Online Bayesian Max-Margin Subspace Multi-View Learning

Jia He[1,3], Changying Du[2], Fuzhen Zhuang[1], Xin Yin[1], Qing He[1], Guoping Long[2]
[1]Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing 100190, China
[2]Laboratory of Parallel Software and Computational Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[3]University of Chinese Academy of Sciences, Beijing 100049, China
{hej, zhuangfz, yinx, heq}@ics.ict.ac.cn, {changying, guoping}@iscas.ac.cn

Abstract

Last decades have witnessed a number of studies devoted to multi-view learning algorithms, however, few efforts have been made to handle online multi-view learning scenarios. In this paper, we propose an online Bayesian multi-view learning algorithm to learn predictive subspace with max-margin principle. Specifically, we first define the latent margin loss for classification in the subspace, and then cast the learning problem into a variational Bayesian framework by exploiting the pseudo-likelihood and data augmentation idea. With the variational approximate posterior inferred from the past samples, we can naturally combine historical knowledge with new arrival data, in a Bayesian Passive-Aggressive style. Experiments on various classification tasks show that our model have superior performance.

1 Introduction

Nowadays, multi-view data are often generated from multiple information channels continuously, e.g., hundreds of YouTube videos consisting of visual, audio and text features are uploaded every minute. Multi-view learning arouses amounts of interests in the past decades [Blum and Mitchell, 1998; Yarowsky, 1995; Gönen and Alpaydın, 2011; Quang *et al.*, 2013; Sun and Chao, 2013]. Among them, the multi-view subspace learning approaches aim at obtaining a subspace shared by multiple views and then learning models in the shared subspace [Sharma *et al.*, 2012; Hardoon *et al.*, 2004; Guo and Xiao, 2012]. They are very useful for cross-view classification and retrieval. However, these approaches are prone to overfitting to small training data without considering the maximum margin principle. In [Chen *et al.*, 2012], a large-margin harmonium model (MMH) based on latent subspace Markov network is developed for multi-view data. But MMH is under the maximum entropy discrimination framework and cannot infer the penalty parameter of max-margin models in Bayesian style automatically. In [Du *et al.*, 2015], a posterior-regularized Bayesian approach is proposed to combine Principal Component Analysis (PCA) with the max-margin learning, which can infer the penalty parameter of max-margin models but cannot address multi-view data.

On the other hand, multi-view data often cannot be collected in a single time due to temporal and spatial constrictions in applications, while the traditional multi-view algorithm need store the entire training samples. Online learning is an efficient method to address this problem. Many efforts have been made on the studies of online learning [Cesa-Bianchi and Lugosi, 2006; Hazan *et al.*, 2007; Chechik *et al.*, 2010]. Unfortunately, there are few studies about online multi-view learning. OPMV is one of the few online multi-view learning [Zhu *et al.*, 2015]. OPMV is not in a Bayesian framework and does not introduce the max-margin principle, thus it is prone to overfitting to small training data. And OPMV is formulated as a point estimate by optimizing some deterministic objective function. Online Passive-Aggressive (PA) learning provides a method for online large-margin learning [Crammer *et al.*, 2006]. Although it enjoys strong discriminative ability suitable for predictive tasks, it is also formulated as a point estimate by optimizing some deterministic objective function. The point estimate can be affected seriously by inappropriate regularization, outliers and noises, especially when the training data arrive sequentially. Based on the online PA learning, Shi proposes a Bayesian PA learning method [Shi and Zhu, 2013] which infers a posterior under the Bayesian framework instead of a point estimate. Nevertheless, these online learning methods cannot process multi-view data. To the best of our knowledge, there has been few efforts focused on online multi-view learning under the Bayesian framework.

In this paper, we address the aforementioned problems by developing an online Bayesian multi-view subspace learning method with max-margin principle. Specifically, we first propose a predictive subspace learning method based on factor analysis and define a latent margin loss for classification in the subspace. Then we cast the learning problem into a variational Bayesian framework by exploiting the pseudo-likelihood and data augmentation idea which allows us to automatically infer the penalty parameter. With the variational approximate posterior inferred from the past samples, we can naturally combine historical knowledge with new arriving data, in a Bayesian Passive-Aggressive style. We update our model with the training data coming one by one, instead of storing all training data. Experiments on synthetic and various real classification tasks show both our batch and online model have superior performance, compared with a number of competitors.

Related Work

The earliest works of multi-view learning are introduced by Blum and Mitchell [Blum and Mitchell, 1998] and Yarowsky [Yarowsky, 1995]. Nowadays, there are many multi-view learning approaches, e.g., multiple kernel learning [Gönen and Alpaydın, 2011], disagreement-based multi-view learning [Blum and Mitchell, 1998] and late fusion methods which combine outputs of the models constructed from different view features [Ye *et al.*, 2012]. Especially, the multi-view subspace learning algorithms learn latent salient representation of multi-view data [Sharma *et al.*, 2012; Hardoon *et al.*, 2004]. This approach aims at obtaining a subspace shared by multiple views and then learn models in the shared subspace.

Online learning starts from the Perceptron algorithm [Rosenblatt, 1958] and has attracted much attention during the past years [Cesa-Bianchi and Lugosi, 2006; Hazan *et al.*, 2007; Grangier and Bengio, 2008; Chechik *et al.*, 2010]. Crammer proposes the Online Passive-Aggressive (PA) learning which provides a general framework for online large-margin learning[Crammer *et al.*, 2006], with many applications [Chiang *et al.*, 2008]. Online Bayesian Passive-Aggressive learning presents a generic framework of performing online learning for Bayesian max-margin models [Shi and Zhu, 2013].

2 The Model

In this section, we firstly propose the max-margin subspace learning based on factor analysis. Then we develop a multi-view classification with max-margin subspace learning under the Bayesian framework. Finally, we extend the batch model to the online scenario which trains the model with the samples coming one by one.

