}$ that is the design matrix of $\mathcal{X}$,
$S \in \mathbb{R}^{n\times n}$, $S(i,j) = 1$ if $(x_i, x_j) \in \mathcal{S}$ and $S(i,j) = 0$ otherwise,
$D \in \mathbb{R}^{n\times n}$, $D(i,j) = 1$ if $(x_i, x_j) \in \mathcal{D}$ and $D(i,j) = 0$ otherwise,
$Q \in \mathbb{R}^{n\times n}$, $Q(i,j) = q(+1 \mid x_i, x_j)$ for $(x_i, x_j) \in \mathcal{U}$

Output: $\nabla\mathcal{F}(A)$

1: Compute all pairwise Mahalanobis distances by

$$\bar{x} \leftarrow \mathrm{diag}(XAX^\top), M \leftarrow \mathrm{repmat}(\bar{x}, 1, n) + \mathrm{repmat}(\bar{x}^\top, n, 1) - 2XAX^\top.$$

2: Compute

$$P \leftarrow 1./(1 + \exp(M - \eta)),$$

where $./$ and $\exp$ are the element-wise matrix division and exponential function.

3: Let $C \in \mathbb{R}^{n\times n}$ that will store all the coefficients of $(x_i - x_j)(x_i - x_j)^\top$. Initialize it as $C \leftarrow 0_{n\times n}$.

4: Let $O \leftarrow 1_{n\times n}$, and subsequently

$$C_S \leftarrow P_S - O_S, C_D \leftarrow P_D,$$

where the subscripts S and D mean that the matrix operations are done only for the entries corresponding to $S(i,j) = 1$ or $D(i,j) = 1$.

5: Get the matrix form of $\mathcal{U}$ by

$$U \leftarrow O - S - D - I_n,$$

and compute

$$C_U \leftarrow \mu(Q_U .* (P_U - O_U) + (O_U - Q_U) .* P_U)$$

where $.*$ is the element-wise matrix multiplication.

6: Finally,

$$\nabla\mathcal{F}(A) \leftarrow X^\top(\mathrm{repmat}(\mathrm{sum}(C, 2), 1, m) .* X) - X^\top CX - \lambda I_m.$$

3.4 Discussions

We have left out a few theoretical arguments when we proposed the model of SERAPH in order to keep the presentation as concise as possible. Therefore, we discuss the sparsity issue in the sense of metric learning and give two additional justifications of our formulation in this section.

3.4.1 Posterior Sparsity and Projection Sparsity

Sparse metric learning may have different meanings, since we learn a metric with low-rank linear projections by optimizing a conditional probability, where the optimization variable is actually a square matrix. First of all, we would like to explain the meaning of our sparsity and claim that we can obtain the *posterior sparsity* (Graça et al., 2009) by entropy regularization and the *projection sparsity* (Ying et al., 2009) by trace-norm regularization. The arguments are as follows.

By a "sparse" posterior distribution, we mean that the uncertainty (e.g., the entropy or variance) of the probability $p^A_{i,j}$ for $(x_i, x_j) \in \mathcal{U}$ is low, such that (x_i, x_j) can be classified to be a similar or dissimilar pair with high confidence.

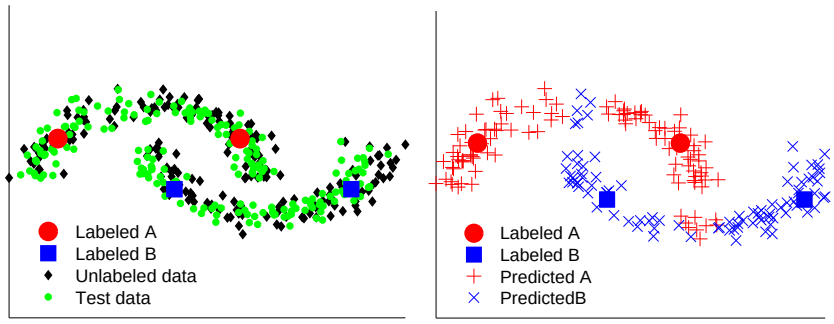
Figure 3.3 is an illustrative example of metric learning with the sparse and non-sparse posterior distributions. Recall that the supervised metric learning aims at learning a Mahalanobis distance metric under which data in the same class are close and data from different classes are far apart. It would result in the metric which ignores the horizontal feature and only focuses on the vertical feature. Nonetheless, the horizontal feature is useful, and taking care of the posterior sparsity would lead to a better metric as shown in subfigures (e) and (f). As a consequence, we prefer taking the posterior sparsity into account in addition to the aforementioned goal of supervised metric learning, and then the risk of overfitting weakly labeled data can be significantly reduced.

We can rewrite $\mathcal{L}_2(A, \kappa)$ as a soft posterior regularization objective function (Graça et al., 2009; Gillenwater et al., 2011). Let the auxiliary feature function be

$$g(x, x', y) = -\ln p^A(y \mid x, x'),$$

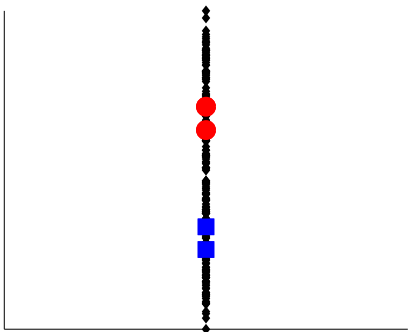
then maximizing $\mathcal{L}_2(A, \kappa)$ is equivalent to

$$\max_{A,\kappa} \ \mathcal{L}_1(A, \kappa) - \mu \sum\nolimits_{\mathcal{U}} \mathbb{E}_{p^A_{i,j}}[g(x_i, x_j, y)]. \tag{3.15}$$

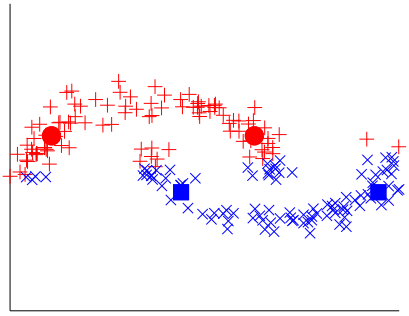


(a) Data set

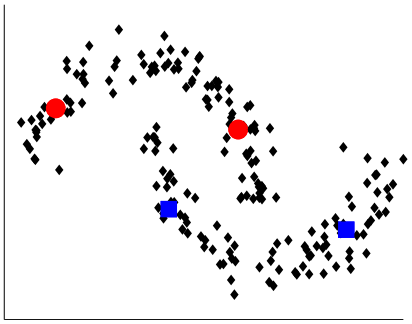
(b) Euclidean



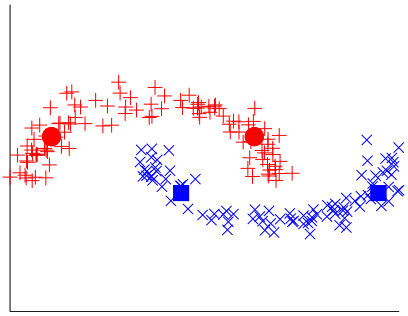
(c) Non-sparse



(d) Non-sparse



(e) Sparse



(f) Sparse

Figure 3.3: Sparse vs. non-sparse posterior distributions. In this example, six weak labels are constructed according to the four class labels. The left three panels show the original data and the projected data by metrics learned with/without the posterior sparsity. The right three panels exhibit one-nearest-neighbor classification results based on the Euclidean distance and two learned metrics.

On the other hand, according to optimization (7) of Graça et al. (2009), the soft posterior regularization objective should take a form as

$$\begin{aligned} \max_{A,\kappa} \ & \mathcal{L}_1(A,\kappa) - \min_q \Big(\mathrm{KL}(q \mid\mid p^A) + \mu \sum\nolimits_{\mathcal{U}} \xi_{i,j}\Big) \\ \text{s.t.} \ & \mathbb{E}_q[g(x_i,x_j,y)] \le \xi_{i,j}, \forall (x_i,x_j) \in \mathcal{U}, \end{aligned} \tag{3.16}$$

where $\xi_{i,j}$ are slack variables. Since q is unconstrained, we can optimize q with respect to fixed A and κ. It is easy to see that q should be p^A restricted on $\mathcal{U}$, so the KL divergence term is zero and the expectation term is the entropy, which implies the equivalence of optimizations (3.15) and (3.16).

Besides the posterior sparsity, we also hope for the projection sparsity, which may guide the learned metric to a better generalization performance. Figure 3.4 illustrates its effectiveness, where the horizontal feature is dominant and the vertical feature is uninformative.

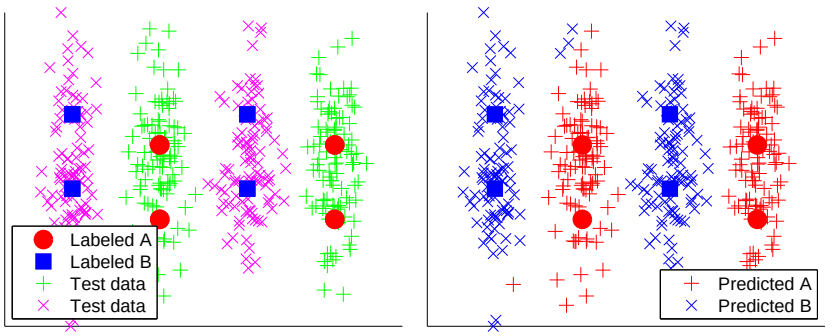
The underlying technique of the projection sparsity is the mixed-norm regularization (Argyriou et al., 2007) or group lasso (Yuan and Lin, 2006). Denote the $\ell_{(2,1)}$-norm of a symmetric matrix M as

$$\|M\|_{(2,1)} = \sum_{k=1}^{m} \left(\sum_{k'=1}^{m} M_{k,k'}^2 \right)^{1/2}.$$

Similarly to Ying et al. (2009), let $P \in \mathbb{R}^{m\times m}$ be a projection, and $W = P^\top P$ be the metric induced from P. Let P_i and W_i be the i-th column of P and W. If P_i is identically zero, the i-th component of x has no contribution to $z = Px$. Since the column-wise sparsity of W and P are equivalent, we can penalize $\|W\|_{(2,1)}$ to reach the column-wise sparsity of P.

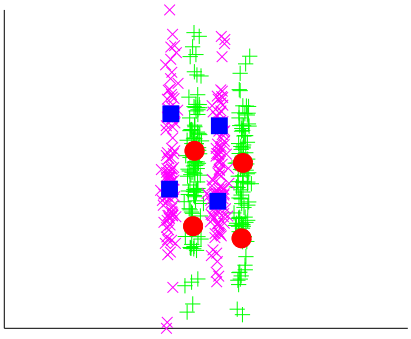
Nevertheless, this is the ability of feature selection rather than dimensionality reduction. Recall that the goal is to select a few most representative directions of input data which are not restricted to the coordinate axes. The solution is to pick an extra transformation $V \in \mathcal{O}^m$ to rotate x before the projection where $\mathcal{O}^m$ is the set of orthonormal matrices of size m, and add V to the optimization variables. Consequently, we penalize $\|W\|_{(2,1)}$, project x to $z = PVx$, and since $A = (PV)^\top(PV) = V^\top W V$, we arrive at

$$\begin{aligned} \max_{A,\kappa,W,V} \ & \mathcal{L}_2(A,\kappa) - \lambda \|W\|_{(2,1)} \\ \text{s.t.} \ & A = V^\top W V, W = W^\top, W \succeq 0, V \in \mathcal{O}^m. \end{aligned} \tag{3.17}$$

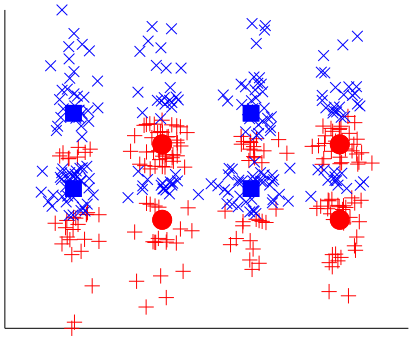


(a) Data set

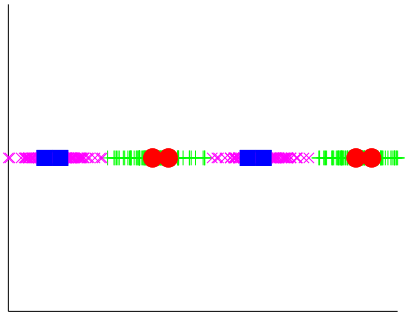
(b) Euclidean



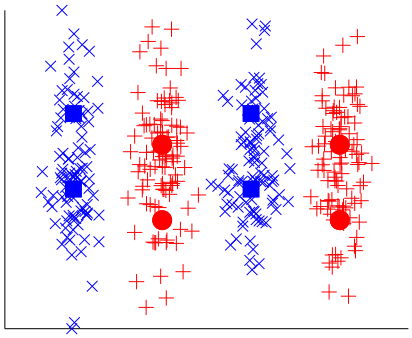
(c) Non-sparse



(d) Non-sparse



(e) Sparse



(f) Sparse

Figure 3.4: Sparse vs. non-sparse projections. In this example, twenty-eight weak labels are constructed according to the eight class labels. The left three panels show the original data and the projected data by metrics learned with/without the projection sparsity. The right three panels exhibit one-nearest-neighbor classification results based on the Euclidean distance and two learned metrics.

Remember that the final model of SERAPH is given by optimization (3.7) as

$$\max_{A,\kappa} \ \mathcal{L}_2(A,\kappa) - \lambda \operatorname{tr}(A).$$

The equivalence of optimizations (3.7) and (3.17) is guaranteed by Lemma 1 of Ying et al. (2009). By unifying the posterior sparsity and the projection sparsity mentioned above, we obtain a property that we call the *hyper-sparsity*.

3.4.2 Generalized Maximum Entropy Principle

The basic model defined in optimization (3.2) contains an inequality constraint instead of an equality constraint, since the regularization term $-\gamma\kappa^2/2$ is indispensable for $\mathcal{L}_1(A,\kappa)$. Otherwise, we will have $\kappa^* = +\infty$ for the optimal solution (A^*,κ^*). In other words, the optimization will be degenerated, and the learned metric may easily overfit weakly labeled training data. This phenomenon is owing to the single-point prior of the expected data moment from the empirical data moment. An ℓ_2-regularization on the dual variable reflects the Gaussian prior of the expected data moment from the empirical data moment according to the generalized maximum entropy principle (Dudík and Schapire, 2006), while the ordinary maximum entropy principle (Jaynes, 1957; Berger et al., 1996) assumes the single-point prior and applies no regularization onto the dual variable.

The potential function underlies the generalized maximum entropy estimation. By the potential function and the slack variable, we could obtain the same dual problem for ℓ_2-regularization but different dual problems for ℓ_1-regularization on the gap of expected data moment and empirical data moment.

Let the potential function $U_f(\cdot)$ and its target value u_f be

$$\begin{aligned} U_f(x) &= \frac{1}{2\gamma}(x-u_f)^2, \\ u_f &= \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} f(x_i,x_j,y_{i,j}). \end{aligned}$$

Redefine optimization (3.2) as an equivalent form

$$\max_{A,p_{i,j}^A} \ \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} H(p_{i,j}^A) - U_f\left(\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \mathbb{E}_{p_{i,j}^A}[f(x_i,x_j,y)]\right),$$

where the equivalence is due to *Fenchel's Duality Theorem* of Dudík and Schapire (2006) plus the fact that the conjugate of $U_f(x)$ is $U_f^*(\kappa) = \gamma\kappa^2/2$. Subsequently,

$$\max_{A,\kappa} \mathcal{L}_2(A,\kappa) = \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln p_{i,j}^A(y_{i,j}) - U_f^*(-\kappa) - \mu U_g\left(\sum\nolimits_{\mathcal{U}} \mathbb{E}_{p_{i,j}^A}[g(x_i,x_j,y)]\right)$$

is an optimization problem with two potential functions $U_f(\cdot)$ and $U_g(\cdot)$ under the posterior regularization framework (Graça et al., 2008, 2009; Bellare et al., 2009; Gillenwater et al., 2011), and thus SERAPH can be viewed as a semi-supervised maximum entropy estimation equipped with the additional projection sparsity.

3.4.3 Information Maximization Principle

The final model of SERAPH defined in optimization (3.7) can also be viewed as an information maximization approach for semi-supervised metric learning. The regularized information maximization framework (Gomes et al., 2010) follows the *information maximization principle*, and it advocates the preference for maximizing the mutual information between data and labels as well as the necessity of regularization on model parameters.

Let $p(y)$ be the prior distribution

$$p(y) = \iint_{\mathbb{R}^m\times\mathbb{R}^m} p^A(y \mid x,x')p(x)p(x')\mathrm{d}x\mathrm{d}x',$$

and $\hat{p}(y)$ be its estimate

$$\hat{p}(y) = \frac{1}{\#\mathcal{U}}\sum\nolimits_{\mathcal{U}} p_{i,j}^A(y).$$

Let $I(y;x,x')$ be the mutual information between the data pair and the weak label

$$I(y;x,x') = \iint_{\mathbb{R}^m\times\mathbb{R}^m} \sum\nolimits_y p^A(y \mid x,x')p(x)p(x')\ln\left(\frac{p^A(y \mid x,x')}{p(y)}\right)\mathrm{d}x\mathrm{d}x',$$

and $I(y;\mathcal{U})$ be its estimate, that is, the mutual information between unlabeled data and unobserved weak labels

$$I(y;\mathcal{U}) = \frac{1}{\#\mathcal{U}}\sum\nolimits_{\mathcal{U}}\sum\nolimits_y p_{i,j}^A(y)\ln\left(\frac{p_{i,j}^A(y)}{\hat{p}(y)}\right).$$

Given the supervised part of SERAPH, regularized information maximization would suggest

$$\max_{A,\kappa} \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \ln p_{i,j}^{A}(y_{i,j}) + \mu' I(y;\mathcal{U}) - \frac{\gamma}{2}\kappa^2 - \lambda \operatorname{tr}(A),$$

where we assume the regularization parameter μ' satisfies $\mu' = \#\mathcal{U}\mu$. By decomposing $I(y;\mathcal{U})$, it could be rewritten as

$$\max_{A,\kappa} \mathcal{L}(A,\kappa) + \mu' H(\hat{p}(y)).$$

The entropy term encourages a balanced prior distribution of y under the metric $d(x,x')$. However, the number of similar and dissimilar data pairs (i.e., $y=+1$ and $y=-1$) are inherently imbalanced in all metric learning problem settings. Therefore, we simply drop the regularization term $\mu' H(\hat{p}(y))$ and attain optimization (3.7).

3.5 Related Works

Xing et al. (2003) initiated the research of metric learning based on pairwise similarity and dissimilarity constraints by global distance metric learning (GDM). Inspired by miscellaneous motivations, several excellent metric learning methods have been developed in the last decade, such as neighborhood component analysis (NCA; Goldberger et al., 2005), large margin nearest neighbor classification (LMNN; Weinberger et al., 2006), information-theoretic metric learning (ITML; Davis et al., 2007), and so on.

Both ITML and SERAPH are information-theoretic, but the ideas and models are quite different. ITML defines a generative Gaussian model

$$p^A(x) = \frac{1}{Z}\exp\left(-\frac{1}{2}\|x-\mu\|_A^2\right),$$

where μ is the unknown mean value, Z is a normalizing constant, and both of them can be canceled out in the constrained optimization. Compared with GDM, ITML regularizes the Kullback-Leibler divergence between $p^{A_0}(x)$ and $p^A(x)$ where A_0 is the prior metric, and then transforms this term to a *Log-Det regularization*. By specifying $A_0 = I_m/n$, it becomes the maximum entropy estimation of $p^A(x)$. Thus, it prefers the distance metric close to the Euclidean distance. On the other

hand, the supervised part of SERAPH also follows the maximum entropy principle, but the probabilistic model $p^A(y \mid x, x')$ is discriminative.

A probabilistic GDM was designed intuitively as a baseline method in the experimental part of Yang et al. (2006). It can be viewed as a special case of our supervised part, but the final model of SERAPH is much more general. Please refer to Section 3.2.2 for details.

Due to the limitation of supervised metric learning when few labeled data are available, semi-supervised models and algorithms that incorporate off-the-shelf unsupervised techniques to existing supervised approaches have been proposed in recent years. Local distance metric learning (LDM; Yang et al., 2006) is the pioneer. Unlike later manifold-based methods, it embeds the unsupervised information by assuming that the eigenvectors of the optimal A are the principal components of all training data. Hoi et al. (2008) borrows the idea of Laplacian eigenmaps (Belkin and Niyogi, 2002) and combines manifold regularization to the min-max principle of GDM. Baghshah and Shouraki (2009) then shows that Fisher discriminant analysis can be regularized by locally linear embedding (Roweis and Saul, 2000), and the resulting manifold Fisher discriminant analysis (MFDA) is extremely computational-efficient. Liu et al. (2010) brings the element-wise matrix sparsity of A to Hoi et al. (2008). In principle, any unsupervised embedding method that preserves the local neighborhood information can be modified into a semi-supervised extension. Check *DistLearnKit*[6] for a partial list.

The manifold extension is so general that it can be attached to all metric learning methods, whereas our information-theoretic extension can only be applied to probabilistic metric learning methods. Nevertheless, any probabilistic method with an explicit expression of the posterior distribution such as NCA, LDM and SERAPH can have two semi-supervised extensions, while deterministic methods like GDM, LMNN and MFDA cannot benefit from our semi-supervised extension. ITML utilizes a generative Gaussian model whose parameters are not estimated by the algorithm, so it is non-trivial to apply our extension to it.

Notice that we leave out sparse metric learning and robust metric learning. Instead, we recommend Huang et al. (2009, pp. 8–9) and Huang et al. (2010, p. 2) for the reviews of sparse and robust metric learning respectively.

[6] A Matlab toolkit for distance metric learning: http://www.cs.cmu.edu/~liuy/distlearn.htm.

3.6 Experiments

In this section, we numerically evaluate the performance of metric learning algorithms.

3.6.1 Setup

We compared the proposed SERAPH with six representative metric learning algorithms (plus the Euclidean distance):

- Global distance metric learning (GDM; Xing et al., 2003)[7];
- Neighborhood component analysis (NCA; Goldberger et al., 2005)[8];
- Large margin nearest neighbor classification (LMNN; Weinberger et al., 2006)[9];
- Information-theoretic metric learning (ITML; Davis et al., 2007)[10];
- Local distance metric learning (LDM; Yang et al., 2006)[11];
- Manifold Fisher discriminant analysis (MFDA; Baghshah and Shouraki, 2009)[12].

