Noname manuscript No.
(will be inserted by the editor)

Multiple Instance Learning via Gaussian Processes

Minyoung Kim and Fernando De la Torre

Received: date / Accepted: date

Abstract Multiple instance learning (MIL) is a binary classification problem with loosely supervised data where a class label is assigned only to a bag of instances indicating presence/absence of positive instances. In this paper we introduce a novel MIL algorithm using Gaussian processes (GP). The bag labeling protocol of the MIL can be effectively modeled by the sigmoid likelihood through the max function over GP latent variables. As the non-continuous max function makes exact GP inference and learning infeasible, we propose two approximations: the soft-max approximation and the introduction of witness indicator variables. Compared to the state-of-the-art MIL approaches, especially those based on the Support Vector Machine (SVM), our model enjoys two most crucial benefits: (i) the kernel parameters can be learned in a principled manner, thus avoiding grid search and being able to exploit a variety of kernel families with complex forms, and (ii) the efficient gradient search for kernel parameter learning effectively leads to feature selection to extract most relevant features while discarding noise. We demonstrate that our approaches attain superior or comparable performance to existing methods on several real-world MIL datasets including large-scale content-based image retrieval problems.

Keywords Multiple Instance Learning · Gaussian Processes · Kernel Machines · Probabilistic Models

Minyoung Kim
Department of Electronics and IT Media Engineering,
Seoul National University of Science & Technology
Tel.: +82-2-970-9020
Fax: +82-2-970-7903
E-mail: mikim21@gmail.com

Fernando De la Torre
The Robotics Institute,
Carnegie Mellon University, Pittsburgh, PA 15213, USA
Tel.: +1-412-268-4706
Fax: +1-412-268-5571
E-mail: ftorre@cs.cmu.edu

1 Introduction

The paper deals with the multiple instance problem, a very important topic in machine learning and data mining. We begin with its formal definition. In the standard supervised classification setup, we have training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ which are labeled at the instance level. In the binary classification setting, $y_i \in \{+1, -1\}$. In multiple instance learning (MIL) (Dietterich et al, 1997) problem, on the other hand, the assumption is rather loosen in the following manner: (i) one is given B bags of instances $\{\mathbf{X}_b\}_{b=1}^{B}$ where each bag, $\mathbf{X}_b = \{\mathbf{x}_{b,1}, \ldots, \mathbf{x}_{b,n_b}\}$, consists of n_b instances ($\sum_b n_b = n$), and (ii) the labels are provided only at the bag-level in a way that for each bag b, $Y_b = -1$ if $y_i = -1$ for *all* $i \in I_b$, and $Y_b = +1$ if $y_i = +1$ for *some* $i \in I_b$, where I_b indicates the index set for instances in bag b.

The MIL is considered to be more realistic than standard classification setup due to its notion of bag in conjunction with the bag labeling protocol. Two most typical applications are image retrieval (Zhang et al, 2002; Gehler and Chapelle, 2007) and text classification (Andrews et al, 2003). For instance, the content-based image retrieval fits well the MIL framework as an image can be seen as a bag comprised of smaller regions/patches (i.e., instances). Given a query for a particular object, one may be interested in deciding only whether the image contains the queried object ($Y_b = +1$) or not ($Y_b = -1$), instead of solving the more involved (and probably less relevant) problem of labeling every single patch in the image. In text classification, one is more concerned with the concept/topic (i.e., bag label) of an entire paragraph than labeling each of the sentences that comprise the paragraph. The MIL framework is also directly compatible with other application tasks including the object detection (Viola et al, 2005) and the identification of proteins (Tao et al, 2004).

Although we only consider the original MIL problem defined as above, there are other alternative problems and generalization. For instance, the *multiple instance regression* deals with the real-valued outputs instead of binary labels (Dooly et al, 2002; Ray and Page, 2001), and the *multiple instance clustering* tackles the multiple instance problems in unsupervised situations (Zhang and Zhou, 2009; Zhang et al, 2011). The MIL problem can be extended to more generalized forms. One typical generalization is to modify the bag positive condition to be determined by some *collection* of positive instances, instead of a single one (Scott et al, 2003). Essentially and more generally, one can obtain other types of generalized MIL problems by specifying how the collection of underlying instance-level concepts is combined to form the label of the bag (Weidmann et al, 2003).

Traditionally, the MIL problem was tackled by specially tailored algorithms; for example, the hypothesis class of axis-parallel rectangles in the feature space has been introduced in (Dietterich et al, 1997), which is iteratively estimated to contain instances from positive bags. In (Maron and Lozano-Perez, 1998), the so-called diverse density (DD) is defined to measure proximity between a bag and a positive intersection point. Another line of research considers the MIL problem as a standard classification problem at a bag-level via proper development of kernels or distance measures on the bag space (Wang and Zucker, 2000; Gärtner et al, 2002; Tao et al, 2004; Chen et al, 2006). Particularly it subsumes the set kernels for SVMs (Gärtner

et al, 2002; Tao et al, 2004; Chen et al, 2006) and the Hausdorff set distances (Wang and Zucker, 2000).

A different perspective that regards the MIL as a missing-label problem was recently emerged. Unlike the negative instances which are all labeled negatively, the labels of instances in the positive bags are considered as latent variables. The latent labels have additional positive bag constraints, namely that at least one of them is positive, or equivalently, $\sum_i \frac{y_i+1}{2} \geq 1$ for $i \in I_b$ such that $Y_b = +1$. In this treatment, a fairly straightforward approach would be to formulate a standard (instance-level) classification problem (e.g., SVM) that can be optimized over the model and the latent variables simultaneously. The *mi-SVM* approach of (Andrews et al, 2003) is derived in this manner.

More specifically, the following optimization problem is solved for both the hyperplane parameter vector $\mathbf{w}$ and the output variables $\{y_i\}$:

$$\begin{aligned} \min_{\{y_i\},\mathbf{w},\{\xi_i\}} \; & \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ \text{s.t. } & \xi_i \geq 1 - y_i\mathbf{w}^\top\mathbf{x}_i, \;\; \xi_i \geq 0 \;\text{ for all }\; i, \\ & y_i = -1 \;\text{ for all }\; i \in I_b \;\text{ s.t. }\; Y_b = -1. \\ & \sum_{i\in I_b} \frac{y_i+1}{2} \geq 1 \;\text{ for all }\; b \;\text{ s.t. } Y_b = +1. \end{aligned} \tag{1}$$

Although the latent instance label treatments are mathematically appealing, a drawback of such approaches is that they involve a (mixed) integer programming which is generally difficult to solve. There have been several heuristics or approximate solutions such as those proposed in (Andrews et al, 2003). Recently, the deterministic annealing (DA) algorithm has been employed (Gehler and Chapelle, 2007), which approximates the original problem to a continuous optimization by introducing binary random variables in conjunction with the temperature-scaled (convex) entropy term. The DA algorithm begins with a high temperature to solve a relatively easy convex-like problem, and iteratively reduces the temperature with the warm starts (initialized at the solution obtained from the previous iteration).

Instead of dealing with all instances in a positive bag individually, a more insightful strategy is to focus on the *most positive* instance, often referred to as the *witness*, which is responsible for determining the label of a positive bag. In the SVM formulation, the *MI-SVM* of (Andrews et al, 2003) directly aims at maximizing the margin of the instance with the most positive confidence w.r.t. the current model $\mathbf{w}$ (i.e., $\max_{i\in I_b}\langle \mathbf{w}, \mathbf{x}_{b,i}\rangle$). An alternative formulation has been introduced in the *MICA* algorithm (Mangasarian and Wild, 2008), where they indirectly form a witness using convex combination over all instances in a positive bag. The *EM-DD* algorithm of (Zhang et al, 2002) extends the diverse density framework of (Maron and Lozano-Perez, 1998) by incorporating the witnesses. In (Gehler and Chapelle, 2007) the DA algorithms have also been applied to the witness-identifying SVMs, exhibiting superior prediction performance to existing approaches.

Even though some of these MIL algorithms, especially the SVM-based discriminative methods, are quite effective for a variety of situations, most approaches are

non-probabilistic, thus unable to capture the underlying generative process of the MIL data formation. In this paper[1] we introduce a novel MIL algorithm using the Gaussian process (GP), which we call *GPMIL*. Motivated from the fact that a bag label is solely determined by the instance that has the highest confidence toward the positive class, we design the bag class likelihood as the sigmoid function over the maximum GP latent variables on the instances. By marginalizing out the latent variables, we have a nonparametric, nonlinear probabilistic model $P(Y_b|\mathbf{X}_b)$ that fully respects the bag labeling protocol of the MIL.

Dealing with a probabilistic bag class model is not completely new. For instance, the Noisy-OR model suggested by (Viola et al, 2005) represents a bag label probability distribution, where the learning is formulated within the functional gradient boosting framework (Friedman, 1999). A similar Noisy-OR modeling has recently been proposed with Bayesian treatment by (Raykar et al, 2008). In their approaches, however, the bag class model is built from the *instance-level* classification models $P(y_i|\mathbf{x}_i)$, more specifically, $P(Y_b = -1|\mathbf{X}_b) = \prod_{i \in I_b} P(y_i = -1|\mathbf{x}_i)$ and $P(Y_b = +1|\mathbf{X}_b) = 1 - P(Y_b = -1|\mathbf{X}_b)$, which may incur several drawbacks. First of all, it involves additional modeling effort for the instance-level classifiers, which may be unnecessary, or only indirectly relevant to the bag class decision. Moreover, the Noisy-OR model combines the instance-level classifiers in a product form, treating each instance independently. This ignores the impact of potential interaction among the neighboring instances, which may be crucial for the accurate bag class prediction. On the other hand, our GPMIL represents the bag class model directly without employing probably unnecessary instance-level classifiers. The interaction among the instances is also incorporated through the GP prior that essentially enforces the smoothness regularization along the neighboring structure of the instances.

In addition to the above-mentioned advantages, the most important benefit of the GPMIL, especially contrasted with the SVM-based approaches, is that the kernel hyperparameters can be learned in a principled manner (e.g., empirical Bayes), thus avoiding grid search and being able to exploit a variety of kernel families with complex forms. Another promising aspect is that the efficient gradient search for kernel parameter learning effectively leads to feature selection to extract most relevant features while discarding noise. One caveat of the GPMIL is that it is intractable to perform exact GP inference and learning due to the non-continuous max function. We remedy it by proposing two approximation strategies: the soft-max approximation and the use of witness indicator variables which can be further optimized by the deterministic annealing schedule. Both approaches often exhibit more accurate prediction than most recent SVM variants.

The paper is organized as follows. After briefly reviewing the Gaussian process and introducing notations used throughout the paper in Sec. 2, our GPMIL framework is introduced in Sec. 3 with the soft-max approximation for inference and learning. The witness variable based approximation for GPMIL is described in Sec. 4, while we

[1] It is an extension of our earlier work (conference paper) published in (Kim and De la Torre, 2010). We extend the previous work broadly in two aspects: i) More technical details and complete derivations are provided for Gaussian process and our approaches based on it, which makes the manuscript comprehensive and self-contained, and ii) The experimental evaluation includes more extensive multiple instance learning datasets including the SIVAL image retrieval database and the drug activity datasets.

also suggest the deterministic annealing optimization. After discussing relationship with existing MIL approaches in Sec. 5, the experimental results of the proposed methods on both synthetic data and real-world MIL benchmark datasets are provided in Sec. 6. We conclude the paper in Sec. 7.

