Heterogeneous Learning and Its Applications

BY

XIAOXIAO SHI
B.E., Sun Yat-sen University, China, 2007
M.S., Sun Yat-sen University, China, 2009
M.S., Applied Math, University of Illinois at Chicago, 2012

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Chicago, 2013

Chicago, Illinois

Defense Committee:

Philip S. Yu, Chair and Advisor
Bing Liu, Professor at Computer Science Dept.
John Lillis, Professor at Computer Science Dept.
Junhui Wang, Assistant Professor at Dept. of Math, Statistics, and Computer Science
Jing Wang, Assistant Professor at Dept. of Math, Statistics, and Computer Science

To Wanying Zhang,

my ever supportive, always faithful, loving wife.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Prof. Philip S. Yu, for his continual support and advice throughout my Ph.D. study, research and life. His knowledge and kindness never ceased to impress and educate me. Without him this work could not have been completed. I would also like to thank my collaborators including Dr. Wei Fan, Prof. Qiang Yang, Dr. Qi Liu, Dr. Kevin Chang, Dr. Vijay Narayanan, Dr. Vanja Josifovski, Dr. Jean-Francois Paiement and Prof. Alex Smola. They enrich my research work by providing various insightful and valuable opinions and discussions. Furthermore, I would like to thank my committee members for their support and suggestions to improve this work.

XS

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

SUMMARY

With the rapid growth of big data mining, multiple related data sources containing different types of features may be available for a given task. For instance, users' profiles can be used to build recommendation systems; in addition, a model can also use users' historical behaviors and social networks to infer users' interests on related products. We argue that it is desirable to collectively use any available multiple heterogeneous data sources in order to build effective learning models. We call this framework *heterogeneous learning*.

There are mainly two challenges in heterogeneous learning as follows:

- Learning from data with different statistical properties. For example, the data from different data sources violate the iid assumption (53)(50), or the data from different sources have different feature spaces (58)(60)(59), or the data have different prediction labels (different posterior) (54), or the combination of the above cases (59).
- Learning from data with different structures. For example, some of the data sources contain traditional vector-based features (e.g., user profiles), while others are graph relational data (e.g., social networks) (31)(55)(57)(61), or the data sources are chemical graphs with different structures (56).

In this thesis, we explore the above challenges from the views of supervised learning, unsupervised learning and feature projection respectively, and apply them to solve real world problems. These real world applications include drug efficiency prediction (58)(60)(59), document classification (54)(53), image classification (54), movie rating prediction (61), chemical graph classification (56), collective classification (31)(57), and several datasets from the UCI database (50)(59). It shows that heterogeneous

learning improves the learning accuracy significantly in some applications. For example, in the task of drug efficiency prediction, heterogeneous learning can reduce the error rate by over 50% by using a projection approach.

CHAPTER 1

INTRODUCTION

In traditional supervised and semi-supervised learning, there is usually a large gap between the number of labeled examples needed to obtain high prediction accuracy, and the number of labeled examples that could be realistically obtained. These problems can be found in web mining, behavior targeting, spam filtering, objective recognition, and bioinformatics applications. At the same time, however, there is usually a large number of labeled examples, or other supervision knowledge (e.g., linkage information) from various related applications, such as, labeled documents in social tagging systems (e.g., wikipedia, ODP) for web mining, labeled chemical compounds from NCI database for bioinformatics, and classified images from social sites such as Flickr for object recognition and image classification. With the rapid growth of big data mining techniques, the above phenomena can be found in various applications; that is, multiple related data sources may be available for a given task. One may ask: can these available source data provide useful supervision to a related target task? Four challenging sub-issues need to be solved:

1. The source data may be drawn from a distribution different from the target data. For example, the source data is dominated by a Gaussian distribution while the target data is dominated by a multinomial distribution, which violates the i.i.d. assumption.

2. The source and target data may have totally different output spaces. For example, the regression of stock price is totally different from the regression of the foreign exchange price, although they are correlated.

3. The source data may be generated from a different feature space from the target data (e.g., source is text data while target is image data).

4. The source and target data may have different structures. For example, the target data may be graph data, while the source may contain vector-based data.

Among the four issues, the last three issues (e.g. unifying heterogeneous outputs, features, and learning from data with different structures) are most challenging, which, so far as we know, has not been specifically addressed systematically. In this thesis, we explore solutions to solve the above issues, and introduce several related applications.

1.1 Dissertation Framework

We explore the problem of heterogeneous learning mainly in views of supervised learning, unsupervised learning and feature projection. We first explore the related works of the problem in Charter 2. We then construct the dissertation as the flow in Fig. 1. More specifically, we first propose a general supervised heterogeneous learning framework called **G**radient **B**oosting **C**onsensus (GBC) to improve the accuracy of learning models by using multiple heterogeneous datasets. We then explore an unsupervised learning model called **Co**llective **C**omponent **A**nalysis (CoCA) and discuss its applications. Furthermore, a projection based algorithm HeMap is analyzed in heterogeneous learning with its applications in bioinformatics. Finally, we conclude the paper by summarizing the key techniques and principles.

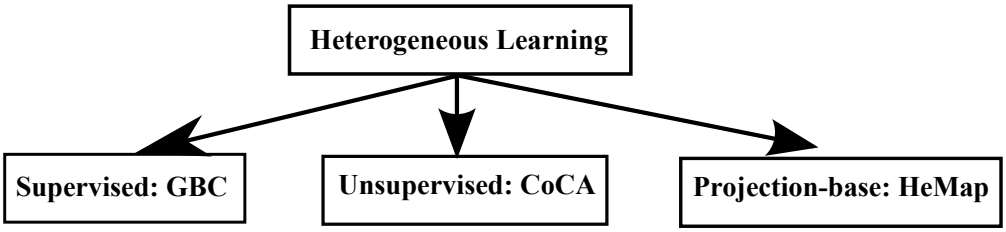


Figure 1. Dissertation framework.

1.2 Heterogeneous Learning

With the exploration of big data analysis, heterogeneous learning attracts more and more attentions in recent years. In addition to the technical challenges, we can also analyze the area from the view of supervised learning, unsupervised learning, and its extension in feature projection.

1.2.1 Supervised Heterogeneous Learning via GBC

Multiple data sources containing different types of features may be available for a given task. For instance, users' profiles can be used to build recommendation systems. In addition, a model can also use users' historical behaviors and social networks to infer users' interests on related products. We argue that it is desirable to collectively use any available multiple heterogeneous data sources in order to build effective learning models, even though they have different data structures. In our proposed setting, data sources can include (i) non-overlapping features, (ii) non-overlapping instances, and (iii) multiple networks (i.e. graphs) that connect instances. In Chapter 3, we propose a general optimization framework for learning from heterogeneous structures, and devise a corresponding learning model from gradient boosting. The idea is to minimize the empirical loss with two constraints: (1) There should be consensus among the predictions of overlapping instances (if any) from different data sources; (2) Connected

instances in graph datasets may have similar predictions. The objective function is solved by stochastic gradient boosting trees. Furthermore, a weighting strategy is designed to emphasize informative data sources, and deemphasize the noisy ones. We formally prove that the proposed strategy leads to a tighter error bound. This approach consistently outperforms a standard concatenation of data sources on movie rating prediction, number recognition and terrorist attack detection tasks. We observe that the proposed model can improve out-of-sample error rate by as much as 80%.

1.2.2 Unsupervised Heterogeneous Learning via CoCA

Combining correlated data sources may help improve the learning performance of a given task. For example, in recommendation problems, one can combine (1) user profile database (e.g. genders, age, etc.), (2) users' log data (e.g., clickthrough data, purchasing records, etc.), and (3) users' social network (useful in social targeting) to build a recommendation model. All these data sources provide informative but heterogeneous features. For instance, user profile database usually has nominal features reflecting users' background, log data provides term-based features about users' historical behaviors, and social network database has graph relational features. Given multiple heterogeneous data sources, one important challenge is to find a unified feature subspace that captures the knowledge from all sources. To this aim, Chapter 4 proposes a principle of *collective component analysis* (CoCA), in order to capture the correlations across a mixture of vector-based features and graph relational features. The CoCA principle is to find a feature subspace with maximal variance under two constraints. First, there should be consensus among the projections from different feature spaces. Second, the similarity between connected data (in any of the network databases) should be maximized. The optimal solution is obtained by solving an

eigenvalue problem. Moreover, we discuss how to use prior knowledge to distinguish informative data sources, and optimally weight them in CoCA.

1.2.3 Heterogeneous Projection via HeMap

Labeled examples are often expensive and time-consuming to obtain. One practically important problem is: can the labeled data from other related sources help predict the target task, even if they have (a) different feature spaces (e.g., image vs. text data), (b) different data distributions, and (c) different output spaces? Among the three problems, learning with different feature spaces is the most challenging one. Chapter 5 proposes a solution and discusses the conditions where this is possible and highly likely to produce better results. It works by first using spectral embedding to unify the different feature spaces of the target and source data sets, even when they have completely different feature spaces. The principle is to cast into an optimization objective that preserves the original structure of the data, while at the same time, maximizes the similarity between the two. Second, a judicious sample selection strategy is applied to select only those related source examples. At last, a Bayesian-based approach is applied to model the relationship between different output spaces. The three steps can bridge related heterogeneous sources in order to learn the target task. Among the 12 experiment data sets, for example, the images with wavelet-transformed-based features are used to predict another set of images whose features are constructed from color-histogram space. By using these extracted examples from heterogeneous sources, the models can reduce the error rate by as much as 50%, compared with the methods using only the examples from the target task.

CHAPTER 2

RELATED WORK

There are several lines of related works that we build upon. The first one is Multi-view learning (e.g., (7)(45)). For example, in (45), a co-training algorithm is proposed to classify the web pages by the text on the web page, and the text on hyperlinks pointing to the web page. Furthermore, there is another type of method called canonical correlation analysis (e.g., (64)), which aims at finding a hidden space to maximize the correlation between the data with different types of features. However, it is assumed that (1) the data from different views are the same set of objects, and that (2) the data correspondences are already given. In other words, the multi-view data are the same set of objects only with different feature spaces. Different from these works, we study the problem of unifying the feature spaces of two sets of instances, and there may be no correspondence between instances in these spaces.

Another related field is transfer learning (e.g., (19)(2)(47)(54)) which aims at learning the target task from a related out-of-domain source task. A survey of transfer learning can be found at (48). One line of research is to find a new feature space in which the training and test data have strong similarities (e.g., (10)(27)(36)(28)(70)). For instance, (2) applies sparse learning techniques to transfer the knowledge across multiple tasks. Furthermore, (47) proposes to find a RKHS feature space to enable transfer learning via maximum mean discrepancy. Recently, (70) proposes a projection model to find the alignment of manifolds to enable knowledge transfer. Different from this work, our projection-based model HeMap does not just force the projections from different domains to be totally the same. Instead, we also require the projections to keep the original structure of the data in order to reduce

negative transfer. Furthermore, (14) is among the first works to exploit using second-order logic for transfer learning based on relational network data. There is also a set of works enable transfer learning on document-related tasks. For example, (27) finds the commonality among different tasks, such as common words, to attack NLP tasks. However, documents are usually viewed as coming from the same feature space, since any document can be represented as "bag of words" where the dictionary contains all possible words. Different from these works, we do not require the original training and test data are in the same feature space, or have a subset of common features. Instead, they can be from completely different feature spaces such as the image and text data. Recently, (72)(9)(81) propose methods to improve image clustering and classification with the help of text data. The major idea is to take advantage of the image tags to build up the relationship between image and text, and further use text features to enrich image features to improve the learning accuracy. There are at least two differences between this paper and (72)(9)(81). First, the proposed models aim at studying in a more *general* setting which is not limited to image and text data. Second, no auxiliary source (such as image tags) is given to provide clue of the correlation among the instances in different feature spaces. In addition to the feature based transfer learning techniques, the proposed model is also related to a set of research works that focus on knowledge transfer among data sets with different outputs (e.g., (54)(49)). For example, (49) studies the problem of using arbitrary images to help find a more robust basis for a new image classification task. (54) finds a latent space for the outputs and tries to align the different outputs in the latent space.

The third related area is graph mining. With the explosive development of social network, graph based approaches attract intensive attentions in recent years. Several approaches have been proposed

to perform inference on a given network (e.g., (78)(71)(73)(76)(41)(75)). For instance, (77) proposes to propagate the labels through the network, and assigns each node with the label that is most likely to be visited from the node. However, this category of algorithms only considers the graph structure or only consider the feature vectors, and it cannot handle the case directly when there is feature vector in a graph. For example, in a citation network, each node is a paper and the links represent a citation relationship. In addition to the linkage information among the nodes, we also have the "bag of words" feature vector for each node. Traditional graph based models cannot directly handle this kind of network data with feature vectors. Collective classification is then proposed to solve the problem. Its key idea is to combine the supervision knowledge from the traditional tuple-based feature vectors, as well as the linkage information from the network. It has been applied to various applications such as part-of-speech tagging (34), classification of hypertext documents using hyperlinks (65)(18), link prediction in friend-of-a-friend networks (67), predicting disulphide bonds in protein molecules (66), etc.. Iterative Classification Algorithm (ICA (43)) is among the first works proposed to tackle collective classification. The key step is to transform the network summary into feature vectors and treat the network as ordinary features. It is reported in (22) that ICA is a fairly accurate method with robust performance to different strategies of updating the labels. Moreover, Gibbs sampling (29) is further integrated into the ICA framework to enrich the statistical foundation of the algorithm. This is the major reason that we choose ICA and Gibbs sampling as the major collective classifiers to be compared with. In recent years, there are several works proposed to use a similar schema as ICA but with different basic classifiers or in different scenarios (29)(44)(65)(40). In this thesis, we adopt the idea of ICA, but study a more general problem where there are multiple sets of relational and unrelational data.

CHAPTER 3

SUPERVISED HETEROGENEOUS LEARNING VIA GRADIENT BOOSTING CONSENSUS

In this chapter, we explore a supervised gradient boosting model to learn from multiple heterogeneous datasets.

3.1 Introduction

With the rapid development of big data technologies, multiple related data sources can be used to build prediction models given a target task. Each of the related data sources may have a distinct set of features and instances, and we argue that the combination of all data sources may yield better prediction results. An example is illustrated in Fig. 16. The task is to predict movie ratings in the Internet Movie Database (IMDB[1]), which has been used in movie recommendation (42). For example, in Fig. 2(a), given that we observe that the rating for "The Godfather" is 9.2 (out of 10), and "The Giant Spider Invasion" is 2.8, what are the ratings for "Apocalypse Now" and "Monster a-Go Go"? Note that in this task, there are multiple available databases that record various information about movies. For instance, there is a genre database (Fig. 2(b)), a sound technique database (Fig. 2(c)), a running times database (Fig. 2(d)), an actor graph database that links two movies together if the same actor/actress performs in the movies (Fig. 2(e)), and a director graph database that links two movies if they are directed by the same director (Fig. 2(f)). Note that these multiple data sources have the following properties:

[1] http://www.imdb.com/

Name	Ratings
The Godfather	9.2
Apocalypse Now	?
Monster a-Go Go	?
The Giant Spider Invasion	2.8

(a) Movie rating prediction.

Name	Genre
The Godfather	Drama, Crime
Apocalypse Now	Drama, War
The Giant Spider Invasion	Horror, Sci-Fi

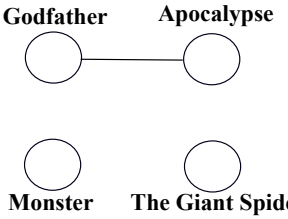
(b) Genre database.

Name	Sound Technique
Apocalypse Now	DTS, Digital, 6-Track
Monster a-Go Go	Mono
The Giant Spider Invasion	Mono

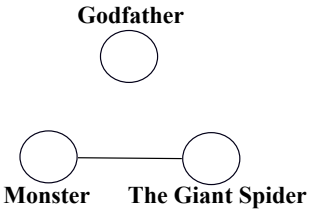
(c) Sound technique database.

Name	Running times (mins)
The Godfather	175
Monster a-Go Go	70
The Giant Spider Invasion	84

(d) Running times.



(e) Actor graph.



(f) Director graph that does not have record on "Apocalypse Now".

Figure 2. Combining different sources to infer movie ratings. The true rating for "Apocalypse Now" is 8.6, while the rating for "Monster a-Go Go" is 1.5.

- Firstly, each data source can have its own feature sets. For example, the running times database (Fig. 2(d)) has numerical features; the genre database (Fig. 2(b)) has nominal features, and the actor graph database (Fig. 2(e)) provides graph relational features.
- Secondly, each data source can have its own set of instances. For example, the genre database does not have the record for "Monster a-Go Go"; the running times database does not have any record of "Apocalypse Now".

Note that it is difficult to build an accurate prediction model by using only one of the five databases, since the information in each of them is incomplete. However, if we consider the five data sources collectively, we are able to infer that the rating of "Apocalypse Now" (ground truth: 8.6) may be close to that of "The Godfather", since they are similar in genre and they are connected in the actor graph. Similarly, one can infer that the rating for "Monster a-Go Go" (ground truth: 1.5) is similar to that of "The Giant Spider Invasion".

In the past, multi-view learning (7; 46) was proposed to study a related problem where each instance can have different views. However, it usually does not consider graph data with relational features, especially when there are multiple graphs and each graph may only contain a subset of the relation features. Hence, we study a more general learning scenario called *heterogeneous learning* where the data can come from multiple sources. Specifically, the data sources can (1) have unique feature spaces (i.e., new features in certain data sources), (2) have some incomplete instances (i.e., instances do not record in some data sources), and (3) contain multiple network (i.e. weighted graphs) datasets. Furthermore, some of the data sources may contain substantial noise or low-quality data. Our aim is to utilize all data sources collectively and judiciously, in order to improve the learning performance.

A general objective function is proposed to make good use of the information from these multiple data sources. The intuition is to learn a prediction function from each data source to minimize the empirical loss with two constraints. First, the predictions of the same instance should be similar even when learning from different data sources. Second, the predictions of connected data (i.e., instances connected in any of the graphs) should be similar. Finally, the prediction models are judiciously combined (with different weights) to generate a global prediction model. In order to solve the objective function, we

borrow ideas from gradient boosting decision trees (GBDT), which is an iterated algorithm that generates a sequence of decision trees, where each tree fits the gradient residual of the objective function. We call our proposed algorithm Gradient Boosting Consensus (GBC) because each data source generates a set of trees, and the consensus of the decision trees makes the final prediction. Moreover, GBC has the following properties.

- Deep-ensemble. Recall that the traditional boosting tree model is an iterated algorithm that builds new trees based on the previous iterations (residuals). Usually, these new trees are generated based on the residual of only one data source. However, as shown in Fig. 3, GBC generates new trees collectively from all data sources (horizontally) in each iteration (vertically). We call it "deep ensemble" since it ensembles models both horizontally and vertically to make the final prediction.
- Network-friendly. Unlike traditional boosting trees, GBC can take advantage of multiple graph datasets to improve learning. In other words, it can take advantage of traditional vector-based features and graph relational features simultaneously.
- Robust. Some data sources may contain substantial noise. A weighting strategy is incorporated into GBC to emphasize informative data sources and deemphasize the noisy ones. This weighting strategy is further proven to have a tighter error bound in both inductive and transductive settings.

We conducted four sets of experiments. These experiments include IMDB movie rating prediction, UCI number recognition, terrorist attack detection, and a demographic prediction task in a big dataset from a nationa wide phone provider with over 500,000 samples, 91 different data sources, and over

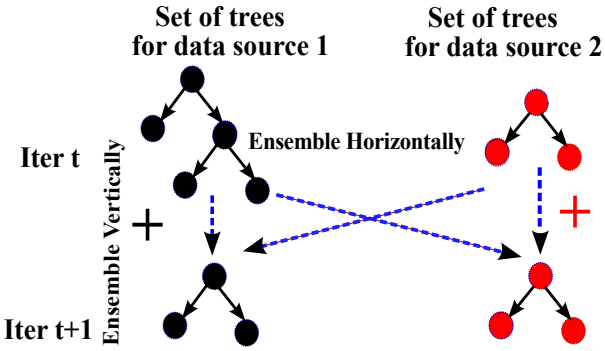


Figure 3. Gradient Boosting Consensus.

45,000,000 joined features. Each task has a set of data sources with heterogeneous features. For example, in the IMDB movie rating prediction task, we have data sources about the plots of the movies (text data), technologies used by the movies (nominal features), running times of the movies (numerical features), and several movie graphs (such as director graph, actor graph). All these mixture types of data sources were used collectively to build a prediction model. Since there is no previous model that can handle the problem directly, we have constructed a straightforward baseline which first appends all data sources together into a single database, and uses traditional learning models to make predictions. Experiments show that the proposed GBC model consistently outperforms our baseline, and can decrease the error rate by as much as 80%.

3.2 Related Work for GBC

There are several areas of related works upon which our proposed model is built. First, multi-view learning (e.g., (7; 45; 38)) is proposed to learn from instances which have multiple views in different feature spaces. For example, in (45), a co-training algorithm is proposed to classify the web pages

by the text on the web page, and the text on hyperlinks pointing to the web page. In (38), a general clustering framework is proposed to reconcile the clustering results from different views. In (20), a term called consensus learning is proposed. The general idea is to perform learning on each heterogeneous feature space independently and then summarize the results via ensemble. Recently, (1) proposes a recommendation model (collaborative filtering) that can combine information from different contexts. It finds a latent factor that connects all data sources, and propagate information through the latent factor. There are mainly two differences between our work and the previous approaches. First, most of the previous works do not consider the vector-based features and the relational features simultaneously. Second and foremost, most of the previous works require the data sources to have records of all instances in order to enable the mapping, while the proposed GBC model does not have this constraint.

Another area of related work is collective classification (e.g., (51)) that aims at predicting the class label from a network. Its key idea is to combine the supervision knowledge from traditional vector-based feature vectors, as well as the linkage information from the network. It has been applied to various applications such as part-of-speech tagging (34), classification of hypertext documents using hyperlinks (65), etc. Most of these works study the case when there is only one vector-based feature space and only one relational feature space, and the focus is how to combine the two. Different from traditional collective classification framework, we consider multiple vector-based features and multiple relational features simultaneously. Specifically, (15) proposes an approach to combine multiple graphs to improve the learning. The basic idea is to average the predictions during training. There are three differences between the previous works and the current model. Firstly, we allow different data sources to have new instances that do not record in other data sources. Secondly, we introduce a weight learning

process to filter out noisy data sources. Thirdly, we consider *multiple* vector-based sources and *multiple* graphs at the same time. Hence, all the aforementioned methods could not effectively learn from the datasets described in Section 4, as they all contain multiple vector-based data sources and relational graphs.

Another related field is transfer learning (e.g., (6; 8; 69; 60; 4)) which aims at learning the target task from a related out-of-domain source task. Most research work on transfer learning focus on how to make good use of the training data that distributes differently with the test data. A general approach is based on re-sampling (e.g., (6)), where the motivation of it is to "emphasize" the knowledge among "similar" and discriminating instances. Another line of research is to find a new feature space in which the training and test data have strong similarities (e.g., (10; 27; 36; 28; 70)). For instance, (2) applies sparse learning techniques to transfer the knowledge across multiple tasks. There is also a set of works that enable transfer learning on document-related tasks. For example, (27) finds the commonality among different tasks, such as common words, to attack NLP tasks. However, documents are usually viewed as coming from the same feature space, since any document can be represented as "bag of words" where the dictionary contains all possible words. Different from these works, we do not require the original training and test datasets to be in the same feature space, or have a subset of common features. Instead, they can be from completely different feature spaces, and they can even be graph datasets.

The proposed model is also related to the research of ensemble learning. For example, gradient boosting tree (17) is a work proposed by J. H. Friedman to approach the learning objective by building an ensemble of weak learners (i.e., decision tree in this case). It is achieved by fitting a decision tree to the residual error of the model at each iteration, and the final model is composed of a weighted summation

of all the fitted trees. GBDT is a new model widely used in the industry. For instance, it is used in Yahoo and the search engine company Yandex in the field of "learning to rank" (37); it is also adopted by the winning team of the Netflix competition (32). In this paper, we borrow the idea of gradient boosting to solve the problem of aggregating multiple heterogeneous data sources. There are mainly two reasons that we choose this technique. First, in heterogeneous learning, ensemble approach is a natural choice since it is not straightforward to come up with a single model to deal with multiple heterogeneous data sources. Second, it is required that the prediction model to be efficient since large dataset is used as input. As such, gradient boosting tree is a clear option, especially given that it can be easily deployed in a distributed computing environment. Note that there are some other ensemble learning researches developed with the idea of consensus. For instance, (21) is proposed to perform clustering via an ensemble learning approach. The basic idea is to model the pairwise relationships from multiple sources, and construct the "belief" graphs that maximizes the consensus among the data sources. Furthermore, a generalized unsupervised learning model is proposed in (62), which also adopts the idea of aggregating multiple data sources via consensus principle. However, so far as we know, although there are several ensemble methods proposed to aggregate heterogeneous sources in the unsupervised learning framework (as in (21; 62)), there is only a few proposed to conduct supervised heterogeneous learning. Furthermore, the proposed model also borrows the idea from the research of multiple kernel learning (24). The idea in MKL research is to use multiple kernels in formulating a learning process. In this way, the learning model itself will pick the best kernels to improve the result. In this paper, the proposed model attacks a more general but more challenging problem. It aims at choosing the optimal data sources, among which some of them have totally different statistics and data structures.

TABLE I. Symbol definition for GBC

Symbol	Definition
$\mathbf{x}_j^{(i)} \in \mathbb{R}^{d_i}$	The j-th data (column vector) in the i-th source (the i-th feature space).
G_g	The g-th relational graph.
$\mathcal{U}_i$	The set of unlabeled data in the i-th data source.
$f_i(\mathbf{x})$	The prediction model built from the i-th data source.
$\mathcal{C}$	Consensus constraint.
$\mathcal{G}$	Graph connectivity constraint.
$\mathcal{T}$	Set of labeled data.

3.3 Problem Formulation

In this section, we formally define the problem of heterogeneous learning, and then introduce a general learning objective. In heterogeneous learning, data can be described in heterogeneous feature spaces from multiple sources. Traditional vector-based features are denoted with the column vectors $\mathbf{x}_i^{(j)} \in \mathbb{R}^{d_j}$ corresponding to the i-th data in the j-th source (or the j-th feature space) whose dimension is d_j. In matrix form, $\mathbf{X}^{(j)} = [\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, \cdots, \mathbf{x}_m^{(j)}] \in \mathbb{R}^{d_j \times m}$ is the dataset in the j-th feature space where m is the sample size. Different from vector-based features, graph relational features describe the relationships between instances. In other words, they are graphs representing connectivity/similarity of the data. Specifically, we denote $G_g =< V_g, E_g >$ as the g-th graph where V_g is the set of nodes and $E_g \subseteq V_g \times V_g$ is the set of edges. We assume that the features from the same data source are from the same feature space, and hence each data source has a corresponding feature space. Furthermore, different data sources may provide different sets of instances. In other words, some instances exist in some data sources, but are missing in the others. Thus, heterogeneous learning is a machine learning scenario where we consider data from different sources, but they may (1) have different sets of instances,

(2) have different feature spaces, and (3) have multiple network based (graph) datasets. Hence, we have p data sources providing vector-based features $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(p)}$ and q data sources providing relational networks $G_1, \cdots, G_q$. The aim is to derive learning models (classification, regression or clustering) by collectively and judiciously using the $p + q$ data sources. A set of important symbols in the remaining of the paper are summarized in Table V.