GDM, NCA, LMNN and ITML are supervised methods, and LDM and MFDA are semi-supervised methods. SERAPH as well as GDM, ITML and LDM utilize the global metric information. On the other hand, NCA, LMNN and MFDA benefit from the local metric information.

Table 3.1 describes the specification of benchmark data sets in our experiments. The top six data sets (i.e., iris, wine, ionosphere, balance, breast cancer, and diabetes) come from the *UCI machine learning repository*[13], while *USPS* and *MNIST* come from the homepage of the late Sam Roweis[14]. The gray-scale images of handwritten digits in USPS are downsampled to 8×8 pixel resolution

[7] http://www.cs.cmu.edu/~epxing/papers/Old_papers/code_Metric_online.tar.gz.
[8] http://www.cs.berkeley.edu/~fowlkes/software/nca/nca_demo.tar.gz.
[9] http://www.cse.wustl.edu/~kilian/code/files/mLMNN.zip.
[10] http://www.cs.utexas.edu/~pjain/itml/download/itml-1.2.tar.gz.
[11] http://www.cs.cmu.edu/~liuy/ldm_scripts_2.zip.
[12] Implemented based on locally linear embedding, http://www.cs.nyu.edu/~roweis/lle/code.html.
[13] http://archive.ics.uci.edu/ml/.
[14] http://cs.nyu.edu/~roweis/data.html.

resulting in 64-dimensional vectors. Similarly, the gray-scale images in MNIST are downsampled to 14×14 pixel resolution resulting in 196-dimensional vectors. The symbol $\text{USPS}_{1-5,20}$ means 20 training data from each of the first 5 classes, $\text{USPS}_{1-10,40}$ means 40 training data from each of all 10 classes, $\text{MNIST}_{1,7}$ means digits 1 versus 7, and so forth. Note that in the last two tasks, the dimensionality of data is greater than the size of all training data.

All algorithms were repeatedly run on 50 random samplings of a given task. For each random sampling, we construct the sets of similar and dissimilar data pairs $\mathcal{S}$ and $\mathcal{D}$ according to the class labels of the first few training data. The sizes of $\mathcal{S}$ and $\mathcal{D}$ were dependent on the specific random sampling of each UCI task, but fixed for all samplings of each USPS and MNIST task. We measured the performance of the one-nearest-neighbor classifiers based on the learned metrics and the computation time for learning the metrics.

We fixed $\eta = 1$ for simplicity. Then, four settings of SERAPH were considered in our experiments (except on two artificial data sets):

- $\text{SERAPH}_{\text{none}}$ stands for $\mu = 0$ and $\lambda = 0$;
- $\text{SERAPH}_{\text{post}}$ stands for $\mu = \frac{\#(\mathcal{S} \cup \mathcal{D})}{\#\mathcal{U}}$ and $\lambda = 0$;
- $\text{SERAPH}_{\text{proj}}$ stands for $\mu = 0$ and $\lambda = 1$;
- $\text{SERAPH}_{\text{hyper}}$ stands for $\mu = \frac{\#(\mathcal{S} \cup \mathcal{D})}{\#\mathcal{U}}$ and $\lambda = 1$.

There was no cross-validation for each random sampling, because we would like the learned metrics of different algorithms to be independent of the nearest-neighbor classifiers whose performance had a large deviation given limited supervised information. The hyperparameters of other algorithms, e.g., the number of reduced dimensions, the number of nearest neighbors, and the percentage of principal components, were selected as the best value based on another 10 random samplings (which serve as additional validation data) if the default values or heuristics were not provided by the original authors.

3.6.2 Results

Figures 3.3 and 3.4 had already displayed the visually comprehensive results of the posterior and projection sparsity regularization on two artificial data sets respectively. Subfigures (c) and (d) in both figures were obtained by SERAPH with

	#classes	#features	#training	#test	#class labels	$\mathbb{E}\#\mathcal{S}$	$\mathbb{E}\#\mathcal{D}$	$\#\mathcal{U}$
iris	3	4	100	38	10	15.10	29.90	4905
wine	3	13	100	78	10	13.98	31.02	4905
ionosphere	2	34	100	251	20	97.50	92.50	4760
balance	3	4	100	465	10	20.38	24.62	4905
breast cancer	2	30	100	469	10	23.54	21.46	4905
diabetes	2	8	100	668	10	23.02	21.98	4905

	#classes	#features	#training	#test	#class labels	$\#\mathcal{S}$	$\#\mathcal{D}$	$\#\mathcal{U}$
$\text{USPS}_{1-5,20}$	5	64	100	2500	10	5	40	4905
$\text{USPS}_{1-5,40}$	5	64	200	2500	20	30	160	19710
$\text{USPS}_{1-10,20}$	10	64	200	2500	20	10	180	19710
$\text{USPS}_{1-10,40}$	10	64	400	2500	40	60	720	79020
$\text{MNIST}_{1,7}$	2	196	100	1000	4	2	4	4944
$\text{MNIST}_{3,5,8}$	3	196	150	1500	9	9	27	11139

Table 3.1: Specification of benchmark data sets

	iris	wine	ionosphere	balance	breast cancer	diabetes
EUCLIDEAN	9.58 ± 0.73	12.93 ± 0.83	23.60 ± 0.89	27.15 ± 0.75	14.11 ± 1.07	32.94 ± 0.65
GDM	8.95 ± 0.71	11.52 ± 0.77	$\mathbf{20.82 \pm 0.82}$	22.89 ± 1.08	11.86 ± 0.83	$\mathbf{30.73 \pm 0.59}$
NCA	10.32 ± 0.83	15.03 ± 1.12	26.68 ± 0.82	32.97 ± 1.31	14.63 ± 1.09	32.95 ± 0.65
LMNN	9.81 ± 0.79	14.83 ± 0.97	22.25 ± 0.75	24.00 ± 1.34	13.86 ± 0.84	32.02 ± 0.60
ITML	$\mathbf{5.57 \pm 0.53}$	$\mathbf{8.22 \pm 0.66}$	$\mathbf{20.35 \pm 0.64}$	22.04 ± 0.80	$\mathbf{9.60 \pm 0.49}$	$\mathbf{31.21 \pm 0.73}$
LDM	7.27 ± 0.72	17.21 ± 1.41	24.54 ± 0.92	$\mathbf{21.22 \pm 0.93}$	14.85 ± 0.92	34.33 ± 0.60
MFDA	6.58 ± 0.54	11.55 ± 1.03	23.66 ± 0.91	23.61 ± 1.00	11.21 ± 0.80	31.64 ± 0.62
SERAPH$_{\text{none}}$	6.21 ± 0.48	$\mathbf{8.13 \pm 0.58}$	$\mathbf{19.70 \pm 0.43}$	$\mathbf{20.25 \pm 0.64}$	11.39 ± 0.49	$\mathbf{29.86 \pm 0.61}$
SERAPH$_{\text{post}}$	$\mathbf{4.79 \pm 0.37}$	$\mathbf{7.46 \pm 0.51}$	$\mathbf{19.64 \pm 0.45}$	$\mathbf{19.98 \pm 0.67}$	11.33 ± 0.50	$\mathbf{29.87 \pm 0.57}$
SERAPH$_{\text{proj}}$	$\mathbf{5.79 \pm 0.54}$	$\mathbf{7.39 \pm 0.50}$	$\mathbf{19.53 \pm 0.46}$	$\mathbf{20.94 \pm 0.64}$	$\mathbf{9.61 \pm 0.49}$	$\mathbf{30.43 \pm 0.65}$
SERAPH$_{\text{hyper}}$	$\mathbf{5.31 \pm 0.43}$	$\mathbf{7.38 \pm 0.49}$	$\mathbf{19.33 \pm 0.42}$	$\mathbf{20.15 \pm 0.63}$	$\mathbf{10.04 \pm 0.52}$	$\mathbf{30.02 \pm 0.63}$

Table 3.2: Means with standard errors of the nearest-neighbor misclassification rate (in %) on UCI benchmarks. For each data set, the best algorithm and comparable ones based on the unpaired t-test at the significance level 5% are highlighted in boldface.

	$\text{USPS}_{1-5,20}$	$\text{USPS}_{1-5,40}$	$\text{USPS}_{1-10,20}$	$\text{USPS}_{1-10,40}$	$\text{MNIST}_{1,7}$	$\text{MNIST}_{3,5,8}$
EUCLIDEAN	36.63 ± 0.80	28.43 ± 0.60	49.17 ± 0.50	39.30 ± 0.39	10.42 ± 0.67	$\mathbf{37.30 \pm 0.81}$
GDM	37.62 ± 0.77	-	-	-	-	-
NCA	37.55 ± 0.84	28.39 ± 0.60	57.01 ± 0.82	49.21 ± 0.66	10.42 ± 0.67	$\mathbf{37.75 \pm 0.92}$
LMNN	36.43 ± 0.78	28.93 ± 0.61	48.12 ± 0.57	43.68 ± 0.58	9.99 ± 0.71	$\mathbf{36.49 \pm 0.82}$
ITML	35.86 ± 0.74	27.40 ± 0.65	47.40 ± 0.60	39.44 ± 0.57	9.94 ± 0.69	40.83 ± 0.93
LDM	47.19 ± 1.51	32.52 ± 0.85	59.13 ± 0.73	43.18 ± 0.53	14.54 ± 1.41	45.53 ± 1.16
MFDA	42.52 ± 0.82	28.82 ± 0.62	52.13 ± 0.59	37.78 ± 0.50	9.35 ± 0.72	42.39 ± 0.92
$\text{SERAPH}_{\text{none}}$	36.08 ± 0.75	27.41 ± 0.60	47.29 ± 0.58	38.36 ± 0.55	9.97 ± 0.71	$\mathbf{36.44 \pm 0.84}$
$\text{SERAPH}_{\text{post}}$	35.79 ± 0.75	27.37 ± 0.60	47.12 ± 0.58	38.20 ± 0.55	10.98 ± 0.79	$\mathbf{36.45 \pm 0.84}$
$\text{SERAPH}_{\text{proj}}$	36.01 ± 0.75	$\mathbf{26.17 \pm 0.57}$	47.42 ± 0.62	35.42 ± 0.54	9.28 ± 0.72	$\mathbf{36.55 \pm 0.80}$
$\text{SERAPH}_{\text{hyper}}$	$\mathbf{32.79 \pm 0.77}$	$\mathbf{25.26 \pm 0.56}$	$\mathbf{44.89 \pm 0.58}$	$\mathbf{33.41 \pm 0.47}$	$\mathbf{7.61 \pm 0.57}$	$\mathbf{35.71 \pm 0.84}$

Table 3.3: Means with standard errors of the nearest-neighbor misclassification rate (in %) on USPS and MNIST. For each data set, the best algorithm and comparable ones based on the unpaired t-test at the significance level 5% are highlighted in boldface.

$\mu = \lambda = 0$, while (e) and (f) were generated by SERAPH with $\mu = 10 \cdot \frac{\#(\mathcal{S} \cup \mathcal{D})}{\#\mathcal{U}}, \lambda = 0$ in Figure 3.3 and $\mu = 0, \lambda = 100$ in Figure 3.4. We can see from Figures 3.3 and 3.4 that the sparsity regularization can dramatically improve the supervised maximum entropy estimation.

The experimental results in terms of the nearest-neighbor misclassification rate are reported in Tables 3.2 and 3.3, where GDM was very slow for high-dimensional data and excluded from the comparison. SERAPH is fairly promising, especially the hyper-sparsity setting ($\mu = \frac{\#(\mathcal{S} \cup \mathcal{D})}{\#\mathcal{U}}$ and $\lambda = 1$). It was best or tie over all twelve tasks. It often outperformed other algorithms statistically significantly except ITML on six UCI data sets. $\text{SERAPH}_{\text{hyper}}$ was superior to all other algorithms on four USPS tasks including $\text{SERAPH}_{\text{post}}$ and $\text{SERAPH}_{\text{proj}}$. Moreover, it successfully improved the accuracy even on two ill-posed MNIST tasks, though the improvement was insignificant on $\text{MNIST}_{3,5,8}$. To sum up, SERAPH can reduce the risk of overfitting weakly labeled data with the help of unlabeled data, and hence our sparsity regularization would be reasonable and practical.

In vivid contrast with SERAPH that exhibited the nice generalization capability, supervised algorithms might learn a metric even worse than the Euclidean distance due to overfitting problems, especially NCA that optimized the expected leave-one-out classification error on a limited amount of labeled data. The powerful LMNN did not behave satisfyingly, since it was hardly fulfilled to find a lot of neighbors belonging to the same class within labeled data. ITML worked very well despite that it can only access weakly labeled data, but it became less useful for high-dimensional data. On the other hand, we observed that LDM might fail when the principal components of all training data were not close to the eigenvectors of the optimal target matrix, and MFDA might fail if the amount of training data cannot afford to recover the intrinsic geometric structure.

An observation is that the global metric learning often outperformed the local one in our experiments where the supervised information was insufficient, which is opposite to the phenomena observed in supervised metric learning problem settings. It indicates that the local metric learning tends to fit the neighborhood information exceedingly and suffers from overfitting problems, since the neighborhood information has a relatively large variance for a small amount of data.

Finally, we report the computation time of each algorithm (excluding GDM)

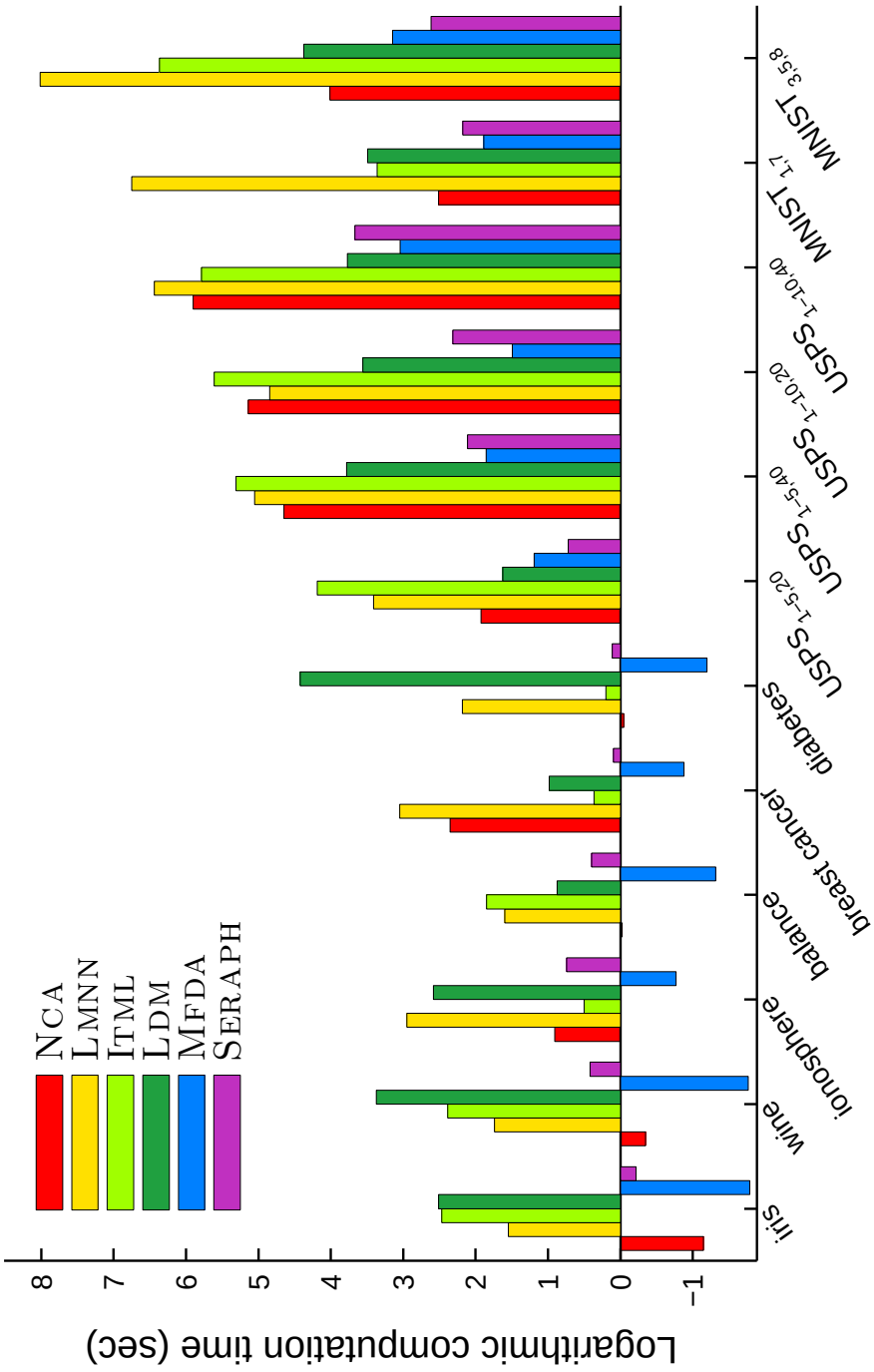


Figure 3.5: Computation time (per run) of different metric learning algorithms

on each task in Figure 3.5. Generally speaking, SERAPH was the second most computationally-efficient algorithm, and the most computationally-efficient algorithm MFDA consists of only solving a linear system in locally linear embedding (Roweis and Saul, 2000) and then a generalized eigenvalue problem as in Fisher discriminant analysis (Fisher, 1936). Improvements may be expected if we program in Matlab together with C/C++ as NCA and LMNN.

3.7 Proofs of Theoretical Results

3.7.1 Proof of Theorem 3.1

To simplify our notations and make the proof compact, let us denote

$$\begin{aligned}
p_{i,j}^{+} &\triangleq p_{i,j}^{A}(+1),\\
p_{i,j}^{-} &\triangleq p_{i,j}^{A}(-1),\\
f_{i,j}^{+} &\triangleq f(x_i, x_j, +1),\\
f_{i,j}^{-} &\triangleq f(x_i, x_j, -1),\\
\tilde{f}_{i,j} &\triangleq f(x_i, x_j, y_{i,j}),
\end{aligned}$$

respectively.

Foremost, expand optimization (3.2) into its complete form:

$$\begin{aligned}
\max_{A, p_{i,j}^{A}, \xi} \; & -\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^{+} \ln p_{i,j}^{+} + p_{i,j}^{-} \ln p_{i,j}^{-}) - \frac{1}{2\gamma}\xi^2\\
\text{s.t.} \; & \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^{+} f_{i,j}^{+} + p_{i,j}^{-} f_{i,j}^{-}) - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \tilde{f}_{i,j} - \xi \leq 0,\\
& \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \tilde{f}_{i,j} - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^{+} f_{i,j}^{+} + p_{i,j}^{-} f_{i,j}^{-}) - \xi \leq 0,\\
& p_{i,j}^{+} + p_{i,j}^{-} = 1, \forall (x_i, x_j) \in \mathcal{S}\cup\mathcal{D}.
\end{aligned}$$

The terms $\ln p_{i,j}^{+}$ and $\ln p_{i,j}^{-}$ in the objective function plus $p_{i,j}^{+} + p_{i,j}^{-} = 1$ in the constraints have already implied that

$$0 \leq p_{i,j}^{+}, p_{i,j}^{-} \leq 1.$$

By introducing dual variables $\kappa_1 \geq 0, \kappa_2 \geq 0$ for the first and second constraints,

and $\delta_{i,j} \in \mathbb{R}$ for the third group of constraints, the Lagrangian is expressed as

$$\begin{aligned} L(A, p_{i,j}^A, \xi, \kappa_1, \kappa_2, \delta_{i,j}) = & -\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^+ \ln p_{i,j}^+ + p_{i,j}^- \ln p_{i,j}^-) - \frac{1}{2\gamma}\xi^2 \\ & - \kappa_1 \left(\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^+ f_{i,j}^+ + p_{i,j}^- f_{i,j}^-) - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \tilde{f}_{i,j} - \xi \right) \\ & - \kappa_2 \left(\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \tilde{f}_{i,j} - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^+ f_{i,j}^+ + p_{i,j}^- f_{i,j}^-) - \xi \right) \\ & + \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \delta_{i,j} (p_{i,j}^+ + p_{i,j}^- - 1). \end{aligned}$$

Differentiating the function $L(A, p_{i,j}^A, \xi, \kappa_1, \kappa_2, \delta_{i,j})$ with respect to $p_{i,j}^+$ and $p_{i,j}^-$, and equating the derivatives to zero will give us

$$\begin{aligned} \ln p_{i,j}^+ &= \kappa f_{i,j}^+ + \delta_{i,j} - 1, \\ \ln p_{i,j}^- &= \kappa f_{i,j}^- + \delta_{i,j} - 1, \end{aligned} \tag{3.18}$$

where $\kappa = \kappa_2 - \kappa_1 \in \mathbb{R}$. Note that Eq. (3.18) says that

$$\frac{p_{i,j}^+}{p_{i,j}^-} = \exp(\kappa f_{i,j}^+ - \kappa f_{i,j}^-). \tag{3.19}$$

Hence Eq. (3.3) follows with

$$\delta_{i,j} = 1 - \ln Z_{i,j}^A. \tag{3.20}$$

Next, differentiating $L(A, p_{i,j}^A, \xi, \kappa_1, \kappa_2, \delta_{i,j})$ with respect to ξ and equating the derivative to zero will give us

$$\xi = \gamma(\kappa_1 + \kappa_2). \tag{3.21}$$

According to the Karush-Kuhn-Tucker condition about the dual complementary slackness, i.e.,

$$\begin{aligned} \kappa_1 \left(\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^+ f_{i,j}^+ + p_{i,j}^- f_{i,j}^-) - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \tilde{f}_{i,j} - \xi \right) = 0, \\ \kappa_2 \left(\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} \tilde{f}_{i,j} - \sum\nolimits_{\mathcal{S}\cup\mathcal{D}} (p_{i,j}^+ f_{i,j}^+ + p_{i,j}^- f_{i,j}^-) - \xi \right) = 0, \end{aligned}$$

we know that $\kappa_1 \kappa_2 = 0$, which means

$$(\kappa_1 + \kappa_2)^2 = (\kappa_1 - \kappa_2)^2 = \kappa^2. \tag{3.22}$$

Substituting Eq. (3.18)–Eq. (3.22) into $L(A, p_{i,j}^A, \xi, \kappa_1, \kappa_2, \delta_{i,j})$ accomplishes dual problem (3.4).

