ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-387

Bayesian Multiview Dimensionality Reduction for Learning Predictive Subspaces

Mehmet Gönen[1] and **Gülefşan Bozkurt Gönen**[2] and **Fikret Gürgen**[3]

Abstract. Multiview learning basically tries to exploit different feature representations to obtain better learners. For example, in video and image recognition problems, there are many possible feature representations such as color- and texture-based features. There are two common ways of exploiting multiple views: forcing similarity (i) in predictions and (ii) in latent subspace. In this paper, we introduce a novel *Bayesian multiview dimensionality reduction* method coupled with supervised learning to find predictive subspaces and its inference details. Experiments show that our proposed method obtains very good results on image recognition tasks in terms of classification and retrieval performances.

1 INTRODUCTION

Multiview learning considers problems that can describe data points with different feature representations (i.e., *views* or *modalities*). The main idea is to exploit these different views to obtain better learners than the learners that can be found from each view separately. We can also transfer information from a subset of views (i.e., *source views*) to a particular view (i.e., *target view*) if we do not have enough training instances in the latter to build a reliable learner, which is known as *transfer learning*. There are two common approaches for multiview learning: (i) training separate learners for each view in a coupled manner by forcing them to have similar predictions on matching data points, (ii) projecting the data points from each view into a unified subspace and training a common learner in this subspace.

The first attempt to exploit multiple views is proposed for semi-supervised learning with two views, which is known as *co-training* [3]. In this approach, two distinct learners are trained separately using a small set of labeled instances from both views. Then, the unlabeled examples that are classified most confidently by these two learners are added to the set of labeled data points. Recently, the co-training idea is reformulated with a Bayesian approach applicable to a large set of problems [19]. One other strategy is minimizing the regularization errors of all views by training distinct learners simultaneously and a regularization term that penalizes the disagreement between views at the same time [4, 7, 8, 16, 20].

We can also exploit multiple views by finding a unified subspace from them. *Canonical correlation analysis* (CCA) [12] and kernel CCA (KCCA) [11], which extract a shared representation from two multivariate variables, are the first two methods that come to mind. The main restriction of such methods is that they are required to have matching samples from the views. [15] proposes a probabilistic KCCA variant using Gaussian process regression to find a shared representation from two views. [13] formulates an algorithm to find shared and private representations for each view using structured sparsity. [13, 15] obtain good performances for human pose estimation from image features (i.e., inferring missing data of one view using the other). [17, 18] extend *spectral embedding* and *stochastic neighborhood embedding* for multiview learning, respectively, and perform experiments on image and video retrieval tasks. However, the generalization performances of these unsupervised methods may not be good enough for prediction tasks due to their unsupervised nature.

[5, 6] propose a supervised algorithm, which is called *max-margin harmonium* (MMH), for finding a predictive subspace from multiple views using an undirected latent space Markov network with a large margin approach. MMH obtains better results than its competitor algorithms on video and image recognition data sets in terms of classification, annotation, and retrieval performances. [14] introduces a multiview metric learning algorithm that tries to preserve cross-view neighborhood by placing similarly labeled data points from different views nearby in the projected subspace. The proposed method outperforms CCA on an image retrieval task, where k-nearest neighbor strategy is used for retrieval.

In this paper, we propose a novel *Bayesian multiview dimensionality reduction* (BMDR) method, where data points from different views are projected into a unified subspace without the restriction of having matching data samples from these views. We make the following contributions: In §2, we give the graphical model of our approach for multiclass classification. §3 introduces an efficient variational approximation approach in a detailed manner. We report our experimental results in §4 and conclude in §5.

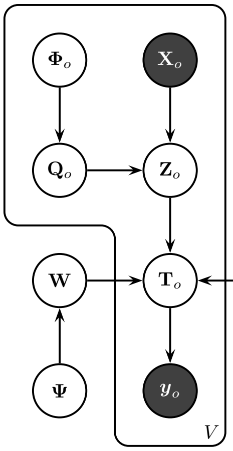
2 BAYESIAN MULTIVIEW DIMENSIONALITY REDUCTION FOR LEARNING PREDICTIVE SUBSPACES

We propose to combine linear dimensionality reduction and linear supervised learning in a joint probabilistic model to obtain predictive subspaces for multiview learning problems. The main idea is to map the training instances of different views to a unified subspace using linear projection matrices and to estimate the target outputs in this projected subspace. Performing dimensionality reduction and supervised learning separately (generally with two different objective functions) may not result in a predictive subspace and may have low generalization performance. For multiview learning problems,

[1] Department of Computational Biology, Sage Bionetworks, Seattle, WA 98109, USA, email:mehmet.gonen@sagebase.org
Present address: Department of Biomedical Engineering, Oregon Health & Science University, Portland, OR 97239, USA, email: gonen@ohsu.edu

[2] Department of Computer Engineering, Boğaziçi University, İstanbul, Turkey, email: gulefsanbozkurt@gmail.com

