Multiple Information Sources Cooperative Learning *

Xingquan Zhu
Faculty of Eng. & Info. Technology
University of Technology, Sydney, Australia
xqzhu@it.uts.edu.au

Ruoming Jin
Dept. of Computer Science
Kent State University, Kent, USA
jin@cs.kent.edu

Abstract

Many applications are facing the problem of learning from an objective dataset, whereas information from other auxiliary sources may be beneficial but cannot be integrated into the objective dataset for learning. In this paper, we propose an omni-view learning approach to enable learning from multiple data collections. The theme is to organize heterogeneous data sources into a unified table with global data view. To achieve the omni-view learning goal, we consider that the objective dataset and the auxiliary datasets share some instance-level dependency structures. We then propose a relational *k*-means to cluster instances in each auxiliary dataset, such that clusters can help build new features to capture correlations between the objective and auxiliary datasets. Experimental results demonstrate that omni-view learning can help build models which outperform the ones learned from the objective dataset only. Comparisons with the co-training algorithm further assert that omni-view learning provides an alternative, yet effective, way for semi-supervised learning.

1 Introduction

As data collection sources and channels continuously evolve, learning and correlating information from multiple data sources (*In this paper, multi-source and multiple data source are equivalent terms*) is almost ubiquitous and crosses business, government/enterprise organizations, and many scientific disciplines. Many key bioinformatic research problems, such as gene function annotation, regulatory motif finding, and protein complex identification, have recently been shown to greatly benefit from the techniques which utilize more than one type of data [Fujibuchi and Kato, 2007], this is because many learners can significantly improve their performance if they can integrate information from multiple relevant sources. Consider a major movie data repository which maintains a large number of movie records (IMDb currently maintains more than 750,000 movies and TV episodes worldwide). Movie rental companies (such as Netflix) which maintain and utilize such large data repository will usually have two major tables, namely movie table and user table shown in Figures (1a) and (1b), respectively. The main objective of the movie table is to use movie content such as keyword descriptions to predict a new movie's expected category, whereas the user table is to predict which types of movies the users are interested in or internal categorization of the users for the company's own purposes (*e.g.* finding high-attrition customers who may turn to other companies). Although the prediction of each task can be solved by learning from each individual table respectively, the prediction solely relying on a single table may not, in practice, generate a good result. Information collected from other sources, such as actors, can be provided as auxiliary data, to enhance the learning. The challenging issue is how to integrate such heterogeneous data sources for learning?

Multi-source learning can also help solve traditional machine learning problems, such as semi-supervised learning [Blum and Mitchell, 1998] and transfer learning [Baxter, 1997], by generalizing the problems as multi-source learning tasks. In semi-supervised learning, data are collected as labeled and unlabeled sets, and the main goal is to use unlabeled set to improve the learning. Common approaches, such as co-training [Blum and Mitchell, 1998], are to assign class labels to unlabeled instances and further include them into the training set for learning. Such an "automatic" labeling process may add a significant amount of class noise [Zhu and Wu, 2003] to the training set. Alternatively, because the representations (*i.e.* features) of the labeled and unlabeled sets are the same, we can directly calculate the relationship between each labeled and unlabeled instances, which may, in practice, provide useful information to improve the learning of the labeled set.

Harnessing multi-source data and unleashing their full potential for learning is, however, a very challenging problem. This is mainly because of the following reasons:

- **Heterogeneous data sources:** Multi-source data normally have different data representations and schemas, which make traditional data integration based approaches practically inefficient to aggregate the data for multi-source learning.

*The support of NSFC under Grant No. 60674109 is acknowledged.

- **Multi-Source data representation:** No common platform currently exists to represent multi-source data in a unified view for multi-source cooperative learning.
- **Multi-Source knowledge transferring:** Multi-source data are usually collected from different sources and used for different learning tasks. Enabling multi-source learning will have to face the problem of transferring knowledge from one dataset to another, or vice versa.

In this paper, we propose a multi-source omni-view learning framework. The purpose is to deliver a common platform to leverage information from multiple data sources for learning. In our problem setting, multi-source data are assumed to be heterogeneous with completely different data representations, and no schema mapping or data integration is required for learning. Our problem setting will not only make the solutions widely applicable for real-world multi-source data, but also helps differentiate the proposed work from existing research such as relational mining, which we will address in Section 2.

To achieve the goal, we propose a data representation model which lays the foundation to represent multi-source data in a global multi-source view. Further more, we propose an omni-view multi-source learning framework and a relational *k*-means clustering method to cluster instances by leveraging information from different sources, such that the clusters can help build new features to capture correlations between data collections.

2 Related Work

Existing work in learning from multiple data sources mainly falls into the following three categories: (1) data integration; (2) model integration; and (3) relational learning.

Data Integration: Data integration intends to integrate heterogeneous data collections, such that the learning can be carried out on the integrated data. A crucial step is to map schemas between different data collections [Cohen, 2000]. Instead of relying on schema mapping for data integration, we seek to unify multi-source data through data correlations, so complex schema mapping process can be avoided.

