Unsupervised Cluster Matching via Probabilistic Latent Variable Models

Tomoharu Iwata and **Tsutomu Hirao** and **Naonori Ueda**
NTT Communication Science Laboratories
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan

Abstract

We propose a probabilistic latent variable model for unsupervised cluster matching, which is the task of finding correspondences between clusters of objects in different domains. Existing object matching methods find one-to-one matching. The proposed model finds many-to-many matching, and can handle multiple domains with different numbers of objects. The proposed model assumes that there are an infinite number of latent vectors that are shared by all domains, and that each object is generated using one of the latent vectors and a domain-specific linear projection. By inferring a latent vector to be used for generating each object, objects in different domains are clustered in shared groups, and thus we can find matching between clusters in an unsupervised manner. We present efficient inference procedures for the proposed model based on a stochastic EM algorithm. The effectiveness of the proposed model is demonstrated with experiments using synthetic and real data sets.

1 Introduction

Object matching is an important task for finding correspondences between objects in different domains. Examples of object matching include matching an image with an annotation (Socher and Fei-Fei 2010), an English word with a French word (Tripathi, Klami, and Virpioja 2010), and a user identification with a user identification in a different database for recommendation (Li, Yang, and Xue 2009). Most object matching methods require similarity measures between objects in different domains, or paired data that contain correspondence information.

Similarity measures and correspondences might not be available. Defining similarities and generating correspondences incur a cost and require time, and they are sometimes unobtainable because of the need to preserve privacy. For this situation, unsupervised object matching methods have been proposed, such as kernelized sorting (Quadrianto et al. 2010), least squares object matching (Yamada and Sugiyama 2011), matching canonical correlation analysis (Haghighi et al. 2008), and variational Bayesian matching (Klami 2012). These methods find one-to-one matching. However, matching is not necessarily one-to-one in some applications. For example, in image annotation, an annotation could be attached to multiple images that show the same thing, and an image that shows multiple things could have multiple annotations. When matching English and French vocabularies, multiple English words with the same meaning could correspond to multiple French words with the same meaning. Other limitations of these methods are that the number of domains needs to be two, and the numbers of objects in different domains must be the same. There can be more than two domains in some applications, for example matching multilingual vocabularies such as English, French and German, and the vocabulary size of the languages is usually different.

In this paper, we propose a probabilistic latent variable model for unsupervised cluster matching, which is a task that involves the many-to-many matching of objects in multiple domains. The proposed model can handle more than two domains with different numbers of objects. The proposed model is called the *many-to-many matching latent variable model* (MMLVM), and assumes that there are an infinite number of latent vectors that are shared by all domains. Each object is generated using one of the latent vectors and a domain-specific linear projection. The latent vector to be used for generating an object is unknown. By assigning a latent vector for each object, we can cluster objects in different domains, and find matching between clusters. The number of clusters is automatically inferred from the given data by using a Dirichlet process prior. Because the proposed model is a probabilistic generative model, we can extend it in a probabilistically principled manner, and use it, for example, for handling missing data, integration with other probabilistic models, and generalization to exponential family distributions. The inference of the proposed model is based on a stochastic EM algorithm, in which the Gibbs sampling of cluster assignments and the maximum joint likelihood estimation of projection matrices are alternately iterated while marginalizing out latent vectors.

The remainder of this paper is organized as follows. We formulate the proposed model in Section 2, and describe efficient inference procedures for the proposed model in Section 3. In Section 4, we briefly review related work. In Section 5, we demonstrate the effectiveness of the proposed model with experiments using synthetic and real data sets.

Table 1: Notation.

Symbol	Description
D	number of domains
N_d	number of objects in the dth domain
M_d	dimensionality of the dth domain
K	dimensionality of a latent vector

Finally, we present concluding remarks and a discussion of future work in Section 6.

2 Proposed model

Suppose that we are given objects in D domains $\boldsymbol{X} = \{\boldsymbol{X}_d\}_{d=1}^{D}$, where $\boldsymbol{X}_d = \{\boldsymbol{x}_{dn}\}_{n=1}^{N_d}$ is a set of objects in the dth domain, and $\boldsymbol{x}_{dn} \in \mathbb{R}^{M_d}$ is the feature vector of the nth object in the dth domain. Our notation is summarized in Table 1. Note that we are unaware of any correspondences between objects in different domains. The number of objects N_d and the dimensionality M_d for each domain can be different from those of other domains. The task is to match clusters of objects across multiple domains in an unsupervised manner.

