Max-Margin Discriminant Projection via Data Augmentation

Bo Chen, *Member, IEEE*, Hao Zhang, Xuefeng Zhang, Wei Wen, Hongwei Liu, *Member, IEEE*, and Jun Liu

Abstract—In this paper, we introduce a new max-margin discriminant projection method, which takes advantage of the latent variable representation for support vector machine (SVM) as the classification criterion. Specifically, the proposed model jointly learns the discriminative subspace and classifier in a Bayesian framework by conditioning on augmented variables. Moreover, an extended nonlinear model is developed based on the kernel trick, where the similar model can be used in this setting with few modifications. To explore the sparsity in the kernel expansion, we use the spike-and-slab prior to seek basis vectors (BVs) from the corresponding candidates. Unlike existing methods, which employ BVs to approximate the original feature space, in our method BVs are sought to associate the final classification task. Thanks to the conditionally conjugate property, the parameters in our models can be inferred via the simple and efficient Gibbs sampler. Finally, we test our methods on synthesized and real-world data, including large-scale data sets to demonstrate their efficiency and effectiveness.

Index Terms—Max-margin, Gibbs sampling, feature extraction, kernel methods, radar automatic target recognition (RATR)

1 Introduction

As a common and necessary step in many machine learning applications, supervised feature extraction has been widely studied over the past few decades [35]. Linear discriminant analysis (LDA) [15] is a fundamental method for linear supervised feature extraction. However, LDA highly depends on the data distribution and can only deal with linearly separable problems well, which may limit its capability in practice [7], [49]. Many researchers have proposed different ways to handle those weaknesses [9], [12], [38], [43]. In order to take advantage of Bayesian modeling and label information, various techniques have been also produced to learn the discriminative subspace for the classifiers [16], [23], [44].

In classification, the best known example is the support vector machine (SVM) [36], which maximizes the margin between different classes. However, it is difficult to formulate the max-margin criterion under a Bayesian framework, which results in the case that an optimization-based SVM solver has to be accessed per-iteration during the Bayesian inference [48]. Recently, Polson and Scott [28] bridge the gap via the data augmentation technique, which encourages us to develop a new max-margin feature extraction method. Specifically, the proposed model jointly learns the discriminative subspace and max-margin classifier, where the parameters can be effectively inferred via the efficient Gibbs sampler [17], [18], since it has good conjugacy conditioned on augmented variables. We call our linear model max-margin linear discriminant projection (MMLDP). Unlike LDA, the proposed method does not assume any prior distribution on the data.

Moreover, we extend the linear model to nonlinear version based on kernel trick and explore the sparse kernel expansion [1], [32]. We employ the spike-and-slab prior [39] to infer the sparsity in the kernel expansion and find basis vectors (BVs) from a set of candidates, either local centers or data samples. In the experiments, we found that in the model a relatively small number of BVs have been able to preserve the classification performance of kernel expansion, as long as they capture the underlying structure of data distribution, such as the multimodal structure in the data distribution.

The remainder of this paper is organized as follows. Section 3 briefly introduces the Bayesian SVM. Section 4 describes the proposed model, max-margin discriminant projection, which includes the linear projection and nonlinear versions. We also give a way to model the sparsity in the kernel expansion. In Section 5, we conduct large-scale experiments on synthetic, benchmark, and measured radar high-resolution range profile (HRRP) data sets to evaluate our methods.

2 Related Work

LDA is a representative supervised subspace analysis method and can be solved by eigenvalue decomposition as a spectral method. However, LDA has two underlying assumptions: the samples in each class satisfy the Gaussian distribution and the classes are linearly separable. To solve the problems, many improved LDA variants have been proposed [9], [12], [38], [43]. To overcome the small sample size problem, Wang and Tang [38] combined the conventional LDA and null-space LDA to propose a dual-space LDA. In order to address the singularity problem, Ye and Wang [43] presented an efficient algorithm to calculate the projective

• *The authors are with the National Lab of Radar Signal Processing, Xidian University, Xi'an, Shaanxi 710071, China. E-mail: bchen@mail.xidian.edu.cn, 411800739@qq.com, {zxf0913, wenwei8114}@163.com, hwliu@xidian.edu.cn, jun_liu_math@hotmail.com.*

Manuscript received 31 Dec. 2013; revised 19 Dec. 2014; accepted 9 Jan. 2015. Date of publication 1 Feb. 2015; date of current version 1 June 2015. Recommended for acceptance by J. Bailey. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TKDE.2015.2397444

functions of Regularized Discriminant Analysis (RDA). To incorporate sparsity into the LDA transformation, Clemmensen et al. [12] provided a Sparse Discriminant Analysis (SDA), which imposes a sparseness constraint on projection vectors with a set of interpretable features for classification. Although these methods produce high-quality results, they do not change the Fisher's discriminant criterion in light of the separability aspect. From the graph embedding point of view, Local Discriminant Embedding (LDE, [9]) and Marginal Fisher Analysis (MFA, [41]), both of which are essentially same, are proposed as the combinations of the locality preserving technique and the discriminant analysis techniques. Different from the conventional LDA, they treat the data locally and can obtain more projection directions to better characterize the separability of different classes. All the above methods, however, have two common issues: 1. the learned feature subspaces have no direct connection to the final classifier; 2. they lack probabilistic interpretation.

On the contrast, probabilistic approaches aim to infer an entire distribution profile for the latent structures given the observations and some prior distributions following a Bayes' rule. In the present paper, we focus on Bayesian approaches, since it can naturally incorporate diverse prior knowledge and be extensible to nonparametric methods. Yu et al. [44] proposed a linear supervised probabilistic PCA and an efficient solution method with a decoder form, while the algorithm is developed only for real outputs. Rai and Daumé III [29] developed a supervised Bayesian probabilistic canonical component analysis (BCCA), which learns the subspace shared by real observed data and labels. However, both of models consider the classification problem as the imputation of missing values without using any classification criterion. Along the other line, some methods simultaneously learn discriminative subspace and a specific classifier by modeling the joint distribution of the latent representations and the labels. The work in this paper falls in that category and is most related to Bayesian supervised dimension reduction (BSDR, [16]) in terms of the form of latent representation and projection. Nevertheless, as other existing supervised Bayesian model [10], [19], BSDR uses a conventional generalized linear model (GLMs) as a separation criterion to learn the discriminative subspace. In our work, we will utilize the latent max-margin classifier to infer the discriminative projection matrix.

Based on max-margin criterion, Zhu and his colleagues [40], [47], [48] proposed a series of max-margin supervised model for document analysis and collaborative filtering, all of which train the discriminative latent topics/factors. Typically, the infinite latent SVM (iLSVM) uses the binary infinite latent representation as the input for the following max-margin classifier solved by a standard SVM solver. Via the data augmentation technique, they further proposed the Gibbs max-margin topic model [47], which can use Gibbs sampling to efficiently infer the parameters. In our work, we will address more general data rather than the count matrix. Meanwhile, in their mode they use the decoder/decomposition form for data analysis, which has to learn both the topics/factors and corresponding coefficients to reconstruct data well, whereas our model is an encoder model that only focuses on the projection matrix without any requirement for data reconstruction. We also extend the linear projection to kernelized max-margin projection, where, different from other kernel methods, the kernel matrix does not need to be Gram matrix, so that we are able to explicitly exploit and analyze the sparsity in the kernel expansion.

Considering the approximation of kernel matrix, emerging methods are focused on forming an approximation to the original kernel feature space and capturing the geometrical structure of the whole data in feature space [1], [3], [4]. What's more, they learn BVs and train the model separately, hence the quality of selected BVs is independent of the class separability criterion, as well as the classification performance in the new feature space. In this paper we will present a way to simultaneously infer the BVs, discriminative feature space and classifier via the spike-and-slab prior. In other words, it seeks the BVs of interest to help the final classifier prediction.

TABLE 1
List of Notations

N	Number of training instances
L	Number of classes
P	Dimensionality of input space
K	Dimensionality of projected subspace
M	Number of basis vectors
$\mathbf{X}$	$P \times N$ matrix of data instances
$\mathbf{A}$	$P \times K$ matrix of projection variables
$\mathbf{Z}$	$K \times N$ matrix of projected data instances
Θ	$M \times K$ matrix of combination variables in kernel model
$\boldsymbol{y}$	$N \times 1$ vector of binary labels
w	$K \times 1$ vector of SVM coefficient
$\boldsymbol{\lambda}$	$N \times 1$ vector of augmented varables
$\boldsymbol{\alpha}$	$K \times 1$ vector of precision priors over projection variables
σ	scalar of precision prior over SVM coefficienet
$\boldsymbol{\pi}$	$M \times 1$ vector of potential variables

3 Bayesian SVM

Given a labeled training set $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$ with the data vectors $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and their labels $\boldsymbol{y} = (y_1, \ldots, y_N) \in \{-1, +1\}^N$, where each feature vector $\boldsymbol{x}_n$ is P-dimensional, $\boldsymbol{x}_n \in \mathbb{R}^P$. For clear demonstration, we list all of notations in Table 1. Conventional L_1-SVM, that penalizes the L_1-norm errors, finds a hyperplane $f(x) = \boldsymbol{w}^T\boldsymbol{x} + b$ by solving the following optimization problem [36]:

$$\begin{aligned} \min_{\boldsymbol{w},b,\xi_n} \quad & \frac{1}{2} \parallel \boldsymbol{w} \parallel^{\alpha} + C\sum_{n=1}^{N} \xi_n, \\ \text{s.}t. \quad & y_n(\boldsymbol{w}^T\boldsymbol{x}_n + b) \geq 1 - \xi_n, \\ & \xi_n \geq 0, \qquad n = 1, \ldots, N \end{aligned} \tag{1}$$

where the underlying discriminative objective is a linear hinge loss function, $\max(1 - y_n(\boldsymbol{x}_n^T\boldsymbol{w} + b), 0)$, which seems to make traditional Bayesian analysis difficult to model. According to the sign of $\boldsymbol{x}_n^T\boldsymbol{w} + b$, one can classify the observation $\boldsymbol{x}_n$ as $+1$ or -1. The most popular selection for α is 1 corresponding to the lasso prior, because it leads to posterior distributions that push many elements of $\boldsymbol{w}$ close to zero [33]. In this paper, for the convenience of modeling, we employ a Student-t prior, implemented via

the hierarchical construction of normal-gamma distribution on $\boldsymbol{w}$ [34], instead of the double-exponential (Laplace) prior, since they have the similar sparseness-promoting behavior [6], [27]

$$\boldsymbol{w} \sim \mathcal{N}(0, \sigma^{-2}\mathbf{I}), \qquad \sigma \sim \mathrm{Ga}(a_0, b_0). \tag{2}$$