[3] Department of Computer Engineering, Boğaziçi University, İstanbul, Turkey, email: gurgen@boun.edu.tr



$$\begin{aligned}
\phi^f_{o,s} &\sim \mathcal{G}(\phi^f_{o,s}; \alpha_\phi, \beta_\phi) && \forall(o,f,s)\\
q^f_{o,s}|\phi^f_{o,s} &\sim \mathcal{N}(q^f_{o,s}; 0, \phi_s^{-1}) && \forall(o,f,s)\\
z^s_{o,i}|\boldsymbol{q}_{o,s}, \boldsymbol{x}_{o,i} &\sim \mathcal{N}(z^s_{o,i}; \boldsymbol{q}_{o,s}^\top \boldsymbol{x}_{o,i}, 1) && \forall(o,s,i)\\
\lambda_c &\sim \mathcal{G}(\lambda_c; \alpha_\lambda, \beta_\lambda) && \forall c\\
b_c|\lambda_c &\sim \mathcal{N}(b_c; 0, \lambda_c^{-1}) && \forall c\\
\psi^s_c &\sim \mathcal{G}(\psi^s_c; \alpha_\psi, \beta_\psi) && \forall(s,c)\\
w^s_c|\psi^s_c &\sim \mathcal{N}(w^s_c; 0, (\psi^s_c)^{-1}) && \forall(s,c)\\
t^c_{o,i}|b_c, \boldsymbol{w}_c, \boldsymbol{z}_{o,i} &\sim \mathcal{N}(t^c_{o,i}; \boldsymbol{w}_c^\top \boldsymbol{z}_{o,i} + b_c, 1) && \forall(o,c,i)\\
y_{o,i}|\boldsymbol{t}_{o,i} &\sim \prod_{c \neq y_{o,i}} \delta(t^{y_{o,i}}_{o,i} > t^c_{o,i}) && \forall(o,i)
\end{aligned}$$

Figure 1. Graphical model and distributional assumptions of Bayesian multiview dimensionality reduction for learning predictive subspaces.

we should consider the predictive performance of the unified projected subspace while learning the projection matrices. We give detailed derivations for multiclass classification, but our derivations can easily be extended to binary classification and regression.

Figure 1 illustrates the proposed probabilistic model with a graphical model and its distributional assumptions. $\mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes the normal distribution with the mean vector $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. $\mathcal{G}(\cdot; \alpha, \beta)$ denotes the gamma distribution with the shape parameter α and the scale parameter β. $\delta(\cdot)$ denotes the Kronecker delta function that returns 1 if its argument is true and 0 otherwise. The reason for choosing these specific distributions in our probabilistic model becomes clear when we explain our inference procedure in the following section. The notation we use throughout the manuscript is summarized in Table 1. As short-hand notations, all prior variables in the model are denoted by $\boldsymbol{\Xi} = \{\boldsymbol{\lambda}, \{\boldsymbol{\Phi}_o\}_{o=1}^V, \boldsymbol{\Psi}\}$, where the remaining variables by $\boldsymbol{\Theta} = \{\boldsymbol{b}, \{\mathbf{Q}_o\}_{o=1}^V, \{\mathbf{T}_o\}_{o=1}^V, \mathbf{W}, \{\mathbf{Z}\}_{o=1}^V\}$ and the hyper-parameters by $\boldsymbol{\omega} = \{\alpha_\lambda, \beta_\lambda, \alpha_\phi, \beta_\phi, \alpha_\psi, \beta_\psi\}$. Dependence on $\boldsymbol{\omega}$ is omitted for clarity throughout the manuscript.

Table 1. List of notation.

V	Number of views (i.e., feature representations)
N_o	Number of training instances for view o
D_o	Dimensionality of input space for view o
K	Number of classes
R	Dimensionality of unified projected subspace
$\mathbf{X}_o$	$D_o \times N_o$ data matrix for view o
$\mathbf{Q}_o$	$D_o \times R$ matrix of projection variables for view o
$\boldsymbol{\Phi}_o$	$D_o \times R$ matrix of priors over projection variables for view o
$\mathbf{Z}_o$	$R \times N_o$ matrix of projected variables for view o
$\mathbf{W}$	$R \times K$ matrix of weight parameters
$\boldsymbol{\Psi}$	$R \times K$ matrix of priors over weight parameters
$\boldsymbol{b}$	$K \times 1$ vector of bias parameters
$\boldsymbol{\lambda}$	$K \times 1$ vector of priors over bias parameters
$\mathbf{T}_o$	$N_o \times K$ matrix of auxiliary variables for view o
$\boldsymbol{y}_o$	$N_o \times 1$ vector of class labels from $\{1, \ldots, K\}$ for view o

The basic steps of our algorithm can be summarized as follows:

1. The data matrices $\{\mathbf{X}_o\}_{o=1}^V$ are used to project data points into a low-dimensional unified subspace using the projection matrices $\{\mathbf{Q}_o\}_{o=1}^V$.
2. The low-dimensional representations of data points $\{\mathbf{Z}_o\}_{o=1}^V$ and the shared set of classification parameters $\{\mathbf{W}, \boldsymbol{b}\}$ are used to calculate the classification scores.
3. Finally, the given class label vectors $\{\boldsymbol{y}_o\}_{o=1}^V$ are generated from the score matrices $\{\mathbf{T}_o\}_{o=1}^V$.