2 Review on Gaussian Processes

In this section we briefly review the Gaussian process model. The Gaussian process is a nonparametric, nonlinear Bayesian regression model. For the expositional convenience, we first consider a *linear* regression from input $\mathbf{x} \in \mathbb{R}^d$ to output $f \in \mathbb{R}$:

$$f = \mathbf{w}^\top \mathbf{x} + \varepsilon, \quad \text{where } \mathbf{w} \in \mathbb{R}^d \text{ is the model parameter and } \varepsilon \sim \mathcal{N}(0, \eta^2). \quad (2)$$

Given n i.i.d. data points $\{(\mathbf{x}_i, f_i)\}_{i=1}^n$, where we often use vector notations, $\mathbf{f} = [f_1, \ldots, f_n]^\top$ and $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]^\top$, we can express the likelihood as:

$$P(\mathbf{f}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^n P(f_i|\mathbf{x}_i, \mathbf{w}) = \mathcal{N}(\mathbf{f}; \mathbf{X}\mathbf{w}, \eta^2 I). \quad (3)$$

We then turn this into a Bayesian nonparametric model by placing prior on $\mathbf{w}$ and marginalizing it out. With a Gaussian prior $P(\mathbf{w}) = \mathcal{N}(0, I)$, it is easy to see that

$$P(\mathbf{f}|\mathbf{X}) = \int P(\mathbf{f}|\mathbf{X}, \mathbf{w}) P(\mathbf{w}) d\mathbf{w} = \mathcal{N}(\mathbf{f}; 0, \mathbf{X}\mathbf{X}^\top + \eta^2 I) \rightsquigarrow \mathcal{N}(\mathbf{f}; 0, \mathbf{X}\mathbf{X}^\top). \quad (4)$$

Here, we let $\eta \to 0$ to have a noise-free model. Although we restrict ourselves to the training data, adding a new test pair $(\mathbf{x}_*, f_*)$ immediately leads to the following joint Gaussian by concatenating the test point with the training data, namely

$$P(\begin{bmatrix} f_* \\ \mathbf{f} \end{bmatrix} | \begin{bmatrix} \mathbf{x}_* \\ \mathbf{X} \end{bmatrix}) = \mathcal{N}\Big(\begin{bmatrix} f_* \\ \mathbf{f} \end{bmatrix}; \begin{bmatrix} 0 \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{x}_*^\top \mathbf{x}_* & \mathbf{x}_*^\top \mathbf{X}^\top \\ \mathbf{X}\mathbf{x}_* & \mathbf{X}\mathbf{X}^\top \end{bmatrix} \Big). \quad (5)$$

From (5), the predictive distribution for f_* is analytically available as a conditional Gaussian:

$$P(f_*|\mathbf{x}_*, \mathbf{f}, \mathbf{X}) = \mathcal{N}(f_*; \mathbf{x}_*^\top \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{f}, \mathbf{x}_*^\top \mathbf{x}_* - \mathbf{x}_*^\top \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1}\mathbf{X}\mathbf{x}_*). \quad (6)$$

A nonlinear extension of (4) is straightforward by replacing the finite dimensional vector $\mathbf{w}$ by an infinite dimensional nonlinear function $f(\cdot)$ [2]. The *Gaussian process* (GP) is a particular choice of prior distribution on functions, which is characterized by the *covariance function* (i.e., the kernel function) $k(\cdot, \cdot)$ defined on the input space. Formally, a GP with $k(\cdot, \cdot)$ satisfies:

$$\mathrm{Cov}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = k(\mathbf{x}_i, \mathbf{x}_j), \quad \text{for any } \mathbf{x}_i \text{ and } \mathbf{x}_j. \quad (7)$$

[2] We abuse the notation f to indicate either a function or a response variable evaluated at $\mathbf{x}$ (i.e., $f(\mathbf{x})$ interchangeably.

In fact, any distribution on f that satisfies (7) is a GP. For f that follows the GP prior with $k(\cdot,\cdot)$, marginalizing out f produces a nonlinear version of (4),

$$P(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}; 0, \mathbf{K}_\beta), \tag{8}$$

where $\mathbf{K}_\beta$ is the kernel matrix on the input data $\mathbf{X}$ (i.e., $[\mathbf{K}_\beta]_{i,j} = k_\beta(i,j)$). Here, β in the subscript indicates the (hyper)parameters of the kernel function (e.g., for the RBF kernel, β includes the length scale, the magnitude, and so on). As the dependency on β is clear, we will sometimes drop the subscript for notational convenience. For a new test input $\mathbf{x}_*$, by letting $\mathbf{k}(\mathbf{x}_*) = [k(\mathbf{x}_1,\mathbf{x}_*),\ldots,k(\mathbf{x}_n,\mathbf{x}_*)]^\top$, we have a predictive distribution for a test response f_*, similar to (6). That is,

$$P(f_*|\mathbf{x}_*,\mathbf{f},\mathbf{X}) = \mathcal{N}(f_*; \mathbf{k}(\mathbf{x}_*)^\top\mathbf{K}^{-1}\mathbf{f}, k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*)^\top\mathbf{K}^{-1}\mathbf{k}(\mathbf{x}_*)). \tag{9}$$

When the GP is applied for regression or classification problems, we often treat $\mathbf{f}$ as *latent* random variables indexed by the training data samples, and introduce the actual (observable) output variables $\mathbf{y} = [y_1,\ldots,y_n]^\top$ which are linked to $\mathbf{f}$ through a likelihood model $P(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^n P(y_i|f_i)$. In the Appendix, we review in greater details two most popular likelihood models that yield GP regression and GP classification.

3 Gaussian Process Multiple Instance Learning (GPMIL)

In this section we introduce a novel Gaussian process (GP) model for the MIL problem, which we denote by *GPMIL*. Our approach builds a bag class likelihood model from the GP latent variables, where the likelihood is the sigmoid of the *maximum* latent variables.

Note that the bag b is comprised of n_b points $\mathbf{X}_b = \{\mathbf{x}_{b,1},\ldots,\mathbf{x}_{b,n_b}\}$. Accordingly, we assign the GP latent variables to the instances in the bag b, and we denote them by $\mathbf{F}_b = \{f_{b,1},\ldots,f_{b,n_b}\}$. One can regard f_i ($\in \mathbf{F}_b$) as a *confidence* score toward the (instance-level) positive class for $\mathbf{x}_i$ ($\in \mathbf{X}_b$). That is, the sign of f_i indicates the (instance-level) class label y_i, and its magnitude implies how confident it is. In the MIL, our goal is to devise a bag class likelihood model $P(Y_b|\mathbf{F}_b)$ instead of the instance-level model $P(y_i|f_i)$. Note that the latter is a special case of the former since an instance can be seen as a singleton bag. Once we have the bag class likelihood model, we can then marginalize out all the latent variables $\mathbf{F} = \{\mathbf{F}_b\}_{b=1}^B$ under the Bayesian formalism using the GP prior $P(\mathbf{F}|\mathbf{X})$ given the entire input $\mathbf{X} = \{\mathbf{X}_b\}_{b=1}^B$.

Now, consider the situation where the bag b is labeled as positive ($Y_b = +1$). The chance is determined solely by the single point that is the *most likely positive* (i.e., the largest f). The larger the confidence f, the higher the chance is. The other instances do not contribute to the bag label prediction no matter what their confidence scores are[3]. Hence, we can let:

$$P(Y_b = +1|\mathbf{F}_b) \propto \exp(\max_{i\in I_b} f_i). \tag{10}$$

[3] We provide a better insight about our argument here. It is true that any other instances can make a bag positive, but it is the instance with the highest confidence score (what we called *most likely positive* instance) that solely determines the bag label. In other words, to have an effect on the label of a bag, an instance needs to get the largest confidence score. In formal probability terms, the instance i can determine the bag label only for the events that assign the highest confidence to i. It is also important to note that even though $\max_{i\in B} f_i$ indicates a single instance, max function examines *all* instances in the bag to find it.

Similarly, the odds of the bag b being labeled as negative ($Y_b = -1$) is affected solely by the single point which is the *least likely negative*. As far as that point has a negative confidence f, the label of the bag is negative, and the larger the confidence $-f$, the higher the chance is. This leads to the model:

$$P(Y_b = -1|\mathbf{F}_b) \propto \exp(\min_{i \in I_b} -f_i). \tag{11}$$

Combining (10) and (11), we have the following bag class likelihood model:

$$P(Y_b|\mathbf{F}_b) = \frac{1}{1+\exp(-Y_b \max_{i \in I_b} f_i)}. \tag{12}$$

Note also that (12), in the limiting case where all the bags become singletons (i.e., classical supervised classification), is equivalent to the standard Gaussian process classification model with the sigmoid link[4].

When incorporating the likelihood model (12) into the GP framework, one bottleneck is that we have non-differentiable formulas due to the max function. We approximate it by the soft-max[5]: $\max(z_1, \ldots, z_m) \approx \log \sum_i \exp(z_i)$. This leads to the approximated bag class likelihood model:

$$\begin{aligned} P(Y_b|\mathbf{F}_b) &\approx \frac{1}{1+\exp(-Y_b \log \sum_{i \in I_b} e^{f_i})} \\ &= \frac{1}{1+(\sum_{i \in I_b} e^{f_i})^{-Y_b}}. \end{aligned} \tag{13}$$

Whereas the soft-max is often good approximation to the max function, it should be noted that unlike the standard GP classification with the sigmoid link, the negative log-likelihood $-\log P(Y_b|\mathbf{F}_b) = \log(1+(\sum_{i \in I_b} e^{f_i})^{-Y_b})$ is not a convex function of $\mathbf{F}_b$ for $Y_b = +1$ (although it is convex for $Y_b = -1$). This corresponds to a non-convex optimization in the approximated GP posterior computation and learning when the Laplace or variational approximation methods are adopted. However, using the (scaled) conjugate gradient search with different starting iterates, one can properly obtain a well-approximated posterior with a meaningful set of hyperparameters.

Before we proceed further to the details of inference and learning, we briefly discuss the benefits of the GPMIL compared to the existing MIL methods. As mentioned earlier, the GPMIL directly models the bag class distribution, without suboptimally introducing instance-level models such as the Noisy-OR model of (Viola et al, 2005). Also, framed in the GP framework, the posterior estimation and the hyperparameter learning can be accomplished by simple gradient search with similar complexity as the standard GP classification, while it enables probabilistic interpretation (e.g., uncertainty in prediction). Moreover, we have a principled way to learn the kernel hyperparameters under the Bayesian formalism, which is not properly handled by other kernel-based MIL methods.

[4] So, it is also possible to have a probit version of (12), namely $P(Y_b|\mathbf{f}_b) = \Phi(Y_b \max_{i \in I_b} f_i)$, where $\Phi(\cdot)$ is the cumulative normal function.

[5] It is well known that the soft-max provides relatively tight bounds for the max, $\max_{i=1}^m z_i \leq \log \sum_{i=1}^m \exp(z_i) \leq \max_{i=1}^m z_i + \log m$. Another nice property is that the soft-max is a convex function.

