# Combining Supervised and Unsupervised Models via Unconstrained Probabilistic Embedding

**Xudong Ma**[1,3], **Ping Luo**[2], **Fuzhen Zhuang**[1,3], **Qing He**[1], **Zhongzhi Shi**[1], **Zhiyong Shen**[2]
[1]The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences. {maxd,zhuangfz,heq,shizz}@ics.ict.ac.cn
[2]Hewlett Packard Labs China. {ping.luo,zhiyongs}@hp.com
[3]Graduate University of Chinese Academy of Sciences

## Abstract

Ensemble learning with output from multiple supervised and unsupervised models aims to improve the classification accuracy of supervised model ensemble by jointly considering the grouping results from unsupervised models. In this paper we cast this ensemble task as an unconstrained probabilistic embedding problem. Specifically, we assume both objects and classes/clusters have latent coordinates without constraints in a $D$-dimensional Euclidean space, and consider the mapping from the embedded space into the space of results from supervised and unsupervised models as a probabilistic generative process. The prediction of an object is then determined by the distances between the object and the classes in the embedded space. A solution of this embedding can be obtained using the quasi-Newton method, resulting in the objects and classes/clusters with high co-occurrence weights being embedded close. We demonstrate the benefits of this unconstrained embedding method by three real applications.

## 1 Introduction

Recently, an interesting problem that combines multiple supervised and unsupervised models at output level has been proposed [Gao *et al.*, 2009]. It is an extension of previous studies on classification ensembles (combining only supervised models) and clustering ensembles (combining only unsupervised models). Here, it combines the output from both supervised models and unsupervised models, and aims to fully utilize the constraints provided by unsupervised clusterings to improve prediction accuracy for classification. The solution to this problem is in great need, where the raw data cannot be accessed due to privacy or other issues and the output from multiple supervised and unsupervised models is provided.

This problem can be formulated as follows. Suppose we have a set of data objects $\mathbf{O} = \{o_1, o_2, \cdots, o_N\}$ from $c$ classes. There are $M$ models that provide the classification information of $\mathbf{O}$, where the first $r$ of them are supervised classifiers, and the left are unsupervised clustering algorithms. Here, we only consider "hard" classification and clustering algorithms, i.e., the object $o$ is predicted to be in exactly one class or cluster. Consider an example where $\mathbf{O} = \{o_1, \cdots, o_6\}$, $c = 2$ and $M = 4$. The output of the four models are: $M_1 = \{1, 2, 1, 1, 2, 2\}$, $M_2 = \{1, 1, 2, 1, 2, 2\}$, $M_3 = \{2, 2, 1, 1, 3, 3\}$, $M_4 = \{2, 2, 2, 2, 1, 1\}$, where $M_1$ and $M_2$ assign each object a class label, whereas $M_3$ and $M_4$ partition the objects into multiple clusters and assign each object a cluster ID.

The objective is to predict the class label of $o_i \in \mathbf{O}$ based on the ensemble of base classification and clustering results. The prediction of this proposed ensemble satisfies as much as possible that 1) it agrees with the base classifiers' prediction, and 2) the objects in the same cluster have the same prediction.

### Object-Group Co-occurrence Matrix

Before briefly introducing our general solution to this problem, we give the concept of *groups*, referring to the classes and clusters from the models, and *object-group co-occurrence matrix*. Note that the taxonomy of data is predefined for classification, and the same class ID assigned by different classification algorithms have the same meaning. However, the clusters from different clustering algorithms

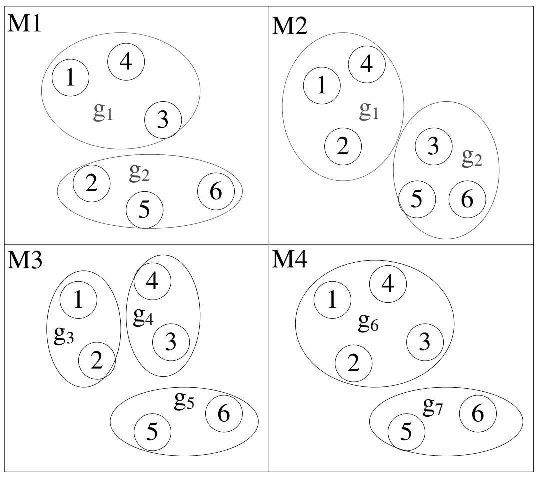

Figure 1: The Multi-Models and Corresponding Groups

have different meaning, and cluster IDs are not related to class labels. Therefore, we can assign a group ID to each class label and each cluster. Assume that clustering algorithm $i$ partitions the data $\mathbf{O}$ into $l_i$ clusters, then the total number of groups is $G = c + \sum_{i=r+1}^{M} l_i$.