The auxiliary variables between the class labels and the projected instances are introduced to make the inference procedures efficient [1]. Exact inference for this probabilistic model is intractable and we instead formulate a deterministic variational approximation in the following section.

3 INFERENCE USING VARIATIONAL APPROXIMATION

Inference using a Gibbs sampling approach is computationally expensive [9]. We instead formulate a deterministic variational approximation, which is more efficient in terms of computation time. The variational methods use a lower bound on the marginal likelihood using an ensemble of factored posteriors to find the joint parameter distribution [2]. Note that there is not a strong coupling between the parameters of our model, although the factorable ensemble implies independence of the approximate posteriors. The factorable ensemble approximation of the required posterior for our model can be written as

$$\begin{aligned}
p(\boldsymbol{\Theta}, \boldsymbol{\Xi}|\{\mathbf{X}_o\}_{o=1}^V, \{\boldsymbol{y}_o\}_{o=1}^V) \approx q(\boldsymbol{\Theta}, \boldsymbol{\Xi}) = \\
q(\{\boldsymbol{\Phi}_o\}_{o=1}^V) q(\{\mathbf{Q}_o\}_{o=1}^V) q(\{\mathbf{Z}_o\}_{o=1}^V) \\
q(\boldsymbol{\lambda}) q(\boldsymbol{\Psi}) q(\boldsymbol{b}, \mathbf{W}) q(\{\mathbf{T}_o\}_{o=1}^V).
\end{aligned}$$

Each factor in the ensemble is defined just like its full conditional distribution:

$$q(\{\boldsymbol{\Phi}_o\}_{o=1}^V) = \prod_{o=1}^V \prod_{f=1}^{D_o} \prod_{s=1}^R \mathcal{G}(\phi^f_{o,s}; \alpha(\phi^f_{o,s}), \beta(\phi^f_{o,s}))$$

$$q(\{\mathbf{Q}_o\}_{o=1}^V) = \prod_{o=1}^V \prod_{s=1}^R \mathcal{N}(\boldsymbol{q}_{o,s}; \mu(\boldsymbol{q}_{o,s}), \Sigma(\boldsymbol{q}_{o,s}))$$

$$q(\{\mathbf{Z}_o\}_{o=1}^V) = \prod_{o=1}^V \prod_{i=1}^{N_o} \mathcal{N}(\boldsymbol{z}_{o,i}; \mu(\boldsymbol{z}_{o,i}), \Sigma(\boldsymbol{z}_{o,i}))$$

$$q(\boldsymbol{\lambda}) = \prod_{c=1}^K \mathcal{G}(\lambda_c; \alpha(\lambda_c), \beta(\lambda_c))$$

$$q(\boldsymbol{\Psi}) = \prod_{s=1}^R \prod_{c=1}^K \mathcal{G}(\psi^s_c; \alpha(\psi^s_c), \beta(\psi^s_c))$$

$$q(\boldsymbol{b}, \mathbf{W}) = \prod_{c=1}^K \mathcal{N}\left(\begin{bmatrix} b_c \\ \boldsymbol{w}_c \end{bmatrix}; \mu(b_c, \boldsymbol{w}_c), \Sigma(b_c, \boldsymbol{w}_c)\right)$$

$$q(\{\mathbf{T}_o\}_{o=1}^V) = \prod_{o=1}^V \prod_{i=1}^{N_o} \mathcal{TN}(\boldsymbol{t}_{o,i}; \mu(\boldsymbol{t}_{o,i}), \Sigma(\boldsymbol{t}_{o,i}), \rho(\boldsymbol{t}_{o,i})),$$

where $\alpha(\cdot)$, $\beta(\cdot)$, $\mu(\cdot)$, and $\Sigma(\cdot)$ denote the shape parameter, the scale parameter, the mean vector, and the covariance matrix for their arguments, respectively. $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot))$ denotes the truncated normal distribution with the mean vector $\boldsymbol{\mu}$, the covariance matrix $\boldsymbol{\Sigma}$, and the truncation rule $\rho(\cdot)$ such that $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) \propto \mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ if $\rho(\cdot)$ is true and $\mathcal{TN}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}, \rho(\cdot)) = 0$ otherwise.

We can bound the marginal likelihood using Jensen's inequality:

$$\log p(\{\boldsymbol{y}_o\}_{o=1}^V | \{\mathbf{X}_o\}_{o=1}^V) \geq$$
$$\mathrm{E}_{q(\boldsymbol{\Theta}, \boldsymbol{\Xi})}[\log p(\{\boldsymbol{y}_o\}_{o=1}^V, \boldsymbol{\Theta}, \boldsymbol{\Xi} | \{\mathbf{X}_o\}_{o=1}^V)] - \mathrm{E}_{q(\boldsymbol{\Theta}, \boldsymbol{\Xi})}[\log q(\boldsymbol{\Theta}, \boldsymbol{\Xi})]$$

and optimize this bound by maximizing with respect to each factor separately until convergence. The approximate posterior distribution of a specific factor $\boldsymbol{\tau}$ can be found as

$$q(\boldsymbol{\tau}) \propto \exp\Big(\mathrm{E}_{q(\{\boldsymbol{\Theta}, \boldsymbol{\Xi}\} \setminus \boldsymbol{\tau})}[\log p(\{\boldsymbol{y}_o\}_{o=1}^V, \boldsymbol{\Theta}, \boldsymbol{\Xi} | \{\mathbf{X}_o\}_{o=1}^V)]\Big).$$

Due to conjugate distributions in our probabilistic model, the resulting approximate posterior distribution of each factor follows the same distribution as the corresponding factor.