3.4 Gradient Boosting Consensus

In this section, we describe the general framework of the proposed GBC model and its theoretical foundations.

3.4.1 The GBC framework

In order to use multiple data sources, the objective function aims at minimizing the overall empirical loss in all data sources, with two more constraints. First, the same instance should have similar predictions from the models trained on different data sources, and we call this the *principle of consensus*. Second, when graph relational data is provided, the connected data should have similar predictions, and we call this the *principle of connectivity similarity*. In summary, the objective function can be written as follows:

$$\begin{aligned} \min \mathcal{L} &= \sum_i w_i \sum_{\mathbf{x} \in \mathcal{T}} \mathbf{L}(f_i(\mathbf{x}), y) \\ \text{s.t. } \mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) &= 0 \\ \mathcal{G}\big(\mathbf{f}, \mathbf{w}\big) &= 0 \end{aligned} \tag{3.1}$$

where $\mathbf{L}(f_i(\mathbf{x}), y)$ is the empirical loss on the set of training data $\mathcal{T}$, w_i is the weight of importance of the i-th data source, which is discussed in Section 3.4.3. Furthermore, the two constraints $\mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) = 0$

and $\mathcal{G}\big(\mathbf{f}, \mathbf{w}\big) = 0$ are the two assumptions discussed above, which are the principle of consensus and principle of connectivity similarity, respectively. More specifically, the consensus constraint $\mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) = 0$ is defined as follows:

$$\begin{aligned} \mathcal{C}\big(\mathbf{f}, \mathbf{w}\big) &= \sum_i w_i \sum_{\mathbf{x} \in \mathcal{U}_i} \mathbf{L}\Big(f_i(\mathbf{x}), \mathbb{E}\big(f(\mathbf{x})\big)\Big) \\ \mathbb{E}\big(f(\mathbf{x})\big) &= \sum_{\{i|\mathbf{x} \in \mathcal{U}_i\}} w_i f_i(\mathbf{x}) \\ \text{s.t. } & \sum_i w_i = 1 \end{aligned} \tag{3.2}$$

It first calculates the expected prediction $\mathbb{E}\big(f(\mathbf{x})\big)$ of a given unlabeled instance $\mathbf{x}$, by summarizing the current predictions from multiple data sources $\sum_{\{i|\mathbf{x} \in \mathcal{U}_i\}} w_i f_i(\mathbf{x})$. This expectation is computed only from the data sources that contain $\mathbf{x}$; in other words, it is from the data sources whose indices are in the set $\{i|\mathbf{x} \in \mathcal{U}_i\}$ where $\mathcal{U}_i$ is the set of unlabeled instances in the i-th data source. Hence, if the j-th data source does not have record of $\mathbf{x}$, it will not be used to calculate the expected prediction. This strategy enables GBC to handle incomplete instances in multiple data sources, and uses complete instances (i.e., those with records in all data sources) to improve the consensus. Eq. Equation 3.2 forces the predictions of $\mathbf{x}$ (e.g., $f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots$) to be close to $\mathbb{E}\big(f(\mathbf{x})\big)$.

Furthermore, according to the principle of connectivity similarity, we introduce another constraint $\mathcal{G}(\mathbf{f}, \mathbf{w})$ as follows:

$$\begin{aligned} \mathcal{G}(\mathbf{f}, \mathbf{w}) &= \sum_i w_i \sum_{\mathbf{x} \in \mathcal{U}_i} \mathbf{L}\big(f_i(\mathbf{x}), \tilde{\mathbb{E}}_i\big(f(\mathbf{x})\big) \\ \tilde{\mathbb{E}}_i\big(f(\mathbf{x})\big) &= \sum_g \frac{\hat{w}_g}{|\{(z, x) \in G_g\}|} \sum_{(\mathbf{z}, \mathbf{x}) \in G_g} f_i(\mathbf{z}) \end{aligned} \tag{3.3}$$

$$\text{s.t.} \sum_g \hat{w}_g = 1$$

The above constraint encourages connected data to have similar predictions. It works by calculating the graph-based expected prediction of $\mathbf{x}$ by looking at the average prediction ($\frac{1}{|\{(z,x) \in G_g\}|} \sum_{(\mathbf{z},\mathbf{x}) \in G_g} f_i(\mathbf{z})$) of all its connected neighbors ($\mathbf{z}$'s). If there are multiple graphs, all the expected predictions are summarized by the weights $\hat{w}_g$.

We use the method of Lagrange multipliers (5) to solve the constraint optimization in Eq. Equation 3.1. The objective function becomes

$$\min \mathcal{L} = \sum_i w_i \sum_{\mathbf{x} \in \mathcal{T}} \mathbf{L}(f_i(\mathbf{x}), y) + \lambda_0 \mathcal{C}(\mathbf{f}, \mathbf{w}) + \lambda_1 \mathcal{G}(\mathbf{f}, \mathbf{w}) \tag{3.4}$$

where the two constraints $\mathcal{C}(\mathbf{f}, \mathbf{w})$ and $\mathcal{G}(\mathbf{f}, \mathbf{w})$ are regularized by Lagrange multipliers λ_0 and λ_1. These parameters are determined by cross-validation, which is detailed in Section 4.5. Note that in Eq. Equation 4.1, the weights w_i and $\hat{w}_g$ ($i, g = 1, 2, \cdots$) are essential. On one hand, the w_is are introduced to assign different weights to different vector-based data sources. Intuitively, if the t-th data source is more informative, w_t should be large. On the other hand, the $\hat{w}_g$s are the weights for the

graph relational data sources. Similarly, the aim is to give high weights to important graph data sources, while deemphasizing the noisy ones. We define different weight symbols (w_i and $\hat{w}_g$) for the data sources with vector-based features (w_i) and graph relational features ($\hat{w}_g$). The values of the weights are automatically learned and updated in the training process, as discussed in Section 3.4.3.

3.4.2 Model training of GBC

We use stochastic gradient descent (17) to solve the optimization problem in Eq. Equation 4.1. In general, it is an iterated algorithm that updates the prediction functions $\mathbf{f}(\mathbf{x})$ in the following way:

$$\mathbf{f}(\mathbf{x}) \leftarrow \mathbf{f}(\mathbf{x}) - \rho \frac{\partial \mathcal{L}}{\partial \mathbf{f}(\mathbf{x})}$$

It is updated iteratively until a convergence condition is satisfied. Specifically, inspired by gradient boosting decision trees (or GBDT (17)), a regression tree is built to fit the gradient $\frac{\partial \mathcal{L}}{\partial \mathbf{f}(\mathbf{x})}$, and the best parameter ρ is explored via line search (17). Note that the calculation of $\frac{\partial \mathcal{L}}{\partial \mathbf{f}(\mathbf{x})}$ depends on the loss function $\mathbf{L}(f, y)$ as reflected in Eq. Equation 3.1. In the following, we use the L-2 loss (for regression problems) and the binary logistic loss (for binary classification problem) as examples:

GBC with L-2 Loss: In order to update the prediction function of the i-th data source, we follow the gradient descent formula as follows.

$$f_i(\mathbf{x}) \leftarrow f_i(\mathbf{x}) - \rho \frac{\partial \mathcal{L}}{\partial f_i(\mathbf{x})} \tag{3.5}$$

If the L-2 loss is used in $\mathcal{L}$, we have

$$\frac{\partial \mathcal{L}}{\partial f_i(\mathbf{x})} = 2w_i\Big(\sum_{\mathbf{x}\in\mathcal{T}} (f_i(\mathbf{x}) - y) + \lambda_0 \sum_{\mathbf{x}\in\mathcal{U}} (f_i(\mathbf{x}) - \mathbb{E}) + \lambda_1 \sum_{\mathbf{x}\in\mathcal{U}} (f_i(\mathbf{x}) - \tilde{\mathbb{E}}_i)\Big)$$

The L-2 loss is a straightforward loss function for the GBC model, and it is used to perform regression tasks in Section 4.5.

GBC with Logistic Loss: With logistic loss, the partial derivative in Eq. Equation 3.5 becomes:

$$\frac{\partial \mathcal{L}}{\partial f_i(\mathbf{x})} = w_i\Big(\sum_{\mathbf{x}\in\mathcal{T}} \frac{-ye^{-yf_i(\mathbf{x})}}{1+e^{-yf_i(\mathbf{x})}} + \lambda_0 \sum_{\mathbf{x}\in\mathcal{U}} \frac{-\mathbb{E}e^{-\mathbb{E}f_i(\mathbf{x})}}{1+e^{-\mathbb{E}f_i(\mathbf{x})}} + \lambda_1 \sum_{\mathbf{x}\in\mathcal{U}} \frac{-\tilde{\mathbb{E}}e^{-\tilde{\mathbb{E}}f_i(\mathbf{x})}}{1+e^{-\tilde{\mathbb{E}}f_i(\mathbf{x})}}\Big)$$

Note that the above formula uses the binary logistic loss where $y = -1$ or $y = 1$, but one can easily extend this model to tackle multi-class problems by using the one-against-others strategy. In Section 4.5, we adopt this strategy to handle multi-class problems.

With the updating rule, we can build the GBC model as described in Algorithm 2. It first finds the initial prediction models for all data sources in Step 1. Then, it goes into the iteration (Step 3 to Step 11) that generates a series of decision trees. The basic idea is to follow the updating rule in Eq. Equation 3.5, and build a decision tree $g_i(x^i)$ to fit the partial derivative of the loss (Step 5). Furthermore, we follow the idea of (17), and let the number of iterations N be set by users. In the experiment, it is determined by cross-validation.

Then given a new data $\mathbf{x}$, the predicted output is

$$\hat{f}(\mathbf{x}) = \mathbf{P}(\sum \omega_i \hat{f}_i(\mathbf{x}^i)) \tag{3.6}$$

where $\mathbf{P}(y)$ is a prediction generation function, where $\mathbf{P}(y) = y$ in regression problems, and $\mathbf{P}(y) = 1$ iff $y > 0$ ($\mathbf{P}(y) = -1$ otherwise) in binary classification problems.

3.4.3 Weight Learning

In the objective function described in Eq. Equation 4.1, one important element is the set of weights (w_i and $\hat{w}_g$) for the data sources. Ideally, informative data sources will have high weights, and noisy data sources will have low weights. As such, the proposed GBC model can judiciously filter out the data sources that are noisy. To this aim, we design the weights by looking at the empirical loss of the model trained from the data source. Specifically, if a data source induces large loss, its weight should be low. Following this intuition, we design the weight as follows:

$$w_i = \exp\Big(-\sum_{\mathbf{x}\in\mathcal{L}} \mathbf{L}\big(f_i(\mathbf{x}), y\big)/z \Big) \tag{3.10}$$

where $\mathbf{L}\big(f_i(\mathbf{x}), y\big)$ is the empirical loss of the model trained from the i-th data source, and z is a normalization constant to ensure the summation of w_is equals to one. Note that the definition of the weight w_i is derived from the weighting matrix in normalized cut (52). The exponential part can effectively give penalty to large loss. Hence, w_i will be large if the empirical loss of the i-th data source is small; it becomes small if the loss is large. It is proven in Theorem 1 that the updating rule of the weights in

Algorithm 1 Gradient Boosting Consensus

Input: Data from different sources (including vector-based and graph data): $\mathcal{X}_1, \mathcal{X}_2, \cdots, \mathcal{X}_p$, Expected outputs (labels or regression values) of a subset of data $\mathcal{Y}$. Number of iterations N.

Output: The prediction model GBC $\hat{f}(\mathbf{x})$.

Initialize $\hat{f}_i(\mathbf{x})$ to be a constant such that $\hat{f}_i(\mathbf{x}) = \arg\min_{\rho_i} \sum_{\mathbf{x}\in\mathcal{L}} \mathbf{L}(\rho_i, y)$ for $i = 1, 2, \cdots, p$.

Initialize $w_i = \frac{1}{p}$.

for $t = 1, 2, \cdots, N$ **do**

  for $i = 1, 2, \cdots, p$ **do**

    For all $\mathbf{x}^{(i)}$, compute the negative gradient with respect to $f(\mathbf{x}^{(i)})$:

$$z_i = -\frac{\partial}{\partial f_i(\mathbf{x}^{(i)})} \mathcal{L}\big(\mathbf{f}(\mathbf{x}), \mathbf{w}\big) \tag{3.7}$$

    where $\mathcal{L}$ is defined in Eq. Equation 4.1 with both vector-based and graph data.

    Fit a regression model $g_i(\mathbf{x}^{(i)})$ that predicts z_i's from $\mathbf{x}^{(i)}$'s.

    Line search to find the optimal gradient descent step size as

$$\rho_i = \arg\min_{\rho_i} \mathcal{L}\big(\hat{f}_i(\mathbf{x}) + \rho_i g_i(\mathbf{x}^{(i)}), \mathbf{w}\big) \tag{3.8}$$

    Update the estimate of $\hat{f}_i(\mathbf{x}^{(i)})$ as

$$\hat{f}_i(\mathbf{x}^{(i)}) \leftarrow \hat{f}_i(\mathbf{x}^{(i)}) + \rho_i g_i(\mathbf{x}^{(i)}) \tag{3.9}$$

  end

  Update $\mathbf{w}$ as Eq. Equation 3.10 and Eq. Equation 3.11.

end

$\hat{f}(\mathbf{x}) = \mathbf{P}(\sum \omega_i \hat{f}_i(\mathbf{x}^{(i)}))$

Eq. Equation 3.10 can result in a smaller error bound. Similarly, we define the weights for graph data sources as follows:

$$w_g = \exp\Big(-\frac{1}{c} \sum_{\mathbf{x}_a^g \sim \mathbf{x}_b^g} \sum_i w_i \mathbf{L}\big(f_i(\mathbf{x}_a), f_i(\mathbf{x}_b)\big)/z\Big) \tag{3.11}$$

where $\mathbf{L}\big(f_i(\mathbf{x}_a), f_i(\mathbf{x}_b)\big)$ is the pairwise loss that evaluates the difference between the two predictions $f_i(\mathbf{x}_a)$ and $f_i(\mathbf{x}_b)$. The idea behind Eq. Equation 3.11 is to evaluate whether a graph can link similar

instances together. If most of the connected instances have similar predictions, the graph is considered to be informative. Note that both the weights in Eq. Equation 3.10 and the weights in Eq. Equation 3.11 are updated at each iteration. By replacing them into Eq. Equation 4.1, one can observe that the objective function of the GBC model is adaptively updated at each iteration. In other words, at the initial step, each data source will be given equal weights; but after several iterations, informative data sources will have higher learning weights, and the objective function will "trust" more the informative data sources. Note that this is a very important setting in dealing with the case that the consensus assumption does not hold. In this situation, the weight learning process will drop the weights of other data sources close to zero. The whole model will be degraded to a normal GBDT.

3.4.4 Generalization bounds

In this section, we consider the incompatibility framework in (3) and (63) to explain the proposed GBC model. Specifically, we show that the weight learning process described in Section 3.4.3 can help reduce an error bound. For the sake of simplicity, we consider the case where we have two data sources $\mathcal{X}_1$ and $\mathcal{X}_2$, and the case with more data sources can be analyzed with similar logic. Note that the goal is to learn a pair of predictors $(f_1; f_2)$, where $f_1 : \mathcal{X}_1 \to \hat{\mathcal{Y}}$ and $f_2 : \mathcal{X}_2 \to \hat{\mathcal{Y}}$, and $\hat{\mathcal{Y}}$ is the prediction space. Further denote $\mathcal{F}_1$ and $\mathcal{F}_2$ as the hypothesis classes of interest, consisting of functions from $\mathcal{X}_1$ (and, respectively, $\mathcal{X}_2$) to the prediction space $\hat{\mathcal{Y}}$. Denote by $L(f_1)$ the expected loss of f_1, and $L(f_2)$ is similarly defined. Let a Bayes optimal predictor with respect to loss L be denoted as f^*. We now apply the incompatibility framework for the multi-view setting (3) to study GBC. We first define the

incompatibility function $\chi : \mathcal{F}_1 \times \mathcal{F}_2 \to \mathbb{R}^+$, and some $t \geq 0$ as those pairs of functions which are compatible to the tune of t, which can be written as:

$$\mathcal{C}^{\chi}(t) = \{(f_1, f_2) : f_1 \in \mathcal{F}_1, f_2 \in \mathcal{F}_2 \text{ and } \mathbb{E}[\chi(f_1, f_2)] \leq t\}$$

Intuitively, the function $\mathcal{C}^{\chi}(t)$ captures the set of function pairs f_1 and f_2 that are compatible with respect to a "maximal expected difference" t. From (3), it is proven that there exists a symmetric function $d : \mathcal{F}_1 \times \mathcal{F}_2$, and a monotonically increasing non-negative function Φ on the reals such that for all f,

$$\mathbb{E}[d(f_1(x); f_2(x))] \leq \Phi(L(f_1) - L(f_2))$$

With these functions at hand, we can derive the following theorems:

Theorem 1 *Let $|L(f_1) - L(f^*)| < \epsilon_1$ and $|L(f_2) - L(f^*)| < \epsilon_2$, then for the incompatibility function $\mathcal{C}^{\chi}(t)$, if we set $\chi = d$, for $t = c_d(\Phi(\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}) + \Phi(\epsilon_{bayes}))$ where c_d is a constant depends on the function d (3), we have*

$$\inf_{(f_1,f_2)\in\mathcal{C}^{\chi}(t)} L_{GBC}(f_1, f_2) \leq L(f^*) + \epsilon_{bayes} + \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1 + \epsilon_2}} \tag{3.12}$$

Proof 1 *Note that* $|L(f_1)-L(f^*)| < \epsilon_1$ *and* $|L(f_2)-L(f^*)| < \epsilon_2$*, and the proposed model GBC adopts a weighted strategy linear to the expected loss, which is approximately* $L_{GBC}(f_1, f_2) = \frac{\epsilon_2}{\epsilon_1+\epsilon_2}L(f_1) + \frac{\epsilon_1}{\epsilon_1+\epsilon_2}L(f_2)$*. According to Lemma 8 in (63), we have* $E[\chi(f_1, f_2)] \leq c_d^2(\Phi(\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}) + \Phi(\epsilon_{bayes}))$*, and*

$$\min_{(f_1,f_2)\in\mathcal{C}^\chi(t)} L_{GBC}(f_1, f_2) \leq L_{GBC}(f*_1, f*_2) + \epsilon_{bayes} \tag{3.13}$$

With Lemma 7 in (63), we can get

$$\min_{(f_1,f_2)\in\mathcal{C}^\chi(t)} L_{GBC}(f_1, f_2) \leq L(f^*) + \epsilon_{bayes} + \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \tag{3.14}$$

□

Similarily, we can derive the error bound of GBC in a transductive setting.

Theorem 2 *Consider the transductive formula Eq. 4 in (63). Given the regularized parameter* $\lambda > 0$*, we denote* $L^\lambda(f)$ *as the expected loss with the regularized parameter* λ*. If we set* $\lambda_c = \frac{\lambda}{4(K+\lambda)^2\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}}$ *then for the pair of functions* $(f_1, f_2) \in \mathcal{F}_1 \times \mathcal{F}_2$ *returned by the transductive learning algorithm, with probability at least* $1-\delta$ *over labeled samples,*

$$\begin{aligned} L^\lambda_{GBC}(f_1, f_2) &\leq L^\lambda(f^*) + \frac{1}{\sqrt{n}}\Big(2 + 3\sqrt{\frac{ln(\frac{2}{\delta})}{2}}\Big) \\ &+2C_{Lip}\hat{R}(\hat{\mathcal{C}}^\chi(\frac{1}{\lambda_c})) + \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \end{aligned} \tag{3.15}$$

where n *is the number of labeled examples, and* C_{Lip} *is the Lipschitz constant for the loss, and* $\hat{R}(\hat{\mathcal{C}}^\chi(\frac{1}{\lambda_c}))$ *is a term bounded by the number of unlabeled examples and the bound of the losses.*

Proof 2 *We first note that* $|L^{\lambda}(f_1) - L^{\lambda}(f^*)| < \epsilon_1$ *and* $|L(f_2)^{\lambda} - L^{\lambda}(f^*)| < \epsilon_2$*. Similar to the logic in Theorem 1, GBC employs a weighting strategy which is linear to the expected loss:* $L^{\lambda}_{GBC}(f_1, f_2) = \frac{\epsilon_2}{\epsilon_1+\epsilon_2}L^{\lambda}(f_1) + \frac{\epsilon_1}{\epsilon_1+\epsilon_2}L^{\lambda}(f_2)$*. With Lemma 7 in (63), we then have*

$$\begin{aligned} &|L^{\lambda}(f_1) - L^{\lambda}_{GBC}| + |L^{\lambda}(f_1) - L^{\lambda}_{GBC}| \\ &\leq \sqrt{\frac{\epsilon_2}{\epsilon_1+\epsilon_2}}\sqrt{\epsilon_1} + \sqrt{\frac{\epsilon_1}{\epsilon_1+\epsilon_2}}\sqrt{\epsilon_2} \\ &= 2\sqrt{\frac{\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \end{aligned} \tag{3.16}$$

Hence, we can further get the following relationship:

$$\begin{aligned} E[\chi(f_1, f_2)] &\leq c_d^2(E[\chi(f_1, y_1)] + E[\chi(y_1, y_2)] + E[\chi(f_2, y_2)]) \\ &\leq c_d^2(L^{\lambda}(f_1) - L^{\lambda}(f^*) + L^{\lambda}(f_2) - L^{\lambda}(f^*) + 2\Phi(\epsilon_{bayes})) \\ &\leq c_d^2(\Phi(\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}) + \Phi(\epsilon_{bayes})) \end{aligned} \tag{3.17}$$

and

$$\min_{(f_1,f_2)\in\mathcal{C}^{\chi}(t)} L^{\lambda}_{GBC}(f_1, f_2) \leq L^{\lambda}_{GBC}(f*_1, f*_2) + \epsilon_{bayes} \tag{3.18}$$

We then have

$$\begin{aligned} L^{\lambda}_{GBC}(f_1, f_2) &\leq L^{\lambda}(f^*) + \frac{1}{\sqrt{n}}\Big(2 + 3\sqrt{\frac{ln(\frac{2}{\delta})}{2}}\Big) \\ +2C_{Lip}\hat{R}(\hat{\mathcal{C}}^{\chi}(\frac{1}{\lambda_c})) &+ \sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \end{aligned} \tag{3.19}$$

□

Note that Theorem 1 and Theorem 2 derive the error bounds of GBC in inductive and transductive setting respectively. In effect, the weighting strategy reduces the last term of the error bound to $\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}}$, as compared to the equal-weighting strategy whose last term is $\sqrt{\frac{\epsilon_1+\epsilon_2}{2}}$ (3). Hence, the weighting strategy induces a tighter bound since $\sqrt{\frac{2\epsilon_1\epsilon_2}{\epsilon_1+\epsilon_2}} \leq \sqrt{\frac{\epsilon_1+\epsilon_2}{2}}$. It is important to note that if the the predictions of different data sources vary significantly ($|\epsilon_1 - \epsilon_2|$ is large), the proposed weighting strategy has a much tighter bound than the equal-weighting strategy. In other words, if there are some noisy data sources that potentially lead to large error rate, GBC can effectively reduce their effect. This is an important property of GBC to handle noisy data sources. This strategy is evaluated empirically in the next section.

3.5 Experiments

In this section, we report four sets of experiments that were conducted in order to evaluate the proposed GBC model applied to multiple data sources. We aim to answer the following questions:

- Can GBC make good use of multiple data sources? Can it beat other more straightforward strategies?
- What is the performance of GBC if there exist incomplete instances (e.g., new instances in new data source that do not have records in other data sources)?
- How does GBC perform on big dataset?

3.5.1 Datasets

The aim of the first set of experiments is to predict movie ratings from the IMDB database.[1] Note that there are 10 data sources in this task. For example, there is a data source about the plots of the

[1] http://www.imdb.com/

TABLE II. IMDB movie rating prediction

Data source	Type of features
Quote Database	Text
Plot Database	Text
Technology Database	Nominal
Sound Technology Database	Nominal
Running Time Database	Real
Genre Database	Binary
Actor Graph	Graph
Actress Graph	Graph
Director Graph	Graph
Writter Graph	Graph

movies, and a data source about the techniques used in the movies (e.g., 3D IMAX). Furthermore, there are several data sources providing different graph relational data about the movies. For example, in a director graph, two movies are connected if they have the same director. A summary of the different data sources can be found in Table II. It is important to note that each of the data sources may provide certain useful information for predicting the ratings of the movies. For instance, the Genre database may reflect that certain types of movies are likely to have high ratings (e.g., Fantasy); the Director graph database implicitly infers movie ratings from similar movies of the same director (e.g., Steven Spielberg has many high-rating movies.). Thus, it is desirable to incorporate different types of data sources to give a more accurate movie rating prediction. This is an essential task for online TV/movie recommendation, such as the famous $1,000,000 Netflix prize (33).

The second set of experiments is about handwritten number recognition. The dataset contains 2000 handwritten numerals ("0"–"9") extracted from a collection of Dutch utility maps.[1] The handwritten numbers are scanned and digitized as binary images. They are represented in terms of the following seven data sources with different vector-based feature spaces: (1) 76 Fourier coefficients of the character shapes, (2) 216 profile correlations, (3) 64 Karhunen-Love coefficients, (4) 240 pixel averages in 2 $\times$ 3 windows, (5) 47 Zernike moments, (6) a graph dataset constructed from the morphological similarity (i.e., two objects are connected if they have similar morphology appearance), and (7) a graph generated with the same method as (6), but with random Gaussian noise imposed in the morphological similarity. This dataset is included to test the performance of GBC on noisy data. The aim is to classify a given object to one of the ten classes ("0"–"9"). The statistics of the dataset are summarized in Table III.

The third set of datasets is downloaded from the UMD collective classification database [2]. The database consists of 1293 different attacks in one of the six labels indicating the type of the attack: arson, bombing, kidnapping, NBCR attack, weapon attack and other attack. Each attack is described by a binary value vector of attributes whose entries indicate the absence or presence of a feature. There are a total of 106 distinct vector-based features, along with three sets of relational features. One set connects the attacks together if they happened in the same location; the other connects the attacks if they are planned by the same organization. In order to perform robust evaluation of the proposed GBC

---

[1] http://archive.ics.uci.edu/ml/datasets/Multiple+Features

[2] http://www.cs.umd.edu/projects/linqs/projects/lbc/index.html

model, we add another data source based on the vector-based dataset, but with a random Gaussian noise $\mathcal{N}(0, 1)$. Again, this is to test the capability of the proposed model to handle noise.