Based on max-margin classification criterion, it is hard to use any prior distribution to generate label y or model the hinge loss function through a formal likelihood. Instead, Polson and Scott utilizes the pseudo-likelihood to represent the likelihood contribution from label y_n via the idea of data augmentation [28]. The unnormalized pseudo-likelihood can be expressed as

$$\begin{aligned}&\phi(y_n|\boldsymbol{w})\\ &\quad= \exp\{-2\max(1 - y_n(\boldsymbol{x}_n^T\boldsymbol{w} + b), 0)\}\\ &\quad= \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_n}} \exp\left(-\frac{(1 + \lambda_n - y_n(\boldsymbol{x}_n^T\boldsymbol{w} + b))^2}{2\lambda_n}\right) d\lambda_n.\end{aligned} \tag{3}$$

The details of the proof is referred to [28]. With the hierarchical Normal Gamma prior on $\boldsymbol{w}$ and conditionally Gaussian linear model expression of the SVM objective function, we can directly write down the complete data pseudo-posterior distribution as

$$\begin{aligned}&p(\boldsymbol{w}, \boldsymbol{\lambda}, \sigma|\boldsymbol{y})\\ &\quad\propto \prod_{n=1}^N \phi(\boldsymbol{y}_n, \boldsymbol{\lambda}_n|\boldsymbol{w}) p(\boldsymbol{w}|\sigma) p(\sigma)\\ &\quad\propto \prod_{n=1}^N \frac{1}{\sqrt{2\pi\boldsymbol{\lambda}_n}} \exp\left(-\sum_{n=1}^N \frac{(1 + \boldsymbol{\lambda}_n - \boldsymbol{y}_n\boldsymbol{x}_n^T\boldsymbol{w})^2}{2\boldsymbol{\lambda}_n}\right)\\ &\quad\times \mathcal{N}(\boldsymbol{w}; 0, \sigma^{-2}\mathbf{I})\mathrm{Ga}(\sigma; a, b).\end{aligned} \tag{4}$$

For simplicity, in the following formulas, we use $\boldsymbol{w}$ to represent the concatenation of coefficient $\boldsymbol{w}$ and bias b, correspondingly, each $\boldsymbol{x}$ represents $[\boldsymbol{x}; 1]$. It is relatively straightforward to develop a Gibbs sampler to repeatedly sample each random variable from its conditional distribution.

4 Max-Margin Discriminant Projection via Data Augmentation

In this section, we will present our max-margin discriminant projection model in detail, which includes the linear projection method and its nonlinear version.

4.1 Max-Margin Linear Discriminant Projection

The latent variable model is to discover latent feature representation of a set of observations. In the linear supervised projection method, we aim to find a matrix $\mathbf{A} \in \mathbb{R}^{P\times K}$, usually $K < P$ such that the projected feature vectors $\{\mathbf{A}^T\boldsymbol{x}_n\}$ are separable according to classes in the new discriminative subspace. Toward this end, we express the generative process of the latent representations as

$$z_n \sim \mathcal{N}(\mathbf{A}^T\boldsymbol{x}_n, \mathbf{I}_K), \boldsymbol{a}_k \sim \mathcal{N}(0, \alpha_k^{-1}\mathbf{I}_P), \alpha_k \sim \mathrm{Ga}(c_0, d_0), \tag{5}$$

where $\mathbf{A} = [\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_K]$ and $\boldsymbol{a}_k$ indicates the kth column in the projection matrix $\mathbf{A}$. α_k is employed for each individual projection coordinate to differentiate their importance in the discriminative subspace. With the Bayesian max-margin classifier stated above, we extend the model, Eq. (5), to the classification problem by introducing the pseudo-likelihood for hinge loss function, which can be explained as a regularized Bayesian inference problem [47]. The pseudo-posterior can be expressed as

$$\begin{aligned}&p(\mathbf{Z}, \mathbf{A}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \sigma|\boldsymbol{y})\\ &\quad\propto \prod_{n=1}^N p(z_n|\mathbf{A})\phi(y_n, \lambda_n|\boldsymbol{w}, z) p(\boldsymbol{w}|\sigma) p(\sigma)\\ &\qquad\prod_{k=1}^K p(\boldsymbol{a}_k|\alpha_k) p(\alpha_k).\end{aligned} \tag{6}$$

Under this setting, the conditional posteriors used to draw samples can be derived analytically. Here we implement the posterior computation by a Markov chain Monte Carlo (MCMC) method based on Gibbs sampling [17], [18], where the posterior distribution is approximated by a sufficient number of samples. At each iteration, the parameters are drawn in sequence from the following posterior distributions conditioned on the most recent values of all the other random variables. [18]

- Sample each column, $\boldsymbol{a}_k$, in the projection matrix $\mathbf{A}$ from $p(\boldsymbol{a}_k|-) = \mathcal{N}(\boldsymbol{a}_k; \boldsymbol{\mu}_{\boldsymbol{a}_k}, \Sigma_{\boldsymbol{a}_k})$ with

$$\begin{aligned}\Sigma_{\boldsymbol{a}_k} &= \left(\sum_{n=1}^N \boldsymbol{x}_n\boldsymbol{x}_n^T + \alpha_k\mathbf{I}_P\right)^{-1},\\ \boldsymbol{\mu}_{\boldsymbol{a}_k} &= \Sigma_{\boldsymbol{a}_k}\left(\sum_{n=1}^N z_{kn}\boldsymbol{x}_n\right).\end{aligned} \tag{7}$$

- Sample the latent feature z_{kn} from $p(z_{kn}|-) = \mathcal{N}(z_{kn}; \mu_{z_{kn}}, \Sigma_{z_{kn}})$

$$\begin{aligned}\Sigma_{z_{kn}} &= \left(\frac{w_k^2}{\lambda_n} + 1\right)^{-1},\\ \mu_{z_{kn}} &= \Sigma_{z_{kn}}\left[\left(\frac{\xi_n^{-k}}{\lambda_n} + 1\right) y_n w_k + \boldsymbol{a}_k^T\boldsymbol{x}_n\right],\end{aligned} \tag{8}$$

  where $\xi_n^{-k} = 1 - y_n\sum_{l=1,l\neq k}^K w_l z_{ln}$. Apparently, z_{kn} is not only related to the projection of $\boldsymbol{x}_n$ in coordinate k but also is adjusted by the supervised signal y_n.
- Sample the classifier coefficients $\boldsymbol{w}$ from $p(\boldsymbol{w}|-) = \mathcal{N}(\boldsymbol{w}; \boldsymbol{\mu}_{\boldsymbol{w}}, \Sigma_{\boldsymbol{w}})$ with

$$\begin{aligned}\Sigma_{\boldsymbol{w}} &= \left(\sum_{n=1}^N \frac{\boldsymbol{z}_n\boldsymbol{z}_n^T}{\lambda_n} + \sigma\mathbf{I}_K\right)^{-1},\\ \boldsymbol{\mu}_{\boldsymbol{w}} &= \Sigma_{\boldsymbol{w}}\left(\sum_{n=1}^N \left(1 + \frac{1}{\lambda_n}\right) y_n\boldsymbol{z}_n\right).\end{aligned} \tag{9}$$

- Sample the augmented random variable λ_n from

$$p(\lambda_n|-) = \mathcal{GIG}\left(\lambda_n; \frac{1}{2}, 1, \xi_n^2\right), \tag{10}$$

  where $\mathcal{GIG}(x; p, a, b) = C(p, a, b)x^{p-1}\exp(-\frac{1}{2}(\frac{b}{x} + ax))$ is a generalized inverse Gaussian distribution [47] and $C(p, a, b)$ is a normalization constant.

The remaining Gibbs update equations are relatively standard for the Normal-Gamma construction [46], which are also put in Supplementary Material, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2015.2397444. With the above setups, we can construct a Markov chain which iteratively draws samples from the above conditional distributions with an initial condition as shown in Algorithm 1. To sample from an inverse Gaussian distribution, we apply the transformation method with multiple roots [24].

Algorithm 1. Max-Margin Linear Discriminant Projection

Require: $\{\boldsymbol{x}_n, y_n\}_{n=1}^N$, T, $T_{\text{burn-in}}$, a_0, b_0, c_0, d_0
1: Initialize $\mathbf{A}$, $\mathbf{Z}$, $\boldsymbol{\lambda}$, $\boldsymbol{\alpha}$, σ
2: **for** t = 1 to T **do**
3: Sample $\boldsymbol{w}$ from Eq. (9)
4: Sample $\boldsymbol{\lambda}$ from Eq. (10)
5: Sample $\mathbf{A}$ from Eq. (7)
6: Sample $\mathbf{Z}$ from Eq. (8)
7: Sample $\boldsymbol{\alpha}$ and σ respectively
8: **if** $t > T_{\text{burn-in}}$ **then**
Collect samples.
9: **end if**
10: **end for**

4.2 Discussion

Analogous to BSDR, in MMLDP we also introduce an intermediate latent random variable $z_n \in \mathbb{R}^K$ [16] to record the latent representations. One may argue why not directly put $\{\tilde{\mathbf{A}}^T\boldsymbol{x}_n\}$ in the pseudo-likelihood to learn the projection matrix $\tilde{\mathbf{A}}^T$ as below

$$\phi\left(y_n|\boldsymbol{w}, \tilde{\mathbf{A}}^T\right) = \int_0^\infty \frac{1}{\sqrt{2\pi\lambda_n}} \exp\left(-\frac{\left(1+\lambda_n - y_n\left(\boldsymbol{x}_n^T\tilde{\mathbf{A}}\boldsymbol{w}+b\right)\right)^2}{2\lambda_n}\right) d\lambda_n. \tag{11}$$