Dimensionality reduction part has two sets of parameters: the projection matrices that have normally distributed entries and the prior matrices that determine the precisions for these projection matrices. The approximate posterior distribution of the priors can be formulated as a product of gamma distributions:

$$q(\{\boldsymbol{\Phi}_o\}_{o=1}^V) =$$
$$\prod_{o=1}^V \prod_{f=1}^{D_o} \prod_{s=1}^R \mathcal{G}\left(\phi_{o,s}^f; \alpha_\phi + \frac{1}{2}, \left(\frac{1}{\beta_\phi} + \frac{\widetilde{(q_{o,s}^f)^2}}{2}\right)^{-1}\right),$$

where the tilde notation gives the posterior expectations as usual, i.e., $\widetilde{f(\boldsymbol{\tau})} = \mathrm{E}_{q(\boldsymbol{\tau})}[f(\boldsymbol{\tau})]$. The approximate posterior distribution of the projection matrices is a product of multivariate normal distributions:

$$q(\{\mathbf{Q}_o\}_{o=1}^V) =$$
$$\prod_{o=1}^V \prod_{s=1}^R \mathcal{N}(\boldsymbol{q}_{o,s}; \Sigma(\boldsymbol{q}_{o,s}) \mathbf{X}_o \widetilde{\boldsymbol{z}_o^s}, (\mathrm{diag}(\widetilde{\boldsymbol{\phi_{o,s}}}) + \mathbf{X}_o \mathbf{X}_o^\top)^{-1}).$$

The approximate posterior distribution of the projected instances can be found as a product of multivariate normal distributions:

$$q(\{\mathbf{Z}_o\}_{o=1}^V) =$$
$$\prod_{o=1}^V \prod_{i=1}^{N_o} \mathcal{N}(\boldsymbol{z}_{o,i}; \Sigma(\boldsymbol{z}_{o,i})(\widetilde{\mathbf{Q}_o^\top} \boldsymbol{x}_{o,i} + \widetilde{\mathbf{W}} \widetilde{\boldsymbol{t}_{o,i}} - \widetilde{\mathbf{W}\boldsymbol{b}}), (\mathbf{I} + \widetilde{\mathbf{W}\mathbf{W}^\top})^{-1}).$$

Supervised learning part has two sets of parameters: the bias vector and the weight matrix that have normally distributed entries, and the corresponding priors are from gamma distribution. The approximate posterior distributions of the priors on the bias vector and the weight matrix can be formulated as products of gamma distributions:

$$q(\boldsymbol{\lambda}) = \prod_{c=1}^K \mathcal{G}\left(\lambda_c; \alpha_\lambda + \frac{1}{2}, \left(\frac{1}{\beta_\lambda} + \frac{\widetilde{b_c^2}}{2}\right)^{-1}\right)$$
$$q(\boldsymbol{\Psi}) = \prod_{s=1}^R \prod_{c=1}^K \mathcal{G}\left(\psi_c^s; \alpha_\psi + \frac{1}{2}, \left(\frac{1}{\beta_\psi} + \frac{\widetilde{(w_c^s)^2}}{2}\right)^{-1}\right).$$

The approximate posterior distribution of the supervised learning parameters is a product of multivariate normal distributions:

$$q(\boldsymbol{b}, \mathbf{W}) = \prod_{c=1}^K \mathcal{N}\left(\begin{bmatrix} b_c \\ \boldsymbol{w}_c \end{bmatrix}; \Sigma(b_c, \boldsymbol{w}_c) \begin{bmatrix} \sum\limits_{o=1}^V \mathbf{1}^\top \widetilde{\boldsymbol{t}_o^c} \\ \sum\limits_{o=1}^V \widetilde{\mathbf{Z}_o} \widetilde{\boldsymbol{t}_o^c} \end{bmatrix},\right.$$
$$\left.\begin{bmatrix} \widetilde{\lambda_c} + \sum\limits_{o=1}^V N_o & \sum\limits_{o=1}^V \mathbf{1}^\top \widetilde{\mathbf{Z}_o^\top} \\ \sum\limits_{o=1}^V \widetilde{\mathbf{Z}_o} \mathbf{1} & \mathrm{diag}(\widetilde{\boldsymbol{\psi}_c}) + \sum\limits_{o=1}^V \widetilde{\mathbf{Z}_o \mathbf{Z}_o^\top} \end{bmatrix}^{-1}\right),$$

where we couple different views using the same bias vector and weight matrix for classification. The projection matrix for each view tries to embed corresponding data points accordingly.