Consider our example again. $M_3$ partitions the objects into 3 clusters, while $M_4$ partitions them into 2 clusters. Thus, $G = 7$ in this example. Figure 1 shows the grouping results in the example. As you can see, the groups from the classification algorithms $M_1$ and $M_2$ share the same group IDs, namely $g_1$ and $g_2$. However, the different clusters from the clustering algorithms $M_3$ and $M_4$ use the different group IDs.

Based on the assignment of group IDs, we can generate an *object-group co-occurrence matrix* $\mathcal{M}_{N\times G}$, where $\mathcal{M}_{n,g}$ is equal to the number of times that object $n$ belongs to group $g$. Table 1 shows this co-occurrence matrix for our example. Obviously, when group $g$ is for a class label (say $g_1$ and $g_2$ in the example), $\mathcal{M}_{n,g}$ may be bigger than 1. For example, $\mathcal{M}_{1,1} = 2$ in Table 1 since object 1 belongs to group 1 twice. However, when group $g$ is for a cluster (say $g_3, g_4, g_5, g_6, g_7$ in the example), $\mathcal{M}_{n,g}$ can only be 0 or 1.

| | classification | | clustering 1 | | | clustering 2 | |
|---|---|---|---|---|---|---|---|
| | $g_1$ | $g_2$ | $g_3$ | $g_4$ | $g_5$ | $g_6$ | $g_7$ |
| $o_1$ | 2 | 0 | 1 | 0 | 0 | 1 | 0 |
| $o_2$ | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| $o_3$ | 1 | 1 | 0 | 1 | 0 | 1 | 0 |
| $o_4$ | 2 | 0 | 0 | 1 | 0 | 1 | 0 |
| $o_5$ | 0 | 2 | 0 | 0 | 1 | 0 | 1 |
| $o_6$ | 0 | 2 | 0 | 0 | 1 | 0 | 1 |

Table 1: The Object-Group Co-occurrence Matrix $\mathcal{M}_{6\times 7}$

**Constrained vs. Unconstrained Embedding**

Gao et al. [Gao *et al.*, 2009] proposed a *constrained embedding* solution to this ensemble task. In their method, each object and class/cluster are embedded into a $c$-dimensional cube. Additionally, it is a constrained embedding in the sense that the sum of all the entries in each $c$-dimensional coordinate must be equal to 1. It means that the embedded space is limited as a $c$-dimensional simplex, and the $i$-th entry in the coordinate of an object can be viewed as the probability that this object belongs to the $i$-th class. See our example of binary classification again. The obtained coordinate for each object is a 2-entry probability vector $(v_1, v_2)$, where $v_1 + v_2 = 1$ and $v_1$ (or $v_2$) is the probability that this object belongs to class 1 (or 2).

In this study we solve this ensemble problem by *Unconstrained Probabilistic Embedding* (UPE for short). We assume that each object and group correspond to a latent coordinate without any constraint in a $D$-dimension Euclidean space. With these coordinates the generation of the object-group co-occurrence matrix can be viewed as a probabilistic generative process. Then, by maximizing the posterior of parameters all these coordinates can be obtained, and the prediction of an object is determined by the distances between the object and the groups for the $c$ classes in the embedded space. Different from the constrained embedding method in [Gao *et al.*, 2009], no constraints are imposed into the latent coordinates in our solution. Additionally, the dimension of the embedded space $D$ can be any positive integer while that in the constrained embedding must be $c$, the number of class labels. We argue that with more freedom in fitting the data into any coordinates in any dimension space our solution might achieve better predictions.

The remainder of this paper is organized as follows. Section 2 details our solution of UPE, including the generative process from the coordinates for objects and groups to the object-group co-occurrence matrix, and the derivation of the model parameters by the quasi-Newton method. In Section 3, we demonstrate the effectiveness of UPE by three real applications and compare it with existing alternatives. Section 4 concludes this study.