Finally, the optimization of the regularized maximum log-likelihood estimation is

$$\max_{A,\kappa} \ \mathcal{L}_1(A, \kappa).$$

By plugging the probabilistic model (3.3) into it we get optimization (3.4) exactly, which is the dual problem of the generalized maximum entropy estimation for supervised metric learning defined in optimization (3.2). □

3.7.2 Proof of Theorem 3.2

The proof is constructive.

As mentioned before, there must be $\kappa^* > 0$. Moreover, $\kappa^* < +\infty$ and $\mathrm{tr}(A^*) < +\infty$, since they are penalized in optimization (3.7). Let

$$\begin{aligned}\hat{A} &= \kappa^* A^*,\\ \hat{\eta} &= \kappa^* \eta,\\ \hat{\lambda} &= \lambda/\kappa^*.\end{aligned}$$

Then $\hat{\eta}$ and $\hat{\lambda}$ are well-defined hyperparameters as finite positive real numbers, and $\hat{A}$ is a feasible solution to (3.8) as a finite trace symmetric positive semi-definite matrix.

Differentiate p^A and $\hat{p}^A$ with respect to A,

$$\frac{\partial p^A}{\partial A} = \kappa y p^A (1 - p^A)(x - x')(x - x')^\top, \tag{3.23}$$

$$\frac{\partial \hat{p}^A}{\partial A} = -y \hat{p}^A (1 - \hat{p}^A)(x - x')(x - x')^\top. \tag{3.24}$$

Note that from

$$\hat{p}^A(y \mid x, x'; \hat{A}, \hat{\eta}) = p^A(y \mid x, x'; A^*, \kappa^*, \eta), \tag{3.25}$$

we have

$$\left.\frac{\partial \hat{\mathcal{L}}}{\partial \hat{p}_{i,j}^A}\right|_{A=\hat{A}} = \left.\frac{\partial \mathcal{L}}{\partial p_{i,j}^A}\right|_{A=A^*,\kappa=\kappa^*}.$$

Thus from

$$\frac{\partial \hat{p}^A}{\partial A}\Big|_{A=\hat{A}} = -\frac{1}{\kappa^*}\frac{\partial p^A}{\partial A}\Big|_{A=A^*,\kappa=\kappa^*},$$

$\partial \operatorname{tr}(A)/\partial A = I_m$ where I_m is the identity matrix, and the KKT stationarity condition of optimization (3.7)

$$\frac{\partial \mathcal{L}}{\partial A}\Big|_{A=A^*,\kappa=\kappa^*} = 0_{m\times m}$$

where $0_{m\times m}$ is the zero matrix in $\mathbb{R}^{m\times m}$, we get

$$\frac{\partial \hat{\mathcal{L}}}{\partial A}\Big|_{A=\hat{A}} = -\frac{1}{\kappa^*}\frac{\partial \mathcal{L}}{\partial A}\Big|_{A=A^*,\kappa=\kappa^*} = 0_{m\times m}.$$

This implies that $\hat{A}$ is a stationary point of $\hat{\mathcal{L}}(A)$.

Similarly, we could derive

$$\frac{\partial^2 \hat{\mathcal{L}}}{\partial A^2}\Big|_{A=\hat{A}} = \left(\frac{1}{\kappa^*}\right)^2 \frac{\partial^2 \mathcal{L}}{\partial A^2}\Big|_{A=A^*,\kappa=\kappa^*}.$$

Hence, $\partial_A^2 \hat{\mathcal{L}}(\hat{A}) \preceq 0$ if and only if $\partial_A^2 \mathcal{L}(A^*, \kappa^*) \preceq 0$, and $\hat{A}$ is actually a maximum of $\hat{\mathcal{L}}(A)$.

Remember Eq. (3.25) that $\hat{p}^A(y \mid x, x'; \hat{A}, \hat{\eta})$ is identical to $p^A(y \mid x, x'; A^*, \kappa^*, \eta)$. The theorem follows. □

3.7.3 Proof of Theorem 3.4

By the techniques used in the supplementary material of Graça et al. (2009), the dual problem of optimization (3.10) should be

$$\begin{aligned} \min_{\xi_{i,j}} \quad & \ln\left(\sum\nolimits_y p_{i,j}^A(y)\exp(\xi_{i,j}\ln p_{i,j}^A(y))\right) \\ \text{s.t.} \quad & 0 \le \xi_{i,j} \le \mu, \end{aligned}$$

where $\xi_{i,j}$ is the dual variable, and the primal variable can be recovered by

$$q(y \mid x_i, x_j) = \frac{p_{i,j}^A(y)\exp(\xi_{i,j}\ln p_{i,j}^A(y))}{\sum_{y'} p_{i,j}^A(y')\exp(\xi_{i,j}\ln p_{i,j}^A(y'))}.$$

The optimal $q(y \mid x_i, x_j)$ is given by

$$q(y \mid x_i, x_j) = \frac{p^A_{i,j}(y) \exp(\mu \ln p^A_{i,j}(y))}{\sum_{y'} p^A_{i,j}(y') \exp(\mu \ln p^A_{i,j}(y'))},$$

since the objective of the dual problem is monotonically decreasing with respect to $\xi_{i,j}$.

However, we present here a shorter and direct proof to get Eq. (3.11) for the sake of self-containing.

As before, let us denote

$$\begin{aligned} p^+_{i,j} &\triangleq p^A_{i,j}(+1), \\ p^-_{i,j} &\triangleq p^A_{i,j}(-1), \\ q^+_{i,j} &\triangleq q(+1 \mid x_i, x_j), \\ q^-_{i,j} &\triangleq q(-1 \mid x_i, x_j), \end{aligned}$$

respectively. We expand optimization (3.10) to its complete form:

$$\begin{aligned} \min_{q_{i,j}} \;& q^+_{i,j} \ln \frac{q^+_{i,j}}{p^+_{i,j}} + q^-_{i,j} \ln \frac{q^-_{i,j}}{p^-_{i,j}} - \mu q^+_{i,j} \ln p^+_{i,j} - \mu q^-_{i,j} \ln p^-_{i,j} \\ \text{s.t.} \;& q^+_{i,j} + q^-_{i,j} = 1. \end{aligned}$$

The terms $\ln(q^+_{i,j}/p^+_{i,j})$ and $\ln(q^-_{i,j}/p^-_{i,j})$ in the objective function plus $q^+_{i,j} + q^-_{i,j} = 1$ in the constraints have already implied that

$$0 \le q^+_{i,j}, q^-_{i,j} \le 1.$$

By introducing a dual variable $\xi_{i,j}$, the Lagrangian is expressed as

$$L(q_{i,j}, \xi_{i,j}) = q^+_{i,j} \ln \frac{q^+_{i,j}}{p^+_{i,j}} + q^-_{i,j} \ln \frac{q^-_{i,j}}{p^-_{i,j}} - \mu q^+_{i,j} \ln p^+_{i,j} - \mu q^-_{i,j} \ln p^-_{i,j} + \xi_{i,j}(q^+_{i,j} + q^-_{i,j} - 1).$$

Differentiate the function $L(q_{i,j}, \xi_{i,j})$ with respect to $q^+_{i,j}$ and $q^-_{i,j}$, equate the derivatives to zero, and then we get

$$\begin{aligned} \ln q^+_{i,j} &= \ln p^+_{i,j} + \mu \ln p^+_{i,j} - 1 - \xi_{i,j}, \\ \ln q^-_{i,j} &= \ln p^-_{i,j} + \mu \ln p^-_{i,j} - 1 - \xi_{i,j}, \end{aligned}$$

which says that

$$\frac{q^+_{i,j}}{q^-_{i,j}} = \frac{p^+_{i,j}}{p^-_{i,j}} \exp(\mu \ln p^+_{i,j} - \mu \ln p^-_{i,j}).$$

The analytical solution defined in Eq. (3.11) follows after the normalization. □

3.7.4 Proof of Theorem 3.5

Obviously $\mathcal{F}(A)$ is differentiable as long as we allow unbounded derivatives. Now we prove that the derivative is uniformly bounded for fixed training set $\mathcal{X}$.

The conjugate matrix norm of the Frobenius norm is still the Frobenius norm, namely,

$$\|B\|_F^* = \max_{\|A\|_F \le 1} \operatorname{tr}(A^\top B) = \|B\|_F.$$

Then the best Lipschitz constant of $\mathcal{F}$ with respect to $\|\cdot\|_F$ can be expressed as

$$\operatorname{Lip}_{\|\cdot\|_F}(\mathcal{F}) = \sup_{A \succeq 0} \|\nabla \mathcal{F}\|_F,$$

so it is sufficient to bound $\|(\partial \mathcal{F}/\partial p_{i,j}^A)(\partial p_{i,j}^A/\partial A)\|_F$ from above uniformly.

Recall that the partial derivative of the simplified p^A with respect to A was given by Eq. (3.24) as

$$\frac{\partial p^A}{\partial A} = -y p^A (1 - p^A)(x - x')(x - x')^\top.$$

On the other hand,

$$\frac{\partial \mathcal{F}}{\partial p_{i,j}^A} = \begin{cases} \dfrac{1}{p_{i,j}^A(y_{i,j})} & \text{if } (x_i, x_j) \in \mathcal{S} \cup \mathcal{D} \\ \dfrac{\mu q(y \mid x_i, x_j)}{p_{i,j}^A(y)} & \text{if } (x_i, x_j) \in \mathcal{U}, y \in \{1, -1\}. \end{cases}$$

Hence when $(x_i, x_j) \in \mathcal{S} \cup \mathcal{D}$,

$$\begin{aligned} \left\| \frac{\partial \mathcal{F}}{\partial p_{i,j}^A} \cdot \frac{\partial p_{i,j}^A}{\partial A} \right\|_F &= \| - y_{i,j}(1 - p_{i,j}^A(y_{i,j}))(x_i - x_j)(x_i - x_j)^\top \|_F \\ &\le \|(x_i - x_j)(x_i - x_j)^\top\|_F \\ &= \|x_i - x_j\|_2^2 \\ &\le (\operatorname{diam}(\mathcal{X}))^2, \end{aligned}$$

where we use the fact that

$$\begin{aligned} \|zz^\top\|_F^2 &= \sum_{i,j=1}^m (z_i z_j)^2 \\ &= \left(\sum_{i=1}^m z_i^2 \right) \left(\sum_{j=1}^m z_j^2 \right) \\ &= \|z\|_2^4. \end{aligned}$$

Similarly, we have that when $(x_i, x_j) \in \mathcal{U}$ for fixed y,

$$\left\| \frac{\partial \mathcal{F}}{\partial p^A_{i,j}} \cdot \frac{\partial p^A_{i,j}}{\partial A} \right\|_F \leq \mu q(y \mid x_i, x_j)(\operatorname{diam}(\mathcal{X}))^2,$$

and thus

$$\sum\nolimits_y \left\| \frac{\partial \mathcal{F}}{\partial p^A_{i,j}} \cdot \frac{\partial p^A_{i,j}}{\partial A} \right\|_F \leq \mu(\operatorname{diam}(\mathcal{X}))^2.$$

As a consequence, there exists a finite $\operatorname{Lip}_{\|\cdot\|_F}(\mathcal{F})$. The inequality (3.14) is obtained by applying the triangle inequality of the Frobenius norm. □

Chapter 4

Squared-loss Mutual Information Regularization

In this chapter, we present squared-loss mutual information regularization (SMIR) which is a novel information-theoretic approach to semi-supervised learning. Our contributions can be summarized as three folds.

- We apply the squared-loss mutual information (SMI) (Suzuki et al., 2009) to semi-supervised classification;
- We prove that SMIR is convex under mild conditions, which improves the non-convexity of mutual information regularization;
- Novel data-dependent generalization error bounds are derived.

This chapter is organized as follows. Sections 4.1 and 4.2 include the background and preliminaries. In Section 4.3, we propose SMIR. In Section 4.4, we discuss the data-dependent generalization error bounds. Related works are compared in Section 4.5. Experimental results are reported in Section 4.6.

4.1 Introduction

Semi-supervised learning, which utilizes both labeled and unlabeled data for training, has attracted considerable attention over the last decade in machine learning and data mining communities. Additional assumptions about the joint distribution

of the data and class labels have been made under semi-supervised settings in order to extract certain helpful information from unlabeled data. There are currently two such assumptions of vital importance in semi-supervised learning:

- The *cluster assumption* (Chapelle et al., 2003) assumes that data form discrete clusters, and then data from the same cluster are more likely to have the same class label.
- The *manifold assumption* (Belkin et al., 2006) assumes that data lie on an intrinsic geometric structure of much lower dimensionality than the input space. This structure of data is called the data manifold. Then this assumption suggests that data from the same region of the data manifold are more likely to have the same class label, and thus gives the priority to decision boundaries which are smooth over the data manifold.

These assumptions both are specialized versions of the *smoothness assumption* and originate from the *low-density separation principle*, which advocates that decision boundaries should pass preferentially through the low-density regions where training data hardly appear.

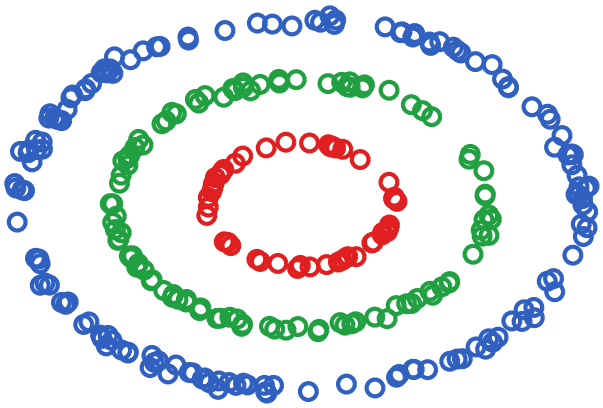
However, the low-density separation principle is not the only direction to go. In this chapter, we propose a novel information-theoretic approach to semi-supervised learning following the *information maximization principle* (IMP).

IMP comes from information maximization clustering (Agakov and Barber, 2007; Gomes et al., 2010; Sugiyama et al., 2011), in which probabilistic classifiers are trained in an unsupervised manner so that a given information measure (e.g., the mutual information) between input data and output cluster assignments is maximized. These clustering methods have shown that IMP is reasonable and powerful and hence can be a useful alternative to the low-density separation principle. Please refer to Figure 4.1 as an illustrative example of IMP. In Figure 4.1, the cluster assignments in panel (a) are intuitively consistent with the cluster structure of data and the corresponding estimated MI is 1.01 nat. On the other hand, the cluster assignments in panel (b) are independent of the cluster structure of data and the corresponding estimated MI is 0.02 nat. The information maximization principle advocates that the clustering shown in (a) is superior to the clustering in (b). Therefore, information maximization clustering which specifies the mutual

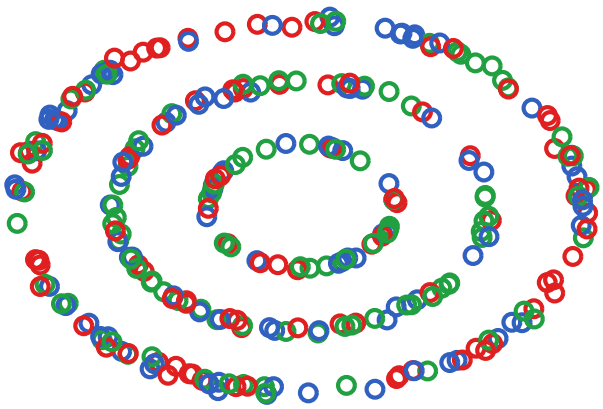
information as the information measure to be maximized prefers the clustering in (a) to (b).

More specifically, in our work, the *squared-loss mutual information* (SMI) (Suzuki et al., 2009) is designated as the information measure to be maximized in IMP. Then, we introduce an unsupervised SMI approximator with no logarithm inside similarly to Sugiyama et al. (2011), and propose the basic model of *SMI regularization* (SMIR). Unlike maximizing the mutual information, SMIR can be strongly convex under two mild conditions, such that the unique globally optimal solution is accessible (see Theorem 4.1). Albeit we can employ any convex loss in principle, SMIR can get rid of the logarithm completely in the involved optimization problem if we use the *squared difference of two probabilities* (also known as the *least-squares posterior fitting*) (Sugiyama et al., 2010), and then it guarantees the analytic expression of the globally optimal solution. Furthermore, SMIR aims at learning *multi-class probabilistic classifiers* that possess the innate ability of multi-class classification with the probabilistic output, and no reduction from the multi-class case to the binary case (cf. Allwein et al., 2000) is needed. These classifiers are inductive models, and thus they can naturally handle unseen data and no explicit out-of-sample extension is required. To the best of our knowledge, SMIR is the unique framework up to the present which leads to semi-supervised algorithms equipped with all aforementioned properties.

In addition, we establish two *data-dependent generalization error bounds* for a reduced SMIR algorithm based on the theory of *Rademacher averages* (Bartlett and Mendelson, 2002). Our error bounds are able to not only consider labeled data but also incorporate the information of unlabeled data. As a consequence, they can reflect certain properties of the particular mechanism generating the data more comprehensively than previous supervised bounds. Thanks to the analytical solution to the involved optimization, our error bounds also have closed-form expression, even though they depend upon the data in terms of Rademacher complexities (see Theorem 4.4). Notice that previous error bounds for semi-supervised learning or transductive learning such as Belkin et al. (2004) and Cortes et al. (2008) just focus on the regression error rather than the classification error, and hence none of semi-supervised or transductive algorithms hitherto have been provided similar theoretical results.



(a) High estimated MI implies good clustering



(b) Low estimated MI implies bad clustering

Figure 4.1: Illustration of high vs. low mutual information (MI) estimated from data in information maximization clustering. In this figure, the cluster assignments in panel (a) are intuitively consistent with the cluster structure of data and the corresponding estimated MI is 1.01 nat. On the other hand, the cluster assignments in panel (b) are independent of the cluster structure of data and the corresponding estimated MI is 0.02 nat. The information maximization principle advocates that the clustering shown in (a) is superior to the clustering in (b). Therefore, information maximization clustering which specifies the mutual information as the information measure to be maximized prefers the clustering in (a) to (b).

4.2 Preliminaries

In this section, we present the preliminaries to squared-loss mutual information regularization.

4.2.1 Problem Setting

Let $\mathcal{X}$ be the input domain, and most often but not necessarily $\mathcal{X} \subset \mathbb{R}^d$ for some natural number d. Let $\mathcal{Y} = \{1, \ldots, c\}$ be the set of class labels for some natural number $c \geq 2$. Assume that $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ are two random variables, and $\mathcal{X} \times \mathcal{Y}$ is furnished with an underlying joint probability distribution $p(x, y)$ with the marginal density $p(x)$ and the marginal probability $p(y)$. Without loss of generality, assume that $p(x)$ is strictly positive over $\mathcal{X}$.

Suppose that we are given l independent and identically distributed labeled data

$$\{(x_1, y_1), \ldots, (x_l, y_l)\} \sim p(x, y),$$

and u independent and identically distributed unlabeled data

$$\{x_{l+1}, \ldots, x_n\} \sim p(x),$$

where $n = l + u$ and typically $l \ll u$. The goal is to estimate the class-posterior probability

$$\hat{p}(y \mid x) \approx p(y \mid x) = \frac{p(x, y)}{p(x)}.$$

Then, we can classify any $x \in \mathcal{X}$ to

$$\hat{y} = \arg\max_{y \in \mathcal{Y}} \hat{p}(y \mid x) \tag{4.1}$$

with confidence $\hat{p}(\hat{y} \mid x)$.

4.2.2 Unsupervised SMI Approximator

Let us review the unsupervised approximator of *squared-loss mutual information* (SMI) proposed in Sugiyama et al. (2011).

As an information measure, SMI (Suzuki et al., 2009) between two random variables X and Y is defined by

$$\mathrm{SMI} := \frac{1}{2}\int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} p(x)p(y)\left(\frac{p(x,y)}{p(x)p(y)} - 1\right)^2 \mathrm{d}x. \tag{4.2}$$

Note that SMI in (4.2) is the *Pearson divergence* (Pearson, 1900) from $p(x,y)$ to $p(x)p(y)$, whereas the ordinary *mutual information* (MI) (Shannon, 1948a) between two random variables X and Y is the *Kullback-Leibler divergence* (Kullback and Leibler, 1951) from $p(x,y)$ to $p(x)p(y)$:

$$\mathrm{MI} := \int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)}\mathrm{d}x.$$

The Pearson and Kullback-Leibler divergences both belong to the family of Ali-Silvey-Csiszár divergences (which is also known as f-divergences, see Ali and Silvey, 1966; Csiszár, 1967), so they share similar properties. For example, SMI is nonnegative, and takes zero if and only if X and Y are statistically independent, as the ordinary MI.

In order to apply SMI to information maximization clustering, a computationally-efficient unsupervised SMI approximator was proposed in Sugiyama et al. (2011) as follows. By assuming a uniform class-prior probability

$$p(y) = 1/c,$$

SMI in (4.2) becomes

$$\begin{aligned}
\mathrm{SMI} &= \frac{1}{2}\int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} p(x)p(y)\left(\frac{p(x,y)}{p(x)p(y)}\right)^2 \mathrm{d}x - \frac{1}{2} \\
&= \frac{1}{2}\int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} p(x)\frac{(p(y\mid x))^2}{p(y)}\mathrm{d}x - \frac{1}{2} \\
&= \frac{c}{2}\int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} (p(y\mid x))^2 p(x)\mathrm{d}x - \frac{1}{2}.
\end{aligned} \tag{4.3}$$

Actually, SMI in (4.3) is an expectation of $\sum_{y\in\mathcal{Y}}(p(y\mid x))^2$ with respect to $p(x)$, i.e.,

$$\mathrm{SMI} = \frac{c}{2}\cdot\mathbb{E}_x\left[\sum\nolimits_{y\in\mathcal{Y}}(p(y\mid x))^2\right] - \frac{1}{2}. \tag{4.4}$$

Then, $p(y\mid x)$ is approximated by a kernel model:

$$q(y\mid x;\boldsymbol{\alpha}) := \sum_{i=1}^{n}\alpha_{y,i}k(x,x_i), \tag{4.5}$$

where $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_c\}$ and $\boldsymbol{\alpha}_y = (\alpha_{y,1}, \ldots, \alpha_{y,n})^\top$ are model parameters, and $k : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ is a kernel function. After approximating the expectation of $\sum_{y \in \mathcal{Y}} (p(y \mid x))^2$ with respect to $p(x)$ in Eq. (4.4) by the empirical average of $\sum_{y \in \mathcal{Y}} (q(y \mid x_i; \boldsymbol{\alpha}))^2$, an SMI approximator is derived as

$$\widehat{\mathrm{SMI}} = \frac{c}{2n} \sum\nolimits_{y \in \mathcal{Y}} \boldsymbol{\alpha}_y^\top K^2 \boldsymbol{\alpha}_y - \frac{1}{2},$$

where $K \in \mathbb{R}^{n \times n}$ is the kernel matrix defined as $K_{i,j} = k(x_i, x_j)$.