The fourth set of datasets is collected from a tier-1 telegraph network provider in the U.S. We study a subset of the demographic database with over 500,000 anonymous users. The database records 196 demographic features per user, which includes education level, age group, language, hobbies, etc. As the objective in (**?**), we aim at predicting four demographic features, which include the age group, gender, credit level, and whether the user is a renter. For the tasks of predicting the age group and credit level, we only consider whether the user belongs to a specific group of interest (e.g., mid age and good credit level). As a result, all four tasks have binary classification labels. Furthermore, we construct 90 different social graphs from the phone call networks similar to the ones introduced in (**?**). These generated social graphs are considered to be close approximations to the real-world social connections, and they cover the social connections among anonymous users in different time periods. In summary, we have one vector-based demographic dataset and 90 graph datasets, and each of them involves a subset of the 500,000 anonymous users. Another very important characteristic is that the dataset contains substantial missing values, owing to the difficulty of obtaining the demographic features (e.g., love fishing? speak Japanese?, etc.). In the dataset, over 50% of samples contain over 60% of missing values. Classic feature based algorithms such as SVM cannot easily handle this case. On the contrary, GBC is designed to fit to the situation with missing values and with many heterogeneous data sources. We design four prediction tasks to predict four important demographic features of the samples, and they include the prediction of the age group, gender, credit level, and whether the user is a renter. Our objective is then to evaluate the four prediction tasks, all of which involve big and heterogeneous data.

3.5.2 Comparison Methods and Evaluations

It is important to emphasize again that there is no previous model that can handle the same problem directly; i.e., building a learning model from multiple graphs and multiple vector-based datasets with some incomplete instances. Furthermore, as far as we know, there is no state-of-the-art approaches that use the benchmark datasets described in the previous section in the same way. For instance, in the movie prediction dataset, we crawl the 10 data sources directly from IMDB and use them collectively in learning. In the case of the number recognition dataset, we have two graph data sources, which are different from previous approaches that only look at vector-based features (68), clustering (16), or feature selection problems (26). In order to evaluate the proposed GBC model, we design a straightforward comparison strategy, which is to directly join all features together. In other words, given the sources with vector-based features $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(p)}$ and the adjacency matrices of the graphs $\mathbf{M}^{(1)}, \cdots, \mathbf{M}^{(q)}$, the joined features can be represented as follows:

$$\mathbf{X} = [{\mathbf{X}^{(1)}}^T, \cdots, {\mathbf{X}^{(p)}}^T, {\mathbf{M}^{(1)}}^T, \cdots, {\mathbf{M}^{(q)}}^T]^T \tag{3.20}$$

Since there is only one set of joined features, traditional learning algorithms can be applied on it to give predictions (each row is an instance; each column is a feature from a specific source). We include support vector machines (SVM) in the experiments as it is used widely in practice. Note that in GBC, the consensus term in Eq. Equation 3.2 and the graph similarity term in Eq. Equation 3.3 can use unlabeled data to improve the learning. Hence, we also compare it with semi-supervised learning models. Specifically, semi-supervised SVM (Semi-SVM) with a self-learning technique (79) is used as

the second comparison model. Note that we have four tasks in the experiment where one of them (i.e., the movie rating prediction task) is a regression task. In this task, regression SVM (35) is used to give predictions. Additionaly, since the proposed model is derived from gradient boosting decision trees, GBDT (17) is used as the third comparison model, and its semi-supervised version (79) is included as well. It is important to note that in order to use the joined features from Eq. Equation 3.20, these comparison models require that there is no incomplete instances. In other words, all data sources should have records of all instances; otherwise, the joined features will have many missing values since some data sources may not have records of the corresponding instances. To evaluate GBC more comprehensively, we thus conducted the experiments on two settings:

- Setting with all complete instances: the first setting is to force all data sources to contain records of all instances. We only look at the instances that have records in all data sources. Table III presents the statistics of the datasets in this setting. In this case, we can easily join the features from different sources as in Eq. Equation 3.20. Note that we do not perform "the setting with complete instances" in the demographic prediction task. The reason is that there is only a couple of users that appear in all 91 different data sources. The complete-instance setting thus cannot generate statistically significant results on this big and sparse dataset.
- Setting with incomplete instances: the second setting is to allow different data sources to have some incomplete instances. Thus, an instance described in one data source may not appear in other data sources. This setting is more realistic, as the example in Fig. 16. The proposed GBC model is able to handle this case, since it allows incomplete instances. However, for the comparison method, there will be many missing values in the joined features as discussed above. In this case,

TABLE III. Data descriptions

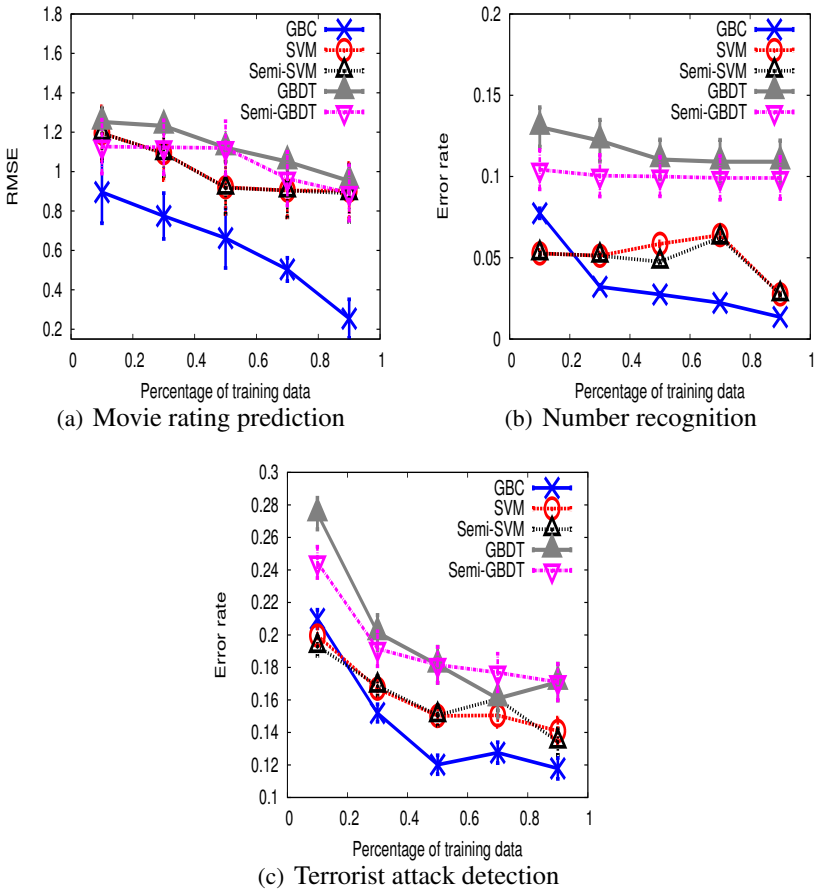
Task	# of data	# of data sources	Average dimension	Predictions
Movie Rating Prediction	3000	10 (4 graph and 6 others)	896 ($\times 10$)	Regression
Number Recognition	2000	7 (2 graph and 5 others)	378 ($\times 7$)	10 labels
Terrorist Attack Classification	1293	4 (2 graph and 2 others)	422 ($\times 4$)	6 labels
Demographic Prediction	500,000	91	499,997 ($\times 91$)	2 labels

we replaced the missing values with the average values of the corresponding features. In this setting, 30% of the instances do not have records in half of the data sources.

We conducted experiments on the above two settings. During each run, we randomly selected a certain portion of examples as training data, keeping the others as test data. For the same training set size, we randomly selected the set of training data 10 times and the rests were used as test data, and the results were averaged over the 10 runs. The experiment results are reported with different training set sizes. Note that the proposed GBC model can be used for both classification and regression. We used error rate to evaluate the results for classification tasks, and root mean square error (RMSE) for regression tasks.

3.5.3 Analysis of the Experiments

Our aim is to study the performance of the proposed GBC model in the two setting described above: the settings with complete instances and that allowing incomplete instances. The experiment results are summarized in Fig. 4 and Fig. 5, respectively. The x-axes record different percentage of training data (while the remainder of the data is used for evaluation), and the y-axes report the errors of the corresponding learning model.

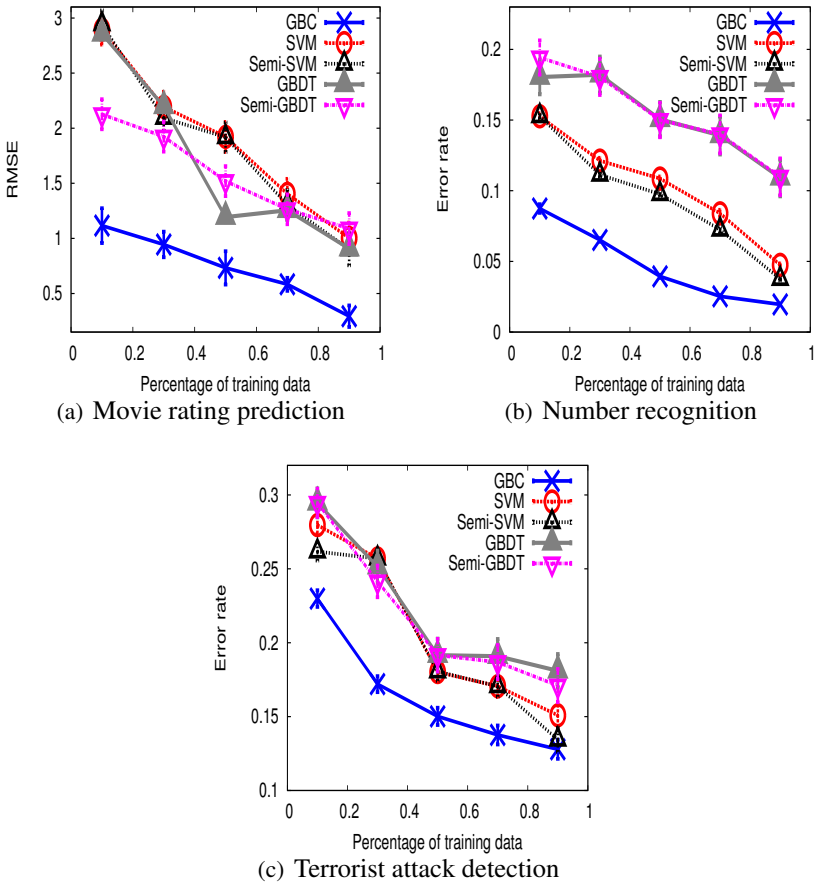


(a) Movie rating prediction (b) Number recognition

(c) Terrorist attack detection

Figure 4. All data sources record the same set of objects.

We observe two major phenomena in the experiments. Firstly, the proposed GBC model effectively reduces the error rate as compared to the other learning models in both settings. It is especially obvious in the movie rating prediction dataset where 10 data sources are used to build the model. In this dataset, GBC reduces the error rate by as much as 80% in the first setting (when there are 90% of training instances), and 60% in the second setting (when there are 10% of training instances). This shows that



(a) Movie rating prediction (b) Number recognition

(c) Terrorist attack detection

Figure 5. Each data source is independent (with 30% of incomplete instances that have no record in some data sources).

GBC is especially advantageous when a large number of data sources are available. We further analyze this phenomenon in the next section with Table IV. On the other hand, the comparison models have to deal with a longer and noisier feature vector. GBC beats the four approaches by judiciously reducing the noise (as discussed in Section 3.4.3 and 3.4.4). Secondly, we can observe that GBC outperforms the other approaches significantly and substantially in the second setting (Fig. 5) where some instances do

not have records on all data sources. As analyzed in the previous section, this is one of the advantages of GBC over the comparison models that have to deal with missing values.

It is also interesting to analyze the performance of GBC on the demographic prediction task. Recall that the learning task has at least three challenges. First, it contains a large set of samples (over 500,000) as compared to normal machine learning tasks. Second, it has 91 different data sources, among which one is a demographic dataset with around 200 features, and the other 90 data sources are social graphs. As a result, it generates over 45,000,000 joined features (as in Eq. Equation 3.20). Third, it contains substantial missing values (50% of samples miss 60% of features), which brings in more difficulty in finding the valuable information. This is also the reason we cannot conduct complete-instance setting on this big dataset, since there is only a couple of users that appear in all 91 different data sources. The complete-instance setting thus cannot generate statistically significant results. All the experiments reported in Fig. 7 were run on a distributed computing system with the Condor framework. It can be clearly observed that GBC outperforms the other comparison models in all four tasks. For instance, in the task of predicting genders, the error rate has been reduced around 20% when there is 50% of training data, and 25% when there is 90% of training examples. The good performance of GBC comes from its unique design to distill useful information from multiple heterogeneous data sources. Each of the individual data source may be very noisy owing to the missing values and unknown data quality. However, GBC is able to distinguish the useful data sources for the learning tasks, and judiciously uses them in improving learning results. The distribution of the learned weights of the data sources are plotted in Fig. 6. It can be shown that among the 91 data sources, some of them are very useful

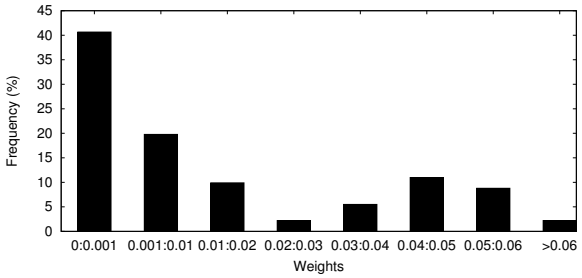


Figure 6. Frequency of the Learned Weights

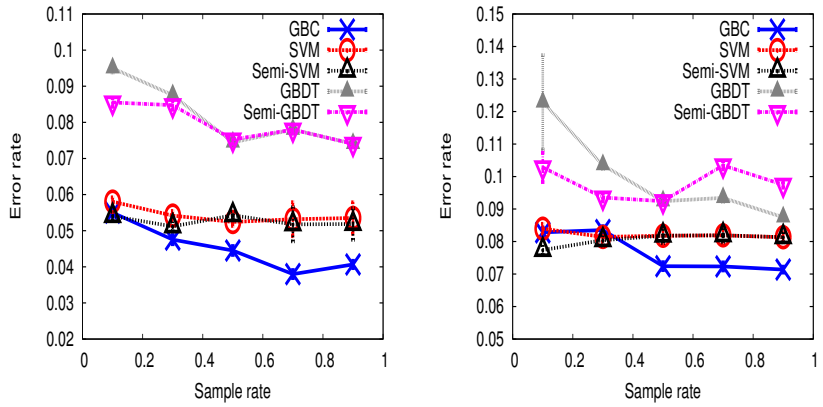
(with large weights), while many of them are quite noisy and are assigned low weights. GBC is able to judiciously treat them differently to achieve a better model.

3.6 Discussion

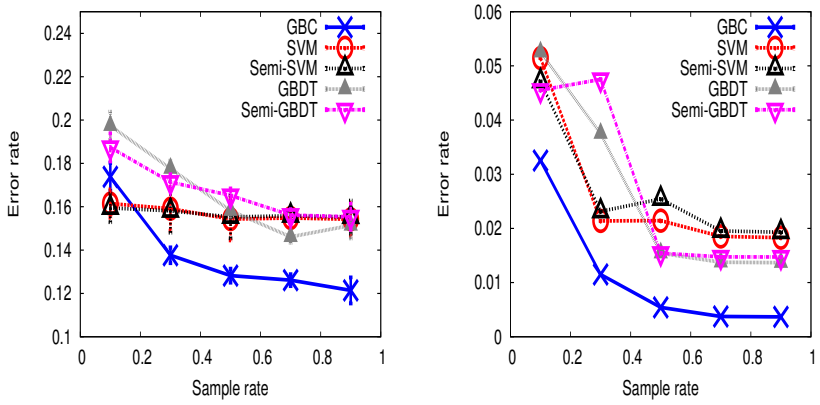
In this section, we would like to answer the following questions:

- To what extent GBC helps to integrate the knowledge from multiple sources, compared to learn from each source independently? Specifically, how do the principles of consensus and connectivity similarity help GBC?
- Is the weight learning algorithm necessary?
- Do we need multiple data sources? Does the number of data sources affects the performance?

In GBC, both λ_0 and λ_1 (from Eq. Equation 4.1) are determined by cross-validation. In the following set of experiments, we tune the values of λ_0 and λ_1 in order to study the specific effects of the consensus and connectivity terms. We compare the proposed GBC model with three algorithms on the number recognition dataset in the complete-instance setting (i.e., all data sources contain all instances):



(a) Credit Level (b) Age Group



(c) Gender (d) Renter

Figure 7. Demographic Prediction

- The first comparison model is to set λ_0 to zero, and determine λ_1 by cross-validation. In this case, the consensus term is removed. In other words, the algorithm can only apply the connectivity similarity principle. We denote this model as "GBC without consensus".

- The second comparison model is to set λ_1 to zero, and let λ_0 be determined by cross-validation. In other words, the connectivity similarity term is removed, and the algorithm only depends on the empirical loss and the consensus term. We denote this model as "GBC without graph".
- The third comparison model is to set both λ_0 and λ_1 to 0. Hence, the remaining term is the empirical loss, and it is identical to the traditional GBDT model.

The empirical results on the number recognition task are presented in Fig. 8. It can be observed that all three models outperforms traditional GBDT. We draw two conclusions from this experiment. First, as was already observed in previous experiments, learning from multiple sources is advantageous. Specifically, the GBDT model builds classifiers for each of the data source independently, and average the predictions at the last step. However, it does not "communicate" the prediction models during training. As a result, it has the worst performance in Fig. 8. Second, both the principle of consensus and the principle of connectivity similarity improve the performance. Furthermore, it shows that the connectivity similarity term helps improve the performance more when the number of training data is limited. For example, when there are only 10% of training instances, the error rate of GBC with only the connectivity term (i.e., GBC without consensus) is less than 9%, while that of GBC with only the consensus term (i.e., GBC without Graph) is around 13%. This is because when the number of labeled training data is limited, the graph connectivity serves as a more important source of information when connecting unlabeled data with the limited labeled data.

Again, it is important to note that in GBC, the weights of different data sources are adjusted at each iteration. The aim is to assign higher weights to the data sources that contain useful information, and filter out the noisy sources. This step is analyzed as an important one in Theorem 1 since it can help

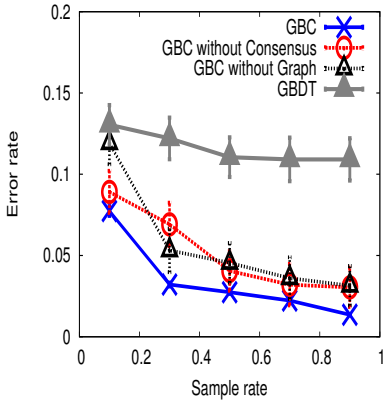


Figure 8. How do consensus principle and connectivity similarity principle help?

reduce the upper bound of the error rate. We specifically evaluate this strategy on the terrorist detection task. Note that in order to perform a robust test, one of the vector-based sources contains Gaussian noise, as described in Section 3.5.1. The empirical results on the terrorist detection task are presented in Fig. 9. It can be clearly observed that the weighting strategy is reducing the error rate by as much as 70%. Hence, an appropriate weighting strategy is an important step when dealing with multiple data sources with unknown noise.

It is also interesting to evaluate to what extent GBC performance is improved as the number of data sources increases. For this purpose, the movie rating prediction dataset is used as an example. We first study the case when there is only one data source. In order to do so, we run GBC on each of the data source independently, then on 2 and 4 data sources. In the experiments with 2 data sources, we randomly selected 2 sources from the pool (Table II) as inputs to GBC. These random selections of data sources were performed 10 times, and the average error is reported in Table IV. A similar strategy was

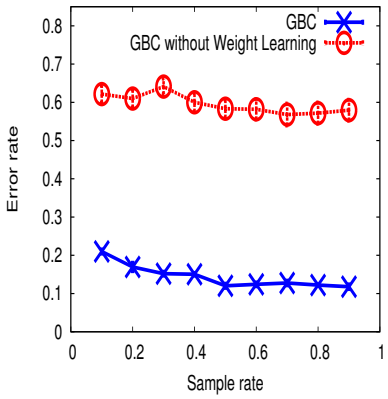


Figure 9. Why is the weighting strategy necessary?

TABLE IV. Effect of different number of sources. Reported results are RMSE with variance in parenthesis.

# of sources	10%	30%	50%	70%	90%
1	1.356 (0.358)	1.223 (0.290)	1.192 (0.257)	1.189 (0.258)	1.077 (0.223)
2	1.255 (0.237)	1.247 (0.293)	1.193 (0.229)	1.193 (0.150)	0.973 (0.216)
4	1.135 (0.138)	**0.914 (0.114)**	**0.732 (0.153)**	0.593 (0.105)	0.314 (0.092)
all	**1.115 (0.158)**	0.945 (0.116)	**0.732 (0.152)**	**0.583 (0.059)**	**0.294 (0. 097)**

implemented to conduct the experiment with 4 data sources. In Table IV, the results are reported with different percentages of training data, and the best performances are highlighted with bold letters. It can be observed that the performance with only one data source is the worst, and has high root mean square error and high variance. With more data sources available, the performance of GBC tends to be better. This is because each data source provides complementary information useful to build a comprehensive model of the whole dataset.

3.7 Conclusion

This paper studies the problem of building a learning model from heterogeneous data sources. Each source can contain traditional vector-based features or graph relational features, with potentially incomplete sets of instances. As far as we know, there is no previous model that can be directly applied to solve this problem. We propose a general framework derived from gradient boosting, called gradient boosting consensus (GBC). The basic idea is to solve an optimization problem that (1) minimizes the empirical loss, (2) encourages the predictions from different data sources to be similar, and (3) encourages the predictions of connected data to be similar. The objective function is solved by stochastic gradient boosting, with an incorporated weighting strategy to adjust the importance of different data sources according to their usefulness. Four sets of experiments were conducted, including movie rating prediction, number recognition, terrorist detection, and demographic prediction tasks with over 500,000 samples and 91 data sources. We show that the proposed GBC model substantially reduce prediction error rate by as much as 80%. Finally, several extended experiments are conducted to study specific properties of the proposed algorithm and its robustness.

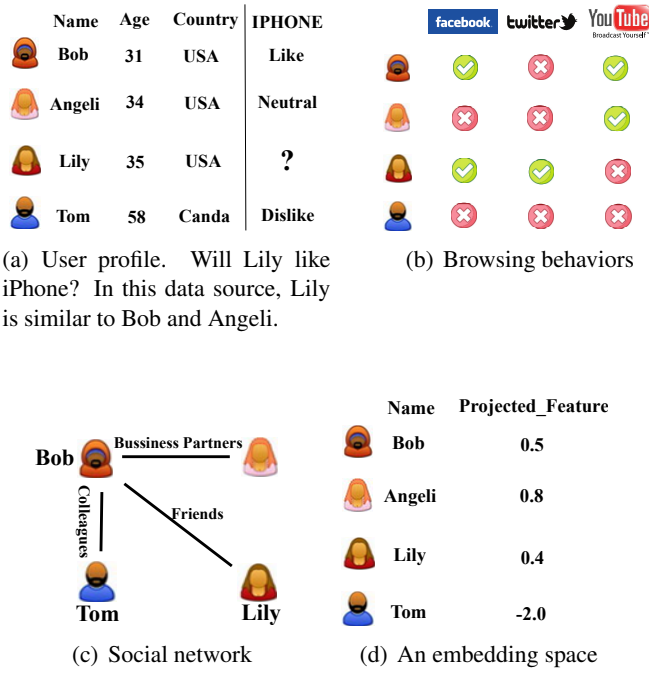
CHAPTER 4

UNSUPERVISED HETEROGENEOUS LEARNING VIA COLLECTIVE COMPONENT ANALYSIS

4.1 Introduction

Multiple related data sources may be available for a given task. An example is user-oriented recommendation system. For this task, related data sources can be (1) user profile database (as shown in Fig. 10(a)), (2) users' log data (as shown in Fig. 10(b)), and (3) users' social connections with other users (as shown in Fig. 10(c)). Each single data source may not be informative enough to build an accurate model, but the combination of them may provide important knowledge. For instance, in Fig. 10(a), given that we observe several users' opinions on the new iPhone, what is Lily's opinion about iPhone? Note that it is difficult to infer the answer just from any one of the databases (Fig. 10(a), Fig. 10(b) or Fig. 10(c)). However, if we combine the knowledge from all three data sources, it can be inferred that Lily has a similar background/behavior to Bob; thus, she may like iPhone as Bob likes. These mixture types of data are rich in information, and it is desirable to build learning models by using the heterogeneous features collectively. We call this framework *heterogeneous learning*. More specifically, the task can have (1) multiple vector-based data sources, and (2) multiple graph relational data sources. The aim is to devise learning algorithms to make good use of all types of features collectively.

Note that one essential challenge of heterogeneous learning is to conduct dimensionality reduction, in order to find a unified embedding feature space that captures the information from all sources. How-



(a) User profile. Will Lily like iPhone? In this data source, Lily is similar to Bob and Angeli.

(b) Browsing behaviors

(c) Social network

(d) An embedding space

Figure 10. Heterogeneous feature spaces in computational advertising. It is difficult to analyze Lily's opinion on iPhone if we just consider only one of the data sources (Fig. 10(a), Fig. 10(b), or Fig. 10(c)). However, if we combine the information from all three sources, we can infer Lily's opinion based on her similarity (background, behavior, social connection) to Bob. A unified embedding space generated from this example is given in Fig. 10(d).

ever, this is a challenging task. First, it is not clear how to find only one (unified) feature subspace, given the heterogeneous features from multiple sources. Second, it is not obvious how to find such embedding with multiple graph relational features (e.g., Fig. 10(c)), and for an even more daunting task with a mixture of vector-based features and relational features. To solve the problem, we propose a principle **Co**llective **C**omponent **A**nalysis (CoCA). The CoCA principle is to find a feature subspace to maximize the variance of the projected data with the following two constraints.

1. First, all vector-based feature spaces have to be projected onto the same reduced feature space consensually. For example, Fig. 10(a) and Fig. 10(b) provide two sets of heterogeneous vector-based features. They both have to agree on and to be projected to a common feature subspace as Fig. 10(d), such that their original data structures (e.g., data similarity) are preserved.
2. Second, if there are relational features (as in Fig. 10(c)), the connected data are preferred to be similar in the reduced feature space. For example, Lily and Tom do not have connections in the graph Fig. 10(c). Hence, as in the unified feature subspace Fig. 10(d), Lily and Bob are more similar than Lily and Tom.

We first analyze the CoCA principle under two special cases. They are cases where (1) there are multiple vector-based data sources (as shown in Fig. 10(a) and Fig. 10(b)), and (2) there are multiple graph relational data sources (as shown in Fig. 10(c)). We then combine the two cases, and discuss the situation where we have a mixture of vector-based features and relational features. A corresponding optimization problem is devised, and the final solution is obtained via the eigenvectors of certain matrix. Furthermore, given multiple sources, we design a quadratic programming problem to identify sources that are more informative, and optimally weight them to learn a better projection. This is an important step since some of the data sources may contain substantial noise, and it is desirable to reduce their effect. Empirical studies include three sets of real-world datasets ranging from handwritten numbers recognition, document classification and terrorist attack detection. Since there was no previous method that can solve the same problem, we devised a straightforward comparison embedding model that performed dimension reduction on the concatenation of the data sources. In all experiments, CoCA substantially outperforms

the comparison method by as much as 50%. In summary, the proposed CoCA model has the following properties.

