Nonlinear Maximum Margin Multi-view Learning with Adaptive Kernel

Jia He[1,4], **Changying Du**[2], **Changde Du**[3,4], **Fuzhen Zhuang**[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]Lab of Parallel Software and Computational Science, Institute of Software, CAS, Beijing, China
[3]Research Center for Brain-Inspired Intelligence, Institute of Automation, CAS, Beijing, China
[4]University of Chinese Academy of Sciences, Beijing 100049, China
{hej, zhuangfz, heq}@ics.ict.ac.cn, {changying, guoping}@iscas.ac.cn, duchangde2016@ia.ac.cn

Abstract

Existing multi-view learning methods based on kernel function either require the user to select and tune a single predefined kernel or have to compute and store many Gram matrices to perform multiple kernel learning. Apart from the huge consumption of manpower, computation and memory resources, most of these models seek point estimation of their parameters, and are prone to overfitting to small training data. This paper presents an adaptive kernel nonlinear max-margin multi-view learning model under the Bayesian framework. Specifically, we regularize the posterior of an efficient multi-view latent variable model by explicitly mapping the latent representations extracted from multiple data views to a random Fourier feature space where max-margin classification constraints are imposed. Assuming these random features are drawn from Dirichlet process Gaussian mixtures, we can adaptively learn shift-invariant kernels from data according to Bochners theorem. For inference, we employ the data augmentation idea for hinge loss, and design an efficient gradient-based MCMC sampler in the augmented space. Having no need to compute the Gram matrix, our algorithm scales linearly with the size of training set. Extensive experiments on real-world datasets demonstrate that our method has superior performance.

1 Introduction

Nowadays data typically can be collected from various information channels, e.g., a piece of news is consisted of text, audio, video clip and hyperlink. How to effectively and efficiently fuse the information from different channels for specific learning tasks is a problem (known as multi-view learning) attracting more and more attention. For supervised learning, e.g., news classification, we have not only these multi-channel features but also the corresponding category labels. Making full use of the label information usually is critical for the construction of predictive multi-view models since it can help to learn more discriminative features for classification. A popular choice is to exploit the maximum margin principle to guarantee the learned model to have a good generalization ability [Xu *et al.*, 2014; He *et al.*, 2016]. Though improved performance is reported on many problems, these methods have made linearity assumption on the data, which may be inappropriate for multi-view data revealing nonlinearities.

Kernel method [Hofmann *et al.*, 2007] is a principled way for introducing nonlinearity into linear models, and a kernel machine can approximate any function or decision boundary arbitrarily well by tuning its kernel parameter. Along this line, many single kernel multi-view learning methods have been proposed [Farquhar *et al.*, 2005; Szedmak and Shawe-Taylor, 2007; Fang and Zhang, 2012; Sun and Chao, 2013; Quang *et al.*, 2013; Xu *et al.*, 2014], which typically require the user to select and tune a predefined kernel for every view. Choosing an appropriate kernel for real-world situations is usually not easy for users without enough domain knowledge. The performance of these models may be greatly affected by the choice of kernel. An alternative solution to resolve this problem is provided by multiple kernel learning (MKL), which can predefine different kernels for each data view and then integrate the kernels by algorithms such as semi-definite programming (SDP) [Lanckriet *et al.*, 2004], semi-infinite linear programming (SILP) [Sonnenburg *et al.*, 2006], and simple MKL [Rakotomamonjy *et al.*, 2008]. However, MKL models inherently have to compute and store many Gram matrices to get good performance while computing a Gram matrix for a data set with N instances and D features needs $O(N^2D)$ operations and storing too many Gram matrices often leads to out of memory on commonly used computers. Besides, all the aforementioned kernelized models seek single point estimation of their parameters, thus are prone to overfitting to small training data. Under the Bayesian framework, BEMKL [Gonen, 2012] is a state-of-the-art variational method that estimates the entire posterior distribution of model weights. Unfortunately, BEMKL has to perform time-consuming matrix inversions to compute the posterior covariances of sample and kernel weights. Moreover, BEMKL's performance may be limited by its mean-field assumption on the approximate posterior and the absence of max-margin principle.

To address the aforementioned problems, we propose an adaptive kernel maximum margin multi-view learning (M^3L) model with low computational complexity. Specifically, we

firstly propose an efficient multi-view latent variable model (LVM) based on the traditional Bayesian Canonical Correlation Analysis (BCCA) [Wang, 2007], which learns the shared latent representations for all views. To adaptively learn the kernel, we introduce the random Fourier features which constructs an approximate primal space to estimate kernel evaluations $K(x, x')$ as the dot product of finite vectors $\varphi(x)^T\varphi(x')$ [Rahimi and Recht, 2007]. Assuming these random features are drawn from Dirichlet process (DP) Gaussian mixtures, we can adaptively learn shift-invariant kernels from data according to Bochners theorem. With such a stochastic random frequency distribution, it is more general than the traditional kernel method approach with a fixed kernel. Secondly, to make full use of the label information, we impose max-margin classification constraints on the explicit expressions $\varphi(x)$ in random Fourier feature space. Next, we regularize the posterior of the efficient multi-view LVM by explicitly mapping the latent representations to a random Fourier feature space where max-margin classification constraints are imposed. Our model is based on the data augmentation idea for max-margin learning in the Bayesian framework, which allows us automatically infer parameters of adaptive kernel and the penalty parameter of max-margin learning model. For inference, we devise a hybrid Markov chain Monte Carlo (MCMC) sampler. To infer random frequencies, we use an effective distributed DP mixture models (DPMM) [Ge *et al.*, 2015]. Moreover, some key parameters don't have corresponding conjugate priors. So we adopt the gradient-based MCMC sampler Hamiltonian Monte Carlo (HMC) [Neal, 2011] for faster convergence. The computational complexity of our algorithm is linear w.r.t. the number of instances N. Extensive experiments on real-world datasets demonstrate our method has a superior performance, compared with a number of competitors.