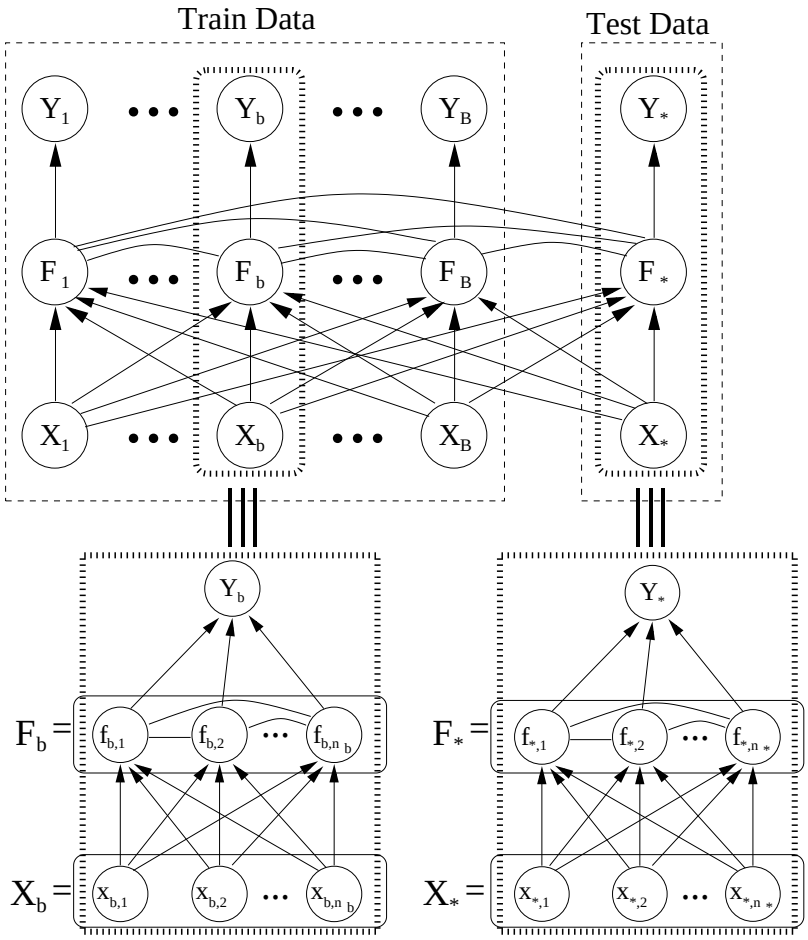

Fig. 1 Graphical model for GPMIL.

3.1 Posterior, Evidence, and Prediction

From the latent-to-output likelihood model (13), our generative GPMIL model can be depicted in a graphical representation as Fig. 1. Following the GP framework, all the latent variables $\mathbf{F} = \{\mathbf{F}_1, \ldots, \mathbf{F}_B\} = \{f_{b,i}\}_{b,i}$ are dependent on one another as well as on all the training input points $\mathbf{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_B\} = \{\mathbf{x}_{b,i}\}_{b,i}$, conforming to the following distribution:

$$P(\mathbf{F}|\mathbf{X}) = \mathcal{N}(\mathbf{F}; 0, \mathbf{K}), \tag{14}$$

Similarly, for a new test bag $\mathbf{X}_* = \{\mathbf{x}_{*,1}, \ldots, \mathbf{x}_{*,n_*}\}$ together with the corresponding latent variables $\mathbf{F}_* = \{f_{*,1}, \ldots, f_{*,n_*}\}$, we have a joint Gaussian prior on the concatenated latent variables, $\{\mathbf{F}_*, \mathbf{F}\}$, from which the predictive distribution on $\mathbf{F}_*$ can be

derived as (by conditional Gaussian):

$$P(\mathbf{F}_*|\mathbf{X}_*,\mathbf{F},\mathbf{X}) = \mathscr{N}\Big(\mathbf{F}_*;\ \mathbf{k}(\mathbf{X}_*)^\top\mathbf{K}^{-1}\mathbf{F},\ k(\mathbf{X}_*,\mathbf{X}_*) - \mathbf{k}(\mathbf{X}_*)^\top\mathbf{K}^{-1}\mathbf{k}(\mathbf{X}_*)\Big), \quad (15)$$

where $\mathbf{k}(\mathbf{X}_*)$ is the $(n \times n_*)$ train-test kernel matrix whose ij-th element is $k(\mathbf{x}_i, \mathbf{x}_{*,j})$, and $k(\mathbf{X}_*,\mathbf{X}_*)$ is the $(n_* \times n_*)$ test-test kernel matrix whose ij-th element is $k(\mathbf{x}_{*,i}, \mathbf{x}_{*,j})$.

Under the usual i.i.d. assumption, the entire likelihood $P(\mathbf{Y} = [Y_1,\ldots,Y_B]|\mathbf{F})$ is the product of the individual bag likelihoods $P(Y_b|\mathbf{F}_b)$ in (13). That is,

$$P(\mathbf{Y}|\mathbf{F}) = \prod_{b=1}^{B} P(Y_b|\mathbf{F}_b) \approx \prod_{b=1}^{B} \frac{1}{1+(\sum_{i\in I_b} e^{f_i})^{-Y_b}}. \quad (16)$$

Equipped with (14) and (16), one can compute the posterior distribution $P(\mathbf{F}|\mathbf{Y},\mathbf{X}) \propto P(\mathbf{F}|\mathbf{X})P(\mathbf{Y}|\mathbf{F})$ and the evidence (or the data likelihood) $P(\mathbf{Y}|\mathbf{X}) = \int_{\mathbf{F}} P(\mathbf{F}|\mathbf{X})P(\mathbf{Y}|\mathbf{F})$, where the GP learning maximizes the evidence w.r.t. the kernel hyperparameters (also known as the empirical Bayes). However, similar to the GP classification cases, the non-Gaussian likelihood term (16) causes intractability in the exact computation, and we resort to some approximation. Here we focus on the Laplace approximation[6] where its application to standard GP classification is very popular and reviewed in Appendix.

The Laplace approximation essentially replaces the product $P(\mathbf{Y}|\mathbf{F})P(\mathbf{F}|\mathbf{X})$ by a Gaussian with the mean equal to the mode of the product, and the covariance equal to the inverse Hessian of the product evaluated at the mode. For this purpose, we rewrite

$$\begin{aligned} &P(\mathbf{Y}|\mathbf{F})P(\mathbf{F}|\mathbf{X}) = \exp(-S(\mathbf{F}))\cdot|\mathbf{K}|^{-1/2}\cdot(2\pi)^{-n/2}, \\ &\text{where}\ \ S(\mathbf{F}) = \sum_{b=1}^{B} l(Y_b,\mathbf{F}_b) + \frac{1}{2}\mathbf{F}^\top\mathbf{K}^{-1}\mathbf{F}, \\ &\qquad l(Y_b,\mathbf{F}_b) = -\log P(Y_b|\mathbf{F}_b) \approx \log\Big(1+(\sum_{i\in I_b} e^{f_i})^{-Y_b}\Big). \end{aligned} \quad (17)$$

We first find the minimum of $S(\mathbf{F})$, namely

$$\widehat{\mathbf{F}} = \arg\min_{\mathbf{F}} S(\mathbf{F}), \quad (18)$$

where the optimum is denoted by $\widehat{\mathbf{F}}$. Solving (18) can be done by gradient search as usual. Unlike the standard GP classification, however, notice that $S(\mathbf{F})$ is a non-convex function of $\mathbf{F}$ since the Hessian of $S(\mathbf{F})$, $\mathbf{H}+\mathbf{K}^{-1}$, is generally not positive definite, where $\mathbf{H}$ is the block diagonal matrix whose b-th block has the ij-th entry $[\mathbf{H}_b]_{ij} = \frac{\partial^2 l(Y_b,\mathbf{F}_b)}{\partial f_i \partial f_j}$ for $i,j \in I_b$. Although this may hinder obtaining the global minimum easily, $S(\mathbf{F})$ is bounded below by 0 (from (17)), and the (scaled) conjugate or

[6] Although it is feasible, here we do not take the variational approximation into consideration for simplicity. Unlike the standard GP classification, it is difficult to perform, for instance, the Expectation Propagation (EP) approximation since the moment matching, the core step in EP that minimizes the KL divergence between the marginal posteriors, requires the integration over the likelihood function in (13), which requires further elaboration.

Newton-type gradient search with different initial iterates can yield a reliable solution.

We then approximate $S(\mathbf{F})$ by a quadratic function using its Hessian evaluated at $\widehat{\mathbf{F}}$, namely $\mathbf{H}(\widehat{\mathbf{F}})+\mathbf{K}^{-1}$. Yet, in order to enforce a convex quadratic form, we need to address the case that $\mathbf{H}+\mathbf{K}^{-1}$ is not positive definite, which although very rare, could happen as gradient search only discovers a point close (not exactly the same) to local minima. We approximate it to the closest positive definite matrix by projecting it onto the PSD cone. More specifically, we let $\mathbf{Q}\approx\mathbf{H}+\mathbf{K}^{-1}$, with $\mathbf{Q}=\sum_i \max(\lambda_i,\varepsilon)\mathbf{v}_i\mathbf{v}_i^\top$, where λ and $\mathbf{v}$ are the eigenvalues/vectors of $\mathbf{H}+\mathbf{K}^{-1}$, and ε is a small positive constant. In this way $\mathbf{Q}$ is a positive definite matrix closest to the Hessian with precision ε. Letting $\widehat{\mathbf{Q}}$ be $\mathbf{Q}$ evaluated at $\widehat{\mathbf{F}}$, we approximate $S(\mathbf{F})$ by the following quadratic function (i.e., using the Taylor expansion)

$$S(\mathbf{F})\approx S(\widehat{\mathbf{F}})+\frac{1}{2}(\mathbf{F}-\widehat{\mathbf{F}})^\top\widehat{\mathbf{Q}}(\mathbf{F}-\widehat{\mathbf{F}}), \tag{19}$$

which leads to Gaussian approximation for $P(\mathbf{F}|\mathbf{Y},\mathbf{X})$

$$P(\mathbf{F}|\mathbf{Y},\mathbf{X})\approx\mathcal{N}(\mathbf{F};\widehat{\mathbf{F}},\widehat{\mathbf{Q}}^{-1}). \tag{20}$$

The data likelihood (i.e., evidence) immediately follows from the similar approximation,

$$P(\mathbf{Y}|\mathbf{X},\theta)\approx\exp(-S(\widehat{\mathbf{F}}))|\widehat{\mathbf{Q}}|^{-1/2}|\mathbf{K}|^{-1/2}. \tag{21}$$

We then maximize (21) (so-called *evidence maximization* or *empirical Bayes*) w.r.t. the kernel parameters θ by gradient search.