The auxiliary variables of each point follow a truncated multivariate normal distribution whose mean vector depends on the weight matrix, the bias vector, and the corresponding projected instance. The approximate posterior distribution of the auxiliary variables is a product of truncated multivariate normal distributions:

$$q(\{\mathbf{T}_o\}_{o=1}^V) =$$
$$\prod_{o=1}^V \prod_{i=1}^{N_o} \mathcal{TN}\left(\boldsymbol{t}_{o,i}; \widetilde{\mathbf{W}^\top} \widetilde{\boldsymbol{z}_{o,i}} + \widetilde{\boldsymbol{b}}, \mathbf{I}, \prod_{c \neq y_{o,i}} \delta(t_{o,i}^{y_{o,i}} > t_{o,i}^c)\right).$$

However, we need to find the posterior expectations of the auxiliary variables to update the approximate posterior distributions of the projected instances and the supervised learning parameters. We can approximate these expectations using a naive sampling approach [10].

Updating the projection matrices $\{\mathbf{Q}_o\}_{o=1}^V$ is the most time-consuming step, which requires inverting $D_o \times D_o$ matrices for the covariance calculations and dominates the overall running time. When we have high-dimensional views, we can use an unsupervised dimensionality reduction method (e.g., principal component analysis) before running the algorithm to reduce the computational complexity of our algorithm.

After convergence, we have a separate projection matrix for each view and a unified set of classification parameters for the projected subspace. For a test data point, we can perform dimensionality reduction and classification using only the available views. $p(\mathbf{Q}_o | \{\mathbf{X}_o\}_{o=1}^V, \{\boldsymbol{y}_o\}_{o=1}^V)$ can be replaced with its approximate posterior distribution $q(\mathbf{Q}_o)$ for the prediction step. We obtain the predictive distribution of the projected instance $\boldsymbol{z}_{o,\star}$ for a new data point $\boldsymbol{x}_{o,\star}$ from a particular view as

$$p(\boldsymbol{z}_{o,\star} | \boldsymbol{x}_{o,\star}, \{\mathbf{X}_o\}_{o=1}^V, \{\boldsymbol{y}_o\}_{o=1}^V) =$$
$$\prod_{s=1}^R \mathcal{N}(z_{o,\star}^s; \mu(\boldsymbol{q}_{o,s})^\top \boldsymbol{x}_{o,\star}, 1 + \boldsymbol{x}_{o,\star}^\top \Sigma(\boldsymbol{q}_{o,s}) \boldsymbol{x}_{o,\star}).$$

The predictive distribution of the auxiliary variables $\boldsymbol{t}_{o,\star}$ can also be found by replacing $p(\boldsymbol{b}, \mathbf{W} | \{\mathbf{X}_o\}_{o=1}^V, \{\boldsymbol{y}_o\}_{o=1}^V)$ with its approximate posterior distribution $q(\boldsymbol{b}, \mathbf{W})$:

$$p(\boldsymbol{t}_{o,\star} | \{\mathbf{X}_o\}_{o=1}^V, \{\boldsymbol{y}_o\}_{o=1}^V, \boldsymbol{z}_{o,\star}) =$$
$$\prod_{c=1}^K \mathcal{N}\left(t_{o,\star}^c; \mu(b_c, \boldsymbol{w}_c)^\top \begin{bmatrix} 1 \\ \boldsymbol{z}_{o,\star} \end{bmatrix}, 1 + \begin{bmatrix} 1 & \boldsymbol{z}_{o,\star} \end{bmatrix} \Sigma(b_c, \boldsymbol{w}_c) \begin{bmatrix} 1 \\ \boldsymbol{z}_{o,\star} \end{bmatrix}\right)$$

and the predictive distribution of the class label $y_{o,\star}$ can be formulated using these auxiliary variables:

$$p(y_{o,\star} = c|\boldsymbol{x}_{o,\star}, \{\mathbf{X}_o\}_{o=1}^V, \{\boldsymbol{y}_o\}_{o=1}^V) = \mathrm{E}_{p(u)}\left[\prod_{j\neq c}\Phi\left(\Sigma(t_{o,\star}^j)^{-1}(u\Sigma(t_{o,\star}^c)+\mu(t_{o,\star}^c)-\mu(t_{o,\star}^j))\right)\right],$$

where the random variable u is standardized normal and $\Phi(\cdot)$ is the standardized normal cumulative distribution function. The expectation can be found using a naive sampling approach. If we have more than one view for testing, we can find the predictive distribution for each view separately and calculate the average probability to estimate the class label.

4 EXPERIMENTS

We test our algorithm BMDR by performing classification and retrieval experiments on FLICKR image data set from [5, 6], which contains 3411 images from 13 animal categories, namely, squirrel, cow, cat, zebra, tiger, lion, elephant, whales, rabbit, snake, antlers, wolf, and hawk. Each animal image is represented using 500-dimensional SIFT features and 634-dimensional low-level image features (e.g., color histogram, edge direction histogram, etc.). We use 2054 images for training and the rest for testing as provided. We implement our algorithm in Matlab, which is publicly available at https://github.com/mehmetgonen/bmdr. The default hyper-parameter values for BMDR are selected as $(\alpha_\lambda, \beta_\lambda) = (\alpha_\phi, \beta_\phi) = (\alpha_\psi, \beta_\psi) = (1, 1)$. We run our algorithm for 500 iterations.