2 Model

BCCA [Wang, 2007] assumes the following generative process to learn the shared latent representations from multiple views $\{\mathbf{x}_i\}_{i=1}^{N_v}$, where N_v is the number of views:

$$\begin{aligned}\mathbf{h} &\sim \mathcal{N}(\mathbf{0},\, \mathbf{I})\\ \mathbf{x}_i &\sim \mathcal{N}(\mathbf{W}_i\mathbf{h}, \mathbf{\Psi}_i),\end{aligned}$$

where $\mathcal{N}(\cdot)$ denotes the normal distribution, $\mathbf{h} \in \mathbb{R}^m$ is the shared latent variable, $\mathbf{W}_i \in \mathbb{R}^{D_i\times m}$ is the linear transformation and $\mathbf{\Psi}_i \in \mathbb{R}^{D_i\times D_i}$ denotes covariance matrix. A commonly used prior is the Wishart distribution for $\mathbf{\Psi}_i^{-1}$.

2.1 An Efficient Multi-view LVM

However, BCCA has to perform time-consuming inversions of the high-dimension covariance matrix $\mathbf{\Psi}_i$ which could result in a severe computational problem. To solve the problem, we propose an efficient model by introducing additional latent variables. We assume the following generative process:

$$\begin{aligned}\mathbf{h},\, \mathbf{u}_i &\sim \mathcal{N}(\mathbf{0},\, \mathbf{I})\\ \mathbf{x}_i &\sim \mathcal{N}(\mathbf{W}_i\mathbf{h} + \mathbf{V}_i\mathbf{u}_i, \tau_i^{-1}\mathbf{I}),\end{aligned}$$

where $\mathbf{u}_i \in \mathbb{R}^{K_i}$ is the additional latent variable. Gamma prior can be used for τ_i and popular automatic relevance determination (ARD) prior can be imposed on projection matrices $\mathbf{W}_i$, $\mathbf{V}_i \in \mathbb{R}^{D_i\times K_i}$, i.e.,

$$\begin{aligned}\mathbf{r}_i &\sim \textstyle\prod_{j=1}^{m}\Gamma(r_{ij}|a_r, b_r)\\ \mathbf{W}_i &\sim \textstyle\prod_{j=1}^{m}\mathcal{N}(\mathbf{w}_{i,\cdot j}|\mathbf{0}, r_{ij}^{-1}\mathbf{I})\\ \mathbf{V}_i &\sim \textstyle\prod_{j=1}^{K_i}\mathcal{N}(\mathbf{v}_{i,\cdot j}|\mathbf{0},\, \eta^{-1}\mathbf{I})\\ \tau_i &\sim \Gamma(\tau_i|a_\tau, b_\tau),\end{aligned}$$

where $i = 1, \cdots, N_v$, $\Gamma(\cdot)$ denotes Gamma distribution, $\mathbf{w}_{i,\cdot j}$ represents the j-th column of the transformation matrix $\mathbf{W}_i$ and $\mathbf{v}_{i,\cdot j}$ represents the j-th column of $\mathbf{V}_i$. This model can be shown to be equivalent to imposing a low-rank assumption $\mathbf{\Psi}_i = \mathbf{V}_i\mathbf{V}_i^{\mathrm{T}} + \tau_i^{-1}\mathbf{I}$ for the covariances, which allows decreasing the computational complexity.

We define that the data matrix of the i-th view is $\mathbf{X}_i \in \mathbb{R}^{D_i\times N}$ consisting of N observations $\{\mathbf{x}_i^n\}_{n=1}^N$, $\widetilde{\mathbf{X}} = \{\mathbf{X}_i\}_{i=1}^{N_v}$, $\mathbf{H} = \{\mathbf{h}^n\}_{n=1}^N$ and $\mathbf{U}_i = \{\mathbf{u}_i^n\}_{n=1}^N$. For simplicity, let $\Omega = (\mathbf{r}_i, \mathbf{V}_i, \mathbf{W}_i, \eta, \tau_i, \mathbf{H}, \mathbf{U}_i)$ be the parameters of the multi-view LVM and $p_0(\Omega)$ be the prior of Ω. We can verify that the Bayesian posterior distribution $p(\Omega|\widetilde{\mathbf{X}}) = p_0(\Omega)p(\widetilde{\mathbf{X}}|\Omega)/p(\widetilde{\mathbf{X}})$ can be equivalently obtained by solving the following optimization problem:

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

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

2.2 Adaptive Kernel M^{3}L

From the description above, we can see that the multi-view LVM is an unsupervised model which learns the shared latent variables from the observations without using any label information. In general, we prefer that the shared latent representation can not only explain the observed data well but also help to learn a predictive model, which predicts the responses of new observations as accurate as possible.

Moreover, this multi-view LVM is a linear multi-view representation learning algorithm, but multi-view data usually reveal nonlinearities in many scenarios of real-world. As is well known, kernel methods are attractive because they can approximate any function or decision boundary arbitrarily well. So in this section we propose an adaptive kernel max-margin multi-view learning (M^{3}L) model. With the method named random features for the approximation of kernels [Rahimi and Recht, 2007], we can get explicit expression of the latent variable $\mathbf{h}$ in random feature space. Then we can classify these explicit expression linearly by introducing the max-margin principle which has good generalization performance. To incorporate the nonlinear max-margin method to the unsupervised multi-view LVM, we adopt the posterior regularization strategy [Jaakkola *et al.*, 1999; Zhu *et al.*, 2012]. Suppose we have a $1 \times N$ label vector $\mathbf{y}$ with its element $y^n \in \{+1, -1\}$, $n = 1, \cdots, N$. Then

we define the following pseudo-likelihood function of latent representation $\mathbf{h}^n$ for the n-th observation $\{\mathbf{x}_i^n\}_{i=1}^{N_v}$:

$$\ell(y^n|\tilde{\varphi}(\mathbf{h}^n),\boldsymbol{\beta}) = \exp\{-2C\cdot\max(0,1-y^n\boldsymbol{\beta}^{\mathrm{T}}\tilde{\varphi}(\mathbf{h}^n))\},$$
$$\varphi(\mathbf{h}^n) = \frac{1}{\sqrt{M}}[\cos(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{h}^n),\cdots,\cos(\boldsymbol{\omega}_M^{\mathrm{T}}\mathbf{h}^n), \sin(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{h}^n),\cdots,\sin(\boldsymbol{\omega}_M^{\mathrm{T}}\mathbf{h}^n)]^{\mathrm{T}},$$

where C is the regularization parameter, $\tilde{\varphi}(\mathbf{h}^n)$ is the explicit expression of the latent variable $\mathbf{h}^n$ in random Fourier feature space and $\tilde{\varphi}(\mathbf{h}^n) = (\varphi(\mathbf{h}^n)^{\mathrm{T}},1)^{\mathrm{T}}$. $\boldsymbol{\omega}_i \in \mathbb{R}^m$ denotes random frequency vector and $\boldsymbol{\beta}^{\mathrm{T}}\tilde{\varphi}(\mathbf{h}^n)$ is a discrimination function parameterized by $\boldsymbol{\beta} \in \mathbb{R}^{2M+1}$.

Bochners theorem states that a continuous shift-invariant kernel $K(\mathbf{h},\bar{\mathbf{h}}) = k(\mathbf{h}-\bar{\mathbf{h}})$ is a positive definite function if and only if $k(t)$ is the Fourier transform of a non-negative measure $\rho(\boldsymbol{\omega})$ [Rudin, 2011]. Further, we note that if $k(0) = 1$, $\rho(\boldsymbol{\omega})$ will be a normalized density. So we can get

$$\begin{aligned} k(\mathbf{h}-\bar{\mathbf{h}}) &= \int_{\mathbb{R}^m}\rho(\boldsymbol{\omega})\exp(i\boldsymbol{\omega}^{\mathrm{T}}(\mathbf{h}-\bar{\mathbf{h}}))d\boldsymbol{\omega} \\ &= \mathbb{E}_{\boldsymbol{\omega}\sim\rho}[\exp(i\boldsymbol{\omega}^{\mathrm{T}}\mathbf{h})\exp(i\boldsymbol{\omega}^{\mathrm{T}}\bar{\mathbf{h}})^*] \\ &\approx \frac{1}{M}\sum_{j=1}^{M}\exp(i\boldsymbol{\omega}_j^{\mathrm{T}}\mathbf{h})\exp(i\boldsymbol{\omega}_j^{\mathrm{T}}\bar{\mathbf{h}})^*. \end{aligned}$$

If the kernel k is real-valued, we can discard the imaginary part:

$$k(\mathbf{h}-\bar{\mathbf{h}}) \approx \varphi(\mathbf{h})^{\mathrm{T}}\varphi(\bar{\mathbf{h}})$$
$$\varphi(\mathbf{h}) \equiv \frac{1}{\sqrt{M}}[\cos(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{h}),\cdots,\cos(\boldsymbol{\omega}_M^{\mathrm{T}}\mathbf{h}), \sin(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{h}),\cdots,\sin(\boldsymbol{\omega}_M^{\mathrm{T}}\mathbf{h})]^{\mathrm{T}}.$$

A robust and flexible choice of $\rho(\boldsymbol{\omega})$ is a Gaussian mixture model. Mixture models based on DPs treat the number of represented mixture components as a latent variable, and infer it automatically from observed data. DP Gaussian mixture prior is widely used for density estimation [Oliva *et al.*, 2016]. Assuming these random features are drawn from DP Gaussian mixtures, we can adaptively learn shift-invariant kernels from data according to Bochners theorem. We impose DP Gaussian mixture prior for the variables $\boldsymbol{\omega}_j, j = 1,\cdots,M$. Suppose the DP has base distribution G_0 and concentration parameter α, then we have

$$\zeta_k \sim G_0,\ \ \nu_k \sim \text{Beta}(1,\alpha),\ \ \varpi_k = \nu_k\prod_{i=1}^{k-1}(1-\nu_i)$$
$$z_j \sim \text{Cat}(\boldsymbol{\varpi}),\quad \boldsymbol{\omega}_j \sim \mathcal{N}(\zeta_{z_j}),$$

where $k = 1,\cdots,\infty$ and $\zeta_k = (\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$ contains the mean and covariance parameters of the k-th Gaussian component. Popular choice would be Normal-Inverse-Wishart prior G_0 for the mixture components:

$$\boldsymbol{\Sigma}_k \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}_0,\nu_0),\quad \boldsymbol{\mu}_k \sim \mathcal{N}(\boldsymbol{\mu}_0,\frac{1}{\kappa_0}\boldsymbol{\Sigma}_k),$$

where $\mathcal{W}^{-1}(\cdot)$ denotes Inverse-Wishart distribution. Next, we impose prior on $\boldsymbol{\beta}$ as the following form

$$p(\boldsymbol{\beta}|v) \quad\sim\quad \mathcal{N}(\boldsymbol{\beta}|\mathbf{0},v^{-1}\mathbf{I}_{(2M+1)}),$$

where a_v and b_v are hyper-parameters and v plays a similar role as the penalty parameter in SVM.