More specifically, the negative log-likelihood, $NLL=-\log P(\mathbf{Y}|\mathbf{X},\theta)$, can be approximately written as:

$$NLL=\sum_{b=1}^{B} l(\mathbf{Y}_b,\widehat{\mathbf{F}}_b)+\frac{1}{2}\widehat{\mathbf{F}}^\top\mathbf{K}^{-1}\widehat{\mathbf{F}}+\frac{1}{2}\log|\mathbf{I}+\mathbf{K}\mathbf{H}|. \tag{22}$$

The gradient of the negative log-likelihood with respect to a (scalar) kernel parameter θ_m (i.e., $\theta=\{\theta_m\}$) can then be derived easily as follows:

$$\frac{\partial NLL}{\partial\theta_m}=-\frac{1}{2}\alpha^\top\Big(\frac{\partial\mathbf{K}}{\partial\theta_m}\Big)\alpha+\frac{1}{2}tr\bigg((\mathbf{H}^{-1}+\mathbf{K})^{-1}\Big(\frac{\partial\mathbf{K}}{\partial\theta_m}\Big)\bigg)+\frac{1}{2}tr\bigg((\mathbf{H}+\mathbf{K}^{-1})^{-1}\Big(\frac{\partial\mathbf{H}}{\partial\theta_m}\Big)\bigg), \tag{23}$$

where

$$\alpha=\mathbf{K}^{-1}\widehat{\mathbf{F}}\ \text{ and }\ \Big(\frac{\partial\mathbf{H}}{\partial\theta_m}\Big)_{i,j}=tr\bigg(\Big(\frac{\partial\mathbf{H}_{i,j}}{\partial\mathbf{F}}\Big)^\top(\mathbf{I}+\mathbf{K}\mathbf{H})^{-1}\Big(\frac{\partial\mathbf{K}}{\partial\theta_m}\Big)\alpha\bigg). \tag{24}$$

Here, $tr(A)$ is the trace of the matrix A. In the implementation, one can exploit the fact that $\frac{\partial\mathbf{H}_{i,j}}{\partial\mathbf{F}}$ is highly sparse (only the corresponding block of $\mathbf{F}_b$ can be non-zero).

The overall learning algorithm is depicted in Algorithm 1.

Given a new test bag $\mathbf{X}_*=\{\mathbf{x}_{*,1},\ldots,\mathbf{x}_{*,n_*}\}$, it is easy to derive the predictive distribution for the corresponding latent variables $\mathbf{F}_*=\{f_{*,1},\ldots,f_{*,n_*}\}$. Using the

Algorithm 1 GPMIL Learning

Input: Initial guess θ, the tolerance parameter τ.
Output: Learned hyperparameters θ.
(a) Find $\widehat{\mathbf{F}}$ from (18) for current θ.
(b) Compute $\widehat{\mathbf{Q}}$ using the PSD cone projection.
(c) Maximize (21) w.r.t. θ.
if $||\theta - \theta^{old}|| > \tau$ **then**
  Go to (a).
else
  Return θ.
end if

Gaussian approximated posterior (20) together with the conditional Gaussian prior (15), we have:

$$\begin{aligned} P(\mathbf{F}_*|\mathbf{X}_*,\mathbf{Y},\mathbf{X}) &= \int P(\mathbf{F}_*|\mathbf{X}_*,\mathbf{F},\mathbf{X})P(\mathbf{F}|\mathbf{Y},\mathbf{X})d\mathbf{F} \\ &\approx \int P(\mathbf{F}_*|\mathbf{X}_*,\mathbf{F},\mathbf{X})\mathcal{N}(\mathbf{F};\widehat{\mathbf{F}},\widehat{\mathbf{Q}}^{-1})d\mathbf{F} \\ &= \mathcal{N}\Big(\mathbf{F}_*;\ \mathbf{k}(\mathbf{X}_*)^\top\mathbf{K}^{-1}\widehat{\mathbf{F}},\ k(\mathbf{X}_*,\mathbf{X}_*)+\mathbf{k}(\mathbf{X}_*)^\top(\mathbf{K}^{-1}\widehat{\mathbf{Q}}^{-1}\mathbf{K}^{-1}-\mathbf{K}^{-1})\mathbf{k}(\mathbf{X}_*)\Big). \end{aligned} \tag{25}$$

Finally, the predictive distribution for the test bag class label Y_* can be obtained by marginalizing out $\mathbf{F}_*$, namely

$$P(Y_*|\mathbf{X}_*,\mathbf{Y},\mathbf{X}) = \int P(\mathbf{F}_*|\mathbf{X}_*,\mathbf{Y},\mathbf{X})P(Y_*|\mathbf{F}_*)d\mathbf{F}_*. \tag{26}$$

The integration in (26) generally needs further approximation. If one is only interested in the mean prediction (i.e., the predicted class label), it is possible to approximate $P(\mathbf{F}_*|\mathbf{X}_*,\mathbf{Y},\mathbf{X})$ by a delta function at its mean (mode), $\mu := \mathbf{k}(\mathbf{X}_*)^\top\mathbf{K}^{-1}\widehat{\mathbf{F}}$, which yields the test prediction:

$$\text{Class}(Y_*) \approx \text{sign}\left(\frac{1}{1+(\sum_{i\in *} e^{\mu_i})^{-1}} - 0.5\right). \tag{27}$$

4 GPMIL using Witness Variables

Although the approach in Sec. 3 is reasonable, one drawback is that the target function we approximate (i.e., $S(\mathbf{F})$) is not in general a convex function (due to the non-convexity of $-\log P(Y_b|\mathbf{F}_b)$), where we perform the PSD projection step to find the closest convex function in the Laplace approximation. In this section, we address this issue in a different way by introducing the so-called *witness latent variables* which indicate the most probably positive instances in the bags.

For each bag b, we introduce the witness indicator random variables $\mathbf{P}_b = [p_{b,1},\ldots,p_{b,n_b}]^\top$, where $p_{b,i}$ represents the probability that $\mathbf{x}_{b,i}$ is considered as a *witness* of the bag b. We call an instance a *witness* if it contributes to the likelihood $P(Y_b|\mathbf{F}_b)$. Note that $\sum_i p_{b,i} = 1$, and $p_{b,i} \geq 0$ for all $i \in I_b$. In the MIL, as $P(Y_b|\mathbf{F}_b)$ is solely dependent on

the most likely positive instance, it is ideal to put all the probability mass to a single instance as:

$$p_{b,i} = \begin{cases} 1 \text{ if } i = \arg\max_j f_{b,j} \\ 0 \text{ otherwise} \end{cases} \tag{28}$$

Alternatively, it is also possible to define a soft witness assignment[7] using a sigmoid function:

$$p_{b,i} = \frac{\exp(\lambda f_{b,i})}{\sum_{j \in I_b} \exp(\lambda f_{b,j})}, \tag{29}$$

where λ is the parameter that controls the smoothness of the assignment.

Once $\mathbf{P}_b$ is given, we then define the likelihood as a sigmoid of the weighted sum of f_i's with weights p_i's:

$$P(Y_b|\mathbf{F}_b, \mathbf{P}_b) = \frac{1}{1 + \exp(-Y_b \sum_i p_{b,i} f_{b,i})}. \tag{30}$$

The aim here is to replace the *max* or the *soft-max* function in the original derivation by the *expectation*, $\sum_i p_{b,i} f_{b,i}$, a linear function of $\mathbf{F}_b$ given the witness assignment $\mathbf{P}_b$. Notice that given $\mathbf{P}_b$, the negative log-likelihood of (30) is a convex function of $\mathbf{F}_b$.

In the full Bayesian treatment, one marginalizes out $\mathbf{P}_b$, namely

$$P(Y_b|\mathbf{F}_b) = \int P(Y_b|\mathbf{F}_b, \mathbf{P}_b) P(\mathbf{P}_b|\mathbf{F}_b) d\mathbf{P}_b, \tag{31}$$

where $P(\mathbf{P}_b|\mathbf{F}_b)$ is a Dirac's delta function with the point support given as (28) or (29). However, this simply leads to the very non-convexity raised by the original version of our GPMIL. Rather we pursue the coordinate-wise convex optimization by separating the process of approximating $P(Y_b|\mathbf{F}_b)$ into two individual steps: (i) find the witness indicator $\mathbf{P}_b$ from $\mathbf{F}_b$ using (28) or (29), and (ii) (while fixing $\mathbf{P}_b$) represent the likelihood as the sigmoid of the weighted sum (30), and perform posterior approximation. We alternate these two steps until convergence. Note that in this setting the Laplace approximation becomes quite similar to that of the standard GP classification, having the additional alternating optimization as an inner loop.

4.1 (Optional) Deterministic Annealing

When we adopt the soft witness assignment in the above formulation, it is easy to see that (29) is very similar to the probability assignment in the deterministic annealing (i.e., Eq. (11) of (Gehler and Chapelle, 2007)) while the smoothness parameter λ now acts as the inverse temperature in the annealing schedule. Motivated by this, we can have a annealed version of posterior approximation. More specifically, it initially begins with a small λ (large temperature) corresponding to a uniform-like $\mathbf{P}_b$, and repeats the followings: perform a posterior approximation starting from the optimum $\mathbf{F}_b$ in the previous stage to get a new $\mathbf{F}_b$, then increase λ to reduce the entropy of $\mathbf{P}_b$.

[7] This has a close relation to (Gehler and Chapelle, 2007)'s deterministic annealing approach to SVM. Similar to (Gehler and Chapelle, 2007), one can also consider a scheduled annealing, where the inverse of the smoothness parameter λ in (29) serves as the annealing temperature. See Sec. 4.1 for further details.

5 Related Work

This section briefly reviews/summarizes some typical approaches to MIL problems that are related to our models.

The pioneering work by (Dietterich et al, 1997) introduces the multiple instance problem, where they suggest a specific type of hypothesis class that can be learned iteratively to meet the positive bag/instance constraints. Since this work, there have been considerable research results on MIL problems. Most of the existing approaches can roughly belong to one of two different techniques: 1) directly learning a distance/similarity metric between bags, and 2) learning a predictor model while properly dealing with missing labels.

The former category includes: the diverse density (DD) of (Maron and Lozano-Perez, 1998) that aims to estimate proximity between a bag and a positive intersection point, the EM-DD of (Zhang et al, 2002) that extends the DD by introducing witness variables, and several kernel/distance measures proposed by (Wang and Zucker, 2000; Gärtner et al, 2002; Tao et al, 2004; Chen et al, 2006). In the other category, the most popular and sophisticated SVM framework has been exploited to find reasonable predictors. The *mi-SVM* (Andrews et al, 2003) is derived by formulating SVM-like optimization with the MIL's bag constraints. The difficult integer programming has been mitigated by the technique of deterministic annealing (Gehler and Chapelle, 2007).

Apart from instance-level predictors, the idea of focusing on the *most positive* instance or the *witness*, has been studied considerably. In the SVM framework, *MI-SVM* of (Andrews et al, 2003) directly maximizes the margin of the instance with the most positive confidence. Alternatively, the *MICA* algorithm (Mangasarian and Wild, 2008) parameterized witnesses as linear weighted sums over all instances in positive bags. Our GPMIL model can also be seen as a witness-based approach as the bag class likelihood is dominated by the maximally confident instance either via sigmoid soft-max modeling or via introducing witness random variables.

Some of the recent multiple instance algorithms have close relationships with the witness technique similar to ours. We briefly discuss two interesting approaches. In (Li et al, 2009), the CBIR problem is particularly considered where the regions of interest can be seen as witnesses or key instances in positive bags. They formed a convex optimization problem iteratively by finding violated key instances and combining them via multiple kernel learning. The optimization involves a series of standard SVM subproblems, and can be solved efficiently by a cutting plane algorithm. In (Liu et al, 2012), the key instance detection problem is tackled/formulated within a graph-based voting framework, which is formed either by a random walk or an iterative rejection algorithm.

