Small-variance Asymptotics for Dirichlet Process Mixtures of SVMs

Yining Wang
The Institute for Theoretical Computer Science
Institute for Interdisciplinary Information Sciences
Tsinghua University, Beijing, China
ynwang.yining@gmail.com

Jun Zhu
Department of Computer Science and Technology
TNList Lab, State Key Lab of Intell. Tech. & Sys.
Tsinghua University, Beijing, China
dcszj@mail.tsinghua.edu.cn

Abstract

Infinite SVM (iSVM) is a Dirichlet process (DP) mixture of large-margin classifiers. Though flexible in learning nonlinear classifiers and discovering latent clustering structures, iSVM has a difficult inference task and existing methods could hinder its applicability to large-scale problems. This paper presents a small-variance asymptotic analysis to derive a simple and efficient algorithm, which monotonically optimizes a *max-margin DP-means* (M^2DPM) problem, an extension of DP-means for both predictive learning and descriptive clustering. Our analysis is built on Gibbs infinite SVMs, an alternative DP mixture of large-margin machines, which admits a partially collapsed Gibbs sampler without truncation by exploring data augmentation techniques. Experimental results show that M^2DPM runs much faster than similar algorithms without sacrificing prediction accuracies.

Introduction

Clustering is a fundamental task in descriptive unsupervised learning with many popular methods such as K-means and various Bayesian models. It also plays an important role in predictive supervised learning for discovering subgroup structures and improving time efficiency. For example, when learning SVM classifiers, it could be computationally expensive to directly solve a large optimization problem on all training data. To improve efficiency and/or disclose descriptive structures, practitioners have used clustering methods to partition the data into subgroups and learn a simple classifier within each cluster (Fu, Robles-Kelly, and Zhou 2010).

Recent work on DP mixtures of generalized linear models (Shahbaba and Neal 2009; Hannah, Blei, and Powell 2011) provides flexible solutions to jointly learn classifiers and perform clustering; meanwhile these methods automatically resolve the unknown number of clusters, thereby bypassing the model selection problem of K-means and parametric mixture models. Along this line, infinite SVM (iSVM) (Zhu, Chen, and Xing 2011), a DP mixture of large-margin machines, provides an alternative approach that enjoys the advantages of Bayesian nonparametrics to resolve the number of components as well as the discriminative property of large-margin machines. However, these nonparametric methods normally have difficult inference problems, for which both variational and Monte Carlo methods could be too expensive to be applied to large-scale applications.

Small-variance asymptotics (SVA) offers useful techniques to setup conceptual links between probabilistic and non-probabilistic models and derive new algorithms that can be simple and scalable. For instance, connections between probabilistic PCA (pPCA) and standard PCA can be made by letting the covariance of the likelihood in pPCA approach zero (Tipping and Bishop 1999); and similarly the K-means algorithm can be obtained from the EM algorithm for Gaussian mixtures when the covariances of Gaussian components go to zero. Recent progress has been made on deriving new computational methods. For example, DP-means is a deterministic extension to K-means by applying SVA analysis to the Gibbs sampling algorithm of DP mixtures (Kulis and Jordan 2012); and the work (Broderick, Kulis, and Jordan 2013) provides a generic SVA analysis on MAP estimates and presents an example of latent feature learning. However, SVA analysis has been exclusively performed in unsupervised learning, while little work has been done for supervised learning, which imposes new challenges and requires more careful analysis as shown later.

This paper presents an SVA analysis to DP mixtures of SVMs and derives simple and scalable algorithms that perform descriptive clustering and predictive classifier learning jointly. Technically, our analysis is built on Gibbs infinite SVMs (Gibbs-iSVM) which learns component-wise classifiers by exploring the ideas of Gibbs classifiers, a powerful paradigm in learning theory (Germain et al. 2009). For Gibbs-iSVM with the generic exponential family likelihood, we present a partially collapsed Gibbs sampler by exploring data augmentation (Tanner and Wong 1987; Polson and Scott 2011; Zhu et al. 2014) techniques. Compared to the variational algorithm of iSVM, the Gibbs sampler does not require truncated mean-field assumptions, thus converging to the true posterior. However, some expectations in the Gibbs sampler are hard to compute in closed forms. To improve efficiency, we perform SVA analysis to the Gibbs sampler and derive a simple deterministic algorithm by scaling both the covariances of the likelihood and the regularization parameters appropriately, where scaling the regularization parameters is a key new feature for extend-

ing SVA analysis to supervised models. In addition, by expressing the Chinese Restaurant Process (CRP, a marginalized version of DP) prior as exchangeable partition probability functions (EPPF) (Blackwell and MacQueen 1973; Pitman 1995; Aldous 1985) and adopting the same set of scalings to the posterior directly, we obtain a *max-margin DP-means* (M^2DPM) optimization problem, an extension to the descriptive DP-means for both predictive learning and descriptive clustering. We prove that the deterministic algorithm monotonically optimizes the M^2DPM problem for local optimums. Experimental results on both synthetic and real datasets demonstrate the efficiency and effectiveness of the M^2DPM algorithm compared to other competitors.

Preliminaries

Exponential family distributions

An exponential family distribution can be characterized as follows (Barndorff-Nielsen 1978):

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \exp(\langle \boldsymbol{x}, \boldsymbol{\theta} \rangle - \psi(\boldsymbol{\theta}) - h(\boldsymbol{x})), \quad (1)$$

where $\boldsymbol{\theta}$ is the natural parameter and $\psi(\boldsymbol{\theta})$ is the log-partition function. The mean and covariance of an exponential family are given by $\nabla\psi(\boldsymbol{\theta})$ and $\nabla^2\psi(\boldsymbol{\theta})$ respectively. In the Bayesian setting, a convenient prior would be the conjugate prior (Agarwal and Daume 2010)

$$p(\boldsymbol{\theta}|\boldsymbol{\tau}, \kappa) = \exp(\langle \boldsymbol{\theta}, \boldsymbol{\tau} \rangle - \kappa\psi(\boldsymbol{\theta}) - m(\boldsymbol{\tau}, \kappa)), \quad (2)$$

where $\boldsymbol{\tau}$ and κ are hyper-parameters.

Given a convex set $S \subseteq \mathbb{R}^d$ and a differentiable, strictly convex function $\varphi : S \to \mathbb{R}$, Bregman divergence (Bregman 1967) $D_\varphi(\cdot, \cdot)$ is defined over pairs of points $\boldsymbol{x}, \boldsymbol{y} \in S$ as $D_\varphi(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x}) - \varphi(\boldsymbol{y}) - \langle \boldsymbol{x} - \boldsymbol{y}, \nabla\varphi(\boldsymbol{y}) \rangle$. In (Banerjee et al. 2005), a bijection between exponential family distributions and Bregman divergence was established. Specifically, we can use the mean parameter of an exponential family distribution to equivalently characterize the distribution and conjugate prior as:

$$p(\boldsymbol{x}|\boldsymbol{\mu}) = \exp(-D_\varphi(\boldsymbol{x}, \boldsymbol{\mu}))f_\varphi(\boldsymbol{x}), \quad (3)$$

$$p(\boldsymbol{\mu}|\boldsymbol{\tau}, \kappa) = \exp\left(-\kappa D_\varphi\left(\frac{\boldsymbol{\tau}}{\kappa}, \boldsymbol{\mu}\right)\right) g_\varphi(\boldsymbol{\tau}, \kappa), \quad (4)$$

where $\varphi(\cdot)$ is the Legendre-conjugate function of the log-partition function $\psi(\cdot)$ and $f_\varphi(\boldsymbol{x}) = \exp(\varphi(\boldsymbol{x}) - h(\boldsymbol{x}))$. These representations can greatly simplify our small-variance analysis (Jiang, Kulis, and Jordan 2012).