4.3 Squared-loss Mutual Information Regularization

In this section, we propose squared-loss mutual information regularization (SMIR).

4.3.1 Alternative SMI Approximator

Instead of $q(y \mid x; \boldsymbol{\alpha})$ defined in Eq. (4.5), we introduce our alternative kernel model and alternative SMI approximator for SMIR. The reason will be explained in Remark 4.2.

Let the empirical kernel map (Schölkopf and Smola, 2001, p. 42) be

$$\begin{aligned} \Phi_n : \mathcal{X} &\mapsto \mathbb{R}^n \\ x &\mapsto (k(x, x_1), \ldots, k(x, x_n))^\top, \end{aligned}$$

the degree of x_i be

$$d_i = \sum_{j=1}^{n} k(x_i, x_j),$$

and the degree matrix be

$$D = \mathrm{diag}(d_1, \ldots, d_n).^1$$

[1]The kernel matrix K is regarded as a similarity matrix, i.e., the weighted adjacency matrix of a similarity graph.

Here, we propose to approximate the class-posterior probability $p(y \mid x)$ by[2]

$$q(y \mid x; \boldsymbol{\alpha}) := \langle K^{-1/2}\Phi_n(x), D^{-1/2}\boldsymbol{\alpha}_y \rangle, \tag{4.6}$$

where $\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle = \sum_{j=1}^{n} \alpha_j \beta_j$ means the inner product of vectors in $\mathbb{R}^n$. In Eq. (4.6), we make use of the kernel PCA map $K^{-1/2}\Phi_n(x)$ (Schölkopf and Smola, 2001, p. 43), and balance the influence of high-density and low-density regions by dividing the parameter $\alpha_{y,i}$ associated with each x_i by its degree $\sqrt{d_i}$.

As a result,

$$q(y \mid x_i; \boldsymbol{\alpha}) = \boldsymbol{k}_i^\top K^{-1/2} D^{-1/2} \boldsymbol{\alpha}_y, \tag{4.7}$$

where $\boldsymbol{k}_i$ is the i-th column of the kernel matrix K. Approximating the expectation in Eq. (4.4) and plugging Eq. (4.7) into the corresponding average gives us an alternative SMI approximator:

$$\widehat{\mathrm{SMI}} = \frac{c}{2n} \sum\nolimits_{y \in \mathcal{Y}} \boldsymbol{\alpha}_y^\top D^{-1/2} K D^{-1/2} \boldsymbol{\alpha}_y - \frac{1}{2},$$

or equivalently,

$$\widehat{\mathrm{SMI}} = \frac{c}{2n} \operatorname{tr}\left(A^\top D^{-1/2} K D^{-1/2} A\right) - \frac{1}{2}, \tag{4.8}$$

where $A = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_c) \in \mathbb{R}^{n \times c}$ is the matrix representation of model parameters.

4.3.2 Basic Model

In our basic model, we employ $\widehat{\mathrm{SMI}}$ in (4.8) to regularize a loss function $\Delta(p, q)$ that is convex with respect to $q(y \mid x; \boldsymbol{\alpha})$ for $y = 1, \ldots, c$. More specifically, we have three objectives:

- To minimize the loss function $\Delta(p, q)$;
- To maximize the approximator $\widehat{\mathrm{SMI}}$; and
- To regularize the parameters $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_c$.

[2] We assume that the kernel matrix K is full-rank, and then $K^{-1/2}$ is a well-defined symmetric and positive definite matrix. The Gaussian kernel matrix is full-rank as long as $\forall i \neq j, x_i \neq x_j$. As we will see, $K^{1/2}$ is enough for classifying training data, and $K^{-1/2}$ will not be used until classifying unseen data.

Therefore, we formulate the optimization problem of SMIR as

$$\min_{\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_c\in\mathbb{R}^n} \Delta(p,q) - \gamma\widehat{\mathrm{SMI}} + \lambda\sum\nolimits_{y\in\mathcal{Y}} \frac{1}{2}\|\boldsymbol{\alpha}_y\|_2^2, \tag{4.9}$$

where $\gamma > 0$ and $\lambda > 0$ are user-specified regularization parameters for a trade-off between those objectives, and $\|\cdot\|_2$ means the ℓ_2-norm of a vector.

A remarkable characteristic of optimization (4.9) is its convexity, as long as the kernel function k is nonnegative and $\lambda > \gamma c/n$:

Theorem 4.1. *Assume that $k : \mathcal{X}\times\mathcal{X}\mapsto\mathbb{R}_+$ and $\lambda > \gamma c/n$. Then optimization* (4.9) *is strongly convex with a strong convexity constant $(\lambda - \gamma c/n)$, and there exists a unique globally optimal solution.*[3]

Proof. Denote the *unnormalized graph Laplacian* by

$$L = D - K,$$

and the *normalized graph Laplacian* by

$$\mathcal{L} = D^{-1/2}LD^{-1/2} = I_n - D^{-1/2}KD^{-1/2},$$

where I_n is the identity matrix of size n (Chung, 1997). Optimization (4.9) can be rewritten as

$$\min_{\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_c\in\mathbb{R}^n} \Delta(p,q) + \gamma'\sum\nolimits_{y\in\mathcal{Y}} \boldsymbol{\alpha}_y^\top\mathcal{L}\boldsymbol{\alpha}_y + \lambda'\sum\nolimits_{y\in\mathcal{Y}} \frac{1}{2}\|\boldsymbol{\alpha}_y\|_2^2, \tag{4.10}$$

where $\gamma' = \gamma c/2n > 0$ and $\lambda' = \lambda - \gamma c/n > 0$ are new regularization parameters that depend on γ, λ, c and n. Notice that $\forall y\in\mathcal{Y}$,

$$\boldsymbol{\alpha}_y^\top\mathcal{L}\boldsymbol{\alpha}_y = \frac{1}{2}\sum_{i,j=1}^{n}\left(\frac{\alpha_{y,i}}{d_i} - \frac{\alpha_{y,j}}{d_j}\right)^2 K_{i,j},$$

and then the second term of (4.10) is convex since $K_{i,j}\geq 0$.

We know that the loss function $\Delta(p,q)$ is convex with respect to $q(y \mid x;\boldsymbol{\alpha})$, and $q(y \mid x;\boldsymbol{\alpha})$ is linear with respect to $\boldsymbol{\alpha}_y$, so $\Delta(p,q)$ is convex with respect to $\boldsymbol{\alpha}_1,\dots,\boldsymbol{\alpha}_c$. Moreover, the ℓ_2-norm is strongly convex with the strong convexity

[3] In the rest of this chapter, we will assume that k is nonnegative and $\lambda > \gamma c/n$ to guarantee the convexity of SMIR.

constant 2, and consequently the third term of (4.10) is strongly convex with a strong convexity constant $\lambda' > 0$. Therefore, the objective function of (4.10) (or of (4.9) equivalently) is strongly convex with the strong convexity constant λ', and there exists a unique globally optimal solution, since the strong convexity implies the strict convexity. □

Theorem 4.1 has an interesting implication: According to optimization (4.10), we smooth $\boldsymbol{\alpha}_y$ over the intrinsic geometric structure in SMIR. Despite that maximizing SMI and smoothing the model parameters are quite different in general, the effects would be similar if we regularize these model parameters at the same time.

Remark 4.2. We introduced the alternative kernel model Eq. (4.6) due to two reasons, one theoretical and one practical.

Firstly, any kernel model that is linear with respect to $\boldsymbol{\alpha}_y$ may be used to approximate $p(y \mid x)$ in principle, and the maximization of $\widehat{\mathrm{SMI}}$ alone is non-convex under all circumstances. However, optimization (4.9) will be convex sooner or later by increasing λ, since the convexity of minimizing $\sum_y \|\boldsymbol{\alpha}_y\|_2^2$ will dominate the non-convexity of maximizing $\widehat{\mathrm{SMI}}$ if λ is large enough. Hence, only λ above a certain threshold is acceptable in order to guarantee the convexity of SMIR:

- The threshold of the kernel model in Eq. (4.6) is $\gamma c/n$, which is data-independent.
- The threshold of the kernel model in Eq. (4.5) is $\|K\|_2^2 \cdot \gamma c/n$, where $\|K\|_2$ means the spectral norm (also known as the operator norm induced by the ℓ_2-norm) of K. It depends upon all the training data thoroughly and is usually much larger than $\gamma c/n$.

Secondly, we found that Eq. (4.6) experimentally outperformed Eq. (4.5).

4.3.3 Proposed Algorithm

Among popular divergence measures, we choose the squared difference of the probability $p(y \mid x)$ and its approximation $q(y \mid x; \boldsymbol{\alpha})$ as the loss function (Sugiyama et al., 2010; Kanamori et al., 2009):

$$\Delta_2(p, q) := \frac{1}{2} \int_{\mathcal{X}} \sum\nolimits_{y \in \mathcal{Y}} (p(y \mid x) - q(y \mid x; \boldsymbol{\alpha}))^2 p(x) \mathrm{d}x. \tag{4.11}$$

It enables the analytical solution and facilitates our future theoretical analysis a lot. By expanding (4.11), we have

$$\Delta_2(p,q) = \text{Const.} - \int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} q(y \mid x;\boldsymbol{\alpha}) p(x,y)\mathrm{d}x + \frac{1}{2}\int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} (q(y \mid x;\boldsymbol{\alpha}))^2 p(x)\mathrm{d}x,$$

where the first term only contains $p(y \mid x)$ and $p(x)$ so it is viewed as a constant. Note that

$$\int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} q(y \mid x;\boldsymbol{\alpha}) p(x,y)\mathrm{d}x = \mathbb{E}_{(x,y)}[q(y \mid x;\boldsymbol{\alpha})],$$
$$\int_{\mathcal{X}} \sum\nolimits_{y\in\mathcal{Y}} (q(y \mid x;\boldsymbol{\alpha}))^2 p(x)\mathrm{d}x = \mathbb{E}_x\left[\sum\nolimits_{y\in\mathcal{Y}} (q(y \mid x;\boldsymbol{\alpha}))^2\right].$$

We thus approximate the expectation of $q(y \mid x;\boldsymbol{\alpha})$ with respect to $p(x,y)$ and the expectation of $\sum_{y\in\mathcal{Y}}(q(y \mid x;\boldsymbol{\alpha}))^2$ with respect to $p(x)$ by their corresponding averages over l labeled data, and the empirical version of $\Delta_2(p,q)$ is given by

$$\widehat{\Delta}_2(p,q) = \text{Const.} - \frac{1}{l}\sum_{i=1}^{l} q(y_i \mid x_i;\boldsymbol{\alpha}) + \frac{1}{2l}\sum_{i=1}^{l}\sum_{y=1}^{c} (q(y \mid x_i;\boldsymbol{\alpha}))^2. \quad (4.12)$$

Let $Y \in \mathbb{R}^{l\times c}$ be the class indicator matrix such that $Y_{i,j} = 1$ if $y_i = j$ and $Y_{i,j} = 0$ otherwise, and $B = (I_l; \mathbf{0}_{u\times l}) \in \mathbb{R}^{n\times l}$ where $\mathbf{0}_{u\times l}$ is the all-zero matrix in $\mathbb{R}^{u\times l}$. Subsequently,

$$\begin{aligned}\sum_{i=1}^{l} q(y_i \mid x_i;\boldsymbol{\alpha}) &= \sum_{i=1}^{l} \boldsymbol{k}_i^\top K^{-1/2} D^{-1/2} \boldsymbol{\alpha}_{y_i} \\ &= \mathrm{tr}\left((KB)^\top K^{-1/2} D^{-1/2} (AY^\top)\right) \\ &= \mathrm{tr}\left(Y^\top B^\top K^{1/2} D^{-1/2} A\right).\end{aligned}$$

Similarly,

$$\begin{aligned}\sum_{i=1}^{l}\sum_{y=1}^{c} (q(y \mid x_i;\boldsymbol{\alpha}))^2 &= \sum_{i=1}^{l}\sum_{y=1}^{c} \boldsymbol{\alpha}_y^\top D^{-1/2} K^{-1/2} \boldsymbol{k}_i \boldsymbol{k}_i^\top K^{-1/2} D^{-1/2} \boldsymbol{\alpha}_y \\ &= \sum_{y=1}^{c} \boldsymbol{\alpha}_y^\top D^{-1/2} K^{-1/2} (KB)(KB)^\top K^{-1/2} D^{-1/2} \boldsymbol{\alpha}_y \\ &= \mathrm{tr}\left(A^\top D^{-1/2} K^{1/2} B B^\top K^{1/2} D^{-1/2} A\right).\end{aligned}$$

As a consequence, $\widehat{\Delta}_2(p,q)$ in (4.12) can be expressed by

$$\begin{aligned}\widehat{\Delta}_2(p,q) = \text{Const.} - \frac{1}{l}\operatorname{tr}\left(Y^\top B^\top K^{1/2} D^{-1/2} A\right) \\ + \frac{1}{2l}\operatorname{tr}\left(A^\top D^{-1/2} K^{1/2} B B^\top K^{1/2} D^{-1/2} A\right).\end{aligned} \tag{4.13}$$

Dropping the constant term in Eq. (4.13) and substituting it into optimization (4.9), we will get the following objective function:

$$\begin{aligned}\mathcal{F}(A) = -\frac{1}{l}\operatorname{tr}\left(Y^\top B^\top K^{1/2} D^{-1/2} A\right) \\ + \frac{1}{2l}\operatorname{tr}\left(A^\top D^{-1/2} K^{1/2} B B^\top K^{1/2} D^{-1/2} A\right) \\ - \frac{\gamma c}{2n}\operatorname{tr}\left(A^\top D^{-1/2} K D^{-1/2} A\right) + \frac{\lambda}{2}\operatorname{tr}\left(A^\top A\right).\end{aligned} \tag{4.14}$$

At last, by equating the gradient matrix $\nabla\mathcal{F}$ to the zero matrix, i.e.,

$$\begin{aligned}\mathbf{0}_{n\times c} = \nabla\mathcal{F}(A) = -\frac{1}{l} D^{-1/2} K^{1/2} B Y \\ + \frac{1}{l} D^{-1/2} K^{1/2} B B^\top K^{1/2} D^{-1/2} A \\ - \frac{\gamma c}{n} D^{-1/2} K D^{-1/2} A + \lambda A,\end{aligned}$$

and solving the above equation, we obtain the analytic expression of the globally optimal solution to unconstrained optimization problem (4.9):

$$\begin{aligned}A^*_{\mathcal{F}} = n\left(n D^{-1/2} K^{1/2} B B^\top K^{1/2} D^{-1/2} + \lambda n l I_n \right. \\ \left. -\gamma l c D^{-1/2} K D^{-1/2}\right)^{-1} D^{-1/2} K^{1/2} B Y.\end{aligned} \tag{4.15}$$

The optimal solution $\boldsymbol{\alpha}^*_y$ is then the y-th column of $A^*_{\mathcal{F}}$ for $y = 1, \ldots, c$.

Remark 4.3. Let μ_x be the underlying probability measure of $p(x)$. Since $\mathcal{Y}$ has a finite cardinality, a measure μ on $\mathcal{X} \times \mathcal{Y}$ such that $\mathrm{d}\mu = (1/c)\mathrm{d}\mu_x = (p(x)/c)\mathrm{d}x$ is uniquely defined as

$$\mu(x, E_y) = (\#E_y/c) \cdot \mu_x(x),$$

where $E_y \subseteq \mathcal{Y}$ is a collection on $\mathcal{Y}$ and $\#$ measures the cardinality of a set. Although this μ is the underlying probability measure of $p'(x,y) = p(x)/c$ rather

than $p(x, y)$, it enables us to construct a Lebesgue space $L^2(\mathcal{X} \times \mathcal{Y}, \mu)$ with the norm

$$\|f\|_{L^2} = \left(\int_{\mathcal{X}} \sum\nolimits_{y \in \mathcal{Y}} (f(x, y))^2 \mathrm{d}\mu \right)^{1/2}.$$

Recall that we have assumed that $p(x)$ is strictly positive over $\mathcal{X}$. If so, $L^2(\mathcal{X} \times \mathcal{Y}, \mu)$ is a normed vector space rather than a seminormed vector space of measurable functions, i.e., $\|f\|_{L^2} = 0$ if and only if $f(x, y)$ is identically zero. In this sense, the loss function $\Delta_2(p, q)$ in (4.11) can be rewritten as $(c/2)\|p - q\|_{L^2}^2$, which is the least-squares fitting of the true class-posterior probability $p(y \mid x)$ in $L^2(\mathcal{X} \times \mathcal{Y}, \mu)$.

4.3.4 Post-processing

Since $p(y)$ was set to be uniform, we have the following normalization condition

$$\frac{1}{n} \mathbf{1}_n^\top K^{1/2} D^{-1/2} \boldsymbol{\beta}_y \approx \frac{1}{c},$$

where $\mathbf{1}_n$ is the all-one vector in $\mathbb{R}^n$ and $\boldsymbol{\beta}_y$ is a normalized version of $\boldsymbol{\alpha}_y^*$ for $y = 1, \ldots, c$, so $\boldsymbol{\beta}_y$ should be

$$\boldsymbol{\beta}_y = \frac{n}{c} \cdot \frac{K^{-1/2} D^{-1/2} \boldsymbol{\alpha}_y^*}{\mathbf{1}_n^\top K^{1/2} D^{-1/2} \boldsymbol{\alpha}_y^*}.$$

However, we recommend to enforce the class-prior probability on all training data by

$$\boldsymbol{\beta}_y = n\pi_y \cdot \frac{K^{-1/2} D^{-1/2} \boldsymbol{\alpha}_y^*}{\mathbf{1}_n^\top K^{1/2} D^{-1/2} \boldsymbol{\alpha}_y^*}, \tag{4.16}$$

where π_y is an unbiased estimate of $p(y)$ based on labeled data. Or we can simply set

$$\boldsymbol{\beta}_y = K^{-1/2} D^{-1/2} \boldsymbol{\alpha}_y^* \tag{4.17}$$

without any normalization for model parameters.

In addition, probability estimates should be nonnegative, and thus we round up $\langle \Phi_n(x), \boldsymbol{\beta}_y \rangle$ to zero if it is negative for $y = 1, \ldots, c$ and normalize the results once more. Taking these issues into account, our final solution can be expressed as follows (cf. Yamada et al., 2011):

$$\hat{p}(y \mid x) = \frac{\max(0, \langle \Phi_n(x), \boldsymbol{\beta}_y \rangle)}{\sum_{y'=1}^{c} \max(0, \langle \Phi_n(x), \boldsymbol{\beta}_{y'} \rangle)}. \tag{4.18}$$

As mentioned before, $K^{-1/2}$ is necessary just for classifying unseen data, and $K^{1/2}$ is enough for classifying training data. Let

$$\hat{\boldsymbol{p}}_y = (\hat{p}(y \mid x_1), \ldots, \hat{p}(y \mid x_n))^\top$$

be the probability estimates for all training data concerning the y-th class, then

$$\hat{\boldsymbol{p}}_y = n\pi_y \cdot \frac{K^{1/2} D^{-1/2} \boldsymbol{\alpha}_y^*}{\mathbf{1}_n^\top K^{1/2} D^{-1/2} \boldsymbol{\alpha}_y^*},$$

where we normalize $\boldsymbol{\alpha}_y^*$ according to π_y as in Eq. (4.16). In this case, the kernel matrix K must be a positive semi-definite matrix instead of a positive definite matrix, and low-rank approximations of K are allowed for the computational efficiency.

Although $q(y \mid x; \boldsymbol{\alpha}^*)$ might be negative or unnormalized, Kanamori et al. (2012) implies that minimizing the loss function $\Delta_2(p, q)$ could achieve the optimal non-parametric convergence rate from $q(y \mid x; \boldsymbol{\alpha})$ to $p(y \mid x)$, and when we have enough labeled data $q(y \mid x; \boldsymbol{\alpha}^*)$ is automatically a probability (i.e., non-negative and normalized). Anyway, the post-processing should be performed in practice since we usually have limited labeled data in semi-supervised learning.