## 2 Unconstrained Probabilistic Embedding

For clarity and convenience, the notation and denotation used in this study are summarized in Table 2.

| | |
|---|---|
| $\mathcal{M}$ | The object-group co-occurrence matrix |
| $M$ | The number of models |
| $r$ | The number of classification algorithms in the $M$ models |
| $l_i$ | The number of clusters from the $i$-th clustering algorithm |
| $c$ | The number of class labels |
| $G$ | The total number of groups |
| $N$ | The number of objects to be predicted |
| $g$ | The index of groups |
| $n$ | The index of objects |
| $\mathbf{x_n}$ | The coordinate of the $n$-th object in the embedded space |
| $\mathbf{X}$ | The set of all the object coordinates |
| $\phi_\mathbf{g}$ | The coordinate of the $g$-th group in the embedded space |
| $\mathbf{\Phi}$ | The set of all the group coordinates |
| $D$ | The dimensionality of the embedded space |

Table 2: The Notation and Denotation

### 2.1 From Latent Space to Object-Group Co-occurrence Matrix

Now we embed the objects and groups to a new space. We assume the objects have latent coordinates $\mathbf{X} = \{\mathbf{x_n}\}_{n=1}^{N}$, where $\mathbf{x_n} = (x_{n1}, \cdots, x_{nD})$ is a coordinate of the $n$-th object in the embedded space, and $D$ is its dimensionality. Similarly, we assume each group $g$ also has its associated latent coordinate $\phi_\mathbf{g} = \{\phi_{g1}, \cdots, \phi_{gD}\}$ in the embedded space. The probability of an object belonging to a certain group is determined by its Euclidean distances from each group in the embedded space as follows:

$$P(g|\mathbf{x_n}, \mathbf{\Phi}) = \frac{\exp(-\frac{1}{2}\|\mathbf{x_n} - \phi_\mathbf{g}\|^2)}{\sum_{g'=1}^{G}\exp(-\frac{1}{2}\|\mathbf{x_n} - \phi_\mathbf{g'}\|^2)}. \quad (1)$$

where $\sum_{g=1}^{G} P(g|\mathbf{x_n}, \mathbf{\Phi}) = 1$, and $\|\cdot\|$ represents the Euclidean norm in the latent space. Obviously, if the Euclidean distance between $\mathbf{x_n}$ and $\phi_\mathbf{g}$ is small, the probability $P(g|\mathbf{x_n}, \mathbf{\Phi})$ is high.

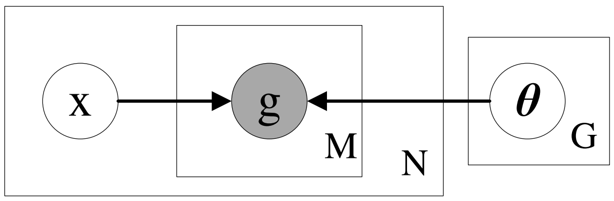

Figure 2: The Graphical Model of UPE

Our model assumes the probabilistic generative process of the object-group co-occurrence matrix $\mathcal{M}$ as shown in Algorithm 1. First, the coordinates $\mathbf{x}_n$ and $\phi_\mathbf{g}$ are assumed to be generated by the zero-mean spherical Gaussian distributions. Then, for an object $n$ we draw the group ID $M$ times using the multinomial distribution $Mult(\{P(g|\mathbf{x_n}, \mathbf{\Phi})\}_{g=1}^{G})$. Each entry in the parameters of this multinomial distribution is calculated by $\mathbf{x}_n$ and $\mathbf{\Phi}$ via Equation (1). If $g'$ is drawn for $n$, we add 1 to $\mathcal{M}_{n,g'}$. This process ends after we sample the groups for all the $N$ objects. Figure 2 shows the graphical model representation of UPE, where the observed and latent variables are indicated by shaded and unshaded nodes, respectively.

**Algorithm 1** Generative Process for the Object-Group Co-occurrence Matrix

```
1. Initialize the object-group co-occurrence matrix :
       $\mathcal{M}_{n,g} = 0$
       where $n = 1, \cdots, N$ and $g = 1, \cdots, G$
2. For each group $g = 1, \cdots, G$:
     Draw group coordinate
       $\phi_\mathbf{g} \sim Normal(\mathbf{0}, \beta^{-1}\mathbf{I})$.
3. For each object $n = 1, \cdots, N$:
     (a) Draw object coordinate
       $\mathbf{x_n} \sim Normal(\mathbf{0}, \gamma^{-1}\mathbf{I})$
     (b) For $m = 1, \cdots, M$:
       i. Draw group
       $g' \sim Mult(\{P(g|\mathbf{x_n}, \mathbf{\Phi})\}_{g=1}^{G})$.
       ii. Increase corresponding weight by 1,
       $\mathcal{M}_{n,g'}++$.
```