- Firstly, it finds a unified feature subspace from *multiple* vector-based data sources and *multiple* graph relational datasets.
- Secondly, it is a robust model since a quadratic programming framework is incorporated to reduce the effect of noise.
- Thirdly, it is an efficient model. It has the same algorithm complexity as traditional dimension reduction algorithms such as PCA, even though (1) heterogeneous data sources are considered, and (2) a quadratic programming framework is used. More details can be found at Section 4.4.5.

4.2 Related Work for CoCA

There are several areas of related works that we built upon. First, multi-view learning (e.g., (7; 45; 38; 25)) is proposed to learn from instances which have multiple views in different feature spaces. For example, in (38), a framework is proposed to reconcile the clustering results from different views. In (20), a term called consensus learning is proposed. The general idea is to perform learning on each heterogeneous feature space independently and then ensemble the results. Furthermore, (1) proposes an approach to combine different data sources for recommendation systems. There are mainly two differences between our work with these previous approaches. First, so far as we know, most of the previous works do not consider vector-based features and relational features simultaneously. Second, most of these models aim at tackling classification or clustering problem. In this paper, we instead aim

at solving low-dimensional embedding especially for heterogeneous learning, to compress the features from multiple sources.

Another area of related work is collective classification (e.g., (51)) that aims at predicting the class label from a network. Its key idea is to combine the supervision knowledge from traditional vector-based feature vectors, as well as the linkage information from the network. It has been applied to various applications such as part-of-speech tagging (34), classification of hypertext documents using hyperlinks (65), etc. Most of these works study the case when there is only one vector-based feature space and only one relational feature space, and the focus is how to combine the two. Different from traditional collective classification framework, we consider multiple vector-based features and multiple relational features simultaneously. Furthermore, we solve embedding problem instead of classification problem since "the curse of dimension" is more likely to happen in heterogeneous learning (increasing number of features from multiple sources).

The proposed model is also related to dimensionality reduction and kernel learning (e.g., (74)). One of the surveys can be found at (11). There are two differences of the current work with previous works. First, most of the previous works do not consider vector-based features and graph relational features simultaneously. Second, most of approaches treat all the data sources the same, while the proposed CoCA has a schema to identify informative data source from the noisy ones. Robustness is a very important property of CoCA since the multiple data sources may contain substantial unobserved noise.

Specifically, the work in (30) also attacks dimensionality reduction on "heterogeneous" data. However, it is totally different from the present work. The objective of (30) is to design an implementation for PCA in the distributed computing environment and the term "heterogeneous data" refers to the data

TABLE V. Symbol definition for CoCA

Symbol	Definition
$\mathbf{X}^{(j)} = [\mathbf{x}_1^{(j)}, \mathbf{x}_2^{(j)}, \cdots, \mathbf{x}_m^{(j)}] \in \mathbb{R}^{d_j \times m}$	The dataset in the j-th source where $\mathbf{x}_i^{(j)} \in \mathbb{R}^{d_j}$ is the i-th data.
$\mathbf{M}^{(k)} \in \mathbb{R}^{m \times m}$	The adjacency matrix of the k-th relational graph.
$\mathbf{u}^{(j)}$	Projection basis for the j-th source (the j-th feature space).
$\mathbf{\Phi}$	The embedding space.

stored in different computers. The present work instead aims at developing an algorithm on a new learning schema called heterogeneous learning where "heterogeneous" refers to the different feature spaces of the objects, and they can have different physical meanings.

4.3 Problem Formulation

In this section, we first formally define heterogeneous learning, and then introduce the CoCA principle for heterogeneous embedding. In heterogeneous learning, the data can be described in heterogeneous feature spaces from multiple sources. Note that in this paper, we mainly study two categories of heterogeneous features: vector-based features and graph relational features. For traditional vector-based features, we use column vector $\mathbf{x}_i^{(j)} \in \mathbb{R}^{d_j}$ to denote the i-th data in the j-th source (or the j-th feature space) whose dimension is d_j. In a matrix form, we denote $\mathbf{X}^{(j)} \in \mathbb{R}^{d_j \times m}$ as the set of data in the j-th feature space where m is the sample size. Different from vector-based features, graph relational features describe the relationships among the data. In other words, they are graphs representing similarity of the data, and the adjacency matrices of the graphs are considered as relational features. Specifically, we denote $\mathbf{M}^{(k)} \in \mathbb{R}^{m \times m}$ as the adjacency matrix of the k-th graph where m is the sample size. Note that the matrix $\mathbf{M}$ can be a "soft" adjacency matrix such that $\mathbf{M}^{(k)}(i, j)$ is the degree of similarity between

the i-th data and the j-th data in the k-th graph. It is large (closed to 1) if the two data are strongly correlated; otherwise it is 0. We assume that the features from the same data source are from the same feature space, and hence each data source has a corresponding feature space. Thus, heterogeneous learning is a machine learning scenario where the data have heterogeneous feature spaces. For example, we have p data sources providing vector-based features $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(p)}$ and q data sources providing relational features $\mathbf{M}^{(1)}, \cdots, \mathbf{M}^{(q)}$. The aim is to derive learning models (classification or clustering) by collectively using the $p + q$ feature spaces.

One innate challenge of heterogeneous learning is finding a low-dimensional embedding space. First, low-dimensional embedding is a necessary step to increase the learning accuracy and efficiency. Let us consider a straightforward strategy of utilizing the heterogeneous features by directly joining the $p+q$ sources together. In such a case, the dimension goes up to $\sum_i^p d_i+qm$, which is approximately $p+q$ times of the original dimension (if assuming that the data size is the same as the feature size). In a linear classification model, it means that the number of variables increases by $p+q$ times. Given a fixed number of labeled examples, it may easily encounter the under-fitting problem. In other words, the number of labeled examples is far from enough to determine the increasing number of model variables. Hence, it is necessary to reduce its overall dimension. Second, low-dimensional embedding is a necessary step to reduce noise. This is because different sources may have different data qualities. Some may contain substantial noise that hurt the learning performance. Hence, low-dimensional embedding is also a necessary step to dig out informative features to reduce noise. However, it is a very challenging task. The first challenge is how to find a unified reduced feature subspace based on multiple feature spaces. The second challenge is how to use the graph relational features in finding the feature subspace.

We devise a CoCA principle for finding the optimal embedding for heterogeneous data, and further perform low-dimensional embedding. The idea is derived from PCA, which aims at maximizing the variance of the projected data to maximize the data separation. In addition, we introduce two more constraints. The first is to force all heterogeneous feature spaces to be projected onto the same subspace. The second is to maximize the similarity of the data with connections as reflected by the relational features. Similar to PCA, the data are normalized to have zero mean, and the aim is to find the orthogonal linear projection basis $\mathbf{u}^{(j)}$ for all heterogeneous feature spaces ($j = 1, 2, \cdots$). The general objective function can be written as follows:

$$\max_{\mathbf{u}^{(1)},\mathbf{u}^{(2)},\cdots} \gamma(\mathbf{u}) + c(\mathbf{u}) + \alpha\chi(\mathbf{u}) \tag{4.1}$$

where $\gamma(\mathbf{u})$ is the variance of the projected data, which depends on the data and the projection basis $\mathbf{u} = [{\mathbf{u}^{(1)}}^T, {\mathbf{u}^{(2)}}^T, \cdots]^T$[1]. The second term $c(\mathbf{u})$ is the similarity among connected data. In other words, if the i-th data and the j-th data are connected in any of the graph data sources, they should also be similar in the projected space.

The last term $\chi(\mathbf{u})$ forces the projections from different data sources to be similar. More specifically, if the projections of the i-th and j-th data sources are Φ_i and Φ_j respectively, then the difference between Φ_i and Φ_j should be very small. For example, suppose we have two objects and two data sources, denote the projections learned from the two data sources are $\hat{\Phi}_1$ and $\hat{\Phi}_2$ respectively. Further assume that $\hat{\Phi}_1 = [0, 1]^T$ and $\hat{\Phi}_2 = [1, 0]^T$. If we maximize the similarity of the two projected spaces via

[1]The dependency of $\gamma()$ on the data is obvious, and we do not write it explicitly for simplicity.

$\chi(\mathbf{u})$, we will get the final projections as $\Phi_1 = \hat{\Phi}_1$ and $\Phi_2 = \hat{\Phi}_2\hat{I}$ where $\hat{I} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$. As such, both feature spaces are projected into $[0,1]^T$. It is equally likely to be both projected to $[1,0]^T$, but the structure (separation) of the data is the same. Hence, the term $\chi(\mathbf{u})$ tries to find a consensus feature subspace across all data sources. Moreover, α is a parameter to control how strongly we want the data to be projected onto a unified subspace.

Note that by solving Eq. Equation 4.1, we can obtain multiple projections $\mathbf{u}^{(j)}$ where $j = 1, 2, \cdots$ (one for each feature space). According to the CoCA principle, all these projections are similar to each other. Hence, the final projection is the expected value of the feature subspace:

$$\mathbf{\Phi} = \mathbb{E}(\mathbf{X}^T\mathbf{u}) = \frac{1}{n}\sum\nolimits_{j=1}^{n} \mathbf{X}^{(j)^T}\mathbf{u}^{(j)} \tag{4.2}$$

where n is the number of data sources (feature spaces), and $\mathbf{X}^{(j)^T}\mathbf{u}^{(j)}$ is the projected space of the j-th data source, and $\mathbf{\Phi} \in \mathbb{R}^{m\times d}$ is the desired unified feature subspace with reduced dimension d.

4.4 Collective Component Analysis

In the last section, we introduce collective component analysis principle for unsupervised heterogeneous embedding, which is described in Eq. Equation 4.1. In this section, for the sake of clarity, we first study the CoCA principle on two special cases where (1) all sources provide only vector-based features, and where (2) all sources provide only graph relational features. We then combine the two cases, and derive a general CoCA algorithm for vector-based features and relational features simultaneously. The most important challenges of the problem can be summarized as follows:

- How should we formulate the term $\chi(\mathbf{u})$ to force the similarity of the projected spaces?
- How should we solve the optimization in Eq. Equation 4.1, especially given that it contains multiple objectives?
- How should we reduce the impact of the noisy data sources?

4.4.1 CoCA on multiple vector-based feature spaces.

We first consider the problem of finding a consensus feature subspace across multiple sources with only vector-based features. Without loss of generality, we first discuss the case where the desired dimension is one.

Definition 1 *Given p data sources with different feature spaces $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$, $\cdots$, $\mathbf{X}^{(p)}$, the aim is to find a feature subspace that (1) maximizes the variance of the projected data, and that (2) maximizes the "agreement" of all projections:*

$$\begin{aligned}\max_{\mathbf{u}^{(1)},\cdots,\mathbf{u}^{(p)}} & \sum_{j=1}^{p}\omega_j\sum_{i=1}^{m}(\mathbf{x}_i^{(j)^T}\mathbf{u}^{(j)})^2 \\ & -\alpha\sum_{j=1}^{p}\sum_{h=1}^{p}\sum_{i=1}^{m}\|(\mathbf{x}_i^{(j)^T}\mathbf{u}^{(j)})-(\mathbf{x}_i^{(h)^T}\mathbf{u}^{(h)})\|^2 \\ & \textit{s.t. } \|\mathbf{u}^{(j)}\|=1 \textit{ where } j=1,\cdots,p\end{aligned} \tag{4.3}$$

where $\mathbf{x}_i^{(j)^T}\mathbf{u}^{(j)}$ is the length of the projection of $\mathbf{x}_i^{(j)}$ on $\mathbf{u}^{(j)}$, and $\|*\|$ is the Euclidean norm to ensure the agreement of projections from different data sources. In other words, although the same object (e.g., $\mathbf{x}_i$) has different features from different feature spaces (e.g., $\mathbf{x}_i^{(j)}$ and $\mathbf{x}_i^{(h)}$), it has only one projected value in the final subspace (i.e., to force $\mathbf{x}_i^{(j)^T}\mathbf{u}^{(j)} = \mathbf{x}_i^{(h)^T}\mathbf{u}^{(h)}$). Furthermore, ω_j is the weight of

importance of the j-th data source (or the j-th feature space). Ideally, we should give higher weights for informative sources. At the moment, we set all weights to be equal. In Section 4.4.4, a separate method is introduced to obtain optimal weights to better use the informative sources. Note that although the optimization problem in Eq. Equation 4.3 is straightforward, it is not easy to solve. We derive an equivalent optimization problem as follows:

Lemma 1 *The optimization problem in Eq. Equation 4.3 is equivalent to the following optimization problem:*

$$\begin{aligned}\max_{\mathbf{u}^{(1)},\cdots,\mathbf{u}^{(p)}} & \sum_{j=1}^{p}\omega_j\sum_{i=1}^{m}({\mathbf{x}_i^{(j)}}^T\mathbf{u}^{(j)})^2 \\ & + \alpha\sum_{j=1}^{p}\sum_{h=1}^{p}\sum_{i=1}^{m}({\mathbf{x}_i^{(j)}}^T\mathbf{u}^{(j)})^T({\mathbf{x}_i^{(h)}}^T\mathbf{u}^{(h)}) \\ & \textit{s.t. } \|\mathbf{u}^{(j)}\| = 1 \textit{ where } j = 1,\cdots,p\end{aligned} \tag{4.4}$$

Proof 3 *The difference between Eq. Equation 4.3 and Eq. Equation 4.4 is the last term in Eq. Equation 4.3 and the last term in Eq. Equation 4.4. We next prove that they are equivalent. Note that*

$$\begin{aligned}& -\|({\mathbf{x}_i^{(j)}}^T\mathbf{u}^{(j)}) - ({\mathbf{x}_i^{(h)}}^T\mathbf{u}^{(h)})\|^2 \\ =& -({\mathbf{x}_i^{(j)}}^T\mathbf{u}^{(j)})^T({\mathbf{x}_i^{(j)}}^T\mathbf{u}^{(j)}) - ({\mathbf{x}_i^{(h)}}^T\mathbf{u}^{(h)})^T({\mathbf{x}_i^{(h)}}^T\mathbf{u}^{(h)}) \\ & + 2({\mathbf{x}_i^{(j)}}^T\mathbf{u}^{(j)})^T({\mathbf{x}_i^{(h)}}^T\mathbf{u}^{(h)})\end{aligned} \tag{4.5}$$

Note that the first two terms can be incorporated into the first term in Eq. Equation 4.3, and the last term is controlled by the parameter α. Thus, maximizing Eq. Equation 4.3 is equivalent to maximize Eq. Equation 4.4.

The optimization problem in Eq. Equation 4.4 can be written in a matrix form as follows:

$$\begin{aligned}\max_{\mathbf{u}^{(1)},\cdots,\mathbf{u}^{(p)}} & \sum_{j=1}^{p} \omega_j {\mathbf{u}^{(j)}}^T \mathbf{X}^{(j)} {\mathbf{X}^{(j)}}^T \mathbf{u}^{(j)} \\ & + \alpha\Big(\sum_{j=1}^{p}\sum_{h=1}^{p} {\mathbf{u}^{(j)}}^T \mathbf{X}^{(j)} {\mathbf{X}^{(h)}}^T \mathbf{u}^{(h)}\Big) \\ & \text{s.t. } \|\mathbf{u}^{(j)}\| = 1 \text{ where } j = 1, \cdots, p\end{aligned} \tag{4.6}$$

Note that at the moment, we set the weight ω_j to be $\frac{1}{p}$. In Section 4.4.4, a separate method is introduced to obtain the optimal ω_j with some prior knowledge. Hence, the unknown variables in Eq. Equation 4.6 are the projection bases $\mathbf{u}^{(1)}, \cdots, \mathbf{u}^{(p)}$. We further rewrite the optimization problem into a more compact form:

$$\begin{aligned}\max_{\mathbf{u}} \ & \mathbf{u}^T \mathbf{X} \mathbf{u} \\ & \text{s.t. } \|\mathbf{u}^{(j)}\| = 1 \text{ where } j = 1, \cdots, p\end{aligned} \tag{4.7}$$

where $\mathbf{u} = [{\mathbf{u}^{(1)}}^T, \cdots, {\mathbf{u}^{(p)}}^T]^T \in \mathbb{R}^{(d_1+\cdots+d_p)\times 1}$ and

$$\mathbf{X} = \begin{bmatrix} \omega_1 \mathbf{X}^{(1)} {\mathbf{X}^{(1)}}^T & \alpha \mathbf{X}^{(1)} {\mathbf{X}^{(2)}}^T & \cdots & \alpha \mathbf{X}^{(1)} {\mathbf{X}^{(p)}}^T \\ \alpha \mathbf{X}^{(2)} {\mathbf{X}^{(1)}}^T & \omega_2 \mathbf{X}^{(2)} {\mathbf{X}^{(2)}}^T & \cdots & \alpha \mathbf{X}^{(2)} {\mathbf{X}^{(p)}}^T \\ \cdots & \cdots & \cdots & \cdots \\ \alpha \mathbf{X}^{(p)} {\mathbf{X}^{(1)}}^T & \alpha \mathbf{X}^{(p)} {\mathbf{X}^{(2)}}^T & \cdots & \omega_p \mathbf{X}^{(p)} {\mathbf{X}^{(p)}}^T \end{bmatrix} \tag{4.8}$$

To solve Eq. Equation 4.7, we first form its Lagrangian formula

$$\max_{\mathbf{u}} \ \mathbf{u}^T\mathbf{X}\mathbf{u} - \lambda\sum_{j=1}^{p}(\mathbf{u}^{(j)^T}\mathbf{u}^{(j)} - 1) = \mathbf{u}^T\mathbf{X}\mathbf{u} - \lambda\mathbf{u}^T\mathbf{u} + \lambda p \tag{4.9}$$

where λ is the Lagrangian multiplier associated with the constraints. It is well known that the eigenvector corresponding to the largest eigenvalue of $\mathbf{X}$ maximizes Eq. Equation 4.9 (e.g., in (23)). The final projected data is $\mathbf{\Phi} = \frac{1}{p}\sum_{j=1}^{p}\mathbf{X}^{(j)^T}\mathbf{u}^{(j)}$ as discussed in Eq. Equation 4.2. Note that although we analyze the case where the desired dimension is one, it is not difficult to extend it to multiple dimensions by using the eigenvectors of the top d eigenvalues.

4.4.2 CoCA on multiple relational feature spaces.

For relational features, the aim is to find a subspace that maximizes the similarity among the data with connections. In other words, we maximize the second term in Eq. Equation 4.1. Note that since there are no vector-based features, we do not have to find the linear projection basis $\mathbf{u}$ as in Section 4.4.1. Instead, we can find the projected data $\mathbf{\Phi}$ directly. The principle is: if data a and data b are connected, they should have similar projected values $\mathbf{\Phi}(a)$ and $\mathbf{\Phi}(b)$. Also note that since the projected data $\mathbf{\Phi}$ are normalized to have zero mean as in PCA, maximizing the similarity of $\mathbf{\Phi}(a)$ and $\mathbf{\Phi}(b)$ is approximately equivalent to maximizing their product $\mathbf{\Phi}(a)\cdot\Big(\mathbf{\Phi}(b)\Big)^T$, in order to ensure that they have the same sign. Hence, the optimization problem is as follows:

$$\begin{aligned} &\max\sum_{j=1}^{q}\tilde{\omega}_j\sum_{a=1}^{m}\sum_{b=1}^{m}\mathbf{M}^{(j)}(a,b)\cdot\mathbf{\Phi}(a)\cdot\Big(\mathbf{\Phi}(b)\Big)^T \\ &\text{s.t. } \|\mathbf{\Phi}\| = 1 \end{aligned} \tag{4.10}$$

where $\mathbf{M}^{(j)}(a,b)$ is large if data a and data b have strong similarity in the j-th relational graph (e.g., equals to 1 if connected, 0 otherwise) , and $\mathbf{\Phi}(a), \mathbf{\Phi}(b)$ are the projected features of a and b respectively. Moreover, $\tilde{\omega}_j$ is the weight of importance of the j-th source. In Section 4.4.4, a separate method is introduced to learn the best $\tilde{\omega}_j$ from the dataset. We rewrite Eq. Equation 4.10 in a matrix form as follows:

$$\begin{aligned} &\max \sum_{j=1}^{q} \tilde{\omega}_j \mathbf{\Phi} \mathbf{M}^{(j)} \mathbf{\Phi}^T = \mathbf{\Phi}\Big(\sum_{j=1}^{q} \tilde{\omega}_j \mathbf{M}^{(j)}\Big)\mathbf{\Phi}^T \\ &\text{s.t. } \|\mathbf{\Phi}\| = 1 \end{aligned} \tag{4.11}$$

Hence, the optimal $\mathbf{\Phi}$ is the set of eigenvectors of the top d eigenvalues of the matrix sum $\sum_{j=1}^{q} \tilde{\omega}_j \mathbf{M}^{(j)}$.

4.4.3 A general CoCA algorithm.

Section 4.4.1 and Section 4.4.2 introduce two special cases of the CoCA principle. In this section, we consider the general scenario where we have a mixture of vector-based features and relational features. The objective function contains three components as discussed in Eq. Equation 4.1. The first term maximizes the variance of the projected data. The second term is to ensure a common feature subspace

as in Section 4.4.1. The last term is to maximize the similarity of the data with connections as in Section 4.4.2. Hence, the objective function can be written as follows:

$$\begin{aligned}
&\max_{\mathbf{u}^{(1)},\cdots,\mathbf{u}^{(p)}} \sum_{j=1}^{p} \omega_j \sum_{i=1}^{m} (\mathbf{x}_i^{(j)^T} \mathbf{u}^{(j)})^2 \\
&+\alpha \sum_{j=1}^{p} \sum_{h=1}^{p} \sum_{i=1}^{m} (\mathbf{x}_i^{(j)^T} \mathbf{u}^{(j)})(\mathbf{x}_i^{(h)^T} \mathbf{u}^{(h)}) \\
&+\sum_{j=1}^{p} \sum_{a} \sum_{b} \Big(\sum_{t=1}^{q} \tilde{\omega}_t \mathbf{M}^{(t)}(a,b) \cdot \big(\mathbf{x}_a^{(j)^T} \mathbf{u}^{(j)}\big)^T \cdot \mathbf{x}_b^{(j)^T} \mathbf{u}^{(j)} \Big) \\
&\text{s.t. } \|\mathbf{u}^{(j)}\| = 1 \text{ where } j = 1, \cdots, p
\end{aligned} \tag{4.12}$$

where the first term is to maximize the variance, and the third term is to maximize the similarity of connected data, and the second term is to ensure the multiple feature spaces to be projected to the same feature subspace. In a matrix form, it can be written as:

$$\begin{aligned}
&\max_{\mathbf{u}^{(1)},\cdots,\mathbf{u}^{(p)}} \mathbf{u}^T \mathbf{X} \mathbf{u} + \mathbf{u}^T \tilde{\mathbf{X}} \mathbf{u} \\
&\text{s.t. } \|\mathbf{u}^{(j)}\| = 1 \text{ where } j = 1, \cdots, p
\end{aligned} \tag{4.13}$$

where $\mathbf{X}$ is as defined in Eq. Equation 4.8 of Section 4.4.1 and

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{Z}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{Z}_2 & \cdots & \mathbf{0} \\ \cdots & \cdots & \cdots & \cdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{Z}_p \end{bmatrix} \tag{4.14}$$

where

$$\mathbf{Z}_i = \mathbf{X}^{(i)}\Big(\sum_{t}^{q} \tilde{\omega}_t \mathbf{M}^{(t)}\Big)\mathbf{X}^{(i)^T}$$

Hence, the optimal solution is given by the eigenvectors of the top d eigenvalues of the matrix $\mathbf{X} + \hat{\mathbf{X}}$. The output is the expected value of the projections $\mathbf{\Phi} = \frac{1}{p}\sum_{j=1}^{p}(\mathbf{X}^{(j)} + \hat{\mathbf{X}}^{(j)})^T\mathbf{u}^{(j)}$.

It is important to note that Eq. Equation 4.12 is a general solution, and it includes Eq. Equation 4.7 in Section 4.4.1 and Eq. Equation 4.10 in Section 4.4.2 as special cases. In the experiment, we use Eq. Equation 4.12 to generate the reduced feature spaces since it can handle all three cases: (1) learning with only vector-based features, (2) learning with only relational features, and (3) learning with both vector-based features and relational features.

4.4.4 Weight learning.

In Eq. Equation 4.12, we set the weights $\omega_j = \frac{1}{p}$ and $\tilde{\omega}_t = \frac{1}{q}$ where $1 \leq j \leq p$ and $1 \leq t \leq q$. In other words, all sources providing vector-based features have the same weight; all sources with relational features have the equal weight. However, in some cases, some of the sources may be more informative than the others. In this situation, a better strategy is to increase the weights of the informative sources, in order to obtain a more informative feature subspace. In the following, we study how to obtain the optimal weights with the "must-link" and "cannot-link" constraints as in semi-supervised learning (77). We first introduce how to learn optimal weights for data sources with vector-based features. The general idea is to first use PCA to obtain projections $\mathbf{\Phi}^{(k)} = \mathbf{X}^{(k)^T}\mathbf{u}^{(k)}$ individually for each source where $\mathbf{u}^{(k)}$

is given by PCA, and then find which feature subspaces $\mathbf{\Phi}^{(k)}$ can better satisfy the constraints. To do so, we first denote a $m \times m$ constraint matrix $\mathbf{C}$ as:

$$\mathbf{C}(i,j) = \begin{cases} 1 & \text{if the } i\text{-th and } j\text{-th data should have} \\ & \text{similar projected values} \\ 0 & \text{no preference} \\ -1 & \text{if the } i\text{-th and } j\text{-th data should have} \\ & \text{dissimilar projected values} \end{cases} \tag{4.15}$$

In other words, $\mathbf{C}(i,j) = 1$ if it is a "must-link" constraint; $\mathbf{C}(i,j) = -1$ if it is a "cannot-link" constraint; otherwise it is 0. We further define a *truncated similarity matrix* $\mathbf{E}^{(k)}$ for the k-th source such that

$$\mathbf{E}^{(k)}(i,j) = |\mathbf{C}(i,j)| \times \left(\mathbf{\Phi}^{(k)}(i)\mathbf{\Phi}^{(k)}(j)^T\right) \tag{4.16}$$

where $|*|$ takes the absolute value, and hence $|\mathbf{C}(i,j)| = 1$ iff the i-th data and the j-th data have certain constraint, regardless it is a "must-link" or a "cannot-link". Furthermore, $\mathbf{\Phi}^{(k)} = \mathbf{X}^{(k)^T}\mathbf{u}^{(k)}$ is the reduced subspace from the k-th source given by PCA. Hence, if the i-th data and the j-th data do not have any constraint, $\mathbf{E}^{(k)}(i,j) = 0$; otherwise

$$\mathbf{E}^{(k)}(i,j) = \left(\mathbf{\Phi}^{(k)}(i)\right)\left(\mathbf{\Phi}^{(k)}(j)\right)^T \tag{4.17}$$

We call $\mathbf{E}^{(k)}$ "truncated similarity matrix" since it only shows the similarity of the data with constraints. Moreover, since $\mathbf{\Phi}^{(k)}$ has zero mean as in PCA, $\mathbf{E}^{(k)}(i,j) > 0$ iff the i-th and the j-th data have

similar projected values of the same sign, and $\mathbf{E}^{(k)}(i,j) < 0$ iff they are with opposite signs. The ideal truncated similarity matrix should have the same values as the constraint matrix. Hence, given the truncated similarity matrices $\mathbf{E}^{(k)}$ from all sources where $k = 1, 2, \cdots$, the objective function is to find a linear combination strategy that best satisfies the constraint matrix $\mathbf{C}$:

$$\begin{aligned} &\min_{\omega_{\mathbf{1}},\omega_{\mathbf{2}},\cdots} \|\Big(\sum_{k=1}^{p} \omega_k \mathbf{E}^{(k)}\Big) - \mathbf{C}\|_F^2 \\ &\text{s.t. } \textstyle\sum_{k=1}^{p} \omega_k = 1, \omega_k \geq 0 \text{ for } k = 1, \cdots, p \end{aligned} \tag{4.18}$$

where $\| * \|_F^2$ is the Frobenius norm. The sources with better truncated similarity matrices are learned to have higher weights, and we use these weights in the general CoCA algorithm Eq. Equation 4.12. We next solve Eq. Equation 4.18.