Model Integration: Model integration focus on combining models learnt from each individual sources, under the assumption that schema mapping between data sources is available or a primary key is available to link data sources [Fujibuchi and Kato, 2007; BenDavid *et al.*, 2002]. This has been commonly used for privacy preserving clustering. In comparison, we do not rely on any complex model integration approaches but directly build learners from the data.

Relational Mining: When instances between data collections are relational, a number of works exist for probabilistic relational classification and clustering [Taskar *et al.*, 2001; Deodhar and Ghosh, 2007; Long *et al.*, 2007]. Crammer *et al.* [Crammer *et al.*, 2008] provides a general theory of selectively combining multi-source data, under the assumption that data sources share identical representations. Multi-task, transfer, and semi-supervised learning [Baxter, 1997; Caruana, 1997] represent another body of work where learning can gain benefits from relevant data sources with identical data representation. In comparison, we are dealing with heterogeneous data sources with distinct data representations.

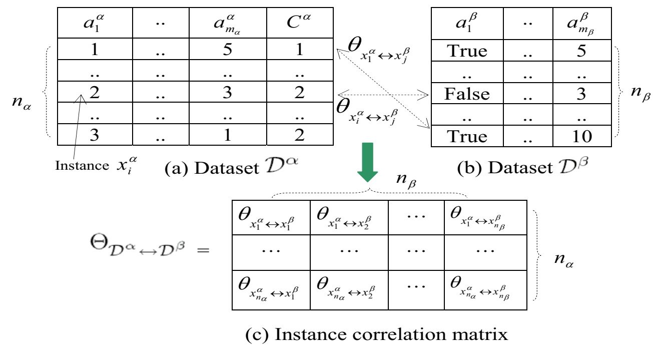

Figure 2: Unified data representation model for multi-source data

3 Multi-Source Data Representation

To support multi-source learning, we propose to utilize a unified data representation model to integrate multi-source data through a matrix which specifies correlations between data sources. Denoting $\mathcal{D}^{\alpha}$ and $\mathcal{D}^{\beta}$ two data collections, each contains a set of instances $x_1^{\alpha}, \cdots, x_{n_\alpha}^{\alpha}$, and $x_1^{\beta}, \cdots, x_{n_\beta}^{\beta}$, where x_i^{α} specifies an instance in $\mathcal{D}^{\alpha}$ and n_α denotes the number of instances in $\mathcal{D}^{\alpha}$. Each instance is represented by a set of features/attributes $a_1^{\alpha}, ..., a_{m_\alpha}^{\alpha}$ where m_α specifies the number of features for instances in $\mathcal{D}^{\alpha}$. If $\mathcal{D}^{\alpha}$ and $\mathcal{D}^{\beta}$ were formed for classification purposes, a class label (C^{α} or C^{β}) will be assigned to each instance.

To bridge connections between different data collections, we assume that each instance in one data collection has a correlation with one or multiple instances in other data collections, as shown in Figure 2. Given two datasets $\mathcal{D}^{\alpha}$ and $\mathcal{D}^{\beta}$, $\theta_{x_i^{\alpha} \leftrightarrow x_j^{\beta}}$ denotes the correlation value between the i^{th} instance in $\mathcal{D}^{\alpha}$ and the j^{th} instance in $\mathcal{D}^{\beta}$. In addition, we also assume that the correlation between i and j is symmetric, *i.e.* $\theta_{\mathcal{D}_i^{\alpha} \leftrightarrow \mathcal{D}_j^{\beta}} = \theta_{\mathcal{D}_j^{\beta} \leftrightarrow \mathcal{D}_i^{\alpha}}$. If we take the correlation for instances without correlation as zero, then all correlation values between instances in $\mathcal{D}^{\alpha}$ and $\mathcal{D}^{\beta}$ form a matrix $\Theta_{\mathcal{D}^{\alpha} \leftrightarrow \mathcal{D}^{\beta}}$, as shown on the right side of Figure 2.

4 Omni-View Cooperative Learning

4.1 Omni-View Dataset Construction

Given a set of data collections $\mathcal{D}^{\alpha}, \mathcal{D}^{\beta}, \cdots, \mathcal{D}^{\gamma}$, each contains a set of instances with or without class labels. For learning purposes, we assume that at least one data collection should be labeled. Since our data representation model intends to map data from one dataset to another, the one to which the data are mapped is called an *objective dataset*, and all others are called *auxiliary datasets*.

Our solution for multi-source learning is to follow the data model in Section 3 and build a special training set for the objective learning task, by leveraging information from all data collections. More specifically, we can list instances in the objective dataset $\mathcal{D}^{\alpha}$ as rows, and map instances from other auxiliary datasets as features(attributes), with each feature value

Movie ID	K ord	..	$K\ ord_k$	Runtime (min)	Production Budget	Year	**Movie roups**
001	2	..	3	178	93 m	2001	**Action**
002	4	..	3	222	N/A	1939	**Comedy**
..	..	..	..	..		..	..
n	1	..	3	127	48 m	1989	**Romance**