In this section, we are going to demonstrate the difference between them. Following Eq. (11), the resulting posterior distribution of $\tilde{\boldsymbol{a}}_k$ can be written as $p(\tilde{\boldsymbol{a}}_k|-) = \mathcal{N}(\tilde{\boldsymbol{a}}_k; \boldsymbol{\mu}_{\tilde{\boldsymbol{a}}_k}, \boldsymbol{\Sigma}_{\tilde{\boldsymbol{a}}_k})$ with

$$\boldsymbol{\Sigma}_{\tilde{\boldsymbol{a}}_k} = \left(w_k^2 \sum_{n=1}^N \frac{\boldsymbol{x}_n\boldsymbol{x}_n^T}{\lambda_n} + \alpha_k \mathbf{I}_P\right)^{-1},$$

$$\boldsymbol{\mu}_{\tilde{\boldsymbol{a}}_k} = \boldsymbol{\Sigma}_{\tilde{\boldsymbol{a}}_k} w_k \sum_{n=1}^N \left(\frac{\xi_n^{-k} + \lambda_n}{\lambda_n} y_n \boldsymbol{x}_n\right). \tag{12}$$

For clarity, we use $\tilde{\mathbf{A}}$ and $\tilde{\boldsymbol{a}}$ in this model. In order to clearly illustrate the influence of the intermediate representation on the inference, we integrate out z_n in Eq. (6) to get the collapsed posterior distribution of $\boldsymbol{a}_k$, $p(\boldsymbol{a}_k|-) = \mathcal{N}(\boldsymbol{a}_k; \boldsymbol{\mu}_{\boldsymbol{a}_k}, \boldsymbol{\Sigma}_{\boldsymbol{a}_k})$

$$\boldsymbol{\Sigma}_{\boldsymbol{a}_k} = \left(w_k^2 \sum_{n=1}^N \frac{\boldsymbol{x}_n\boldsymbol{x}_n^T}{\lambda_n + w_k^2} + \alpha_k \mathbf{I}_P\right)^{-1},$$

$$\boldsymbol{\mu}_{\boldsymbol{a}_k} = \boldsymbol{\Sigma}_{\boldsymbol{a}_k} w_k \sum_{n=1}^N \left(\frac{\xi_n^{-k} + \lambda_n}{\lambda_n + w_k^2} y_n \boldsymbol{x}_n\right). \tag{13}$$

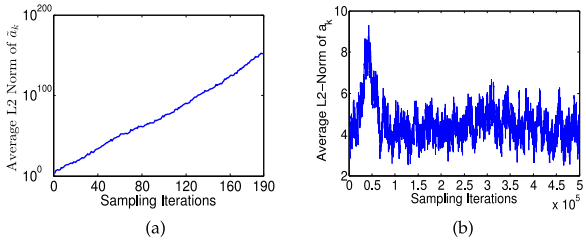


Fig. 1. The variation of average L_2-norm of $\boldsymbol{a}_k$ with the increase of sampling iterations. (a) Without z_n; (b) with z_n.

The detailed derivation is deferred to supplementary material, available online. Comparing Eqs. (12) and (13), we can see that the difference lies in the denominators, where w_k^2 is brought in by z_n. Intuitively, w_k^2 seems to play a role of balancing the power between projection matrix and classifier coefficients to prevent them too large or small. Fig. 1a and 1b describe the variation of average L_2-norm of $\boldsymbol{a}_k$, $\frac{1}{M}\sum_{k=1}^M \|\boldsymbol{a}_k\|_2$, with the increase of sampling iterations. Apparently, under the help of the intermediate latent representation z_n, the amplitude of $\boldsymbol{a}_k$ walks around in a normal range, but without z_n, $\tilde{\boldsymbol{a}}_k$ goes to infinity very quickly which leads to the inference collapse.

In addition, different from the common form of the probabilistic latent-variable model, which deploys the decoder-type decomposition [11], [44], our model directly uses data projection to construct the projected features via the encoder [21]. Namely, the encoder provides a bottom-up mapping from the input to latent feature space while the decoder maps the latent features back to the input, hopefully giving a reconstruction close to the original input [45]. On one hand, our objective is to find a discriminative subspace rather than the reconstruction. On the other hand, using decoder to get the latent representation is relatively difficult, because it requires solving an inference problem with multiple elements in the latent features competing to explain each part of the input. Furthermore, an encoder can perform faster inference to compute the latent representation at test time.

4.3 Kernel Max-Margin Discriminant Projection (KMMDP)

For data with complicated multimodal distributions, linear transformation is unlikely to be sufficiently flexible to reveal useful structures. In this section, we follow the approaches of kernel PCA and kernel LDA (KLDA) to kernelize the above linear feature extractor [25], [31]. That is, we employ some kernel function to implicitly perform a nonlinear mapping Φ to transform the data into a feature space $\mathcal{F}$, in which the linear projection is applied.

Suppose that the euclidean space $\mathbb{R}^P$ is mapped to a Hilbert space $\mathcal{F}$ through a nonlinear mapping function Φ: $\mathbb{R}^P \rightarrow \mathcal{F}$. Let $\Phi(\boldsymbol{x}_n)$ denote the data in the Hilbert space. So Eq. (5) in the Hilbert space can be written as follows:

$$z_n \sim \mathcal{N}(\mathbf{A}^T\Phi(\boldsymbol{x}_n), \mathbf{I}_K). \tag{14}$$

Since the dimensionality of $\Phi(\boldsymbol{x}_n)$ may be infinite, it is not possible to directly put a multivariate Normal distribution on each column of projection matrix, $\boldsymbol{a}_k$, that is the reason

why factor analysis model is very difficult to kernelize. Instead, under the assumption that any vector $\boldsymbol{a}_k$ in the Hilbert space lies in the span of a set of data vectors, $\Phi(\mathbf{V}) = [\Phi(\boldsymbol{v}_1), \Phi(\boldsymbol{v}_2), \ldots, \Phi(\boldsymbol{v}_M)]$, which are usually called basis vectors [3], [4], [31], we can find the corresponding coefficients $\boldsymbol{\theta}_k \in \mathbb{R}^M$, $k = 1, 2, \ldots, K$ for each $\boldsymbol{a}_k$

$$\boldsymbol{a}_k = \Phi(\mathbf{V})\boldsymbol{\theta}_k. \tag{15}$$

To generalize MMLDP to the nonlinear case, we formulate it in a way that uses dot product. Concretely, we consider an expression of dot product on the Hilbert space $\mathcal{F}$ given by the following kernel function:

$$K(\boldsymbol{x}_n, \boldsymbol{x}_m) = \langle \Phi(\boldsymbol{x}_n), \Phi(\boldsymbol{x}_m) \rangle = \Phi(\boldsymbol{x}_n)^T \Phi(\boldsymbol{x}_m), \tag{16}$$

where $\langle \cdot, \cdot \rangle$ represents the dot product. By means of Eqs. (15) and (16), Eq. (14) can be rewritten as

$$z_n \sim \mathcal{N}(\boldsymbol{\Theta}^T K(\mathbf{V}, \boldsymbol{x}_n), \mathbf{I}_K), \tag{17}$$

where $K(\mathbf{V}, \boldsymbol{x}_n) = [K(\boldsymbol{v}_1, \boldsymbol{x}_n), K(\boldsymbol{v}_2, \boldsymbol{x}_n), \ldots, K(\boldsymbol{v}_M, \boldsymbol{x}_n)]^T$ and $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_K]$. The inference of each projection coordinate is equivalent to learning the corresponding coefficients $\boldsymbol{\theta}_k$. Interestingly, from the other point of view, we can regard Eq. (17) as a way to find a linear subspace embedding of nonlinearly transformed $\mathbf{X}$ [37]. The kernel encoder $K(\mathbf{V}, \boldsymbol{x}_n)$ is equivalent to a nonlinear transformation $h(\boldsymbol{x}_n) \in \mathbb{R}^M$ and $\boldsymbol{\Theta}$ is the linear projection that we would like to infer.

We let each $\boldsymbol{\theta}_k$ follow the multivariate Normal distribution

$$\boldsymbol{\theta}_k \sim \mathcal{N}(0, \alpha_k^{-1}\mathbf{I}_M), \qquad \alpha_k \sim \mathrm{Ga}(c_0, d_0). \tag{18}$$

As a result, we have the pseudo-posterior of a kernelized max-margin discriminant projection

$$\begin{aligned} &p(\boldsymbol{\Theta}, \mathbf{Z}, \boldsymbol{w}, \boldsymbol{\lambda}, \boldsymbol{\alpha}, \sigma | \boldsymbol{y}, \mathbf{V}) \\ &\quad \propto \prod_{n=1}^{N} p(\boldsymbol{z}_n | \boldsymbol{\Theta}, \mathbf{V}, \boldsymbol{x}_n) \phi(y_n, \lambda_n | \boldsymbol{w}) \\ &\qquad \times p(\boldsymbol{w}|\sigma) p(\sigma) \prod_{k=1}^{K} p(\boldsymbol{\theta}_k | \alpha_k) p(\alpha_k). \end{aligned} \tag{19}$$

Since KMMDP is different from MMLDP by the coefficients $\boldsymbol{\Theta}$ and a predefined kernel function in terms of the formulation, we only give the posterior distribution of each $\boldsymbol{\theta}_k$, $p(\boldsymbol{\theta}_k | -) = \mathcal{N}(\boldsymbol{\theta}_k; \boldsymbol{\mu}_\theta, \boldsymbol{\Sigma}_\theta)$

$$\begin{aligned} \boldsymbol{\Sigma}_\theta &= \left(\alpha_k \mathbf{I}_N + \sum_{n=1}^{N} \boldsymbol{K}(\mathbf{V}, \boldsymbol{x}_n) \boldsymbol{K}(\mathbf{V}, \boldsymbol{x}_n)^T \right)^{-1}, \\ \boldsymbol{\mu}_\theta &= \boldsymbol{\Sigma}_\theta \left(\sum_{n=1}^{N} z_{kn} \boldsymbol{K}(\mathbf{V}, \boldsymbol{x}_n) \right). \end{aligned} \tag{20}$$

It is remarkable that the proposed kernel supervised feature extractor is different from kernel PCA plus linear SVM (LiSVM) as mentioned in [31], since our model jointly learns the discriminative subspace and the linear classifier in the feature space rather than separately. Furthermore, in our method we do not require our kernel as a centerized Gram matrix.