The model proposed for this task is a probabilistic latent variable model. The proposed model assumes that there are potentially a countably infinite number of clusters, and each cluster j has a latent vector $\boldsymbol{z}_j \in \mathbb{R}^K$ in a K-dimensional latent space. Each object $\boldsymbol{x}_{dn}$ in the dth domain is generated depending on a domain-specific projection matrix $\boldsymbol{W}_d \in \mathbb{R}^{M_d \times K}$ and a latent vector $\boldsymbol{z}_{s_{dn}}$ that is selected from a set of latent vectors $\boldsymbol{Z} = \{\boldsymbol{z}_j\}_{j=1}^{\infty}$. Here, $s_{dn} \in \{1, \cdots, \infty\}$ is the latent cluster assignment of object $\boldsymbol{x}_{dn}$. Objects that use the same latent vector, or that have the same cluster assignment s_{dn}, are considered to match. Figure 1 shows the relationship between latent vectors and objects in two domains, where arrows that indicate the corresponding latent vectors for each object are hidden.

Specifically, the proposed model is an infinite mixture model, where the probability of object $\boldsymbol{x}_{dn}$ is given by

$$p(\boldsymbol{x}_{dn}|\boldsymbol{Z}, \boldsymbol{W}, \boldsymbol{\theta}) = \sum_{j=1}^{\infty} \theta_j \mathcal{N}(\boldsymbol{x}_{dn}|\boldsymbol{W}_d \boldsymbol{z}_j, \alpha^{-1}\boldsymbol{I}), \quad (1)$$

where $\boldsymbol{W} = \{\boldsymbol{W}_d\}_{d=1}^{D}$ is a set of projection matrices, $\boldsymbol{\theta} = (\theta_j)_{j=1}^{\infty}$ is mixture weights, θ_j represents the probability that the jth cluster is chosen, and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. In the proposed model, a set of latent vectors $\boldsymbol{Z}$ are shared among multiple domains, but projection matrix $\boldsymbol{W}_d$ depends on the domain. By sharing the latent vectors, we can assign objects in different domains to common clusters, and find matching between clusters. By employing domain-specific projection matrices, we can handle multiple domains with different dimensionalities and different properties. Given latent vectors, an arbitrary number of objects can be generated for each domain independently. Therefore, we can handle domains with different numbers of objects.

In summary, the proposed model generates objects in multiple domains $\boldsymbol{X}$ according to the following process,

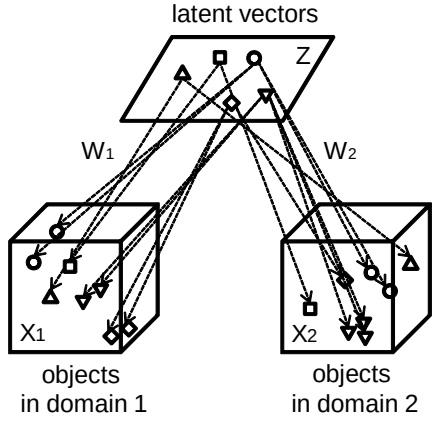

Figure 1: Relationship between latent vectors and objects in two domains.

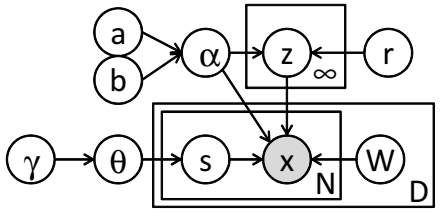

Figure 2: Graphical model representations of the proposed model.

1. Draw cluster proportions
   $\boldsymbol{\theta} \sim \text{GEM}(\gamma)$
2. Draw a precision parameter
   $\alpha \sim \text{Gamma}(a, b)$
3. For each cluster: $j = 1, \ldots, \infty$
   (a) Draw a latent vector
   $\boldsymbol{z}_j \sim \mathcal{N}(\boldsymbol{0}, (\alpha r)^{-1}\boldsymbol{I})$
4. For each domain: $d = 1, \ldots, D$
   (a) For each object: $n = 1, \ldots, N_d$
      i. Draw a cluster assignment
      $s_{dn} \sim \text{Discrete}(\boldsymbol{\theta})$
      ii. Draw an observation vector
      $\boldsymbol{x}_{dn} \sim \mathcal{N}(\boldsymbol{W}_d \boldsymbol{z}_{s_{dn}}, \alpha^{-1}\boldsymbol{I})$

Here, GEM(γ) is the stick-breaking process (Sethuraman 1994) that generates mixture weights for a Dirichlet process with concentration parameter γ. By using a Dirichlet process, we can automatically find the number of clusters from the given data. Figure 2 shows graphical model representations of the proposed model.