(a) Movie table

θ_{22} θ_{n1} θ_{2m}

User ID	Age	Gender	Occupation		**Interests**	**ser roups**
001	32	Male	Engineer		**Action Movie**	**Profitable Customer**
002	44	Male	Artist		**All**	**Professional Reviewer**
..	..	..	..	..		..
m	19	Female	Student		**Comedy Movie**	**High-attrition Customer**

(b) User table

α_{12} α_{22} α_{nk} β_{11} β_{km}

Actor	Year of Birth	Year of Active		Origin	Box Office Grosses (Domestic)	Box Office Grosses (Internal.)	Academy Awards
H. Ford	1942	1966		IL (USA)	3100m	60000m	1
C. Gable	1901	1924		OH(USA)	N/A	N/A	1
..	..	..		..	..	..	..
V. Leigh	1913	1935		British India	N/A	N/A	2

(c) Actor table

Figure 1: A heterogeneous multiple data source example: (a) movie table used to assign movies into different groups; (b) user table used to predict customer interests, loyalty, and the risk of customer churn. θ_{ij} indicates the correlation between a movie i and a user j; (c) actor table listing actors and demographic information α and β specify the movie-actor and user-actor correlations, respectively.

a_1^α	a_2^α	..	$a_{m_\alpha}^\alpha$	C^α
1	2	..	5	1
..	..	..	..	..
..	..	..	..	..
3	7	..	1	2

(a) Objective dataset $\mathcal{D}^\alpha$

a_1^β	..	$a_{m_\beta}^\beta$
True	..	5
..	..	..
..	..	..
True	..	10

(b) Auxiliary dataset $\mathcal{D}^\beta$

a_1^γ	..	$a_{m_\gamma}^\gamma$
A	..	7
..	..	..
..	..	..
A	..	2

(c) Auxiliary dataset $\mathcal{D}^\gamma$

Information from $\mathcal{D}^\alpha$ Information from $\mathcal{D}^\beta$ Information from $\mathcal{D}^\gamma$ Class label for $\mathcal{D}^\alpha$

a_1^α	a_2^α	..	$a_{m_\alpha}^\alpha$	$\mathcal{G}_1^\beta$	..	$\mathcal{G}_{g_\beta}^\beta$	$\mathcal{G}_1^\gamma$	..	$\mathcal{G}_{g_\gamma}^\gamma$	C^α
1	2	..	5	$f(x_1^\alpha,\mathcal{G}_1^\beta)$	..	$f(x_1^\alpha,\mathcal{G}_{g_\beta}^\beta)$	$f(x_1^\alpha,\mathcal{G}_1^\gamma)$	..	$f(x_1^\alpha,\mathcal{G}_{g_\gamma}^\gamma)$	1
..	..	..	..	..	..	..	..	..	..	..
3	7	..	1	$f(x_{n_\alpha}^\alpha,\mathcal{G}_1^\beta)$	..	$f(x_{n_\alpha}^\alpha,\mathcal{G}_{g_\beta}^\beta)$	$f(x_{n_\alpha}^\alpha,\mathcal{G}_1^\gamma)$	..	$f(x_{n_\alpha}^\alpha,\mathcal{G}_{g_\gamma}^\gamma)$	2

(d) Omni-view dataset for $\mathcal{D}^\alpha$

Figure 3: Example of omni-view multi-source learning. $\mathcal{D}^\alpha$ is the objective dataset, $\mathcal{D}^\beta$ and $\mathcal{D}^\gamma$ are auxiliary datasets, $f(x_i^\alpha,\mathcal{G}_j^d)$ can be calculated by Eq.(1)

given by two instances' correlation value $\theta_{x_i^\alpha \leftrightarrow x_j^\beta}$. We call such dataset an **omni-view** dataset, as it contains information across data collections. After that, we can apply any existing learning algorithms to the omni-view dataset and generate prediction models.

4.2 Omni-View Learning

The employment of the omni-view dataset ensures that learning has the ability of surveying data records in the objective dataset, as well as their correlations with other data collections (global knowledge).

Two challenges remain in order to carry out the learning on the omni-view dataset:

- **High Dimensionality**. Because we map an instance from one dataset $\mathcal{D}^\beta$ to another dataset $\mathcal{D}^\alpha$ as a new feature. If $\mathcal{D}^\beta$ has a large number of instances, it will result high-dimensional features in $\mathcal{D}^\alpha$'s omni-view dataset. Learning from high-dimensional data is an identified data mining challenge, therefore we must propose solutions to resolve this issue.
- **Test difficulty**. Although a learning model can be built from the omni-view dataset, the correlation values between a test and the auxiliary instances may not be immediately available. As a result, although prediction model is global, the prediction is still based on the test instance's local feature values.