2.1 Max-margin Subspace Learning

Suppose we have a set of N observations $\mathbf{x}^{(n)}$, $n = 1, \cdots, N$ in d-dimension feature space and a $1 \times N$ label vector $\mathbf{y}$ with its element $y_n \in \{+1, -1\}$, $n = 1, \cdots, N$. Factor analysis projects an observation into a low dimensional space that captures the latent feature of data. The generative process for the n-th observation $\mathbf{x}^{(n)}$ is as follows:

$$\begin{array}{rcl} \varepsilon & \sim & \mathcal{N}(\varepsilon|\mathbf{0},\, \Phi) \\ \mathbf{x}^{(n)} & = & \mu + \mathbf{W}\mathbf{z}^{(n)} + \varepsilon, \end{array} \tag{1}$$

where $\varepsilon \in \mathbb{R}^{d \times 1}$ denotes the Gaussian noise, $\Phi \in \mathbb{R}^{d \times d}$ is a variance matrix of ε, $\mu \in \mathbb{R}^{d \times 1}$ is the mean value of $\mathbf{x}^{(n)}$, $\mathbf{W} \in \mathbb{R}^{d \times m}$ is the factor loading matrix, $\mathbf{z}^{(n)}$ is a m-dimensional latent variable.

The estimates of model variables $(\mu, \mathbf{W}, \Phi, \mathbf{Z})$ can be obtained as follows:

$$\max_{\mu,\mathbf{W},\Phi,\mathbf{Z}} \ell_s(\mu, \mathbf{W}, \Phi, \mathbf{Z}) = \max_{\mu,\mathbf{W},\Phi,\mathbf{Z}} \log \prod_{n=1}^{N} \frac{1}{(2\pi)^{d/2}|\Phi|}$$
$$\cdot \exp(-\frac{1}{2}(\mathbf{x}^{(n)} - \mu - \mathbf{W}\mathbf{z}^{(n)})^T \Phi^{-1} (\mathbf{x}^{(n)} - \mu - \mathbf{W}\mathbf{z}^{(n)})).$$

However, factor analysis is an unsupervised model, which learns the latent variables of the observations without using any label information. The max-margin principle can be introduced to incorporate label information into the factor analysis model. We define $\tilde{\mathbf{z}} = [\mathbf{z}^T,\ 1]^T$ as the augmented latent representation of observation $\mathbf{x}$, and let $f(\mathbf{x}; \tilde{\mathbf{z}}, \boldsymbol{\eta}) = \boldsymbol{\eta}^T \tilde{\mathbf{z}}$ be a discriminant function parameterized by $\boldsymbol{\eta}$. Now for fixed values of $\mathbf{Z}$ and $\boldsymbol{\eta}$, we can compute the margin loss on training data $(\mathbf{X}, \mathbf{y})$ by

$$\ell_m(\mathbf{Z}, \eta) = \sum_{n=1}^{N} \max(0, 1 - y_n f(\mathbf{x}^{(n)}; \tilde{\mathbf{z}}^{(n)}, \boldsymbol{\eta})). \tag{2}$$

The max-margin subspace learning model can be formulated as follows:

$$\max_{\mu,\mathbf{W},\Phi,\mathbf{Z},\eta} \ell_s(\mu, \mathbf{W}, \Phi, \mathbf{Z}) - C\ell_m(\mathbf{Z}, \eta), \tag{3}$$

where C is the regularization parameter.

2.2 Multi-view Classification with Bayesian M^{2}SL

Then we propose a Bayesian max-margin subspace multi-view learning (BM^{2}SMVL) model. We assume that N_v is the number of views, N_c is the number of classes, d_i is the dimension of the i-th view, the data matrix of the i-th view is $\mathbf{X}_i \in \mathbb{R}^{d_i \times N}$ consisting of N observations $\mathbf{x}_i^{(n)}$ in d_i-dimension feature space, $\mathbf{x}^{(n)} = \{\mathbf{x}_i^{(n)}, i = 1, \cdots, N_v\}$ denotes the n-th observation, and $\mathbf{Y}$ is a $N_c \times N$ label matrix consisting of N label vectors $\mathbf{y}^{(n)} = \{y_c^{(n)}, c = 1, \cdots, N_c\}$. If the n-th observation's label belongs to the c-th class, we define $y_c^{(n)} = +1$ otherwise $y_c^{(n)} = -1$.

In our BM^{2}SMVL model, each view $\mathbf{x}_i^{(n)}$ of the n-th observation $\mathbf{x}^{(n)}$ is generated from the latent variable $\mathbf{z}^{(n)}$. We impose prior distributions over all variables shown in Eq.(1). The generative process for the n-th observation is as follows:

$$\begin{array}{rcl} \mathbf{z}^{(n)} & \sim & \mathcal{N}(\mathbf{z}^{(n)}|\mathbf{0},\ \mathbf{I}_m) \\ \mu_i & \sim & \mathcal{N}(\mu_i|\mathbf{0},\ \beta_i^{-1}\mathbf{I}_{d_i}) \\ \alpha_i & \sim & \prod_{j=1}^{m} \Gamma(\alpha_{ij}|a_{\alpha_i}, b_{\alpha_i}) \\ \mathbf{W}_{\mathrm{i}}|\alpha_i & \sim & \prod_{j=1}^{d_i} \mathcal{N}(w_{ij}|\mathbf{0},\ \mathrm{diag}(\alpha_{\mathrm{i}})) \\ \phi_i & \sim & \Gamma(\phi_i|a_{\phi_i}, b_{\phi_i}) \\ \mathbf{x}_i^{(n)}|\mathbf{z}^{(n)} & \sim & \mathcal{N}(\mathbf{x}_i^{(n)}|\mathbf{W}_{\mathbf{i}}\mathbf{z}^{(n)} + \mu_{\mathrm{i}},\ \phi_i^{-1}\mathbf{I}_{d_i}), \end{array}$$