4.4 Generalization Error Bounds

In order to theoretically elucidate the generalization capability, we reduce SMIR to binary classification. Now, a class label y is coded as $+1$ or -1, a single vector $\boldsymbol{\alpha} \in \mathbb{R}^n$ is enough to construct a discriminative model, and we classify any $x \in \mathcal{X}$ to

$$\hat{y} = \mathrm{sign}(\langle K^{-1/2} \Phi_n(x), D^{-1/2} \boldsymbol{\alpha} \rangle). \tag{4.19}$$

Let $\boldsymbol{y} = (y_1, \ldots, y_l)^\top \in \mathbb{R}^l$ be the class indicator vector where $y_i \in \{-1, +1\}$. The objective function in (4.14) can be reduced to

$$\begin{aligned}\mathcal{F}(\boldsymbol{\alpha}) = &-\frac{1}{l} \boldsymbol{y}^\top B^\top K^{1/2} D^{-1/2} \boldsymbol{\alpha} \\ &+ \frac{1}{2l} \boldsymbol{\alpha}^\top D^{-1/2} K^{1/2} B B^\top K^{1/2} D^{-1/2} \boldsymbol{\alpha} \\ &- \frac{\gamma c}{2n} \boldsymbol{\alpha}^\top D^{-1/2} K D^{-1/2} \boldsymbol{\alpha} + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \boldsymbol{\alpha},\end{aligned}$$

and accordingly the optimal solution in (4.15) can be reduced to

$$\begin{aligned}\boldsymbol{\alpha}^*_{\mathcal{F}} = n\big(&nD^{-1/2}K^{1/2}BB^\top K^{1/2}D^{-1/2} + \lambda nlI_n \\ &-\gamma lcD^{-1/2}KD^{-1/2}\big)^{-1} D^{-1/2}K^{1/2}B\boldsymbol{y}.\end{aligned} \tag{4.20}$$

This solution is consistent with the one in Eq. (4.15) when $c = 2$. In binary classification, we have a positive class and a negative class, and both of Y and A have two columns. Denote the columns of Y by $\boldsymbol{y}_+$ and $\boldsymbol{y}_-$ and the columns of $A^*_{\mathcal{F}}$ by $\boldsymbol{\alpha}^*_+$ and $\boldsymbol{\alpha}^*_-$. As a result, (4.20) can be recovered from (4.15) using $\boldsymbol{y} = \boldsymbol{y}_+ - \boldsymbol{y}_-$ and $\boldsymbol{\alpha}^*_{\mathcal{F}} = \boldsymbol{\alpha}^*_+ - \boldsymbol{\alpha}^*_-$, and the classification rule in (4.19) is equivalent to the classification rule in (4.1) based on $\hat{p}(y \mid x)$ in (4.18) with $\boldsymbol{\beta}_y$ in (4.17). For convenience, we define the decision function as

$$f(x) = \langle K^{-1/2}\Phi_n(x), D^{-1/2}\boldsymbol{\alpha}^*_{\mathcal{F}}\rangle. \tag{4.21}$$

Let $\mathbb{E}[\cdot]$ and $\hat{\mathbb{E}}[\cdot]$ stand for the true expectation and the empirical expectation, $\ell(z)$ be the *indicator loss* such that

$$\ell(z) = \begin{cases} 0 & \text{if } z > 0, \\ 1 & \text{if } z \leq 0, \end{cases}$$

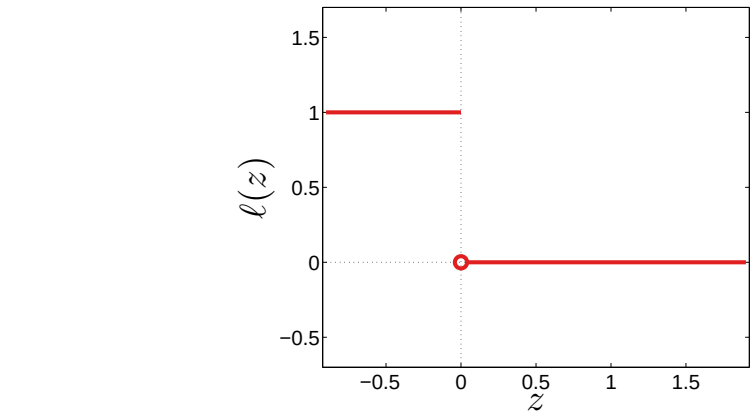
and $\ell_\eta(z)$ be the *surrogate loss* (Bartlett and Mendelson, 2002) such that

$$\ell_\eta(z) = \begin{cases} 0 & \text{if } z > \eta, \\ 1 - z/\eta & \text{if } 0 < z \leq \eta, \\ 1 & \text{if } z \leq 0, \end{cases}$$

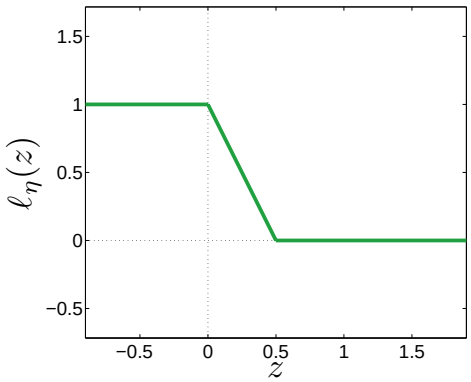
as illustrated in Figure 4.2. For any $\eta > 0$, $\ell_\eta(z)$ is lower bounded by $\ell(z)$ and approaches $\ell(z)$ as η approaches zero. In the following, we bound the expected classification error $\mathbb{E}\ell(yf)$ based on the theory of *Rademacher averages* (Bartlett and Mendelson, 2002). Moreover, we can bound $\mathbb{E}\ell(yf)$ more tightly, if all ground truth class labels are available for evaluating the empirical classification error $\hat{\mathbb{E}}\ell_\eta(yf)$. We state the theoretical result in Theorem 4.4 and prove it in Section 4.7.

Theorem 4.4. *Assume that*

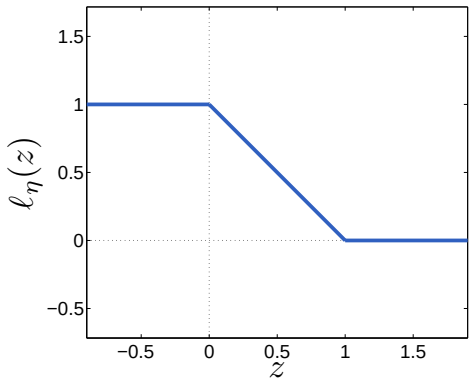
$$\exists B_k > 0, \forall x, x' \in \mathcal{X}, k(x, x') \leq B_k^2.$$



(a) $\ell(z)$



(b) $\ell_\eta(z), \eta = 0.5$



(c) $\ell_\eta(z), \eta = 1$

Figure 4.2: Illustration of loss functions

Let $\boldsymbol{\alpha}^*_{\mathcal{F}}$ *and* $f(x)$ *be the optimal solution and the decision function defined in Eqs.* (4.20) *and* (4.21) *respectively, and*

$$B_{\mathcal{F}} = \|D^{-1/2}\boldsymbol{\alpha}^*_{\mathcal{F}}\|_2$$
$$B'_{\mathcal{F}} = \|K^{-1/2}D^{-1/2}\boldsymbol{\alpha}^*_{\mathcal{F}}\|_1.$$

For any $\eta > 0$ *and* $0 < \delta < 1$*, with probability at least* $1 - \delta$*, we have*

$$\begin{aligned}\mathbb{E}\ell(yf(x)) \leq \frac{1}{l}\sum_{i=1}^{l}\ell_\eta(y_i f(x_i)) + \frac{2B_k B_{\mathcal{F}}}{\eta\sqrt{l}} \\ + \min\left(3, 1 + \frac{4B_k^2 B'_{\mathcal{F}}}{\eta}\right)\sqrt{\frac{\ln(2/\delta)}{2l}}.\end{aligned} \tag{4.22}$$

If the ground truth class labels $y_{l+1}, \ldots, y_n$ are also available for evaluation, with probability at least $1 - \delta$, we have

$$\begin{aligned}\mathbb{E}\ell(yf(x)) \leq \frac{1}{n}\sum_{i=1}^{n}\ell_\eta(y_i f(x_i)) + \frac{2B_k B_{\mathcal{F}}}{\eta\sqrt{n}} \\ + \min\left(3, 1 + \frac{4B_k^2 B'_{\mathcal{F}}}{\eta}\right)\sqrt{\frac{\ln(2/\delta)}{2n}}.\end{aligned} \tag{4.23}$$

Theorem 4.4 gives the tightest upper bounds (i.e., the coefficients of the third terms in the right-hand sides of (4.22) and (4.23) are smallest under each given scenario) based on the existing inductive definitions of Rademacher complexity (cf. Koltchinskii, 2001; Bartlett and Mendelson, 2002; Meir and Zhang, 2003; El-Yaniv and Pechyony, 2009). The bound in (4.22) is asymptotically $O(1/\sqrt{l})$, if we only know the first l class labels. In such cases, we may benefit from unlabeled data by a lower empirical error. The bound becomes asymptotically $O(1/\sqrt{n})$ in (4.23) if we can access the other u class labels, even though these labels are not used for training. Due to the smaller deviation of the empirical error and the empirical Rademacher complexity when they are estimated over all n data, we can improve the order of the bound from $O(1/\sqrt{l})$ to $O(1/\sqrt{n})$. Nevertheless, there is no free lunch: In (4.23), the empirical error is also evaluated over l labeled data and u unlabeled data, and it may be significantly higher than the empirical error evaluated without u unlabeled data. Basically, inequality (4.22) or inequality (4.23), which right-hand side is smaller reflects whether the underlying information maximization principle befits the data set or not.

Note that there is no unknown constant in our error bounds, and it is possible to roughly estimate how many data are needed to achieve a certain accuracy. For example, let us denote the expected classification error and the empirical classification error by E and E' respectively, then we know that

$$E - E' \leq \frac{2B_k B_{\mathcal{F}}}{\eta\sqrt{l}} + \min\left(3, 1 + \frac{4B_k^2 B'_{\mathcal{F}}}{\eta}\right)\sqrt{\frac{\ln(2/\delta)}{2l}}.$$

Fixing an upper bound of E, we would like to investigate the lower bound of l to guarantee that E is indeed below that upper bound. For Gaussian kernel function, B_k is identically one, and we may solve SMIR once to obtain E', $B_{\mathcal{F}}$ and $B'_{\mathcal{F}}$. If assuming that E', $B_{\mathcal{F}}$ and $B'_{\mathcal{F}}$ do not change rapidly with respect to l and u, for

any desired expected classification error E, Lipschitz constant η, and significance level δ, we have the following lower bound of l:

$$l \geq \left(\frac{2B_k B_{\mathcal{F}}/\eta + \min(3, 1 + 4B_k^2 B'_{\mathcal{F}}/\eta)\sqrt{\ln(2/\delta)/2}}{E - E'} \right)^2 .$$

The theoretical result in Theorem 4.4 is fairly novel. It is argued with both theoretical and experimental evidence that error bounds with some complexity measures which do not depend on given training data cannot be universally effective (Kearns et al., 1997). Fortunately, our data-dependent generalization error bounds can take both labeled and unlabeled data into account, and hence they can reflect certain properties of the particular mechanism generating the data. Thanks to the analytical solution in (4.20), our error bounds in (4.22) and (4.23) involve no optimization and then possess closed-form expression, even though they depend on given training data in terms of Rademacher complexities. Note that previous generalization and transductive error bounds for semi-supervised learning or transductive learning (e.g., Belkin et al., 2004 and Cortes et al., 2008) just focus on the regression error instead of the classification error. Therefore, to the best of our knowledge, neither semi-supervised nor transductive algorithm hitherto has been provided closed-form generalization error bounds similar to ours.

4.5 Related Works

Generally speaking, information-theoretic semi-supervised approaches directly constrain $p(y \mid x)$ by unlabeled data or some $p(x)$ given as the prior knowledge. *Information regularization* (IR; Szummer and Jaakkola, 2003) is the pioneer for this purpose. Compared with information maximization methods, IR minimizes the mutual information (MI) based on a key observation: Within a small region $Q \subset \mathcal{X}$, MI_Q is low/high if the label information is pure/chaotic. Subsequently, IR estimates a cover $\mathcal{C}$ of $\mathcal{X}$ from $\{x_1, \ldots, x_n\}$, and minimizes the maximal MI_Q for $Q \in \mathcal{C}$, subject to the class constraints provided by labeled data. The advantage of IR is its flexibility and convexity, while the drawback is that it is unclear how to define $\mathcal{C}$ properly. Each Q should be small enough to preserve the locality of the label information in a single region, and each pair Q and Q' should be

	Analytical solution	Out-of-sample classification	Multi-class classification	Probabilistic output
Geometric				
Transductive SVM (Joachims, 1999)	×	◯	△	×
Semi-supervised SVM (Bennett and Demiriz, 1999)	×	◯	△	×
Laplacian SVM (Belkin et al., 2006)	×	◯	△	×
Laplacian Regularized Least Squares (Belkin et al., 2006)	◯	◯	△	×
Markov Random Walks (Szummer and Jaakkola, 2002)	×	×	◯	◯
Local and Global Consistency (Zhou et al., 2004)	◯	△	◯	×
Spectral Graph Transducer (Joachims, 2003)	◯	×	×	×
Harmonic Energy Minimization (Zhu et al., 2003)	◯	×	×	◯
Sparse Eigenfunction Bases (Sinha and Belkin, 2009)	×	◯	×	×
Information-theoretic				
Information Regularization (Szummer and Jaakkola, 2003)	×	◯	◯	◯
Entropy Regularization (Grandvalet and Bengio, 2005)	×	◯	◯	◯
Expectation Regularization (Mann and McCallum, 2007)	×	◯	◯	◯
Regularized Information Maximization (Gomes et al., 2010)	×	◯	◯	◯
Squared-loss Mutual Information Regularization	◯	◯	◯	◯

◯: Yes ×: No △: Extension has been proposed

Table 4.1: Summary of existing semi-supervised learning methods

connected to ensure the dependence of $p(y \mid x)$ over all regions, which implies a great number of tiny regions.

By employing the Shannon entropy of $p(y \mid x)$ as a measure of class overlap, *entropy regularization* (ER; Grandvalet and Bengio, 2005) minimizes the entropy from a viewpoint of *maximum a posteriori* estimation. Specifically, ER regularizes the maximum log-likelihood estimation of a logistic regression or kernel logistic regression model by an entropy term:

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \ln q(y_i \mid x_i; \boldsymbol{\alpha}) \\ + \gamma \sum_{i=l+1}^{n} \sum\nolimits_{y \in \mathcal{Y}} q(y \mid x_i; \boldsymbol{\alpha}) \ln q(y \mid x_i; \boldsymbol{\alpha}). \end{aligned} \tag{4.24}$$

ER also favors low-density separations, since the low/high entropy means that the class overlap is mild/intensive. At a first glance, ER and IR seem opposite, because MI equals the difference of the entropies of class prior and posterior. However, IR minimizes MI *locally* and ER minimizes the entropy *globally*, so both of them highly penalize the variations of the class-posterior probability in high-density regions. A recent framework called *regularized information maximization* (RIM; Gomes et al., 2010) follows ER directly and further maximizes the entropy of the class-prior probability to encourage balanced classes. Compared with IR, ER and RIM do not model $p(x)$ explicitly, which is a big improvement. The disadvantage of ER and RIM is the non-convexity of their optimization problems.

Although ER does not model $p(x)$ explicitly, *expectation regularization* (XR; Mann and McCallum, 2007) goes one step further, namely, it does not use $p(x)$ at all. Therefore, XR is absolutely independent of low-density separations, and can even handle highly overlapped classes. The spirit of XR is to encourage model predictions on unlabeled data to match some designer-provided expectation by minimizing the KL-divergence between the expectations predicted by the model and provided as the prior knowledge. On the other hand, if there is no prior knowledge, XR matches the class prior of unlabeled data with that of labeled

data:

$$\begin{aligned}\max_{\boldsymbol{\alpha}} \sum_{i=1}^{l} \ln q(y_i \mid x_i; \boldsymbol{\alpha}) - \lambda \sum\nolimits_{y \in \mathcal{Y}} \frac{1}{2} \|\boldsymbol{\alpha}_y\|_2^2 \\ + \gamma \sum\nolimits_{y \in \mathcal{Y}} \pi_y \ln \left(\sum_{i=l+1}^{n} q(y \mid x_i; \boldsymbol{\alpha}) \right),\end{aligned} \tag{4.25}$$

where π_y is an estimate of $p(y)$ through labeled data, and $q(y \mid x; \boldsymbol{\alpha})$ is a logistic or kernel logistic regression model. Unlike IR and ER, neither parametric nor non-parametric XR prefers low-density separations. As a result, XR cannot deal with low-dimensional datasets with explicit nonlinear structures (such as the famous *two-moons* and *two-circles*), if there are not enough labeled data.

On the other hand, there are lots of geometric methods for semi-supervised learning. See Table 4.1 as a list of representative methods. Note that all geometric methods in Table 4.1 are in the style of either large margins or similarity graphs, and they favor the cluster or manifold assumption. According to Table 4.1, we could know that many methods based on similarity graphs (Szummer and Jaakkola, 2002; Zhou et al., 2004; Joachims, 2003; Zhu et al., 2003) are transductive, while the information-theoretic methods are all inductive; only two geometric methods (Szummer and Jaakkola, 2002; Zhou et al., 2004) could deal with multi-class data directly, while it is an inherent property of all information-theoretic methods. However, none of previous information-theoretic methods have analytical solutions, due to the logarithms in the entropy, MI or KL-divergence. Thanks to SMI, the proposed SMIR involves a strictly convex optimization problem with no logarithm inside and consequently has the analytic expression of the unique globally optimal solution.

The similarity between ER and SMIR is intriguing. Historically, RIM followed ER. Nonetheless, if we start from MI maximization with the uniform $p(y)$, we will get ER as

$$\max_{\boldsymbol{\alpha}} \int_{\mathcal{X}} \sum\nolimits_{y \in \mathcal{Y}} q(y \mid x; \boldsymbol{\alpha}) \ln q(y \mid x; \boldsymbol{\alpha}) p(x) \mathrm{d}x.$$

Recall that SMI maximization under the assumption of the uniform $p(y)$ is expressed by

$$\max_{\boldsymbol{\alpha}} \int_{\mathcal{X}} \sum\nolimits_{y \in \mathcal{Y}} q(y \mid x; \boldsymbol{\alpha}) q(y \mid x; \boldsymbol{\alpha}) p(x) \mathrm{d}x.$$

As a consequence, ER and SMIR have the similar preference, since the logarithm is strictly monotonically increasing. The vital difference is the convexity and the analytical solution: SMIR is convex and the globally optimal solution can be obtained analytically, whereas ER is non-convex so any locally optimal solution has to be found numerically.[4]

SMIR is not equivalent to the geometric framework known as *manifold regularization* (MR; Belkin et al., 2006), that is, we cannot reduce the optimization of SMIR to MR, or vice versa, despite that SMIR smoothes the parameters $\boldsymbol{\alpha}_y$ and MR smoothes the outputs $\boldsymbol{q}_y = (q(y \mid x_1), \ldots, q(y \mid x_n))^\top$ by minimizing $\boldsymbol{q}_y^\top L \boldsymbol{q}_y$ or $\boldsymbol{q}_y^\top \mathcal{L} \boldsymbol{q}_y$. Due to

$$\begin{aligned} \boldsymbol{\alpha}_y^\top \mathcal{L} \boldsymbol{\alpha}_y &= \|\mathcal{L}^{1/2} \boldsymbol{\alpha}_y\|_2^2, \\ \boldsymbol{q}_y^\top \mathcal{L} \boldsymbol{q}_y &= \|\mathcal{L}^{1/2} K^{1/2} D^{-1/2} \boldsymbol{\alpha}_y\|_2^2, \\ \det(\mathcal{L}^{1/2} K^{1/2} D^{-1/2}) &= \det(\mathcal{L}^{1/2}) \det(K^{1/2} D^{-1/2}), \end{aligned}$$

we know $\mathcal{L}^{1/2} K^{1/2} D^{-1/2}$ also has a zero eigenvalue, as $\mathcal{L}^{1/2}$. However, the associated eigenvectors are different, and hence $\|\mathcal{L}^{1/2} \boldsymbol{\alpha}_y\|_2$ cannot be *uniformly* bounded by a constant times $\|\mathcal{L}^{1/2} K^{1/2} D^{-1/2} \boldsymbol{\alpha}_y\|_2$, or vice versa. As a result, SMIR is a novel regularization approach which is different from MR.

4.6 Experiments

In this section, we numerically evaluate SMIR. The specification of benchmark data sets is summarized in Table 4.2. Besides four well-tried benchmarks in the first block (i.e., USPS, MNIST, 20Newsgroups and Isolet), there are eight benchmarks from a book entitled *Semi-Supervised Learning* (Chapelle et al., 2006)[5] in the second block, and eight benchmarks from the *UCI machine learning repository*[6] in the third block except that Senseval-2 is from a *word sense disambiguation* workshop[7] in conjunction with ACL 2001. Detailed explanation of benchmarks is omitted. Our experiments consist of three parts:

[4] In practice, SMIR may also be solved numerically in consideration of the computational efficiency for large n.

[5] http://olivier.chapelle.cc/ssl-book/benchmarks.html.

[6] http://archive.ics.uci.edu/ml/.

[7] http://www.senseval.org/.

	# Classes	# Dimensions	# Data	Balance of classes (in %)
USPS	10	256	11000	10 per class
MNIST	10	784	70000	11.3 / 10.0 / 10.2 / 9.8 / 9.0 / 9.8 / 10.4 / 9.8 / 9.9 / 9.9
20Newsgroups	7	53975	11269	4.3 / 25.8 / 5.2 / 21.1 / 21.0 / 5.3 / 17.3
Isolet	26	617	7797	3.85 per class
g241c	2	241	1500	50.0 / 50.0
g241n	2	241	1500	50.1 / 49.9
Digit1	2	241	1500	51.1 / 48.9
USPS	2	241	1500	80.0 / 20.0
COIL	6	241	1500	16.7 per class
COIL2	2	241	1500	50.0 / 50.0
BCI	2	117	400	50.0 / 50.0
Text	2	11960	1500	50.0 / 50.0
Diabetes	2	8	768	65.1 / 34.9
Wine	3	13	178	33.1 / 39.9 / 27.0
Vowel	11	13	990	9.1 per class
Image	2	18	1155	42.9 / 57.1
Vehicle	4	18	846	25.1 / 25.7 / 25.8 / 23.5
German	2	20	1000	70.0 / 30.0
Satimage	6	36	6435	23.8 / 10.9 / 21.1 / 9.7 / 11.0 / 23.4
Senseval-2	3	50	534	33.3 per class

Table 4.2: Specification of benchmark data sets

Firstly, we compare SMIR with entropy regularization (ER; Grandvalet and Bengio, 2005) and expectation regularization (XR; Mann and McCallum, 2007). Two probabilistic models for ER and XR are considered: The logistic regression

$$q(y \mid x; \boldsymbol{\alpha}) \propto \exp\langle x, \boldsymbol{\alpha}_y \rangle, \boldsymbol{\alpha}_y \in \mathbb{R}^d,$$

and the kernel logistic regression (Ker)

$$q(y \mid x; \boldsymbol{\alpha}) \propto \exp\langle \Phi_n(x), \boldsymbol{\alpha}_y \rangle, \boldsymbol{\alpha}_y \in \mathbb{R}^n,$$

where $\langle \cdot, \cdot \rangle$ is the inner product, Φ_n is the empirical kernel map of the Gaussian kernel. SMIR also applies the Gaussian kernel, so there are three kernel methods which allow nonlinear decision boundaries in $\mathbb{R}^d$. The two-fold cross-validation is performed to select the hyperparameters. The kernel width is the median of all pairwise distances times the best value among

$$\{1/15, 1/10, 1/5, 1/2, 1\}.$$

A Gaussian prior of parameters, which is same as the third term of optimization (4.9), is included for XR and KerXR (Mann and McCallum, 2007). No extra prior is added to ER or KerER, since ER itself is a prior from a viewpoint of maximum a posteriori estimation (Grandvalet and Bengio, 2005). Therefore, ER/KerER has one regularization parameter whereas XR/KerXR and SMIR have two. The candidate list of regularization parameters is

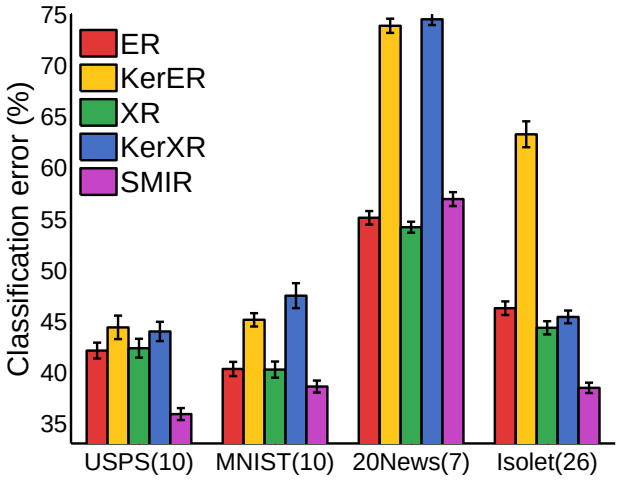
$$10\hat{}\{-7, -3, -1, 1, 3\},$$

except that λ is chosen from

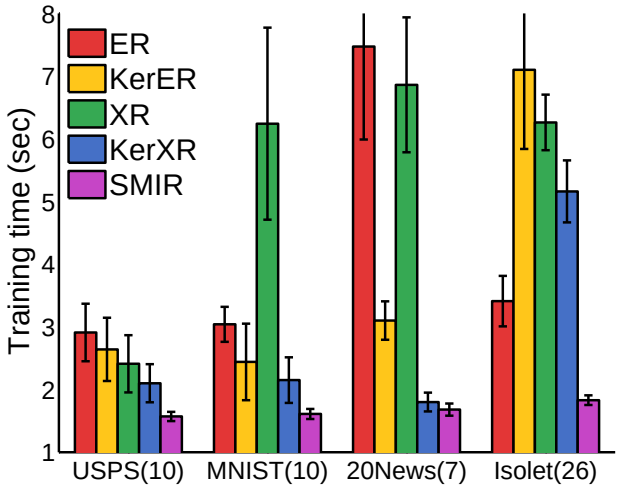
$$\gamma c/n + 10\hat{}\{-10, -8, -6, -4, -2\}$$

for SMIR to ensure the convexity. The *minFunc*[8] package for unconstrained optimization using line-search methods (the quasi-Newton limited-memory BFGS updates, by default) is utilized to solve ER/KerER and XR/KerXR. Since minimizing the entropy is non-convex, we initialize ER/KerER with the globally optimal solution of its supervised part.