In classification experiments, we use both views for training and only image features for testing (i.e., 634-dimensional low-level image features). We evaluate the classification results using the test accuracy. Table 2 shows the classification results on FLICKR data set. We compare our results with only the results of [5,6] because MMH outperforms several algorithms significantly in terms of classification accuracy using 30 latent topics. BMDR obtains higher test accuracies than MMH using 10 or 15 dimensions. Figure 2 displays eight training images, which corresponds the images with four smallest and four largest coordinate values, for each dimension obtained by BMDR with $R = 10$. We can easily see that most of the dimensions have clear meanings. For example, the dimensions #1, #4, #8, and #10 aim to separate zebra, whales, tiger, and lion categories, respectively, from other categories.

Table 2. Classification results on FLICKR data set.

Algorithm	Test Accuracy
MMH (30 topics)	51.70
BMDR ($R = 5$)	48.34
BMDR ($R = 10$)	54.02
BMDR ($R = 15$)	54.68

In retrieval experiments, each test image is considered as a separate query and training images are ranked based on their cosine similarities with the given test image. The cosine similarity is calculated using the subspace projections obtained using only image features. A training image is taken as relevant if it belongs to the category of the test image. We evaluate the retrieval results using the mean average precision score. Table 3 shows the retrieval results on FLICKR data set. We again compare our results with only the results of [5, 6] because MMH outperforms several algorithms significantly in terms of average precision using 60 latent topics. BMDR obtains significantly higher average precisions than MMH independent of the subspace dimensionality. Figure 3 displays one test image from each category and the first seven training images in the ranked result list for that test image. We see that the initial images in the result list are very meaningful for most of the categories even though there are some mistakes for confusing category groups such as {cat, tiger, lion, wolf}.

Table 3. Retrieval results on FLICKR data set.

Algorithm	Average Precision
MMH (60 topics)	0.163
BMDR ($R = 5$)	0.341
BMDR ($R = 10$)	0.383
BMDR ($R = 15$)	0.395

Our method also decreases the computational complexity of retrieval tasks due to low-dimensional representation used for images as in *indexing* and *hashing* schemes. When we need to retrieve images similar to a query image, we can calculate the similarities between the query image and other images very fast.

5 CONCLUSIONS

We introduce a Bayesian multiview dimensionality reduction method coupled with supervised learning to find predictive subspaces. We learn a unified subspace from multiple views (i.e., feature representations) by exploiting the correlation information between them. This approach can also be interpreted as transfer learning between different views. We give detailed derivations for multiclass classification using a variational approximation scheme and extensions to binary classification and regression are straightforward. Experimental results on FLICKR image data set show that the proposed method obtains a unified predictive subspace for classification and retrieval task using different views.

ACKNOWLEDGEMENTS

Most of this work has been done while the first author was working at the Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, Espoo, Finland. This work was financially supported by the Integrative Cancer Biology Program of the National Cancer Institute (grant no 1U54CA149237) and the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN, grant no 251170).

REFERENCES

[1] J. H. Albert and S. Chib, 'Bayesian analysis of binary and polychotomous response data', *Journal of the American Statistical Association*, **88**(422), 669–679, (1993).

[2] M. J. Beal, *Variational Algorithms for Approximate Bayesian Inference*, Ph.D. dissertation, The Gatsby Computational Neuroscience Unit, University College London, 2003.

[3] A. Blum and T. Mitchell, 'Combining labeled and unlabeled data with co-training', in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, (1998).

[4] U. Brefeld, T. Gärtner, T. Scheffer, and S. Wrobel, 'Efficient co-regularised least squares regression', in *Proceedings of the 23rd International Conference on Machine Learning*, (2006).

[5] N. Chen, J. Zhu, F. Sun, and E. P. Xing, 'Large-margin predictive latent subspace learning for multiview data analysis', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**(12), 2365–2378, (2012).

[6] N. Chen, J. Zhu, and E. P. Xing, 'Predictive subspace learning for multi-view data: A large margin approach', in *Advances in Neural Information Processing Systems 23*, (2010).