To resolve the dimensionality challenge, we propose a relational *k*-means clustering method to merge instances in an auxiliary dataset $\mathcal{D}^\beta$ into k groups, $\mathcal{G}_1^\beta,\cdots,\mathcal{G}_k^\beta$, and map each group as a new omni-view feature in the objective dataset $\mathcal{D}^\alpha$, as shown in Figure 3. The value of the new feature is calculated such that for each instance x_i^α in $\mathcal{D}^\alpha$, its new feature value is the average of the mapping values between x_i^α and all instances in group $\mathcal{G}_j^d$, as defined by Eq.(1), where $|\theta_{x_i^\alpha \leftrightarrow \mathcal{G}_j^d}|$ denotes the number of instances in $\mathcal{G}_j^d$ which have mapping relationship values with x_i^α.

$$f(x_i^\alpha,\mathcal{G}_j^d)=\frac{1}{|\theta_{x_i^\alpha \leftrightarrow \mathcal{G}_j^d}|}\sum_{x_\tau^d\in\mathcal{G}_j^d}\theta_{x_i^\alpha \leftrightarrow x_\tau^d} \quad (1)$$

Although the above clustering process can significantly reduce the dimension of the omni-view features, we may still have to face the reality that the cluster number k in the auxiliary dataset $\mathcal{D}^\beta$ may be much larger than the dimension of the objective dataset $\mathcal{D}^\alpha$. In addition, the dimension of the omni-features will cumulate if multiple auxiliary datasets $\mathcal{D}^1,\cdots,\mathcal{D}^d$ are involved. Consequently, we employ the principle component analysis (PCA) process to the omni-view features, and reduce the dimension to be the same as the number of attributes of the objective dataset $\mathcal{D}^\alpha$.

After the above process, the objective dataset $\mathcal{D}^\alpha$ will be transformed into an omni-view dataset. Assuming a prediction model is built from $\mathcal{D}^\alpha$'s transformed omni-view dataset, the test difficulty arises when classifying a test instance x_τ^α. This is because x_τ^α may only contain feature values with respect to $\mathcal{D}^\alpha$, but has no values corresponding to $\mathcal{D}^\alpha$'s omni-view features, as the correlations between x_τ^α and instances in the auxiliary datasets may not be immediately available. Our solution is to use each test instance's nearest neighbours Δ in the objective dataset $\mathcal{D}^\alpha$ to estimate x_τ^α's new feature values, as defined by Eq. (2).

$$f(x_\tau^\alpha,\mathcal{G}_j^d)=\frac{1}{|\Delta|}\sum_{x_\mu^\alpha\in\Delta}f(x_\mu^\alpha,\mathcal{G}_j^d) \quad (2)$$

Based on the above articulations, the major steps of the omni-view learning framework are given in Algorithm 1.

Algorithm 1 OFL: Omni-view Feature Learning

Require: Objective set: $\mathcal{D}^\alpha$, d auxiliary sets $\mathcal{D}^1, \cdots, \mathcal{D}^d$, and mapping relationships $\Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{D}^1}, \cdots, \Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{D}^d}$
  for $\tau \leftarrow 1$ to d **do**
    $\mathcal{G}_1^\tau, \cdots, \mathcal{G}_k^\tau \leftarrow$ relational-k-means($\mathcal{D}^\tau, \Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{D}^\tau}, k, T$)
  end for
  for each training instance x_i^α in $\mathcal{D}^\alpha$ **do**
    $f_i^\alpha \leftarrow$ calculate omni-view features using Eq.(1)
    $F = F \cup f_i^\alpha$ //aggregate omni-view features
  end for
  $F' \leftarrow$ Apply Principle Component Analysis (PCA) to F
  $\mathcal{D}^{'\alpha} \leftarrow$ create omni-view dataset using $\mathcal{D}^\alpha$ and F'
  $\hbar(\mathcal{D}^{'\alpha}) \leftarrow$ train classifier from $\mathcal{D}^{'\alpha}$
  ==================TEST==================
  for each test instance x_τ^α **do**
    $\Delta \leftarrow$ find x_τ^α's K nearest neighbours in $\mathcal{D}^\alpha$
    $f_\tau^\alpha \leftarrow$ calculate omni-view features using Eq.(2)
    $f_\tau^{'\alpha} \leftarrow$ PCA transformed feature
    $x_\tau^\alpha \leftarrow x_\tau^\alpha \cup f_\tau^{'\alpha}$: test instance with omni-view feature
    $\hbar(\mathcal{D}^{'\alpha}, x_\tau^\alpha)$: classify
  end for
  return prediction accuracy

4.3 Relational *k*-Means Clustering

The purpose of the relational k-means clustering is to cluster instances in an auxiliary dataset $\mathcal{D}^\beta$ into a number of k clusters $\mathcal{G}_1^\beta, \cdots, \mathcal{G}_k^\beta$, under matrix $\Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{D}^\beta}$ which specifies the mapping relationships between instances in $\mathcal{D}^\alpha$ and $\mathcal{D}^\beta$. For general k-means, the objective function is to maximize the intra-cluster instance similarities over all clusters [Duda *et al.*, 2001], as defined by Eq. (3), where $S(x, x')$ specifies the similarity between instances x and x', and $S_{\mathcal{G}_i^\beta}$ is the average intra-cluster pair-wise similarity for cluster $\mathcal{G}_i^\beta$.

$$J = max\sum_{i=1}^{k} S_{\mathcal{G}_i^\beta} = max\sum_{i=1}^{k}\{\frac{1}{n^2_{\mathcal{G}_i^\beta}}\sum_{x\in\mathcal{G}_i^\beta}\sum_{x'\in\mathcal{G}_i^\beta} S(x,x')\} \tag{3}$$