where $\Gamma(\cdot)$ is the Gamma distribution, β_i, a_{α_i}, b_{α_i}, a_{ϕ_i}, b_{ϕ_i} are the hyper-parameters, and $\mathbf{W}_{\mathrm{i}} \in \mathbb{R}^{d_i \times m}$. The prior on $\mathbf{W}_{\mathrm{i}}$ and α_i is introduced according to the automatic relevance determination [Reents and Urbanczik, 1998]. In order to improve the efficiency of our algorithm, we define the variance matrix Φ_i of the $\mathbf{x}_i^{(n)}$ as a diagonal matrix $\phi_i^{-1}\mathbf{I}_{d_i}$. Let $\Omega = (\mu, \alpha, \mathbf{W}, \phi, \mathbf{Z})$ denote all variables. $p_0(\Omega) = p_0(\mu)p_0(\mathbf{W}, \alpha)p_0(\phi)p_0(\mathbf{z})$ is the prior of Ω. We can verify the Bayesian posterior distribution $p(\Omega|\mathbf{X}) = p_0(\Omega)p(\mathbf{X}|\Omega)/p(\mathbf{X})$ is equal to the solution of the following optimization problem:

$$\min_{q(\Omega)\in\mathcal{P}} \mathrm{KL}(q(\Omega)\|p_0(\Omega)) - \mathbb{E}_{q(\Omega)}[\log p(\mathbf{X}|\Omega)], \tag{4}$$

where $\mathrm{KL}(q\|p)$ is the Kullback-Leibler divergence, and $\mathcal{P}$ is the space of probability distributions. When the observations are given, $p(\mathbf{X})$ is a constant.

Next, we adapt our model with the one-VS-rest strategy like that for SVM for multi-class classification problems. We have N_c classifiers, and take the c-th classification for an example: $f_c(\mathbf{x}^{(n)}; \tilde{\mathbf{z}}^{(n)}, \boldsymbol{\eta}_c) = \boldsymbol{\eta}_c^T \tilde{\mathbf{z}}^{(n)}$ denotes a discriminant function. Under the Bayesian framework, we impose a prior on $\boldsymbol{\eta}_c$ as follows:

$$\nu_c \sim p_0(\nu_c) = \Gamma(\nu | a_{\nu,c}, b_{\nu,c})$$
$$p(\boldsymbol{\eta}_c | \nu_c) = \mathcal{N}(\boldsymbol{\eta}_c | \mathbf{0}, \nu_c^{-1} \mathbf{I}_{(m+1)}),$$

where $a_{\nu,c}$ and $b_{\nu,c}$ are hyper-parameters. For simplify, let $\Theta = \{(\boldsymbol{\eta}_c, \nu_c)\}_{c=1}^{N_c}$.

Then we can replace the margin loss with the expected margin loss for the classification. We introduce

$$\varphi(\mathbf{Y}, \boldsymbol{\lambda} | \mathbf{Z}, \boldsymbol{\eta}) = \prod_{n=1}^{N} \prod_{c=1}^{N_c} \exp\{-2C \cdot \max(0, 1 - y_c^{(n)} \boldsymbol{\eta}_c^T \tilde{\mathbf{z}}^{(n)}\} \quad (5)$$

as the pseudo-likelihood of the n-th data's label variable. We can get our final model as follows:

$$\min_{q(\Omega,\Theta) \in \mathcal{P}} \mathrm{KL}(q(\Omega,\Theta) \| p_0(\Omega,\Theta)) - \mathbb{E}_{q(\Omega)}[\mathrm{log}p(\mathbf{X}|\Omega)]$$
$$- \mathbb{E}_{q(\Omega,\Theta)}[\log(\varphi(\mathbf{Y}|\mathbf{Z}, \boldsymbol{\eta}))], \quad (6)$$

where $p_0(\Omega,\Theta)$ is the prior, $p_0(\Omega,\Theta) = p_0(\Omega)p_0(\Theta)$, $p_0(\Theta_c) = p(\boldsymbol{\eta}_c|\nu_c)p_0(\nu_c)$ and C is the regularization parameter. Solving problem (6), we can get the posterior distribution

$$q(\Omega,\Theta) = \frac{p_0(\Omega,\Theta)p(\mathbf{X}|\Omega)\varphi(\mathbf{Y}|\mathbf{Z},\boldsymbol{\eta})}{\phi(\mathbf{X},\mathbf{Y})}, \quad (7)$$

where $\phi(\mathbf{X},\mathbf{Y})$ is the normalization constant. In order to approximate $q(\Omega,\Theta)$ we use variational approximate inference which is introduced in the Section 3.

2.3 Online BM^{2}SMVL

The goal of online learning is to minimize the cumulative loss for a certain prediction task from the sequentially arriving training samples. In this section, we present an online BM^{2}SMVL (OBM^{2}SMVL) based on the online Passive-Aggressive learning framework [Crammer *et al.*, 2006]. This generic framework for online large-margin learning has been used in many applications [Chiang *et al.*, 2008]. Online Bayesian Passive-Aggressive learning was presented for online Bayesian max-margin topic models [Shi and Zhu, 2013].

Assuming we have already got the posterior $q_t(\Omega,\Theta)$ at time t, when a new data $(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})$ is coming, we need update the new posterior distribution $q_{t+1}(\Omega,\Theta)$. For simplify, We denote $\mathbf{x}^{(t+1)} = \{\mathbf{x}_i^{(t+1)}\}_{i=1}^{N_v}, \mathbf{y}^{(t+1)} = \{y_c^{(t+1)}\}_{c=1}^{N_c}$.

Generally, we define ω as the parameterized model and $\ell(\omega; \mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})$ as the loss for the new data $(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})$. Our OBM^{2}SMVL sequentially infers a new posterior distribution $q_{t+1}(\omega)$ on the arrival of new data $(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})$ by solving the following optimization problem:

$$\min_{q(\omega) \in \mathcal{P}} \mathrm{KL}(q(\omega) \| q_t(\omega)) - \mathbb{E}_{q(\omega)}[\mathrm{log}p(\mathbf{x}^{(t+1)}|\omega)]$$
$$+ \ell(\omega; \mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)}).$$

The online model includes three main updating rule. Firstly, we hope $\mathrm{KL}(q(\omega)\|q_t(\omega))$ is as small as possible. It means that $q_{t+1}(\omega)$ is close to $q_t(\omega)$. Secondly, the likelihood of the new data $\mathbb{E}_{q(\omega)}[\mathrm{log}p(\mathbf{x}^{(t+1)}|\omega)]$ is high enough. Thirdly, the loss of the new data $\ell(\omega; \mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})$ is as small as possible. It means that the new model $q_{t+1}(\omega)$ suffers little loss from the new data.