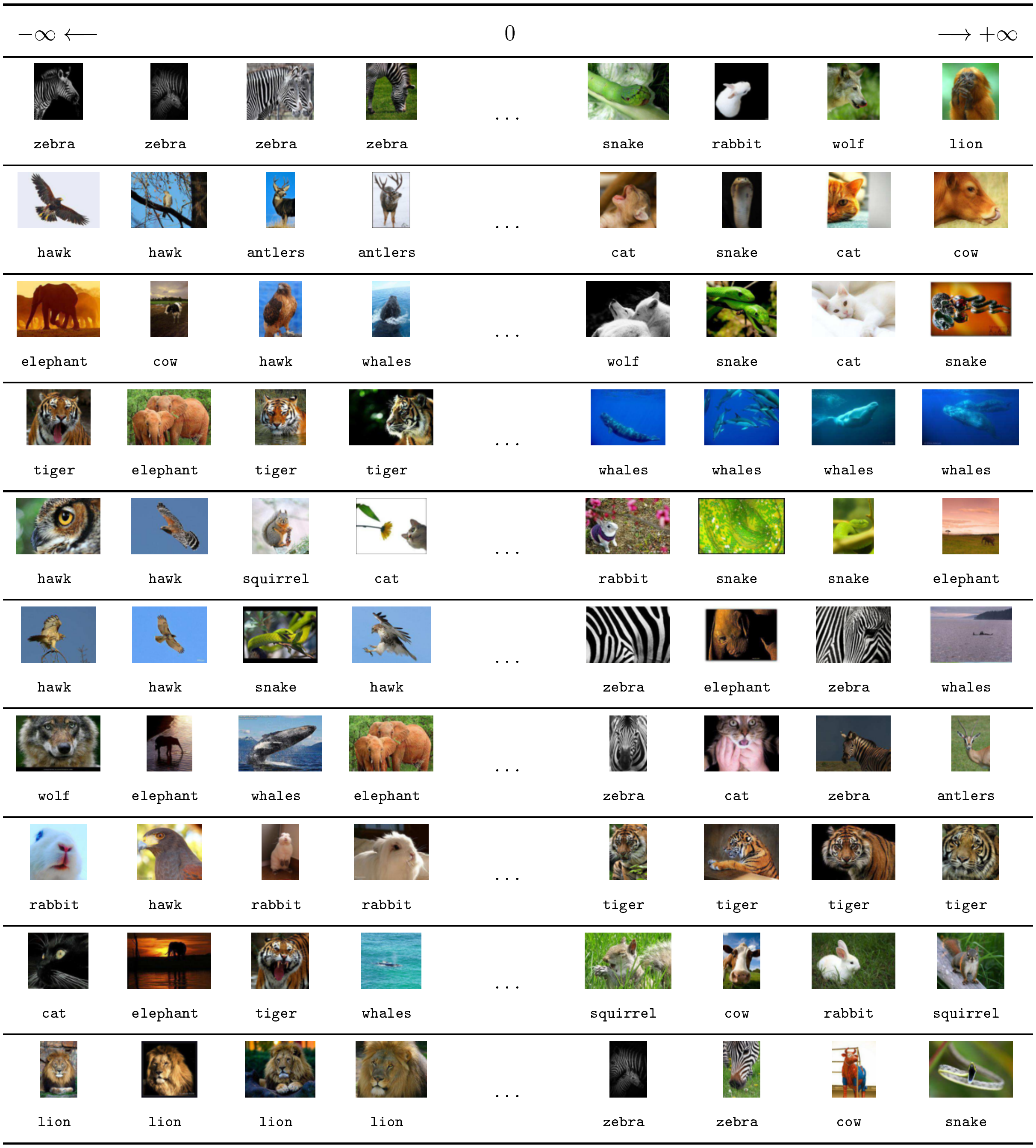


Figure 2. Training images of FLICKR data set projected on the dimensions obtained by BMDR with $R = 10$.

[7] T. Diethe, D. R. Hardoon, and J. Shawe-Taylor, ‘Constructing nonlinear discriminants from multiple data views’, in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, (2010).

[8] J. D. R. Farquhar, D. Hardoon, Hongying Meng, J. Shawe-Taylor, and S. Szedmak, ‘Two view learning: SVM-2K, theory and practice’, in *Advances in Neural Information Processing Systems 18*, (2006).

[9] A. E. Gelfand and A. F. M. Smith, ‘Sampling-based approaches to calculating marginal densities’, *Journal of the American Statistical Association*, **85**, 398–409, (1990).

[10] M. Girolami and S. Rogers, ‘Variational Bayesian multinomial probit regression with Gaussian process priors’, *Neural Computation*, **18**(8), 1790–1817, (2006).

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

[12] H. Hotelling, ‘Relations between two sets of variates’, *Biometrika*, **28**(3/4), 321–327, (1936).