Our model is motivated by PLSV (Probabilistic Latent Semantic Visualization) [Iwata *et al.*, 2008], which is a topic modeling based visualization method for documents. In that model the document-word occurrence matrix is taken as input, and the documents and latent topics are embedded into a 2 or 3 dimension space for visualization. To compare PLSV with our model of UPE, we may view *objects* in our problem as *documents*, and *groups* as *words*. Then, the difference between UPE and PLSV is as follows. First, there are not latent topics in UPE. Second, in UPE not only documents but also words are embedded into a $D$ dimension space. Third, since PLSV is used for visualization, the dimension of the embedded space must be 2 or 3, while that for UPE can be any positive integer.

## 2.2 Parameter Estimation

The unknown parameters are a set of object coordinates $\mathbf{X}$ and a set of group coordinates $\mathbf{\Phi}$. We estimate these parameters based on Maximum A Posteriori (MAP) estimation. The posterior of the parameters is given as follows,

$$p(\mathbf{X}, \mathbf{\Phi}|\mathcal{M}) = \frac{p(\mathcal{M}|\mathbf{X}, \mathbf{\Phi})p(\mathbf{X})p(\mathbf{\Phi})}{p(\mathcal{M})}, \tag{2}$$

where

$$P(\mathcal{M}|\mathbf{X}, \mathbf{\Phi}) = \prod_{n=1}^{N}\prod_{g=1}^{G} P(g|\mathbf{x_n}, \mathbf{\Phi})^{\mathcal{M}_{n,g}}, \tag{3}$$

$$p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x_n}), \tag{4}$$

$$p(\mathbf{\Phi}) = \prod_{g=1}^{G} p(\phi_\mathbf{g}) \tag{5}$$

In the generative process described in the previous subsection, we use a Gaussian prior with a zero mean and a spherical covariance for distribution of the coordinates of $\mathbf{x_n}$ and $\phi_\mathbf{g}$. Then, we have

$$p(\phi_\mathbf{g}) = (\frac{\beta}{2\pi})^{\frac{D}{2}} \exp(-\frac{\beta}{2}\|\phi_\mathbf{g}\|^2), \tag{6}$$

$$p(\mathbf{x_n}) = (\frac{\gamma}{2\pi})^{\frac{D}{2}} \exp(-\frac{\gamma}{2}\|\mathbf{x_n}\|^2). \tag{7}$$

where $\beta$ and $\gamma$ are hyper-parameters. We choose $\beta = \lambda N$, $\gamma = \lambda G$, and $\lambda$ $(0 \leq \lambda \leq 1)$ is a coefficient. In the experiment section we will discuss the influence of different values of $\lambda$.

Next, we estimate parameters $\mathbf{X}, \mathbf{\Phi}$ by maximizing the posterior $p(\mathbf{X}, \mathbf{\Phi}|\mathcal{M})$ in Equation (2). By applying the log function, it is equivalent to the following optimization problem,

$$\{\mathbf{X}, \mathbf{\Phi}\} = \max_{\mathbf{X}, \mathbf{\Phi}} Q(\mathbf{X}, \mathbf{\Phi}) \tag{8}$$

where

$$\begin{aligned} Q(\mathbf{X}, \mathbf{\Phi}) = &\sum_{n=1}^{N}\sum_{g=1}^{G} \mathcal{M}_{n,g} \log P(g|\mathbf{x_n}, \mathbf{\Phi}) + \\ &\sum_{n=1}^{N} \log p(\mathbf{x_n}) + \sum_{g=1}^{G} \log p(\phi_\mathbf{g}) \end{aligned} \tag{9}$$

We use the quasi-Newton method [Liu and Nocedal, 1989] to maximize $Q(\mathbf{X}, \mathbf{\Phi})$. The partial differentials of $Q(\mathbf{X}, \mathbf{\Phi})$ w.r.t. $\mathbf{x_n}$ and $\phi_\mathbf{g}$ are as follows respectively,

$$\frac{\partial Q}{\partial \mathbf{x_n}} = \sum_{g=1}^{G} (M \cdot P(g|\mathbf{x_n}, \mathbf{\Phi}) - \mathcal{M}_{n,g})(\mathbf{x_n} - \phi_g) - \gamma \mathbf{x_n} \tag{10}$$

$$\frac{\partial Q}{\partial \phi_\mathbf{g}} = \sum_{n=1}^{N} (M \cdot P(g|\mathbf{x_n}, \mathbf{\Phi}) - \mathcal{M}_{n,g})(\phi_g - \mathbf{x_n}) - \beta \phi_\mathbf{g} \tag{11}$$