The joint probability of the data $\boldsymbol{X}$ and the cluster assignments $\boldsymbol{S} = \{\{s_{dn}\}_{n=1}^{N_d}\}_{d=1}^{D}$ is given by

$$p(\boldsymbol{X}, \boldsymbol{S}|\boldsymbol{W}, a, b, r, \gamma) = p(\boldsymbol{S}|\gamma)p(\boldsymbol{X}|\boldsymbol{S}, \boldsymbol{W}, a, b, r). \quad (2)$$

By marginalizing out mixture weights $\boldsymbol{\theta}$, the first factor is calculated by

$$p(\boldsymbol{S}|\gamma) = \frac{\gamma^J \prod_{j=1}^{J}(N_{\cdot j}-1)!}{\gamma(\gamma+1)\cdots(\gamma+N-1)}, \tag{3}$$

where $N = \sum_{d=1}^{D} N_d$ is the total number of objects, $N_{\cdot j}$ represents the number of objects assigned to cluster j, and J is the number of clusters for which $N_{\cdot j} > 0$. By marginalizing out latent vectors $\boldsymbol{Z}$ and precision parameter α, the second factor of (2) is calculated by

$$\begin{aligned} &p(\boldsymbol{X}|\boldsymbol{S},\boldsymbol{W},a,b,r) \\ &= (2\pi)^{-\frac{\sum_d M_d N_d}{2}} r^{\frac{KJ}{2}} \frac{b^a}{b'^{a'}} \frac{\Gamma(a')}{\Gamma(a)} \prod_{j=1}^{J} |\boldsymbol{C}_j|^{\frac{1}{2}}. \end{aligned} \tag{4}$$

Here,

$$a' = a + \frac{\sum_{d=1}^{D} M_d N_d}{2}, \tag{5}$$

$$b' = b + \frac{1}{2}\sum_{d=1}^{D}\sum_{n=1}^{N_d} \boldsymbol{x}_{dn}^\top \boldsymbol{x}_{dn} - \frac{1}{2}\sum_{j=1}^{J} \boldsymbol{\mu}_j^\top \boldsymbol{C}_j^{-1} \boldsymbol{\mu}_j, \tag{6}$$

$$\boldsymbol{\mu}_j = \boldsymbol{C}_j \sum_{d=1}^{D} \boldsymbol{W}_d^\top \sum_{n:s_{dn}=j} \boldsymbol{x}_{dn}, \tag{7}$$

$$\boldsymbol{C}_j^{-1} = \sum_{d=1}^{D} N_{dj} \boldsymbol{W}_d^\top \boldsymbol{W}_d + r\boldsymbol{I}, \tag{8}$$

where N_{dj} is the number of objects assigned to cluster j in domain d. The posterior for the precision parameter α is given by

$$p(\alpha|\boldsymbol{X},\boldsymbol{S},\boldsymbol{W},a,b) = \text{Gamma}(a',b'), \tag{9}$$

and the posterior for the latent vector $\boldsymbol{z}_j$ is given by

$$p(\boldsymbol{z}_j|\boldsymbol{X},\boldsymbol{S},\boldsymbol{W},r) = \mathcal{N}(\boldsymbol{\mu}_j, \alpha^{-1}\boldsymbol{C}_j). \tag{10}$$

3 Inference

We describe the inference procedures for the proposed model based on a stochastic EM algorithm, in which collapsed Gibbs sampling of cluster assignments $\boldsymbol{S}$ and the maximum joint likelihood estimation of projection matrices $\boldsymbol{W}$ are alternately iterated while marginalizing out the latent vectors $\boldsymbol{Z}$ and the precision parameter α.

In the E-step, given the current state of all but one latent assignment s_{dn}, a new value for s_{dn} is sampled from the following probability,

$$\begin{aligned} &p(s_{dn}=j|\boldsymbol{X},\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r,\gamma) \\ &= \frac{p(s_{dn}=j,\boldsymbol{S}_{\backslash dn}|\gamma)p(\boldsymbol{X}|s_{dn}=j,\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r)}{p(\boldsymbol{S}_{\backslash dn}|\gamma)p(\boldsymbol{X}_{\backslash dn}|\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r)p(\boldsymbol{x}_{dn}|\boldsymbol{W},a,b,r)} \\ &\propto \frac{p(s_{dn}=j,\boldsymbol{S}_{\backslash dn}|\gamma)}{p(\boldsymbol{S}_{\backslash dn}|\gamma)} \cdot \frac{p(\boldsymbol{X}|s_{dn}=j,\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r)}{p(\boldsymbol{X}_{\backslash dn}|\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r)}, \end{aligned} \tag{11}$$

where $\backslash dn$ represents a value or set excluding the nth object in the dth domain, and we use the fact that $p(s_{dn}=j|\boldsymbol{X},\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r,\gamma)$ does not depend on $p(\boldsymbol{x}_{dn}|\boldsymbol{W},a,b,r)$. The first factor is given by

$$\frac{p(s_{dn}=j,\boldsymbol{S}_{\backslash dn}|\gamma)}{p(\boldsymbol{S}_{\backslash dn}|\gamma)} = \begin{cases} \frac{N_{\cdot j\backslash dn}}{N-1+\gamma} & \text{for existing cluster} \\ \frac{\gamma}{N-1+\gamma} & \text{for new cluster,} \end{cases} \tag{12}$$