[8] http://www.di.ens.fr/~mschmidt/Software/minFunc.html.

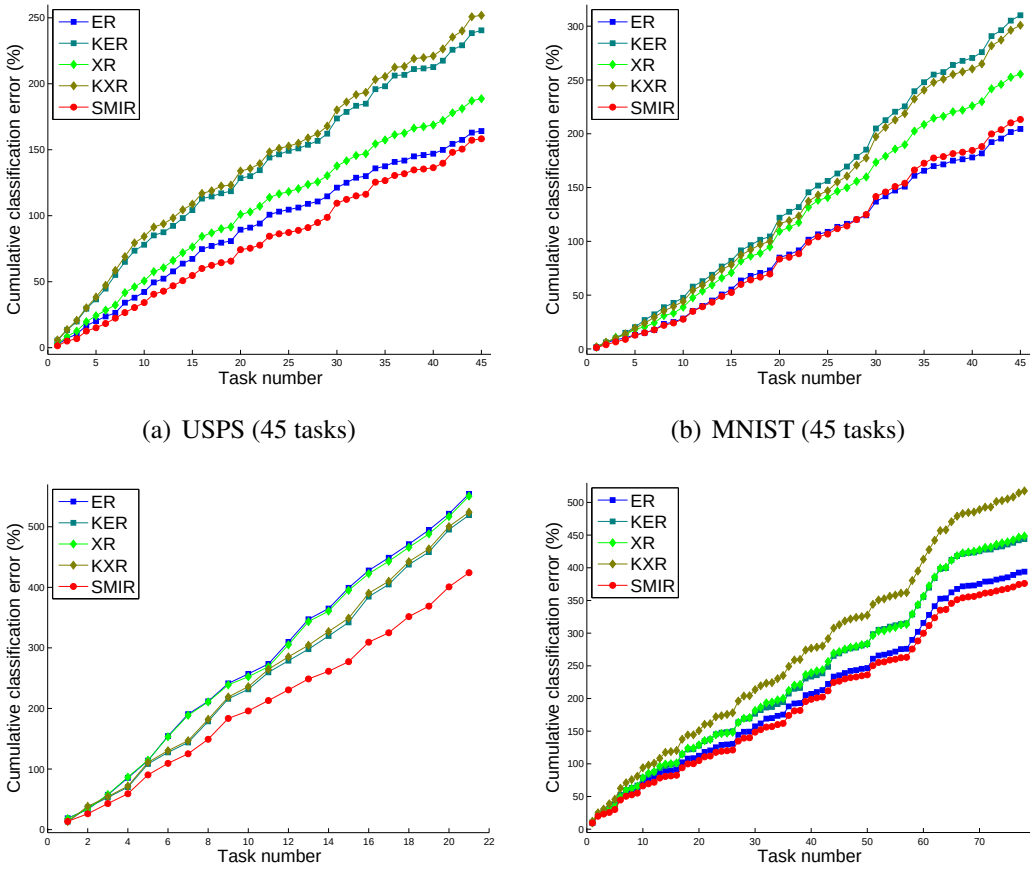


(a) Classification error



(b) Training time

Figure 4.3: Experimental results of the multi-class classification tasks. Means with standard errors are shown by bar charts.



(a) USPS (45 tasks)

(b) MNIST (45 tasks)

(c) 20News (21 tasks)

(d) Isolet (78 tasks)

Figure 4.4: Experimental results of the simple classification tasks. The cumulative classification error at the k-th task means the sum of classification errors from the first to k-th tasks, and non-cumulative standard deviations are shown along the curves.

	ER	KerER	XR	KerXR	SMIR
USPS, best or comparable (%)	45.65	15.22	21.74	17.39	**73.91**
MNIST, best or comparable (%)	**86.95**	0.00	19.57	2.17	80.43
20News, best or comparable (%)	36.36	18.18	36.36	18.18	**63.64**
Isolet, best or comparable (%)	60.76	62.03	68.35	48.10	**81.01**
USPS, training time (sec)	1.545	1.906	**1.149**	1.770	1.608
MNIST, training time (sec)	2.367	1.676	2.060	**1.536**	1.575
20News, training time (sec)	3.987	2.023	4.144	1.917	**1.654**
Isolet, training time (sec)	2.377	1.842	2.194	1.728	**1.723**

Table 4.3: Summary of all experimental results on USPS, MNIST, 20Newsgroups and Isolet. For each method, we measure how frequently it is the best or a comparable method based on the unpaired t-test at the significance level 5%, and the CPU time is averaged over all samplings of all tasks. The most accurate and computationally-efficient methods are highlighted in boldface.

	LapRLS	LGC	SMIR
USPS	39.64 ± 0.55	$\mathbf{36.53 \pm 0.53}$	$\mathbf{35.87 \pm 0.59}$
MNIST	42.34 ± 0.67	42.70 ± 0.60	$\mathbf{38.56 \pm 0.59}$
20News	64.85 ± 0.61	73.03 ± 0.24	$\mathbf{56.90 \pm 0.68}$
Isolet	39.98 ± 0.56	40.62 ± 0.47	$\mathbf{38.43 \pm 0.51}$

Table 4.4: Comparisons of LapRLS, LGC and SMIR, by means with standard errors of the classification error (in %) on the multi-class tasks. For each data set, the best and comparable methods based on the 5% unpaired t-test are highlighted in boldface.

We evaluated them on USPS, MNIST, 20Newsgroups and Isolet. Pearson's correlation (Hall, 2000) was used to select 1000 most informative words for 20Newsgroups. For each data set, we prepared a multi-class task, namely, the classification tasks involving 10 classes of USPS and MNIST, 7 classes of 20Newsgroups, as well as 26 classes of Isolet. In addition, extensive experiments of simple classification tasks were conducted, including 45 binary tasks of USPS, 45 binary tasks of MNIST and 21 binary tasks of 20Newsgroups. Isolet may lead to too many binary tasks and these tasks are often too easy, so we combined 26 letters into 13 groups (e.g., 'a' with 'b', 'c' with 'd' etc.) and treated each group as a single class resulting in 78 simple classification tasks. For each task, we repeatedly ran all methods on 100 random samplings, where the sample size was fixed to 500. Each random sampling was partitioned into a training set and a test set with 80% and 20% data, and 10% class labels of training data were revealed to construct labeled data.

Figure 4.3 and Figure 4.4 report the experimental results of the multi-class tasks and the simple tasks, and Table 4.3 summarizes all of experimental results. We can see from Figure 4.3 that SMIR outperformed others on the multi-class tasks of USPS, MNIST and Isolet. Likewise Figure 4.3 indicates that SMIR was the most computationally-efficient algorithm on all four multi-class tasks. Next, according to the curves in Figure 4.4, SMIR was the best on the simple tasks of USPS, 20Newsgroups and Isolet, but was slightly inferior to plain ER on MNIST. Note that there were 12 highly imbalanced tasks among 21 simple tasks of 20Newsgroups, which says that the uniform class-prior assumption will not affect the performance of SMIR essentially, if the tasks are not very complicated. The experiments of Isolet further imply that SMIR is fairly good at multi-modal data, since all classes there had two clusters. Compared with KerER and KerXR, plain ER and XR were better on USPS, MNIST and Isolet, but worse on 20Newsgroups. Nonetheless, ER/XR always outperformed KerER/KerXR in Table 4.3. Even though other algorithms quite often converged very quickly on the simple tasks, SMIR was still a computationally-efficient algorithm after taking these simple tasks into account.

Secondly, we compare SMIR with two geometric methods: Laplacian regularized least squares (LapRLS; Belkin et al., 2006) with a multi-class extension, as

	ER	KerER	XR	KerXR	LapRLS	LGC	SMIR
g241c	30.14 ± 0.55	$\mathbf{24.86 \pm 0.66}$	31.66 ± 0.81	$\mathbf{24.42 \pm 0.69}$	34.12 ± 0.69	36.53 ± 0.74	31.69 ± 0.66
g241n	$\mathbf{33.07 \pm 0.58}$	35.65 ± 0.98	$\mathbf{33.90 \pm 0.83}$	36.67 ± 0.99	35.07 ± 0.65	38.15 ± 0.72	$\mathbf{33.76 \pm 0.65}$
Digit1	12.12 ± 0.41	$\mathbf{9.31 \pm 0.32}$	12.47 ± 0.49	$\mathbf{9.68 \pm 0.62}$	11.44 ± 0.43	11.87 ± 0.46	10.23 ± 0.40
USPS	26.60 ± 0.59	17.58 ± 0.33	27.07 ± 0.90	18.02 ± 0.72	12.45 ± 0.34	$\mathbf{10.27 \pm 0.33}$	12.23 ± 0.40
COIL	46.16 ± 0.78	38.58 ± 0.98	50.55 ± 1.20	39.81 ± 1.06	37.03 ± 0.81	$\mathbf{32.95 \pm 0.88}$	$\mathbf{33.62 \pm 0.82}$
COIL2	28.83 ± 0.72	25.81 ± 0.75	31.54 ± 1.02	27.73 ± 0.98	26.52 ± 0.65	$\mathbf{23.39 \pm 0.71}$	$\mathbf{24.12 \pm 0.69}$
BCI	$\mathbf{40.58 \pm 0.67}$	47.76 ± 0.45	43.21 ± 0.70	48.35 ± 0.46	43.46 ± 0.63	48.70 ± 0.44	47.27 ± 0.57
Text	$\mathbf{34.92 \pm 0.56}$	44.36 ± 0.58	$\mathbf{35.38 \pm 0.54}$	43.79 ± 0.65	44.50 ± 0.54	49.53 ± 0.18	38.80 ± 0.64

Table 4.5: Means with standard errors of the classification error (in %) on benchmarks from Chapelle et al. (2006). For each data set, the best method and comparable ones based on the unpaired t-test at the significance level 5% are highlighted in boldface.

	ER	KerER	XR	KerXR	LapRLS	LGC	SMIR
Diabetes	$\mathbf{27.26 \pm 0.41}$	29.70 ± 0.50	28.41 ± 0.53	30.16 ± 0.72	32.01 ± 0.62	32.32 ± 0.42	29.87 ± 0.57
Wine	8.09 ± 0.44	$\mathbf{4.21 \pm 0.44}$	10.56 ± 1.21	6.56 ± 0.95	8.21 ± 0.45	7.71 ± 0.44	6.91 ± 0.54
Vowel	70.65 ± 0.78	63.03 ± 0.78	69.70 ± 0.77	$\mathbf{61.32 \pm 0.68}$	63.90 ± 0.65	64.13 ± 0.66	$\mathbf{62.77 \pm 0.65}$
Image	27.32 ± 0.64	22.38 ± 0.67	26.91 ± 0.75	23.07 ± 0.90	$\mathbf{18.80 \pm 0.66}$	$\mathbf{19.45 \pm 0.65}$	$\mathbf{19.82 \pm 0.67}$
Vehicle	39.43 ± 0.90	45.61 ± 0.78	48.44 ± 1.10	46.86 ± 0.91	$\mathbf{38.22 \pm 0.79}$	43.01 ± 0.54	$\mathbf{37.48 \pm 0.74}$
German	32.30 ± 0.55	$\mathbf{29.31 \pm 0.31}$	32.76 ± 0.65	$\mathbf{29.45 \pm 0.35}$	30.96 ± 0.42	30.94 ± 0.33	30.62 ± 0.43
Satimage	31.01 ± 0.73	22.59 ± 0.58	34.79 ± 0.68	25.12 ± 1.43	20.15 ± 0.40	$\mathbf{18.75 \pm 0.34}$	$\mathbf{18.96 \pm 0.39}$
Senseval-2	$\mathbf{32.72 \pm 0.62}$	35.56 ± 0.73	37.14 ± 1.10	36.37 ± 0.83	34.66 ± 0.71	37.77 ± 0.67	$\mathbf{33.11 \pm 0.74}$

Table 4.6: Means with standard errors of the classification error (in %) on seven UCI benchmarks and Senseval-2. For each data set, the best method and comparable ones based on the unpaired t-test at the significance level 5% are highlighted in boldface.

well as learning with local and global consistency (LGC; Zhou et al., 2004) with an out-of-sample extension. LapRLS and LGC could represent respectively state-of-the-art manifold regularization and graph transduction. Similarly to SMIR, their optimizations are convex and can be solved analytically. LapRLS is extended using the one-vs-rest trick, and LGC is extended via the Nadaraya-Watson estimator (Delalleau et al., 2005). The experimental setup and the candidates of hyperparameters are same as before, except that the regularization parameter α of LGC is chosen from $\{0.2, 0.4, 0.6, 0.8, 0.99\}$. SMIR was always best or tie according to the experimental results in Table 4.4, and thus it can be a useful alternative to pure geometric methods on these benchmarks.

Finally, we take all seven methods and compare their performance on the sixteen benchmarks listed in Table 4.2. The experimental results are reported in Tables 4.5 and 4.6 respectively, where the experimental setup and the candidates of hyperparameters are same as before. To be clear, there are two benchmarks, BCI and Wine, whose sample size is less than 500. As a result, each of their random samplings included the whole set, and the randomness or the difference of the classification error was actually from how the training, test and cross-validation data were split and also how labeled data were selected. We can see that in Table 4.5, ER, LGC and SMIR were best or comparable on three benchmarks, and KerER, XR and KerXR were best or comparable on two benchmarks. Moreover, in Table 4.6, SMIR won or tied five times, while all other methods except XR won or tied two times. Therefore, the principle of SMIR, which adopts SMI as the information measure to be maximized, is reasonable and practical, likewise the proposed SMIR is a promising information-theoretic approach to semi-supervised learning.

4.7 Proof of the Generalization Error Bounds

4.7.1 Definitions

To begin with, we state the inductive definition of Rademacher complexity following (El-Yaniv and Pechyony, 2009).

Definition 4.5. *Suppose that $x_1, \ldots, x_n$ are independent observations according*

to $p(x)$. Let $\mathcal{F}$ be a class of functions mapping from $\mathcal{X}$ to $\mathbb{R}$, and $\sigma_1, \ldots, \sigma_n$ be independent uniformly $\{\pm 1\}$-valued random variables, i.e., Rademacher variables. Subsequently, the empirical Rademacher complexity conditioned on $x_1, \ldots, x_n$ is defined as

$$\widehat{\mathcal{R}}_n(\mathcal{F}) := \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f\in\mathcal{F}} \frac{2}{n}\sum_{i=1}^{n}\sigma_i f(x_i)\right\},$$

and the inductive Rademacher complexity is defined as

$$\mathcal{R}_n(\mathcal{F}) := \mathbb{E}_{x_1,\ldots,x_n}\left\{\widehat{\mathcal{R}}_n(\mathcal{F})\right\}.$$

There exist various definitions of $\widehat{\mathcal{R}}_n(\mathcal{F})$: The definition in Bartlett and Mendelson (2002) is

$$\widehat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f\in\mathcal{F}} \frac{2}{n}\left|\sum_{i=1}^{n}\sigma_i f(x_i)\right|\right\},$$

the definition in Koltchinskii (2001) uses

$$\widehat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f\in\mathcal{F}} \frac{1}{n}\left|\sum_{i=1}^{n}\sigma_i f(x_i)\right|\right\},$$

while the definition in Meir and Zhang (2003) adopt

$$\widehat{\mathcal{R}}_n(\mathcal{F}) = \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n}\sigma_i f(x_i)\right\}.$$

The definition in El-Yaniv and Pechyony (2009) is consistent with Bartlett and Mendelson (2002) for function classes that are closed under negation, and is always equal to or less than the one in Bartlett and Mendelson (2002).

Nevertheless, a vital disagreement arises when considering comparison theorems and thus the famous *contraction principle* of Rademacher averages. If $\psi : \mathbb{R} \mapsto \mathbb{R}$ is Lipschitz continuous with a Lipschitz constant L_ψ and satisfies $\psi(0) = 0$, then

$$\widehat{\mathcal{R}}_n(\psi \circ \mathcal{F}) \leq L_\psi \widehat{\mathcal{R}}_n(\mathcal{F})$$

for El-Yaniv and Pechyony (2009) and

$$\widehat{\mathcal{R}}_n(\psi \circ \mathcal{F}) \leq 2L_\psi \widehat{\mathcal{R}}_n(\mathcal{F})$$

for Bartlett and Mendelson (2002). When all involved error bounds are single-sided concentration results, those definitions without the absolute value in the argument of the supremum (El-Yaniv and Pechyony, 2009; Meir and Zhang, 2003) are more natural and powerful.

4.7.2 Proof of Theorem 4.4

Let $\boldsymbol{\beta}_{\mathcal{F}} = K^{-1/2}D^{-1/2}\boldsymbol{\alpha}^*_{\mathcal{F}}$, then

$$B^2_{\mathcal{F}} = \|D^{-1/2}\boldsymbol{\alpha}^*_{\mathcal{F}}\|^2_2 = \boldsymbol{\beta}^\top_{\mathcal{F}} K \boldsymbol{\beta}_{\mathcal{F}},$$
$$B'_{\mathcal{F}} = \|K^{-1/2}D^{-1/2}\boldsymbol{\alpha}^*_{\mathcal{F}}\|_1 = \|\boldsymbol{\beta}_{\mathcal{F}}\|_1.$$

Define the class of functions $\mathcal{F}$ as

$$\mathcal{F} := \left\{ f : x \mapsto \sum_{i=1}^{n} \beta_i k(x, x'_i) \;\middle|\; x'_i \in \mathcal{X}, \beta_i \in \mathbb{R}, \right.$$
$$\left. \sum_{i=1}^{n} |\beta_i| \le B'_{\mathcal{F}}, \sum_{i,j=1}^{n} \beta_i \beta_j k(x'_i, x'_j) \le B^2_{\mathcal{F}} \right\}.$$

It is easy to verify that $f(x) = \langle \Phi_n(x), \boldsymbol{\beta}_{\mathcal{F}} \rangle \in \mathcal{F}$, where $f(x)$ is the decision function defined in Eq. (4.21). By Lemma 22 of Bartlett and Mendelson (2002), we get

$$\widehat{\mathcal{R}}_n(\mathcal{F}) \le \frac{2B_{\mathcal{F}}}{n} \left(\sum_{i=1}^{n} k(x_i, x_i) \right)^{1/2} \le \frac{2B_k B_{\mathcal{F}}}{\sqrt{n}}. \tag{4.26}$$

Applying Lemma 22 of Bartlett and Mendelson (2002) again gives us

$$\widehat{\mathcal{R}}_l(\mathcal{F}) \le \frac{2B_{\mathcal{F}}}{l} \left(\sum_{i=1}^{l} k(x_i, x_i) \right)^{1/2} \le \frac{2B_k B_{\mathcal{F}}}{\sqrt{l}}. \tag{4.27}$$

where $\widehat{\mathcal{R}}_l(\mathcal{F})$ is the empirical Rademacher complexities of $\mathcal{F}$ conditioned only on $x_1, \ldots, x_l$.

In the following, we only focus on the proof of inequality (4.23) based on inequality (4.26). Inequality (4.22) can be derived by the exactly same way based on inequality (4.27). Let

$$\ell_\eta \circ \mathcal{F} := \{(x, y) \mapsto \ell_\eta(yf(x)) \mid f \in \mathcal{F}\},$$

which is a class of functions mapping from $\mathcal{X} \times \mathcal{Y}$ to the interval $[0, 1]$. The rest of the proof consists of two steps. The first step bounds $\mathcal{R}_n(\ell_\eta \circ \mathcal{F})$ from above, and the second step bounds $\mathbb{E}\ell(yf(x))$ using $\mathcal{R}_n(\ell_\eta \circ \mathcal{F})$.

Step 1

The following lemma relates the inductive Rademacher complexity of a class of bounded functions to the corresponding empirical Rademacher complexity.