Theorem 3 *The optimization in Eq. Equation 4.18 is equivalent to the following quadratic programming problem:*

$$\begin{aligned} &\min_{\omega} \omega^T(\mathbf{K})\omega - 2\mathbf{v}^T\omega \\ &\textit{s.t. } \textstyle\sum_{k=1}^{p} \omega_k = 1, \omega_k \geq 0 \textit{ for } k = 1, \cdots, p \end{aligned} \tag{4.19}$$

where $\mathbf{K}$ *is a* $p \times p$ *matrix, and* $\mathbf{v}$ *is a* $p \times 1$ *vector such that*

$$\mathbf{K}(i,j) = tr\Big((\mathbf{E}^{(i)})^T\mathbf{E}^{(j)}\Big)$$

$$\mathbf{v}(i) = tr(\mathbf{C}\mathbf{E}^{(i)})$$

Proof 4 *The optimization function in Eq. Equation 4.18 can be written as*

$$\begin{aligned}
&\|\Big(\sum_{k=1}^{p}\omega_k\mathbf{E}^{(k)}\Big)-\mathbf{C}\|_F^2\\
=&tr\Big(\big(\sum_{k=1}^{p}\omega_k\mathbf{E}^{(k)}\big)^T\big(\sum_{k=1}^{p}\omega_k\mathbf{E}^{(k)}\big)\\
&-2\mathbf{C}^T\big(\sum_{k=1}^{p}\omega_k\mathbf{E}^{(k)}\big)+\mathbf{C}^T\mathbf{C}\Big)\\
=&\omega^T(\mathbf{K})\omega-2\mathbf{v}^T\omega+tr(\mathbf{C}^T\mathbf{C})
\end{aligned}$$

Note that $tr(\mathbf{C}^T\mathbf{C})$ is a constant. Hence, the optimization in Eq. Equation 4.18 is equivalent to that Eq. Equation 4.19.

The quadratic programming problem in Eq. Equation 4.19 can be conveniently solved with standard solvers such as MATLAB. Note that if the given heterogeneous features are relational features $\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \cdots$, we can replace $\mathbf{\Phi}^{(k)}\mathbf{\Phi}^{(k)^T}$ by $\mathbf{M}^{(k)}$ in Eq. Equation 4.17 with some modifications on $\mathbf{M}^{(i)}$. We have to set $\mathbf{M}^{(k)}(i,j) = -1$ if the i-th and the j-th data is disconnected. This modification is to reflect the "cannot-link" constraints when forming $\mathbf{E}^{(k)}$. All the formulas and properties still hold for relational features.

4.4.5 Algorithm flow and complexity analysis

The CoCA approach is described in Algorithm 2. It takes the data from various sources as input, and its aim is to construct an embedding with reduced dimensions. The algorithm has an optional input, which is the constraint matrix $\mathbf{C}$. If $\mathbf{C}$ is available, the CoCA algorithm can take advantage of $\mathbf{C}$ to learn the optimal weights (ω's) as in Eq. Equation 4.19; otherwise, CoCA assigns equal weights to the sources to learn the target feature subspace. It is important to mention that although the CoCA algorithm applies

Algorithm 2 A general algorithm with the CoCA principle

Input: p sources with vector-based features: $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \cdots, \mathbf{X}^{(p)}$, q sources with relational features: $\mathbf{M}^{(1)}, \mathbf{M}^{(2)}, \cdots, \mathbf{M}^{(q)}$ where either p or q can be zero, the parameter α and the desired dimension d, and the constraint matrix $\mathbf{C}$ (optional).

Output: The dataset with reduced dimension $\mathbf{\Phi}$.

```
if the constraint matrix C is available then
    Learn ω's and ω̃'s as in Eq. Equation 4.19;
else
    if p = 0 then
        Set ω's to be 0 and ω̃'s to be 1/q;
    else
        if q = 0 then
            Set ω's to be 1/p and ω̃'s to be 0;
        else
            Set ω's to be 1/p and ω̃'s to be 1/q;
        end
    end
end
Construct the matrix X as in Eq. Equation 4.8;
Construct the matrix X̃ as in Eq. Equation 4.14;
Find the eigenvectors u of the top d eigenvalues of the matrix X + X̂;
Construct Φ with u as in Eq. Equation 4.2.
```

quadratic programming to obtain the optimal weights, this step can actually be finished in constant time. This is because the complexity of the quadratic programming solver (e.g., MATLAB) is $O(u^3)$ where u is the number of unknown parameters, which is equal to the number of sources. In practice, the number of sources is usually not large, and it can be viewed as a constant. Hence, the quadratic programming step can take constant time to finish. The most expensive step of the algorithm is to get the eigenvectors. The complexity of the whole algorithm is $O\big(dIN\big)$ where d is the number of eigenvectors desired, I is

the number of Lanczos iteration steps, and N is the non-zero entries of the matrix. In the worst case, $N = (\sum_j d_j)^2$ as $\mathbf{X} + \hat{\mathbf{X}}$ in Eq. Equation 4.12 where $\sum_j d_j$ is the total dimension of all feature spaces.

4.5 Experiments

In this section, we analyze the proposed model Collective Component Analysis (CoCA) on three sets of datasets with heterogeneous features.

4.5.1 Comparison Approaches.

Since there was no previous model that can be directly used to handle the same problem, we compared CoCA with a straightforward strategy. The comparison strategy is to directly join all features together. In other words, given the sources with vector-based features $\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(p)}$ and sources with relational features $\mathbf{M}^{(1)}, \cdots, \mathbf{M}^{(q)}$, the joined features can be represented as follows:

$$\mathbf{X} = [\mathbf{X}^{(1)^T}, \cdots, \mathbf{X}^{(p)^T}, \mathbf{M}^{(1)^T}, \cdots, \mathbf{M}^{(q)^T}]^T \tag{4.20}$$

Hence, traditional dimensionality reduction methods can be applied on the joined features to obtain a reduced space. Note that CoCA is a general principle to handle heterogeneous features. The principle can be incorporated by most of the unsupervised dimensionality reduction models by adding the projection consensus and graph constraints. In this paper, we take PCA as an example. In other words, the proposed CoCA model adopts PCA's strategy to maximize the variance of the data but with the CoCA principle, and PCA was used as baseline that run on the joined dataset (Eq. Equation 4.20).

4.5.2 Evaluation Strategy.

Note that the output of CoCA is the dataset with a reduced feature space. In order to evaluate the quality of the reduced feature space, we performed clustering on the output, and the clustering result was evaluated by normalized mutual information (NMI). Note that NMI equals to zero when clustering is random, and it is close to one when the clustering result is good. Furthermore, k-means was chosen as the based clustering algorithm since it is efficient and widely used in practice. Note that k-means is sensitive to initial seed selection. Hence, we run k-means 10 times on each parameter setting, and report the summarized NMI with mean value and standard deviation. It is also important to note that we propose two versions of CoCA, which are different in how to obtain the weights of the sources. In other words, we have two versions of CoCA depending on whether the optional constraint matrix $\mathbf{C}$ in Algorithm 2 is available or not. If the constraint matrix $\mathbf{C}$ is not available, we directly assign equal weights to the sources; otherwise we learn the optimal weights as in Section 4.4.4. In the experiment, we denote the CoCA without the constraint matrix $\mathbf{C}$ as "CoCA (equal weights)" and the one with $\mathbf{C}$ as "CoCA (unequal weights)". To construct constraint matrix $\mathbf{C}$, we sampled 30% data and generated "must-links" among the data with the same class label, and "cannot-links" among the data with different labels. We report the results with increasing number of desired dimensions.

4.5.3 Handwritten dutch numbers recognition.

The first dataset contains 2000 handwritten numerals ("0"–"9") extracted from a collection of Dutch utility maps (13). The handwritten numbers are scanned and digitized as binary images. They are represented in terms of the following six feature spaces (six sources) with different vector-based features: (1) 76 Fourier coefficients of the character shapes, (2) 216 profile correlations, (3) 64 Karhunen-Love

coefficients, (4) 240 pixel averages in 2×3 windows, (5) 47 Zernike moments, and (6) 6 morphological features. All these features are conventional vector-based features but in different feature spaces. The aim is to find the optimal embedding with reduced dimensions collectively with the six different sources. As mentioned before, we use two versions of CoCA to find the reduced feature spaces. One is to assign equal weights to the sources, and the other is to learn the optimal weights from prior knowledge. The results are presented in Fig. 11. As it can be observed, CoCA outperforms the comparison method substantially. For example, when the dimension is around 8, the NMI obtained from the two versions of CoCA are around 0.65 while that of the comparison method is only around 0.45. This shows that CoCA performs better than the intuitive strategy which directly uses the joined features. Between the two versions of CoCA, the unequal-weight solution is noticeably better, especially when the dimension is low. This is because some prior knowledge ("must-links" and "cannot-links") is used to emphasize those informative sources and reduce noise.

4.5.4 Citeseer document dataset.

In this experiment, we studied high-dimensional text data. The dataset (12) is originally from Citeseer with 3312 scientific papers under six different topics: Agents, AI, DB, IR, ML, and HCI. It includes two types of features. One is the vector-based features that describe the papers with "bag of words" representation. In other words, each feature represents one word and the feature value reflects the presence or absence of the word in the document. The relational features come from the citations that link papers together if they have citation relations, which are converted into an undirected graph. Note that in order to examine the weight learning algorithm described in Section 4.4.4, we split the vector-based "bag of words" representation into five different subspaces. This was done by first calculating the mutual

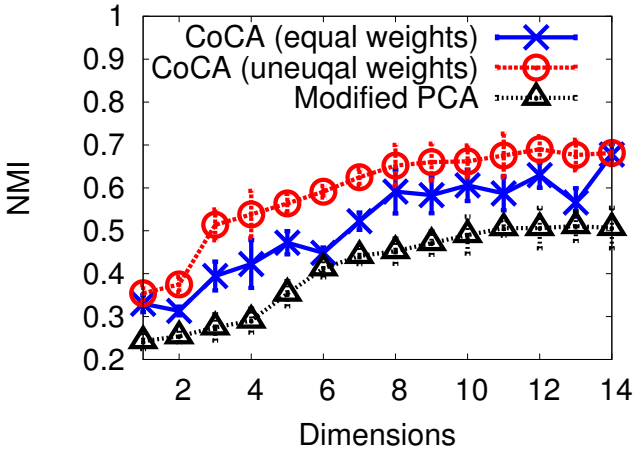


Figure 11. Handwritten numbers recognition.

information of each feature (word) with respect to the class labels (topics), and then sorting them in descending order. We then split the sorted features into five groups where the first group contains the top 20% of features with the largest mutual information, and the second group is the following 20% of features, and so on. Furthermore, we randomly generated additional sets of vector-based features and relational features by generating matrices with binary entry values (0 or 1) randomly. As a result, we have 8 sources which contain 6 sets of vector-based features and 2 sets of relational features. Note that some of the sources are highly correlated with the class label (informative), while others are very noisy. The weight learning algorithm discussed in Section 4.4.4 is supposed to be able to identify the informative sources.

The results summarized in 10 runs are presented in Fig. 12 with the changing value of dimensions. Since we were dealing with a document dataset that has over 3000 features, we thus extended the

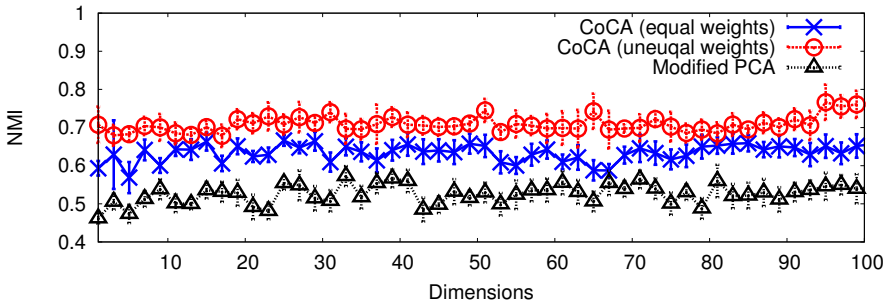


Figure 12. Citeseer dataset.

maximal dimension up to 100. As it can be observed, the CoCA with unequal weights has the best performance while the CoCA with equal weights ranks the second. Both approaches beat the comparison modified PCA substantially and significantly. It shows that the naïve strategy is far from enough to build an accurate learner. Instead, CoCA can collectively take different sources into consideration, and obtain a better reduced feature space. Furthermore, some prior knowledge can help CoCA get better weights of the sources, and the results are further improved.

4.5.5 Terrorist attack detection.

The third dataset is about terrorist attack (12) that consists of 1293 different attacks in one of the six labels indicating the type of the attack: arson, bombing, kidnapping, NBCR attack, weapon attack and other attack. Each attack is described by a binary value vector of attributes whose entries indicate the absence or presence of a feature. There is a total of 106 distinct vector-based features, along with two sets of relational features. One set connects the attacks from the same location, and the other connects the attacks if they are planned by the same organization. As in the previous experiment, we split the vector-based features into five different groups, and randomly generated one additional set of vector-

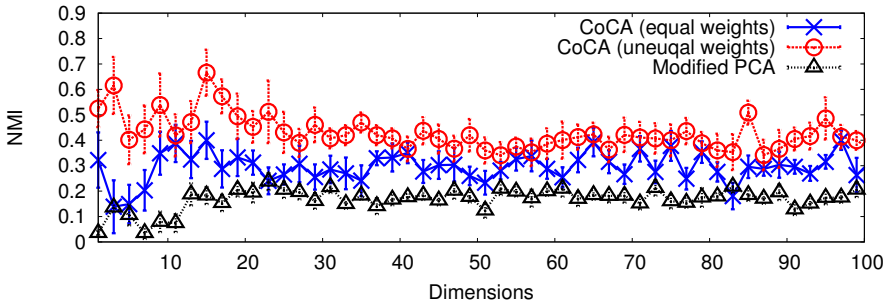


Figure 13. Terrorist attack detection dataset.

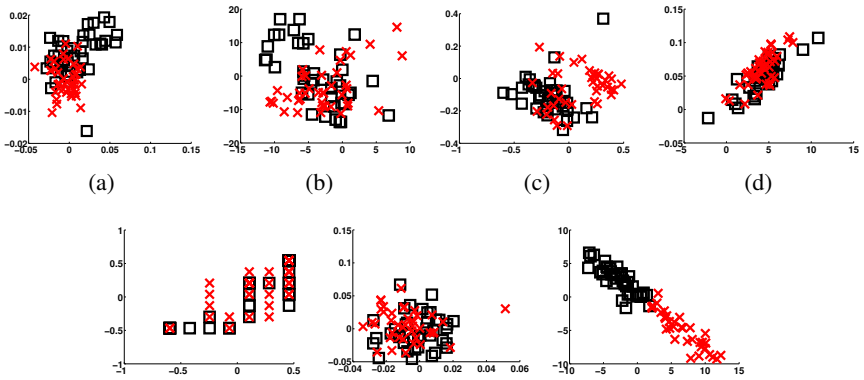
based features and one additional set of relational features. In summary, this dataset contains features from 9 sources including 6 sets of vector-based features and 3 sets of relational features.

The results with the changing value of dimensions are presented in Fig. 13. It is clear that the equal-weight CoCA beats the modified PCA by as much as 50%, and the weighted CoCA can beat the comparison model by as much as three times in NMI. Furthermore, when the number of features reaches 15, the proposed model gets the best performance. Note that selecting the optimal dimension is a nontrivial model selection problem, but it is beyond the scope of the present paper. In practice, one can apply cross-validation or validation set to obtain the best dimension.

4.5.6 Discussion

In this section, we aim at analyzing CoCA more in detail in order to answer the following three questions:

1. How does the embedding look like geometrically?



(a) (b) (c) (d)

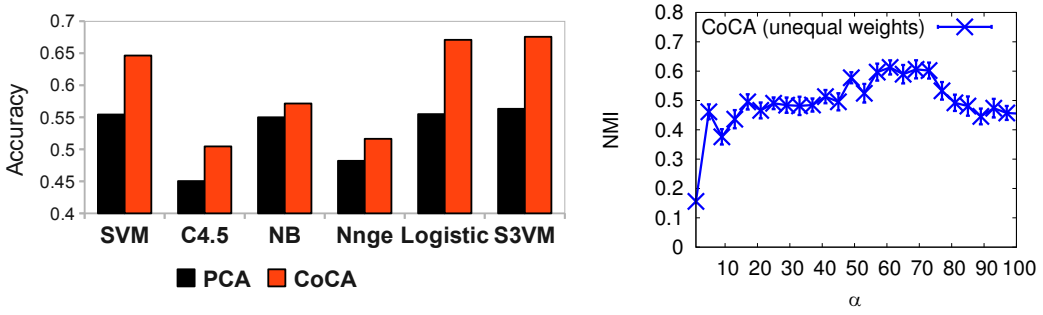
(e) (f) (g) Projected feature space

Figure 14. Samples of the projected data: □s are the data of the same class, while Xs are the data of another class.

2. Can the embedding data with reduced dimensions work well on other learners (such as different classifiers in classification problem)?

3. How does the parameter affect the algorithm?

Fig. 14 plots a subset of the hand-written number data in certain feature spaces. The black squares represent the data belonging to one class, while the red crosses are the data of another class. Specifically, Fig. 14(a)∼Fig. 14(f) are the feature spaces from the six sources, and the shown dimensions are the ones that are most correlated with the class labels. In other words, in each source, the two best features (evaluated by mutual information) were chosen to show in Fig. 14. It can be clearly observed that some feature spaces are relatively informative (e.g., Fig. 14(a) and Fig. 14(c) help separate part of the data), while some are quite noisy (e.g., Fig. 14(e)). However, the final reduced feature space in Fig. 14(g) best captures the characteristics of the dataset by taking all sources into consideration.



(a) Classification capability (b) Parameter sensitivity

Figure 15. Classification capability and parameter sensitivity analysis

Note that in the experiment section, we evaluated the embedding with reduced dimensions via clustering algorithm k-means. It is also interesting to see how the low-dimensional data performs on classification problems. Hence, we used the 10-classes hand-written number recognition dataset as an example to analyze the classification capability of the low-dimensional data. To isolate the effect of prior knowledge, we considered the equal weight CoCA. The modified PCA was again used as a comparison method. Several classifiers were trained on the dataset with reduced dimensions and we compared their average accuracy with 10-fold cross validation. The results are shown in Fig. 15(a) with 6 different classifiers SVM, C4.5, Naive Bayes, Nearest Neighbor, Logistic Regression and the semi-supervised SVM. All six experiments show that CoCA better captures the characteristics of the data and obtains a better classification accuracy. For example, when the accuracy given by the modified PCA is only 55% with SVM, the accuracy of CoCA reaches 65%, which is 18% better.

In the proposed algorithm, we have one parameter α that controls how much we want the different feature spaces to be projected onto the same subspace. We use the terrorist dataset as an example and

plot the results with different values of α as shown in Fig. 15(b). It can be observed that when α is small, the result does not look good because the heterogeneous feature spaces are not bounded to find a consensus subspace. When α is around 60, the result tends to be stable, and we choose α to be 60 in the experiment. However, when α is larger than 75, the performance drops a little because the learning model encounters overfitting by forcing all feature spaces (including the noisy ones) "vote" for the consensus subspace. In practice, one can use cross-validation or directly use the root mean square error (the objective function of k-means) as a criterion to choose α. In the experiment, we chose $\alpha = 60$.

4.6 Conclusion

In this paper, we study the problem of finding the optimal embedding given multiple sets of vector-based features and multiple sets of graph relational features. We propose a CoCA principle to solve the problem by finding a reduced feature subspace that maximizes the variance of the data, with two constraints. First, all the heterogeneous feature spaces have to be consensually projected on the same subspace. Second, if relational features are available, the data with connections are preferred to be similar in the projected space. An optimization problem is derived from the CoCA principle, and the solution is obtained by solving an eigenvalue problem. Furthermore, we extend the method by using prior knowledge to identify better data sources and construct informative feature subspaces. Three sets of experiments were performed to evaluate CoCA. It can be clearly observed that the proposed CoCA model outperforms the comparison algorithm by as much as 50%, and the weighted CoCA beats the comparison model even more (by as much as three times in NMI). We also analyze the performance of CoCA on classification problems in the discussion section, where we can observe an 18% gain over comparison model.

CHAPTER 5

HETEROGENEOUS PROJECTION AND ITS APPLICATIONS

In this chapter, we explore the approaches to learn from heterogeneous feature spaces.

5.1 Introduction

In supervised and semi-supervised learning, there is usually a large gap between the number of labeled examples needed to obtain high prediction accuracy, and the number of labeled examples that could be realistically obtained. These problems can be found in web mining, behavior targeting, spam filtering, objective recognition, and bioinformatics applications. At the same time, however, there is usually a large number of labeled examples from various related applications, such as, labeled documents in social tagging systems (eg, wikipedia, ODP) for web mining, labeled chemical compounds from NCI database for bioinformatics, and classified images from social sites such as Flickr for object recognition and image classification. One may ask: can these free labeled source data provide useful supervision to a related target task? Three challenging sub-issues need to be solved:

1. The source data may be generated from a different feature space from the target data (e.g., source is text data while target is image data).
2. The source data may be drawn from a distribution different from the target data. For example, the source data is dominated by a Gaussian distribution while the target data is dominated by a multinomial distribution, which violates the i.i.d. assumption.
3. The source and target data may have totally different output spaces.

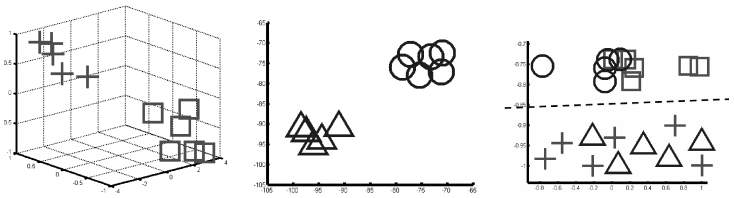
(a) Bacteria genome sequence expression

(b) Yeast DNA microarray

Figure 16. Two related examples with (1) heterogeneous feature spaces and data distributions; (2) heterogeneous output spaces

We illustrate in Figure 16 as an example. The left figure describes certain bacteria genome sequence, while the right figure is the microarray expression data of certain yeast. The two data sets are heterogeneous because: (1) different feature spaces and distributions: one has text like features, while the other is microarray expression with numerical features; (2) different output spaces: one is labeled by the category of bacteria, while the other is labeled by the category of yeast. Intuitively, however, the two heterogeneous data sets may provide some useful knowledge to each other. The simple intuition is that yeast and bacteria all belong to micro-organisms, and they may share some homologue genes. But the question is how to handle the heterogeneity issues and find their similarities.

Among the three sub-issues, the problem on unifying completely different feature spaces is most challenging, which, so far as we know, has not been specifically addressed. There are some works (e.g. (9; 72)) proposed to apply image tags to align image data and text data; but the algorithms rely on image tags to build up the alignment and are especially designed for image and text data, which is not for general case. There are also some works proposed to solve the other two problems separately. For example, multi-task learning (e.g., (8; 6)), transfer learning (e.g., (4; 14)), etc, are proposed to handle the case where data distributions are different but mainly in the same feature space. Multi-view



(a) 3-D data (b) 2-D data (c) Projected space

Figure 17. An Illustration of Feature Projection

learning (7; 45) is proposed to manipulate data with different feature spaces, but the data distributions and class labels are assumed the same. In addition, (60) applies the labeled data with different output spaces to help learn the target task, but also in the same feature space. Different from these works, one focus of this work is to investigate a general model to use the data from completely different feature space. We also integrate and improve (60) to deal with data with different distributions and output spaces. A summary is presented in Table VI. The objective of the proposed model is to solve the three sub-issues simultaneously, and automatically draw related heterogeneous training examples for a given target task. It is also important to note that in some transfer learning applications, certain concrete objects are used as bridges to enable knowledge transfer. For example, in document classification, words are used as bridges to enable knowledge transfer. However, in this paper, we do not assume the "bridge" is known. Instead, the "bridge" is automatically detected, and it rejects the transfer if the "bridge" cannot be found, implying that the two tasks are too different (the last step of the algorithm in Fig. 3). This step helps alleviate negative transfer.

TABLE VI. Different learning schemas

Learning Schema	Data Distributions	Feature Spaces	Class Labels
Transfear Learning	Different	Same	Same
Multi-view Learning	Same	Different	Same
Cross-label transfer learning (60)	Different	Same	Different
The proposed model	Different	Different	Different

The main idea is to find a common feature subspace for two heterogeneous tasks. For instance, given the 3-dimension data as in Figure 17(a) and the 2-dimension data as in Figure 17(b), the proposed model explores a common projected space as in Figure 17(c) in which (i) the original structure of the data is preserved where discriminative examples are still far apart, and (ii) the two distributions are similar in the projected space even they look different in their respective original 3-dimension and 2-dimension spaces. The conjecture here is that if the two sets of data are similar in distribution while still keep their original data structure, they should share similar decision boundaries in the projected space as in Figure 17(c). Hence the key requirement is to find a sound transformed feature subspace that maximizes the similarity among the two data sets, and preserves the intrinsic similarity among the source examples, as well as that among the target examples. Two solutions are proposed to find the optimal projection. The first approach explores the projected data from a strict linear transformation of the original data. The second approach relaxes the linear assumption by ensuring and checking that the clustering structure is preserved after spectral transformation; that is, instances in the same cluster are still together in the projected space. Both approaches are derived as trace norm maximization problems solved with their closed forms resulting from the selected eigenvectors. Finally, in the projected space,

a sample selection strategy is applied to select only the related examples as new training data, and a Bayesian-based approach is derived to model the relationship between different output spaces. The algorithm flow is summarized in Figure 18. As such, the proposed algorithms can extract training data from related heterogeneous sources to help learn the target data. Experiments involve 20 data sets, including drug efficacy prediction, image classification, etc. For example, in image classification, the data set constructed from wavelet space is applied to improve the accuracy of another category of images constructed from histogram space by around 20%.

5.2 Related Work for HeMap

There are mainly two research areas that are related to the proposed model. The first one is Multi-view learning (e.g., (7; 45)). For example, in (45), a co-training algorithm is proposed to classify the web pages by the text on the web page, and the text on hyperlinks pointing to the web page. Furthermore, there is another type of method called canonical correlation analysis (e.g., (64)), which aims at finding a hidden space to maximize the correlation between the data with different types of features. However, it is assumed that (1) the data from different views are the same set of objects, and that (2) the data correspondences are already given. In other words, the multi-view data are the same set of objects only with different feature spaces. Different from these works, we study the problem to unify the feature spaces of two sets of instances, and there may be no correspondence between instances in these spaces.