using (3). By using (4), the second factor is given by

$$\begin{aligned} &\frac{p(\boldsymbol{X}|s_{dn}=j,\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r)}{p(\boldsymbol{X}_{\backslash dn}|\boldsymbol{S}_{\backslash dn},\boldsymbol{W},a,b,r)} \\ &= (2\pi)^{-\frac{M_d}{2}} \frac{b'^{a'_{\backslash dn}}_{\backslash dn}}{b'^{a'_{s_{dn}=j}}_{s_{dn}=j}} \frac{\Gamma(a'_{s_{dn}=j})}{\Gamma(a'_{\backslash dn})} \frac{|\boldsymbol{C}_{j,s_{dn}=j}|^{\frac{1}{2}}}{|\boldsymbol{C}_{j\backslash dn}|^{\frac{1}{2}}}, \end{aligned} \tag{13}$$

where subscript $s_{dn}=j$ indicates the value when object $\boldsymbol{x}_{dn}$ is assigned to cluster j as follows,

$$a'_{s_{dn}=j} = a', \tag{14}$$

$$\begin{aligned} b'_{s_{dn}=j} = b'_{\backslash dn} &+ \frac{1}{2}\boldsymbol{x}_{dn}^\top \boldsymbol{x}_{dn} + \frac{1}{2}\boldsymbol{\mu}_{j\backslash dn}^\top \boldsymbol{C}_{j\backslash dn}^{-1} \boldsymbol{\mu}_{j\backslash dn} \\ &- \frac{1}{2}\boldsymbol{\mu}_{j,s_{dn}=j}^\top \boldsymbol{C}_{j,s_{dn}=j}^{-1} \boldsymbol{\mu}_{j,s_{dn}=j}, \end{aligned} \tag{15}$$

$$\boldsymbol{\mu}_{j,s_{dn}=j} = \boldsymbol{C}_{j,s_{dn}=j}(\boldsymbol{W}_d^\top \boldsymbol{x}_{dn} + \boldsymbol{C}_{j\backslash dn}^{-1}\boldsymbol{\mu}_{j\backslash dn}), \tag{16}$$

$$\boldsymbol{C}_{j,s_{dn}=j}^{-1} = \boldsymbol{W}_d^\top \boldsymbol{W}_d + \boldsymbol{C}_{j\backslash dn}^{-1}. \tag{17}$$

In the M-step, the projection matrices $\boldsymbol{W}$ are estimated by maximizing the logarithm of the joint likelihood (2). We can maximize it by using a gradient-based numerical optimization method such as the quasi-Newton method (Nocedal 1980). The gradient of the joint log likelihood is calculated by

$$\begin{aligned} \frac{\partial \log p(\boldsymbol{X},\boldsymbol{S}|\boldsymbol{W},a,b,r,\gamma)}{\partial \boldsymbol{W}_d} &= -\boldsymbol{W}_d \sum_{j=1}^{J} N_{dj}\boldsymbol{C}_j \\ &- \frac{a'}{b'}\sum_{j=1}^{J}\Big(N_{dj}\boldsymbol{W}_d\boldsymbol{\mu}_j\boldsymbol{\mu}_j^\top - \sum_{n:s_{dn}=j}\boldsymbol{x}_{dn}\boldsymbol{\mu}_j^\top\Big). \end{aligned} \tag{18}$$

By iterating the E-step that samples the cluster assignment s_{dn} by employing (11) for all objects $n = 1,\ldots,N_d$ in each domain $d = 1,\ldots,D$, and the M-step that maximizes the joint likelihood using (18) with respect to the projection matrix $\boldsymbol{W}_d$ for all domains $d = 1,\ldots,D$, we can obtain an estimate of the cluster assignments and projection matrices.

We can use cross-validation to select an appropriate dimensionality for the latent space K. With cross-validation, we assume that some features are missing in the given data, and infer the model with different K. Then, we select the K value that performed the best at predicting missing values.

Missing data

The proposed model can handle missing data. Let $\boldsymbol{h}_{dn} = (h_{dnm})_{m=1}^{M_d}$ be a vector indicating observed indexes, where $h_{dnm} = 1$ if x_{dnm} is observed, $h_{dnm} = 0$ otherwise, and M_{dn} is the number of observed values for object $\boldsymbol{x}_{dn}$. The posterior parameters are calculated as follows,

$$a' = a + \frac{\sum_{d=1}^{D}\sum_{n=1}^{N_d} M_{dn}}{2}, \tag{19}$$

$$b' = b + \frac{1}{2}\sum_{d=1}^{D}\sum_{n=1}^{N_d}(\boldsymbol{h}_{dn} \circ \boldsymbol{x}_{dn})^\top(\boldsymbol{h}_{dn} \circ \boldsymbol{x}_{dn}) - \frac{1}{2}\sum_{j=1}^{J}\boldsymbol{\mu}_j^\top \boldsymbol{C}_j^{-1}\boldsymbol{\mu}_j, \tag{20}$$