For simplicity, let $\Theta = (\boldsymbol{\beta},v,\nu_k,\varpi_k,\zeta_k,z_i,\boldsymbol{\omega}_i,\boldsymbol{\mu}_k,\boldsymbol{\Sigma}_k)$ be the variables of the nonlinear max-margin prediction model. Now, we can formulate our final modal as

$$\min_{q(\Omega,\Theta)\in\mathcal{P}} \text{KL}(q(\Omega,\Theta)\|p_0(\Omega,\Theta)) - \mathbb{E}_{q(\Omega)}[\log p(\widetilde{\mathbf{X}}|\Omega)] - \mathbb{E}_{q(\Omega,\Theta)}[\log(\ell(\mathbf{y}|\mathbf{H},\Theta)],$$

where $p_0(\Omega,\Theta)$ is the prior, $p_0(\Omega,\Theta) = p_0(\Omega)p_0(\Theta)$ and $p_0(\Theta)$ is the prior of Θ. By solving the optimization above, we get the desired post-data posterior distribution [Ghosh and Ramamoorthi, 2003]

$$q(\Omega,\Theta) = \frac{p_0(\Omega,\Theta)p(\widetilde{\mathbf{X}}|\Omega)\ell(\mathbf{y}|\mathbf{H},\Theta)}{\Xi(\widetilde{\mathbf{X}},\mathbf{y})},$$

where $\Xi(\widetilde{\mathbf{X}},\mathbf{y})$ is the normalization constant.

3 Post-Data Posterior Sampling with HMC

As we can see, the post-data posterior above is intractable to compute. Firstly, the pseudo-likelihood function $\ell(\cdot)$ involves a max operater which mixes the posterior inference difficult and inefficient. We introduce the data augmentation idea [Polson and Scott, 2011] to solve this problem. Secondly, the form of $\tilde{\varphi}(\mathbf{h}^n)$ mixes local conjugacy. So we adopt the gradient-based MCMC sampler for faster convergence. Thirdly, we introduce the ditributed DPMM to improve the efficiency. In the following, we devise a hybrid MCMC sampling algorithm that generates a sample from the post-data posterior distribution of each variable in turn, conditional on the current values of the other variables. It can be shown that the sequence of samples constitutes a Markov chain and the stationary distribution of that Markov chain is just the joint posterior.

3.1 Updating variables $\boldsymbol{\beta}$, λ^n, $\boldsymbol{\omega}_j$ and $\mathbf{h}^n$

In this part, we develop a MCMC sampler for $\boldsymbol{\beta}$, λ^n, $\boldsymbol{\omega}_j$ and $\mathbf{h}^n$ by introducing augmented variables.

Data augmentation

The pseudo-likelihood function $\ell(\cdot)$ involves a max operater which mixes the posterior inference difficult and inefficient. So we re-express the pseudo-likelihood function into the integration of a function with augmented variable based on the data augmentation idea:

$$\ell(y^n|\tilde{\varphi}(\mathbf{h}^n),\boldsymbol{\beta}) = \int_0^\infty \frac{\exp\{\frac{-[\lambda^n+C(1-y^n\boldsymbol{\beta}^{\mathrm{T}}\tilde{\varphi}(\mathbf{h}^n))]^2}{2\lambda^n}\}}{\sqrt{2\pi\lambda^n}}d\lambda^n.$$

Then we can get the non-normalized joint distribution of $\mathbf{y}$ and $\boldsymbol{\lambda}$ conditional on $\mathbf{H}$ and $\boldsymbol{\Theta}$:

$$\ell(\mathbf{y},\boldsymbol{\lambda}|\mathbf{H},\boldsymbol{\Theta}) = \prod_{n=1}^{N}\frac{\exp\{\frac{-1}{2\lambda^n}[\lambda^n+C(1-y^n\boldsymbol{\beta}^{\mathrm{T}}\tilde{\varphi}(\mathbf{h}^n))]^2\}}{\sqrt{2\pi\lambda^n}}.$$

Sampling $\boldsymbol{\beta}$:
The conditional distribution of $\boldsymbol{\beta}$ is

$$q(\boldsymbol{\beta}|v,\mathbf{H},\boldsymbol{\omega},\mathbf{y},\boldsymbol{\lambda}) \sim p(\boldsymbol{\beta}|v)\prod_{n=1}^{N}\ell(y^n|\tilde{\varphi}(\mathbf{h}^n),\boldsymbol{\beta}) \propto \exp(-\frac{v||\boldsymbol{\beta}||^2}{2}-\sum_{n=1}^{N}\frac{[\lambda^n+\Lambda]^2}{2\lambda^n}),$$

where $\Lambda = C(1-y^n\boldsymbol{\beta}^{\mathrm{T}}\tilde{\varphi}(\mathbf{h}^n))$. This conditional distribution is a Gaussian distribution with covariance $\boldsymbol{\Sigma}_{\boldsymbol{\beta}} = \{v\mathbf{I}_{2M+1} + \sum_{n=1}^{N}\frac{C^2\tilde{\varphi}(\mathbf{h}^n)\tilde{\varphi}(\mathbf{h}^n)^{\mathrm{T}}}{\lambda^n}\}^{-1}$ and mean $\boldsymbol{\mu}_{\boldsymbol{\beta}} = \boldsymbol{\Sigma}_{\boldsymbol{\beta}}\sum_{n=1}^{N}(\frac{C\lambda^n+C^2}{\lambda^n})y^n\tilde{\varphi}(\mathbf{h}^n))$.