Infinite SVMs: a DP mixture of SVMs

A DP mixture (DPM) is a nonparametric Bayesian mixture model (Hjort et al. 2010), where the number of cluster components is unbounded. Given an instance $\boldsymbol{x}_i \in \mathbb{R}^d$ and its component assignment $z_i \in \mathbb{N}$, we consider the general exponential family likelihood of the data instance:

$$p(\boldsymbol{x}_i|z_i, \boldsymbol{\mu}) = \exp(-D_\varphi(\boldsymbol{x}_i, \boldsymbol{\mu}_{z_i}))f_\varphi(\boldsymbol{x}_i), \quad (5)$$

where $\boldsymbol{\mu}_k$ is the mean parameter of component k. For priors, the cluster assignments $\boldsymbol{z}_{1:n}$ follow a CRP with the concentration parameter α, and the parameter of each component $\boldsymbol{\mu}_k$ follows a conjugate prior as in (4) with the hyper-parameters $(\boldsymbol{\tau}, \kappa)$.

Given a set of data $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n$, we can apply Bayes' rule to obtain the posterior distribution $p(\boldsymbol{z}, \boldsymbol{\mu}|\boldsymbol{X})$, which is equivalent to the solution of the convex problem:

$$\min_{q(\boldsymbol{z},\boldsymbol{\mu})\in\mathcal{P}} \mathrm{KL}(q(\boldsymbol{z}, \boldsymbol{\mu})\|p_0(\boldsymbol{z}, \boldsymbol{\mu})) - \mathbb{E}_q[\log p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\mu})], \quad (6)$$

where p_0 is the prior; $p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\mu}) = \prod_{i=1}^n p(\boldsymbol{x}_i|z_i, \boldsymbol{\mu})$ is the likelihood; and $\mathcal{P}$ is a space of normalized distrubtions.

Infinite SVMs (iSVM) extends the unsupervised DPM for both predictive supervised learning and descriptive clustering. Let's consider binary classification, where each instance is a pair of input features $\boldsymbol{x}_i$ and class label $y_i \in \{-1, +1\}$. In iSVM, each cluster k is associated with a mean parameter $\boldsymbol{\mu}_k$ to characterize the likelihood of $\boldsymbol{x}$ and a classifier $\boldsymbol{\eta}_k$ to predict y. For the common linear classifiers, the discriminant function is

$$f(\boldsymbol{x}_i; \boldsymbol{\eta}, z_i) = \boldsymbol{\eta}_{z_i}^\top \boldsymbol{x}_i = \sum_{k=1}^\infty \delta_{z_i,k} \boldsymbol{\eta}_k^\top \boldsymbol{x}_i. \quad (7)$$

To resolve the uncertainty of $\Theta := \{\boldsymbol{\mu}, \boldsymbol{\eta}, \boldsymbol{z}\}$, characterized by a distribution $q(\Theta)$, iSVMs defines the averaging (or expected) discriminant function $f(\boldsymbol{x}_i; q) = \mathbb{E}_q[f(\boldsymbol{x}_i; \boldsymbol{\eta}, z_i)]$, and makes predictions using the rule $\hat{y} = \mathrm{sign} f(\boldsymbol{x}_i; q)$. For this averaging classifier, we can measure its performance by using the hinge loss, $\mathcal{R}(q, \boldsymbol{X}) = \sum_{i=1}^n (l - y_i f(\boldsymbol{x}_i; q))_+$, where $(x)_+ = \max(0, x)$ and $l \geq 1$ is the cost of a wrong prediction. The hinge loss is an upper bound of training error. Then, iSVM solves the regularized Bayesian inference (RegBayes) (Zhu, Chen, and Xing 2014) problem

$$\min_{q(\Theta)\in\mathcal{P}} \mathcal{L}(q(\Theta)) + 2c \cdot \mathcal{R}(q(\Theta), \boldsymbol{X}), \quad (8)$$

where $\mathcal{L}(q(\Theta)) = \mathrm{KL}(q\|p_0) - \mathbb{E}_q[\log p(\boldsymbol{X}|\boldsymbol{z}, \boldsymbol{\eta})]$ is the objective of Bayesian inference for the DPM, and c is a positive regularization parameter balancing the two terms. For $\boldsymbol{\eta}$, a common Gaussian prior is $\boldsymbol{\eta}_k \sim \mathcal{N}(\boldsymbol{\eta}_k|\boldsymbol{0}, \nu^2\boldsymbol{I}_d)$.

Variational methods have been developed for approximate inference in iSVM, with a truncated mean-field assumption for tractability (Zhu, Chen, and Xing 2011). Generally, it is hard to develop Monte Carlo methods for such Bayesian max-margin models, which may involve solving a dual problem still with mean-field assumptions (Jiang et al. 2012).

Gibbs Infinite SVMs

We present Gibbs iSVMs with the generic exponential family likelihood and develop a Gibbs sampler that grounds our small-variance asymptotic (SVA) analysis. Gibbs iSVMs is formulated as a DP mixture of Gibbs classifiers (Germain et al. 2009; Zhu et al. 2014; Zhang, Zhu, and Zhang 2014).

Learning with an expected hinge loss

For our DP mixtures, a Gibbs classifier randomly draws a sample $(\boldsymbol{\eta}, \boldsymbol{z})$ from the target posterior $q(\Theta)$, and makes predictions using the linear discriminant function (7) via the rule $\hat{y}_i = \mathrm{sign} f(\boldsymbol{x}_i; \boldsymbol{\eta}, z_i)$. We measure the performance of this latent classifier using the hinge loss as a surrogate to the training error, $\mathcal{R}'(\boldsymbol{\eta}, \boldsymbol{z}, \boldsymbol{X}) = \sum_{i=1}^n (l - y_i \boldsymbol{\eta}_{z_i}^\top \boldsymbol{x}_i)_+$. To resolve the uncertainty, we take the expectation and define the expected hinge loss

$$\mathcal{R}'(q(\Theta), \boldsymbol{X}) = \mathbb{E}_q[\mathcal{R}'(\boldsymbol{\eta}, \boldsymbol{z}, \boldsymbol{X})]. \quad (9)$$

Then, Gibbs iSVMs solves the new RegBayes problem:

$$\min_{q(\Theta)\in\mathcal{P}} \quad \mathcal{L}(q(\Theta)) + 2c\cdot\mathcal{R}'(q(\Theta),\boldsymbol{X}). \tag{10}$$

Note: by using Jensen's inequality, we can show the relationship between the two hinge losses as: $\mathcal{R}'(q,\boldsymbol{X}) \geq \mathcal{R}(q,\boldsymbol{X})$.

Representation with data augmentation

One nice property of Gibbs iSVMs is that we can develop a Gibbs sampler without making truncated mean-field assumptions. This sampler will lead to a simple deterministic algorithm via our SVA analysis, as shown soon later. Specifically, we can express the solution to (10) as

$$q(\Theta) = \frac{p_0(\Theta)\prod_{i=1}^n p(\boldsymbol{x}_i|z_i,\boldsymbol{\mu})\phi(y_i|z_i,\boldsymbol{\eta})}{Z(\boldsymbol{D})}, \tag{11}$$

where $\phi(y_i|z_i,\boldsymbol{\eta}) = \exp(-2c(\zeta_i^{z_i})_+)$ is an unnormalized likelihood corresponding to the hinge loss, $\zeta_i^k = l - y_i\boldsymbol{\eta}_k^\top\boldsymbol{x}_i$ is the margin achieved by applying classifier $\boldsymbol{\eta}_k$ on data $\boldsymbol{x}_i$ and $Z(\boldsymbol{D})$ is the normalization factor. As in (Polson and Scott 2011), we can show that

$$\phi(y_i|z_i,\boldsymbol{\eta}) = \int_0^\infty \frac{1}{\sqrt{2\pi\omega_i}}\exp\left(-\frac{(\omega_i + c\zeta_i^{z_i})^2}{2\omega_i}\right)\mathrm{d}\omega_i, \tag{12}$$

where $\omega_i > 0$ is an augmented variable. Then, $q(\Theta)$ can be written as a marginal of the complete distribution:

$$q(\Theta,\boldsymbol{\omega}) = \frac{p_0(\Theta)\prod_{i=1}^n p(\boldsymbol{x}_i|z_i,\boldsymbol{\mu})\phi(y_i,\omega_i|z_i,\boldsymbol{\eta})}{Z(\boldsymbol{D})}, \tag{13}$$

where $\phi(y_i,\omega_i|z_i,\boldsymbol{\eta}) = \frac{1}{\sqrt{2\pi\omega_i}}\exp(-\frac{(\omega_i+c\zeta_i^{z_i})^2}{2\omega_i})$ is the augmented (unnormalized) likelihood for the hinge loss.