From Eqs. (20) and (15), we can see that the BV plays an important role in the model, which dominates the computational complexity of the inference and may influence the classification performance [1]. In the next section, we will present a way to infer them.

4.4 Modeling the Sparsity in Kernel Expansion via Spike-and-Slab Prior (ssKMMDP)

Currently, there are two choices for BVs: one is to use all or a randomly selected set of observed data samples as BVs, which is employed by most of kernel methods, such as kernel SVMs, kernel PCA, kernel LDA and so on; the other is to directly learn BVs through minimizing some reconstruction error [3]. But they face several difficulties: (i) since kernel methods are involved in an eigen-decompostion or kernel matrix inversion [4], [31], they scale cubically in the number of selected BVs, M; (ii) for large N, random selection may not be accurate enough while it will be expensive as increasing M for good accuracy; (iii) both types of methods learn BVs offline to best characterize the space of the data set, independent of the final task [3], [4], [31]. In this section, we present a way, falling into the first type of method, to select BVs of interest from a set of candidates.

In order to analyze what are suited well to span the subspace in the feature space, we will extend our nonlinear model with the help of spike-and-slab [5] prior to adaptively find BVs. Specifically, we put the spike-and-slab prior on the coefficients matrix ($\boldsymbol{\Theta}$), since the associations among latent features and BVs are sparse. (Note, placing the spike-and-slab prior on the kernel matrix will bring trouble, because it leads to BVs data-dependent and unsuitable for realtime decision making at test phase.) Recall the coefficient matrix $\boldsymbol{\Theta} \in \mathbb{R}^{r \times K}$, where r defines the number of BVs responsible for the kth component and it is not known in general, and needs inferring. Within the analysis we will consider M candidates (M rows of $\boldsymbol{\Theta}$), with M set to a value anticipated to be large relative to r. We then infer the number of rows of $\boldsymbol{\Theta}$ needed to represent each component in the feature space, with this number used as an estimate of r. Finally, we will infer a posterior density function on r. Lets assume $\boldsymbol{\Theta}$ has M rows, with the understanding that we wish to infer the $r < M$ rows that are actually needed to represent the kernel components. We employ the spike-and-slab prior to define sparseness in $\boldsymbol{\theta}_k$

$$\begin{aligned} &\theta_{lk} \sim (1 - \pi_l)\delta_0 + \pi_l \mathcal{N}\left(0, \alpha_k^{-1}\right), \\ &\pi_l \sim \mathbf{Beta}(e_0, f_0), \quad \alpha_k \sim \mathbf{Ga}(c_0, d_0), \end{aligned} \tag{21}$$

where (e_0, f_0) are selected as to strongly favor $\pi_l \to 0$, δ_0 is a distribution concentrated as zero, and $l = 1, \ldots, M$. The advantage of Eq. (21) is that sparseness is imposed explicitly (many elements of $\boldsymbol{\theta}_k$ are exactly zero). Each π_l represents the global potential of the lth candidate selected by all of components. Then we have the following conditional posterior distributions for $\boldsymbol{\theta}_k$, α_k and π_l [20]:

TABLE 2
Computational Complexities of MMLDP, KMMDP, and ssKMMDP at Each Iteration

Parameter	$\mathbf{A}$	$\mathbf{Z}$	w	$\boldsymbol{\lambda}$	$\boldsymbol{\alpha}$	σ	Θ	$\boldsymbol{\pi}$
MMLDP	$\mathcal{O}(KP^3 + NPK)$	$\mathcal{O}(NPK)$	$\mathcal{O}(K^3 + NK^2)$	$\mathcal{O}(NK)$	$\mathcal{O}(KP)$	$\mathcal{O}(K)$	—	—
KMMDP	$\mathcal{O}(KM^3 + NMK)$	$\mathcal{O}(NMK)$	$\mathcal{O}(K^3 + NK^2)$	$\mathcal{O}(NK)$	$\mathcal{O}(KM)$	$\mathcal{O}(K)$	—	—
ssKMMDP	$\mathcal{O}(KM^3 + NMK)$	$\mathcal{O}(NMK)$	$\mathcal{O}(K^3 + NK^2)$	$\mathcal{O}(NK)$	$\mathcal{O}(KM)$	$\mathcal{O}(K)$	$\mathcal{O}(NKM)$	$\mathcal{O}(KM)$

- $p(\boldsymbol{\theta}_{lk}|-) = (1-\tilde{\pi}_{lk})\delta_0 + \tilde{\pi}_{lk}\mathcal{N}(\mu_{\theta_{lk}}, \Sigma_{\theta_{lk}})$, where

$$\Sigma_{\theta_{lk}} = \left(\alpha_k + \sum_{n=1}^{N} K(\boldsymbol{v}_l, \boldsymbol{x}_n)^2\right)^{-1},$$
$$\mu_{\theta_{lk}} = \Sigma_{\theta_{lk}} \sum_{n=1}^{N} z_{kn}^{(l)} K(\boldsymbol{v}_l, \boldsymbol{x}_n), \tag{22}$$
$$\frac{\tilde{\pi}_{lk}}{1-\tilde{\pi}_{lk}} = \frac{\pi_l}{1-\pi_l} \frac{\mathcal{N}\left(0|0, \alpha_k^{-1}\right)}{\mathcal{N}\left(0|\mu_{\theta_{lk}}, \Sigma_{\theta_{lk}}\right)},$$

  where $z_{kn}^{(l)} = z_{kn} - \sum_{m=1, m\neq l}^{M} \theta_{mk} K(\boldsymbol{v}_m, \boldsymbol{x}_n)$.
- $p(\alpha_k|-) = \mathrm{Ga}(c_0 + 1/2\sum_{m=1}^{M} 1(\theta_{mk} \neq 0), d_0 + 1/2\sum_{m=1}^{M} \theta_{mk}^2)$, where $1(x) = 1$ if x is true and 0 otherwise.
- $p(\pi_l|-) = \mathrm{Beta}(e_0 + \sum_{k=1}^{K} 1(\theta_{lk} \neq 0), f_0 + \sum_{k=1}^{K} 1(\theta_{lk} = 0))$.

4.5 Prediction

For the linear projection model, after the burn-in stage we can average over all of the collected samples from Gibbs sampler to predict the label of a new data $\boldsymbol{x}^*$ as below

$$y^* = \mathrm{sign}\left(\frac{1}{T - T_{\mathrm{burn-in}}} \sum_{t=T_{\mathrm{burn-in}}+1}^{T} \boldsymbol{w}_t^T \boldsymbol{z}_t^*\right), \tag{23}$$

where $\{z_t^* = \mathbf{A}_t^T \boldsymbol{x}^*\}_{t=T_{\mathrm{burn-in}}+1}^{T}$ is drawn from its corresponding posterior based on $T - T_{\mathrm{burn-in}}$ collected samples. Thus, instead of integrating the latent variables, we average the final outputs from the max-margin classifier. For the nonlinear projection model, we get the similar prediction form except for the latent representation $\{z_t^* = \boldsymbol{\theta}_t^T K(\mathbf{V}, \boldsymbol{x}^*)\}_{t=T_{\mathrm{burn-in}}+1}^{T}$. It is worth of noticing that in the Bayesian factor analysis, it is not easy to average the random variables due to the exchangeability issue. Usually one selects a most likely sample as a point estimate, which may not make use of integration power in Gibbs sampling method. Fortunately, since our model is built on latent max-margin representation and optimized for the final classification task, what we are principled interested in is the output for each data sample which is a scalar as Eq. (23). Therefore, it is convenient to get the average of all the outputs from the classifier after burn-in, which is able to boost the performance.

Although the basic SVM classifier is for binary classification, there have been several strategies to realize multi-class classification via binary classification, such as one-vs-all and one-vs-one. In this paper, we choose one-vs-all strategy, since its effectiveness has been analyzed and proved in [30] theoretically and experimentally. We have also tried the multi-class SVM strategy proposed by Crammer and Singer [13] , which produces the similar performance and running time with the one-vs-all scheme. Suppose we have L different classes. When it is desired to classify a new example, the classifier which outputs the largest (most positive) value is chosen. If we define $U_l^* = \frac{1}{T - T_{\mathrm{burn-in}}} \sum_{t=T_{\mathrm{burn-in}}+1}^{T} (\boldsymbol{w}_t^l)^T \boldsymbol{z}_t^*$ as the output from the lth classifier after the burn-in stage, the discriminate decision function can be written as

$$y^* = \underset{l=1,\ldots,L}{\mathrm{argmax}} \left(U_l^*\right). \tag{24}$$

We put the detailed implementation of multi-class problem in Supplementary Material, available online.

4.6 Computational Complexity

In Table 2, we list the per-iteration complexity of each parameter in MMLDP, KMMDP and ssKMMDP respectively. For MMLDP, to sample the projection matrix $\mathbf{A}$ column by column (Step 5), we need inverse a $P \times P$ matrix, which makes MMLDP not practical for high-dimensional data. For other parameters, inferring them are not dominant since $K \ll P$. For KMMDP, we can precalculate the kernel matrix between data and BVs offline, which costs $\mathcal{O}(NMP)$. But from Eq. (20), we can see that the most time-consuming step of KMMDP, $\mathcal{O}(KM^3 + NMK)$, is dependent on the number of BVs and independent of the dimensionality of original data. Accordingly, KMMDP is also a way to deal with very high dimensional data. Considering ssKMMDP, because the elements of the binary matrix are coupled each other, we have to update them one by one via *for* loop and cannot use the efficient matrix-vector calculation. Moreover, in ssKMMDP, M is usually truncated at a large value, consequently, it takes longer time in the training phase relative to KMMDP.