Sampling λ^n :
The conditional distribution over the augmented variable λ^n is a generalized inverse Gaussian distribution:

$$q(\lambda^n|\mathbf{h}^n,\boldsymbol{\omega},y^n,\boldsymbol{\beta}) \propto \exp(-\frac{1}{2\lambda^n}\{\lambda^n+C[1-y^n\boldsymbol{\beta}^{\mathrm{T}}\tilde{\varphi}(\mathbf{h}^n)]\}^2) \sim \mathcal{GIG}(\lambda^n|\frac{1}{2},1,C^2[1-y^n\boldsymbol{\beta}^{\mathrm{T}}\tilde{\varphi}(\mathbf{h}^n)]^2).$$

Sampling $\boldsymbol{\omega}_j$, $\mathbf{h}^n$:
Unfortunately, we find the conditional distribution over $\boldsymbol{\omega}_j$

$$q(\boldsymbol{\omega}_j|\mathbf{H},\boldsymbol{\lambda},\boldsymbol{\beta},\mathbf{y}) \sim p(\boldsymbol{\omega}_j|\boldsymbol{\mu}_{z_j},\boldsymbol{\Sigma}_{z_j})\prod_{n=1}^{N}\ell(y^n|\tilde{\varphi}(\mathbf{h}^n),\boldsymbol{\beta}),$$

where $\tilde{\varphi}(\mathbf{h}^n) = (\varphi(\mathbf{h}^n)^{\mathrm{T}},1)^{\mathrm{T}} = \{\frac{1}{\sqrt{M}}[\cos(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{h}^n),\cdots,\cos(\boldsymbol{\omega}_M^{\mathrm{T}}\mathbf{h}^n),\sin(\boldsymbol{\omega}_1^{\mathrm{T}}\mathbf{h}^n),\cdots,\sin(\boldsymbol{\omega}_M^{\mathrm{T}}\mathbf{h}^n)],1\}^{\mathrm{T}}$ is too complex and the prior is non-conjugated. So it is hard to get the analytical form of the above distribution. To generate a sample from $q(\boldsymbol{\omega}_j|\mathbf{H},\boldsymbol{\lambda},\boldsymbol{\beta},\mathbf{y})$ with its non-normalized density, we appeal to the HMC method [Neal, 2011]. Sampling from a distribution with HMC requires translating the density function for this distribution to a potential energy function and introducing 'momentum' variables to go with the original variables of interest ('position' variables). We need to constitute a Markov chain. Each iteration has two steps. In the first step, we sample the momentum which needs the gradient of the potential energy function. In the second step, we do a Metropolis update with a proposal found by using Hamiltonian dynamics.

Similar to $\boldsymbol{\omega}_j$, we also find the conditional posterior distribution over $\mathbf{h}^n$

$$q(\mathbf{h}^n|\boldsymbol{\omega},\boldsymbol{\lambda},\boldsymbol{\beta},\mathbf{y},\mathbf{X}) \sim p(\mathbf{h}^n|\mathbf{0},\mathbf{I})\prod_{i=1}^{N_v}p(\mathbf{x}_i^n|\mathbf{W}_i,\mathbf{h}^n,\mathbf{u}_i^n,\tau_i) \cdot \ell(y^n|\tilde{\varphi}(\mathbf{h}^n),\boldsymbol{\beta}),$$

doesn't have the analytical form. So we can sample $\mathbf{h}^n$ with the HMC sampler.

3.2 Updating the variables in distributed DPMM

Unlike a Gibbs sampler using the marginal representation of the DP mixtures, slice sampling methods [Walker, 2007; Kalli *et al.*, 2011; Ge *et al.*, 2015] employ the random measure G directly. It consists of the imputation of G and subsequent Gibbs sampling of the component assignments from their posteriors. To be able to represent the infinite number of components in G_0, we have to introduce some auxiliary (slice) variables t_j, $j = 1,\cdots,M$. Through introducing an auxiliary variable t_j, the joint density of $\boldsymbol{\omega}_j$ and the latent variable t_j becomes

$$p(\boldsymbol{\omega}_j,t_j|\boldsymbol{\varpi},\boldsymbol{\zeta}) = \sum_{k=1}^{\infty}\varpi_k\mathrm{Unif}(t_j|0,\varpi_k)p(\boldsymbol{\omega}_j|\zeta_k) = \sum_{k=1}^{\infty}\mathbb{1}(t_j\le\varpi_k)p(\boldsymbol{\omega}_j|\zeta_k),$$

where $\mathbb{1}(\cdot)$ is the indicator function. We can easily verify that when t_j is integrated over, the joint density is equivalent to $p(\boldsymbol{\omega}_j|\boldsymbol{\varpi},\boldsymbol{\zeta})$. Thus the interesting fact is that given the latent variable t_j, the number of mixtures needed to be represented is finite.

Through introducing the slice variable, the conditional posterior distribution of $\boldsymbol{\omega}_j$ becomes independent so it is possible to derive a parallel sampler for the DP mixture model under the Map-Reduce framework. Our parallel sampler in distributed DPMM is similar as that in [Ge *et al.*, 2015], thus is omitted here.

3.3 Updating the other variables

In this part, the other variables all have the conjugate prior. So we get the analytic conditional posterior distributions of them.

Sampling $\mathbf{u}_i^n$:
The conditional posterior of $\mathbf{u}_i^n$ is

$$q(\mathbf{u}_i^n|\tau_i,\mathbf{h}^n,\mathbf{x}_i^n,\mathbf{W}_i) \sim p(\mathbf{u}_i^n|\mathbf{0},\mathbf{I})p(\mathbf{x}_i^n|\mathbf{W}_i,\mathbf{h}^n,\mathbf{u}_i^n,\mathbf{V}_i,\tau_i) \propto \exp\{-\frac{1}{2}(||\mathbf{u}_i^n||^2-\tau_i||\mathbf{x}_i^n-\mathbf{W}_i\mathbf{h}^n-\mathbf{V}_i\mathbf{u}_i^n||^2)\},$$

a Gaussian distribution with covariance $\boldsymbol{\Sigma}_{\mathbf{u}_i^n} = (\mathbf{I}+\tau_i{\mathbf{V}_i}^{\mathrm{T}}\mathbf{V}_i)^{-1}$ and mean $\boldsymbol{\mu}_{\mathbf{u}_i^n} = \boldsymbol{\Sigma}_{\mathbf{u}_i^n}[{\mathbf{V}_i}^{\mathrm{T}}(\mathbf{x}_i^n-\mathbf{W}_i\mathbf{h}^n)\tau_i]$.