If instances in $\mathcal{D}^\beta$ are subject to some mapping relationships $\Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{D}^\beta}$ with the objective dataset $\mathcal{D}^\alpha$, we expect that each cluster $\mathcal{G}_i^\beta$ in $\mathcal{D}^\beta$ will maximize its intra-cluster similarity $S_{\mathcal{G}_i^\beta}$, and the instances in $\mathcal{G}_i^\beta$ should also minimize the mapping relationship variance with respect to $\mathcal{D}^\alpha$. In other words, if two instances x_i^β and x_j^β are indeed similar to each other, they should have large similarity value $S(x_i^\beta, x_j^\beta)$, and share similar mapping relationship values with other instances, *i.e.*, low variance.

The above observation motivates a new clustering measure which combines both intra-cluster similarity and the mapping relationship variance. Given each instance x_i^β in an auxiliary dataset D^β, assume $\Theta_{\mathcal{D}^\alpha \leftrightarrow x_i^\beta} = \in \mathbb{R}^{n_\alpha \times 1}$ denotes the relationship of x_i^β to all (n_α) instances in $\mathcal{D}^\alpha$. Assume the existence of a cluster $\mathcal{G}_\tau^\beta$ in $\mathcal{D}^\beta$, the mean mapping relationship of all instances in $\mathcal{G}_\tau^\beta$ is denoted by Eq. (4).

$$\Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{G}_\tau^\beta} = \frac{1}{|\mathcal{G}_\tau^\beta|}\sum_{x_j^\beta \in \mathcal{G}_\tau^\beta} \Theta_{\mathcal{D}^\alpha \leftrightarrow x_j^\beta} \tag{4}$$

The variance of the mapping relationship of all instances in $\mathcal{G}_\tau^\beta$ (w.r.t. D^α) is given in Eq. (5), where $\Delta_{\Theta(x_j^\beta, \mathcal{G}_\tau^\beta)} = \Theta_{\mathcal{D}^\alpha \leftrightarrow x_j^\beta} - \Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{G}_\tau^\beta}$

$$\delta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{G}_\tau^\beta} = \frac{1}{|\mathcal{G}_k^\beta|}\sum_{x_j^\beta \in \mathcal{G}_\tau^\beta} \Delta^T_{\Theta(x_j^\beta, \mathcal{G}_\tau^\beta)} \Delta_{\Theta(x_j^\beta, \mathcal{G}_\tau^\beta)} \tag{5}$$

As a result, the objective function of the relational k-means is given in Eq.(6)

$$J' = max\sum_{\tau=1}^{k} J_{\mathcal{G}_\tau^\beta} = max\sum_{\tau=1}^{k} \frac{S_{\mathcal{G}_\tau^\beta}}{\delta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{G}_\tau^\beta}} \tag{6}$$

Explicitly solving Eq.(6) is difficult, alternatively, we can employ a recursive hill-climbing search process to find solutions. Assume instances in $\mathcal{D}^\beta$ are clustered into k clusters, $\mathcal{G}_1^\beta, \cdots, \mathcal{G}_k^\beta$, moving an instance x from cluster $\mathcal{G}_i^\beta$ to cluster $\mathcal{G}_j^\beta$ will only change the cluster objective values $J_{\mathcal{G}_i^\beta}$ and $J_{\mathcal{G}_j^\beta}$. Therefore, in order to maximize Eq. (6), at each step t, we can randomly select an instance x from a cluster $\mathcal{G}_i^\beta$, and move x to cluster $\mathcal{G}_j^\beta$. We accept the movement only if the Inequity (7) reaches a larger value at step $t+1$ (*i.e.* after the movement).

$$J_{\mathcal{G}_i^\beta}(t) + J_{\mathcal{G}_j^\beta}(t) < J_{\mathcal{G}_i^\beta}(t+1) + J_{\mathcal{G}_j^\beta}(t+1) \tag{7}$$

Based on the search process in Inequity (7), major steps of the relational k-means are listed in the Algorithm 2.

Algorithm 2 Relational-k-Means

Require: $\mathcal{D}^\beta$, $\Theta_{\mathcal{D}^\alpha \leftrightarrow \mathcal{D}^\beta}$, k and T
  $\mathcal{G}_1^\beta, \cdots, \mathcal{G}_k^\beta \leftarrow$ Apply k-means to $\mathcal{D}^\beta$
  for $t \leftarrow 1$ to T **do**
    $x \leftarrow$ Randomly select an instance from $\mathcal{D}^\beta$
    $\mathcal{G}_i^\beta \leftarrow$ current cluster of instance x
    $J_{\mathcal{G}_i^\beta}(t) \leftarrow$ Calculate $\mathcal{G}_i^\beta$'s objective value
    $J_{\mathcal{G}_i^\beta}(t+1) \leftarrow \mathcal{G}_i^\beta$'s new value after excluding x
    for $j \leftarrow 1$ to k, $j \neq i$ **do**
      $J_{\mathcal{G}_j^\beta}(t) \leftarrow$ Calculate $\mathcal{G}_j^\beta$'s objective value
      $J_{\mathcal{G}_j^\beta}(t+1) \leftarrow \mathcal{G}_j^\beta$'s new value after including x
      if Inequity (7) is $true$ **then**
        $\mathcal{G}_j^\beta \leftarrow \mathcal{G}_j^\beta \cup x$; $\mathcal{G}_i^\beta \leftarrow \mathcal{G}_i^\beta \setminus x$
        $Break$
      end if
    end for
  end for
  return $\mathcal{G}_1^\beta, \cdots, \mathcal{G}_k^\beta$