By applying the quasi-Newton method until convergence, we obtain a local optimum solution for $\{\mathbf{X}, \mathbf{\Phi}\}$, including the coordinates of objects and groups in the embedded space. Then we can predict the label of object $n$ as follows,

$$g' = \max_{1 \le g \le c} P(g|\mathbf{x_n}, \mathbf{\Phi}) \tag{12}$$

Remember that our task is for classification. Thus, in the prediction process we only consider the first $c$ groups, corresponding to the $c$ class labels, for comparison. This prediction process is equivalent to

$$g' = \min_{1 \le g \le c} \|\mathbf{x_n} - \phi_\mathbf{g}\|^2, \tag{13}$$

where $\| \cdot \|$ represents the Euclidean norm in the latent space.

## 3 Experiments

### 3.1 Data Set

We apply the proposed method on the data set from [Gao *et al.*, 2009], including 11 classification tasks from three real world applications. In each task, the data set is separated into training and test sets. Clustering algorithms are performed on the test set to obtain the grouping results. On the other hand, we learn classification model from training set and apply it to the test set. The proposed algorithm generates a consolidated classification solution for the test set based on both classification and clustering results. The details of each application is elaborated in the following.

**20 Newsgroups categorization.** The 20 newsgroups data set[1] contains approximately 20,000 newsgroup documents, partitioned across 20 different newsgroups nearly evenly. From the data sets, six learning problems are constructed, each of which has documents from four different topics to distinguish. The logistic regression [Genkin *et al.*, 2005] and SVM models [Chang and Lin, 2001] are learned from the training sets, and apply, as well as K-means and min-cut clustering algorithms [Karypis, 2006] on the test sets.

**Cora research paper classification.** The Cora data set [McCallum *et al.*, 2000] contains around 37,000 research papers that are classified into a topic hierarchy with 73 leaves. The citations among papers are around 75,000 entries. Four test sets are extracted, each of which includes papers from around four areas. The training sets contain research papers that are different from those in the test sets. Both training and test sets have two views, the paper abstracts, and the paper citations. The logistic regression classifiers and K-means clustering algorithms are applied on the two views of the target sets.

**DBLP network.** Around 4,000 authors are retrieved from DBLP network [Ley, 2001]. We try to predict their research areas in four candidate fields. There are also two views for both training and test sets, namely the publication network and the textual content of the publications. The amount of papers an author published in the conference can be regarded as link feature, whereas the pool of titles that an author published is the text feature. Logistic regression and K-means clustering algorithms are used to derive the predictions on the test set. The test set is labeled manually for evaluation.

### 3.2 Baseline Methods

On each test set, two classification models and two clustering models (denoted as M1 to M4) are obtained by applying four different algorithms. Then we applied the ensemble methods on the four models. In [Gao *et al.*, 2009] the constrained embedding solution, namely Bipartite Graph-based Consensus Maximization (BGCM for short) is compared with the two clustering ensemble approaches, MCLA [Strehl and Ghosh, 2003] and HBGF [Fern and Brodley, 2004], which regard all the base models as unsupervised clustering, and integrate all their output. They show that BGCM improves the accuracy of the best single model by $2\%$ to $10\%$.

Here, we focus on the comparison between UPE and BGCM, and also give the results of the baselines in [Gao *et al.*, 2009]. To evaluate classification accuracy, The output of all the clustering algorithms (the base models, and the ensembles) are mapped to the best possible class predictions with the help of hungarian method[2] [Kuhn, 1955], where cluster IDs are matched with class labels. In addition, the results of Majority Voting on only the classification models (denoted as MV) are presented. We denote the proposed methods as UPE-$D$, where $D$ refer to the number of dimension used for the embedded space. $D$ is set from 1 to 4 in the experiments. Note that UPE has some random property since the initial parameters are drawn from a spherical Gaussian distribution. Thus, we run our solution 20 times with different initialization for each task, and calculate the average value as the result.