Partially collapsed Gibbs sampling

With data augmentation, we develop a Gibbs sampler that iteratively samples $\boldsymbol{z}$, $\boldsymbol{\mu}$, $\boldsymbol{\eta}$ and $\boldsymbol{\omega}$ from the posterior (13); thus the samples of Gibbs iSVM by dropping the variables $\boldsymbol{\omega}$. To improve the convergence rate, we also adopt a partially collapsed approach (van Dyk and Park 2008) when sampling $\boldsymbol{z}$ by integrating out $\boldsymbol{\omega}$. The Gibbs sampler iteratively samples from the following conditional distributions:

For $\boldsymbol{\mu}_k$: the conditional distribution is $q(\boldsymbol{\mu}_k|\boldsymbol{z}) \propto p(\boldsymbol{\mu}_k|\boldsymbol{\tau},\kappa)\prod_{i\in\mathcal{N}_k} p(\boldsymbol{x}_i|\boldsymbol{\mu}_k)$, where $\mathcal{N}_k := \{i : z_i = k\}$ is the set of instances assigned to cluster k. With the conjugate prior, $q(\boldsymbol{\mu}_k|\boldsymbol{z})$ is still an exponential family distribution:

$$q(\boldsymbol{\mu}_k|\boldsymbol{z}) \propto \exp\left(-\kappa' D_\varphi\left(\frac{\boldsymbol{\tau}'}{\kappa'},\boldsymbol{\mu}_k\right)\right), \tag{14}$$

where $\boldsymbol{\tau}' = \boldsymbol{\tau} + \sum_{i\in\mathcal{N}_k}\boldsymbol{x}_i$, $\kappa' = \kappa + n_k$ and $n_k = |\mathcal{N}_k|$ is the number of data samples in cluster k.

For z_i: let $n_{-i,k}$ be the number of instances other than $\boldsymbol{x}_i$ that belong to cluster k. Then, for cluster k with $n_{-i,k} > 0$, the conditional probability of getting $z_i = k$ is

$$\begin{aligned} q(z_i = k|\boldsymbol{z}_{-i},\boldsymbol{\eta}) &\propto p_0(z_i = k|\boldsymbol{z}_{-i})p(\boldsymbol{x}_i|\boldsymbol{\mu}_k)\phi(y_i|z_i = k) \\ &\propto n_{-i,k}p(\boldsymbol{x}_i|\boldsymbol{\mu}_k)\phi(y_i|z_i = k,\boldsymbol{\eta}). \end{aligned} \tag{15}$$

The probability of assigning $\boldsymbol{x}_i$ to a new cluster is

$$q(z_i = z_{\text{new}}|\alpha) \propto \alpha p(\boldsymbol{x}_i)\int\phi(y_i|\boldsymbol{\eta})p_0(\boldsymbol{\eta})\mathrm{d}\boldsymbol{\eta}, \tag{16}$$

where $p(\boldsymbol{x}_i) = \int p(\boldsymbol{x}_i|\boldsymbol{\mu})p_0(\boldsymbol{\mu})\mathrm{d}\boldsymbol{\mu}$ and $\phi(y_i|\boldsymbol{\eta}) = \exp(-2c(l - y_i\boldsymbol{\eta}^\top\boldsymbol{x}_i)_+)$. Though $p(\boldsymbol{x}_i)$ has closed forms because of conjugate priors, the integral $\int\phi(y_i|\boldsymbol{\eta})p_0(\boldsymbol{\eta})\mathrm{d}\boldsymbol{\eta}$ does not have a simple closed form due to the hinge loss term $\phi(y_i|\boldsymbol{\eta})$ and approximation is needed in practice.

For ω_i: when the margin $\zeta_i^{z_i}$ is fixed, the conditional distribution over the augmented variable ω_i is a generalized inverse Gaussian distribution (Devroye 1986). Consequently, ω_i^{-1} follows an inverse Gaussian distribution

$$q(\omega_i^{-1}|\boldsymbol{z},\boldsymbol{\eta}) = \mathcal{IG}\Big(\omega_i^{-1};\frac{1}{c|\zeta_i^{z_i}|},1\Big), \tag{17}$$

where $\mathcal{IG}(x;a,b) = \sqrt{\frac{b}{2\pi x^3}}\exp(-\frac{b(x-a)^2}{2a^2x})$ for $a > 0$ and $b > 0$. We can sample ω_i^{-1} from an inverse Gaussian distribution in $O(1)$ time (Michael, Schucany, and Haas 1976).

For $\boldsymbol{\eta}_k$: the conditional distribution of $\boldsymbol{\eta}_k$ is

$$q(\boldsymbol{\eta}_k|\boldsymbol{z},\boldsymbol{\omega},\boldsymbol{\mu}) \propto \exp\Big(-\frac{\|\boldsymbol{\eta}_k\|^2}{2\nu^2} - \sum_{i\in\mathcal{N}_k}\frac{(\omega_i + c\zeta_i^k)^2}{2\omega_i}\Big), \tag{18}$$

a Gaussian distribution with covariance and mean: $\boldsymbol{\Lambda}_k = (\frac{1}{\nu^2}I + c^2\sum_{i\in\mathcal{N}_k}\frac{\boldsymbol{x}_i\boldsymbol{x}_i^\top}{\omega_i})^{-1}$; $\boldsymbol{\lambda}_k = \boldsymbol{\Lambda}_k(c\sum_{i\in\mathcal{N}_k} y_i\frac{\omega_i+cl}{\omega_i}\boldsymbol{x}_i)$.

Small-Variance Asymptotics

In the above sampler, the integral in Eq. (16) is normally hard to compute. Though we can approximate it, e.g., using importance sampling, it could be inaccurate and may significantly slow down the algorithm. Also, in practice, deterministic algorithms like K-means are sometimes preferred because they are fast and easy to implement. This motivates us to perform SVA analysis to derive new algorithms.

Asymptotic behavior of the Gibbs sampler

We first analyze the asymptotic behavior of the Gibbs sampler under the small-variance setting. Our SVA analysis will use one fact that under the scaling of exponential family parameters (i.e. $\tilde{\boldsymbol{\theta}} = \beta\boldsymbol{\theta}$), the mean remains the same while the variance is scaled down. This fact is formulated in Lemma 1, following (Jiang, Kulis, and Jordan 2012).

Lemma 1. *Denote $\boldsymbol{\mu}(\boldsymbol{\theta})$ as the mean and cov$(\boldsymbol{\theta})$ as the covariance of $p(\boldsymbol{x}|\boldsymbol{\theta})$ with log-partition $\psi(\boldsymbol{\theta})$. For the scaled exponential family with $\tilde{\boldsymbol{\theta}} = \beta\boldsymbol{\theta}$ and $\tilde{\psi}(\tilde{\boldsymbol{\theta}}) = \beta\psi(\tilde{\boldsymbol{\theta}}/\beta)$, its mean and covariance are $\tilde{\boldsymbol{\mu}}(\tilde{\boldsymbol{\theta}}) = \boldsymbol{\mu}(\boldsymbol{\theta})$ and $\tilde{cov}(\tilde{\boldsymbol{\theta}}) =$ cov$(\boldsymbol{\theta})/\beta$, respectively.*

Therefore, with the scaling $\tilde{\boldsymbol{\theta}} = \beta\boldsymbol{\theta}$, Eq. (1) and (2) become Eq. (19) and (20) respectively, where $\tilde{\varphi} := \beta\varphi$.

$$p(\boldsymbol{x}|\tilde{\boldsymbol{\theta}}(\boldsymbol{\mu})) = \exp(-\beta D_\varphi(\boldsymbol{x},\boldsymbol{\mu}))f_{\tilde{\varphi}}(\boldsymbol{x}), \tag{19}$$

$$p(\tilde{\boldsymbol{\theta}}(\boldsymbol{\mu})|\boldsymbol{\tau},\kappa,\beta) = \exp\Big(-\kappa D_\varphi\Big(\frac{\boldsymbol{\tau}}{\kappa},\boldsymbol{\mu}\Big)\Big)g_{\tilde{\varphi}}\Big(\frac{\boldsymbol{\tau}}{\beta},\frac{\kappa}{\beta}\Big), \tag{20}$$

Below, we present the detailed SVA analysis of each step of the Gibbs sampler, and provide insights of our new algorithm. Our analysis will focus on the cluster assignments $\boldsymbol{z}$, classifiers $\boldsymbol{\eta}$ and augmented variables $\boldsymbol{\omega}$. Compared to previous work on SVA, both $\boldsymbol{\eta}$ and $\boldsymbol{\omega}$ are unique to our methods; and the analysis of $\boldsymbol{z}$ also needs new techniques, as we shall see.