Accelerated Search Process

In Algorithm 2, each time when we test to move an instance x from cluster $\mathcal{G}_i^\beta$ to $\mathcal{G}_j^\beta$, we need to recalculate the objective values for both $\mathcal{G}_i^\beta$ and $\mathcal{G}_j^\beta$. This is a computational expensive process. Using simple math, we can show that accelerated search process exists for calculating $J_{\mathcal{G}_i^\beta}(t+1)$ and $J_{\mathcal{G}_j^\beta}(t+1)$ from $J_{\mathcal{G}_i^\beta}(t)$ and $J_{\mathcal{G}_j^\beta}(t)$ directly. Due to the page limit, we omit the technical details in this paper.

5 Experimental Results

5.1 Experimental Settings

We implement the omni-view feature learning (OFL) framework in Java by integrating the WEKA machine learning tool [Witten and Frank, 2005]. In our experiments, the number of nearest neighbors of a test instance is automatically determined by using hold-one-out cross validation (provided by WEKA tool), and the number of clusters for each auxiliary dataset is set as the number of instances divided by 20, with 100 clusters being the maximum. Three learning algorithms used in our experiments include C4.5 decision trees [Quinlan, 1993], Naive Bayes, and Support Vector Machines (the Libsvm package). In all experiments, we use 10 times 10-fold cross-validation to evaluate the algorithm performance.

Benchmark Data: For evaluation purposes, we use 10 UCI machine learning benchmark datasets [Asuncion and Newman, 2007] as our test bed. To create heterogeneous data sources, we sort attributes of each dataset in an descending order, according to their information gain values [Quinlan, 1993]. We sequentially assign each sorted attribute (from the top to the bottom) to one data source, which results in multi-source data containing distinct attribute values. We randomly select one data source as the objective dataset, and the remaining data sources are treated as the auxiliary datsets. Although labeling information is available for both objective and auxiliary instances, we only keep class labels for the objective instances, with auxiliary instances remaining unlabeled. We first calculate the Euclidean distance between any instance pairs (in the original complete feature space). Then, we assign the mapping relationship between an objective instance and an auxiliary instance as the normalized Euclidean distance ([0,1]) between their corresponding instances in the original feature space (computed in the first step).

Benchmark Methods: To evaluate the performance of our method, we compare the results with two baseline approaches. The first method is to train classifiers using local features of the objective dataset only. Such a local feature based learning approach is intended to serve as the **lower-bound** of all the benchmark methods. The second method assumes that each instance in the objective dataset is able to build some new features identical to the representation of the auxiliary dataset, so the classifiers are trained from the "global" features of the objective dataset. Notice that "global" features are hard to collect unless the schema mapping and a primary index key exist to link objective and auxiliary datasets. In our experiments, because both objective and auxiliary datasets are split from one dataset, we know objective dataset's "global" features. The results of such a "global" feature based learning provide an **upper-bound** for all the methods. To demonstrate that omni-view learning is effective for traditional learning tasks, such as semi-supervised learning, we also compare OFL with the classical co-training method (denoted by Co-Tr). For comparison purposes, we also report the results of OFL in an ideal situation where the omni-feature values of a test instance are calculated according to Eq. (1) based on a test instance's known mapping relationships with auxiliary instances (denoted by OFL⋄). A simple description of the above methods are listed in Table 2.

Table 1: Description of the benchmark data

Dataset	Sizes	Dimensions	Classes
Anneal	898	38	5
Credit	690	14	2
Digits	3689	256	10
Letter-7	5329	16	7
Kr-vs-kp	3196	36	2
Segment	2310	19	7
Sonar	208	60	2
Soybean	683	35	19
Vowel	990	13	11
Waveform	5000	40	3

Table 2: Abbreviation of the benchmark methods

OFL	classifiers from Omni-view Feature Learning
OFL⋄	OFL with known correlations for test instances
LFL	classifiers from Local Feature Learning
GFL	classifiers from "Global" Features Learning
CoTr	Co-Training classifiers [Blum and Mitchell, 1998]

5.2 Omni-view multi-source learning results

In Table 3, we report OFL, LFL, and GFL's performance on the benchmark datasets where each dataset is equally split into two sources (one object set and one auxiliary set) with non-overlapping features.