Lemma 4.6 (Concentration Lemma)**.** *Let $\mathcal{F}_C$ be a class of functions mapping to the interval $[-C, C]$. With probability at least $1 - \delta/2$, we have*

$$\mathcal{R}_n(\mathcal{F}_C) \leq \widehat{\mathcal{R}}_n(\mathcal{F}_C) + 4C\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Similarly, let $\mathcal{F}_C^+$ be a class of functions mapping to the interval $[0, C]$. With probability at least $1 - \delta/2$, we have

$$\mathcal{R}_n(\mathcal{F}_C^+) \leq \widehat{\mathcal{R}}_n(\mathcal{F}_C^+) + 2C\sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Proof. Recall that $\widehat{\mathcal{R}}_n(\mathcal{F}_C)$ conditioned on $x_1, \ldots, x_n$ is a random variable defined as

$$\widehat{\mathcal{R}}_n(\mathcal{F}_C) = \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f \in \mathcal{F}_C} \frac{2}{n}\sum_{i=1}^{n} \sigma_i f(x_i)\right\}.$$

When an observation x_i changes to x_i', the change of $\widehat{\mathcal{R}}_n(\mathcal{F}_C)$ is no more than $4C/n$, and thus *McDiarmid's inequality* (McDiarmid, 1989) implies that

$$\Pr\left\{\mathcal{R}_n(\mathcal{F}_C) - \widehat{\mathcal{R}}_n(\mathcal{F}_C) \geq \epsilon\right\} \leq \exp\left(-\frac{\epsilon^2 n}{8C^2}\right).$$

The first bound can be obtained by equating the right-hand side of the above inequality to $\delta/2$.

For $\mathcal{F}_C^+$, when an observation x_i changes to x_i', the change of $\widehat{\mathcal{R}}_n(\mathcal{F}_C^+)$ is no more than $2C/n$. The lemma follows by the same argument as above. □

The next lemma is a variation of the comparison lemma in Meir and Zhang (2003), where the comparison is done for two sets of functions under a Bayesian framework, and its validity follows Lemma 5 of El-Yaniv and Pechyony (2009) by setting $p = 1/2$.

Lemma 4.7 (Comparison Lemma)**.** *Let*

$$\mathcal{H} := \{\boldsymbol{h} = (h_1, \ldots, h_n)^\top \mid h_i = y_i f(x_i), f \in \mathcal{F}\},$$

and $\psi, \psi' : \mathbb{R} \mapsto \mathbb{R}$ *be real-valued functions. If for all* $\boldsymbol{h}, \boldsymbol{h}' \in \mathcal{H}$ *and* $i = 1, \ldots, n$*,*

$$|\psi(h_i) - \psi(h_i')| \leq |\psi'(h_i) - \psi'(h_i')|,$$

then

$$\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{h\in\mathcal{H}} \sum_{i=1}^n \sigma_i \psi(h_i)\right\} \leq \mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{h\in\mathcal{H}} \sum_{i=1}^n \sigma_i \psi'(h_i)\right\}.$$

Now $\widehat{\mathcal{R}}_n(\ell_\eta \circ \mathcal{F})$ and $\mathcal{R}_n(\ell_\eta \circ \mathcal{F})$ can be bounded from above by $\widehat{\mathcal{R}}_n(\mathcal{F})$ and $\mathcal{R}_n(\mathcal{F})$ based on the comparison lemma.

Lemma 4.8 (Contraction Lemma)**.** *For any* $\eta > 0$*, we have*

$$\widehat{\mathcal{R}}_n(\ell_\eta \circ \mathcal{F}) \leq \frac{1}{\eta}\widehat{\mathcal{R}}_n(\mathcal{F}),$$
$$\mathcal{R}_n(\ell_\eta \circ \mathcal{F}) \leq \frac{1}{\eta}\mathcal{R}_n(\mathcal{F}).$$

Proof. Note that $\ell_\eta(z)$ satisfies the Lipschitz condition

$$|\ell_\eta(z) - \ell_\eta(z')| \leq \frac{1}{\eta}|z - z'|, \quad \forall z, z' \in \mathbb{R}.$$

Let $\psi(h_i) = \ell_\eta(y_i f(x_i))$ and $\psi'(h_i) = y_i f(x_i)/\eta$, then

$$\begin{aligned}\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f\in\mathcal{F}} \sum_{i=1}^n \sigma_i \ell_\eta(y_i f(x_i))\right\} &\leq \frac{1}{\eta}\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f\in\mathcal{F}} \sum_{i=1}^n \sigma_i y_i f(x_i)\right\} \\ &= \frac{1}{\eta}\mathbb{E}_{\sigma_1,\ldots,\sigma_n}\left\{\sup_{f\in\mathcal{F}} \sum_{i=1}^n \sigma_i f(x_i)\right\},\end{aligned}$$

where the first step is a corollary of the comparison lemma, and the second step is due to the same distribution of each $\sigma_i y_i$ and σ_i. This completes the proof. □

As a result, if we contract $\widehat{\mathcal{R}}_n(\mathcal{F})$ and then concentrate $\widehat{\mathcal{R}}_n(\ell_\eta \circ \mathcal{F})$, we could know

$$\begin{aligned}\mathcal{R}_n(\ell_\eta \circ \mathcal{F}) &\leq \widehat{\mathcal{R}}_n(\ell_\eta \circ \mathcal{F}) + 2\sqrt{\frac{\ln(2/\delta)}{2n}} \\ &\leq \frac{2B_k B_{\mathcal{F}}}{\eta\sqrt{n}} + 2\sqrt{\frac{\ln(2/\delta)}{2n}},\end{aligned} \tag{4.28}$$

since ℓ_η maps to the interval $[0, 1]$. On the other hand, for any $f \in \mathcal{F}$,

$$\|f\|_\infty = \sup_{x \in \mathcal{X}} \left| \sum_{i=1}^{n} \beta_i k(x, x_i') \right| \leq B_k^2 B_{\mathcal{F}}',$$

which says that $\mathcal{F}$ is a class of functions mapping to the interval $[-B_k^2 B_{\mathcal{F}}', B_k^2 B_{\mathcal{F}}']$. Thus, if we concentrate $\widehat{\mathcal{R}}_n(\mathcal{F})$ before contract $\mathcal{R}_n(\mathcal{F})$, we can obtain

$$\begin{aligned} \mathcal{R}_n(\ell_\eta \circ \mathcal{F}) &\leq \frac{1}{\eta} \mathcal{R}_n(\mathcal{F}) \\ &\leq \frac{1}{\eta} \left(\frac{2 B_k B_{\mathcal{F}}}{\sqrt{n}} + 4 B_k^2 B_{\mathcal{F}}' \sqrt{\frac{\ln(2/\delta)}{2n}} \right). \end{aligned} \tag{4.29}$$

Combining inequalities (4.28) and (4.29) finalizes the first step of the proof, that is,

$$\mathcal{R}_n(\ell_\eta \circ \mathcal{F}) \leq \frac{2 B_k B_{\mathcal{F}}}{\eta \sqrt{n}} + \min \left(2, \frac{4 B_k^2 B_{\mathcal{F}}'}{\eta} \right) \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

Step 2

This step is composed of a single concentration inequality, that is, with probability at least $1 - \delta/2$,

$$\mathbb{E}\ell(yf(x)) \leq \hat{\mathbb{E}}_n \ell_\eta(yf(x)) + \mathcal{R}_n(\ell_\eta \circ \mathcal{F}) + \sqrt{\frac{\ln(2/\delta)}{2n}}. \tag{4.30}$$

Since $\forall z \in \mathbb{R}$, $\ell(z)$ is always equal to or less than $\ell_\eta(z)$, for any $f \in \mathcal{F}$ we can write

$$\begin{aligned} \mathbb{E}\ell(yf(x)) &\leq \mathbb{E}\ell_\eta(yf(x)) \\ &\leq \hat{\mathbb{E}}_n \ell_\eta(yf(x)) + \sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi). \end{aligned}$$

Any function $\psi(x, y) = \ell_\eta(yf(x)) \in \ell_\eta \circ \mathcal{F}$ satisfies $0 \leq \psi(x, y) \leq 1$, so when (x_i, y_i) changes to (x_i', y_i'), the change of $\sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi)$ cannot be more than $1/n$. Hence, McDiarmid's inequality implies that

$$\Pr \left\{ \sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi) - \mathbb{E}_{(x_1, y_1), \ldots, (x_n, y_n)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi) \geq \epsilon \right\} \leq \exp(-2\epsilon^2 n),$$

or equivalently, with probability at least $1 - \delta/2$,

$$\sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi) \leq \mathbb{E}_{(x_1,y_1),\ldots,(x_n,y_n)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi) + \sqrt{\frac{\ln(2/\delta)}{2n}}.$$

It remains to bound the expectation $\mathbb{E}_{(x_1,y_1),\ldots,(x_n,y_n)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi)$ by the complexity $\mathcal{R}_n(\ell_\eta \circ \mathcal{F})$. Suppose that

$$\{(x'_1, y'_1), \ldots, (x'_n, y'_n) \mid (x'_i, y'_i) \sim p(x, y)\}$$

is a ghost sample for symmetrization, then

$$\begin{aligned}
&\mathbb{E}_{(x_i,y_i)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} (\mathbb{E}\psi - \hat{\mathbb{E}}_n \psi) \\
&\quad = \mathbb{E}_{(x_i,y_i)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \Big(\mathbb{E}_{(x'_i,y'_i)} [\hat{\mathbb{E}}_n \psi(x'_i, y'_i)] - \hat{\mathbb{E}}_n \psi(x_i, y_i) \Big) \\
&\quad = \mathbb{E}_{(x_i,y_i)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \Big(\mathbb{E}_{(x'_i,y'_i)} [\hat{\mathbb{E}}_n \psi(x'_i, y'_i) - \hat{\mathbb{E}}_n \psi(x_i, y_i)] \Big) \\
&\quad \leq \mathbb{E}_{(x_i,y_i),(x'_i,y'_i)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \Big(\hat{\mathbb{E}}_n \psi(x'_i, y'_i) - \hat{\mathbb{E}}_n \psi(x_i, y_i) \Big) \qquad (4.31) \\
&\quad = \mathbb{E}_{(x_i,y_i),(x'_i,y'_i)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (\psi(x'_i, y'_i) - \psi(x_i, y_i)) \\
&\quad = \mathbb{E}_{\sigma_i,(x_i,y_i),(x'_i,y'_i)} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i (\psi(x'_i, y'_i) - \psi(x_i, y_i)) \qquad (4.32) \\
&\quad \leq \mathbb{E}_{(x'_i,y'_i),\sigma_i} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i \psi(x'_i, y'_i) \\
&\qquad + \mathbb{E}_{(x_i,y_i),\sigma_i} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \frac{1}{n} \sum_{i=1}^n (-\sigma_i) \psi(x_i, y_i) \\
&\quad = 2\mathbb{E}_{(x_i,y_i),\sigma_i} \sup_{\psi \in \ell_\eta \circ \mathcal{F}} \frac{1}{n} \sum_{i=1}^n \sigma_i \psi(x_i, y_i) \qquad (4.33) \\
&\quad = \mathcal{R}_n(\ell_\eta \circ \mathcal{F}),
\end{aligned}$$

where (4.31) uses the fact that the supremum is a convex function and then we apply *Jensen's inequality*, (4.32) is due to the symmetry of the ghost sample and the original sample and thus the same distribution of $\psi(x'_i, y'_i) - \psi(x_i, y_i)$ and $\sigma_i(\psi(x'_i, y'_i) - \psi(x_i, y_i))$, and (4.33) is valid since σ_i and $-\sigma_i$ have the same distribution while the original and ghost samples also have the same distribution. □

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis was devoted to developing discriminative methods with imperfect supervision in machine learning. We formulated six major settings of learning with imperfect supervision, and then focused on three typical imperfectly supervised learning problems: Discriminative clustering, (weak-supervised) metric learning, and semi-supervised classification.

In Chapter 2, we proposed *maximum volume clustering* (MVC), a novel discriminative approach to clustering. It partitions the data to be clustered based on the large volume principle so that hypotheses lying in an equivalence class with a larger volume are more preferable. Two algorithms were developed for approximating the basic model of MVC:

- MVC-HL relaxes MVC to a semi-definite programming problem which is convex but time-consuming;
- MVC-SL employs sequential quadratic programming which is non-convex but computationally-efficient.

Then, we showed that MVC includes the optimizations of some famous clustering methods as special limit cases, and discussed in great detail the finite sample stability and the data-dependent error bound of MVC-SL. Given the encouraging experimental results on many artificial and benchmark data sets, we conclude that the proposed MVC approach is promising, especially for images and text.

In Chapter 3, we proposed SERAPH (which is named after *SEmi-supervised metRic leArning Paradigm with Hyper-sparsity*), a novel discriminative approach to semi-supervised metric learning. It serves as an alternative to the methods applying manifold regularization or manifold embedding to semi-supervised metric learning. To begin with, the generalized maximum entropy distribution estimation for supervised metric learning was our foundation. Then, a semi-supervised extension that can achieve the posterior sparsity was obtained by entropy regularization. Moreover, we enforced a trace-norm regularization that encourages the projection sparsity. The constrained optimization problem of SERAPH was finally solved by an EM-like scheme with some nice algorithmic properties. Experiments on many benchmark data sets demonstrated that given limited supervised information, SERAPH usually outperformed state-of-the-art metric learning methods, and the learned metric possessed high discriminability even under a noisy environment.

In Chapter 4, we proposed *squared-loss mutual information regularization* (SMIR), a novel discriminative approach to semi-supervised learning. It serves as an information-theoretic regularization technique for learning multi-class probabilistic classifiers. In contrast to other information-theoretic regularization techniques, SMIR is convex with no logarithm in the involved optimization problem, and thus enables the analytic expression of the globally optimal solution. Compared with the geometric methods for semi-supervised classification, SMIR directly deals with multi-class out-of-sample classification problems. Moreover, we established novel data-dependent generalization error bounds that take the information of unlabeled data into account and can test whether the information maximization principle befits the data set or not. We evaluated the proposed SMIR on twenty benchmark data sets, and the results demonstrated that it compared favorably with state-of-the-art semi-supervised learning methods.

5.2 Problems for the Future

In the final section of this thesis, we show some important problems for the future. The first subsection includes possible directions of MVC, the second subsection includes possible directions of SERAPH, and the last subsection includes possible

directions of SMIR.

5.2.1 Future Directions of MVC

Recall that without any weak labels (similarity and dissimilarity constraints), the current model of MVC partitions the data samples into two clusters based on the large volume principle, and the primal problem of sofe-label MVC can be written as

$$\min_{\boldsymbol{h}\in\mathbb{R}^n} \ -2\|\boldsymbol{h}\|_1 + \gamma \boldsymbol{h}^\top Q \boldsymbol{h} \qquad \text{s.t. } \|\boldsymbol{h}\|_2 = 1.$$

When we have some weak labels at hand, we can modify MVC as follows. Put x_i and x_j into the same cluster if they are similar, i.e., $h_i h_j > 0$ if $(x_i, x_j) \in \mathcal{S}$; put x_i and x_j into different clusters if they are dissimilar, i.e., $h_i h_j < 0$ if $(x_i, x_j) \in \mathcal{D}$. The optimization problem becomes

$$\begin{aligned} \min_{\boldsymbol{h}\in\mathbb{R}^n} \ & -2\|\boldsymbol{h}\|_1 + \gamma \boldsymbol{h}^\top Q \boldsymbol{h} \\ \text{s.t.} \ & \|\boldsymbol{h}\|_2 = 1 \\ & h_i h_j > 0, \forall (x_i, x_j) \in \mathcal{S} \\ & h_i h_j < 0, \forall (x_i, x_j) \in \mathcal{D}. \end{aligned} \tag{5.1}$$

Optimization (5.1) can no longer be solved by sequential quadratic programming or semi-definite programming, and then in order to solve it we need more advanced optimization methods.

Next, the basic model of MVC is currently binary, and it needs a multi-way extension to partition the data samples into more than two clusters. To this end, we should extend the definition of the volume before extending the basic model of MVC. Unlike the margin, there exists no multi-class definition of the volume hitherto. We may borrow the idea of the multi-class definition of the margin in Crammer and Singer (2001) based on which the first multi-way extension of MMC was proposed (Xu and Schuurmans, 2005).

The proposed approximation schemes and optimization algorithms for MVC may be improved. However, we believe that the improvement cannot be straightforward. We have considered several options and found that none of them befits MVC well. Recall that the primal problem of MVC-SL defined in (2.4) is non-convex, and the *concave-convex procedure* (CCP) and the *constrained CCP*

(CCCP) (Yuille and Rangarajan, 2003; Smola et al., 2005) seem able to solve it. In fact, CCP can only be applied to the Lagrange function $L(\boldsymbol{h}, \eta)$, and η as an optimization variable may diverge even though $\boldsymbol{h}$ is guaranteed to converge given constant η. On the other hand, CCCP accepts any first-order equality constraint and any inequality constraint involving the difference of two convex functions, but the second-order equality constraint like $\boldsymbol{h}^\top \boldsymbol{h} = 1$ is unacceptable. If we relax the equality constraint $\boldsymbol{h}^\top \boldsymbol{h} = 1$ into an inequality constraint $\boldsymbol{h}^\top \boldsymbol{h} \leq 1$, we will get

$$\min_{\boldsymbol{h} \in \mathbb{R}^n} \; -2\|\boldsymbol{h}\|_1 + \gamma \boldsymbol{h}^\top Q \boldsymbol{h} \qquad \text{s.t. } \boldsymbol{h}^\top \boldsymbol{h} \leq 1. \tag{5.2}$$

Unfortunately, CCCP fails to solve optimization (5.2) again, since now we cannot assume that $\|\boldsymbol{h}\|_1$ is differentiable and then we cannot easily linearize the concave part of the energy function. Note that the popular trick to cope with ℓ_1-regularization is futile here, since (5.2) is never equivalent to

$$\begin{aligned} \min_{\boldsymbol{h} \in \mathbb{R}^n} \; & -2\boldsymbol{\alpha}^\top \mathbf{1}_n + \gamma \boldsymbol{h}^\top Q \boldsymbol{h} \\ \text{s.t. } & \boldsymbol{h}^\top \boldsymbol{h} \leq 1, -\boldsymbol{\alpha} \leq \boldsymbol{h} \leq \boldsymbol{\alpha}, \boldsymbol{\alpha} \geq \mathbf{0}_n. \end{aligned}$$

Similarly, (5.2) itself is not *quadratically-constrained quadratic programming* (QCQP) (Boyd and Vandenberghe, 2004) due to the minimization of negative ℓ_1-norm, but it can be reformulated as a QCQP with an optimization variable essentially in $\mathbb{R}^{2n}$:

$$\min_{\boldsymbol{y} \in [-1,+1]^n} \min_{\boldsymbol{h} \in \mathbb{R}^n} \; -2\boldsymbol{h}^\top \boldsymbol{y} + \gamma \boldsymbol{h}^\top Q \boldsymbol{h} \qquad \text{s.t. } \boldsymbol{h}^\top \boldsymbol{h} \leq 1. \tag{5.3}$$

Although optimization (5.3) is convex in $\boldsymbol{y}$ and convex in $\boldsymbol{h}$, it is not jointly convex in $\boldsymbol{y}$ and $\boldsymbol{h}$, so no off-the-shelf QCQP solver is applicable and we need relax it via semi-definite programming or *reformulation-linearization technique* (Sherali and Adams, 1998) once more. Actually, the feasible region $[-1, +1]^n$ of $\boldsymbol{y}$ is as difficult as the combinatorial $\{-1, +1\}^n$, and all of optimizations (2.2), (2.4), (5.2) and (5.3) are NP-hard, regardless of the different feasible regions of $\boldsymbol{h}$.

In contrast to MVC-SL, there is much more room for MVC-HL to be improved. GMMC uses a tricky substitution to get (2.24), and that substitution is so specific that it does not work for MVC-HL. Following the idea of LGMMC, we

can obtain an alternative relaxation as

$$\begin{aligned}\min_{\boldsymbol{\mu}\in\mathbb{R}^{2^n}} \min_{\boldsymbol{\alpha}} \; & -2\boldsymbol{\alpha}^\top \mathbf{1}_n + \gamma \boldsymbol{\alpha}^\top \left(\sum\nolimits_{t:-b\le \boldsymbol{y}_t^\top \mathbf{1}_n \le b} \mu_t Q \circ \boldsymbol{y}_t \boldsymbol{y}_t^\top \right) \boldsymbol{\alpha} \\ \text{s.t.} \; & \boldsymbol{\mu}^\top \mathbf{1}_{2^n} = 1, \boldsymbol{\mu} \ge \mathbf{0}_{2^n} \\ & \boldsymbol{\alpha}^\top \boldsymbol{\alpha} = 1, \boldsymbol{\alpha} \ge \mathbf{0}_n.\end{aligned}$$

Similarly, this optimization can also be regarded as a multiple kernel learning problem and solved by the cutting plane method, as LGMMC. However, the inner optimization subproblem is difficult due to $\boldsymbol{\alpha}^\top \boldsymbol{\alpha} = 1$ instead of $\boldsymbol{\alpha}^\top \mathbf{1}_n = 1$ in LGMMC, and we decide to investigate how to solve it in our future study since the computational efficiency is not the main focus of the current research.

In our experiments we always use the best candidate hyperparameters in the hindsight, since there lacks a systematic way to tune the hyperparameters for clustering. Such choices may be okay from the theoretical standpoint but not enough from the practical standpoint. Notice that any validation technique using the clustering error, which is the in-sample test error on the same data to be clustered, simply does not work here. In order to do model selection, a criterion other than the clustering error is required. Fortunately, such criteria exist though they are not uniformly effective for clustering. In Sugiyama et al. (2011), the *mutual information* (MI) (Shannon, 1948b) was used for *MI based clustering* (Gomes et al., 2010) via *maximum likelihood MI* (Suzuki et al., 2008) for model selection, and likewise *squared-loss MI* (SMI) (Suzuki et al., 2009) was used for *SMI based clustering* (Sugiyama et al., 2011) via *least-squares MI* (Suzuki et al., 2009) for model selection.