For $\boldsymbol{\mu}_k$: applying the scaling $\tilde{\boldsymbol{\theta}}_k = \beta\boldsymbol{\theta}_k$, and the fact in Eq. (19), the conditional in Eq. (14) now becomes:

$$q(\boldsymbol{\mu}_k|\boldsymbol{z},\beta) \propto \exp\left(-\kappa'' D_\varphi\left(\frac{\boldsymbol{\tau}''}{\kappa''},\boldsymbol{\mu}_k\right)\right), \quad (21)$$

where $\kappa'' = \kappa + \beta n_k$ and $\boldsymbol{\tau}'' = \boldsymbol{\tau} + \beta\sum_{i\in\mathcal{N}_k}\boldsymbol{x}_i$. Similar as in (Jiang, Kulis, and Jordan 2012) we can show that as $\beta \to \infty$ the conditional distribution of $\boldsymbol{\mu}_k$ will be concentrated on the empirical mean. See Appendix A for details.

For z_i: since we are performing both classification and clustering, we need both the likelihood (e.g. $D_\varphi(\boldsymbol{x}_i,\boldsymbol{\mu}_k)$) and hinge loss (e.g. $\phi(y_i|z_i,\boldsymbol{\eta})$) to play a role in determining z_i. A straightforward application of existing SVA techniques will drop the hinge loss. To avoid such artifacts, we develop a new analysis by scaling the regularization parameter c. Specifically, let $\tilde{c} = \beta' c$ and $\phi(y_i|z_i = k,\boldsymbol{\eta},\beta') = \exp(-2\tilde{c}(\zeta_i^k)_+)$. We also set a connection between the two scaling constants $\boldsymbol{\beta} := (\beta,\beta')$ as $\beta = s\beta'$, where s is a constant. Then, by Eq. (19), the conditional in Eq. (15) becomes

$$q(z_i = k|\boldsymbol{z}_{-i},\boldsymbol{\mu},\boldsymbol{\eta},\boldsymbol{\beta}) = \frac{n_{-i,k}}{Z}\cdot f_{\tilde{\varphi}}(\boldsymbol{x}_i) \cdot \exp\left(-\beta'\left(s\cdot D_\varphi(\boldsymbol{x}_i,\boldsymbol{\mu}_k) + 2c(\zeta_i^k)_+\right)\right). \quad (22)$$

When generating new clusters, we need to retain the regularization term (i.e. prior) of $\boldsymbol{\eta}$. Therefore, the variance of the prior distribution, ν^2, should be properly scaled. To this end, we use the scaling $\tilde{\nu}^2 = \nu^2/\beta'$. Then, the conditional probability in Eq. (16) can be expressed as

$$q(z_i = z_{\text{new}}|\alpha,\boldsymbol{\beta}) = \frac{\alpha}{Z}\cdot I_1\cdot I_2, \quad (23)$$

where the two integrals are $I_1 := \int p(\boldsymbol{x}_i|\tilde{\boldsymbol{\theta}})p_0(\tilde{\boldsymbol{\theta}})\mathrm{d}\tilde{\boldsymbol{\theta}}$ and $I_2 := \int \phi(y_i|z_i,\boldsymbol{\eta},\beta')p_0(\boldsymbol{\eta}|\tilde{\nu}^2)\mathrm{d}\boldsymbol{\eta}$. Though the integrals are hard to compute, we can apply the Laplace's method to derive meaningful SVA results, as detailed below.

For integral I_1, using Eqs. (19,20), we can expand it as

$$I_1 = f_{\tilde{\varphi}}(\boldsymbol{x}_i)g_{\tilde{\varphi}}\left(\frac{\boldsymbol{\tau}}{\beta},\frac{\kappa}{\beta}\right)A_{(\tilde{\varphi},\boldsymbol{\tau},\kappa,\beta)}(\boldsymbol{x}_i)\cdot\beta^d J, \quad (24)$$

where $A_{(\tilde{\varphi},\boldsymbol{\tau},\kappa,\beta)}(\boldsymbol{x}_i) := \exp(-(\beta\varphi(\boldsymbol{x}_i) + \kappa\varphi(\frac{\boldsymbol{\tau}}{\beta}) - (\beta+\kappa)\varphi(\frac{\beta\boldsymbol{x}_i+\boldsymbol{\tau}}{\beta+\kappa})))$ arises when combining the exponential family distribution with its conjugate prior; and $J := \int \exp\left(-(\beta+\kappa)D_\varphi\left(\frac{\beta\boldsymbol{x}_i+\boldsymbol{\tau}}{\beta+\kappa},\boldsymbol{\mu}(\boldsymbol{\theta})\right)\right)\mathrm{d}\boldsymbol{\theta}$ is another integral, which can be further expanded via the Laplace's method. Let $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}(\hat{\boldsymbol{\mu}})$ be a local minimum of $D(\boldsymbol{\theta}) = D_\varphi(\frac{\beta\boldsymbol{x}_i+\boldsymbol{\tau}}{\beta+\kappa},\boldsymbol{\mu}(\boldsymbol{\theta}))$. We then have

$$J = \frac{\exp\big(-(\beta+\kappa)D(\hat{\boldsymbol{\theta}})\big)}{(2\pi/(\beta+\kappa))^{-d/2}}\left(\left|\frac{\partial^2 D(\hat{\boldsymbol{\theta}})}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}^\top}\right|^{-1/2} + o(1)\right). \quad (25)$$

As $\lim_{\beta\to\infty}\frac{\beta\boldsymbol{x}_i+\boldsymbol{\tau}}{\beta+\kappa} = \boldsymbol{x}_i$, we have $D(\hat{\boldsymbol{\theta}}) = D(\boldsymbol{\theta}(\boldsymbol{x}_i)) = 0$. Thus, $\lim_{\beta\to\infty} J = (2\pi/(\beta+\kappa))^{d/2}\cdot\mathrm{cov}(\boldsymbol{x}_i)^{-1/2}$. Note also that $\lim_{\beta\to\infty}A_{(\tilde{\varphi},\boldsymbol{\tau},\kappa,\beta)}(\boldsymbol{x}_i) = \exp(-\kappa(\varphi(\boldsymbol{\tau}/\kappa) - \varphi(\boldsymbol{x}_i)))$ (Jiang, Kulis, and Jordan 2012). Since $\mathrm{cov}(\boldsymbol{x}_i)$ and $A_{(\tilde{\varphi},\boldsymbol{\tau},\kappa,\beta)}(\boldsymbol{x}_i)$ do not scale with β, we have

$$I_1 = f_{\tilde{\varphi}}(\boldsymbol{x}_i)g_{\tilde{\varphi}}\left(\frac{\boldsymbol{\tau}}{\beta},\frac{\kappa}{\beta}\right)\cdot(2\pi\beta)^{\frac{d}{2}}\cdot C_1(\boldsymbol{x}_i), \quad (26)$$

as $\beta\to\infty$, where $C_1(\cdot)$ is a function independent of β.