$$\boldsymbol{\mu}_j = \boldsymbol{C}_j \sum_{d=1}^{D}\boldsymbol{W}_d^\top \sum_{n:s_{dn}=j} \boldsymbol{h}_{dn} \circ \boldsymbol{x}_{dn}, \tag{21}$$

$$\boldsymbol{C}_j^{-1} = \sum_{d=1}^{D}\boldsymbol{W}_d^\top\Big(\sum_{n:s_{dn}=j}\mathrm{diag}(\boldsymbol{h}_{dn})\Big)\boldsymbol{W}_d + r\boldsymbol{I}, \tag{22}$$

where $\circ$ represents the Hadamard product, or element-wise product, and $\mathrm{diag}(\boldsymbol{h}_{dn})$ returns a diagonal matrix whose diagonal elements are $h_{dn1}, \ldots, h_{dnM_d}$.

4 Related work

Unsupervised object matching is a task that involves finding correspondences between objects in different domains without correspondence information. For example, kernelized sorting (Quadrianto et al. 2010) finds correspondences by permuting a set to maximize the dependence between two domains. Here, the Hilbert Schmidt Independence Criterion (HSIC) is used for measuring dependence. Kernelized sorting requires the numbers of objects in different domains to be the same. Kernelized sorting is extended to convex kernelized sorting (Djuric, Grbovic, and Vucetic 2012), which is guaranteed to find a globally optimal solution. Matching canonical correlation analysis (MCCA) (Haghighi et al. 2008) is another unsupervised object matching method, where bilingual translation lexicons are learned from two monolingual corpora. MCCA simultaneously finds latent variables that represent correspondences and latent vectors so that the latent vectors of corresponding objects exhibit the maximum correlation. (Tripathi et al. 2011) also proposed a method for unsupervised object matching that is related to MCCA. These methods assume the one-to-one matching of objects in two domains. On the other hand, the proposed model can find many-to-many matching, and is applicable to objects in more than two domains.

Manifold alignment is related to the proposed model because they both find latent vectors of multiple sets in a joint latent space. The unsupervised manifold alignment method (Wang and Mahadevan 2009) finds latent vectors of different domains in a joint latent space in an unsupervised manner. The method first identifies all the possible matches for each example leveraging its local geometry, and then finds an embedding in the latent space. The method requires permutations of the factorial of the number of neighborhoods to match the local geometry.

There has been some work on improving the learning performance of a classification task by using labeled objects in different domains without correspondence information. For example, multiple outlook mapping (MOMAP) (Harel and Mannor 2011) improves the performance by matching the moments of the empirical distributions for each class of two domains. (Shi et al. 2010) proposed a transfer learning method that improves the learning performance by embedding both source and target domains in a joint latent space when a limited number of target objects are labeled. These methods require labeled objects. On the other hand, the proposed method does not require any labeled objects.

The proposed model is an extension of probabilistic canonical correlation analysis (CCA) (Bach and Jordan 2005), which finds dependences between objects in two domains by projecting objects in a latent space. Probabilistic CCA requires correspondence information between different domains since it takes a set of paired objects as input. On the other hand, the proposed model can find dependences in an unsupervised manner without correspondence information by taking a set of objects for each domain as input. CCA is successfully used for a wide variety of applications, such as multi-label prediction (Rai and Daumé III 2009; Sun, Ji, and Ye 2011), information retrieval (Hardoon, Szedmak, and Shawe-Taylor 2004), and image annotation (Kimura et al. 2010). The proposed model can be used for these applications when supervised data are unavailable.

The proposed model can be seen as a generalization of the infinite Gaussian mixture model (Rasmussen 2000). When the dimensionality of the latent space is the same as that of the observed space $D = K$ and $\boldsymbol{W}_d = \boldsymbol{I}$ for all domains, the proposed model corresponds to the infinite Gaussian mixture model. The proposed model is different from the mixture of PCA (Tipping and Bishop 1999), where latent vectors are not shared among different objects.