Another related field is transfer learning (e.g., (19; 2; 47)) which aims at learning the target task from a related out-of-domain source task. A survey of transfer learning can be found at (48). One line of research is to find a new feature space in which the training and test data have strong similarities (e.g., (10; 27; 36; 28; 70)). For instance, (2) applies sparse learning techniques to transfer the knowledge

across multiple tasks. Furthermore, (47) proposes to find a RKHS feature space to enable transfer learning via maximum mean discrepancy. Recently, (70) proposes a projection model to find the alignment of manifolds to enable knowledge transfer. Different from this work, we do not just force the projections from different domains to be totally the same. Instead, we also require the projections to keep the original structure of the data in order to reduce negative transfer. Furthermore, (14) is among the first works to exploit using second-order logic for transfer learning based on relational network data. There is also a set of works enable transfer learning on document-related tasks. For example, (27) finds the commonality among different tasks, such as common words, to attack NLP tasks. However, documents are usually viewed as coming from the same feature space, since any document can be represented as "bag of words" where the dictionary contains all possible words. Different from these works, we do not require the original training and test data are in the same feature space, or have a subset of common features. Instead, they can be from completely different feature spaces such as the image and text data. Recently, (72; 9; 81) propose methods to improve image clustering and classification with the help of text data. The major idea is to take advantage of the image tags to build up the relationship between image and text, and further use text features to enrich image features to improve the learning accuracy. There are at least two differences between this paper and (72; 9; 81). First, the proposed models aim at studying in a more *general* setting which is not limited to image and text data. Second, no auxiliary source (such as image tags) is given to provide clue of the correlation among the instances in different feature spaces. In addition to the feature based transfer learning techniques, the proposed model is also related to a set of research works that focus on knowledge transfer among data sets with different outputs (e.g., (60; 49)). For example, (49) studies the problem of using arbitrary images to help find a

TABLE VII. Notation Descriptions for HeMap

Notations	Descriptions
$\mathbf{T}$	Target data set (target data matrix)
$\mathcal{Y}$	Target output space
$\mathbf{B_T}$	Projected target data set
$\mathbf{P_T}$	Linear mapping function (to the target data space)
$\mathbf{C_T}$	Partition matrix of the target data
$\mathbf{P_{TC}}$	Linear mapping function (to the space of the target partition matrix)
$\mathbf{S}$	Source data set (source data matrix)
$\mathcal{V}$	Source output space
$\mathbf{B_S}$	Projected source data set
$\mathbf{P_S}$	Linear mapping function (to the source data space)
$\mathbf{C_S}$	Partition matrix of the source data
$\mathbf{P_{SC}}$	Linear mapping function (to the space of the source partition matrix)

more robust basis for a new image classification task. (60) finds a latent space for the outputs and tries to align the different outputs in the latent space. Although we also study the case when the source and target data sets have different outputs, the focus of the paper is how to unify the different feature spaces. We next introduce the problem and the general optimization objective.

5.3 Problem Formulation

Denote the target data set as $\mathbf{T} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_r)^\top$ where $\mathbf{x}_i \in \mathbb{R}^m$ is a column vector data, and $\mathbf{x}_i \sim p_t(\mathbf{x})$ is drawn from a distribution $p_t(\mathbf{x})$. From a matrix point of view, we use $\mathbf{T}(i, j)$ or $[\mathbf{T}]_{i,j}$ to indicate the j-th feature of the i-th target instance, which is also the (i, j)-th element of the matrix $\mathbf{T}$. Let the outputs of the target data (i.e., class labels or regression values) be $\mathbf{Y} = (y_1, y_2, \cdots, y_r)^\top$ where $y_i \in \mathcal{Y}$ is the output of x_i, and it is drawn from the output space $\mathcal{Y}$. Assume that in the model training

process, we can only observe the outputs of the first t target data where $t \ll r$. Our goal is then to use the t target training data to predict the outputs of the remaining $r - t$ test data. Furthermore, we are also given an auxiliary source data set $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_q)^\top$, in which $\mathbf{s}_i \in \mathbb{R}^n$ is drawn from the marginal distribution $p_s(\mathbf{s})$. Denote the outputs of source data as $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_q)$, where the output of $\mathbf{s}_i$ is $\mathbf{v}_i \in \mathcal{V}$ drawn from the output space $\mathcal{V}$. The important notations are summarized in Tabel VII. Note that one assumption in the above formulation is that the source and target data are all continuous such that $\mathbf{x}_i \in \mathbb{R}^m$ and $\mathbf{s}_i \in \mathbb{R}^n$.

Note that the target data $\mathbf{x} \in \mathbb{R}^m$ and source data $\mathbf{s} \in \mathbb{R}^n$ have (1) different feature spaces ($\mathbb{R}^m \neq \mathbb{R}^n$), (2) different data distributions ($p_t(\mathbf{x}) \neq p_s(\mathbf{s})$), and (3) different output spaces ($\mathcal{Y} \neq \mathcal{V}$). The heterogeneous source data cannot be directly used as training data to learn the target task. In this paper, we mainly focus on resolving the feature heterogeneity issue by investigating a common feature space for the source and target data. We then apply and improve the method proposed in (60) to tackle the problem of different data distributions and output spaces.

General Optimization Objective We aim at finding a common feature subspace for the source and target data. The optimal projected space is defined as follows:

Definition 2 *Given the target data matrix* $\mathbf{T}$ *and the source data matrix* $\mathbf{S}$*, the optimal projection of the target data* $\mathbf{B_T}$*, and that of the source data* $\mathbf{B_S}$ *are given by the following optimization objective:*

$$\min_{\mathbf{B_T}, \mathbf{B_S}} \ell\big(\mathbf{B_T}, \mathbf{T}\big) + \ell\big(\mathbf{B_S}, \mathbf{S}\big) + \beta \cdot \mathbf{D}\big(\mathbf{B_T}, \mathbf{B_S}\big) \tag{5.1}$$

where $\mathbf{B_T} \in \mathbb{R}^{r\times k}, \mathbf{B_S} \in \mathbb{R}^{q\times k}$ *are the projected matrices of* $\mathbf{T}$ *and* $\mathbf{S}$ *respectively. Furthermore,* $\ell(*,*)$ *is a distortion function that evaluates the difference between the projected data and the original data (e.g.,* $\mathbf{B_T}$ *and* $\mathbf{T}$*). The difference between the two projected data sets are denoted as* $\mathbf{D}\big(\mathbf{B_T}, \mathbf{B_S}\big)$*. It is important to mention that* $\mathbf{B_T}$ *and* $\mathbf{B_S}$ *are obtained by linear transformations (discussed in Section 5.4) such as rotation, scaling, permutation of row vectors and column vectors, etc. The initial instance order (or order of row vectors) and scale of the original target and source datasets* $\mathbf{T}$ *and* $\mathbf{S}$ *will not affect the result. The algorithms are expected to find the optimal linear projection by using any possible operations. In Eq. Equation 5.1,* β *is a parameter to control how desirable the two data sets are similar. We further define* $\mathbf{D}\big(\mathbf{B_T}, \mathbf{B_S}\big)$ *in terms of* $\ell(*,*)$ *as*

$$\mathbf{D}(\mathbf{B_T}, \mathbf{B_S}) = \frac{1}{2}\big(\ell(\mathbf{B_T}, \mathbf{S}) + \ell(\mathbf{B_S}, \mathbf{T})\big) \tag{5.2}$$

which is the average of the difference between the projected target data and the original source data, and that between the projected source data and the original target data.

Note that the definition of $\mathbf{D}(\mathbf{B_T}, \mathbf{B_S})$ is somewhat equivalent to $\|\mathbf{B_T} - \mathbf{B_S}\|^2$ since if $\mathbf{B_T}$ and $\mathbf{B_S}$ are similar, both $\mathbf{D}(\mathbf{B_T}, \mathbf{B_S})$ and $\|\mathbf{B_T} - \mathbf{B_S}\|^2$ will decrease. But for computational convenience, we adopt the definition in Eq. Equation 5.2. The distortion function $\ell(*,*)$ serves as the key component. Two different definitions of $\ell(*,*)$ with two solutions are presented in the next section. According to Definition 2, on one hand, the projected data should preserve the "structures" of the original data regulated by $\ell\big(\mathbf{B_T}, \mathbf{T}\big)$ and $\ell\big(\mathbf{B_S}, \mathbf{S}\big)$; on the other hand, the projected source and target data are constrained to be similar by minimizing the difference function $\mathbf{D}\big(\mathbf{B_T}, \mathbf{B_S}\big)$. This allows the projection to maximize

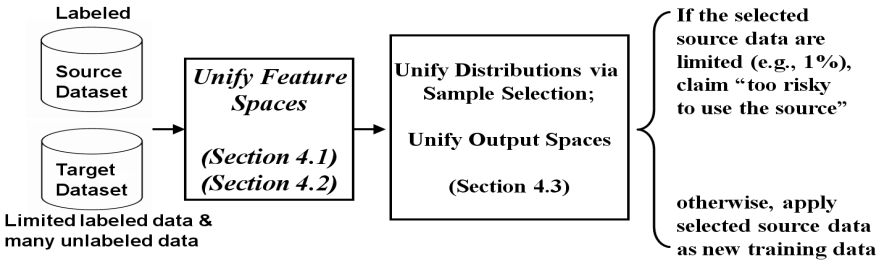


Figure 18. Training data extraction from heterogeneous sources. The focus of the paper is the first step to unify heterogeneous feature spaces.

the similarity among the data, without overly distorting their original structures. Thus, if the source and target data are totally unrelated, their projected data may be still different in distribution in order to preserve their original structures ($\ell\big(\mathbf{B_T}, \mathbf{T}\big)$ and $\ell\big(\mathbf{B_S}, \mathbf{S}\big)$). Finally, a sample selection algorithm is applied to avoid the unrelated examples, and a Bayesian-based approach is proposed to unify the different output spaces. The flow is depicted in Figure 18. Note that if the source task is too different from the target task, the framework will not use any source data but claim "too risky" in the last step. This step is also discussed in Section 5.5.2. It is important to mention again that the focus of this paper is on the first step to unify the different feature spaces by solving Eq. (Equation 5.1). In the next section, two approaches are proposed which are different in how to define the distortion function $\ell(*, *)$.

5.4 Spectral Cross Feature-space Embedding

For the sake of computation, the target data set and the source data set are preprocessed to have the same number of instances. The objective is to make the projected data matrices $\mathbf{B_T}$ and $\mathbf{B_S}$ be of the same size, such that the difference between the two matrices can be conveniently expressed in a matrix form. Note that this preprocessing can be easily done by random sampling to increase the size of

the smaller data set. The data with high frequency are more likely to be sampled, which preserves the original data distribution. In the following, we present the proposed algorithm HeMap (**He**terogeneous **Map**ping) with linear and nonlinear transformation respectively.

5.4.1 Heterogeneous Mapping via Linear Transformation

Linear transformation is adopted to find the projected space since it can be easily expressed in a matrix form, and it has an intuitive meaning in many applications. Denote $\mathbf{P_T} \in \mathbb{R}^{k\times m}$ and $\mathbf{P_S} \in \mathbb{R}^{k\times n}$ as the linear mapping matrices to the target and source data respectively. One straightforward approach to define $\ell(*,*)$ in Eq. (Equation 5.1) is thus:

$$\ell(\mathbf{B_T}, \mathbf{T}) = \|\mathbf{B_T}\mathbf{P_T} - \mathbf{T}\|^2, \ell(\mathbf{B_S}, \mathbf{S}) = \|\mathbf{B_S}\mathbf{P_S} - \mathbf{S}\|^2 \tag{5.3}$$

where $\|\mathbf{X}\|^2 = \sum_{ij} \mathbf{X}_{ij}^2$ is the Frobenius norm which can also be expressed as matrix trace norm. Note that in the above definition, $\ell(\mathbf{B_T}, \mathbf{T})$ is the difference between $\mathbf{B_T}$ and $\mathbf{T}$ in the space expanded by $\mathbf{T}$. An alternative definition is $\ell(\mathbf{B_T}, \mathbf{T}) = \|\mathbf{T}\mathbf{P}_\mathbf{T}^\top - \mathbf{B_T}\|^2$; that is, to evaluate their difference in the space expanded by $\mathbf{B_T}$. However, the latter definition will always lead to a trivial solution $\mathbf{B_T} = \mathbf{0}$ and $\mathbf{P_T} = \mathbf{0}$, because $\mathbf{T}\mathbf{P}_\mathbf{T}^\top = \mathbf{B_T} = \mathbf{0}$ can always minimize the optimization objective. We thus apply the first definition.

Optimization Objective 1: With Eq. (Equation 5.1) and Eq. (Equation 5.3), the optimization objective can then be rewritten as:

$$\begin{aligned} & \min_{\mathbf{B_T^\top B_T}=\mathbf{I},\, \mathbf{B_S^\top B_S}=\mathbf{I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) \\ = & \min_{\mathbf{B_T^\top B_T}=\mathbf{I},\, \mathbf{B_S^\top B_S}=\mathbf{I}} \|\mathbf{T} - \mathbf{B_T P_T}\|^2 + \|\mathbf{S} - \mathbf{B_S P_S}\|^2 \\ & + \beta \times (\frac{1}{2} \cdot \|\mathbf{T} - \mathbf{B_S P_T}\|^2 + \frac{1}{2} \cdot \|\mathbf{S} - \mathbf{B_T P_S}\|^2) \end{aligned} \tag{5.4}$$

where $\mathbf{B_T}$ is assumed orthogonal to reduce redundancy as PCA, similarly for $\mathbf{B_S}$, and β is a nuisance parameter to control how desirable the two data sets are similar. It is interesting to note that in the above formula, the linear projection will automatically perform rotation, scaling and permutation on the target matrix to minimize the difference. In this way, the order of instances, or the order of row vectors in $\mathbf{T}$ and $\mathbf{S}$ will not affect the result.

Lemma 2 *The optimal* $\mathbf{B_T}$, $\mathbf{P_T}$, $\mathbf{B_S}$, $\mathbf{P_S}$ *to Eq. (Equation 5.4) have the following properties:*

$$\begin{aligned} \mathbf{P_T} &= \frac{1}{2+\beta}(2 \cdot \mathbf{B_T^\top T} + \beta \cdot \mathbf{B_S^\top T}) \\ \mathbf{P_S} &= \frac{1}{2+\beta}(2 \cdot \mathbf{B_S^\top S} + \beta \cdot \mathbf{B_T^\top S}) \end{aligned} \tag{5.5}$$

Proof 5 *Note that*

$$\|\mathbf{T} - \mathbf{B_T P_T}\|^2 = \mathbf{tr}(\mathbf{T}^\top \mathbf{T}) - 2\mathbf{tr}(\mathbf{B_T^\top T P_T^\top}) + \mathbf{tr}(\mathbf{P_T^\top P_T})$$

In the above deduction, we use $\mathbf{B_T^\top B_T = I}$ *and the cyclic permutation property of trace:* $\mathbf{tr}(\mathbf{P_T^\top B_T^\top T}) = \mathbf{tr}(\mathbf{B_T^\top T P_T^\top})$. *Let* $\|\mathbf{S - B_S P_S}\|^2$, $\|\mathbf{T - B_S P_T}\|^2$ *and* $\|\mathbf{S - B_T P_S}\|^2$ *be expanded in the same way. The optimization objective in Eq. (Equation 5.4) can then be written as:*

$$
\begin{aligned}
&\min_{\mathbf{B_T^\top B_T = I},\, \mathbf{B_S^\top B_S = I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) \\
=&\min_{\mathbf{B_T^\top B_T = I},\, \mathbf{B_S^\top B_S = I}} (1 + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{T}^\top \mathbf{T}) + (1 + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{S}^\top \mathbf{S}) \\
&+ (1 + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{P_T^\top P_T}) + (1 + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{P_S^\top P_S}) \\
&- 2 \cdot \mathbf{tr}(\mathbf{B_T^\top T P_T^\top}) - 2 \cdot \mathbf{tr}(\mathbf{B_S^\top S P_S^\top}) \\
&- \beta \cdot \mathbf{tr}(\mathbf{B_S^\top T P_T^\top}) - \beta \cdot \mathbf{tr}(\mathbf{B_T^\top S P_S^\top})
\end{aligned}
$$

Taking the derivative of G *w.r.t.* $\mathbf{P_T}$ *and* $\mathbf{P_S}$ *respectively, we obtain*

$$
\begin{aligned}
\nabla G(\mathbf{P_T}) &= -2 \cdot \mathbf{B_T^\top T} - \beta \cdot \mathbf{B_S^\top T} + (2 + \beta) \cdot \mathbf{P_T} \\
\nabla G(\mathbf{P_S}) &= -2 \cdot \mathbf{B_S^\top S} - \beta \cdot \mathbf{B_T^\top S} + (2 + \beta) \cdot \mathbf{P_S}
\end{aligned}
$$

According to the Karush-Kuhn-Tucker conditions, we have $\nabla G(\mathbf{P_T}) = 0$ *and* $\nabla G(\mathbf{P_S}) = 0$ *in the optimal solution (39). We then obtain Eq. (Equation 5.5).*

$$
\begin{aligned}
\mathbf{P_T} &= \frac{1}{2 + \beta}(2 \cdot \mathbf{B_T^\top T} + \beta \cdot \mathbf{B_S^\top T}) \\
\mathbf{P_S} &= \frac{1}{2 + \beta}(2 \cdot \mathbf{B_S^\top S} + \beta \cdot \mathbf{B_T^\top S})
\end{aligned}
$$

Lemma 2 provides the formula of the optimal $\mathbf{P_T}$ and $\mathbf{P_S}$ in terms of $\mathbf{B_T}$ and $\mathbf{B_S}$. We can then use Eq. (Equation 5.5) in the optimization objective in Eq. (Equation 5.4) to derive the closed form of the optimal $\mathbf{B_T}$ and $\mathbf{B_S}$ as the following theorem.

Theorem 4 *The minimization problem in Eq. (Equation 5.4) is equivalent to the following maximization problem:*

$$\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G = \max_{\mathbf{B^\top B=I}} \mathbf{tr}(\mathbf{B}^\top \mathbf{A} \mathbf{B}) \tag{5.6}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{B_T} \\ \\ \mathbf{B_S} \end{bmatrix},\ \mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix}. \tag{5.7}$$

and

$$\begin{aligned} &\mathbf{A}_1 = 2\mathbf{T}\mathbf{T}^\top + \frac{\beta^2}{2}\mathbf{S}\mathbf{S}^\top, \mathbf{A}_4 = \frac{\beta^2}{2}\mathbf{T}\mathbf{T}^\top + 2\mathbf{S}\mathbf{S}^\top \\ &\mathbf{A}_2 = \mathbf{A}_3^\top = \beta(\mathbf{S}\mathbf{S}^\top + \mathbf{T}\mathbf{T}^\top) \end{aligned} \tag{5.8}$$

Proof 6 *With Eq. (Equation 5.5) in Lemma 2, the optimization function in Eq. (Equation 5.4) can be written as:*

$$\begin{aligned} &\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) \\ =&\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} (1+\frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{T}^\top \mathbf{T}) + (1+\frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{S}^\top \mathbf{S}) \\ &- \frac{1}{2+\beta}\mathbf{B}^\top \mathbf{A} \mathbf{B} \end{aligned} \tag{5.9}$$

Since $\mathbf{tr}(\mathbf{T}^\top\mathbf{T})$ *and* $\mathbf{tr}(\mathbf{S}^\top\mathbf{S})$ *are constants, the minimization in Eq. (Equation 5.16) is equivalent to:*

$$\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) = \max_{\mathbf{B^\top B=I}} \mathbf{tr}(\mathbf{B}^\top\mathbf{AB})$$

Theorem 5 *The matrix* $\mathbf{A}$ *in Eq. (Equation 5.7) is a symmetric matrix such that* $\mathbf{A}^\top = \mathbf{A}$.

Proof 7 *Note that matrices* $\mathbf{TT}^\top$, $\mathbf{SS}^\top$ *are all symmetric. As a result,* $\mathbf{A}_1$ *and* $\mathbf{A}_4$ *are symmetric such that* $\mathbf{A}_1^\top = \mathbf{A}_1, \mathbf{A}_4^\top = \mathbf{A}_4$. *Then,*

$$\mathbf{A}^\top = \begin{bmatrix} \mathbf{A}_1^\top & \mathbf{A}_3^\top \\ \mathbf{A}_2^\top & \mathbf{A}_4^\top \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_2 \\ \mathbf{A}_3 & \mathbf{A}_4 \end{bmatrix} = \mathbf{A}.$$

Theorem 5 presents that $\mathbf{A}$ is a symmetric matrix. Under the constraint $\mathbf{B}^\top\mathbf{B} = \mathbf{I}$, the maximization function in Eq. (Equation 5.6) has a closed-form solution.

Theorem 6 *(Ky-Fan theorem (23)) Let* $\mathbf{M}$ *be a symmetric matrix with eigenvalues* $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_k$, *and the corresponding eigenvectors* $\mathbf{U} = [\mathbf{u}_1, \cdots, \mathbf{u}_k]$. *Then* $\sum_{i=1}^k \theta_i = \max_{\mathbf{X}^\top\mathbf{X}=\mathbf{I}_k} \mathbf{tr}(\mathbf{X}^\top\mathbf{MX})$. *Moreover, the optimal* $\mathbf{X}$ *is given by* $[\mathbf{u}_1, \cdots, \mathbf{u}_k]\mathbf{Q}$ *where* $\mathbf{Q}$ *is an arbitrary orthogonal matrix.*

According to Theorem 6, the optimal $\mathbf{B}$ in Eq. (Equation 5.6) is given as the top k eigenvectors of the matrix $\mathbf{A}$. The algorithm is described in Algorithm 3. The main computational cost is to obtain the top k eigenvectors of the matrix $\mathbf{A}$. Consider Lanczos as the method used to calculate the eigenvectors. The complexity of the algorithm is thus $O(kNA_{nz})$ where k is the number of eigenvectors desired, and N is the number of Lanczos iteration steps and A_{nz} denotes the number of non-zeros in the matrix $\mathbf{A}$.

Algorithm 3 HeMap with Linear Transformation

Input: Target data $\mathbf{T}$; Source data $\mathbf{S}$; similarity confidence parameter β (default as 1); #dimentions of the new feature space k

Output: Projected target data $\mathbf{B_T}$; Projected source data $\mathbf{B_S}$

30 Construct matrix $\mathbf{A}$ as Eq. (Equation 5.7) and Eq. (Equation 5.8).

31 Calculate the top-k eigenvalues of $\mathbf{A}$, and their corresponding eigenvectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k]$.

32 $\mathbf{B_T}$ is the first half rows of $\mathbf{U}$

$$\mathbf{B_T} = \begin{bmatrix} \mathbf{U}(1,1) & \cdots & \mathbf{U}(1,k) \\ \cdots & & \\ \mathbf{U}(\frac{l}{2},1) & \cdots & \mathbf{U}(\frac{l}{2},k) \end{bmatrix}$$

where l is the number of rows of $\mathbf{U}$. $\mathbf{B_S}$ is the second half rows of $\mathbf{U}$.

33 Return $\mathbf{B_T}$ and $\mathbf{B_S}$.

5.4.2 Nonlinear Transformation via Incorporating Clustering Information

When the instances are strictly linearly transformed from the original space, it might be still too difficult to find the linear space where the source and target instances are similar. In this way, there may be no source examples selected to help learn the target task because of the different distributions. In this section, we relax the strict linear requirement by allowing some deviations of the projection as long as the clustering structure of the data is preserved. Hence, we first perform clustering on the two data sets respectively and obtain the partition matrix $\mathbf{C_T}$ and $\mathbf{C_S}$ where the partition matrix of the target data $[\mathbf{C_T}]_{i,j} = 1$ iff the i-th instance belongs to the j-th cluster, and $\mathbf{C_S}$ is the partition matrix of the source data. In this case, the distortion function is defined as

$$\begin{aligned} \ell(\mathbf{B_T}, \mathbf{T}) &= \theta \cdot \|\mathbf{B_T P_T} - \mathbf{T}\|^2 + (1-\theta) \cdot \|\mathbf{B_T P_{TC}} - \mathbf{C_T}\|^2 \\ \ell(\mathbf{B_S}, \mathbf{S}) &= \theta \cdot \|\mathbf{B_S P_S} - \mathbf{S}\|^2 + (1-\theta) \cdot \|\mathbf{B_S P_{SC}} - \mathbf{C_S}\|^2 \end{aligned} \tag{5.10}$$

Algorithm 4 HeMap with Nonlinear Transformation

Input: Target data $\mathbf{T}$; Source data $\mathbf{S}$; #dimentions of the new feature space k; Nuisance parameters $\beta = 1$ and $\theta = 0.5$

Output: Projected target data $\mathbf{B_T}$; Projected source data $\mathbf{B_S}$

34 Generate the partition matrix $\mathbf{C_T}$ for the target data by a clustering method; generate the partition matrix $\mathbf{C_S}$ for the source data according to the class label where the data with the same label belongs to the same cluster.

35 Construct the matrix $\hat{\mathbf{A}}$ as Eq. (Equation 5.14) and Eq. (Equation 5.15) in Theorem 7.

36 Calculate the top-k eigenvalues of $\hat{\mathbf{A}}$, and their corresponding eigenvectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_k]$.

37 $\mathbf{B_T}$ is the first half rows of $\mathbf{U}$, while $\mathbf{B_S}$ is the second half rows of $\mathbf{U}$.

38 Return $\mathbf{B_T}$ and $\mathbf{B_S}$.

In addition, $\mathbf{P_{TC}}$ is the linear mapping matrix that projects $\mathbf{B_T}$ onto the space expanded by the partition matrix $\mathbf{C_T}$. $\mathbf{P_{SC}}$ is defined similarly. θ serves as the nuisance parameter to control the importance of preserving the cluster structures. The above formulation thus requires the projected matrices $\mathbf{B_T}$ and $\mathbf{B_S}$ to reflect the partition structures of the data; that is, they can be directly mapped to the partition matrices via certain simple linear transformations.