The integral I_2 can be analyzed similarly via the Laplace's method. Let $L(\boldsymbol{\eta}) = 2c(l - y_i\boldsymbol{\eta}^\top\boldsymbol{x}_i)_+ + \frac{\|\boldsymbol{\eta}\|^2}{2\nu^2}$ be the hinge loss and regularization term induced by the classifier $\boldsymbol{\eta}$. Let $\boldsymbol{\eta}^*$ be the classifier that minimizes $L(\boldsymbol{\eta})$. Then we have $I_2 = (2\pi\nu^2/\beta')^{-d/2}\int\exp(-\beta' L(\boldsymbol{\eta}))\mathrm{d}\boldsymbol{\eta}$ expanded as

$$I_2 = \frac{\exp(-\beta' L(\boldsymbol{\eta}^*))}{(2\pi\nu^2/\beta')^{\frac{d}{2}}}\left(\frac{2\pi}{\beta'}\right)^{\frac{d}{2}}\left(\left|\frac{\partial^2 L(\boldsymbol{\eta}^*)}{\partial\boldsymbol{\eta}\partial\boldsymbol{\eta}^\top}\right|^{-\frac{1}{2}} + o(1)\right). \quad (27)$$

Since the terms $2\pi/\beta'$ cancel out in Eq. (27), ν^2 and $\mathrm{cov}(\boldsymbol{\eta}^*)$ do not scale with β', the integral I_2 can be written as

$$I_2 = \exp(-\beta' L(\boldsymbol{\eta}^*))\cdot C_2(\boldsymbol{x}_i), \quad (28)$$

as $\beta'\to\infty$, where $C_2(\cdot)$ is a function independent of β'.

Substituting Eqs. (26,28) into Eq. (23), we obtain

$$q(z_i = z_{\text{new}}|\alpha,\boldsymbol{\beta}) = \frac{\alpha}{Z}\cdot f_{\tilde{\varphi}}(\boldsymbol{x}_i)g_{\tilde{\varphi}}\left(\frac{\boldsymbol{\tau}}{\beta},\frac{\kappa}{\beta}\right)\cdot(2\pi\beta)^{\frac{d}{2}} \cdot\exp(-\beta' L(\boldsymbol{\eta}^*))\cdot C_1(\boldsymbol{x}_i)C_2(\boldsymbol{x}_i). \quad (29)$$

Further, we scale the concentration parameter α with $\boldsymbol{\beta}$, but independent of the data $\boldsymbol{x}_i$, aa in Eq. (30)

$$\alpha = \left(g_{\tilde{\varphi}}\left(\frac{\boldsymbol{\tau}}{\beta},\frac{\kappa}{\beta}\right)\cdot(2\pi\beta)^{\frac{d}{2}}\right)^{-1}\cdot\exp(-\beta'\lambda), \quad (30)$$

for some parameter λ and cancel out the $f_{\tilde{\varphi}}(\boldsymbol{x}_i)$ term in both Eq. (22) and (29). Clearly, as $\beta\to\infty$, the exponential term dominates the other terms in both probabilities. In other words, define "cost functions" $Q_i(k)$ as

$$Q_i(k) = s\cdot D_\varphi(\boldsymbol{x}_i,\boldsymbol{\mu}_k) + 2c(l - y_i\boldsymbol{\eta}_k^\top\boldsymbol{x}_i)_+ \quad (31)$$

for existing components and

$$Q_i(z_{\text{new}}) = \lambda + 2c(l - y_i\boldsymbol{\eta}^{*\top}\boldsymbol{x}_i)_+ + \|\boldsymbol{\eta}^*\|^2/(2\nu^2) \quad (32)$$

for new components. The conditional distribution of z_i is then concentrated on the component with the smallest cost $Q_i(\cdot)$ when $\beta\to\infty$.

For ω_i and $\boldsymbol{\eta}_k$: under the same scalings (i.e., $\tilde{c} = \beta' c$ and $\tilde{\nu}^2 = \nu^2/\beta'$), Eq. (17) becomes $q(\tilde{\omega}_i^{-1}|\boldsymbol{z},\boldsymbol{\eta},) = \mathcal{IG}\big(\tilde{\omega}_i^{-1};\frac{1}{\tilde{c}|\zeta_i^{z_i}|},1\big)$. Since the variance of $\mathcal{IG}(x;a,b)$ is $\frac{a^3}{b}$ and as $\beta'\to\infty$ we have $a\to 0$, the distribution of $\tilde{\omega}_i^{-1}$ will concentrate on its mean $\frac{1}{\tilde{c}|\zeta_i^{z_i}|}$. Thus, $\tilde{\omega}_i$ will concentrate on $\beta'\cdot c|\zeta_i^{z_i}|$. Finally, replacing c,ν^2 and ω_i with $\tilde{c},\tilde{\nu}^2$ and $\tilde{\omega}_i$ in Eq. (18), we have $\tilde{\boldsymbol{\lambda}}_k = \boldsymbol{\lambda}_k$ and $\tilde{\boldsymbol{\Lambda}}_k = \boldsymbol{\Lambda}_k/\beta'$. Thus, the conditional distribution of $\boldsymbol{\eta}_k$ will also concentrate on its posterior mean $\boldsymbol{\lambda}_k$.

In summary, the small-variance analysis results in an iterative algorithm, as outlined in Alg. 1. The algorithm is easier to implement compared to the Gibbs sampler and also runs faster because it avoids sampling from complex distributions. Note that $\boldsymbol{\eta}^*$, the solution to the "one-point SVM" that minimizes $L(\boldsymbol{\eta}) = 2c(l - y_i\boldsymbol{\eta}^\top\boldsymbol{x}_i)_+ + \frac{\|\boldsymbol{\eta}\|^2}{2\nu^2}$, has the closed-form solution (Crammer et al. 2006): $\boldsymbol{\eta}^* = \min(2c\nu^2, 1/\|\boldsymbol{x}_i\|^2)\cdot y_i\boldsymbol{x}_i$. Thus, we can compute $\boldsymbol{\eta}^*$ for each instance $\boldsymbol{x}_i$ in a constant time. Finally, the above analysis can be generalized to multi-class classification. See Appendix C for details.

Algorithm 1 The M^{2}DPM algorithm

Initialize: $\boldsymbol{z}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\eta}^{(0)}, \boldsymbol{\omega}^{(0)}$; $g \leftarrow 0$.
repeat
  For each instance i: $z_i^{(g+1)} \leftarrow \text{argmin}_k Q_i(k)$.
  For each cluster k: $\boldsymbol{\mu}_k^{(g+1)} \leftarrow (\sum_i \delta_{z_i,k}\boldsymbol{x}_i)/(\sum_i \delta_{z_i,k})$.
  For each instance i: $\omega_i^{(g+1)} \leftarrow c \cdot |\zeta_i^{z_i}|$.
  For each cluster k: $\boldsymbol{\eta}_k^{(g+1)} \leftarrow \boldsymbol{\lambda}_k$.
  Update: $g \leftarrow g + 1$.
until converge