We finally list some of more recent MIL algorithms. In (Antić and Ommer, 2012), a MIL problem is tackled by two alternating tasks of learning regular classifiers and imputing missing labels. They introduced the so-called *superbags*, a random ensemble of sets of bags, aimed for decoupling two tasks to avoid overfitting and improve robustness. Instead of building bag-level distance measures, (Wang et al, 2011) proposes a new approach of forming a *class-to-bag* distance metric. The goal is to reflect

the semantic similarities between class labels and bags. The maximum margin optimization is formulated and solved by parameterizing the distance metrics.

Apart from typical treatments that consider bags having a finite number of instances, the approach in (Babenko et al, 2011) regards bags as low dimensional manifolds embedded in high dimensional feature space. The geometric manifold structure of the manifold bags is then learned from data. This can be essentially seen as employing a particular bag kernel that preserves the geometric constraints that reside in data. In (Fu et al, 2011), they focus on the problem of efficient instance selection under large instance spaces. An adaptive instance selection algorithm is introduced, which alternates between instance selection and classifier learning in an iterative manner. In particular, the instance selection is seeded by a simple kernel density estimator on negative instances.

There are several unique benefits of having the GP framework in MIL problems. First, by using GP, choosing parameters of the kernel/model can be done in a principled manner (e.g., empirical Bayes of maximizing data likelihood) unlike some ad-hoc methods by SVM. Also, the parameters in GP models are random variables, and hence can be marginalized out within the Bayesian probabilistic framework to yield more flexible models. Furthermore, apart from other non-parametric kernel machines, one can interpret the underlying kernels more directly as the *covariance functions*, for which certain domain knowledge can be effectively exploited. In our MIL formulation, the bag label scoring model process is specifically treated as a covariance function over the input instance space. More importantly, we have observed empirically that the proposed GPMIL approaches achieve often times much more accurate prediction than existing methods including recent SVM-based MIL algorithms.

6 Experiments

In this section we conduct experimental evaluation for both artificially generated data and several real-world benchmark datasets. The latter includes the MUSK datasets (Dietterich et al, 1997), image annotation, and text classification datasets traditionally well-framed in multiple instance learning setup. Furthermore, we test the proposed algorithms on the large-scale content-based image retrieval task using the SIVAL dataset (Rahmani and Goldman, 2006).

We run two different approximation schemes for our GPMIL, which are denoted by: (a) `GP-SMX` = the soft-max approximation with the PSD projection described in Sec. 3, and (b) `GP-WDA` = the approximation using the witness indicator variables with the deterministic annealing optimization discussed in Sec. 4. In the `GP-SMX`, the GP inference/learning optimization is done by the (scaled) conjugate gradient search with different starting iterates. In the `GP-WDA`, we start from a large temperature (e.g., $\lambda = 1e-1$), and decrease it in log-scale (e.g., $\lambda \leftarrow 10 \cdot \lambda$) until there is no significant change in the quantities to be estimated. For both methods, we first estimate kernel hyperparameters by empirical Bayes (i.e., maximizing the evidence likelihood), then use the learned hyperparameters to the test prediction. The GPMIL is implemented in Matlab based on the publicly available GP codes from (Rasmussen and Williams, 2006).

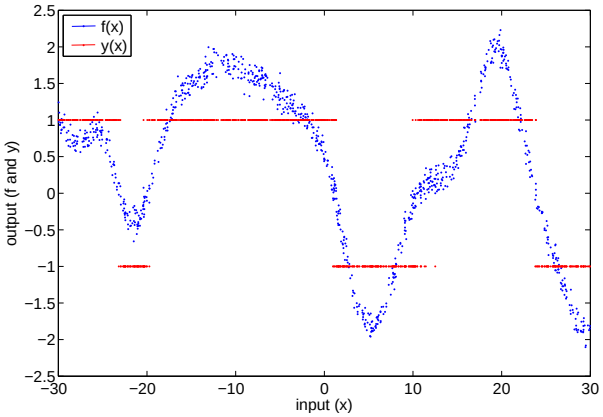

Fig. 2 Visualization of the synthetic 1D dataset. It depicts the instance-level input and output samples, where the bag formation is done by randomly grouping the instances. See text for details.

In the following section, we first demonstrate the performance of the proposed algorithms on artificially generated data.

6.1 Synthetic Data

In this synthetic setup, we test the capability of the GPMIL on estimating the kernel hyperparameters accurately from data. We construct the synthetic 1D dataset generated by a GP prior with random formation of the bags. The first step is to sample the input data points $\mathbf{x}$ uniformly from the real line $[-30,30]$. For the 1000 samples generated, the latent variables f are randomly generated from the GP prior distribution with the covariance matrix set equal to the (1000×1000) kernel matrix from the input samples. The kernel has a particular form, specifically the RBF kernel $k(\mathbf{x},\mathbf{x}') = \exp(-||\mathbf{x}-\mathbf{x}'||^2/2\sigma^2)$, where the hyperparameter is set to $\sigma = 3.0$. The RBF kernel form is assumed known to the algorithms, and the goal is to estimate σ as accurately as possible. Given the sampled f, the actual instance-level class output y is then determined by: $y = \mathtt{sign}(f)$. Fig. 2 depicts the instance-level input and output samples (i.e., f (and y) vs. $\mathbf{x}$).

To form the bags, we perform the following procedure. For each bag b, we randomly assign the bag label Y_b uniformly from $\{+1,-1\}$. The number of instances n_b is also chosen uniformly at random from $\{1,\ldots,10\}$. When $Y_b = -1$, we randomly select n_b instances from the negative instances of the 1000-sample pool. On the other hand, when $Y_b = +1$, we flip the 10-side fair coin to decide the positive instance portion $pp \in \{0.1,0.2,\ldots,1.0\}$, with which the bag is constructed from $\lceil pp \times n_b \rceil$ instances selected randomly from the positive instances and the rest (also randomly) from the negative instance pool. We generate 100 bags. We repeat the above procedure randomly 20 times.

We then perform the GPMIL hyperparameter learning starting from the initial guess $\sigma = 1.0$. We compute the average σ estimated for 20 trials. The results are: $\mathbf{3.2038 \pm 0.2700}$ for the `GP-SMX` approach, and $\mathbf{3.0513 \pm 0.2149}$ for the `GP-WDA`,

which are very close to the true value $\sigma = 3.0$. This experimental result highlights unique benefit of our GPMIL algorithms, namely that we can estimate the kernel parameters precisely in a principled manner, which is difficult to be achieved by other existing MIL approaches that rely on heuristic grid search on the hyperparameter space.

6.2 Competing Approaches

In this section we perform extensive comparison study of our GPMIL approaches against the state-of-the-art MIL algorithms. The competing algorithms are summarized below. The datasets, evaluation setups, and prediction results are provided in the following sections.

- **GP-SMX**: The proposed GPMIL algorithm that implements the soft-max approximation with the PSD projection.
- **GP-WDA**: The proposed GPMIL algorithm that incorporates the witness indicator random variables optimized by deterministic annealing.
- **mi-SVM**: The instance-level SVM formulation by treating the labels of instances in positive bags as missing variables to be optimized (Andrews et al, 2003).
- **MI-SVM**: The bag-level SVM formulation that aims to maximize the margin of the most positive instance (i.e., witness) with respect to the current model (Andrews et al, 2003).
- **AL-SVM**: The deterministic annealing extension of the instance-level **mi-SVM** by introducing binary random variables that indicate the positivity/negativity of the instances (Gehler and Chapelle, 2007).
- **ALP-SVM**: Further extension of **AL-SVM** by incorporating extra constraints on the expected number of positive instances per bag (Gehler and Chapelle, 2007).
- **AW-SVM**: The deterministic annealing extension of the witness-based **MI-SVM** approach (Gehler and Chapelle, 2007).
- **EMDD**: Probabilistic approach to find witnesses of the positive bags to estimate diverse densities (Zhang et al, 2002). We use Jun Yang's implementation, referred to as `Multiple Instance Learning Library`[8].
- **MICA**: SVM formulation that indirectly represents the witnesses using convex combination over instances in positive bags (Mangasarian and Wild, 2008). The linear program formulation for the MICA has been implemented in MATLAB.

Unless stated otherwise, all the kernel machines including our GPMIL algorithms employ the RBF kernel. For the SVM-based methods, the scale parameter of the RBF kernel is chosen as the median of the pairwise pattern distances. The hyperparameters are optimized using cross validation. Other parameters are selected randomly.

[8] Available at http://www.cs.cmu.edu/∼juny/MILL.

6.3 Standard Benchmark Datasets

6.3.1 The MUSK Datasets

The MUSK datasets (Dietterich et al, 1997) have served widely as the benchmark dataset for demonstrating performance of MIL algorithms. The datasets consist of the description of molecules using multiple low-energy conformations. The feature vector $\mathbf{x}$ is of 166-dimensional. There are two different types of bag formation denoted by MUSK1 and MUSK2, where the MUSK1 has approximately $n_b = 6$ conformations (instances) per bag, while the MUSK2 takes $n_b = 60$ instances per bag on average.

For comparison with the existing MIL algorithms, we follow the experimental setting similar to that of (Andrews et al, 2003; Gehler and Chapelle, 2007), where we conduct 10-fold cross validation. This is further repeated 5 times with different (random) partitions, and the average accuracies are reported. The test accuracies are shown in Table 1.

Our GPMIL algorithms, for both approximation strategies `WDA` and `SOFT-MAX`, exhibit superior classification performance to the existing approaches for the two MUSK datasets. One exception is the MICA where their reported error is the smallest on the MUSK2 dataset. This can be mainly due to the use of L1-regularizer in the MICA that yields a sparse solution suitable for the large-scale MUSK2 dataset. As is also alluded in (Gehler and Chapelle, 2007), it may not be directly comparable with the other methods.

6.3.2 Image Annotation

We test the algorithms on the image annotation datasets devised by (Andrews et al, 2003) from the COREL image database. Each image is treated as a bag comprised of the segments (instances) that are represented as feature vectors of color, text, and shape descriptors. Three datasets are formed for the object categories, tiger, elephant, and fox, regarding images containing the object as positive, and the rest as negative.

We follow the same setting as the original paper: There are 100/100 positive/negative bags, each of which contains $2 \sim 13$ instances. Similar to (Andrews et al, 2003; Gehler and Chapelle, 2007), we conduct 10-fold cross validation. This is further repeated 5 times with different (random) partitions. Table 1 shows the test accuracies. The proposed GPMIL algorithms achieve significantly higher accuracy than the best competing approaches most of the time. Comparing two approximation methods for GPMIL, `GP-WDA` often outperforms `GP-SMX`, implying that the approximation based on witness variables followed by a proper deterministic annealing schedule can be more effective than the soft-max approximation with the spectral convexification.

6.3.3 Text Classification

We next demonstrate the effectiveness of the GPMIL algorithm on the text categorization task. We use the MIL datasets provided by (Andrews et al, 2003) obtained from

Table 1 Test accuracies (%) on MUSK and Image Annotation Datasets. We report the accuracies of the proposed GPMIL algorithms, `GP-SMX` (soft-max approximation) and `GP-WDA` (witness variables with deterministic annealing). In AW-SVM and AL-SVM, for the two annealing schedules suggested by (Gehler and Chapelle, 2007), we only show the ones with smaller errors. Boldfaced numbers indicate the best results.