5 Experimental Results

5.1 Parameter Setting

While the hierarchical model construction may appear relatively complex, the number of parameters that need to be set is not particularly large, and they are set in a noninformative way [26], [34], [46]. Specifically, for the spike-and-slab model $e_0 = 1$ and $f_0 = 1$, and for the gamma distributions $a_0 = b_0 = 1$ and $c_0 = d_0 = 1$. The same parameters were used in all examples unless otherwise noted. The following feature extraction methods are compared: (1) Linear SVM; (2) Gaussian Kernel SVM (GKSVM); (3) Polynomial Kernel SVM (PKSVM); (4) LDA; (5) Local Discriminant Embedding; (6) Sparse Discriminant Analysis; (7) Kernel LDA; (8) Kernel LDE (KLDE); (9) Infinite Latent SVM; (10) Bayesian Supervised Dimension Reduction; (11) MMLDP; (12) Kernel MMDP (KMMDP); (13) KMMDP with Spike-and-Slab Prior (ssKMMDP). For LDA and KLDA, since the rank of its between-class scatter matrix is at most $N_c - 1$ [25], for the

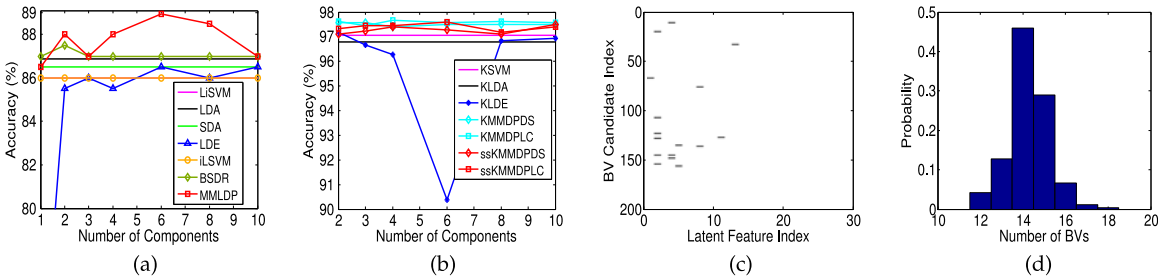


Fig. 2. The performance comparison on two synthesized data sets. (a) Test accuracy for linear case; (b) Test accuracy for nonlinear case; (c) BV candidate usage (black means used, white means not used); (d) approximate posterior distribution on the number of BVs.

sake of convenience, we always set the number of their components to $N_c - 1$, where N_c is the number of classes. Except BSDR and our methods, for the fair comparison, the features extracted by other methods are fed into the latent SVM implementation. All the algorithms are implemented in MATLAB and run on an Intel Core i7-3520M 2.90 GHz CPU with 8.00 GB of RAM.

5.2 Synthesized Example

We first consider two synthesized data. The first data includes two overlapping classes and 400 samples, each consisting of two Gaussians. The second data set includes three classes and nine local clusters, $3{,}600$ samples totally. The three classes are linearly nonseparable in general. The details about these data are included in Supplementary Material, available online. We average the accuracy over ten random $50/50$ splits of the training and test data. We consider $20{,}000$ Gibbs iterations, with the first burn-in $10{,}000$ samples discarded and every tenth sample collected afterwards. We ran this many samples because of the computational efficiency for this problem; good results are obtained with far fewer samples.

In the first case, we only tested linear methods and list the accuracies in Fig. 2a. We found several phenomena as follows: 1. The fact that LDA and SDA used only one component in this binary classification case, leads to their performance inferior to LDE, which employs an improved Fisher criterion and can extract more feature components; 2. SDA worked even worse than LDA and we analyze that in SDA the feature selection step may result in information loss, since there is no redundant feature in that case; 3. BSDR and MMLDP were leading other methods. We attributed that to the fact that both of them take advantage of the joint learning framework to construct the discriminative feature space for the classifier rather than separately, while thanks to the max-margin criterion and efficient Gibbs sampler, MMLDP yielded the best performance; 4. iLSVM did not work satisfyingly in spite of using the similar classifier and joint framework, because it only learns a binary latent representation for data construction and classification, which barriers its performance.

On the second data, we aim to evaluate different kernel methods. Meanwhile, we would like to compare the influences of two types of BV candidates on the classification performance of ssKMMDP, e.g., local cluster centers and a random subset of training samples. ssKMMDP with local centers is indicated as ssKMMDPLC and with data samples as ssKMMDPDS. The number of BV candidates was truncated as 200, covering all of local regions for fair comparison. We set the number of latent components as 30 and utilized Gaussian kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2)$ with $\gamma = 0.01$. For KLDA and KLDE, we varied γ to show the best results. Fig. 2b shows the test accuracies on different numbers of features extracted by the various algorithms. For visualization, the distribution of data in different feature spaces constructed by different approaches were depicted in Fig. 3. Apparently, in this nonlinear case our methods outperformed others. ssKMMDP was comparable to KMMDP and ssKMMDPLC slightly better than ssKMMDPDS. Relatively, ssKMMDPLC gave more parsimonious number of BVs. In detail, ssKMMDPLC got 14 BVs, while ssKMMDPDS 25 BVs. We attributed that to the fact that usually the local centers are more stable and represent many neighboring data samples, so it is more helpful for the model to make choice. In some real applications, local cluster centers have revealed better robustness to noise compared to the single sample [14]. To investigate the role of spike-and-slab prior in kernel expansion, we plot the underlying candidate usage matrix in Fig. 2d. For simplicity, we only show the results got by ssKMMDP with local centers. Surprisingly, the candidate usage matrix is highly sparse and each latent feature is only involved few candidates, which further supports that it is too redundant to use all of data samples as BVs. More interestingly, from Fig. 2c

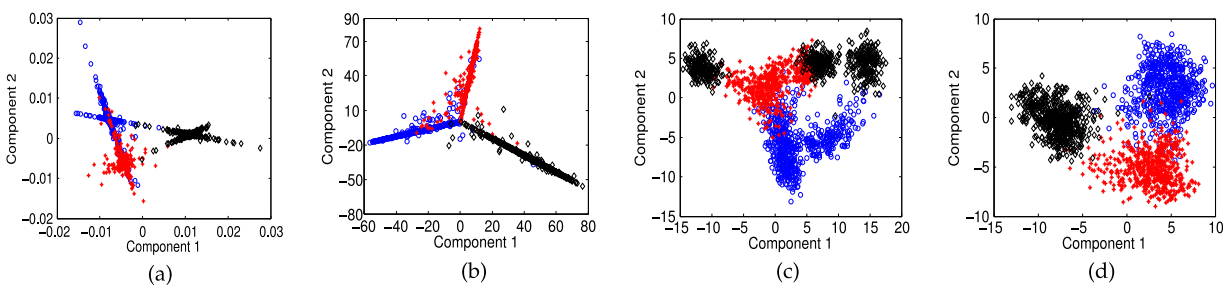


Fig. 3. Visualize the second synthesized data by different methods. (a) KLDA; (b) KLDE; (c) ssKMMDPDS; (d) ssKMMDPLC.

TABLE 3
Mean Classification Accuracies (Percent) and Corresponding Standard Deviations (Percent) for Different Methods on Benchmark Data

Dataset	Diabiets	Heart	Ringnorm	Waveform	German	Satimage	Arcene	Dorothea	Leukemia	Isolete	DNA	USPS
LiSVM	72.2(2.1)	83.7(3.7)	**77.6(0.5)**	88.0(0.8)	67.9(1.0)	74.6(0.6)	68.7(4.1)	62.3(3.0)	97.9(2.5)	**93.9(0.8)**	94.2(0.6)	92.9(0.5)
LDA	72.4(1.6)	85.6(3.3)	76.6(0.3)	88.2(0.5)	68.4(0.9)	74.8(0.8)	67.3(3.9)	51.0(2.5)	92.1(2.6)	92.0(1.1)	93.5(0.5)	89.5(0.7)
SDA	74.4(1.3)	83.5(3.3)	76.7(0.4)	84.7(0.6)	69.6(1.9)	71.9(0.8)	**71.1(3.8)**	**90.1(1.3)**	95.7(2.7)	80.7(1.2)	93.5(0.5)	89.1(0.5)
LDE	73.8(2.0)	82.9(3.6)	76.7(0.4)	**88.6(0.5)**	69.9(1.1)	74.0(0.8)	—	—	96.4(2.6)	93.4(1.1)	**94.8(1.8)**	**94.8(0.5)**
iLSVM	73.1(2.0)	82.5(3.0)	75.8(0.4)	83.8(1.1)	68.7(2.3)	75.7(1.2)	62.7(3.9)	75.8(3.1)	91.1(2.0)	79.5(1.4)	91.1(0.9)	87.2(1.0)
BSDR	76.9(1.2)	85.6(3.6)	76.0(0.3)	88.2(0.3)	70.3(1.2)	73.8(0.6)	—	—	—	84.7(0.9)	89.7(0.7)	89.3(0.9)
MMLDP	**77.7(1.9)**	**86.1(3.6)**	76.3(0.4)	88.2(0.7)	**70.6(1.1)**	**76.0(0.9)**	—	—	**98.7(2.0)**	93.7(0.8)	94.2(0.9)	93.6(0.6)
GKSVM	79.4(1.8)	86.0(3.5)	97.5(0.2)	88.9(0.4)	77.1(1.8)	86.6(0.7)	50.0(0.0)	50.0(0.0)	96.2(2.0)	90.4(0.5)	94.7(0.1)	93.9(0.5)
PKSVM	78.9(2.3)	85.7(3.3)	96.8(0.2)	88.4(0.4)	76.5(2.3)	**89.6(0.6)**	86.0(3.8)	75.9(3.2)	96.2(1.9)	89.6(0.4)	92.7(1)	93.7(0.7)
GKLDA	78.5(2.1)	87.0(3.5)	91.0(0.3)	87.5(0.1)	76.0(1.8)	88.8(1.1)	50.0(0.0)	50.0(0.0)	94.3(2.3)	94.3(1.0)	94.0(0.8)	96.0(1.0)
PKLDA	79.4(2.2)	83.6(3.6)	90.0(0.4)	87.6(0.6)	75.2(2.0)	87.9(1.1)	70.5(3.8)	80.1(2.7)	96.4(2.1)	94.1(0.4)	83.0(1.4)	90.8(1.2)
GKLDE	77.7(1.8)	85.6(3.0)	96.1(0.3)	88.6(0.4)	76.9(1.9)	88.7(1.1)	50.0(0.0)	50.0(0.0)	95.0(2.0)	**95.6(1.0)**	**95.8(0.9)**	96.1(1.1)
PKLDE	79.7(2.0)	87.3(3.6)	77.7(0.2)	88.0(0.3)	75.5(1.9)	86.1(1.2)	65.8(3.9)	75.0(3.1)	96.4(2.2)	95.2(0.9)	95.7(1.0)	90.4(1.0)
GKMMDP	80.1(1.7)	88.2(3.7)	**98.8(0.3)**	**89.7(0.6)**	78.2(1.6)	88.7(0.7)	72.4(4.0)	89.7(2.7)	96.7(2.3)	94.7(0.5)	94.4(0.8)	**96.8(0.8)**
PKMMDP	78.2(2.2)	85.2(3.3)	97.5(0.1)	88.5(0.7)	**79.0(1.1)**	86.3(0.8)	81.7(4.2)	**90.8(2.6)**	**97.4(2.2)**	92.2(0.6)	94.8(0.7)	95.8(0.6)
ssGKMMDP	**80.8(1.9)**	**88.3(3.3)**	98.0(0.3)	89.1(0.8)	77.9(2.3)	**89.6(0.9)**	70.8(4.1)	84.7(2.9)	93.2(2.3)	94.8(0.8)	91.0(0.9)	96.1(0.8)
ssPKMMDP	80.0(2.0)	88.1(3.2)	97.7(0.2)	87.9(0.8)	72.5(2.7)	86.1(0.9)	**82.4(4.2)**	88.9(3.2)	93.6(2.1)	93.4(0.8)	95.1(0.8)	95.2(0.7)