Asymptotics of the posterior distribution

We now perform SVA analysis to posterior in Eq. (11) directly and derive an optimization objective. The analysis is based on the fact that the CRP prior can be written as an exchangeable partition probability function (Pitman 1995; Aldous 1985):

$$p_0(\boldsymbol{z}|\alpha) = \alpha^{K-1}\frac{\Gamma(\alpha+1)}{\Gamma(\alpha+n)}\prod_{k=1}^{K}(n_k-1)!, \tag{33}$$

where K is the number of nonempty clusters. Since $\frac{\Gamma(\alpha+1)}{\Gamma(\alpha+n)}$ only depends on $\boldsymbol{D}$, the posterior (11) can be written as

$$q(\Theta) \propto \alpha^{K-1}\psi(\boldsymbol{n})\prod_{k=1}^{K} p_0(\boldsymbol{\mu}_k, \boldsymbol{\eta}_k)\prod_{i=1}^{n} p(\boldsymbol{x}_i|\boldsymbol{\mu}_{z_i})\phi(y_i|z_i,\boldsymbol{\eta}),$$

where $\psi(\boldsymbol{n}) := \prod_{k=1}^{K}(n_k-1)!$. Scaling $\boldsymbol{\theta}$, c and ν^2 by putting $\tilde{\boldsymbol{\theta}} = \beta\boldsymbol{\theta}$, $\tilde{c} = \beta' c$, $\tilde{\nu}^2 = \nu^2/\beta'$ and $\beta = s\beta'$, the posterior $q(\Theta)$ becomes[1]

$$q(\Theta|\boldsymbol{\beta}) \propto \alpha^K \psi(\boldsymbol{n})\left(g_{\tilde{\varphi}}\left(\frac{\boldsymbol{\tau}}{\beta},\frac{\kappa}{\beta}\right)\right)^K \prod_{i=1}^{n} f_{\tilde{\varphi}}(\boldsymbol{x}_i) \cdot \exp\left(-\beta'\mathcal{L}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\eta}) - \sum_{k=1}^{K}\kappa D_\varphi\left(\frac{\boldsymbol{\tau}}{\kappa},\boldsymbol{\mu}_k\right)\right), \tag{34}$$

where $\mathcal{L}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\eta}) = \sum_{k=1}^{K}\frac{\|\boldsymbol{\eta}_k\|^2}{2\nu^2} + 2c\sum_{i=1}^{n}(\zeta_i^{z_i})_+ + s\sum_{i=1}^{n} D_\varphi(\boldsymbol{x}_i, \boldsymbol{\mu}_{z_i})$. Assuming the scaling of the concentration parameter α in Eq. (30) and noting the fact that $f_{\tilde{\varphi}}(\boldsymbol{x}_i)$ only depends on data $\boldsymbol{x}_i$, the posterior distribution (34) then concentrate on the values of $\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\eta}$ that minimize the loss function in Eq. (35) when β and β' approach infinity[2]:

$$\mathcal{L}^{\text{sv}}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\eta}) = \mathcal{L}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\eta}) + \lambda \cdot K. \tag{35}$$

For the terms in the right-hand side of Eq. (35), we can see that the first and second terms of $\mathcal{L}$ indicate the error of the supervised classification task, one for the hinge loss and the other for the regularization; the third term of $\mathcal{L}$ characterizes the inference error on input feature vectors; and the last term of $\mathcal{L}^{\text{sv}}$ is a penalty term on model complexity, which resembles the classic AIC criteria (Akaike 1974). This penalty term originates from the CRP prior and was also derived in the work of (Kulis and Jordan 2012).

[1] We multiply α into the posterior to simplify scaling. It doesn't change the posterior as α is a constant independent of $\boldsymbol{z}$, $\boldsymbol{\mu}$ and $\boldsymbol{\eta}$.

[2] Actually we only need $\alpha = \left(g_{\tilde{\varphi}}\left(\frac{\boldsymbol{\tau}}{\beta},\frac{\kappa}{\beta}\right)\right)^{-1}\cdot\exp(-\beta'\lambda)$, but the extra terms in Eq. (30) do not affect the result.

Theorem 1 (with a proof in Appendix B) characterizes the consistency between Alg. 1 and the objective in Eq. (35). Specifically, it states that each iteration of Alg. 1 decreases the loss function in Eq. (35) monotonically. This consistency is desirable, as we are adopting the same set of scaling assumptions in small-variance asymptotic analysis on both the Gibbs sampler and the posterior distribution.

Theorem 1. *Let $\mathcal{L}^{sv}(\boldsymbol{z},\boldsymbol{\mu},\boldsymbol{\eta})$ be the loss function defined in Eq. (35). After each iteration of Algorithm 1, we have*

$$\mathcal{L}^{sv}(\boldsymbol{z}^{(g+1)},\boldsymbol{\mu}^{(g+1)},\boldsymbol{\eta}^{(g+1)}) \le \mathcal{L}^{sv}(\boldsymbol{z}^{(g)},\boldsymbol{\mu}^{(g)},\boldsymbol{\eta}^{(g)}). \tag{36}$$

Experiments

Results on synthetic datasets

We first compare M^{2}DPM with various competitors on synthetic data generated under 2 settings. For each setting, we generate 20 datasets at random and report the average accuracy and running time. In each dataset, we randomly pick 80% instances for training and use the rest 20% for testing. Hyper-parameters are determined by 5-fold cross-validation on training data. The algorithm terminates when the relative change of the loss function is less than $\varepsilon = 10^{-3}$.

Setting I: we first assign each of the 1,000 instances to a cluster according to a CRP with the concentration parameter $\alpha = 1$. For each instance in cluster k, we then sample its 10-dimensional feature vector from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_k, \sigma^2\boldsymbol{I}_d)$, where we set $\sigma = 0.5$ and $\boldsymbol{\mu}_k = (k,\cdots,k)^\top$. Note that the instances from different clusters may overlap due to the random samples from Gaussian distributions. The maximum number of clusters is 10. For each cluster k, we sample a linear classifier $\boldsymbol{\eta}_k$ from a Gaussian distribution $\mathcal{N}(0,\boldsymbol{I}_d)$. We sample binary labels from a Bernoulli model

$$p(y_i = 1|\boldsymbol{x}_i, k) = \text{Sigmoid}\left(\boldsymbol{\eta}_k^\top(\boldsymbol{x}_i - \boldsymbol{\mu}_k)\right). \tag{37}$$

Setting II: we uniformly divide 10,000 10-dimensional instances into 10 clusters. For each cluster k, values of each feature are sampled from a uniform distribution $U[k-0.5, k+0.5]$, and a linear classifier is generated as in Setting I. The true labels are sampled from the same Bernoulli model as in Eq. (37). Note that the clusters are disjoint.

Table 1 shows the average accuracy and running time of M^{2}DPM (std in parentheses), compared to Gibbs iSVM[3], dpMNL (Shahbaba and Neal 2009) and other baseline algorithms. We also compare with a pipeline algorithm that first uses DP-means to cluster the data and then trains an SVM classifier for each cluster. The results show that M^{2}DPM achieves comparable performance with the best methods in both settings, while requires much less running time than other sampling methods (e.g. dpMNL, Gibbs-iSVM) and kernel SVMs (e.g. RBF-SVM) when the data size is large.

Though RBF-SVM has slightly higher accuracy in Setting II, M^{2}DPM has the advantage of identifying latent cluster structures. As a nonparametric method, M^{2}DPM can infer the number of clusters adaptively. Table 2 illustrates how the number of clusters is automatically resolved from

[3] iSVM using truncated mean-field is worse; omitted for space.

Table 1: Averaging testing accuracy (%) and running time (s) of different algorithms on synthetic datasets.

		Linear-SVM	RBF-SVM	MNL	dpMNL	DPmeans+SVM	Gibbs-iSVM	M^{2}DPM
Setting I	accuracy	66.4 (6.9)	69.5 (5.5)	66.2 (7.2)	68.8 (7.5)	70.8 (5.6)	70.9 (4.9)	71.1 (5.2)
	time (s)	0.1 (0.0)	0.1 (0.0)	0.2 (0.0)	29.8 (2.8)	0.1 (0.0)	4.2 (0.1)	0.1 (0.0)
Setting II	accuracy	54.4 (1.7)	65.5 (1.2)	54.6 (1.6)	54.6 (1.8)	58.9 (1.4)	62.9 (1.2)	64.4 (1.2)
	time (s)	11.1 (0.6)	14.3 (0.2)	1.2 (0.1)	126.2 (11.0)	2.4 (0.8)	40.7 (2.4)	3.6 (1.4)

Table 2: The actual number of components (K_0) compared to the one inferred by M^{2}DPM (K)

n_0	100	300	1000	3000	10000
K_0	8	9	11	13	14
K	8	8	11	12	14
time (s)	0.03	0.07	0.54	0.91	5.45

Table 3: Performance and running time of different models on the protein classification task.