The results in Table 3 show that when considering all benchmark data as a whole, omni-view feature learning (OFL) provides an effective solution to leverage multi-source data to improve the learning on the objective dataset. This is valid for all three learning algorithms, which asserts that OFL is effective for common learners. Compared to the classifiers built from the local features (*i.e.* LFL), the classifiers trained from OFL are usually more accurate with $1\% \sim 15\%$ or larger (*i.e.* the Vowel dataset) accuracy gains. If the mapping relationships of a test instance were known (*i.e.*, OFL⋄), OFL can further improve the accuracies to a large extent. On the other hand, the results of GFL show the maximum gains (the upper-bound) we may expect if multi-source data were perfectly aligned (notice that in real-world environments, the results of GFL may not be achievable unless the schema mapping and a primary key both exist to link multi-source data). For some datasets, models trained from partially observed local features may actually outperform the ones learnt from the global features (e.g. the Credit (SVM) dataset). We believe this is mainly because some datasets contain redundant features so using a subset features may help build a better model

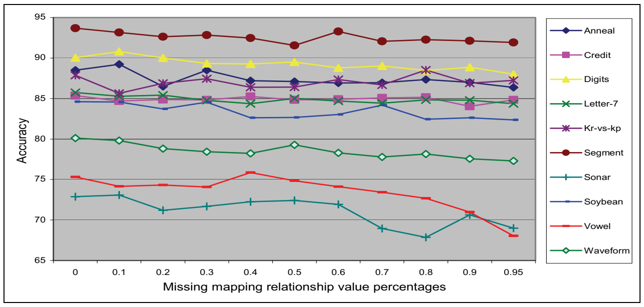

Figure 4: OFL results with different percentages of missing mapping relationship values

(this also explains why feature selection is helpful). Overall, we can expect that OFL significantly outperforms LFL, and occasionally beats the upper-bound GFL (Credit:(C4.5, NB, SVM), Digits:C4.5, Sonar: NB, Waveform: C4.5).

To simulate real-world situations where mapping relationships between some instances are unknown. We randomly select $\epsilon \times 100\%$ of mapping relationships and replace them with a constant value 0.5 (which is the half-way point of the general mapping relationship values [0, 1]). We report the results in Figure 4,which show that although OFL prefers complete mapping relationship values, in practice, it only needs very little mapping information to build learners outperforming LFL. This is mainly because each auxiliary dataset is clustered into a number of groups, so for each instance x in the objective dataset, knowing a small portion of mapping relationships between x and objective instances is sufficient for x to build omni-view features. This observation asserts that although instance mapping relationships between data sources are important, they do not have to be complete to support omni-view learning.

5.3 Omni-view semi-supervised learning results

When using OFL for semi-supervised learning, we split a dataset into two subsets: objective set and auxiliary set, where both sets have the same features as the original data. We discard the labels of the auxiliary instances which results in an unlabeled sample set, and only the objective set contains labeled examples. Then we use co-training and OFL to build prediction models from the objective set, by using auxiliary (unlabeled) instances. For comparison purposes, we also report the results of LFL and GFL. LFL simply trains the models using objective instances only. For GFL, we aggregate both objective and auxiliary instances and assign the genuine class labels to the auxiliary instances and train classifiers from the aggregated data. The results of GFL demonstrate the maximum gains that Co-Tr can possibly achieve.

In Table 4, we report the results of all datasets. In our experiments, we randomly select 20% of instances in each dataset as the labeled set and the remaining instances are taken as the unlabeled set. In addition, we use Co-Tr to assign class labels to 20% of unlabeled instances in 50 repetitions (all parameters are selected to reveal general experimental settings). As shown in Table 4, OFL outperforms Co-Tr from 8 out of 10 datasets. In many situations, the results of Co-Tr are just marginally better than LFL whereas OFL can bring significant improvement, compared to LFL and Co-Tr.

The main disadvantage of the traditional semi-supervised learning methods, such as co-training, is that each unlabeled example is explicitly assigned a class label and included in the training set. Notice that class labels play vital roles for classification, instances with incorrect class labels will act as noise and deteriorate the learner performances [Zhu and Wu, 2003]. To validate our hypothesis, we report the accuracy of the class labels assigned to the unlabeled instances by Co-Tr in the last row of Table 4. As we can see, Co-Tr's average labeling accuracy is between 60% to 90%. This suggests that a significant amount of labeled instances (which are included in the training set) are noisy. When using OFL for semi-supervised learning, although incorrect omni-feature values may also exist, errors in the attributes are actually much less harmful than class errors. In the worst scenario, assume an omni-view feature contains random values, its impact will be easily reduced by OFL's succeeding principle component analysis process.

6 Conclusions

In this paper, we proposed a multiple information sources cooperative learning framework (OFL). Given an objective dataset and its learning tasks, we assume that a number of heterogeneous auxiliary datasets relevant to the objective dataset exist and instances between the objective and the auxiliary datasets may be subject to some mapping relationships. Our solution is to build omni-view features by leveraging information from multiple data collections to achieve multi-source learning goal. Experimental results and comparisons asserted that OFL delivers a simple yet practical platform to support multi-source learning. Such an OFL framework is also beneficial for traditional learning tasks such as semi-supervised learning. The main contribution of the paper, compared to other existing work, is threefold: (1) a common platform to support multi-source learning; (2) a relational k-means for clustering across multiple data collections; and (3) an alternative solution to solve traditional learning problems by using multi-source learning.