Optimization Objective 2: With Eq. (Equation 5.10) and the same definition of $\mathbf{D}(*,*)$ in Eq. (Equation 5.2), the optimization objective can then be written as:

$$\begin{aligned}
&\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) \\
=&\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} \theta \cdot \|\mathbf{T} - \mathbf{B_T P_T}\|^2 \\
&+(1-\theta) \cdot \|\mathbf{C_T} - \mathbf{B_T P_{TC}}\|^2 \\
&+\theta \cdot \|\mathbf{S} - \mathbf{B_S P_S}\|^2 + (1-\theta) \cdot \|\mathbf{C_S} - \mathbf{B_S P_{SC}}\|^2 \\
&+\frac{\beta}{2} \cdot \|\mathbf{T} - \mathbf{B_S P_T}\|^2 + \frac{\beta}{2} \cdot \|\mathbf{S} - \mathbf{B_T P_S}\|^2
\end{aligned} \tag{5.11}$$

Lemma 3 *The optimal* $\mathbf{B_T}$, $\mathbf{P_T}$, $\mathbf{B_S}$, $\mathbf{P_S}$, $\mathbf{P_{TC}}$, $\mathbf{P_{SC}}$ *to Eq. (Equation 5.11) have the following properties:*

$$\begin{aligned}
\mathbf{P_T} &= \frac{1}{\beta + 2\theta}(2\theta \cdot \mathbf{B_T^\top T} + \beta \cdot \mathbf{B_S^\top T}) \\
\mathbf{P_S} &= \frac{1}{\beta + 2\theta}(2\theta \cdot \mathbf{B_S^\top S} + \beta \cdot \mathbf{B_T^\top S}) \\
\mathbf{P_{TC}} &= \mathbf{B_T^\top C_T},\ \ \mathbf{P_{SC}} = \mathbf{B_S^\top C_S}
\end{aligned} \tag{5.12}$$

Proof 8 *Note that*

$$\|\mathbf{T} - \mathbf{B_T P_T}\|^2 = \mathbf{tr}(\mathbf{T}^\top \mathbf{T}) - 2\mathbf{tr}(\mathbf{B_T^\top T P_T^\top}) + \mathbf{tr}(\mathbf{P_T^\top P_T})$$

In the above deduction, we use $\mathbf{B_T^\top B_T} = \mathbf{I}$ *and the cyclic permutation property of trace:* $\mathbf{tr}(\mathbf{P_T^\top B_T^\top T}) = \mathbf{tr}(\mathbf{B_T^\top T P_T^\top})$. *Let* $\|\mathbf{S} - \mathbf{B_S P_S}\|^2$, $\|\mathbf{T} - \mathbf{B_S P_T}\|^2$, $\|\mathbf{S} - \mathbf{B_T P_S}\|^2$, $\|\mathbf{C_T} - \mathbf{B_T P_{TC}}\|^2$ *and* $\|\mathbf{C_S} -$

$\mathbf{B_S P_{SC}}\|^2$ *be expanded in the same way. The optimization objective in Eq. (Equation 5.4) can then be written as:*

$$
\begin{aligned}
&\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) \\
=&\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} (\theta + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{T}^\top \mathbf{T}) + (\theta + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{S}^\top \mathbf{S}) \\
&+ (\theta + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{P_T^\top P_T}) + (\theta + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{P_S^\top P_S}) \\
&- 2\theta \cdot \mathbf{tr}(\mathbf{B_T^\top T P_T^\top}) - 2\theta \cdot \mathbf{tr}(\mathbf{B_S^\top S P_S^\top}) \\
&- \beta \cdot \mathbf{tr}(\mathbf{B_S^\top T P_T^\top}) - \beta \cdot \mathbf{tr}(\mathbf{B_T^\top S P_S^\top}) \\
&+ (1-\theta) \cdot \mathbf{tr}(\mathbf{C_T^\top C_T}) + (1-\theta) \cdot \mathbf{tr}(\mathbf{C_S^\top C_S}) \\
&+ (1-\theta) \cdot \mathbf{tr}(\mathbf{P_{TC}^\top P_{TC}}) + (1-\theta) \cdot \mathbf{tr}(\mathbf{P_{SC}^\top P_{SC}}) \\
&- (2-2\theta) \cdot \mathbf{tr}(\mathbf{B_T^\top C_T P_{TC}^\top}) - (2-2\theta) \cdot \mathbf{tr}(\mathbf{B_S^\top C_S P_{SC}^\top})
\end{aligned}
$$

Taking the derivative of G *w.r.t.* $\mathbf{P_T}$, $\mathbf{P_S}$, $\mathbf{P_{TC}}$ *and* $\mathbf{P_{SC}}$ *respectively, we obtain*

$$
\begin{aligned}
\nabla G(\mathbf{P_T}) &= -2\theta \cdot \mathbf{B_T^\top T} - \beta \cdot \mathbf{B_S^\top T} + (2\theta + \beta) \cdot \mathbf{P_T} \\
\nabla G(\mathbf{P_S}) &= -2\theta \cdot \mathbf{B_S^\top S} - \beta \cdot \mathbf{B_T^\top S} + (2\theta + \beta) \cdot \mathbf{P_S} \\
\nabla G(\mathbf{P_{TC}}) &= -(2-2\theta) \cdot \mathbf{B_T^\top C_T} + (2-2\theta) \cdot \mathbf{P_{TC}} \\
\nabla G(\mathbf{P_{SC}}) &= -(2-2\theta) \cdot \mathbf{B_S^\top C_S} + (2-2\theta) \cdot \mathbf{P_{SC}}
\end{aligned}
$$

According to the Karush-Kuhn-Tucker conditions, we have $\nabla G(\mathbf{P_T}) = 0$, $\nabla G(\mathbf{P_S}) = 0$, $\nabla G(\mathbf{P_{TC}}) = 0$ *and* $\nabla G(\mathbf{P_{SC}}) = 0$ *in the optimal solution (39). We then obtain Eq. (Equation 5.12).*

When we apply Lemma 3 into the optimization objective in Eq. (Equation 5.11), we can obtain the following theorem.

Theorem 7 *The minimization problem in Eq. (Equation 5.11) is equivalent to the following maximization problem:*

$$\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G = \max_{\mathbf{B^\top B=I}} \mathbf{tr}(\mathbf{B}^\top \hat{\mathbf{A}} \mathbf{B}) \tag{5.13}$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{B_T} \\ \\ \mathbf{B_S} \end{bmatrix}, \hat{\mathbf{A}} = \begin{bmatrix} \hat{\mathbf{A}}_1 & \hat{\mathbf{A}}_2 \\ \\ \hat{\mathbf{A}}_3 & \hat{\mathbf{A}}_4 \end{bmatrix}. \tag{5.14}$$

and

$$\begin{aligned} \hat{\mathbf{A}}_1 &= 2\theta^2 \cdot \mathbf{T}\mathbf{T}^\top + \frac{\beta^2}{2} \cdot \mathbf{S}\mathbf{S}^\top + (1-\theta)(\beta+2\theta) \cdot \mathbf{C_T}\mathbf{C_T^\top} \\ \hat{\mathbf{A}}_2 &= \hat{\mathbf{A}}_3 = \beta\theta(\mathbf{T}\mathbf{T}^\top + \mathbf{S}\mathbf{S}^\top) \\ \hat{\mathbf{A}}_4 &= 2\theta^2 \cdot \mathbf{S}\mathbf{S}^\top + \frac{\beta^2}{2} \cdot \mathbf{T}\mathbf{T}^\top + (1-\theta)(\beta+2\theta) \cdot \mathbf{C_S}\mathbf{C_S^\top} \end{aligned} \tag{5.15}$$

Proof 9 *With Eq. (Equation 5.5) in Lemma 2, the optimization function in Eq. (Equation 5.4) can be written as:*

$$\begin{aligned} &\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) \\ =& \min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} (\theta + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{T}^\top \mathbf{T}) + (\theta + \frac{\beta}{2}) \cdot \mathbf{tr}(\mathbf{S}^\top \mathbf{S}) \\ &- \frac{1}{2\theta+\beta} \mathbf{B}^\top \hat{\mathbf{A}} \mathbf{B} \end{aligned} \tag{5.16}$$

Since $\mathbf{tr}(\mathbf{T}^\top\mathbf{T})$ *and* $\mathbf{tr}(\mathbf{S}^\top\mathbf{S})$ *are constants, the minimization in Eq. (Equation 5.16) is equivalent to:*

$$\min_{\mathbf{B_T^\top B_T=I},\, \mathbf{B_S^\top B_S=I}} G(\mathbf{B_T}, \mathbf{B_S}, \mathbf{P_T}, \mathbf{P_S}) = \max_{\mathbf{B^\top B=I}} \mathbf{tr}(\mathbf{B}^\top \hat{\mathbf{A}} \mathbf{B})$$

Theorem 8 *The matrix* $\hat{\mathbf{A}}$ *as defined in Eq. (Equation 5.14) is a symmetric matrix such that* $\hat{\mathbf{A}}^\top = \hat{\mathbf{A}}$.

The flow of the proof of Theorem 8 is similar to that of Theorem 5. We present Algorithm 2 to solve the optimization objective in Eq. (Equation 5.13). The first step of the algorithm is to generate the partition matrices via any clustering model, or directly from the source class labels. Then the algorithm finds the eigenvectors of the top-k eigenvalues of the matrix $\hat{\mathbf{A}}$, based on which the projected matrices are constructed. Note that the parameters β and θ can be directly set as default values $\beta = 1$ and $\theta = 0.5$. We introduce these two parameters only in case when the user want to incorporate some prior knowledge to further improve the result (e.g., strong belief of the similarity of the two data set). More analysis is discussed in the experiment section.

5.4.3 Generalization

Linear and nonlinear models are proposed to project the target and source data onto the same feature space. In this section, we briefly present how to deal with the problem of different data distributions and different output spaces.

Heterogeneous Distributions First, the clustering-based sample selection approach in (60) is applied to draw a subset of related examples from the source task as the training data. Its idea is to first mix the projected source data $\mathbf{B_S}$ and the projected target data $\mathbf{B_T}$, and perform clustering on the combined data

set, and it then selects the source data that are similar to the target data evaluated by the clustering-based KL divergence (60):

$$\begin{aligned} &\mathbf{KL_c}(\mathbf{B_T}||\mathbf{B_S}) = \frac{2}{|\mathbf{B_T}|}\mathbb{U} + \mathbf{log}\frac{|\mathbf{B_S}|}{|\mathbf{B_T}|} \\ &s.t. \quad \forall c, \ \mathbb{E}_{x\in\mathbf{B_T}, x\in c}[x] = \mathbb{E}_{x\in\mathbf{D}, x\in c}[x] \end{aligned} \tag{5.17}$$

where $|\mathbf{B_T}|$, $|\mathbf{B_S}|$ denote the data size of the whole projected target data set $\mathbf{B_T}$ and the projected source data set $\mathbf{B_S}$ accordingly, $\mathbb{E}_{x\in\mathbf{B_T}, x\in c}[x]$ denotes the centroid of data from $\mathbf{B_T}$ in cluster c, and $\mathbb{U}$ is defined as follows:

$$\mathbb{U} = \sum_{\mathbf{C}} \left(\frac{|\mathbf{B_T} \cap \mathbf{C}|^2}{|\mathbf{C}|} \mathbf{log} \frac{|\mathbf{B_T} \cap \mathbf{C}|}{|\mathbf{B_S} \cap \mathbf{C}|}\right) \tag{5.18}$$

where $\mathbf{C}$ denotes the cluster of the combined data $\mathbf{B_T} \cup \mathbf{B_S}$. The intuition behind the clustering-based sample selection algorithm is to bias the examples in the cluster where target and source data are equally mixed.

Heterogeneous Outputs If either the source or the target dataset has continuous outputs, we apply the method in (60) to unify heterogeneous output spaces. If both of them are categorical outputs, we use the following decision rule:

$$p(y|\mathbf{x}) = \sum_{v} (p(v|\mathbf{x})p(y|v)) \tag{5.19}$$

where $\mathbf{x}$ is the data to be predicted; y is the target label; and v denotes the output from the source task. And it is assumed that y is conditional independent of x given the label v. In Eq. (Equation 5.19),

the posterior probability $p(v|\mathbf{x})$ can be obtained from arbitrary classifier trained from source examples. Then $p(y|v)$ is estimated by

$$p(y|v) = \frac{1}{p(v)}p(y,v) = \frac{1}{\sum_{\mathbf{s}} p(\mathbf{s})}\sum_{\mathbf{s}} p(y|\mathbf{s})p(\mathbf{s}) \tag{5.20}$$

where s denotes the source data with class label v. $p(\mathbf{s})$ can be estimated via the proportion of data $\mathbf{s}$, and $p(y|\mathbf{s})$ can be obtained via arbitrary classifier trained from target data.

5.5 Experiments

The proposed model HeMap (**He**terogeneous **Map**ping) with linear and nonlinear transformations are studied on 20 data sets involving image classification, drug efficacy prediction, etc. All classification tasks are binary classification problem. For nonlinear transformation, k-means is used to ensure the cluster similarity since it is efficient and simple. The objective of this experiments is to study the ability of the methods to find a good embedding to unify the heterogeneous feature spaces. The ability to tackle the problem of different distributions and output spaces is not the focus, but they will also be studied in the experiment.

It is important to emphasize that the advantage of the proposed methods is their ability to extract more training examples for most of the previous learning models such as SVM, logisitic regression, decision tree, KNN, S3VM, etc. But to isolate the effect of learners, in this section, we use k-nearest-neighbor where k=10 as the common classifier, and linear ridge regression as the common regression model. The main focus is to analyze the performance of the common learners with different training data (the baseline is totally trained from homogeneous data but the proposed models use heterogeneous

TABLE VIII. Description of the data sets

Data set	#Instances	#Features
UCI Data Sets		
Australian Credit	690	14
German Credit	1000	20
Concrete Strength	1030	9
Concrete Slump	103	10
Drug Data Sets		
Data set 1 (reg.)	93	32
Data set 2 (reg.)	74	28
Data set 3 (reg.)	65	32
Data set 4 (reg.)	83	28
Data set 5 (class.)	76	32
Data set 6 (class.)	88	28
Data set 7 (class.)	83	32
Data set 8 (class.)	90	28
Image Data Sets		
Cartman & bonsai	223	995
H. Simpson & Cactus	211	499
H. Simpson & Coin	189	995
Superman & CD	220	499
Text & Image Mixed		
Baseball & Keyboard	212	995
Comp & Rec (text)	300	28196
AK47 & Baseball-bat	195	995
Rec & Talk (text)	300	24043

data). The performance on different learners is further studied in the discussion section. Error rate is applied to evaluate the classification results, and RMSE (root mean square error) is used to measure the regression tasks. In the experiments, $k = 1$, $\beta = 1$ and $\theta = 0.5$ and the parameter sensitivity is discussed in the discussion section.

5.5.1 Real World Data Sets

Twenty data sets (Table VIII) involving classification and regression problems are applied to evaluate the proposed algorithms.

All experiment results are summarized on 10 runs, and in each run we randomly sample certain fraction of target data as the target training data. Note that among the data sets, some are apparently related, such as the drug efficacy prediction data set, while some are only peripherally related such as the text and image mixed data set.

UCI data sets

Four UCI data sets[1] with totally different feature spaces (Table VIII) are applied to evaluate the proposed models HeMap. There are two sets of experiments. The first set is about credit prediction where we have two credit data sets recorded in different countries (i.e., Australia and Germany). The two data sets have different attributes and different output measurements. We set one of the data sets as the target task and the other as the source. Figure 19(a) and Figure 19(b) show the performance of the two versions of HeMap compared with the baseline that does not use the heterogeneous source. Another set of experiment is about regression with two heterogeneous data sets. As shown in Table VIII, one data set is about concrete compressive strength with 1030 instances and 9 features while the other is about concrete slump test with 103 data and 10 features. They are somehow related although they have different meanings of the outputs and different feature spaces. We plot the results on Figure 19(c) and Figure 19(d). From Figure 19, it can be observed that both the linear version and nonlinear version

[1]http://archive.ics.uci.edu/ml/datasets.html

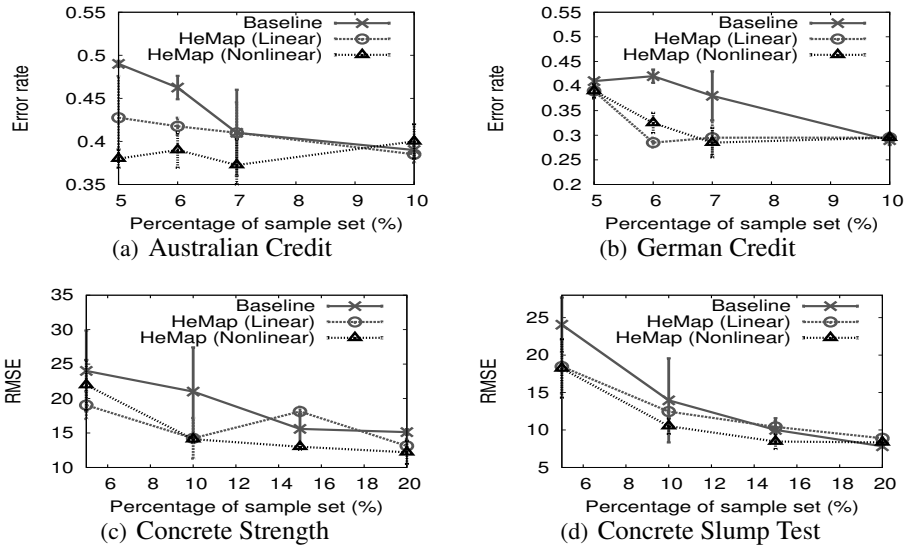


Figure 19. Experiment Results on UCI Data Sets

of HeMap can take advantage of the heterogeneous data to improve the learning accuracy especially when the training data is limited. For instance, in Figure 19(b) when there is only 7% of training data, the error rate of the baseline is about 39% while those of HeMap are less than 30%. Between the two proposed models, the nonlinear model tends to perform better when the amount of target data is small and the source is related because it is more aggressive in picking up source examples. More details are discussed in Section 4.2.

Drug efficacy prediction

The data set is collected by the College of Life Science and Biotechnology of Tongji University. It is to predict the efficacy of drug compounds against certain cell lines. There are four classification tasks (with balanced binary labels) and four regression tasks where the data are generated in two different fea-

ture spaces, i.e., general descriptors and drug-like index. Normally, general descriptors refer to physical prosperities of compounds, while drug-like indexes refer to simple topological indices of compounds. In other words, the general descriptors are the features reflecting whether or not the corresponding drug compound has certain physical property; the drug-like indexes describe whether or not the drug compounds contain certain substructures. Traditionally, because the two feature spaces are heterogeneous, the data using the general descriptors cannot be applied to predict the data expressed as drug-like index, and vice versa, even though they are all for the same application on drug efficacy. In our experiment, we want to analyze the ability of the proposed models to take advantage of these heterogeneous feature spaces, and the target and source data are set to be from these two different feature spaces. The experiment results are plotted in Figure 20, in which one can observe that the proposed models HeMap can reduce the error rate by over 50% especially when there is only a small size of training data. This is because although the target and source dataset have totally different feature spaces and different data distributions, they are all about drug efficacy prediction, and there may be some underlying common prediction rules of different tasks. As such, HeMap explores their similarity in the projected feature space and transfer their hidden similarity to improve the learning accuracy. Furthermore, between the linear version and nonlinear version of HeMap, the nonlinear version works a little better because it has more freedom to maximize the similarity among the source and target datasets. In this situation, more common knowledge is explored and transferred to improve the accuracy. It is also interesting to note that for this drug efficacy prediction data set, the error rates are sometimes larger than 50%, especially when the number of training data is limited. For example, when there is only 15% of training data, the error rate of the baseline is over 55% while that of the proposed nonlinear model is 15% less as shown

in Fig. 20(h). This shows that when there is not enough training data, the baseline is worse than random guessing. Hence, it is necessary to derive the transfer learning model to handle such case.

Image classification

In the field of image classification, there can be various approaches to construct the features. In this set of experiment, we are given 4 different image data sets from the Caltech 256 database[1] (Figure 21). For example, the data set "Cartman & bonsai" is a data set with about 100 images of "Cartman" and 100 images of "bonsai" and the goal is to distinguish the images with the two different contents given certain labeled images. It is important to mention that the 4 image data sets are generated with two different feature spaces. As shown in Table VIII, "Cartman & bonsai" and "Homer simpson & coin" are generated on the basis of the wavelet transformation of the original images and there are 995 features in this space. However, "Homer simpson & Cactus" and "Superman & CD" are constructed according to the histograms of their image colors and there are 499 features in this kind of feature generation. In the experiment, we use the image data set from one feature space to help the image classification task on another set of images which are from a different feature space. For example, the data set "Cartman & bonsai" (995 features) is used to classify the data set "Homer simpson & Cactus" (499 features). Hence, the source and target data sets have (1) different data distributions and different outputs since they have different image contents; (2) different feature spaces. The results are plotted on Figure 22. It can be observed that the proposed model HeMap (linear and nonlinear versions) can reduce the learning error by 20% or more although the source and target data are heterogeneous and only peripherally related

[1]http://www.vision.caltech.edu/Image_Datasets/Caltech256/

as shown in Figure 21. Also note that the baseline performs a little better than the proposed models in some cases when the percentage of labeled data reaches 30% (e.g., Figure 21(c) and Fig 21(d)). This is because when there are sufficient labeled examples (30% in this case), the baseline is good enough to make prediction. However, it is usually not known how many labeled data are needed and if the labeled data is not sufficient (less than 30% in this case), transfer learning can help boost the accuracy.

Text and image data sets with no image tag

In this experiment, we use documents to help classify pure image data with no text annotation. This is in contrast to the case where the image data has text annotation that can be related to the documents. In the first experiment, the target task is to distinguish 212 images of either "Baseball" or "Keyboard", and the source task is a text data set "Comp & Rec" sampled from 20 newsgroup[1]. Similarly, in the second experiment, we have the target task to distinguish "AK47" from "Baseball-bat" and the source task is "Rec & Talk" which is also from 20 newsgroup. On both cases, the source and target data are only tangentially related and they have different data distributions, outputs and feature spaces. Nonetheless, Figure 23 shows that the two proposed models can reduce the error rate by around 16% when there is limited training data.

More interestingly, we plot a subset of examples in the projected feature spaces explored by the non-linear HeMap in Figure 24. It can be observed that in Figure 24(a) and Figure 24(b), the source and target data are similar in distribution in the projected space. Moreover, their original data structures are preserved such that the source data with different class labels are still separable. In these two

[1]http://people.csail.mit.edu/jrennie/20Newsgroups/

TABLE IX. Effect of different sources

Data set	Linear	Nonlinear
Homer & Cactus	5%: **0.33±0.02**	30%: **0.24±0.02**
Homer & Coin	1%: 0.46±0.04	13%: 0.42±0.03
Superman & CD	1%: 0.42±0.01	7%: 0.42±0.01

Note: results are presented as "percentage of examples selected to be training data: error rate". Target task is "Carman vs. bonsai". Baseline error rate is 0.46±0.07

cases, the source data sets help improve the accuracy of the target data sets by transferring the similar decision boundaries. In addition to the two cases, we also plot the found projected space when "AK47 vs. Baseball-bat" is the target data set and the "Comp vs. Rec" is the source data set. In this case, HeMap cannot find a projected space that simultaneously guarantees the two principles: (1) the source and target data are similar; (2) their original data structures are preserved. As shown in Figure 24(c), in order to maintain the original structure of the data sets, the target and source data have different distributions in the projected space. The sample selection algorithm discussed in Section 4.3 will then be executed to exclude those dissimilar source data and thus most of the source examples will not be used in this case. Note that (72; 9) propose models to take advantage of the social tags attached with the images to improve image clustering. Different from these work, in this experiment, we are not given any information about the correlation (e.g., image tags) between the texts and images. The models in (72; 9) are thus not comparable with the proposed methods.

5.5.2 Discussion

In this section, we mainly study the following problems: (1) the comparison between the linear HeMap and nonlinear HeMap when given different sources; (2) the effect of HeMap on various supervised and semi-supervised learning models; (3) parameter sensitivity.

We first study the behavior of HeMap given different sources. Among these sources, some may be strongly related to the target task while some may be weakly related. We use the image data sets as an example. The target task is to distinguish the images between "Carman" and "Bonsai" whose features are generated from the wavelet-based space. Three data sets (Table IX) are used as sources whose features are constructed from the image color histogram space. Among the three sources, "Homer & Cactus" is detected as the most related task since nearly 30% of samples are selected by the nonlinear model. Moreover, it achieves the most significant improvement compared with the baseline method (error rate: 0.46 ± 0.07). This may be because "Homer" is vaguely similar to "Carman", while "Cactus" is somewhat similar to "Bonsai" as shown in Figure 21. On the contrary, only about 10% of samples are selected by the nonlinear model in the other two sources. The reason is that "Superman", "Homer" are still vaguely similar to "Carman" since they are cartoon characters; but "Coin" and "CD" are some unrelated samples to "Bonsai". This shows that the proposed models can judiciously distinguish which task is more related and can draw more examples from it. This is also the reason of the last step of Figure 18 to avoid unrelated data when there is only a few source examples are selected. Furthermore, it can be observed that the nonlinear model can draw more examples than the linear model since it relaxes the projection constraints and allows some deviations. It then encourages more source examples projected as similar to the target task, and more are selected to train the model. As a result, in the

TABLE X. Effect of different learners (error rate)

% of samples	SVM			S3VM			ST-KNN		
	SVM	SVM-L	SVM-NL	S3VM	S3VM-L	S3VM-NL	ST	ST-L	ST-N
5%	0.51	0.49	**0.46**	0.48	0.47	**0.45**	0.34	0.30	**0.28**
10%	0.57	**0.55**	**0.55**	0.55	0.50	**0.49**	0.33	0.27	**0.23**
20%	0.55	0.52	**0.50**	0.54	0.51	**0.47**	0.29	0.27	**0.19**

Note: Best performance of each learner is highlighted in bold. "SVM-L" is the result of SVM trained from the data given by the linear model, and "SVM-NL" is trained from the data given by the nonlinear model, likewise "S3VM-L", "ST-L", etc.

application where large amount of training data are required, nonlinear transformation is a better choice to draw examples from various related sources.

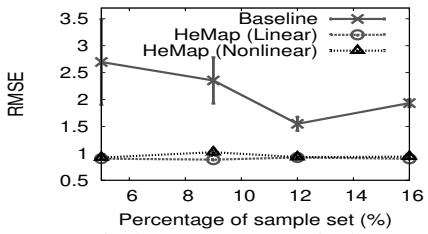
It is important to note that the advantage of the proposed models is their ability to obtain training examples for various learning models to increase their accuracy. We next evaluate the models in different supervised and semi-supervised learning algorithms in Table X. The data set in Figure 20(e) is applied to give the results. Basic learners include SVM, semi-supervised SVM (S3VM (80)), and the self-train model (80) with KNN as the basic classifier. The goal is to study whether the linear version (denoted with suffix "-L") and nonlinear version (denoted with suffix "-NL") can improve the performance of different learners. It can be observed from Table X that both models can consistently improve the learning accuracy of different classifiers and the nonlinear model is better because the source and target data set are related. Another observation is that the KNN model has the best performance. This is because the proposed model uses Frobenius norm (as in Eq. Equation 5.3) to find a feature space with reduced dimensions. Hence, classifiers such as SVM loses its advantage to handle high dimensional data and KNN is beneficial more by the Frobenius norm.

To study the effect of the parameters, the target task is set to be the image data "Carman" and "Bonsai". The source task "Homer & Cactus" is set to be a related task, while the task "Superman & CD" is a partially related task. From the result plotted in Figure 25(a), it can be observed that if the source and target data are relevant, the larger the β, the better is the result. However, if the source and target data are just partially related, a large β may increase the learning error. As a result, without prior knowledge, β is set as 1 to "neutrally" balance the two objectives. Another nuisance parameter θ is used only in the proposed nonlinear model. We set $\theta = 0.5$ to balance the two objectives of nonlinear transformation although it is not very sensitive to the result as shown in Figure 25(b). We also use the data set in Figure 20(e) to study the effect of k that controls the dimensions of the projected data. It can be observed from Figure 25(c) and 25(d) that the two models are not very sensitive to k. In the experiments reported in the last section, $\beta = 1$, $\theta = 0.5$ and $k = 1$, which can be used as the default setting in practice.