Dataset	GP-SMX	GP-WDA	mi-SVM	MI-SVM	AL-SVM	ALP-SVM	AW-SVM	EMDD	MICA
MUSK1	88.5 ± 3.5	$\mathbf{89.5 \pm 3.4}$	87.6 ± 3.5	79.3 ± 3.7	85.7 ± 3.0	86.5 ± 3.4	85.7 ± 3.1	84.6 ± 4.2	84.0 ± 4.4
MUSK2	87.9 ± 3.8	87.2 ± 3.7	83.8 ± 4.8	84.2 ± 4.5	86.3 ± 4.0	86.1 ± 4.7	83.4 ± 4.2	84.7 ± 3.2	$\mathbf{90.3 \pm 5.8}$
TIGER	87.1 ± 3.6	$\mathbf{87.4 \pm 3.6}$	78.7 ± 5.0	83.3 ± 3.3	78.5 ± 4.8	85.2 ± 3.2	82.7 ± 3.5	72.1 ± 4.0	81.3 ± 3.1
ELEPHANT	82.9 ± 4.0	$\mathbf{83.8 \pm 3.8}$	82.7 ± 3.6	81.5 ± 3.5	79.7 ± 2.1	82.8 ± 3.0	81.9 ± 3.4	77.5 ± 3.4	81.7 ± 4.7
FOX	63.2 ± 4.1	$\mathbf{65.7 \pm 4.9}$	58.7 ± 5.7	57.9 ± 5.5	63.7 ± 5.4	$\mathbf{65.7 \pm 4.3}$	63.3 ± 4.2	52.2 ± 5.9	58.3 ± 6.2

Table 2 Test accuracies (%) on text classification. Boldfaced numbers indicate the best results.

Dataset	GP-WDA	mi-SVM	MI-SVM	EMDD
TST1	**94.4**	93.6	93.9	85.8
TST2	**85.3**	78.2	84.5	84.0
TST3	86.1	**87.0**	82.2	69.0
TST4	**85.3**	82.8	82.4	80.5
TST7	80.3	**81.3**	78.0	75.4
TST9	**70.8**	67.5	60.2	65.5
TST10	**80.4**	79.6	79.5	78.5

the well-known TREC9 database. The original data are composed of 54000 MEDLINE documents annotated with 4903 subject terms, each defining a binary concept. Each document (bag) is decomposed into passages (instances) of overlapping windows of 50 or less words. Similar to the settings in (Andrews et al, 2003), a smaller subset is used, where the data are publicly available[9]. The dataset is comprised of 7 concepts (binary classification problems), each of which has roughly the same number (about 1600) of positive/negative instances from 200/200 positive/negative bags.

In Table 2 we report the average test accuracies of the GPMIL with the `WDA` approach, together with those of competing models from (Andrews et al, 2003). For MI-SVM and mi-SVM, only the linear SVM results are shown since the linear kernel outperforms polynomial/RBF kernels most of the time. In the GPMIL we also employ the linear kernel. We see that for a large portion of the problem sets, our GPMIL exhibits significant improvement over the methods provided in the original paper (EM-DD, mi-SVM, and MI-SVM).

6.4 Localized Content-Based Image Retrieval

The task of content-based image retrieval (CBIR) is perfectly fit to the MIL formulation. A typical setup of the CBIR problem is as follows: one is given a collection of training images where each image is labeled as $+1$ (-1) indicating existence (absence) of a particular concept or object in the image. Treating an entire image as a

[9] http://www.cs.columbia.edu/∼andrews/mil/datasets.html.

bag, and the regions/patches in the image as instances, it exactly reduces to the MIL problem: we only have bag-level labels where at least one positive region implies that the image is positive.

For this task, we use the SIVAL (Spatially Independent, Variable Area, and Lighting) dataset (Rahmani and Goldman, 2006). The SIVAL dataset is composed of 1500 images of 25 different object categories (60 images per category). The images of single objects are photographed with highly diverse backgrounds, while the spatial locations of the objects within images are arbitrary. This image acquisition setup can be more realistic and challenging than the popular COREL image database in which objects are mostly centered in the images occupying majorities of images.

The instance/bag formation is as follows: Each image is transformed into the YCbCr color space followed by pre-processing using a wavelet texture filter. This gives rise to six features (three colors and three texture features) per pixel. Then the image is segmented by the IHS segmentation algorithm (Zhang and Fritts, 2005). Each segment (instance) of the image is represented by the 30-dim feature vector by taking averages of color/texture features over pixels in the segment itself as well as those from its four neighbor segments (N, E, S, W). It ends up with 31 or 32 instances per bag.

We then form binary MIL problems via one-vs-all strategy (i.e., considering each of 25 categories as the positive class and the other categories as negative): For each category c, we take 20 random (positive) images from c, and randomly select one image from each of the classes other than c (that is, collecting 24 negative images). These 44 images serve as the training set, and all the rest of the images are used for testing. This procedure is repeated randomly for 5 times, and we report the average performance (with standard errors) in Table 3.

Since the label distributions of the test data are highly unbalanced (for each category, the negative examples take about 97% of the test bags), we used the AUROC (Area-Under ROC) measure instead of the standard error rates. In this result, we excluded MICA not only because its performance is worse than best performing ones for most categories, but also it often takes a large amount of time to converge to optimal solutions. As shown in the results, the proposed GPMIL models perform best most of the problem sets, exhibiting superb or comparable performance to the existing methods. The Gaussian process priors used in our models have effects of smoothing by interpolating the latent score variables across unseen test points, which is shown to be highly useful for improving generalization performance.

Table 3 Test accuracies (AUROC scores) on the SIVAL CBIR dataset. Boldfaced numbers indicate the best results within the margin of significance.

Category	GP-SMX	GP-WDA	mi-SVM	MI-SVM	AL-SVM	ALP-SVM	AW-SVM	EMDD
AjaxOrange	90.05 ± 8.61	**94.20 ± 8.48**	75.47 ± 5.20	63.57 ± 7.60	87.68 ± 3.53	83.72 ± 5.97	86.28 ± 12.66	56.83 ± 11.22
Apple	61.69 ± 2.69	**67.43 ± 3.11**	54.70 ± 4.24	47.20 ± 3.99	50.77 ± 4.93	52.45 ± 2.24	61.26 ± 6.54	54.63 ± 2.79
Banana	**67.69 ± 7.05**	**68.92 ± 4.51**	61.79 ± 5.72	55.82 ± 3.45	60.52 ± 4.62	62.05 ± 4.63	63.89 ± 4.87	59.86 ± 5.18
BlueScrunge	**72.04 ± 6.46**	68.13 ± 1.21	65.04 ± 7.66	67.17 ± 9.40	**71.94 ± 5.20**	67.65 ± 5.01	**71.82 ± 7.60**	66.04 ± 3.03
CandleWithHolder	**89.49 ± 3.35**	86.58 ± 7.54	80.85 ± 2.12	76.73 ± 4.47	76.81 ± 5.57	77.67 ± 5.34	84.70 ± 2.15	69.36 ± 5.86
CardboardBox	**76.35 ± 15.89**	73.53 ± 3.81	65.03 ± 4.03	64.93 ± 5.32	68.86 ± 4.26	67.31 ± 4.72	68.04 ± 3.33	58.42 ± 1.12
CheckeredScarf	**94.85 ± 5.36**	**92.29 ± 8.52**	81.44 ± 1.28	80.01 ± 2.27	88.04 ± 2.06	90.93 ± 2.93	88.63 ± 1.31	89.90 ± 2.37
CokeCan	**96.55 ± 3.14**	**95.14 ± 3.30**	93.61 ± 0.86	79.07 ± 6.74	92.49 ± 2.94	88.39 ± 4.32	92.45 ± 1.16	72.59 ± 5.20
DataMiningBook	**77.07 ± 11.68**	72.82 ± 5.40	69.82 ± 9.63	58.13 ± 2.84	**75.25 ± 6.02**	72.95 ± 6.19	**78.86 ± 6.35**	71.75 ± 7.26
DirtyRunningShoe	79.16 ± 3.18	**82.39 ± 4.88**	74.90 ± 4.83	67.73 ± 2.66	77.71 ± 1.93	**81.00 ± 3.34**	**82.17 ± 4.01**	**80.14 ± 3.89**
DirtyWorkGloves	61.32 ± 4.91	**80.96 ± 3.65**	73.86 ± 3.80	63.07 ± 4.67	76.77 ± 4.84	66.59 ± 3.14	76.96 ± 6.08	66.11 ± 3.12
FabricSoftnerBox	**96.13 ± 6.50**	94.94 ± 3.29	**95.53 ± 0.72**	83.17 ± 6.23	**96.17 ± 2.04**	93.20 ± 3.84	**97.52 ± 2.01**	75.65 ± 12.77
FeltFlowerRug	**92.98 ± 8.71**	87.80 ± 7.47	85.91 ± 2.12	84.52 ± 1.40	89.40 ± 1.89	89.39 ± 3.78	**90.75 ± 3.31**	76.42 ± 7.66
GlazedWoodPot	63.66 ± 8.42	67.40 ± 2.68	50.93 ± 4.34	47.73 ± 6.49	55.75 ± 4.40	57.41 ± 3.58	58.47 ± 5.69	**73.45 ± 6.68**
GoldMedal	82.82 ± 4.20	71.59 ± 5.19	83.40 ± 11.63	52.07 ± 8.25	**86.89 ± 2.79**	**86.18 ± 4.75**	**87.68 ± 4.06**	74.38 ± 6.90
GreenTeaBox	**93.73 ± 2.79**	**93.56 ± 5.00**	88.50 ± 6.93	86.64 ± 7.17	**95.47 ± 2.75**	92.67 ± 1.00	**93.48 ± 1.55**	79.92 ± 7.70
JuliesPot	87.23 ± 9.08	**91.78 ± 11.23**	82.12 ± 17.26	51.87 ± 3.29	84.37 ± 10.52	80.86 ± 10.58	88.88 ± 6.76	83.06 ± 8.39
LargeSpoon	60.01 ± 4.74	**63.30 ± 3.46**	54.16 ± 3.73	57.02 ± 3.18	54.38 ± 1.53	55.08 ± 1.81	54.13 ± 0.76	59.01 ± 1.49
RapBook	**67.43 ± 3.33**	**67.73 ± 6.60**	60.33 ± 3.14	57.18 ± 2.93	60.80 ± 4.60	60.59 ± 1.82	59.31 ± 3.78	55.78 ± 3.25
SmileyFaceDoll	**80.65 ± 2.28**	75.32 ± 5.76	75.52 ± 1.73	74.46 ± 5.98	**81.05 ± 4.54**	68.55 ± 4.72	**81.47 ± 9.52**	65.47 ± 6.77
SpriteCan	**80.31 ± 9.91**	**79.84 ± 7.29**	72.75 ± 3.21	75.38 ± 7.28	74.07 ± 5.91	75.99 ± 7.66	**78.57 ± 7.47**	64.36 ± 5.03
StripedNotebook	**89.29 ± 3.40**	**90.45 ± 5.40**	70.64 ± 7.34	63.26 ± 3.31	**88.10 ± 2.86**	81.08 ± 4.33	**88.90 ± 2.66**	61.47 ± 4.25
TranslucentBowl	**79.71 ± 5.79**	72.62 ± 3.75	**79.33 ± 8.82**	62.48 ± 3.02	**78.96 ± 5.32**	74.72 ± 4.48	77.12 ± 6.58	75.08 ± 6.90
WD40Can	90.41 ± 5.78	79.82 ± 4.42	88.99 ± 3.30	83.02 ± 2.66	**92.02 ± 1.83**	88.34 ± 3.05	**94.10 ± 1.17**	70.57 ± 6.23
WoodRollingPin	71.17 ± 6.73	**75.26 ± 4.47**	54.72 ± 1.62	61.72 ± 4.06	57.09 ± 2.28	59.09 ± 3.80	64.57 ± 2.91	58.90 ± 3.81

6.5 Drug Activity Prediction

Next we consider the drug activity prediction task with the publicly available artificial molecule dataset[10]. In the dataset, the artificial molecules were generated such that each feature value represents the distance from the molecular surface when aligned with the binding sites of the artificial receptor. A molecule bag is then comprised of all likely low-energy configurations or shapes for the molecule. More details on the data synthesis can be found in (Dooly et al, 2002).