To introduce the online idea to the above multi-view classification BM^{2}SMVL, we let (Ω,Θ) denote ω. A new posterior distribution $q_{t+1}(\Omega,\Theta)$ on the arrival of new data $(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})$ can be gotten by solving the following optimization problem:

$$\min_{q(\Omega,\Theta) \in \mathcal{P}} \mathrm{KL}(q(\Omega,\Theta) \| q_t(\Omega,\Theta)) - \mathbb{E}_{q(\Omega,\Theta)}[\mathrm{log}p(\mathbf{x}^{(t+1)}|\Omega,\Theta)]$$
$$+ \ell(\Omega,\Theta; \mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)}).$$

As above, we introduce $\varphi(\cdot)$ function to replace the hinge loss as the pseudo-likelihood. So the formula is replaced by:

$$\min_{q(\Omega,\Theta) \in \mathcal{P}} \mathrm{KL}(q(\Omega,\Theta) \| q_t(\Omega,\Theta)) - \mathbb{E}_{q(\Omega)}[\mathrm{log}p(\mathbf{x}^{(t+1)}|\Omega)]$$
$$- \mathbb{E}_{q(\Omega,\Theta)}[\log(\varphi(\mathbf{y}^{(t+1)}|\tilde{\mathbf{z}}^{(t+1)}, \boldsymbol{\eta}))].$$

Similar to Eq.(7), we can get the posterior distribution:

$$q_{t+1}(\Omega,\Theta) = \frac{q_t(\Omega,\Theta)p(\mathbf{x}^{(t+1)}|\Omega)\varphi(\mathbf{y}^{(t+1)}|\tilde{\mathbf{z}}^{(t+1)}, \boldsymbol{\eta})}{\phi(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})},$$

where $\phi(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})$ is the normalization constant. Note that, the latent variable $\mathbf{z}^{(t)}$ is unrelated to the new posterior, because the variable $\mathbf{z}^{(t+1)}$'s prior is $p_0(\mathbf{z})$. Let $(\Omega, \Theta \backslash \mathbf{z}^{(t)})$ denote all variables in Ω and Θ except $\mathbf{z}^{(t)}$, then we can further get

$$q_{t+1}(\Omega,\Theta) = \frac{q_t(\Omega, \Theta \backslash \mathbf{z}^{(t)})p_0(\mathbf{z})p(\mathbf{x}^{(t+1)}|\Omega)}{\phi(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)})}$$
$$\cdot \varphi(\mathbf{y}^{(t+1)}|\tilde{\mathbf{z}}^{(t+1)}, \boldsymbol{\eta}). \quad (8)$$

In order to approximate $q_{t+1}(\Omega,\Theta)$ we use variational approximate inference which is introduced in Section 3.

3 Variational Inference

Because the posterior is intractable to compute, we apply the variational inference method[Beal, 2003] to approximate the posteriors in (7) for BM^{2}SMVL and in (8) for OBM^{2}SMVL. This method is much more efficient than sampling based methods [Gilks, 2005].

3.1 Data Augmentation

Since the pseudo-likelihood function $\varphi(\cdot)$ involves a max operater which is difficult and inefficient for posterior inference. We re-express the pseudo-likelihood function into the integration of a function with augmented variable based on the data augmentation idea [Polson and Scott, 2011]. For BM^{2}SMVL, we replace the pseudo-likelihood $\varphi(\cdot)$ with:

$$\varphi(y_c^{(n)}|\tilde{\mathbf{z}}^{(n)}, \boldsymbol{\eta}_c) =$$
$$\int_0^\infty \exp\{\frac{-1}{2\lambda_c^{(n)}}[\lambda_c^{(n)} + C(1 - y_c^{(n)}\boldsymbol{\eta}_c^T\tilde{\mathbf{z}}^{(n)})]^2\} \cdot \frac{d\lambda_c^{(n)}}{\sqrt{2\pi\lambda_c^{(n)}}}.$$

Then we can get

$$\varphi(\mathbf{Y},\boldsymbol{\lambda}|\mathbf{Z},\boldsymbol{\eta})=\prod_{n=1}^{N}\prod_{c=1}^{N_c}\frac{\exp\{\frac{-1}{2\lambda_c^{(n)}}[\lambda_c^{(n)}+C(1-y_n^c\boldsymbol{\eta}_c^T\tilde{\mathbf{z}}^{(n)})]^2\}}{\sqrt{2\pi\lambda_c^{(n)}}}.$$

Similarly, we introduce the augmented variable to the pseudo-likelihood function $\varphi(\cdot)$ for OBM^{2}SMVL

$$\begin{aligned}&\varphi(\mathbf{y}^{(t+1)},\boldsymbol{\lambda}^{(t+1)}|\mathbf{z}^{(t+1)},\boldsymbol{\eta})\\&=\prod_{c=1}^{N_c}\frac{\exp\{\frac{-1}{2\lambda_c^{(t+1)}}[\lambda_c^{(t+1)}+C(1-y_c^{(t+1)}\boldsymbol{\eta}_c^T\tilde{\mathbf{z}}^{(t+1)})]^2\}}{\sqrt{2\pi\lambda_c^{(t+1)}}}.\end{aligned}$$

3.2 Variational Approximate Inference

Next, we apply the mean-field variational method to approximating the posterior distributions.