5 Experiments

We evaluated the proposed model quantitatively by using three synthetic and four real data sets. The statistics of the seven data sets are shown in Table 2. There are two domains for all the data sets. Synth3, Synth5 and Synth10 are synthetic data sets with different true dimensionalities of the latent space $K^* = 3, 5$ and 10, respectively. We generated the synthetic data sets using the following procedure. First, we sampled latent vectors $\boldsymbol{z}_j$ for $j = 1, \cdots, J^*$ from a K^*-dimensional normal distribution with mean $\mathbf{0}$ and covariance $\boldsymbol{I}$. Then, we generated projection matrices $\boldsymbol{W}_d$ for $d = 1, 2$, where each element is drawn from a normal distribution with mean 0 and variance 1. Finally, we generated N/J^* objects for each cluster j using a normal distribution with mean $\boldsymbol{W}_d\boldsymbol{z}_j$ and covariance $\alpha^{-1}\boldsymbol{I}$, and obtained N objects in total for each domain $d = 1, 2$. Iris, Glass, Wine and MNIST are the real data sets, which were obtained from LIBSVM multi-class data sets (Chang and Lin 2011), and generated objects in two domains by randomly splitting the features

Table 2: Statistics of the data sets: the number of objects N, the dimensionality of the objects M_d, and the true number of clusters J^*, and the true dimensionality of the latent space K^*.

	N_1/N_2	M_1	M_2	J^*	K^*
Synth3	200	50	50	5	3
Synth5	200	50	50	5	5
Synth10	200	50	50	5	10
Iris	150	2	2	3	N/A
Glass	214	4	5	7	N/A
Wine	178	6	7	3	N/A
MNIST	200	392	392	10	N/A

into two parts for each data set as (Quadrianto et al. 2010; Djuric, Grbovic, and Vucetic 2012) did for their experiments. Because there is no overlapping feature, we cannot calculate similarities between objects in different domains.

For the evaluation measurement, we used the adjusted Rand index (Hubert and Arabie 1985), which quantifies the similarity between inferred clusters and true clusters, and takes the value from -1 to 1, and gives 0 for random clustering. For the real data sets, we assume that the category label of each object is its true cluster assignment. With unsupervised cluster matching, the adjusted Rand index measures how well objects with the same label in different domains are assigned to the same cluster as well as measuring the clustering performance within each domain.

With the proposed method, we used the dimensionality of the latent space $K = 5$, and set the hyperparameters $a = 1, b = 1, r = 1, \gamma = 1$ for all the data sets. To alleviate the local optimum problem, we ran the inference five times with different initial conditions, and selected the result that achieved the highest likelihood. For comparison, we used k-means (KM), convex kernelized sorting (CKS) (Djuric, Grbovic, and Vucetic 2012), and their combinations (KM-CKS and CKS-KM) as described below. The KM method is widely used for clustering. Although KM is not a cluster matching method, we included KM as a baseline method to show the adjusted Rand index when only objects in the same domain were clustered. The CKS method is an unsupervised object matching method. It directly finds correspondence between objects, and does not cluster objects in the same domain. With KM-CKS, first we discovered clusters by using KM for each domain individually, and then found the correspondence between clusters in two domains by using CKS. We used the mean vector of each cluster as the input for matching clusters by CKS. With the CKS-KM, after matching objects using CKS, we combined matched objects in two domains into a vector, and estimated clusters using KM. We employed CKS as comparison methods since it achieved higher performance than kernelized sorting and matching canonical correlation analysis (Djuric, Grbovic, and Vucetic 2012; Jagarlamudi, Juarez, and Daumé III 2010). With KM, KM-CKS and CKS-KM, we used the number of clusters estimated by the proposed model. For comparison with object matching based methods (CKS, CKS-KM), we used data sets that had the same numbers of objects in the two domains $N_1 = N_2$ for each data set.

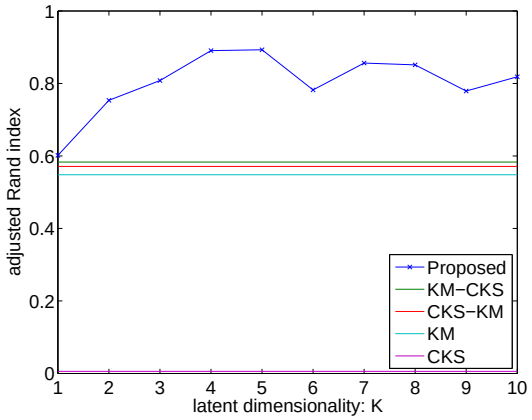

Figure 3: Adjusted Rand index achieved by the proposed model with different latent dimensionalities K using the Synth5 data set whose true latent dimensionality is $K^* = 5$.

Table 3 shows the adjusted Rand index for the seven data sets, which were averaged over 10 experiments for each data set. For all the data sets, the proposed model achieved the highest adjusted Rand index. This result indicates that the proposed model can infer matching clusters by assuming a shared latent space. KM-CKS achieved higher performance than KM by matching clusters as a post-processing step. With KM-CKS, since clusters are inferred individually for each domain, the estimated clusters might be different in different domains. On the other hand, since the proposed model infers clusters simultaneously for all domains, it successfully found shared clusters compared with KM-CKS as shown by the higher adjusted Rand index than KM-CKS. The adjusted Rand index obtained with the CKS method was low, because it does not cluster objects. By clustering the result of CKS, CKS-KM improved the cluster matching performance. However, it did not outperform the proposed model, because errors accumulated in object matching by CKS cannot be corrected in the clustering process with k-means.