The data generation process is quite similar to the MUSK datasets, while a notable aspect is that the real-valued labels are introduced. The label for a molecule is the ratio of the binding energy to the maximum possible binding energy given the artificial receptor, hence representing binding strength, which is real-valued between 0 and 1. We transform it to a binary classification setup by thresholding the label by 0.5.

We select two different types of datasets: one has 166-dim features and the other 283-dim. The former dataset is similar to the MUSK data while the latter aims to mimic the proprietary Affinity dataset from CombiChem (Dooly et al, 2002). In each of the two datasets, there are different setups by having different numbers of relevant features. For the 166-dim dataset, we have two setups of $r = 160$ and $r = 80$ where r indicates the number of relevant features (the rest features can be regarded as noise). For the 283-dim dataset, we consider four setups of $r = 160, 120, 80, 40$.

As the features are blend of relevant and noise ones, one can deal with feature selection. In our GP-based models, the feature selection can be gracefully achieved by the hyperparameter learning with the so-called ARD kernel under the GP framework. The ARD kernel is defined as follows, and allows individual scale parameter for each feature dimension.

$$k(\mathbf{x},\mathbf{x}') = \exp\Big(-\frac{1}{2}(\mathbf{x}-\mathbf{x}')^\top \mathbf{P}^{-1}(\mathbf{x}-\mathbf{x}')\Big), \quad \text{where } \mathbf{P} = diag(p_1^2,\ldots,p_d^2). \tag{32}$$

Here, d is the feature dimension, and $diag(\cdot)$ makes a diagonal matrix with its arguments. Learning the hyperparameters of the ARD kernel in our GPMIL models can be done by efficient gradient search under empirical Bayes (data likelihood maximization). For the other approaches, however, it should be noted that it is computationally infeasible to perform grid search or cross validation like optimization to select relevant features from the large feature dimensions.

For each data setup, we split the data into 180 training bags and 20 test bags, where each bag consists of $3 \sim 5$ instances. The procedure is repeated randomly for 5 times, and we report in Table 4 the means and standard deviations of the test accuracies. For our GPMIL models, the performance of the GP-WDA models are shown (as GP-SMX performs comparably), where we contrast the ARD kernel with the standard isotropic RBF kernel (denoted by ISO). To identify the dataset, we use the notation `#-relevant-features/#-features`, for instance, the dataset $40/283$ indicates that only 40 out of 283 features are relevant, and the rest are noise. For all datasets, each feature vector consists of the relevant features taking the first part, followed by the noise features.

[10] http://www.cs.wustl.edu/∼sg/multi-inst-data/.

Table 4 Test accuracies (%) on the drug activity prediction dataset.

Dataset	GP-WDA (ARD)	GP-WDA (ISO)	mi-SVM	MI-SVM	AL-SVM	ALP-SVM	AW-SVM	EMDD
160/166	100.00 ± 0.00	97.78 ± 4.97	100.00 ± 0.00	97.78 ± 4.97	97.78 ± 4.97	100.00 ± 0.00	100.00 ± 0.00	82.22 ± 9.94
80/166	97.78 ± 4.97	95.56 ± 6.09	77.78 ± 7.86	93.33 ± 6.09	86.67 ± 9.30	93.33 ± 6.09	95.56 ± 6.09	75.56 ± 9.30
160/283	100.00 ± 0.00	100.00 ± 0.00	96.00 ± 4.18	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	96.00 ± 8.94
120/283	100.00 ± 0.00	99.00 ± 2.24	88.00 ± 4.47	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	100.00 ± 0.00	86.00 ± 5.48
80/283	100.00 ± 0.00	98.00 ± 2.74	87.00 ± 6.71	97.00 ± 2.74	100.00 ± 0.00	97.00 ± 2.74	100.00 ± 0.00	82.00 ± 8.37
40/283	99.00 ± 2.24	94.00 ± 5.48	91.00 ± 5.48	92.00 ± 9.08	94.00 ± 4.18	93.00 ± 2.74	92.00 ± 2.74	80.00 ± 7.07

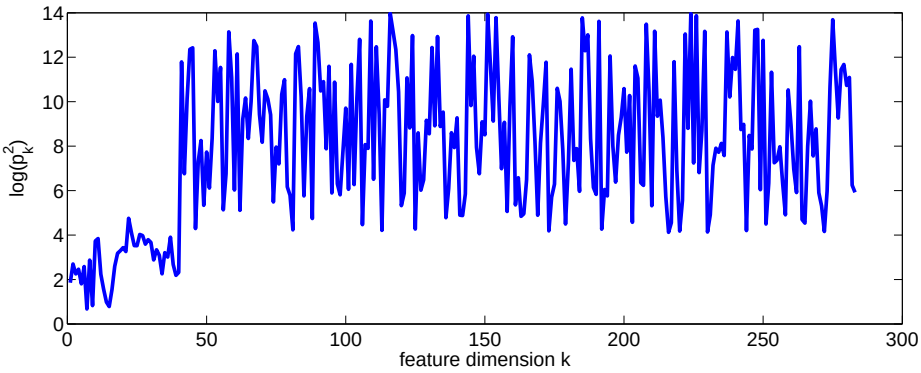

Fig. 3 Learned ARD kernel hyperparameters, $\log(p_k^2)$, $k = 1,\ldots,283$, for the 40/283 dataset. From the definition of ARD kernel (32), a lower value of p_k indicates that the corresponding feature dimension k is more informative.

As demonstrated in the results, most approaches perform equally well when the number of relevant features is large. On the other hand, as the portion of the relevant features decreases, the test performance degrades, and the feature selection becomes more crucial to the classification accuracy. Promisingly our GPMIL model equipped with the ARD kernel feature selection capability performs outstandingly, especially for the 40/283 dataset. For the GP-WDA (ARD), we also depict in Fig. 3 the learned ARD kernel hyperparameters in log-scale, that is, $\log(p_k^2)$ for $k = 1,\ldots,283$ for the 40/283 dataset. From the ARD kernel definition (32), a lower value of p_k indicates that the corresponding feature dimension is more informative. As shown in the figure, the first 40 relevant features are correctly recovered (low p_k values) by the GPMIL learning algorithm.

7 Conclusion

We have proposed novel MIL algorithms by incorporating bag class likelihood models in the GP framework, yielding nonparametric Bayesian probabilistic models that can capture the underlying generative process of the MIL data formation. Under the GP framework, the kernel parameters can be learned in a principled manner using

efficient gradient search, thus avoiding grid search and being able to exploit a variety of kernel families with complex forms. This capability has been further utilized for feature selection, which is shown to yield improved test performance. Moreover, our models provide probabilistic interpretation, informative for better understanding of the MIL prediction problem. To address the intractability in the exact GP inference and learning, we have suggested several approximation schemes including the softmax with the PSD projection and the introduction of the witness latent variables that can be optimized by the deterministic annealing. For several real-world benchmark MIL datasets, we have demonstrated that the proposed methods can yield superior or comparable prediction performance to the existing state-of-the-art approaches.

Unfortunately, the proposed GPMIL algorithms were usually slower in running time than SVM-based methods, mainly due to the overhead of matrix inversion in Gaussian process inference. This is a well-known issue/drawback of most GP-based methods nearly all the time, and we do not rigorously deal with the computational efficiency of the proposed methods here. However, there are several recent approaches to reduce computational complexity of GP inference (e.g., sparse GP or pseudo input methods (Snelson and Ghahramani, 2006; Lázaro-Gredilla et al, 2010)). Their application to our GPMIL framework is left as our future work.

Appendix: Gaussian Process Models and Approximate Inference

We review the Gaussian process regression and classification models as well as the Laplace method for approximate inference.

A.1 Gaussian Process Regression

When the output variable y is a real-valued scalar, one natural likelihood model from f to y is the *additive Gaussian noise* model, namely

$$P(y|f) = \mathcal{N}(y; f, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-f)^2}{2\sigma^2}\right), \tag{33}$$

where σ^2, the output noise variance, is another set of hyperparameters (together with the kernel parameters β). Given the training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and with (33), we can compute the posterior distribution for the training latent variables $\mathbf{f}$ analytically, namely

$$P(\mathbf{f}|\mathbf{y},\mathbf{X}) \propto P(\mathbf{y}|\mathbf{f})P(\mathbf{f}|\mathbf{X}) = \mathcal{N}(\mathbf{f}; \mathbf{K}(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}, (\mathbf{I}-\mathbf{K}(\mathbf{K}+\sigma^2\mathbf{I})^{-1})\mathbf{K}). \tag{34}$$

When the posterior of $\mathbf{f}$ is obtained, we can readily compute the predictive distribution for a test output y_* on $\mathbf{x}_*$. We first derive the posterior for f_*, the latent variable for the test point, by marginalizing out $\mathbf{f}$ as follows:

$$\begin{aligned} P(f_*|\mathbf{x}_*,\mathbf{y},\mathbf{X}) &= \int P(f_*|\mathbf{x}_*,\mathbf{f},\mathbf{X})P(\mathbf{f}|\mathbf{y},\mathbf{X})d\mathbf{f} \\ &= \mathcal{N}(f_*; \mathbf{k}(\mathbf{x}_*)^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}, k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*)^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}_*)). \end{aligned} \tag{35}$$

Then it is not difficult to see that the posterior for y_* can be obtained by marginalizing out f_*, namely

$$\begin{aligned} P(y_*|\mathbf{x}_*,\mathbf{y},\mathbf{X}) &= \int P(y_*|f_*)P(f_*|\mathbf{x}_*,\mathbf{y},\mathbf{X})df_* \\ &= \mathcal{N}(y_*; \mathbf{k}(\mathbf{x}_*)^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}, k(\mathbf{x}_*,\mathbf{x}_*) - \mathbf{k}(\mathbf{x}_*)^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{k}(\mathbf{x}_*) + \sigma^2). \end{aligned} \tag{36}$$

So far, we have assumed that the hyperparameters (denoted by $\theta = \{\beta, \sigma^2\}$) of the GP model are known. However, it is a very important issue to estimate θ from the data, the task often known as the *GP learning*. Following the fully Bayesian treatment, it would be ideal to place a prior on θ and compute a posterior distribution for θ as well, however, due to the difficulty of the integration, it is usually handled by the *evidence maximization*, also referred to as the *empirical Bayes*. The evidence in this case is the data likelihood,

$$P(\mathbf{y}|\mathbf{X},\theta) = \int P(\mathbf{y}|\mathbf{f},\sigma^2)P(\mathbf{f}|\mathbf{X},\beta)d\mathbf{f} = \mathcal{N}(\mathbf{y};\mathbf{0},\mathbf{K}_\beta + \sigma^2\mathbf{I}). \tag{37}$$

We then maximize (37), which is equivalent to solving the following optimization problem:

$$\theta^* = \arg\max_\theta \log P(\mathbf{y}|\mathbf{X},\theta) = \arg\min_\theta |\mathbf{K}_\beta + \sigma^2\mathbf{I}| + \mathbf{y}^\top(\mathbf{K}_\beta + \sigma^2\mathbf{I})^{-1}\mathbf{y}. \tag{38}$$

(38) is non-convex in general, and one can find a local minimum using the quasi-Newton or the conjugate gradient search.