Model	Accuracy (%)	F_1 (%)	Time (s)
MNL	50.0	41.2	2.9
LSVM	50.5	47.3	0.5
RBF-SVM	53.1	49.5	1.6
dpMNL	56.3	49.5	98.2
DP+SVM	51.2	47.9	0.2
Gibbs-iSVM	55.8	50.1	223.4
M^{2}DPM	54.6	49.9	8.1

datasets with different sizes. Here, we generated 10,000 10-dimensional instances from a DP mixture and used the first n_0 ones as training data for each run of M^{2}DPM, under the same set of hyper-parameters. Note that the first n_0 instances may belong to fewer latent clusters due to the nature of DP. We can see that M^{2}DPM gave quite accurate predictions of the number of latent clusters and it can adapt to datasets with different sizes very well. Table 2 also shows that the running time of M^{2}DPM scales near linearly with respect to the data size, a desirable property for learning on large datasets.

Results on Real Datasets

Protein Fold Classification The first real dataset was created in (Ding and Dubchak 2001) for protein fold classification. It consists of 698 samples, each classified into one of the 27 different 3-d folding patterns. Each sample is represented by its percentage composition of 20 amino acids as well as its length of the protein sequence. Our objective is to classify data samples into one of the 27 classes given these types of features. We follow the previous setup that uses 313 samples as training data and the reamining 385 as testing data. The classification accuracy, F_1 scores and running time are reported in Table 3. We select the hyper-parameters of M^{2}DPM by a 5-fold cross-validation on the training set.

We can see that our method outperforms all the other algorithms except dpMNL and Gibbs-iSVM, in terms of both accuracy and F_1 scores. Though dpMNL and Gibbs-iSVM give better accuracy, they require significantly longer time to train and predict. This demonstrates that our algorithm can perform training and predicting with extraordinarily fast speed while still maintaining adequate performance.

Detecting Parkinson's Disease The second real dataset is described in (Little et al. 2009) for detecting Parkinson's disease. It consists of 195 instances (i.e. patients), each associated with 22 real-valued features and a binary label indicating whether the patient has Parkinson's disease. We follow the previous setup (i.e., performing 5-fold cross-validation) and compare with the results reported in (Shahbaba and Neal 2009) in Table 4. The results are reported using hyper-parameters $\lambda = 150$, $s = 0.01$, $\nu = 1$ and $c = 2.5$, which were selected by a 5-fold cross-validation performed on the data set. The results show that our method achieves comparable performance compared to the best competitors while requires much less running time.

Table 4: Accuracy, F_1 scores and running time of different models on the Parkinson's disease detection dataset

Model	Accuracy (%)	F_1 (%)	Time (s)
MNL	85.6 (2.2)	79.1 (2.8)	0.1 (0.0)
LSVM	87.2 (2.3)	80.6 (2.8)	0.1 (0.0)
RBF-SVM	87.2 (2.7)	79.9 (3.2)	0.1 (0.0)
dpMNL	87.7 (3.3)	82.6 (2.5)	22.2 (1.4)
DP+SVM	86.2 (2.1)	78.9 (3.4)	0.1 (0.0)
Gibbs-iSVM	88.9 (1.5)	85.1 (1.3)	1.8 (0.0)
M^{2}DPM	88.7 (2.9)	82.4 (4.8)	0.1 (0.0)

Table 5: Characteristics of patients within each group

	Avg. age	Avg. stage (0-4)
Group I	65.9	1.74
Group II	67.0	1.71
Group III	65.3	1.30
Group IV	77.0	2.3
Group V	65.4	1.50

Apart from having good accuracy, M^{2}DPM is also able to cluster patients into several groups that have different characteristics. Our algorithm divides patients into five groups, and Table 5 shows average ages and disease stages of the five groups. Note that the age and disease stage information are not contained in the original dataset and are cited from Table 1 in (Little et al. 2009). It is clear that patients in the 4-th group are senior than patients in the other groups, and are in more serious conditions in terms of disease stages.

Conclusions

We present max-margin DP-means (M^{2}DPM), an efficient algorithm for both clustering and classification, with the number of clusters automatically resolved from data. M^{2}DPM was developed by performing small-variance asymptotic analysis to a Gibbs sampler of DP mixtures of SVMs. By exploring the similar analysis, we show that M^{2}DPM monotonically minimizes an objective function. Experiments on synthetic and real-world datasets demonstrate that M^{2}DPM runs much faster than various competitors while still maintaining accurate predictions.

Acknowledgments

The work is supported by the National Basic Research Program of China (No. 2013CB329403), National Natural Science Foundation of China (Nos. 61322308, 61332007), and Tsinghua University Initiative Scientific Research Program (No. 20121088071).

References

Agarwal, A., and Daume, H. 2010. A geometric view of conjugate priors. *Machine Learning* 81(1):99–113.

Akaike, H. 1974. A new look at the statistical model identification. *IEEE Transactions on Automatic Control* 19(6):716–723.

Aldous, D. 1985. Exchangeability and related topics. *École d'Été de Probabilités de Saint-Flour XIII1983* 1–198.

Banerjee, A.; Merugu, S.; Dhillon, I.; and Ghosh, J. 2005. Clustering with Bregman divergences. *Journal of Machine Learning Research (JMLR)* 6:1705–1749.

Barndorff-Nielsen, O. 1978. *Information and Exponential Families in Statistical Theory*. Wiley Publishers.

Blackwell, D., and MacQueen, J. 1973. Ferguson distributions via Polya urn schemes. *The Annals of Statistics* 353–355.

Bregman, L. 1967. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics* 7(3):200–217.

Broderick, T.; Kulis, B.; and Jordan, M. 2013. MAD-Bayes: MAP-based asymptotic derivations from Bayes. In *ICML*.

Crammer, K.; Dekel, O.; Keshet, J.; S., S.-S.; and Y., S. 2006. Online passive-aggressive algorithms. *Journal of Machine Learning Research (JMLR)* 7:551–585.

Devroye, L. 1986. *Non-uniform random variate generation*. Springer-Verlag.

Ding, C., and Dubchak, I. 2001. Multi-class protein fold recognition using support vector machines and neural networks. *Bioinformatics* 17(4):349–358.

Fu, Z.; Robles-Kelly, A.; and Zhou, J. 2010. Mixing linear SVMs for nonlinear classification. *IEEE Transactions on Neural Networks* 21(12):1963–1975.

Germain, P.; Lacasse, A.; Laviolette, F.; and Marchand, M. 2009. PAC-Bayesian learning of linear classifiers. In *ICML*.

Hannah, L.; Blei, D.; and Powell, W. 2011. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research (JMLR)* 12:1923–1953.

Hjort, N.; Holmes, C.; Mueller, P.; and Walker, S. 2010. *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press.

Jiang, Q.; Zhu, J.; Sun, M.; and Xing, E. 2012. Monte Carlo methods for maximum margin supervised topic models. In *NIPS*.

Jiang, K.; Kulis, B.; and Jordan, M. 2012. Small-variance asymptotics for exponential family Dirichlet process mixture models. In *NIPS*.

Kulis, B., and Jordan, M. 2012. Revisiting K-Means: New algorithms via Bayesian nonparametrics. In *ICML*.

Little, M.; McSharry, P.; Hunter, E.; Spielman, J.; and Ramig, L. 2009. Suitability of dysphonia measurements for telemonitoring of parkinson's disease. *IEEE Transactions on Biomedical Engineering* 56(4):1015–1022.

Michael, J.; Schucany, W.; and Haas, R. 1976. Generating random variates using transformations with multiple roots. *American Statistician* 30(2):88–90.

Pitman, J. 1995. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields* 102(2):145–158.

Polson, N., and Scott, S. 2011. Data augmentation for support vector machines. *Bayesian Analysis* 6(1):1–24.

Shahbaba, B., and Neal, R. 2009. Nonlinear models using Dirichlet process mixtures. *Journal of Machine Learning Research (JMLR)* 10:1829–1850.