Sampling $\mathbf{W}_i$, $\mathbf{V}_i$:
The conditional posterior distribution of $\mathbf{W}_i$ is proportional to the prior times the likelihood:

$$q(\mathbf{w}_{i,\cdot j}|r_{i,j},\tau_i,\mathbf{H},\mathbf{U}_i,\mathbf{X}_i,\mathbf{V}_i) \sim p(\mathbf{w}_{i,\cdot j}|\mathbf{0},r_{i,j}^{-1}\mathbf{I})p(\mathbf{X}_i|\mathbf{W}_i,\mathbf{H},\mathbf{U}_i,\mathbf{V}_i,\tau_i) \propto \exp\{-\frac{1}{2}(r_{ij}||\mathbf{w}_{i,\cdot j}||^2+\sum_{n=1}^{N}\tau_i||\mathbf{x}_i^n-\mathbf{W}_i\mathbf{h}^n-\mathbf{V}_i\mathbf{u}_i^n||^2)\},$$

where $j = 1,\cdots,m$. This is a Gaussian distribution with covariance $\boldsymbol{\Sigma}_{\mathbf{w}_{i,\cdot j}} = (r_{ij}+\tau_i||\mathbf{h}_{j\cdot}||^2)^{-1}\mathbf{I}$ and mean $\boldsymbol{\mu}_{\mathbf{w}_{i,\cdot j}} = \boldsymbol{\Sigma}_{\mathbf{w}_{i,\cdot j}}\{\sum_{n=1}^{N}[\mathbf{x}_i^n-\sum_{k\neq j}^{m}\mathbf{w}_{i,\cdot k}h_{kn}-\mathbf{V}_iu_{i,jn}]\tau_ih_{jn}\}$.

Similar to $\mathbf{W}_i$, conditional posterior of $\mathbf{V}_i$ is a Gaussian distribution with covariance $\boldsymbol{\Sigma}_{\mathbf{v}_{i,\cdot j}} = (\eta+\tau_i||\mathbf{u}_{i,\cdot j}||^2)^{-1}\mathbf{I}$ and mean $\boldsymbol{\mu}_{\mathbf{v}_{i,\cdot j}} = \boldsymbol{\Sigma}_{\mathbf{v}_{i,\cdot j}}\{\sum_{n=1}^{N}[\mathbf{x}_i^n-\sum_{k\neq j}^{m}\mathbf{v}_{i,\cdot k}u_{i,kn}-\mathbf{V}_iu_{i,jn}]\tau_ih_{jn}\}$.

Sampling $\mathbf{r}_i$, τ_i :
For each $r_{ij}, j = 1, \cdots, m$, its conditional distribution is

$$q(r_{ij}) \sim p(r_{ij}|a_r, b_r)p(\mathbf{w}_{i,\cdot j})|\mathbf{0}, r_{ij}^{-1}\mathbf{I}) \propto r_{ij}^{a_r-1+\frac{D_i}{2}} \exp\{-r_{ij}(b_r + \sum_{d=1}^{D_i} \frac{1}{2}||\mathbf{w}_{i,\cdot j}||^2)\},$$

a Gamma distribution with the shape and rate parameter $a_{r_{ij}} = a_r + \frac{D_i}{2}$, $b_{r_{ij}} = b_r + \sum_{d=1}^{D_i} \frac{1}{2}||\mathbf{w}_{i,\cdot j}||^2$.

Similar to sampling $\mathbf{r}_i$, the conditional distribution of τ_i is a Gamma distribution with the shape and rate parameter $a_{\tau_i} = a_\tau + \frac{ND_i}{2}$, $b_{\tau_i} = b_\tau + \sum_{n=1}^{N} ||\mathbf{x}_i^n - \mathbf{W}_i\mathbf{h}^n - \mathbf{V}_i\mathbf{u}_i^n||^2$.

Computational Complexity

In our post-data posterior sampling, the dominant computation is spent on sampling latent shared variables $\mathbf{H}$. In each round of parameter sampling, our algorithm consumes $O(NL\sum_{i=1}^{N_v} D_i(m + K_i))$ operations where L is the number of steps for the leapfrog method in HMC. So the computational complexity of our algorithm M^{3}LAK is linear w.r.t. the number of instances N.

4 Experiments

In this section, we evaluate our proposed model (M^{3}LAK) on various classification tasks.

4.1 Data Description

The Flickr dataset contains 3,411 images of 13 animals [Chen *et al.*, 2012]. For each image, two types of features are extracted, including 634-dim real-valued features and 500-dim bag of word SIFT features. Trecvid contains 1,078 manually labeled video shots that belongs 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. The web-page 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 web pages from computer science department in three universities, i.e., Cornell, Washington, Wisconsin.

We transform these multi-class data sets into binary ones by following the way in [Zhuang *et al.*, 2012]. For example, the web-page classification with five categories ('course', 'faculty', 'student', 'project', 'staff'), we select category 'student' as a group and the other four categories as another group, since the number of examples in category 'student' is similar with the one belonging to the other four categories. The other data sets are similarly constructed. The details of these data sets are shown in Table 1.

4.2 Baselines

We compare our model with five competitors:

- BM^{2}SMVL [He *et al.*, 2016]: a linear Bayesian max-margin subspace multi-view learning method.

Table 1: Detail description of datasets.

Datasets	Trecvid	Flickr	Cornell	Washington	Wisconsin
Size	1078	3411	195	217	262
D_1	1894	634	1703	1703	1703
D_2	165	500	195	217	262

- MVMED [Sun and Chao, 2013]: a multi-view maximum entropy discrimination model.
- VMRML [Quang *et al.*, 2013]: a vector-valued manifold regularization multi-view learning method.
- MMH [Chen *et al.*, 2012]: a predictive latent subspace Markov network multi-view learning model.
- BEMKL [Gonen, 2012]: a state-of-the-art Bayesian multiple kernel learning method which we use for multi-view learning. For each view, we construct Gaussian kernels with 21 different widths $\{2^{-10}, 2^{-9}, ..., 2^{10}\}$ on all features using its public implementation [1].