Variational Inference in BM^{2}SMVL

Firstly, we define a family of factorized but free-form variational distributions:

$$V(\Omega,\Theta,\boldsymbol{\lambda})=V(\mu)V(\mathbf{W})V(\alpha)V(\phi)V(\mathbf{Z})V(\boldsymbol{\eta})V(\boldsymbol{\lambda})V(\nu).$$

The main idea of variational Bayesian inference is that we need to minimize the KL divergence $\mathrm{KL}(V(\Omega,\Theta,\boldsymbol{\lambda})\|q(\Omega,\Theta,\boldsymbol{\lambda}))$ between the approximating distribution and the target posterior. Next, we initialize the distributions of $V(\Omega,\Theta,\boldsymbol{\lambda})$. Then we iteratively update each parameter of our model by fix other parameters as the current estimates. Now, we give the joint distribution of data and parameters:

$$\begin{aligned}p(\Omega,\Theta,\boldsymbol{\lambda},\mathbf{X},\mathbf{Y})&=p_0(\mu)p(\mathbf{W}|\alpha)p_0(\alpha)p_0(\phi)p_0(\mathbf{Z})p(\boldsymbol{\eta}|\nu)\\&\cdot p_0(\nu)p(\mathbf{X}|\mu,\mathbf{W},\phi,\mathbf{Z})\varphi(\mathbf{Y},\boldsymbol{\lambda}|\mathbf{Z},\boldsymbol{\eta}).\end{aligned}$$

It can be shown that when keeping all other factors fixed the optimal distribution $V^*(\lambda)$ satisfies

$$V^*(\boldsymbol{\lambda})\propto\exp\{\mathbb{E}_{-\boldsymbol{\lambda}}[\log p(\Omega,\Theta,\boldsymbol{\lambda},\mathbf{X},\mathbf{Y})]\},$$

where $\mathbb{E}_{-\boldsymbol{\lambda}}$ denotes the expectation with respect to $V(\Omega,\Theta,\boldsymbol{\lambda})$ over all variables except for $\boldsymbol{\lambda}$. Then we can get the updating formula for $\mathbb{E}_{-\boldsymbol{\lambda}}$:

$$\begin{aligned}V^*(\boldsymbol{\lambda})&=\prod_{c=1}^{N_c}\prod_{n=1}^{N}\mathcal{GIG}(\lambda_c^{(n)}|\frac{1}{2},1,\chi_c^{(n)})\\\chi_c^{(n)}&=C^2\langle(1-y_c^{(n)}\boldsymbol{\eta}_c^T\tilde{\mathbf{z}}^{(n)})^2\rangle,\end{aligned}$$

where $\langle\cdot\rangle$ represents the expectation, $\mathcal{GIG}(\cdot)$ is the generalized inverse Gaussian distribution. Similarly, we can get the updating formulas for all other factors. Since they are tedious and easy to derive, here we only provide the equations for $\mathbf{Z}$, other updating formulas are omitted because of the limited space of the paper,

$$\begin{aligned}V^*(\mathbf{Z})&=\prod_{n=1}^{N}\mathcal{N}(\mathbf{z}^{(n)}|\mu_{\mathbf{z}}^{(n)},\Sigma_{\mathbf{z}}^{(n)})\\\Sigma_{\mathbf{z}}^{(n)}&=\{C^2\sum_{c=1}^{N_c}\langle\tilde{\boldsymbol{\eta}}_c\tilde{\boldsymbol{\eta}}_c{}^T\rangle\langle(\lambda_c^{(n)})^{-1}\rangle+\mathbf{I}_m\\&+\sum_{i=1}^{N_v}\langle\phi_i\rangle\langle\mathbf{W}_{\mathrm{i}}{}^T\mathbf{W}_{\mathrm{i}}\rangle\}^{-1}\\\mu_{\mathbf{z}}^{(n)}&=\Sigma_{\mathbf{z}}^{(n)}\{\sum_{i=1}^{N_v}\langle\phi_i\rangle\langle\mathbf{W}_{\mathrm{i}}{}^T\rangle(\mathbf{x}_i^{(n)}-\langle\mu_{\mathbf{i}}\rangle)\\&+\sum_{i=1}^{N_c}\{C(1+C\langle(\lambda_c^{(n)})^{-1}\rangle)y_c^{(n)}\langle\tilde{\boldsymbol{\eta}}_c\rangle\\&-C^2\langle(\lambda_c^{(n)})^{-1}\rangle\langle\eta_{c,(m+1)}\tilde{\boldsymbol{\eta}}_c\rangle\}\},\end{aligned}$$

where $\tilde{\boldsymbol{\eta}}_c$ denotes the first m dimensions of $\boldsymbol{\eta}_c$, i.e., $\boldsymbol{\eta}_c=[\tilde{\boldsymbol{\eta}}_c,\ \eta_{c,(m+1)}]$.

Variational Inference in OBM^{2}SMVL

Now, we use variational inference to approximate $q_{t+1}(\Omega,\Theta)$ in OBM^{2}SMVL model. Firstly, we give the joint distribution of data and parameters:

$$\begin{aligned}&p(\Omega,\Theta,\boldsymbol{\lambda}^{(t+1)},\mathbf{x}^{(t+1)},\mathbf{y}^{(t+1)})=p_0(\mu)p(\mathbf{W}|\alpha)p_0(\alpha)p_0(\phi)\\&\cdot_0(\mathbf{z})p(\boldsymbol{\eta}|\nu)\,p_0(\nu)p(\mathbf{x}^{(t+1)}|\mu,\mathbf{W},\phi,\mathbf{z})\varphi(\mathbf{y}^{(t+1)},\boldsymbol{\lambda}^{(t+1)}|\mathbf{z},\boldsymbol{\eta}).\end{aligned}$$