It is unclear how to specify the input matrix Q appropriately for a given data set, including a proper similarity measure and the construction of Q from it. According to von Luxburg et al. (2012), the former issue is actually open for all clustering algorithms and it probably has no uniformly effective solution. For the latter issue, we advocate MVC-SL with $Q = L_{\text{sym}} + I_n/n$, where L_{sym} is the normalized graph Laplacian, and the underlying similarity measure can be any similarity suitable for spectral clustering. Note that MVC and NSC have the similar tendency of performance on several benchmark data sets since the same graph Laplacian matrix has been used, but MVC is slightly better. We summarize the corresponding experimental results in Table 5.1, that is, over all tasks how many

	IDA	USPS	MNIST	20News	Isolet
MVC-SL won NSC	3	5	4	6	5
MVC-SL tied NSC	5	1	2	2	1
MVC-SL lost NSC	2	0	0	1	0

Table 5.1: Summary of experimental results concerning MVC vs. NSC

times MVC-SL won/tied/lost NSC statistically significantly.

Then, it is still unsolved when we should use MVC-SL, and when we should use the family of MMC or other clustering algorithms. Unfortunately, there is no answer from a theoretical point of view since clustering has no supervision at all. Nevertheless, MVC-SL may work with high probability in practice if spectral clustering works. We argue that it is the minimization of negative ℓ_1-norm in MVC-SL whose non-sparse optimal solutions improve the performance of spectral clustering, as shown in panel (c) of Figure 2.6.

Theoretically speaking, it is unclear whether the better performance of MVC is actually attributed to the finite sample stability and the clustering error bound, and it is non-trivial to confirm that. The finite sample stability requires the exact locally optimal solution which is difficult from an algorithmic viewpoint, but all other requirements for the data to be clustered can easily be satisfied in practice. In order to test the tightness of the clustering error bound, experiments should be carried out using the same problem setting with the clustering error bound. We have considered this issue when we need to select proper similarity measures in our experiments.

5.2.2 Future Directions of SERAPH

Let us consider the kernel extension of SERAPH. Suppose that we have a kernel function $k : \mathbb{R}^m \times \mathbb{R}^m \mapsto \mathbb{R}$, and a basis $\mathcal{B} = \{\bar{x}_i \mid \bar{x}_i \in \mathbb{R}^m\}_{i=1}^b$ where most often but not necessarily $\mathcal{B} \subseteq \mathcal{X}$. Let the *empirical kernel map* (Schölkopf and Smola, 2001, p. 42) be

$$\begin{aligned} \phi : \mathbb{R}^m &\mapsto \mathbb{R}^b \\ x &\mapsto (k(x, \bar{x}_1), \ldots, k(x, \bar{x}_b))^\top. \end{aligned}$$

Under this scenario, we learn a Mahalanobis distance metric for $\phi(x), \phi(x') \in \mathbb{R}^b$ of the form

$$d(x, x') = \sqrt{(\phi(x) - \phi(x'))^\top A(\phi(x) - \phi(x'))},$$

where $A \in \mathbb{R}^{b \times b}$ is a symmetric positive semi-definite matrix to be learned.

Subsequently, we assume that $\mathcal{B} = \mathcal{X}$. Let $k_1, \ldots, k_n$ be the columns of the kernel matrix K, then for any $x_i, x_j \in \mathcal{X}$,

$$d(x_i, x_j) = \sqrt{(k_i - k_j)^\top A(k_i - k_j)}.$$

All components of SERAPH remain the same just by replacing $x_i \in \mathbb{R}^m$ with the corresponding $k_i \in \mathbb{R}^n$. The resultant metric $d(x, x')$ will be highly non-linear with respect to the original domain. Similarly, one can construct a kernel extension based on the *kernel PCA map* (Schölkopf and Smola, 2001, p. 43) defined as

$$\begin{aligned} \phi : \mathbb{R}^m &\mapsto \mathbb{R}^n \\ x &\mapsto K^{-1/2}(k(x, x_1), \ldots, k(x, x_n))^\top. \end{aligned}$$

SERAPH also has the manifold extension. Without loss of generality, we adopt the kernel matrix K as the adjacency matrix of the underlying similarity graph. Let $d_i = \sum_{j=1}^n K_{i,j}$ be the degree of x_i, and $D = \operatorname{diag}(d_1, \ldots, d_n)$ be the degree matrix, then the *unnormalized graph Laplacian* is given by $L = D - K$.

Let $P \in \mathbb{R}^{m' \times m}$ be the projection matrix associated with A such that $A = P^\top P$, $X = (x_1, \ldots, x_n)^\top \in \mathbb{R}^{n \times m}$ be the design matrix of $\mathcal{X}$, and $Z = (z_1, \ldots, z_n)^\top \in \mathbb{R}^{n \times m'}$ be the design matrix of the projected data such that $z_i = Px_i$ and $Z = XP^\top$. The manifold assumption suggests $\forall x, x' \in \mathcal{X}$, if x and x' are close, z and z' should also be close. As a result, we should minimize the *Laplacian regularization* term defined by

$$\mathcal{M}(A) = \operatorname{tr}(Z^\top L Z) = \operatorname{tr}(X^\top L X A).$$

More specifically, the similarity of x_i and x_j is measured by the value of the kernel function $K_{i,j} = k(x_i, x_j)$, whereas the dissimilarity of z_i and z_j is measured by the Euclidean distance $\|z_i - z_j\|_2$. The manifold assumption translates

into that $\|z_i - z_j\|_2^2$ should be penalized more for larger $K_{i,j}$ than smaller $K_{i,j}$. Consequently, we have

$$\begin{aligned}
\mathcal{M}(A) &= \frac{1}{2}\sum_{i,j=1}^{n} K_{i,j}\|z_i - z_j\|_2^2 \\
&= \sum_{i,j=1}^{n} K_{i,j}(z_i^\top z_i - z_i^\top z_j) \\
&= \sum_{i=1}^{n} d_i z_i^\top z_i - \sum_{i,j=1}^{n} K_{i,j} z_i^\top z_j \\
&= \mathrm{tr}(DZZ^\top) - \mathrm{tr}(KZZ^\top) \\
&= \mathrm{tr}(LZZ^\top) \\
&= \mathrm{tr}(X^\top LXA).
\end{aligned}$$

Note that $\mathcal{M}(A)$ is again linear with respect to A, and it will affect neither the convexity nor the Lipschitz continuity of the M-Step. Let $\omega \geq 0$ be a regularization parameter for $\mathcal{M}(A)$, then the optimization problem becomes

$$\max_A \ \mathcal{L}(A) - \omega\mathcal{M}(A),$$

and at each M-Step, we solve

$$\max_A \ \mathcal{F}(A) - \omega\mathcal{M}(A).$$

The gradient of $\mathcal{F}(A) - \omega\mathcal{M}(A)$ is given by

$$\begin{aligned}
\nabla\mathcal{F}(A) - \omega\nabla\mathcal{M}(A) = &-\sum\nolimits_{\mathcal{S}\cup\mathcal{D}} y_{i,j}(1 - p_{i,j}^A(y_{i,j}))(x_i - x_j)(x_i - x_j)^\top \\
&- \mu\sum\nolimits_{\mathcal{U}}\sum\nolimits_{y} yq(y \mid x_i, x_j)(1 - p_{i,j}^A(y))(x_i - x_j)(x_i - x_j)^\top \\
&- \lambda I_m - \omega X^\top LX.
\end{aligned}$$

It is possible to initialize $q(y \mid x_i, x_j)$ in other ways for the proposed EM-like algorithm. Currently, all unlabeled data are initialized as dissimilar by

$$q(y = -1 \mid x_i, x_j) = 1.$$

It may be better if we estimate the ratio of dissimilar data over similar data, and initialize $q(y \mid x_i, x_j)$ accordingly for each specific data set. Moreover, the probability parameterized by the metric is now logistic that is not robust against noise weak labels. It is interesting to find another probabilistic model that can balance the robustness and the statistical efficiency.

5.2.3 Future Directions of SMIR

As mentioned early, minimizing the squared difference of the probability $p(y \mid x)$ and its approximation $q(y \mid x; \boldsymbol{\alpha})$ as the loss function could achieve the optimal non-parametric convergence rate from $q(y \mid x; \boldsymbol{\alpha})$ to $p(y \mid x)$, though $q(y \mid x; \boldsymbol{\alpha}^*)$ might be negative or unnormalized. However, when taking the regularization for maximizing SMI into account, the order of the convergence rate may still be the same but the constant of the convergence rate should change. Our Theorem 4.4 has shown that under certain conditions, the order of the generalization error bound can change from $O(1/\sqrt{l})$ to $O(1/\sqrt{n})$. Hence, it is interesting to study whether unlabeled data could also improve the order of the convergence rate and under what conditions this phenomenon may happen.

The novel data-dependent generalization error bounds appeared in Theorem 4.4 can only be applied to the reduced SMIR method for binary classification. When Eq. (4.16) instead of Eq. (4.17) is used for the post-precessing, we can no longer prove a generalization error bound using the same technique. Moreover, for the original SMIR method for multi-class classification, generalization error bounds are unavailable even if there is no normalization as Eq. (4.17), since the existing inductive definitions of Rademacher complexity are all binary. Nevertheless, the existing generalization or transductive error bounds for previous geometric methods based on inductive or transductive Rademacher complexity cannot be readily extended to upper bound multi-class classification errors. Previous information-theoretic methods even do not possess a generalization error bound. Therefore, it would be a promising direction to investigate generalization error bounds for the original multi-class SMIR.

Bibliography

F. Agakov and D. Barber. Kernelized infomax clustering. In *Advances in Neural Information Processing Systems 18 (NIPS)*, 2006.

F. Agakov and D. Barber. Kernelized infomax clustering. In *Advances in Neural Information Processing Systems 19 (NIPS)*, 2007.

S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society, Series B*, 28(1):131–142, 1966.

E. Allwein, R. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1: 113–141, 2000.

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19 (NIPS)*, 2007.

D. Arthur and S. Vassilvitskii. How slow is the k-means method? In *Proceedings of 22nd ACM Symposium on Computational Geometry (SoCG)*, 2006.

M. Baghshah and S. Shouraki. Semi-supervised metric learning using pairwise constraints. In *Proceedings of 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

P. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2002.

M. Belkin, I. Matveeva, and P. Niyogi. Regularization and semi-supervised learning on large graphs. In *Proceedings of 15th International Conference on Algorithmic Learning Theory (ALT)*, 2004.

M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: a geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

K. Bellare, G. Druck, and A. McCallum. Alternating projections for learning with expectation constraints. In *Proceedings of 25th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

S. Ben-David, D. Pál, and H. U. Simon. Stability of k-means clustering. In *Proceedings of 20th Annual Conference on Learning Theory (COLT)*, 2007.

K. Bennett and A. Demiriz. Semi-supervised support vector machines. In *Advances in Neural Information Processing Systems 11 (NIPS)*, 1999.

A. Berger, S. Pietra, and V. Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22:39–71, 1996.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta Numerica*, 4:1–51, 1995.

B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of 5th Annual Workshop on Computational Learning Theory (COLT)*, 1992.

O. Bousquet, S. Boucheron, and G. Lugosi. Introduction to statistical learning theory. In O. Bousquet, U. von Luxburg, and G. Rätsch, editors, *Advanced Lectures on Machine Learning*, pages 169–207. Springer, 2004.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

O. Chapelle, J. Weston, and B. Schölkopf. Cluster kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems 15 (NIPS)*, 2003.

O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006.

F. Chiaromonte and R. Cook. Sufficient dimension reduction and graphics in regression. *Annals of the Institute of Statistical Mathematics*, 54:768–795, 2002.

Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3): 273–297, 1995.

C. Cortes, M. Mohri, D. Pechyony, and A. Rastogi. Stability of transductive regression algorithms. In *Proceedings of 25th International Conference on Machine Learning (ICML)*, 2008.

K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

I. Csiszár. Information-type measures of difference of probability distributions and indirect observation. *Studia Scientiarum Mathematicarum Hungarica*, 2: 229–318, 1967.

J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Proceedings of 24th International Conference on Machine Learning (ICML)*, 2007.

T. De Bie and N. Cristianini. Convex methods for transduction. In *Advances in Neural Information Processing Systems 16 (NIPS)*, 2004.

T. De Bie and N. Cristianini. Fast SDP relaxations of graph cut clustering, transduction, and other combinatorial problems. *Journal of Machine Learning Research*, 7:1409–1436, 2006.

O. Delalleau, Y. Bengio, and N. Le Roux. Efficient non-parametric function induction in semi-supervised learning. In *Proceedings of 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2005.

C. Ding and X. He. K-means clustering via principal component analysis. In *Proceedings of 21st International Conference on Machine Learning (ICML)*, 2004.

R. Duda, P. Hart, and D. Stork. *Pattern Classification (2nd edition)*. John Wiley & Sons, 2001.

M. Dudík and R. E. Schapire. Maximum entropy distribution estimation with generalized regularization. In *Proceedings of 19th Annual Conference on Learning Theory (COLT)*, 2006.

R. El-Yaniv and D. Pechyony. Transductive Rademacher complexity and its applications. *Journal of Artificial Intelligence Research*, 35:193–234, 2009.

R. El-Yaniv, D. Pechyony, and V. Vapnik. Large margin vs. large volume in transductive learning. *Machine Learning*, 72(3):173–188, 2008.

L. Faivishevsky and J. Goldberger. A nonparametric information theoretic clustering algorithm. In *Proceedings of 27th International Conference on Machine Learning (ICML)*, 2010.

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

K. Fukumizu, F. Bach, and M. I. Jordan. Kernel dimension reduction in regression. *Annals of Statistics*, 37(4):1871–1905, 2009.

J. Gillenwater, K. Ganchev, J. Graça, F. Pereira, and B. Taskar. Posterior sparsity in usupervised dependency parsing. *Journal of Machine Learning Research*, 12:455–490, 2011.

M. Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Advances in Neural Information Processing Systems 18 (NIPS)*, 2006.

J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighbourhood components analysis. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.

R. Gomes, A. Krause, and P. Perona. Discriminative clustering by regularized information maximization. In *Advances in Neural Information Processing Systems 23 (NIPS)*, 2010.

J. Graça, K. Ganchev, and B. Taskar. Expectation maximization and posterior constraints. In *Advances in Neural Information Processing Systems 20 (NIPS)*, 2008.

J. Graça, K. Ganchev, B. Taskar, and F. Pereira. Posterior vs. parameter sparsity in latent variable models. In *Advances in Neural Information Processing Systems 22 (NIPS)*, 2009.

Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.

Y. Grandvalet and Y. Bengio. Entropy regularization. In O. Chapelle, B. Schölkopf, and A. Zien, editors, *Semi-Supervised Learning*, pages 151–168. MIT Press, 2006.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 1.21 (web page and software). http://cvxr.com/cvx, 2011.

M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of 17th International Conference on Machine Learning (ICML)*, 2000.

J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28:100–108, 1979.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition)*. Springer Verlag, 2009.

S. Hoi, W. Liu, and S.-F. Chang. Semi-supervised distance metric learning for collaborative image retrieval. In *Proceedings of 21st IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

K. Huang, Y. Ying, and C. Campbell. GSML: A unified framework for sparse metric learning. In *Proceedings of 9th IEEE International Conference on Data Mining (ICDM)*, 2009.

K. Huang, R. Jin, Z. Xu, and C. Liu. Robust metric learning by smooth optimization. In *Proceedings of 26th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2010.

E. T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, 1957.

T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of 16th International Conference on Machine Learning (ICML)*, 1999.

T. Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of 20th International Conference on Machine Learning (ICML)*, 2003.

T. Kanamori, S. Hido, and M. Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10:1391–1445, 2009.

T. Kanamori, T. Suzuki, and M. Sugiyama. Statistical analysis of kernel-based least-squares density-ratio estimation. *Machine Learning*, 86(3):335–367, 2012.

M. Kearns, Y. Mansour, A. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning*, 27:7–50, 1997.

V. Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

G. R.G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Y. Li, I. W. Tsang, J. T. Kwok, and Z.-H. Zhou. Tighter and convex maximum margin clustering. In *Proceedings of 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2009.

W. Liu, S. Ma, D. Tao, J. Liu, and P. Liu. Semi-supervised sparse metric learning using alternating linearization optimization. In *Proceedings of 16th ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2010.

J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

G. Mann and A. McCallum. Simple, robust, scalable semi-supervised learning via expectation regularization. In *Proceedings of 24th International Conference on Machine Learning (ICML)*, 2007.

C. McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics*, pages 148–188. Cambridge University Press, 1989.

M. Meila and J. Shi. A random walks view of spectral segmentation. In *Proceedings of 8th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2001.

R. Meir and T. Zhang. Generalization error bounds for Bayesian mixture algorithms. *Journal of Machine Learning Research*, 4:839–860, 2003.

T. M. Mitchell. The discipline of machine learning. Technical Report CMU-ML-06-108, Carnegie Mellon University, 2006.

A. Ng, M. I. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2002.

G. Niu, B. Dai, M. Yamada, and M. Sugiyama. Information-theoretic semi-supervised metric learning via entropy regularization. In *Proceedings of 29th International Conference on Machine Learning (ICML)*, 2012.

K. Pearson. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *Philosophical Magazine*, 50: 157–175, 1900.

B. T. Polyak. A general method for solving extremal problems (in Russian). *Soviet Mathematics Doklady*, 174(1):33–36, 1967.

P. Raghavan and C. Thompson. Randomized rounding: A technique for provably good algorithms and algorithmic proofs. Technical Report UCB/CSD-85-242, UC Berkeley, 1985.

A. Rakhlin and A. Caponnetto. Stability of k-means clustering. In *Advances in Neural Information Processing Systems 19 (NIPS)*, 2007.

S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach (3rd edition)*. Prentice Hall, 2009.

B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, 2001.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 & 623–656, 1948a.

C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 & 623–656, 1948b.

H. Sherali and W. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*. Kluwer Academic Publishers, 1998.

J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

K. Sinha and M. Belkin. Semi-supervised learning using sparse eigenfunction bases. In *Advances in Neural Information Processing Systems 22 (NIPS)*, 2009.

A. Smola, S.V.N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proceedings of 10th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2005.

L. Song, A. Smola, A. Gretton, and K. Borgwardt. A dependence maximization view of clustering. In *Proceedings of 24th International Conference on Machine Learning (ICML)*, 2007.

M. Sugiyama. *A theory of model selection and active learning for supervised learning*. PhD thesis, Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, January 2001.

M. Sugiyama. Active learning in approximately linear regression based on conditional expectation of generalization error. *Journal of Machine Learning Research*, 7:141–166, 2006.

M. Sugiyama. Dimensionality reduction of multimodal labeled data by local Fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027–1061, 2007.

M. Sugiyama, T. Idé, S. Nakajima, and J. Sese. Semi-supervised local Fisher discriminant analysis for dimensionality reduction. *Machine Learning*, 78(1-2):35–61, 2010.

M. Sugiyama, M. Yamada, M. Kimura, and H. Hachiya. On information-maximization clustering: Tuning parameter selection and analytic solution. In *Proceedings of 28th International Conference on Machine Learning (ICML)*, 2011.

R. Sutton and G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

T. Suzuki, M. Sugiyama, J. Sese, and T. Kanamori. Approximating mutual information by maximum likelihood density ratio estimation. In *JMLR Workshop and Conference Proceedings*, volume 4, pages 5–20, 2008.

T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10(1):S52, 2009.

M. Szummer and T. Jaakkola. Partially labeled classification with Markov random walks. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2002.

M. Szummer and T. Jaakkola. Information regularization with partially labeled data. In *Advances in Neural Information Processing Systems 15 (NIPS)*, 2003.

J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

L. Torresani and K. Lee. Large margin component analysis. In *Advances in Neural Information Processing Systems 19 (NIPS)*, 2007.

H. Valizadegan and R. Jin. Generalized maximum margin clustering and unsupervised kernel learning. In *Advances in Neural Information Processing Systems 19 (NIPS)*, 2007.

V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer Verlag, 1982.

V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

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

U. von Luxburg, O. Bousquet, and M. Belkin. Limits of spectral clustering. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.

U. von Luxburg, M. Belkin, and O. Bousquet. Consistency of spectral clustering. *Annals of Statistics*, 36(2):555–586, 2008.

U. von Luxburg, R. Williamson, and I. Guyon. Clustering: Science or art? In *JMLR Workshop and Conference Proceedings*, volume 27, pages 65–80, 2012.

F. Wang, B. Zhao, and C. Zhang. Linear time maximum margin clustering. *IEEE Transactions on Neural Networks*, 21(2):319–332, 2010.

K. Weinberger, J. Blitzer, and L. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18 (NIPS)*, 2006.

E. Xing, A. Ng, M. I. Jordan, and S. Russell. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems 15 (NIPS)*, 2003.

L. Xu and D. Schuurmans. Unsupervised and semi-supervised multi-class support vector machines. In *Proceedings of 20th National Conference on Artificial Intelligence (AAAI)*, 2005.

L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.

M. Yamada, M. Sugiyama, G. Wichern, and J. Simm. Improving the accuracy of least-squares probabilistic classifiers. *IEICE Transactions on Information and Systems*, E94-D(6):1337–1340, 2011.

L. Yang, R. Jin, R. Sukthankar, and Y. Liu. An efficient algorithm for local distance metric learning. In *Proceedings of 21st National Conference on Artificial Intelligence (AAAI)*, 2006.

Y. Ying, K. Huang, and C. Campbell. Sparse metric learning via smooth optimization. In *Advances in Neural Information Processing Systems 22 (NIPS)*, 2009.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society*, Series B, 68(1):49–67, 2006.

A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

L. Zelnik-Manor and P. Perona. Self-tuning spectral clustering. In *Advances in Neural Information Processing Systems 17 (NIPS)*, 2005.

H. Zha, X. He, C. Ding, M. Gu, and H. Simon. Spectral relaxation for k-means clustering. In *Advances in Neural Information Processing Systems 14 (NIPS)*, 2002.

Z. Zha, T. Mei, M. Wang, Z. Wang, and X. Hua. Robust distance metric learning with auxiliary knowledge. In *Proceedings of 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 2009.

K. Zhang, I. W. Tsang, and J. T. Kwok. Maximum margin clustering made practical. In *Proceedings of 24th International Conference on Machine Learning (ICML)*, 2007.

B. Zhao, F. Wang, and C. Zhang. Efficient multiclass maximum margin clustering. In *Proceedings of 25th International Conference on Machine Learning (ICML)*, 2008a.

B. Zhao, F. Wang, and C. Zhang. Efficient maximum margin clustering via cutting plane algorithm. In *Proceedings of 8th SIAM International Conference on Data Mining (SDM)*, 2008b.

D. Zhou, O. Bousquet, T. Navin Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16 (NIPS)*, 2004.

X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using Gaussian fields and harmonic functions. In *Proceedings of 20th International Conference on Machine Learning (ICML)*, 2003.