Figure 3 shows the adjusted Rand index achieved by the proposed model with different latent dimensionalities using the Synth5 data set. The value was highest when the latent dimensionality of the model was the same as the true latent dimensionality $K = K^* = 5$. The proposed model with $K \neq K^*$ also performed better than the other methods. This result indicates that the proposed model is robust to the latent dimensionality setting because of the Bayesian inference.

Figure 4 shows the adjusted Rand index with different numbers of domains D using the Synth5 data set. With KM-CKS, CKS-KM and CKS, we found matching across multiple domains by matching clusters/objects between the Dth domain and each of the other $D-1$ domains, and then combined the results. In general, the adjusted Rand index decreases as the number of domains increases, since the number of possible combinations of cluster matching increases.

Table 3: Average adjusted Rand index and its standard deviation. Values in bold typeface are statistically better from those in normal typeface as indicated by a paired t-test.

	Proposed	KM	KM-CKS	CKS	CKS-KM
Synth3	$\mathbf{0.875} \pm 0.101$	0.525 ± 0.014	0.589 ± 0.117	0.014 ± 0.005	0.699 ± 0.135
Synth5	$\mathbf{0.893} \pm 0.126$	0.548 ± 0.029	0.583 ± 0.198	0.006 ± 0.007	0.571 ± 0.182
Synth10	$\mathbf{0.827} \pm 0.145$	0.556 ± 0.026	0.553 ± 0.165	0.009 ± 0.006	0.678 ± 0.170
Iris	$\mathbf{0.383} \pm 0.189$	0.224 ± 0.091	0.254 ± 0.154	0.003 ± 0.002	0.207 ± 0.089
Glass	$\mathbf{0.160} \pm 0.020$	0.050 ± 0.008	0.052 ± 0.011	0.001 ± 0.001	0.047 ± 0.010
Wine	$\mathbf{0.222} \pm 0.111$	0.125 ± 0.025	0.142 ± 0.046	0.001 ± 0.001	0.107 ± 0.038
MNIST	$\mathbf{0.085} \pm 0.016$	0.030 ± 0.007	0.037 ± 0.008	0.008 ± 0.005	0.041 ± 0.016

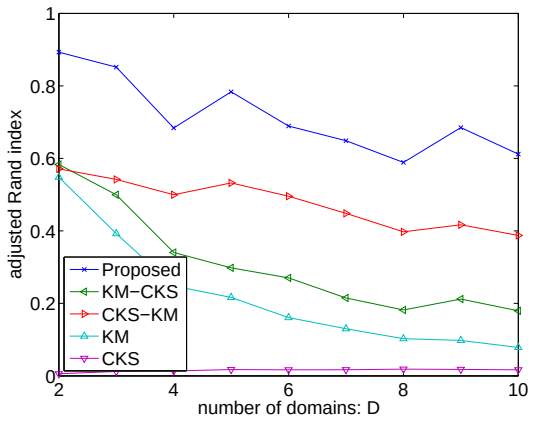

Figure 4: Adjusted Rand index with different numbers of domains D using the Synth5 data set.

However, the proposed method consistently achieved the highest performance for each number of domains.

The computational time for inference of the proposed model with the Synth5 data set was 18 seconds. The demanding part of the inference is E-step. The computational complexity of each E-step is $O(NJK^3)$, where N is the total number of objects, J is the number of clusters, and K is the latent dimensionality.

6 Conclusion

We proposed a generative model approach for unsupervised cluster matching, which we call the many-to-many matching latent variable model. In experiments, we confirmed that the proposed model can perform much better than object matching, clustering and their combinations. Advantages of the proposed model over the existing methods are that it can find many-to-many matching, and can handle multiple domains with different numbers of objects. Because the proposed model is a probabilistic generative model, we can extend it in a probabilistically principled manner.

Although our results have been encouraging as a first step towards unsupervised object clustering, we must extend our approach in a number of directions. First, we would like to extend the proposed model to other types of data. The proposed model assumes real values with Gaussian noise for features. However, the features can be discrete values for bag-of-features represented images and annotations for example. We can handle discrete values in the proposed framework by incorporating topic models (Blei, Ng, and Jordan 2003). The proposed model assumes the linearity of features with respect to their latent vectors. We can relax this assumption by using nonlinear matrix factorization techniques (Lawrence and Urtasun 2009). Second, we would like to evaluate the proposed model in a semi-unsupervised setting (Kimura et al. 2010), where a small number of object correspondences over different domains are available. The information can assist the matching by incorporating a condition stating that the cluster assignments of the corresponding objects become the same. Finally, we would like to use the proposed method for real applications, which include image annotation (Socher and Fei-Fei 2010), cross domain recommendation (Li, Yang, and Xue 2009), multi-lingual corpus analysis (Boyd-Graber and Blei 2009; Iwata, Mochihashi, and Sawada 2010), machine translation (Haghighi et al. 2008), and bioinformatics (Wang and Mahadevan 2008).