It can be shown that when keeping all other factors fixed, the optimal distribution $V^*(\boldsymbol{\lambda}^{(t+1)})$ satisfies

$$V^*(\boldsymbol{\lambda}^{(t+1)})\propto\exp\{\mathbb{E}_{-\boldsymbol{\lambda}^{(t+1)}}[\log p(\Omega,\Theta,\boldsymbol{\lambda}^{(t+1)},\mathbf{x}^{(t+1)},\mathbf{y}^{(t+1)})]\},$$

where $\mathbb{E}_{-\boldsymbol{\lambda}^{(t+1)}}$ denotes the expectation with respect to $V(\Omega,\Theta,\boldsymbol{\lambda}^{(t+1)})$ over all variables except for $\boldsymbol{\lambda}^{(t+1)}$. Then we can get the updating formula for $\mathbb{E}_{-\boldsymbol{\lambda}^{(t+1)}}$:

$$\begin{aligned}V^*(\boldsymbol{\lambda}^{(t+1)})&=\prod_{c=1}^{N_c}\mathcal{GIG}(\lambda_c^{(t+1)}|\frac{1}{2},1,\chi_c^{(t+1)})\\\chi_c^{(t+1)}&=C^2\langle(1-y_c^{(t+1)}\boldsymbol{\eta}_c^T\tilde{\mathbf{z}}^{(t+1)})^2\rangle.\end{aligned}$$

Similarly, we can get the updating formulas for all other factors. Since they are tedious and easy to derive, here we only provide the equations for $\mathbf{z}^{(\mathrm{t}+1)}$, other updating formulas are omitted because of the limited space of the paper,

$$\begin{aligned}V^*(\mathbf{z}^{(t+1)})&=\mathcal{N}(\mathbf{z}^{(t+1)}|\mu_{\mathbf{z}}^{(t+1)},\Sigma_{\mathbf{z}}^{(t+1)})\\\Sigma_{\mathbf{z}}^{(t+1)}&=\{C^2\sum_{c=1}^{N_c}\langle\tilde{\boldsymbol{\eta}}_c(\tilde{\boldsymbol{\eta}}_c)^T\rangle\langle(\lambda_c^{(t+1)})^{-1}\rangle+\mathbf{I}_m\\&+\sum_{i=1}^{N_v}\langle\phi_i\rangle\langle(\mathbf{W}_{\mathrm{i}})^T\mathbf{W}_{\mathrm{i}}\rangle\}^{-1}\\\mu_{\mathbf{z}}^{(t+1)}&=\Sigma_{\mathbf{z}}^{(t+1)}\{\sum_{i=1}^{N_v}\langle\phi_i\rangle\langle(\mathbf{W}_{\mathrm{i}})^T\rangle(\mathbf{x}_i^{(t+1)}-\langle\mu_{\mathbf{i}}\rangle)\\&+\sum_{i=1}^{N_c}\{C(1+C\langle(\lambda_c^{(t+1)})^{-1}\rangle)y_c^{(t+1)}\langle\tilde{\boldsymbol{\eta}}_c\rangle\\&-C^2\langle(\lambda_c^{(t+1)})^{-1}\rangle\langle\eta_{c,(m+1)}\tilde{\boldsymbol{\eta}}_c\rangle\}\},\end{aligned}$$

where $\tilde{\boldsymbol{\eta}}_c$ denotes the first m dimensions of $\boldsymbol{\eta}_c$, i.e., $\boldsymbol{\eta}_c=[\tilde{\boldsymbol{\eta}}_c,\ \eta_{c,(m+1)}]$.

3.3 Computational Complexity

For each iteration of parameter updating in our batch learning BM^{2}SMVL, we need $O(NN_v\bar{d}m^2)$ computation, where $\bar{d}$ is the average dimension of all N_v views. The most computation is spent on the calculation of $\Sigma_{\mathbf{z}}^{(n)}$, $n = 1, \cdots, N$ where the matrix multiplication $\langle \mathbf{W}_{\rm i}{}^T\mathbf{W}_{\rm i}\rangle$ consumes $d_i m^2$ computation. And each iteration of parameter updating in our online learning OBM^{2}SMVL consumes $O(N_v\bar{d}m^2)$ when a new sample is coming.

4 Experiments

We evaluate the proposed batch learning model BM^{2}SMVL and online learning model OBM^{2}SMVL on various classification tasks including image data and text data.

4.1 Real Data Sets

There are four data sets, i.e., Tervid, Washington, Cornell and News4Gv, used in our experiments. Trecvid contains 1,078 manually labeled video shots that belong to five categories [Chen *et al.*, 2012]. And each shot is represented by a 1,894-dim binary vector of text features and a 165-dim vector of HSV color histogram. WebKB data set has two views, including the content features of the web pages and the link features exploited from the link structures. This data set consists of 877 web pages from computer science departments in four universities, i.e., Cornell, Washington, Wisconsin and Texas. And each university has five document classes, i.e., course, faculty, student, project and staff. We select the web pages from Cornell and Washington as our experimental data[1]. These two data sets have five classes with two views. 20Newsgroups data set is widely used for classification. This data set has approximately 20,000 newsgroup documents, which are divided into 20 categories. We follow the way in [Long *et al.*, 2008] to construct multi-view learning problems. We use the tf-idf weighting scheme to represent the document, and the document frequency with the value of 5 is adopted to cut down the number of word features. The details of these data sets are shown in Table 1.

Table 1: Statistics of the multiclass data sets.

Datasets	Trecvid	Washington	Cornell	News4Gv
size	1078	230	195	1500
class	5	5	5	3
V1-Dim	1894	1703	1703	6783
V2-Dim	165	230	195	6307
V3-Dim	-	-	-	7717
V4-Dim	-	-	-	9336

4.2 Competitors

We compare our model with five competitors:

- VMRML [Quang *et al.*, 2013]: it is a vector-valued manifold regularization multi-view learning. The regularization parameters are set as the default value in their paper, and we tune the parameter σ for 'rbf' carefully in each data set;
- MVMED [Sun and Chao, 2013]: it presents a multi-view maximum entropy discrimination model. We use the model with one-VS-rest strategy for multiclass problem. According to the paper, we choose the best parameter from $2^{[-5:5]}$ by executing 5-fold cross-validation for each data set;
- MMH [Chen *et al.*, 2012]: it is a large-margin predictive latent subspace learning for multi-view data. Based on the parameters given in its code[2], we tune the four paramters carefully to choose the best parameters for each data set;
- SVM-FULL: it concatenates all views to form a new single view, and applies SVM for classification. We choose the linear kernel and execute 5-fold cross-validation on training sets to decide the cost parameter c from $10^{[-3:3]}$;
- OPMV [Zhu *et al.*, 2015]: it is an online multi-view learning. According to the paper, the learning rate parameter are chose from $2^{[-8:8]}$, the regularization parameter are chose from $1e^{[-16:0]}$, and the penalty parameters is pre-defined as 1. The parameters are set according to the above rules.