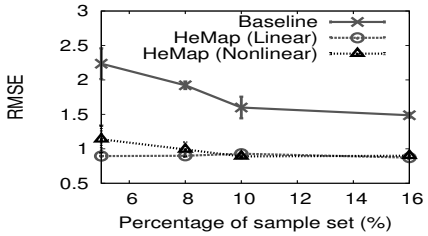
5.6 Conclusions

This paper extends the applicability of supervised learning via spectral embedding, to borrow supervised information from data set with different feature spaces, distributions and output spaces. The main challenge is to find a common projected space for the source and target data sets coming from different feature spaces. This is formulated by two optimization objectives: (1) the original structure of the data is preserved; (2) the projected source and target data are similar in distribution in the new space. Two solutions are proposed to find the optimal embedding with closed forms. The first employs linear transformation to map into a common feature space between source and target, while the second approach makes use of a clustering-based non-linear transformation. Then a sample selection algorithm is incor-

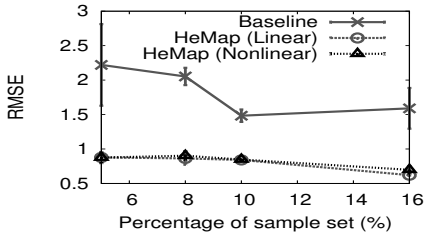
porated to only select those source examples that are most likely to help improve accuracy to model the target domain. Lastly, the differences in output variables between source and target is resolved by a Bayesian-based method that re-scales and calibrates two output variables. Thus, the proposed models can draw training data from heterogeneous related sources. Experiments involve 20 data sets. For example, in the drug efficacy prediction task, the target data describes the physical prosperities of certain compounds; and the source data is related but describes the topological structure of another set of compounds. Although the two tasks are quite different in marginal, conditional distribution and output spaces, HeMap can reduce the error rate by over 50%. For a peripherally related task that aims at using text data to help image classification, the models can still reduce the error rate by about 16%.



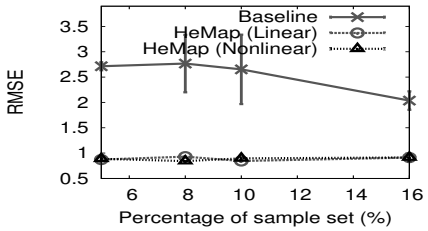
(a) Target is data set 1; source is data set 2



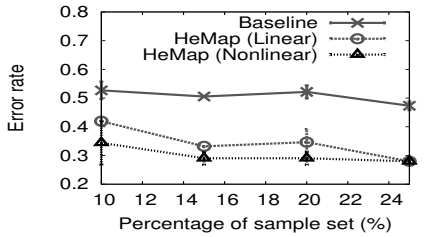
(b) Target is data set 2; source is data set 1



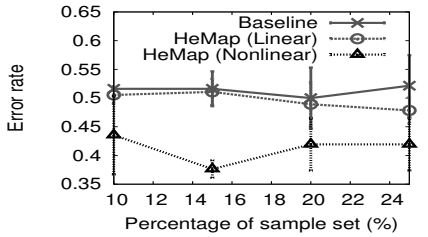
(c) Target is data set 3; source is data set 4



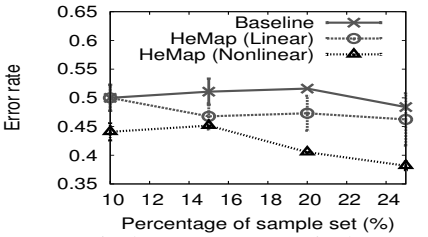
(d) Target is data set 4; source is data set 3



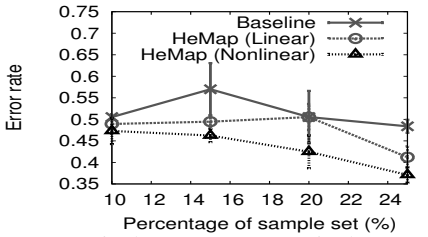
(e) Target is data set 5; source is data set 6



(f) Target is data set 6; source is data set 5



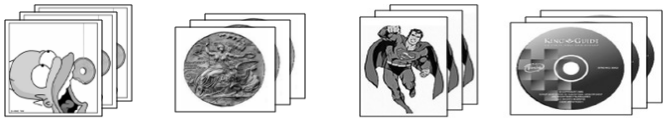
(g) Target is data set 7; source is data set 8



(h) Target is data set 8; source is data set 7

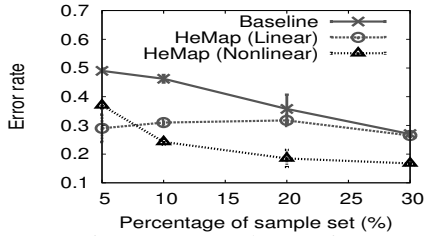
Figure 20. Experiment Results on Drug Efficacy Prediction

(a) Cartman & Bonsai (b) Homer Simpson & Cactus

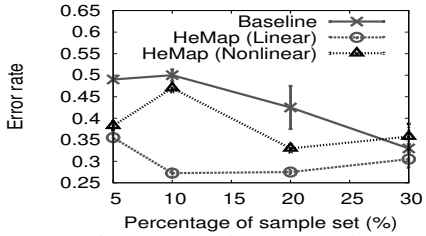


(c) Homer Simpson & Coin (d) Superman & CD

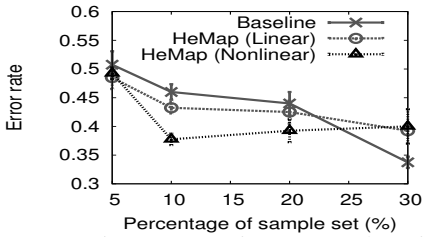
Figure 21. Image Data Sets



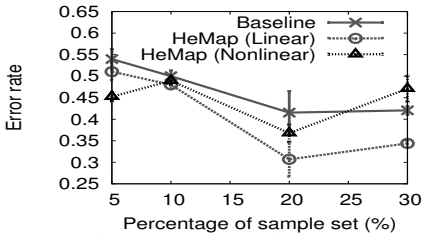
(a) Target is Cartman and Bonsai; source is Homer Simpson and Cactus



(b) Target is Homer Simpson and Cactus; source is Cartman and Bonsai

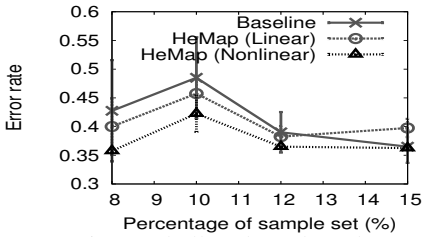


(c) Target is Homer Simpson and Coin; source is Superman and CD

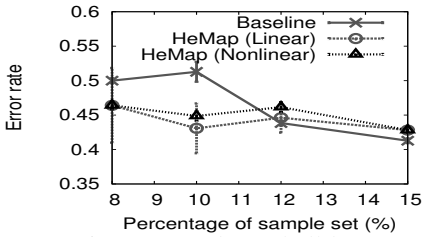


(d) Target is Superman and CD; source is Homer Simpson and Coin

Figure 22. Experiment Results on Image Data Sets

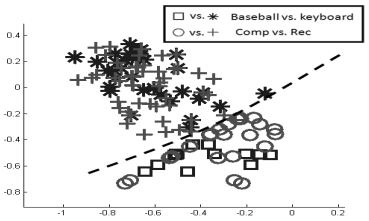


(a) Target is Baseball and Keyboard; source is Comp and Rec

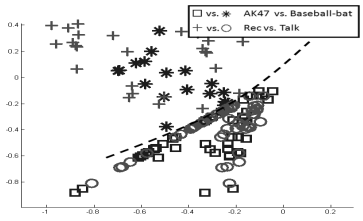


(b) Target is AK47 and Baseball-bat; source is Rec and Talk

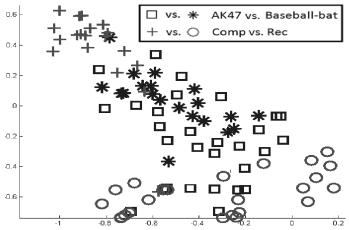
Figure 23. Experiment Results on Text and Image Mixed Data Sets



(a) The source data set (Comp vs. Rec) is forced to keep the original data structure (separable among the data with different labels), while at the same time be similar to the projected target data in distribution. Following the two principles, the optimal projected space is found where the source and target data share the same decision boundary in the projected space.

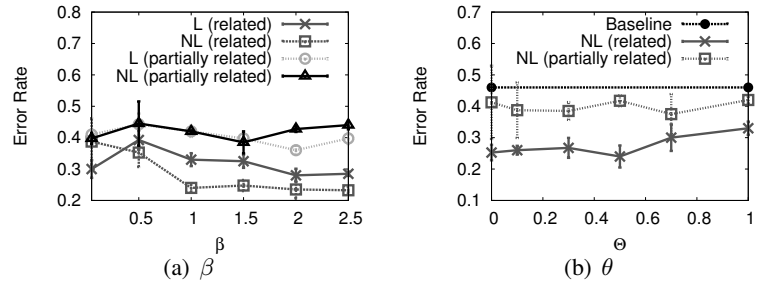


(b) The source data set (Rec vs. Talk) is forced to keep the original data structure and be similar to the target data. They share similar decision boundary in the projected space.



(c) In this case, HeMap cannot find a projected space where the source data set is separable and have similar distribution with the target data. In this situation, the sample selection algorithm described in Section 4.3 will be used to exclude the dissimilar examples from the source task to avoid negative transfer.

Figure 24. In what situations the heterogeneous sources can help the target task?



(a) β (b) θ

(c) k for linear projection (d) k for nonlinear projection

Figure 25. Parameter Sensitivity

CHAPTER 6

CONCLUSIONS

Big data analysis is a hot topic in recent years with the development of advanced database and parallel computing techniques. Given that we have an enormous amount of big datasets available for a given task, it is essential to design a general framework to judiciously use these big datasets collectively. For example, computational advertisement tries to aggregate users' desktop behaviors, users' tablet behaviors, users' purchase records, and their social networks to infer what products users may buy in the late future. In computational finance, the quants are trying to collectively use the market order books, the historical price movement, as well as the financial news feed or even the social media to infer the future price movement. In bioinformatics, we face a similar situation where the scientists are trying to make use of all types of available bio-data (including microarray, experimental data, chemical compound data, etc.) to facilitate their research. In this thesis, we call the process of building models by using multiple heterogeneous datasets as "heterogeneous learning". We analyze the field of heterogeneous learning in the supervised and unsupervised scenario respectively, and we further introduce a heterogeneous projection framework.

In the supervised learning scenario, a stochastic gradient boosting model called GBC is proposed to improve the accuracy of the learning models by using multiple heterogeneous datasets. The key is to cast it as an optimization problem, which (1) minimizes the empirical loss, (2) encourages the predictions from different data sources to be similar, and (3) encourages the predictions of connected data to be similar. Four sets of experiments were conducted, including movie rating prediction, number

recognition, terrorist detection, and demographic prediction tasks with over 500,000 samples and 91 data sources. We show that the proposed GBC model substantially reduce prediction error rate by as much as 80%. In the unsupervised learning scenario, we propose a model called CoCA to find the intrinsic structure of heterogeneous data. The model is derived from PCA, but improved with two important principles. First, all the heterogeneous feature spaces have to be consensually projected on the same subspace. Second, if relational features are available, the data with connections are preferred to be similar in the projected space. An optimization problem is derived from the CoCA principle, and the solution is obtained by solving an eigenvalue problem. It can be clearly observed from the experiments that the proposed CoCA model outperforms the comparison algorithm by as much as 50%, and the weighted CoCA beats the comparison model even more (by as much as three times in NMI). Finally, we introduce a feature projection based approach called HeMap in heterogeneous learning. Conceptually, the projection model satisfies the following two constraints: (1) the original structure of the data is preserved; (2) the projected source and target data are similar in distribution in the new space. Linear projection and nonlinear projection are both introduced to solve the above problem, and the solutions are obtained by solving eigenvalue problems. Experiments involve 20 datasets. For example, in the drug efficacy prediction task, the target data describes the physical prosperities of certain compounds; and the source data is related but describes the topological structure of another set of compounds. Although the two tasks are quite different in marginal, conditional distribution and output spaces, HeMap can reduce the error rate by over 50%. For a peripherally related task that aims at using text data to help image classification, the models can still reduce the error rate by about 16%.

The research of heterogeneous learning is more and more important with the rise of big data analysis. Conceptually, we summarize two important principles to attack the problem. The first principle is called "maximize the consensus" that prefers similar data to have similar predicted results, and the second principle is called "maximize the connectivity" that prefers connected data to have similar properties. We derive the GBC, CoCA, and HeMap based on the two principles. Theoretical and experimental results show that it can improve learning accuracy in many applications.

CITED LITERATURE

1. D. Agarwal, B. Chen, and B. Long. Localized factor models for multi-context recommendation. In *KDD*, pages 609–617, 2011.

2. A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

3. M. Balcan and A. Blum. A pac-style model for learning from labeled and unlabeled data. In *COLT*, pages 111–126, 2005.

4. S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira. Analysis of representations for domain adaptation. In *NIPS*, pages 137–144, 2006.

5. D. P. Bertsekas. *Nonlinear Programming (Second ed.)*. Cambridge, MA.: Athena Scientific, 1999.

6. S. Bickel, J. Bogojeska, T. Lengauer, and T. Scheffer. Multi-task learning for hiv therapy screening. In *ICML*, pages 56–63, 2008.

7. A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

8. R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

9. W. Dai, Y. Chen, G.-R. Xue, Q. Yang, and Y. Yu. Translated learning: Transfer learning across different feature spaces. In *NIPS*, pages 353–360, 2008.

10. W. Dai, G.-R. Xue, Q. Yang, and Y. Yu. Co-clustering based classification for out-of-domain documents. In *KDD*, pages 210–219, 2007.

11. M. Dash and H. Liu. Dimensionality reduction. In *Wiley Encyclopedia of Computer Science and Engineering*, 2008.

12. C. C. Dataset. http://www.cs.umd.edu/projects/linqs/projects /lbc/index.html.

13. H. N. Dataset. http://archive.ics.uci.edu/ml/datasets/multiple +features.

14. J. Davis and P. Domingos. Deep transfer via second-order markov logic. In *ICML*, page 28, 2009.

15. H. Eldardiry and J. Neville. Across-model collective ensemble classification. In *AAAI*, 2011.

16. X. Z. Fern and C. Brodley. Cluster ensembles for high dimensional clustering: An empirical study. *Journal of Machine Learning Research.*, 22(8):888–905, 2004.

17. J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

18. B. Gallagher and T. Eliassi-Rad. Leveraging network structure to infer missing values in relational data. Number UCRL-TR-231993.

19. J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *KDD*, pages 283–291, 2008.

20. J. Gao, W. Fan, Y. Sun, and J. Han. Heterogeneous source consensus learning via decision propagation and negotiation. In *KDD*, pages 339–348, 2009.

21. J. Gao, W. Fan, Y. Sun, and J. Han. Heterogeneous source consensus learning via decision propagation and negotiation. In *KDD*, pages 339–347, 2009.

22. L. Getoor. *Advanced Methods for Knowledge Discovery from Complex Data.* Springer.

23. G. Golub and C. V. Loan. *Matrix computation*. The Johns Hopkins University Press Baltimore, 1996.

24. M. Gönen and E. Alpaydin. Multiple kernel learning algorithms. *Journal of Machine Learning Research*, 12:2211–2268, 2011.

25. D. R. Hardoon, S. Szedmák, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

26. X. He and P. Niyogi. Locality preserving projections. In *NIPS*, 2003.

27. H. D. III. Frustratingly easy domain adaptation. *CoRR*, abs/0907.1815, 2009.

28. T. Jebara. Multi-task feature and kernel selection for svms. In *ICML*, 2004.

29. D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 2004.

30. H. Kargupta, W. Huang, K. Sivakumar, B. Park, and S. Wang. Collective principal component analysis from distributed, heterogeneous data. In *PKDD*, pages 452–457, 2000.

31. X. Kong, X. Shi, and P. S. Yu. Multi-label collective classification. In *SDM*, pages 618–629, 2011.

32. Y. Koren. The bellkor solution to the netflix grand prize, 2009.

33. Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

34. J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.

35. P. Laskov. An improved decomposition algorithm for regression support vector machines. In *NIPS*, pages 484–490, 1999.

36. S.-I. Lee, V. Chatalbashev, D. Vickrey, and D. Koller. Learning a meta-level prior for feature relevance from multiple related tasks. In *ICML*, pages 489–496, 2007.

37. T.-Y. Liu. *Learning to Rank for Information Retrieval*. Springer, 2011.

38. B. Long, P. S. Yu, and Z. Zhang. A general model for multiple view unsupervised learning. In *SDM*, pages 822–833, 2008.

39. B. Long, P. S. Yu, and Z. M. Zhang. A general model for multiple view unsupervised learning. In *SDM*, pages 822–833, 2008.

40. Q. Lu and L. Getoor. Link-based classification. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

41. S. A. Macskassy and F. Provost. A simple relational classifier. In *Proceedings of the Multi-Relational Data MiningWorkshop at the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

42. P. Melville, R. J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.

43. J. Neville and D. Jensen. Iterative classification in relational data. In *Proceeding of the Workshop on Statistical Relational Learning of the National Conference on Artificial Intelligence*, 2000.

44. J. Neville and D. Jensen. Collective classification with relational dependency networks. In *Proceedings of the 2nd Multi-Relational Data Mining Workshop, ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2003.

45. K. Nigam and R. Ghani. Analyzing the effectiveness and applicability of co-training. In *CIKM*, pages 86–93, 2000.

46. S. Oba, M. Kawanabe, K. Müller, and S. Ishii. Heterogeneous component analysis. In *NIPS*, 2007.

47. S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.

48. S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.

49. R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: transfer learning from unlabeled data. In *ICML*, pages 759–766, 2007.

50. J. Ren, X. Shi, W. Fan, and P. S. Yu. Type-independent correction of sample selection bias via structural discovery and re-balancing. In *SDM*, pages 565–576, 2008.

51. P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad. Collective classification in network data. *AI Magazine*, 29(3):93–106, 2008.

52. J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, 2000.

53. X. Shi, W. Fan, and J. Ren. Actively transfer domain knowledge. In *ECML/PKDD (2)*, pages 342–357, 2008.

54. X. Shi, W. Fan, Q. Yang, and J. Ren. Relaxed transfer of different classes via spectral partition. In *ECML/PKDD (2)*, pages 366–381, 2009.

55. X. Shi, W. Fan, J. Zhang, and P. S. Yu. Discovering shakers from evolving entities via cascading graph inference. In *KDD*, pages 1001–1009, 2011.

56. X. Shi, X. Kong, and P. S. Yu. Transfer significant subgraphs across graph databases. In *accepted by SDM*, 2012.

57. X. Shi, Y. Li, and P. S. Yu. Collective prediction with latent graphs. In *CIKM*, pages 1127–1136, 2011.

58. X. Shi, Q. Liu, W. Fan, Q. Yang, and P. S. Yu. Predictive modeling with heterogeneous sources. In *SDM*, pages 814–825, 2010.

59. X. Shi, Q. Liu, W. Fan, and P. S. Yu. Transfer across completely different feature spaces via spectral embedding. *accepted by Transactions on Knowledge and Data Engineering (TKDE)*.

60. X. Shi, Q. Liu, W. Fan, P. S. Yu, and R. Zhu. Transfer learning on heterogenous feature spaces via spectral transformation. In *ICDM*, pages 1049–1054, 2010.

61. X. Shi, J.-F. Paiement, D. Grangier, and P. S. Yu. Learning from heterogeneous sources via gradient boosting consensus. In *accepted by SDM*, 2012.

62. X. Shi and P. S. Yu. Dimensionality reduction on heterogeneous feature space. In *ICDM*, 2012.

63. K. Sridharan and S. M. Kakade. An information theoretic framework for multi-view learning. In *COLT*, pages 403–414, 2008.

64. L. Sun, S. Ji, and J. Ye. A least squares formulation for canonical correlation analysis. In *ICML*, pages 1024–1031, 2008.

65. B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the Annual Conference on Uncertainty in Artificial Intelligence*, 2002.

66. B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin. Learning structured prediction models: A large margin approach. In *Proceedings of the International Conference on Machine Learning*, 2005.

67. B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. Link prediction in relational data. In *Neural Information Processing Systems*, 2003.

68. M. van Breukelen and R. Duin. Neural network initialization by combined classifiers. In *ICPR*, pages 16–20, 1998.

69. B. Wang, J. Tang, W. Fan, S. Chen, Z. Yang, and Y. Liu. Heterogeneous cross domain ranking in latent space. In *CIKM*, pages 987–996, 2009.

70. C. Wang and S. Mahadevan. Manifold alignment using procrustes analysis. In *ICML*, pages 1120–1127, 2008.

71. Y. Weiss. *Advanced Mean Field Methods.* MIT Press.

72. Q. Yang, Y. Chen, G.-R. Xue, W. Dai, and Y. Yu. Heterogeneous transfer learning for image clustering via the socialweb. In *ACL/AFNLP*, pages 1–9, 2009.

73. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51:2282–2312, 2004.

74. Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *ICML*, pages 1151–1157, 2007.

75. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schaolkopf. Learning with local and global consistency. In *Neural Information Processing Systems*, 2004.

76. D. Zhou and B. Scholkopf. Learning from labeled and unlabeled data using random walks. In *Proceedings of the DAGM Symposium*, 2004.

77. X. Zhu. Semi-supervised learning with graphs. Technical report, Carnegie Mellon University, 2005.

78. X. Zhu, Z. Ghahramani, and J. Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the International Conference on Machine Learning*, 2003.

79. X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

80. X. Zhu and A. B. Goldberg. *Introduction to Semi-Supervised Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2009.

81. Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G.-R. Xue, Y. Yu, and Q. Yang. Heterogeneous transfer learning for image classification. In *AAAI*, 2011.

VITA

NAME: Xiaoxiao Shi

EDUCATION:

B.E. in Computer Science, Sun Yat-sen University, China, 2007.

M.S. in Computer Science, Sun Yat-sen University, China, 2009.

M.S. in Applied Math, University of Illinois at Chicago, IL, USA, 2012

PUBLICATIONS

2012 **Xiaoxiao Shi**, Wei Fan and Philip S. Yu, "Dynamic Shaker Detection from Evolving Entities", accepted as full paper in SDM'2013.

2012 Qi Liu, Han Zhou, **Xiaoxiao Shi**, Wei Fan, Ruixin Zhu, Philip S. Yu and Zhiwei Cao, "In-silico Target-specific siRNA Design based on Domain Transfer in Heterogeneous Data", accepted by PLOS Computational Biology (**Impact factor: 5.2**).

2012 **Xiaoxiao Shi** and Philip S. Yu, "Dimensionality Reduction on Heterogeneous Feature Space", accepted as full paper in ICDM'2012 (Acceptance Rate 10.7%).

2012 Xian Wu, Wei Fan, Meilun Sheng, Li Zhang, **Xiaoxiao Shi**, Zhong Su, Yong Yu, "A framework to represent and mine knowledge evolution from Wikipedia revisions", WWW 2012: 633-634.

2012 **Xiaoxiao Shi**, Jean-Francois Paiement, David Grangier, and Philip S. Yu, "Learning from Heterogeneous Sources via Gradient Boosting Consensus", to appear in SDM'2012 (regular paper).

2012 Guan Wang, Yuchen Zhao, **Xiaoxiao Shi**, and Philip S. Yu, "Magnent Community Identification on Social Networks", to appear in KDD'2012.

2012 **Xiaoxiao Shi**, Xiangnan Kong, and Philip S. Yu, "Transfer Significant Subgraphs across Graph Databases", to appear in SDM'2012 (regular paper).

2011 **Xiaoxiao Shi**, Qi Liu, Wei Fan, and Philip S. Yu, "Transfer across Completely Different Feature Spaces via Spectral Embedding", accepted by Transactions on Knowledge and Data Engineering (TKDE).

2011 **Xiaoxiao Shi**, Yao Li, and Philip S. Yu, "Collective Classification with Latent Graphs", the 20th ACM Conference on Information and Knowledge Management (CIKM'11), Glasgow, UK, 2011 (acceptance rate: 15%).

2011 **Xiaoxiao Shi**, Wei Fan, Jianping Zhang, and Philip S. Yu, "Discovering Shaker from Evolving Entities via Cascading Graph Inference", Proceedings of the 17th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'11), San Diego CA, 2011 (acceptance rate: 17.5%).

2011 **Xiaoxiao Shi** and Philip S. Yu, "Limitations of Matrix Completion via Trace Norm Minimization", SIGKDD Explorations, 12(2):16-20, 2011.

2011 Xiangnan Kong, **Xiaoxiao Shi**, and Philip S. Yu, "Multi-label Collective Classification", 2011 SIAM International Conference on Data Mining (SDM'11), Mesa AZ, 2011 (acceptance rate: 25.07%).

2010 **Xiaoxiao Shi**, Wei Fan, and Philip S. Yu, "Efficient Semi-supervised Spectral Co-clustering with Constraints", 2010 IEEE International Conference on Data Mining (ICDM'10), 2010 (acceptance rate: 155/797=19.44%).

2010 **Xiaoxiao Shi**, Qi Liu, Wei Fan, Philip S. Yu, and Ruixin Zhu, "Transfer Learning on Heterogenous Feature Spaces via Spectral Transformation", 2010 IEEE International Conference on Data Mining (ICDM'10), 2010 (acceptance rate: 155/797=19.44%).

2010 **Xiaoxiao Shi**, Kevin Chang, Vijay K. Narayanan, Vanja Josifovski and Alex J. Smola, "A Compression Framework for Generating User Profiles", 2010 ACM SIGIR workshop on feature generation and selection for information retrieval, Geneva, Switzerland, July, 2010.

2010 **Xiaoxiao Shi**, Qi Liu, Wei Fan, Qiang Yang and Philip S. Yu, "Predictive Modeling with Heterogeneous Sources", 2010 SIAM International Conference on Data Mining (SDM 2010), Columbus, Ohio (acceptance rate: 82/351=23.36%).

2009 **Xiaoxiao Shi**, Wei Fan, Qiang Yang and Jiangtao Ren, "Relaxed Transfer of Different Classes via Spectral Partition", 2009 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2009) September 7-11, 2009, Bled, Slovenia (acceptance rate: 105/422=24.88%).

2008 **Xiaoxiao Shi**, Wei Fan, and Jiangtao Ren "Actively Transfer Domain Knowledge", 2008 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2008), Antwerp, Belgium (acceptance rate: 98/521=18.81%).

2008 Jiangtao Ren, **Xiaoxiao Shi**, Wei Fan, and Philip S. Yu "Type Independent Correction of Sample Selection Bias via Structural Discovery and Re-balancing", 2008 SIAM International Conference on Data Mining (SDM 2008), Atlanta, GA, Apr 2008 (acceptance rate: 77/282=27.30%).