### 3.3 Experimental Results

In Table 3, we present the classification accuracy of UPE-$D$ and the other baselines on eleven tasks, and the highest accuracy of each task is bolded. The graphical comparison of UPE and BGCM is also shown as Figure 3. We can see that UPE-2 is already better than BGCM, which indicates that our unconstrained embedding method in 2-dimensional space is already better than the constrained embedding method in a $c$-dimensional space ($c = 4$ in these tasks). Additionally, UPE-3 further improves the accuracy. Among the 11 tasks UPE-3 achieves 7 best results, and it increases the accuracy by $1.3\%$ to $4.4\%$ in 10 cases compared with BGCM. Also UPE-4 achieves similar performance with UPE-3.

As described before, two hyper-parameters $\beta$ and $\gamma$ are used in the prior distribution of the coordinates. They are

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

[2]N. Borlin, http://www.cs.umu.se/ niclas/matlab/assignprob/

| Methods | 20Newsgroup | | | | | | Cora | | | | DBLP | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 1 | 2 | 3 | 4 | 1 | |
| M1 | 0.7967 | 0.8855 | 0.8557 | 0.8826 | 0.8765 | 0.888 | 0.7745 | 0.8858 | 0.8671 | 0.8841 | 0.9337 | 0.8664 |
| M2 | 0.7721 | 0.8611 | 0.8134 | 0.8676 | 0.8358 | 0.8563 | 0.7797 | 0.8594 | 0.8508 | 0.8879 | 0.8766 | 0.8419 |
| M3 | 0.8056 | 0.8796 | 0.8658 | 0.8983 | 0.8716 | 0.902 | 0.7779 | 0.8833 | 0.8646 | 0.8813 | 0.9382 | 0.8698 |
| M4 | 0.777 | 0.8571 | 0.8149 | 0.8467 | 0.8543 | 0.8578 | 0.7476 | 0.8594 | 0.781 | 0.9016 | 0.7949 | 0.8266 |
| MV | 0.7819 | 0.8722 | 0.8239 | 0.8639 | 0.8567 | 0.8631 | 0.8643 | 0.9079 | 0.8161 | 0.88 | 0.9105 | 0.8582 |
| MCLA | 0.7592 | 0.8173 | 0.8253 | 0.8686 | 0.8295 | 0.8546 | 0.8703 | 0.8388 | 0.8892 | 0.8716 | 0.8953 | 0.8472 |
| HBGF | 0.8199 | 0.9244 | 0.8811 | 0.9152 | 0.8991 | 0.9125 | 0.7834 | 0.9111 | 0.8481 | 0.8943 | 0.9357 | 0.8841 |
| BGCM | 0.8128 | 0.9101 | 0.8608 | 0.9125 | 0.8864 | 0.9088 | 0.8687 | **0.9155** | 0.8965 | 0.909 | 0.9417 | 0.8930 |
| UPE-1 | 0.8074 | 0.8948 | 0.8563 | 0.8969 | 0.8851 | 0.8995 | 0.8493 | 0.8833 | 0.9074 | 0.8876 | 0.9032 | 0.8791 |
| UPE-2 | 0.8438 | 0.9383 | **0.9002** | 0.9333 | 0.9211 | 0.9200 | 0.8793 | 0.9038 | **0.9196** | 0.9138 | 0.9530 | 0.9115 |
| UPE-3 | **0.8444** | **0.9389** | 0.8983 | **0.9423** | **0.9211** | **0.9259** | **0.8839** | 0.9069 | 0.9174 | 0.9128 | **0.9541** | **0.9133** |
| UPE-4 | 0.8431 | **0.9389** | 0.8970 | 0.9414 | **0.9211** | **0.9259** | 0.8748 | 0.9069 | 0.9101 | **0.9190** | 0.9492 | 0.9116 |

Table 3: Classification Accuracy Comparison

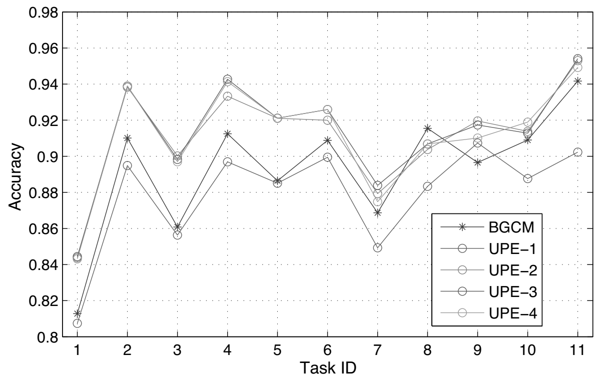

Figure 3: The accuracy of BGCM and UPE-$D$.