A.2 Gaussian Process Classification

In the classification setting where the output variable y takes a binary[11] value from $\{+1,-1\}$, we have several choices for the likelihood model $P(y|f)$. Two most popular ways to link the real-valued variable f to the binary y are the sigmoid and the probit.

$$P(y|f) = \begin{cases} \frac{1}{1+\exp(-yf)} & \text{(sigmoid)} \\ \Phi(yf) & \text{(probit)} \end{cases} \tag{39}$$

where $\Phi(\cdot)$ is the cumulative normal function. We let $l(y,f;\gamma) = -\log P(y|f,\gamma)$, where γ denotes the (hyper)parameters of the likelihood model[12]. Likewise, we often drop the dependency on γ for the notational simplicity.

Unlike the regression case, it is unfortunate that the posterior distribution $P(\mathbf{f}|\mathbf{y},\mathbf{X})$ has no analytic form as it is a product of the non-Gaussian $P(\mathbf{y}|\mathbf{f})$ and the Gaussian $P(\mathbf{f}|\mathbf{X})$. One can consider three standard approximation schemes within the Bayesian framework: (i) Laplace approximation, (ii) variational methods, and (iii) sampling-based approaches (e.g., MCMC). It is often the cases that the third method is avoided due to the computational overhead (as we sample **f**s, n-dimensional vectors, where n is the number of training samples). In the below, we briefly review the Laplace approximation.

A.3 Laplace Approximation

The Laplace approximation essentially replaces the product $P(\mathbf{y}|\mathbf{f})P(\mathbf{f}|\mathbf{X})$ by a Gaussian with the mean equal to the mode of the product, and the covariance equal to the inverse Hessian of the product evaluated at the mode. More specifically, we let

$$S(\mathbf{f}) = -\log\big(P(\mathbf{y}|\mathbf{f})P(\mathbf{f}|\mathbf{X})\big) = \sum_{i=1}^n l(y_i,f_i) + \frac{1}{2}\mathbf{f}^\top\mathbf{K}^{-1}\mathbf{f} + \frac{1}{2}\log|\mathbf{K}| + \frac{n}{2}\log 2\pi. \tag{40}$$

Note that (40) is a convex function of **f** since the Hessian of $S(\mathbf{f})$, $\Lambda + \mathbf{K}^{-1}$, is positive definite where Λ is the diagonal matrix with entries $[\Lambda]_{ii} = \frac{\partial^2 l(y_i,f_i)}{\partial f_i^2}$. The minimum of $S(\mathbf{f})$ can be attained by a gradient search. We denote the optimum by $\mathbf{f}^{MAP} = \arg\max_\mathbf{f} S(\mathbf{f})$. Letting Λ^{MAP} be Λ evaluated at $\mathbf{f}^{MAP}$, we approximate $S(\mathbf{f})$ by the following quadratic function (i.e., using the Taylor expansion)

$$S(\mathbf{f}) \approx S(\mathbf{f}^{MAP}) + \frac{1}{2}(\mathbf{f}-\mathbf{f}^{MAP})^\top(\Lambda^{MAP} + \mathbf{K}^{-1})(\mathbf{f}-\mathbf{f}^{MAP}), \tag{41}$$

[11] We only consider binary classification where the extension to multiclass cases is straightforward.

[12] Although the models in (39) have no parameters involved, we can always consider more general cases. For instance, one may play with a generalized logistic regression (Zhang and Oles, 2000): $P(y|f,\gamma) \propto \big(\frac{1}{1+\exp(\gamma(1-yf))}\big)^\gamma$.

which essentially leads to a Gaussian approximation for $P(\mathbf{f}|\mathbf{y},\mathbf{X})$, namely

$$P(\mathbf{f}|\mathbf{y},\mathbf{X}) \approx \mathcal{N}(\mathbf{f};\mathbf{f}^{MAP},(\Lambda^{MAP}+\mathbf{K}^{-1})^{-1}). \tag{42}$$

The data likelihood (i.e., evidence) immediately follows from the similar approximation,

$$P(\mathbf{y}|\mathbf{X},\theta) \approx \exp(-S(\mathbf{f}^{MAP}))(2\pi)^{n/2}|\Lambda^{MAP}+\mathbf{K}_{\beta}^{-1}|^{-1/2}, \tag{43}$$

which can be maximized by gradient search with respect to the hyperparameters $\theta = \{\beta,\gamma\}$.

Using the Gaussian approximated posterior, it is easy to derive the predictive distribution for f_*:

$$P(f_*|\mathbf{x}_*,\mathbf{y},\mathbf{X}) = \mathcal{N}(f_*;\mathbf{k}(\mathbf{x}_*)^{\top}\mathbf{K}^{-1}\mathbf{f}^{MAP},k(\mathbf{x}_*,\mathbf{x}_*)-\mathbf{k}(\mathbf{x}_*)^{\top}((\Lambda^{MAP})^{-1}+\mathbf{K}^{-1})^{-1}\mathbf{k}(\mathbf{x}_*)). \tag{44}$$

Finally, the predictive distribution for y_* can be obtained by marginalizing out f_*, which has a closed form solution for the probit likelihood model as follows[13]:

$$P(y_*=+1|\mathbf{x}_*,\mathbf{y},\mathbf{X}) = \Phi\Big(\frac{\bar{\mu}}{\sqrt{1+\bar{\sigma}^2}}\Big), \tag{45}$$

where $\bar{\mu}$ and $\bar{\sigma}^2$ are the mean and the variance of $P(f_*|\mathbf{x}_*,\mathbf{y},\mathbf{X})$, respectively.

References

Andrews S, Tsochantaridis I, Hofmann T (2003) Support vector machines for multiple-instance learning. Advances in Neural Information Processing Systems

Antić B, Ommer B (2012) Robust multiple-instance learning with superbags. In Proceedings of the 11th Asian Conference on Computer Vision

Babenko B, Verma N, Dollár P, Belongie S (2011) Multiple instance learning with manifold bags. International Conference on Machine Learning

Chen Y, Bi J, Wang JZ (2006) MILES: Multiple-instance learning via embedded instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(12):1931–1947

Dietterich TG, Lathrop RH, Lozano-Perez T (1997) Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence 89:31–71

Dooly DR, Zhang Q, Goldman SA, Amar RA (2002) Multiple-instance learning of real-valued data. Journal of Machine Learning Research 3:651–678

Friedman J (1999) Greedy function approximation: a gradient boosting machine. Technical Report, Dept. of Statistics, Stanford University

Fu Z, Robles-Kelly A, Zhou J (2011) MILIS: Multiple instance learning with instance selection. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5):958–977

Gärtner T, Flach PA, Kowalczyk A, Smola AJ (2002) Multi-instance kernels. International Conference on Machine Learning

Gehler PV, Chapelle O (2007) Deterministic annealing for multiple-instance learning. AI and Statistics (AISTATS)

Kim M, De la Torre F (2010) Gaussian process multiple instance learning. International Conference on Machine Learning

Lázaro-Gredilla M, Quinonero-Candela J, Rasmussen CE, Figueiras-Vidal AR (2010) Sparse spectrum Gaussian process regression. Journal of Machine Learning Research 11:1865–1881

Li YF, Kwok JT, Tsang IW, Zhou ZH (2009) A convex method for locating regions of interest with multi-instance learning. In Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD'09), Bled, Slovenia

Liu G, Wu J, Zhou ZH (2012) Key instance detection in multi-instance learning. In Proceedings of the 4th Asian Conference on Machine Learning (ACML'12), Singapore

Mangasarian OL, Wild EW (2008) Multiple instance classification via successive linear programming. Journal of Optimization Theory and Applications 137(3):555–568

[13] For the sigmoid model, the integration over f_* cannot be done analytically, and one needs to do further approximation.

Maron O, Lozano-Perez T (1998) A framework for multiple-instance learning. Advances in Neural Information Processing Systems

Rahmani R, Goldman SA (2006) MISSL: Multiple-instance semi-supervised learning. International Conference on Machine Learning

Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. The MIT Press

Ray S, Page D (2001) Multiple instance regression. International Conference on Machine Learning

Raykar VC, Krishnapuram B, Bi J, Dundar M, Rao RB (2008) Bayesian multiple instance learning: Automatic feature selection and inductive transfer. International Conference on Machine Learning

Scott S, Zhang J, Brown J (2003) On generalized multiple instance learning. Technical Report UNL-CSE-2003-5, Department of Computer Science and Engineering, University of Nebraska, Lincoln, NE

Snelson E, Ghahramani Z (2006) Sparse Gaussian processes using pseudo-inputs. Advances in Neural Information Processing Systems

Tao Q, Scott S, Vinodchandran NV, Osugi TT (2004) SVM-based generalized multiple-instance learning via approximate box counting. International Conference on Machine Learning

Viola PA, Platt JC, Zhang C (2005) Multiple instance boosting for object detection. Advances in Neural Information Processing Systems

Wang H, Huang H, Kamangar F, Nie F, Ding C (2011) A maximum margin multi-instance learning framework for image categorization. Advances in Neural Information Processing Systems

Wang J, Zucker JD (2000) Solving the multiple-instance problem: A lazy learning approach. International Conference on Machine Learning

Weidmann N, Frank E, Pfahringer B (2003) A two-level learning method for generalized multi-instance problem. Lecture Notes in Artificial Intelligence 2837

Zhang D, Wang F, Luo S, Li T (2011) Maximum margin multiple instance clustering with its applications to image and text clustering. IEEE Transactions on Neural Networks 22(5):739–751

Zhang H, Fritts J (2005) Improved hierarchical segmentation. Technical Report, Washington University in St Louis

Zhang ML, Zhou ZH (2009) Multi-instance clustering with applications to multi-instance prediction. Applied Intelligence 31(1):47–68

Zhang Q, Goldman SA, Yu W, Fritts JE (2002) Content-based image retrieval using multiple-instance learning. International Conference on Machine Learning

Zhang T, Oles FJ (2000) Text categorization based on regularized linear classification methods. Information Retrieval 4:5–31