Tanner, M., and Wong, W.-H. 1987. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association* 82(398):528–540.

Tipping, M., and Bishop, C. 1999. Probabilistic principle component analysis. *Journal of the Royal Statistical Society, Series B* 21(3):611–622.

van Dyk, D., and Park, T. 2008. Partially collapsed Gibbs samplers: Theory and methods. *Journal of the American Statistical Association* 103(482):790–796.

Zhang, A.; Zhu, J.; and Zhang, B. 2014. Max-margin infinite hidden Markov models. *JMLR W&CP* 32(1):315–323.

Zhu, J.; Chen, N.; Perkins, H.; and Zhang, B. 2014. Gibbs max-margin topic models with data augmentation. *Journal of Machine Learning Research (JMLR)* 15:1073–1110.

Zhu, J.; Chen, N.; and Xing, E. 2011. Infinite SVM: a Dirichlet process mixture of large-margin kernel machines. In *ICML*.

Zhu, J.; Chen, N.; and Xing, E. 2014. Bayesian inference with posterior regularization and applications to infinite latent SVMs. *Journal of Machine Learning Research (JMLR, in press)*.

Appendix A. Derivation of the $\boldsymbol{\mu}_k$ update rule

In Eq. (21) we have expressed the scaled conditional distribution of $\boldsymbol{\mu}_k$ in terms of κ'' and $\boldsymbol{\tau}''$, where $\kappa'' = \kappa + \beta n_k$ and $\boldsymbol{\tau}'' = \boldsymbol{\tau} + \beta \sum_{i \in \mathcal{N}_k} \boldsymbol{x}_i$. As $\beta \to \infty$, note that $\kappa' \to \infty$ and $\frac{\boldsymbol{\tau}''}{\kappa''} = \frac{1}{n_k} \sum_{i \in \mathcal{N}_k} \boldsymbol{x}_i$. Therefore, as β goes to infinity, the conditional distribution of $\boldsymbol{\mu}_k$ will concentrate on the empirical mean of the data in cluster k.

Appendix B. Proof to Theorem 1

In this section, we give a proof to Theorem 1, which says each iteration of Algorithm 1 decreases (or keeps the same) the loss function $\mathcal{L}^{sv}(\boldsymbol{z}, \boldsymbol{\mu}, \boldsymbol{\eta})$ in Eq. (35). The proof is simlilar to the analysis of the K-means algorithm.

Proof. Suppose there are K components before an iteration, with parameter $\boldsymbol{\mu}_{1:K}$ and classifiers $\boldsymbol{\eta}_{1:K}$. We prove that after each update of $\boldsymbol{z}$, $\boldsymbol{\mu}$, $\boldsymbol{\omega}$ and $\boldsymbol{\eta}$ the loss function $\mathcal{L}^{sv}$ does not increase.

Update of z_i: Suppose $z_i^{(g)} = k$ and $z_i^{(g+1)} = k'$. If $k' \leq K$ (i.e. no new component created), we have

$$\Delta \mathcal{L}^{sv} = Q_i(k') - Q_i(k)$$

where $Q_i(k)$ is the cost function defined in Eq. (31) and (32); otherwise, (i.e. $k' = K + 1$), putting $\boldsymbol{\mu}_{K+1} = \boldsymbol{x}_i$ and $\boldsymbol{\eta}_{K+1} = \boldsymbol{\eta}^*$, we have again

$$\Delta \mathcal{L}^{sv} = Q_i(k') - Q_i(k).$$

By the update rule of z_i, it is obvious that $Q_i(k') \geq Q_i(k)$ and hence the loss does not increase.

Update of $\boldsymbol{\mu}_k$: Let $\boldsymbol{X}^k = \{\boldsymbol{x}_i | z_i = k\}$ denote all instances in component k. Suppose $\boldsymbol{\mu}_k^{(g)} = \boldsymbol{\mu}$ and $\boldsymbol{\mu}_k^{(g+1)} = \boldsymbol{\mu}'$. We then have

$$\Delta \mathcal{L}^{sv} = s \cdot (h(\boldsymbol{\mu}) - h(\boldsymbol{\mu}')),$$

where

$$h(\boldsymbol{\mu}) = \sum_{\boldsymbol{x}_i \in \boldsymbol{X}^k} -D_\varphi(\boldsymbol{x}_i, \boldsymbol{\mu}) + \log f_\varphi(\boldsymbol{x}_i) = \log p(\boldsymbol{X}^k | \boldsymbol{\mu}).$$

Since $\boldsymbol{\mu}' = \frac{\sum_{\boldsymbol{x}_i \in \boldsymbol{X}^k} \boldsymbol{x}_i}{|\boldsymbol{X}^k|}$ is the MLE of $p(\boldsymbol{X}^k)$, we have $h(\boldsymbol{\mu}') \geq h(\boldsymbol{\mu})$ and hence the loss $\mathcal{L}^{sv}$ does not increase.

Update of $\boldsymbol{\eta}$ and $\boldsymbol{\omega}$: For each component k, the udpate of $\boldsymbol{\eta}_k$ and $\boldsymbol{\omega}$ is the same as the EM algorithm used in (Polson and Scott 2011) to train an SVM model on data $\boldsymbol{X}^k$. As a result, after an update process of $\boldsymbol{\eta}_k$ and $\boldsymbol{\omega}$, the learning loss

$$\frac{\|\boldsymbol{\eta}_k\|^2}{2\nu^2} + c \cdot \sum_{\boldsymbol{x}_i \in \boldsymbol{X}^k} (\zeta_i^k)_+$$

does not increase while the remaining part of the loss $\mathcal{L}^{sv}$ remains the same.

□

Appendix C. Extension to the Multi-class scenario

In the multi-class scenario, each class y in a particular component k is associated with a classifier $\boldsymbol{\eta}_{k,y}$, and we follow the work (**?**) to define the multi-class hinge loss $\phi^{\mathrm{m}}(y_i | z_i, \boldsymbol{\eta})$ as

$$\phi^{\mathrm{m}}(y_i | z_i, \boldsymbol{\eta}) = \exp(-2c \max_y (\Delta_i^y + \boldsymbol{\eta}_{z_i,y}^\top \boldsymbol{x}_i - \boldsymbol{\eta}_{z_i,y_i}^\top \boldsymbol{x}_i)),$$

where $\Delta_i^y = l \cdot \delta_{y,y_i}$ is the loss imposed on predicting y. As a result, when the component k and the class y is fixed, the conditional distribution of $\boldsymbol{\eta}_{k,y}$ can be expressed as

$$q(\boldsymbol{\eta}_{k,y} | \boldsymbol{z}) \propto \exp\left(-\frac{\|\boldsymbol{\eta}_{k,y}\|^2}{2\nu^2} - 2c \sum_i \delta_{z_i,k} (s_{ik}^y \zeta_{ik}^y - s_{ik}^y \boldsymbol{\eta}_{k,y}^\top \boldsymbol{x}_i)_+ \right), \quad (38)$$

where $s_{ik}^y = 1$ if $y = y_i$, $s_{ik}^y = -1$ if $y \neq y_i$ and $\zeta_{ik}^y = \max_{y' \neq y} (\Delta_i^{y'} + \boldsymbol{\eta}_{k,y'}^\top \boldsymbol{x}_i) - \Delta_i^y$.

It can be seen that the conditional distribution of $\boldsymbol{\eta}_{k,y}$ expressed in Eq. (38) is very similar to the one in Eq. (18), except that we replace l and y_i with $s_{ik}^y \zeta_{ik}^y$ and s_{ik}^y respectively. The Gibbs updates and updates in M^{2}DPM of $\boldsymbol{\eta}$ and $\boldsymbol{\omega}$ follow immediately. The updates for $\boldsymbol{z}$ and $\boldsymbol{\mu}$ also remain nearly the same, except that we should replace the hinge loss $\phi(y_i | z_i, \boldsymbol{\eta})$ with its multi-class version $\phi^{\mathrm{m}}(y_i | z_i, \boldsymbol{\eta})$.