References

[Asuncion and Newman, 2007] A. Asuncion and D. Newman. Uci machine learning repository. 2007.

[Baxter, 1997] J. Baxter. Theoretical models of learning to learn. *Learning to Learn, T. Mitchell and S. Thrun*, 1997.

[BenDavid *et al.*, 2002] S. BenDavid, J. Gehrke, and R. Schuller. A theoretical framework for learning from a pool of disparate data sources. In *Proc. of KDD*, 2002.

[Blum and Mitchell, 1998] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. *Proc. of COLT*, 1998.

[Caruana, 1997] R. Caruana. Multitask learning. In *Machine Learning*, pages 28:41–75, 1997.

[Cohen, 2000] W. Cohen. Data integration using similarity joins and a word-based information representation language. In *ACM TOIS*, pages 18(3):288–321, 2000.

Table 3: Multi-source learning accuracy comparisons (OFL◇ denotes the results of OFL with known correlations between the test and the auxiliary instances. A † indicates a t-test significant, compared to LFL, at the 0.05 level)

Learners	Methods	Benchmark Data (Each was equally split to one objective and one auxiliary set)									
		Anneal	Credit	Digits	Letter-7	Kr-vs-kp	Segment	Sonar	Soybean	Vowel	Waveform
C4.5	LFL	83.57	84.92	81.40	82.51	86.25	90.67	67.33	80.93	58.14	73.04
	OFL	88.49†	85.36	90.04†	85.75	87.87	93.69	72.88†	84.61†	75.32†	80.12†
	OFL◇	90.92†	85.43	90.87†	89.52†	96.02†	93.47†	76.67†	90.05†	75.85†	85.60†
	GFL	93.32	85.65	85.98	94.62	99.46	96.91	73.21	91.05	78.35	75.09
NB	LFL	68.67	80.38	84.22	64.61	78.16	75.26	66.14	77.32	40.94	75.77
	OFL	72.86†	85.71†	84.68	70.33†	81.91†	76.89	67.98	80.20†	56.23†	78.94
	OFL◇	73.93†	85.66†	84.67	69.94†	90.26†	77.14	70.02	84.62†	59.49†	81.56†
	GFL	79.73	77.82	84.88	76.34	87.73	80.00	66.67	92.95	63.46	79.92
SVM	LFL	83.14	67.97	92.41	88.22	91.84	50.43	55.85	77.21	53.03	80.29
	OFL	84.82	71.59	92.96	92.24†	92.63	54.75†	58.92	80.87†	60.44†	80.44
	OFL◇	85.01	71.36	92.53	92.88†	94.24†	58.91†	59.54	81.30†	62.32†	81.41
	GFL	85.86	56.45	94.31	98.18	93.73	62.41	63.81	93.65	87.78	86.68

Table 4: Semi-supervised learning accuracy comparisons using C4.5 ("Co-Tr Labeling" is the accuracy of the class labels assigned by Co-Tr. A † indicates a t-test significant, compared to Co-Tr, at the 0.05 level)

Methods	Benchmark Data (Each was equally split to one objective and one auxiliary set)									
	Anneal	Credit	Digits	Letter-7	Kr-vs-kp	Segment	Sonar	Soybean	Vowel	Waveform
LFL	84.30	83.04	76.33	83.58	96.53	92.33	68.14	75.23	53.03	72.09
OFL	89.51†	83.47	84.52†	88.71†	97.49	93.38	67.64	83.99†	57.76†	82.93†
Co-Tr	82.28	83.62	77.81	84.87	95.70	93.37	68.86	78.05	53.53	75.42
GFL	91.85	85.79	86.85	91.76	99.43	96.53	74.57	91.62	81.11	76.16
Co-Tr Labeling	80.29	81.31	84.60	89.69	93.02	93.82	72.90	91.21	62.81	79.95

[Crammer *et al.*, 2008] Koby Crammer, Michael Kearns, and Jennifer Wortman. Learning from multiple sources. *J. of Machine Learning Research*, 9:1757–1774, 2008.

[Deodhar and Ghosh, 2007] M. Deodhar and J. Ghosh. A framework for simultaneous co-clustering and learning from complex data. In *Proc. of KDD*, 2007.

[Duda *et al.*, 2001] R. Duda, P. Hart, and D. Stork. *Pattern Classification*, John Wiley & Sons 2001.

[Fujibuchi and Kato, 2007] W. Fujibuchi and T. Kato. Classification of heterogeneous microarray data by maximum entropy kernel. In *BMC:Bioinformatics*, page 8, 2007.

[Long *et al.*, 2007] B. Long, Z. Zhang, and P. Yu. A probabilistic framework for relational clustering. In *Proceedings of the* 13^{th} *ACM KDD Conference*, 2007.

[Quinlan, 1993] J. Quinlan. *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers 1993.

[Taskar *et al.*, 2001] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proc. of IJCAI*, 2001.

[Witten and Frank, 2005] I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*, Morgan Kaufmann Publishers 2005.

[Zhu and Wu, 2003] Xingquan Zhu and Xindong Wu. Eliminating class noise in large datasets. *Proc. of ICML*, 2003.