4.3 Experimental Setting

In M^{3}LAK, we perform 5-fold cross-validation on training set to decide the regularization parameter C from the integer set $\{5, 6, 7, 8, 9\}$ for each data set. The rest parameters are fixed as follows for all data sets, i.e., $m = 20$, $M = 100$, $\eta =$ 1e+3, $\alpha = 1$, $a_r =$ 1e-1, $a_\tau = v =$ 1e-2, $b_\tau = b_r =$ 1e-5. For fair comparison, the subspace dimension m in BM^{2}SMVL and MMH is also fixed to 20 for all data sets. For BM^{2}SMVL, the regularization parameter 'C' is chosen from the integer set $\{1, 2, 3\}$ by 5-fold cross-validation on training set according to its original paper. For MVMED, we choose 'c' from $2^{[-5:5]}$ for each data set as suggested in its original paper. For VMRML, the parameters are set as the default values in its paper. For RBF kernel's parameter of MVMED and VMRML, we carefully tune them on each data set separately. For MMH, we tune its parameters as suggested in its original papers. On each data set, we conduct 10-fold cross validation for all the algorithms, where nine folds of the data are used for training while the rest for testing. The averaged accuracies over these 10 runs are reported in Table 2.

4.4 Experimental Results

We have the following insightful observations:

- M^{3}LAK consistently outperforms BM^{2}SMVL. The reason may be that BM^{2}SMVL is a linear multi-view method with limited modeling capabilities.
- On most data sets, M^{3}LAK performs better than MMH. This may be because that unlike MMH which are under the maximum entropy discrimination framework, and can not infer the penalty parameter of max-margin models in a Bayesian style, our method is based on the data augmentation idea for max-margin learning, which allows us to automatically infer the weight parameters and the penalty parameter.
- M^{3}LAK has better performance than the single kernel multi-view learning methods VMRML and MVMED on

[1] http://users.ics.aalto.fi/gonen/bemkl/

Table 2: Comparison of test accuracies (mean ± std) on all datasets. Bold face indicates highest accuracy.

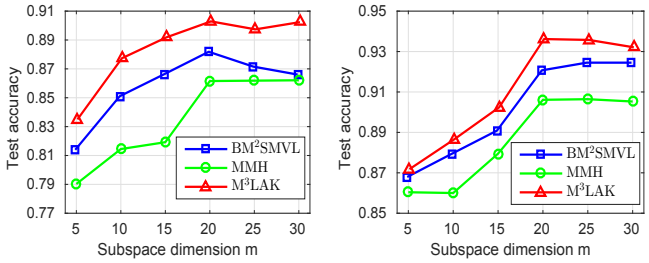
	Trecvid-a	Trecvid-b	Flickr-a	Flickr-b	Cornell	Washington	Wisconsin
MMH	.939 ± .066	**.944 ± .034**	.820 ± .085	.823 ± .055	.862 ± .063	.909 ± .042	.906 ± .043
MVMED	.913 ± .030	.920 ± .068	.854 ± .088	.858 ± .065	.861 ± .080	.852 ± .077	.872 ± .047
VMRML	.921 ± .028	.932 ± .063	.800 ± .068	.834 ± .044	.882 ± .069	.874 ± .069	.883 ± .073
BM^{2}SMVL	.901 ± .019	.912 ± .059	.827 ± .070	.856 ± .048	.882 ± .072	.896 ± .055	.921 ± .072
BEMKL	.944 ± .033	.932 ± .060	**.857 ± .087**	**.871 ± .046**	.861 ± .060	.874 ± .052	.902 ± .072
M^{3}LAK	**.954 ± .033**	.940 ± .054	.855 ± .056	**.871 ± .034**	**.903 ± .065**	**.913 ± .058**	**.936 ± .050**

all data sets. The reason may be that M^{3}LAK 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. And MVMED is also under the maximum entropy discrimination framework, and can not infer the penalty parameter of max-margin models in a Bayesian style.

- M^{3}LAK performs better than BEMKL on most data sets. BEMKL's performance may be limited by its mean-field assumption on the approximate posterior and the absence of max-margin principle while M^{3}LAK introduces the popular max-margin principle which has a great generalization ability. Although BEMKL performs better than M^{3}LAK on some data sets, BEMKL has to perform matrix inversion to compute the posterior covariance of kernel weights in each round of iteration which requires $O(N^3)$ operations. Besides, BEMKL needs to store many Gram matrix to get good performances. However, storing too many Gram matrix leads to out of memory on commonly used computers.

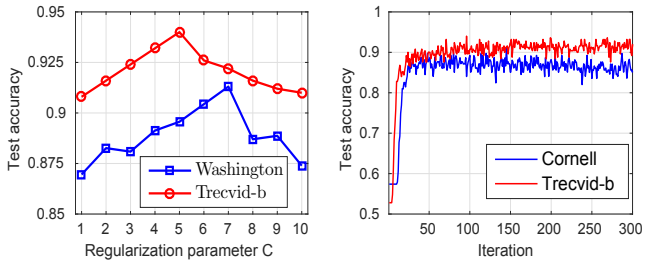
4.5 Parameter Study and Convergence

We study the performance change of the three subspace learning methods (BM^{2}SMVL, MMH and M^{3}LAK). Performances change when the subspace dimension m varies on two datasets (Cornell and Wisconsin). The averaged results are shown in Figure 1. As we can see, different methods prefer different values of m. On some datasets, when m becomes too large, the performances of these three methods become poor. When m ranges from 5 to 30, M^{3}LAK performs better than other subspace learning methods in general.



(a) Cornell (b) Wisconsin

Figure 1: Effect of subspace dimension m.



(a) Effect of C (b) Convergency

Figure 2: (a) Effect of regularization parameter C in M^{3}LAK; (b) Convergency of M^{3}LAK.