Bold fonts highlight linear and nonlinear methods respectively, that were statistically the best.

we also found that besides the rows (candidates) there are some entire columns (latent features) being 0's in the usage matrix. This phenomenon never happens in the standard spike-and-slab applications. It is attributed to the fact that in our model the binary matrix reflects the sparseness between latent features and BV candidates, and both of them might be inactive, since they are all latent and only discriminative latent features are selected for the final prediction. Finally, ssKMMDPLC uses only 8 latent features out of 30 for classification. Since local cluster centers have shown the good sparsity and their size is also far smaller than that of observed data samples, for simplicity and clarity we only display the results of ssKMMDP based on local cluster centers in the experimental section below.

5.3 Benchmark Data Sets

In this section, we perform experiments on 12 benchmark and real-world data sets of varying size and difficulty, and for each we average the accuracy over 50 random splits, 70 percent for training and 30 percent for test with preserving the data ratio of different classes unchanged, except for DNA, Satimage, Isolet and USPS, which were run twenty splits. (The data details are shown in Supplementary Material, available online.) The benchmark data sets can be found either at the UCI or Machine Learning Dataset Repository. For kernel methods, we tried Gaussian and Polynomial kernel $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i' \boldsymbol{x}_j + 1)^d$. The kernel parameter γ is chosen from $\{0.01, 0.1, 1, 10\}$ and d from $\{1, 2, 3, 4, 5, 6\}$ respectively for different methods to get the best performance. For clarity, we briefly denote KSVM, KLDA, KLDE, KMMDP, ssKMMDP with Gaussian kernel as GKSVM, GKLDA, GKLDE, GKMMDP and ssGKMMDP, while those with polynomial kernel as PKSVM, PKLDA, PKLDE, PKMMDP and ssPKMMDP. In our methods, we truncated the number of BV candidates as 200. All of data sets were first normalized to a distribution with zero mean and unity variance in every feature direction.

Table 3 reports the mean and standard deviations of testing accuracies, where for each method only the best over different components are list. (We also plot the variations of accuracies on different numbers of components in Supplementary Material, available online.) For those data with the dimensionality $P > 1{,}000$, we did not test LiSVM, LDE, BSDR and MMLDP, since they are too time-consuming for large P problems. In general, nonlinear methods worked better than linear ones. Although SDA was still inferior to LDE and LDA, it was very efficient on the high-dimensional data, where SDA kept working fast and leading other linear methods, which shows the importance of feature pruning for the high-dimensional problem. LDE also gave leading results on some data, since it leverages the graph matrix to consider the separability locally rather than globally, which is similar to mixture models. However, in LDE the neighboring graph matrix is built in the original space, which only focuses on the local regions and cannot theoretically guarantee the distant samples from different classes still living far away and the samples from same classes close to each other in the new lower-dimensional space. Though LiSVM and kernel SVMs were also very competitive, they employed all of features. Both MMLDP and KMMDP have achieved better accuracies than their counterparts on most of data, and KMMDP worked more stable with both Gaussian and polynomial kernels. We attributed that to the fact that our models, besides the nice max-margin criterion, keep adjusting the discriminative feature subspace in each iteration according to the classifier under the joint learning framework instead of separately, which is more suitable for the classification. We also found that the accuracy is dependent on the selection of kernels and choosing the unsuitable kernel may hurt the final prediction, even though the kernel parameters are carefully tuned. Especially on the high-dimensional data sets, such as Arcene, Dorothea and Leukemia, nonlinear methods with polynomial kernel have yielded better performance than those with Gaussian kernel, some of which even completely fail on those four data

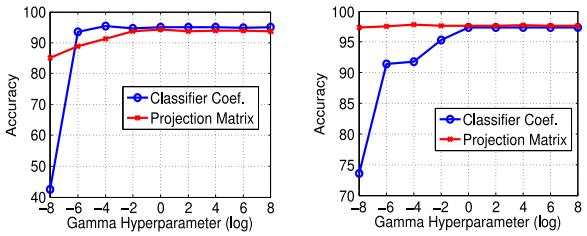


Fig. 4. The effect of different Gamma hyperparameters, b_0 and d_0. (a) DNA; (b) ringnorm

sets. In addition, ssKMMDP is comparable to KMMDP on different kernels with sparser usage of BVs in kernel expansion. We also did some experiments to evaluate the impacts of the number of BVs on classification performance with different kernels and investigated whether the spike-and-slab prior is helpful in our model. Due to the limited space, those results are included in Supplementary Material, available online. It turns out that ssKMMDP is competitive to KMMDP in terms of classification accuracy, although ssKMMDP has explored the sparsity in kernel expansion and only used fewer BVs, especially for Gaussian kernel.

Since the Gamma hyperparameters, b_0 and d_0, may indirectly influence the inference, e.g., classifier coefficients ($\boldsymbol{w}$) and projection matrix ($\mathbf{A}$). To evaluate how the accuracy performance can be effected by varying those parameters, we selected two typical data sets, which have different ratios (ρ) of feature dimensionality to the number of training samples and different number of classes, e.g., DNA and Ringnorm. As shown in Fig. 4, the sparsity of projection matrix is not crucial, while the classification performance is more related to the classifier coefficients.

We now examine the computational costs of the different computational methods on different data, which includes two parts, feature extractor and classifier. For all the methods using Bayesian max-margin classifier, we consider 200 samples with 100 burn-ins and 100 collections. For BSDR, following the setting in the original paper [16], we set the number of VB iterations as 100. For fair comparison, the number of latent components was fixed as $N_c - 1$, which is equal to that extracted by LDA. In KMMDP and ssKMMDP, we employ the minimum of the 20 percent of the training set and 100 as BVs. The results from this experiment are summarized in Table 4, with comparison to several results on bechmark data sets. In terms of the ratio (ρ), we divide those benchmark data sets into three groups: $\rho < 0.01$, $0.01 < \rho < 10$ and $\rho > 10$. As described in Table 2, among linear methods SDA performed most efficient especially on high-dimensional data. When $\rho < 0.01$, MMLDP has the similar computational complexity with LDA and its variants, while BSDR and iLSVM spends more time. Using a subset of the training samples as BVs, KMMDP is clearly beneficial in terms of computational cost, especially when $N \gg M$ and $N \gg K$. KLDA and KLDE have the dominant complexity of $\mathcal{O}(N^3)$, moreover, KLDE has to calculate and save two graph matrices, which is not appropriate for large scale problem. The other hand, when handling large "p", small "n" problem ($\rho > 10$), kernel methods have shown more advantages than their corresponding linear versions concerning the computational complexity. Due to the inference of BV usage matrix element by element, ssKMMDP is slower than KMMDP

5.4 Caltech 101

We also tested our methods on the well-known multiclass computer vision data set: Caltech101 [22] , which contains $9{,}144$ images from 101 object categories and a background category. Following the common experimental setting, we train models on $15/30$ images per category and test on the remaining images. All images are transformed into grayscale form. We followed the strategy in [42] and

TABLE 4
Comparison of CPU Time (Second) of Different Algorithms on 15 Benchmark Data

Dataset	LiSVM	LDA	SDA	LDE	BSDR	iLSVM	MMLDP	KLDA	KLDE	KMMDP	ssKMMDP
Ringnorm	1.57	0.55	0.64	34.04	9.64	11.19	0.86	176.86	—	1.50	4.71
Satimage	1.72	2.66	5.95	79.27	23.73	87.37	6.06	536.70	—	9.22	39.90
USPS	132.87	4.67	266.77	78.25	25.65	271.71	26.90	710.61	—	15.29	51.90
Waveform	0.88	0.78	0.51	15.80	8.06	7.87	2.45	168.76	1820.00	1.95	14.04
Diabetis	0.21	0.25	0.23	0.58	0.27	1.30	0.23	0.42	5.38	0.30	0.78
German	0.36	0.20	0.22	0.89	0.38	1.86	0.28	0.37	13.07	0.34	0.84
Heart	0.18	0.15	0.16	0.27	0.17	0.65	0.19	0.18	0.36	0.21	0.34
DNA	0.55	0.57	5.69	9.25	3.00	17.68	2.02	13.78	534.19	1.34	4.49
Isolet	966.57	28.00	695.26	173.28	197.19	1,129.77	191.06	338.09	—	67.67	147.07
Arcene	—	152.12	1.36	—	—	8.38	—	0.19	0.17	0.18	0.21
Dorothea	—	928.71	24.51	—	—	643.52	—	8.02	7.81	0.40	0.71
Leukemia	327.51	391.25	0.64	45.19	—	2.17	0.65	0.38	0.18	0.29	0.60

"—" denotes the consumed time beyond 3,600 seconds.