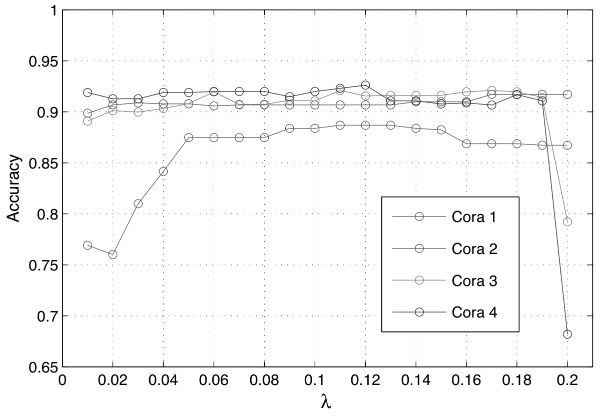

Figure 4: The accuracy on Cora data set with different $\lambda$ value.

set as $\beta = \lambda N$ and $\gamma = \lambda G$. In the previous experiments, we fix $\lambda = 0.1$ for UPE. Here, on the date set of Cora we check the sensitivity of this parameter by changing it from 0.01 to 0.20 with 0.01 as interval. As shown in Figure 4, when $0.05 \leq \lambda \leq 0.19$, the performance is not sensitive.

We also check whether the random initialization of the coordinates affects the performance of UPE. Since we run 20 times with different initialized values for each task, we can calculate the standard deviation of the 20 values. The results show that when $\lambda \leq 0.2$ the standard deviation is less than 0.01. However, when $\lambda > 0.2$ the standard deviation increases, and exceeds 0.15 when $\lambda > 0.4$. Therefore, $\lambda > 0.2$ UPE is not stable in terms of the random initialization. Fortunately, it is very stable when $\lambda \leq 0.2$.

## 4 Conclusions

In this paper we propose an unconstrained probabilistic embedding solution to combine output from multiple supervised and unsupervised models. In our solution each object and group are embedded into a $D$ dimensional space without any constraints. We argue that the more freedom in the embedding process of our solution will result in performance improvement. By considering the mapping from the coordinates in the embedded space into the object-group co-occurrence matrix as a generative process, we can obtain these coordinates by the MAP estimation. This optimization problem is then solved by the quasi-Newton method. With the 11 tasks from three real applications we show the superiority of our solution over the constrained embedding method.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 60933004, 60975039, 61035003, 60903141, 61072085), and National Basic Research Priorities Programme (No.2007CB311004). Thanks to the internship program of Hewlett Packard Labs China.

## References

[Chang and Lin, 2001] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines. 2001.

[Fern and Brodley, 2004] X.Z. Fern and C.E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the twenty-first international conference on Machine learning*, page 36. ACM, 2004.

[Gao *et al.*, 2009] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han. Graph-based consensus maximization among multiple supervised and unsupervised models. *Advances in Neural Information Processing Systems (NIPS)*, 22:585–593, 2009.

[Genkin *et al.*, 2005] A. Genkin, D.D. Lewis, and D. Madigan. Bbr: Bayesian logistic regression software. *Center for Discrete Mathematics and Theoretical Computer Science, Rutgers, New Jersey, USA.[online] URL: http://www. stat. rutgers. edu/˜ madigan/BBR*, 2005.

[Iwata *et al.*, 2008] T. Iwata, T. Yamada, and N. Ueda. Probabilistic latent semantic visualization: topic model for visualizing documents. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 363–371. ACM, 2008.

[Karypis, 2006] G. Karypis. Cluto-family of data clustering software tools, 2006.

[Kuhn, 1955] H.W. Kuhn. The Hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[Ley, 2001] M. Ley. DBLP bibliography. *Available at http://www.informatik.uni-trier.de/ ley/db/*, 2001.

[Liu and Nocedal, 1989] D.C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1):503–528, 1989.

[McCallum *et al.*, 2000] A.K. McCallum, K. Nigam, J. Rennie, and K. Seymore. Automating the construction of internet portals with machine learning. *Information Retrieval*, 3(2):127–163, 2000.

[Strehl and Ghosh, 2003] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.