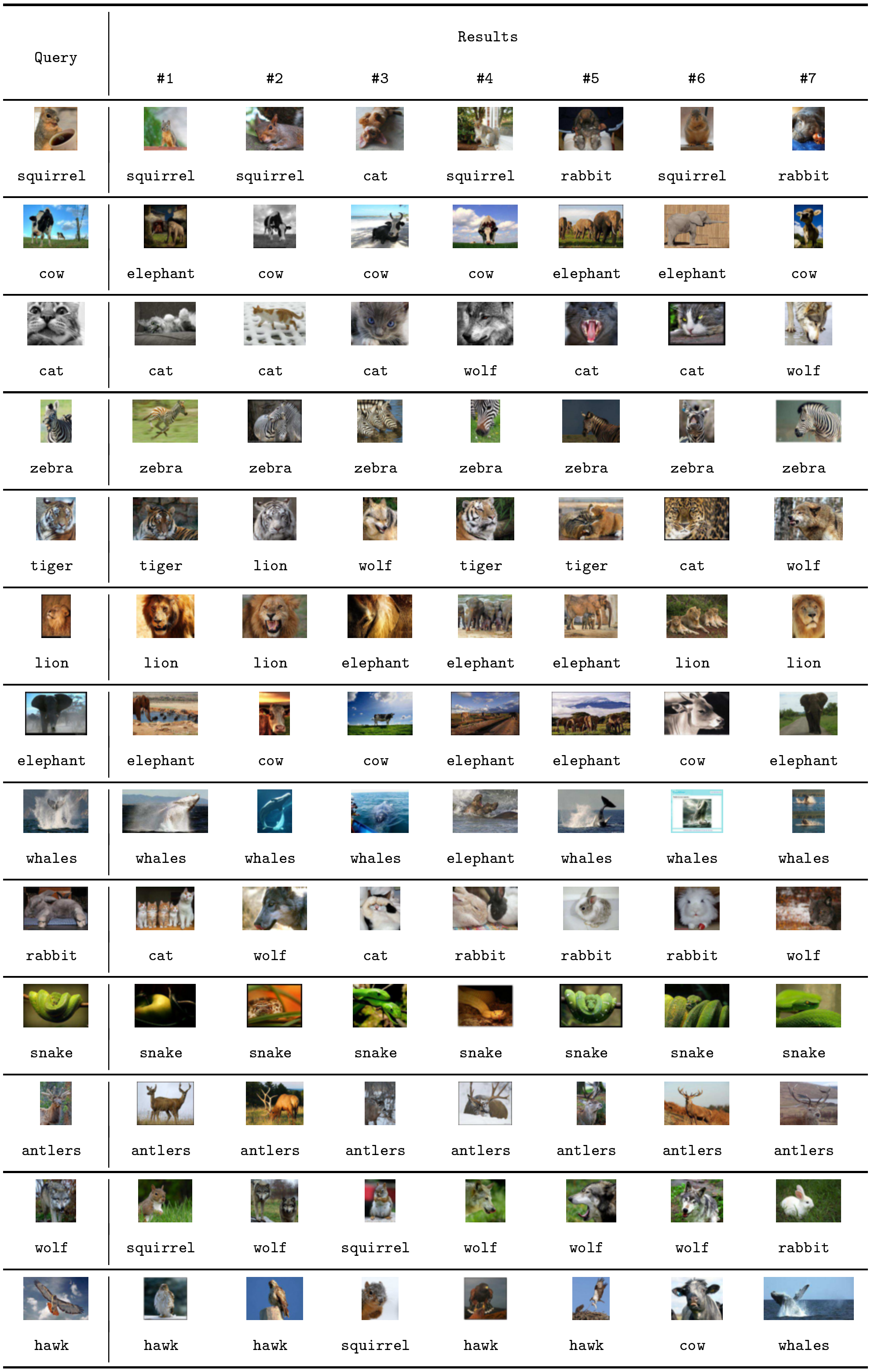
[13] Y. Jia, M. Salzmann, and T. Darrell, ‘Factorized latent spaces with structured sparsity’, in *Advances in Neural Information Processing Systems 23*, (2010).

[14] N. Quadrianto and C. H. Lampert, ‘Learning multi-view neighborhood preserving projections’, in *Proceedings of the 28th International Conference on Machine Learning*, (2011).

[15] A. Shon, K. Grochow, A. Hertzmann, and R. Rao, ‘Learning shared latent structure for image synthesis and robotic imitation’, in *Advances in Neural Information Processing Systems 18*, (2006).

[16] V. Sindhwani and D. S. Rosenberg, ‘An RKHS for multi-view learning and manifold co-regularization’, in *Proceedings of the 25th International Conference on Machine Learning*, (2008).

[17] T. Xia, D. Tao, T. Mei, and Y. Zhang, ‘Multiview spectral embedding’,



Query	Results #1	#2	#3	#4	#5	#6	#7
squirrel	squirrel	squirrel	cat	squirrel	rabbit	squirrel	rabbit
cow	elephant	cow	cow	cow	elephant	elephant	cow
cat	cat	cat	cat	wolf	cat	cat	wolf
zebra	zebra	zebra	zebra	zebra	zebra	zebra	zebra
tiger	tiger	lion	wolf	tiger	tiger	cat	wolf
lion	lion	lion	elephant	elephant	elephant	lion	lion
elephant	elephant	cow	cow	elephant	elephant	cow	elephant
whales	whales	whales	whales	elephant	whales	whales	whales
rabbit	cat	wolf	cat	rabbit	rabbit	rabbit	wolf
snake	snake	snake	snake	snake	snake	snake	snake
antlers	antlers	antlers	antlers	antlers	antlers	antlers	antlers
wolf	squirrel	wolf	squirrel	wolf	wolf	wolf	rabbit
hawk	hawk	hawk	squirrel	hawk	hawk	cow	whales

Figure 3. Sample queries and result images obtained by BMDR with $R = 10$ on FLICKR data set.

IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics, **40**(6), 1438–1446, (2010).

[18] B. Xie, Y. Mu, D. Tao, and K. Huang, 'm-SNE: Multiview stochastic neighbor embedding', *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, **41**(4), 1088–1096, (2011).

[19] S. Yu, B. Krishnapuram, R. Rosales, and R. B. Rao, 'Bayesian co-training', *Journal of Machine Learning Research*, **12**(Sep), 2649–2680, (2011).

[20] D. Zhang, J. He, Y. Liu, L. Si, and R. D. Lawrence, 'Multi-view transfer learning with a large margin approach', in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2011).