References

Bach, F. R., and Jordan, M. I. 2005. A probabilistic interpretation of canonical correlation analysis. Technical Report 688, Department of Statistics, University of California, Berkeley.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research* 3:993–1022.

Boyd-Graber, J., and Blei, D. 2009. Multilingual topic models for unaligned text. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 75–82. AUAI Press.

Chang, C., and Lin, C. 2011. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2(3):27.

Djuric, N.; Grbovic, M.; and Vucetic, S. 2012. Convex kernelized sorting. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Haghighi, A.; Liang, P.; Berg-Kirkpatrick, T.; and Klein, D. 2008. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, 771–779.

Hardoon, D. R.; Szedmak, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: an overview with applica-

tion to learning methods. *Neural Computation* 16(12):2639–2664.

Harel, M., and Mannor, S. 2011. Learning from multiple outlooks. In Getoor, L., and Scheffer, T., eds., *Proceedings of the 28th International Conference on Machine Learning*, ICML '11, 401–408.

Hubert, L., and Arabie, P. 1985. Comparing partitions. *Journal of Classification* 2(1):193–218.

Iwata, T.; Mochihashi, D.; and Sawada, H. 2010. Learning common grammar from multilingual corpus. In *Proceedings of 48th Annual Meeting of the Association for Computational Linguistics*, 184–188. Association for Computational Linguistics.

Jagarlamudi, J.; Juarez, S.; and Daumé III, H. 2010. Kernelized sorting for natural language processing. In *AAAI '10: Proceedings of the 24th AAAI Conference on Artificial Intelligence*.

Kimura, A.; Kameoka, H.; Sugiyama, M.; Nakano, T.; Maeda, E.; Sakano, H.; and Ishiguro, K. 2010. SemiCCA: Efficient semi-supervised learning of canonical correlations. In *Proceedings of IAPR International Conference on Pattern Recognition*, ICPR '10, 2933–2936.

Klami, A. 2012. Variational Bayesian matching. In *Proceedings of Asian Conference on Machine Learning*, 205–220.

Lawrence, N. D., and Urtasun, R. 2009. Non-linear matrix factorization with Gaussian processes. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 601–608.

Li, B.; Yang, Q.; and Xue, X. 2009. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, 617–624.

Nocedal, J. 1980. Updating quasi-Newton matrices with limited storage. *Mathematics of computation* 35(151):773–782.

Quadrianto, N.; Smola, A. J.; Song, L.; and Tuytelaars, T. 2010. Kernelized sorting. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 32(10):1809–1821.

Rai, P., and Daumé III, H. 2009. Multi-label prediction via sparse infinite CCA. In *Advances in Neural Information Processing Systems*, volume 22, 1518–1526.

Rasmussen, C. 2000. The infinite Gaussian mixture model. *Advances in neural information processing systems* 12(5.2):2.

Sethuraman, J. 1994. A constructive definition of Dirichlet priors. *Statistica Sinica* 4:639–650.

Shi, X.; Liu, Q.; Fan, W.; Yu, P. S.; and Zhu, R. 2010. Transfer learning on heterogenous feature spaces via spectral transformation. In *Proceedings of the IEEE International Conference on Data Mining*, ICDM '10, 1049–1054.

Socher, R., and Fei-Fei, L. 2010. Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, 966–973.

Sun, L.; Ji, S.; and Ye, J. 2011. Canonical correlation analysis for multilabel classification: A least-squares formulation, extensions, and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33:194–200.

Tipping, M., and Bishop, C. 1999. Mixtures of probabilistic principal component analyzers. *Neural Computation* 11(2):443–482.

Tripathi, A.; Klami, A.; Orešič, M.; and Kaski, S. 2011. Matching samples of multiple views. *Data Min. Knowl. Discov.* 23:300–321.

Tripathi, A.; Klami, A.; and Virpioja, S. 2010. Bilingual sentence matching using kernel CCA. In *MLSP '10: Proceedings of the 2010 IEEE International Workshop on Machine Learning for Signal Processing*, 130–135.

Wang, C., and Mahadevan, S. 2008. Manifold alignment using Procrustes analysis. In *ICML '08: Proceedings of the 25th International Conference on Machine Learning*, 1120–1127.

Wang, C., and Mahadevan, S. 2009. Manifold alignment without correspondence. In *Proceedings of the 21st International Joint Conference on Artifical Intelligence*, IJCAI'09, 1273–1278.

Yamada, M., and Sugiyama, M. 2011. Cross-domain object matching with model selection. In *In Proceedings of Fourteenth International Conference on Articial Intelligence and Statistics*, AISTATS '11, 807–815.