TABLE 5
Mean Classification Accuracies (Percent) and Corresponding Standard Deviations (Percent) for Different Methods on Caltech101

Methods	LiSVM	GKSVM	PKSVM	GKLDA	PKLDA	GGKMMDP	PGKMMDP	ssGGKMMDP	ssPGKMMDP
15	65.5 (0.9)	59.9 (0.7)	58.8 (0.8)	64.1 (0.8)	65.1 (0.7)	65.5 (0.6)	65.3 (0.5)	61.5 (0.7)	**66.6 (0.8)**
30	72.8 (0.5)	71.1 (0.6)	71.3 (0.6)	70.2 (0.7)	72.4 (0.6)	72.1 (0.5)	**73.9 (0.5)**	71.8 (0.6)	73.7 (0.8)

Bold fonts highlight methods that were statistically the best.

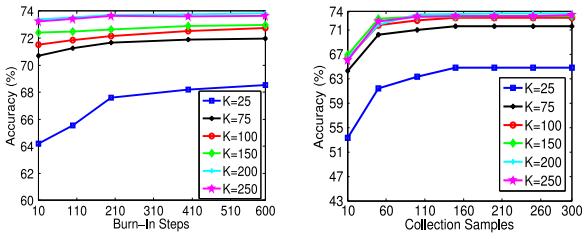


Fig. 5. Classification performance of the KMMDP with different Gibbs sampling settingups on Caltech101 data. (a) Different numbers of burn-in samples; (b) different numbers of collection samples.

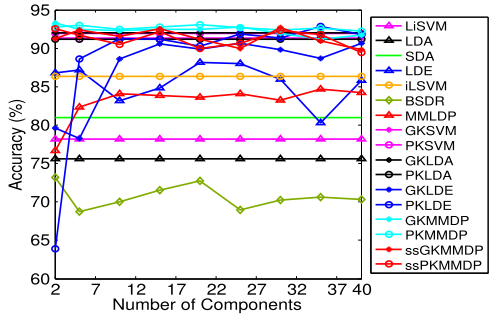


Fig. 7. Test accuracy on HRRP data.

transformed each image to a single $21{,}504$-dimesnional feature vectors as input features. Because of its high dimensionality, in the experiments we only consider KLDA, KMMDP and ssKMMDP with polynomial kernel and Gaussian kernel. We use 300 BVs for KMMDP, while for ssKMMDP the number of BV candidates were truncated as 500. The Beta hyperparameter e_0 in the spike-and-slab prior was set to 500 and $f_0 = 1$, so that enough BVs can be selected to represent the latent feature space, since the Caltech101 data is more complicated and large-scale than those data discussed above.

We show the mean accuracies and standard deviations of several methods in Table 5. (We also plot the mean accuracies as a function of the number of latent features in Supplementary Material, available online.) In both cases, kernel methods with polynomial kernel perform better than those with Gaussian kernel and PKMMDP still achieves the leading performance among all methods, 73.9 percent with 200 components. KLDA did not outperform SVM with the raw features, since it can only extract 101 components limited by its objective function. The best result on Caltech101 was obtained by combining multiple descriptor types. In this paper, we did not list those results, since our main goal is to analyze the strengths of KMMDP relative to other feature extraction methods rather than focus on the image classification task. Therefore, only SIFT descriptors are used for all compared algorithms.

As is known, when the Gibbs sampler is used to estimate posterior distribution [17], a key issue to its implementation is to determine when the procedure has essentially converged. The number of iterations needed for the burn-in procedure varies from cases to cases. In order to investigate the efficiency of the Gibbs sampler in our model, we test the classification performance by varying the numbers of burn-in and collection samples respectively. For the convenience, in this task we only use the kernel max-margin projection with polynomial kernel on 30 training images per class and fix the kernel parameter $d = 5$. Fig. 5a shows the classification accuracy of KMMDP with different numbers of burn-in samples for different numbers of components, where 200 samples are collected with the interval of ten iterations afterwards. We can see that when the number of burn-in samples is larger than 100, the performance has been quite stable, especially when the number of extracted components is large. In Fig. 5b, we present the classification accuracy of KMMDP with different numbers of collection samples for different numbers of components, where 100 burn-in samples are discarded at the beginning. It is apparent that after averaging 50 collection samples, the performance has converged to their stable values, especially when the number of components is large.

In our ssKMMDP, the spike-and-slab prior has been utilized to impose the sparseness on the coefficient matrix, $\mathbf{\Theta}$, in kernel expansion, which is actually l_0 penalized. In the above experiments, the analysis of the number of BVs has been discussed. In this section, we give a detailed examination of the impact of sparseness on various terms of such model. Specifically, we employ the Gibbs sampler and provide a detailed analysis on the effects of the hyperparameter, e_0, on sparseness and model performance of ssKMMDP with Gaussian kernel (ssGKMMDP) and polynomial kernel (ssPKMMDP) respectively. As indicated at the beginning of this section, with $f_0 = 1$, parameter e_0 (the slab strength parameter in Eq. (21)) controls sparseness on the number of BVs employed (via the probability of usage, defined by $\{\pi_l\}$). We fixed the number of extracted components as 200 and still employed 500 BV candidates. In Fig. 6, we presented variations of different model outputs with e_0, such as test accuracy, the number of BVs and the sparsity ratio in the latent usage matrix. These computations were trained on 30 images per category and tested on the remaining images. A wide range of the parameter yield similar good results, all favoring sparsity. As shown in Fig. 6, the

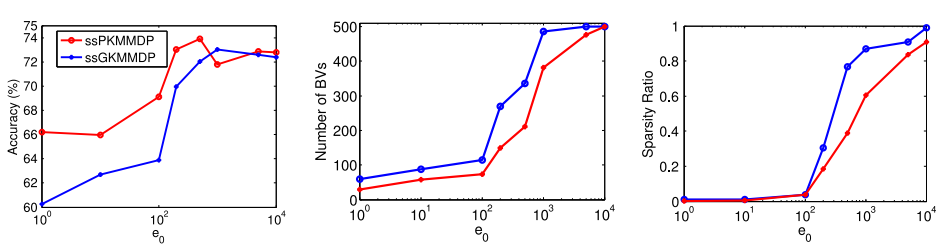


Fig. 6. The effects of the hyperparameter (e_0) in ssKMMDP with Gaussian kernel and Polynomial kernel on Caltech101 data. (a) Test accuracy; (b) the number of BVs; (c) sparsity ratio, where the ratio is calculated via $\frac{\#nonzero\ elements}{\#elements}$ and the lower the sparser.

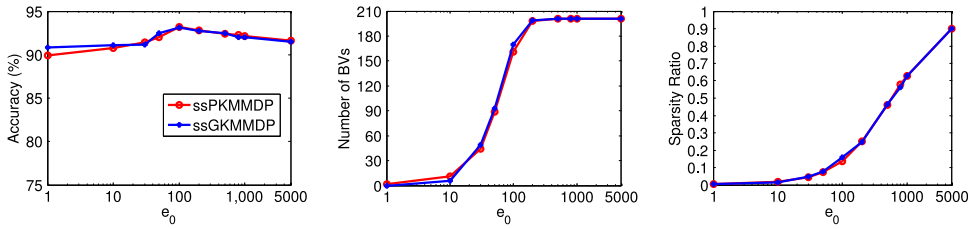


Fig. 8. The effects of the hyperparameter (e_0) in ssKMMDP with Gaussian kernel and Polynomial kernel on HRRP data. (a) Test accuracy; (b) the number of BVs; (c) sparsity ratio.

classification accuracies reach the top, when the matrix Θ has around 40 percent nonzero elements. And Gaussian kernel produced more parsimonious Θ, while polynomial kernel achieved better accuracies. Note that as e_0 increases, a denser use of BVs is encouraged, and as e_0 decreases the number of inferred BVs (in Fig. 6b) deceases as well.

5.5 Radar High-Resolution Range Profile Target Recognition

In this section we investigate how our methods work in a specific target recognition problem, radar HRRP automatic target recognition (ATR), for which it is much necessary to reduce system complexity and processing time. Especially, through pruning the BVs appearing at test phase, we hope to accelerate the decision procedure for the real-time processing demand.

The radar HRRP data of three airplanes, including An-26, Yark-42 and Cessna Citation S/II, were measured continuously when the targets were flying. The detailed parameters about radar and airplanes have been listed in Supplementary Material, available online. Training data (600 HRRPs) and test data (2,400 HRRPs) were taken from different data segments. The target orientations corresponding to the test and training data are different, thus the generalization performance of the recognition methods can be tested. In this experiment, the L_2-norm normalized power spectrum feature of HRRP was used to perform classification on account of its time-shift invariance. And each HRRP sample is a 128-dimensional vector. We consider 2,000 burn-in samples and 1,000 collection samples. Following the parameter setting-up discussed above, we report classification performance of all methods on this application in Fig. 7. According to the test accuracy, the proposed nonlinear methods still performed the best among all methods, but MMLDP achieves test accuracy comparable to LDE. We attribute that to the fact that those three airplanes are not linearly separable in the original feature space, which may not be fitted well by linear methods without considering local structure. Because of the similar reason, BSDR didnot work well, what is worse, BSDR inferred the parameters via variational Bayesian, which only provides an approximation to the posterior distribution. Relatively, concerning the performance variation across different numbers of extracted components, KMMDP is more stable than ssKMMDP due to the sparse effect in ssKMMDP. For the effect of hyperparameter e_0, in ssKMMDP, we did the similar experiments on this data as well as Caltech101. Again, Fig. 8 tells us the importance of sparsity on the performance, but in this case Gaussian kernel worked as good as polynomial kernel.

6 Conclusions and Future Work

The development of supervised feature extraction models has been cast in the form of hierarchical latent max-margin projection, which jointly learns the discriminative subspace and classifier. We also extend the linear projection model to a kernel nonlinear version. In kernel methods, the coefficient matrix in kernel expansion is sparse and characterized by a spike-and-slab prior. The experiments on UCI, Caltech101 and radar measured HRRP data sets have shown their effectiveness and efficiency.