Table 2: Batch learning comparison on multiclass data sets. Listed results are test accuracies (%) averaged over 20 independent runs. Bold face indicates highest accuracy.

	Trecvid	Washington	Cornell	News4Gv
MMH	61.22 ± 0.0	80.98 ± 2.94	74.01 ± 0.19	-
MVMED	63.80 ± 0.0	73.86 ± 2.78	72.27 ± 2.97	94.26 ± 0.83
VMRML	63.27 ± 0.0	79.44 ± 2.61	76.96 ± 4.03	93.34 ± 0.98
SVM-FULL	62.34 ± 0.0	82.91 ± 3.33	76.14 ± 2.40	**99.21 ± 0.30**
BM^{2}SMVL	**65.86 ± 0.0**	**83.48 ± 3.03**	**78.87 ± 3.63**	97.99 ± 0.57

Table 3: Online learning comparison on multiclass data sets. Listed results are test accuracies (%) averaged over 20 independent runs. Bold face indicates highest accuracy.

	Trecvid	Washington	Cornell	News4Gv
OPMV	61.41 ± 0.0	73.44 ± 2.06	66.40 ± 3.83	-
OBM^{2}SMVL	**63.27 ± 0.0**	**77.96 ± 3.87**	**74.18 ± 4.63**	**96.03 ± 0.66**

4.3 Parameter Setting

In our batch learning, the regularization parameter C is chosen from the integer set $\{1, 2, 3\}$ and the subspace dimension m from the integer set $\{20, 30, 50\}$ for each data set by performing 5-fold cross validation on training data. While in our online learning, the regularization parameter C is chosen from the integer set $\{1, 5, 15\}$ and the subspace dimension m from the integer set $\{20, 30, 50\}$. For the rest parameters, both our batch and online learning are set as the same, i.e., $a_\alpha = b_\alpha =$ 1e-3, $a_\phi =$ 1e-2, $a_\nu =$ 1e-1, $b_\phi = b_\nu = \beta =$ 1e-5.

4.4 Experimental Results

Since a normal prior with zero mean is imposed on the observation data, we normalize the observation data to have zero

[1] http://www-2.cs.cmu.edu/ webkb/

[2] http://bigml.cs.tsinghua.edu.cn/ ningchen/MMH.htm

mean and unit variance. In batch learning experiments, we use the same training/testing split of the Trecvid data set as in [Chen *et al.*, 2012]. So there is only one result in this data set. For other data sets the results of all models are averaged over 20 independent runs and. All the results are shown in Table 2. The ratio sampled for training data is 0.5 in the three data set Trecvid, Washington and Cornell, and 0.05 in News4Gv. Since MMH can not address high dimensional data, e.g., News4Gv, so its result is missing for News4Gv in Table 2.

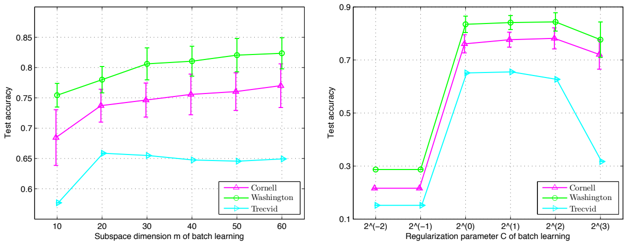
In online learning experiments, we use the same training/testing split of the above batch learning experiments. We sample 0.1 of the training data as the batch training, and the rest come one by one. Since OPMV can only deal with two-view data, so its result is missing for News4Gv in Table 3. From Table 2 and Table 3, we have the following insightful observations:

- Our BM^{2}SMVL achieves the best performance on the Trecvid, Washington and Cornell data sets and performs just a little worse than the SVM-FULL in the News4Gv data. We attribute it to that our method can automatically infer the penalty parameter of max-margin model based on the data augmentation idea, while MVMED and MMH are both under the maximum entropy discrimination framework and cannot infer the penalty parameter. SVM-FULL makes full use of all the information from the observations by concatenating all views to form a new single view. This maybe the reason why it performs better than our BM^{2}SMVL in the News4Gv. But some information from the observations is not helpful for the classification in other data sets. In this case, SVM-FULL cannot achieve a good performance.
- Our method infers a posterior under the Bayesian framework instead of a point estimate as in VMRML. With Baysian model averaging over the posterior, we can make more robust predictions than VMRML.
- We also find that OBM^{2}SMVL performs better than OPMV on all data sets and just a little worse than BM^{2}SMVL. Unlike OPMV, which seeks a point estimate by optimizing some deterministic objective function, our online model infers a posterior under the Bayesian framework. The point estimate can be affected seriously by inappropriate regularization, outliers and noises, especially when the training data arrive sequentially.

4.5 Sensitivity Analysis

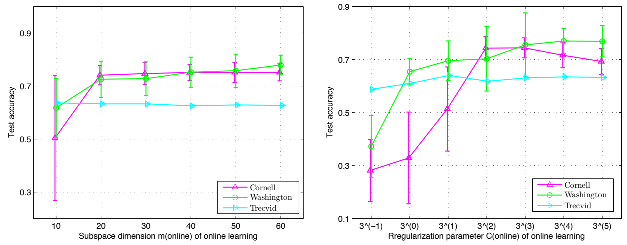
We study the sensitivity of BM^{2}SMVL and OBM^{2}SMVL with respect to the subspace dimension m, and the regularization parameter C.