Also, we set m as 20 and study the influence of regularization parameter C. From the results in Figure 2 (a) , we can find that different data sets may prefer different values of C. C balances the nonlinear classifier with adaptive kernel and the multi-view latent variable model, so M^{3}LAK cannot get the best performance when C is too large or small.

Figure 2 (b) shows the convergency of M^{3}LAK on two data sets. We find that M^{3}LAK has a fast convergence rate, which we contribute to the efficient gradient-based HMC sampler [Neal, 2011].

5 Conclusion

In this paper, we present an adaptive kernel nonlinear max-margin multi-view learning framework. It regularizes the posterior of an efficient multi-view LVM by explicitly mapping the latent representations extracted from multiple data views to a random fourier feature space where max-margin classification constraints are imposed. Having no need to compute the Gram matrix, the computational complexity of our algorithm is linear w.r.t. N. Extensive experiments on real-world datasets demonstrate our method has a superior performance, compared with a number of competitors.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61473273, 61602449, 91546122, 61573335, 61602438, 61473274), National High-tech R&D Program of China (863 Program) (No.2014AA015105), Guangdong provincial science and technology plan projects (No. 2015 B010109005), the Youth Innovation Promotion Association CAS 2017146.

References

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

[Fang and Zhang, 2012] Zheng Fang and Zhongfei Zhang. Simultaneously combining multi-view multi-label learning with maximum margin classification. In *ICDM*, pages 864–869, 2012.

[Farquhar *et al.*, 2005] Jason Farquhar, David Hardoon, Hongying Meng, John S Shawe-taylor, and Sandor Szedmak. Two view learning: Svm-2k, theory and practice. In *Advances in neural information processing systems*, pages 355–362, 2005.

[Ge *et al.*, 2015] Hong Ge, Yutian Chen, ENG CAM, Moquan Wan, and Zoubin Ghahramani. Distributed inference for dirichlet process mixture models. In *ICML*, pages 2276–2284, 2015.

[Ghosh and Ramamoorthi, 2003] Jayanta K Ghosh and RV Ramamoorthi. *Bayesian nonparametrics*, volume 1. Springer New York, 2003.

[Gonen, 2012] Mehmet Gonen. Bayesian efficient multiple kernel learning. In *ICML*, pages 1–8, 2012.

[He *et al.*, 2016] Jia He, Changying Du, Fuzhen Zhuang, Xin Yin, Qing He, and Guoping Long. Online bayesian max-margin subspace multi-view learning. *IJCAI*, pages 1555–1561, 2016.

[Hofmann *et al.*, 2007] Thomas Hofmann, Bernhard Schölkopf, and Alexander J. Smola. Kernel methods in machine learning. *Annals of Statistics*, 36(3):1171–1220, 2007.

[Jaakkola *et al.*, 1999] Tommi Jaakkola, Marina Meila, and Tony Jebara. Maximum entropy discrimination. In *Advances in neural information processing systems*, 1999.

[Kalli *et al.*, 2011] Maria Kalli, Jim E Griffin, and Stephen G Walker. Slice sampling mixture models. *Statistics and computing*, 21(1):93–105, 2011.

[Lanckriet *et al.*, 2004] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the kernel matrix with semidefinite programming. *JMLR*, 5:27–72, 2004.

[Neal, 2011] Radford M Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2:113–162, 2011.

[Oliva *et al.*, 2016] Junier Oliva, Avinava Dubey, Barnabas Poczos, Jeff Schneider, and Eric P. Xing. Bayesian nonparametric kernel learning. *AISTATS*, 2016.

[Polson and Scott, 2011] Nicholas G Polson and Steven L Scott. Data augmentation for support vector machines. *Bayesian Analysis*, 6(1):1–23, 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 *ICML*, pages 100–108, 2013.

[Rahimi and Recht, 2007] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2007.

[Rakotomamonjy *et al.*, 2008] Alain Rakotomamonjy, Francis R Bach, Stéphane Canu, and Yves Grandvalet. Simplemkl. *JMLR*, 9(Nov):2491–2521, 2008.

[Rudin, 2011] Walter Rudin. *Fourier analysis on groups*. John Wiley & Sons, 2011.

[Sonnenburg *et al.*, 2006] Sören Sonnenburg, Gunnar Rätsch, Christin Schäfer, and Bernhard Schölkopf. Large scale multiple kernel learning. *JMLR*, 7(2006):1531–1565, 2006.

[Sun and Chao, 2013] Shiliang Sun and Guoqing Chao. Multi-view maximum entropy discrimination. In *IJCAI*, pages 1706–1712, 2013.

[Szedmak and Shawe-Taylor, 2007] Sandor Szedmak and John Shawe-Taylor. Synthesis of maximum margin and multiview learning using unlabeled data. *Neurocomputing*, 70(7-9):1254–1264, 2007.

[Walker, 2007] Stephen G Walker. Sampling the dirichlet mixture model with slices. *Communications in Statistics-Simulation and Computation®*, 36(1):45–54, 2007.

[Wang, 2007] Chong Wang. Variational bayesian approach to canonical correlation analysis. *Neural Networks*, 18(3):905–910, 2007.

[Xu *et al.*, 2014] Chang Xu, Dacheng Tao, Yangxi Li, and Chao Xu. Large-margin multi-view gaussian process. *Multimedia Systems*, 21(2):147–157, 2014.

[Zhu *et al.*, 2012] Jun Zhu, Amr Ahmed, and Eric P Xing. Medlda: maximum margin supervised topic models. *JMLR*, 13(1):2237–2278, 2012.

[Zhuang *et al.*, 2012] Fuzhen Zhuang, George Karypis, Xia Ning, Qing He, and Zhongzhi Shi. Multi-view learning via probabilistic latent semantic analysis. *Information Sciences*, 199(15):20–30, 2012.