The proposed models are flat and do not consider any complicated distribution in the data, such as linear substructure. Thanks to the latent max-margin criterion, a good opportunity has been provided to incorporate Bayesian nonparametrics [2] to address those data with more complicated structures [19]. Moreover, according to the recent max-margin clustering work [8], in the future we are also interested in building a max-margin feature extraction method in an unsupervised way.

Acknowledgments

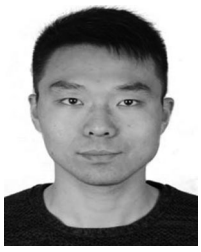
The authors would like to give special thanks to Minjie Xu and Jun Zhu for their valuable discussions and suggestions. This research has been supported by the Program for Young Thousand Talent by Chinese Central government, the National Natural Science Foundation of China (61372132), Program for New Century Excellent Talents in University (NCET-13-0945). Hongwei Liu is the corresponding author.

References

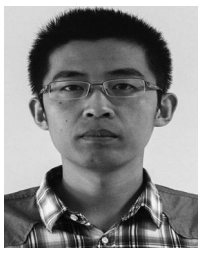
[1] G. Baudat and F. Anouar, "Feature vector selection and projection using kernels," *Neurocomputing*, vol. 55, nos. 1/2, pp. 21–38, 2003.
[2] D. Blei, T. Griffiths, and M. Jordan, "The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies," *J. ACM*, vol. 57, pp. 21–30, 2010.
[3] L. Bo and C. Sminchisescu, "Efficient match kernel between sets of features for visual recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 135–143.
[4] C. J. C. Burges, "Simplified support vector decision rules," in *Proc. Int. Conf. Mach. Learn.*, 1996, pp. 71–77.
[5] C. Carvalho, J. Chang, J. Lucas, J. R. Nevins, Q. Wang, and M. West, "High-dimensional sparse factor modelling: Applications in gene expression genomics," *JASA*, vol. 103, pp. 1438–1456, 2008.
[6] B. Chen, M. Chen, J. Paisley, A. Zaas, C. Woods, G. S Ginsburg, A. Hero, J. Lucas, D. Dunson, and L. Carin, "Bayesian inference of the number of factors in gene-expression analysis: Application to human virus challenge studies," *BMC Bioinformat.*, vol. 11. no. 1, p. 552, 2010.
[7] B. Chen, H. Liu, and Z. Bao, "Optimizing the data-dependent kernel under a unified kernel optimization framework," *Pattern Recognit.*, vol. 41. no. 6, pp. 2107–2119, 2008.

[8] C. Chen, J. Zhu, and X. Zhang, "Robust Bayesian max-margin clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 532–540.
[9] H. Chen, H. Chang, and T. Liu, "Local discriminant embedding and its variants," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 846–853.
[10] M. Chen, D. Carlson, A. Zaas, C. Woods, G. S. Ginsburg, J. Lucas, and L. Carin, "Detection of viruses via statistical gene-expression analysis," *IEEE Trans. Biomed. Eng.*, vol. 58. no. 3, pp. 468–479, Mar. 2011.
[11] M. Chen, J. Silva, J. Paisley, C. Wang, D. Dunson, and L. Carin, "Compressive sensing on manifolds using a nonparametric mixture of factor analyzers: Algorithm and performance bounds," *IEEE Trans. Signal Process.*, vol. 58. no. 12, pp. 6140–6155, Dec. 2010.
[12] L. Clemmensen, T. Hastie, D. Witten, and B. Ersbøll, "Sparse discriminant analysis," *Technometrics*, vol. 53. no. 2, pp. 406–413, 2011.
[13] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *J. Mach. Learn. Res.*, vol. 2, pp. 265–292, 2001.
[14] L. Du, H. Liu, Z. Bao, and M. Xing, "Radar HRRP target recognition based on higher order spectra," *IEEE Trans. Signal Process.*, vol. 53. no. 7, pp. 2359–2368, Jul. 2005.
[15] R. O. Duda, P. E. Hart, and D. H. Stork, *Pattern Classification*, 2nd ed. New York, NY, USA: Wiley Interscience, 2000.
[16] M. Gnen, "Bayesian supervised dimensionality reduction," *IEEE Trans. Cybern.*, vol. 43. no. 6, pp. 2179–2189, Dec. 2013.
[17] A. Gelfand and A. Smith, "Sample based approaches to calculating marginal densities," *J. Amer. Statist. Assoc.*, vol. 85, pp. 398–409, 1990.
[18] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.
[19] L. Hannah, D. Blei, and W. Powell, "Dirichlet process mixtures of generalized linear models," *J. Mach. Learn. Res.*, vol. 12, pp. 1923–1953, 2012.
[20] L. He and L. Carin, "Exploiting structure in wavelet-based Bayesian compressive sensing," *IEEE Trans. Signal Process.*, vol. 57. no. 9, pp. 3488–3497, Sep. 2009.
[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 6, pp. 2278–2324, Nov. 1998.
[22] F. Li, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," in *Proc. Workshop Generative-Model Based Vis.*, 2004, p. 178.
[23] J. Mairal, F. Bach, and J. Ponce, "Task-driven dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34. no. 4, pp. 791–804, Apr. 2012.
[24] J. R. Michael, W. R. Schucany, and R. W. Haas, "Generating random variates using transformations with multiple roots," *Amer. Statist.*, vol. 30. no. 2, pp. 88–90, 1976.
[25] S. Mika, G. Ratsch, J. Weston, B. Schlkopf, and K.-R. Mller, "Fisher discriminant analysis with kernels," in *Proc. Adv. Neural Inf. Process. Syst. IX*, 1999, pp. 41–48.
[26] J. Paisley and L. Carin, "Nonparametric factor analysis with beta process priors," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 777–784.
[27] T. Park and G. Casella, "The Bayesian lasso," *J. Amer. Statist. Assoc.*, vol. 103, pp. 681–686, 2008.
[28] N. G. Polson and S. L. Scott, "Data augmentation for support vector machines," *Bayesian Anal.*, vol. 6. no. 1, pp. 1–24, 2011.
[29] P. Rai and H. Daumé III, "Multi-label prediction via sparse infinite CCA," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1518–1526.
[30] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *J. Mach. Learn. Res.*, vol. 5, pp. 101–141, 2004.
[31] B. Schlkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural Comput.*, vol. 10. no. 5, pp. 1299–1319, 1998.
[32] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[33] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Statist. Soc. Ser. B*, vol. 58, pp. 267–288, 1996.
[34] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.
[35] L. J. P. van der Maaten, E. O. Postma, and H. J. van den Herik, "*Dimensionality reduction: A comparative review*," Tilburg Univ., Tilburg, Netherlands, Tech. Rep. TiCC-TR 2009-005, 2009.
[36] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag, 1995.
[37] M. Wang, F. Sha, and M. Jordan, "Unsupervised kernel dimension reduction," in *Proc. Adv. Neural Inf. Process. Syst. 23*, pp. 2379–2387.
[38] X. Wang and X. Tang, "Dual-space linear discriminant analysis for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2004, pp. II-564–II-569.
[39] M. West, "Bayesian factor regression models in the 'large p, small n' paradigm," in *Bayesian Statist.*, vol. 7, pp. 723–732, 2003.
[40] M. Xu, J. Zhu, and B. Zhang, "Fast max-margin matrix factorization with data augmentation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 978–986.
[41] S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, and S. Lin, "Graph embedding and extension: A general framework for dimensionality reduction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29. no. 1, pp. 40–51, Jan. 2007.
[42] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 1794–1801.
[43] J. Ye and T. Wang, "Regularized discriminant analysis for high-dimensional, low sample size data," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 454–463.
[44] S. Yu, K. Yu, V. Tresp, H. P. Kriegel, and M. Wu, "Supervised probabilistic principal component analysis," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 464–473.
[45] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolution networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 2528–2535.
[46] M. Zhou, H. Chen, J. Paisley, L. Ren, G. Sapiro, and L. Carin, "Non-parametric Bayesian dictionary learning for sparse image representations," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2295–2303.
[47] J. Zhu, N. Chen, H. Perkins, and B. Zhang, "Gibbs max-margin topic models with data augmentation," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1073–1110, 2014.
[48] J. Zhu, N. Chen, and E. P. Xing, "Infinite latent SVM for classification and multi-task learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 1620–1628.
[49] M. Zhu and A. M. Martinez, "Subclass discriminant analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28. no. 8, pp. 1274–1286, Jun. 2006.

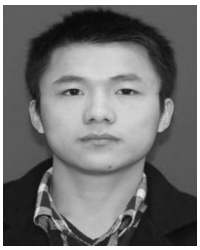
Bo Chen (M'13) received the BS and PhD degrees in electrical engineering from Xidian University, Xian, China, 2003 and 2008, respectively. From 2008 to 2013, he was a research scientist with the ECE Department, Duke University. Since 2013, he was selected to the Young Thousand Talents Program and works as a professor at Xidian University. His research interests include statistical machine learning and radar automatic target recognition. He is a member of the IEEE.



Hao Zhang received the BS degree in electronic engineering from Xidian University in 2012. He is currently working toward the PhD degree at Xidian University. His research interests include statistical machine learning and radar automatic target recognition.



Xuefeng Zhang received the BS degree in electronic engineering from Xidian University in 2010. He is currently working toward the PhD degree at Xidian University. His research interests include statistical machine learning and radar automatic target recognition.



Wei Wen received the BS degree in electronic engineering from Xidian University in 2011. He is currently working toward the PhD degree at Xidian University. His research interests include statistical machine learning and radar automatic target recognition.

Hongwei Liu (M'00) worked at Xidian University. From 2001 to 2002, he was a visiting scholar in the Department of ECE, Duke University. He is currently a professor and the director of the National Laboratory of Radar Signal Processing, Xidian University. His research interests are radar automatic target recognition, radar signal processing, adaptive signal processing, and cognitive radar. He is a member of the IEEE.

Jun Liu received the PhD degree in electronic engineering from Xidian University in 2010. From 2012 to 2014, he was a postdoctral fellow at Stevens Institute of Technology. He joined Xidian University as an associated professor in 2014. His research interests include radar target detection and MIMO radar.