When we study the influence of m, C (batch) is set as 2 for BM^{2}SMVL and C (online) is set as 15 for OBM^{2}SMVL. The averaged results are shown in Figure 1 (a) and Figure 2 (a). We find that the test accuracy increases when m becomes larger. And when m is large enough, the test accuracy remains stable.



(a) m (batch) (b) C (batch)

Figure 1: (a) Results on different data sets with different parameters m in BM^{2}SMVL; (b) Results on different data sets with different regularization parameters C in BM^{2}SMVL.



(a) m (online) (b) C (online)

Figure 2: (a) Results on different data sets with different subspace dimensions m (online) in OBM^{2}SMVL; (b) Results on different data sets with different regularization parameters C (online) in OBM^{2}SMVL.

When we study the influence of C, m is set as 30 for both batch and online learning. From the results in Figure 1 (b) and Figure 2 (b) , we can find that different data sets may prefer different values of C. In batch learning, C (batch) balances the classification model and subspace learning model, so our model cannot get the best performance when C (batch) is too large or too small. C (online) reflects the importance of new arrival data in our online model. When C (online) is too small, the new arrival data plays a tiny role in the online model and offers little help to improve the performance of our online model. For some data sets like Cornell, when C (online) is too large, the performance of OBM^{2}SMVL would become bad because the online model doesn't take full advantage of the historical knowledge. For some other data sets like Trecvid and Washington, they are less sensitive to C (online) when C (online) is large enough.

5 Conclusion

We propose an online Bayesian method to learn predictive subspace for multi-view data. Specifically, the proposed method is based on the data augmentation idea for max-margin learning, which allows us to automatically infer the weight and penalty parameter and find the most appropriate predictive subspace simultaneously under the Bayesian framework. Experiments on various classification tasks show that both our batch model BM^{2}SMVL and online model OBM^{2}SMVL can achieve superior performance, compared with a number of state-of-the-art competitors.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 9154610306, 61573335, 61473273, 61473274), National High-tech R&D Program of China (863 Program) (No.2014AA015105), Guangdong provincial science and technology plan projects (No. 2015 B010109005).

References

[Beal, 2003] Matthew James. Beal. Variational algorithms for approximate bayesian inference. *University College London*, 2003.

[Blum and Mitchell, 1998] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of The Eleventh Annual Conference on Computational learning theory*, pages 92–100, 1998.

[Cesa-Bianchi and Lugosi, 2006] Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.

[Chechik *et al.*, 2010] Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. Large scale online learning of image similarity through ranking. *The Journal of Machine Learning Research*, 11:1109–1135, 2010.

[Chen *et al.*, 2012] Ning Chen, Jun Zhu, Fuchun Sun, and Eric Poe Xing. Large-margin predictive latent subspace learning for multiview data analysis. *Pattern Analysis and Machine Intelligence*, 34(12):2365–2378, 2012.

[Chiang *et al.*, 2008] David Chiang, Yuval Marton, and Philip Resnik. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 224–233, 2008.

[Crammer *et al.*, 2006] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.

[Du *et al.*, 2015] Changying Du, Shandian Zhe, Fuzhen Zhuang, Yuan Qi, Qing He, and Zhongzhi Shi. Bayesian maximum margin principal component analysis. In *The Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[Gilks, 2005] Walter R Gilks. *Markov chain monte carlo*. Wiley Online Library, 2005.

[Gönen and Alpaydın, 2011] Mehmet Gönen and Ethem Alpaydın. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011.

[Grangier and Bengio, 2008] David Grangier and Samy Bengio. A discriminative kernel-based approach to rank images from text queries. *Pattern Analysis and Machine Intelligence*, 30(8):1371–1384, 2008.

[Guo and Xiao, 2012] Yuhong Guo and Min Xiao. Cross language text classification via subspace co-regularized multi-view learning. *Computer Science - Computation and Language*, 2012.

[Hardoon *et al.*, 2004] David R Hardoon, Sandor Szedmak, and John Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, 2004.

[Hazan *et al.*, 2007] Elad Hazan, Amit Agarwal, and Satyen Kale. Logarithmic regret algorithms for online convex optimization. *Machine Learning*, 69(2-3):169–192, 2007.

[Long *et al.*, 2008] Bo Long, Philip S. Yu, and Zhongfei Zhang. A general model for multiple view unsupervised learning. *SIAM International Conference on Data Mining*, pages 822–833, 2008.

[Polson and Scott, 2011] Nicholas G. Polson and Steven L. Scott. Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):43–47, 2011.

[Quang *et al.*, 2013] Minh H Quang, Loris Bazzani, and Vittorio Murino. A unifying framework for vector-valued manifold regularization and multi-view learning. In *International Conference on Machine Learning*, pages 100–108, 2013.

[Reents and Urbanczik, 1998] G. Reents and R. Urbanczik. Self-averaging and on-line learning. *Physical Review Letters*, 80(24):5448, 1998.

[Rosenblatt, 1958] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386, 1958.

[Sharma *et al.*, 2012] Abhishek Sharma, Abhishek Kumar, Hal Daume III, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167, 2012.

[Shi and Zhu, 2013] Tianlin Shi and Jun Zhu. Online bayesian passive-aggressive learning. In *International Conference on Machine Learning*, pages 378–386, 2013.

[Sun and Chao, 2013] Shiliang Sun and Guoqing Chao. Multi-view maximum entropy discrimination. In *International Joint Conference on Artificial Intelligence*, pages 1706–1712, 2013.

[Yarowsky, 1995] David Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of The 33rd Annual Meeting on Association for Computational Linguistics*, pages 189–196, 1995.

[Ye *et al.*, 2012] Guangnan Ye, Dong Liu, I-Hong Jhuo, Shih-Fu Chang, et al. Robust late fusion with rank minimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3021–3028, 2012.

[Zhu *et al.*, 2015] Yue Zhu, Wei Gao, and Zhi-Hua Zhou. One-pass multi-view learning. In *Asian Conference on Machine Learning*, pages 407–422, 2015.