Mach Learn (2012) 88:273–295
DOI 10.1007/s10994-012-5283-x

Bayesian multi-instance multi-label learning using Gaussian process prior

Jianjun He · Hong Gu · Zhelong Wang

Received: 27 September 2010 / Accepted: 21 February 2012 / Published online: 10 March 2012

Abstract Multi-instance multi-label learning (MIML) is a newly proposed framework, in which the multi-label problems are investigated by representing each sample with multiple feature vectors named instances. In this framework, the multi-label learning task becomes to learn a many-to-many relationship, and it also offers a possibility for explaining why a concerned sample has the certain class labels. The connections between instances and labels as well as the correlations among labels are equally crucial information for MIML. However, the existing MIML algorithms can rarely exploit them simultaneously. In this paper, a new MIML algorithm is proposed based on Gaussian process. The basic idea is to suppose a latent function with Gaussian process prior in the instance space for each label and infer the predictive probability of labels by integrating over uncertainties in these functions using the Bayesian approach, so that the connection between instances and every label can be exploited by defining a likelihood function and the correlations among labels can be identified by the covariance matrix of the latent functions. Moreover, since different relationships between instances and labels can be captured by defining different likelihood functions, the algorithm may be used to deal with the problems with various multi-instance assumptions. Experimental results on several benchmark data sets show that the proposed algorithm is valid and can achieve superior performance to the existing ones.

Keywords Multi-label learning · Gaussian process · Multi-instance multi-label learning · Laplace approximation

Editors: Grigorios Tsoumakas, Min-Ling Zhang, and Zhi-Hua Zhou.

This work was supported in part by China Earthquake Research Funds (No. 200808075), and in part by Liaoning Provincial Natural Science Foundation of China (No. 20102025).

J. He · H. Gu (✉) · Z. Wang
Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, Liaoning, 116024, China
e-mail: guhong@dlut.edu.cn

J. He
e-mail: jianjunhe@live.com

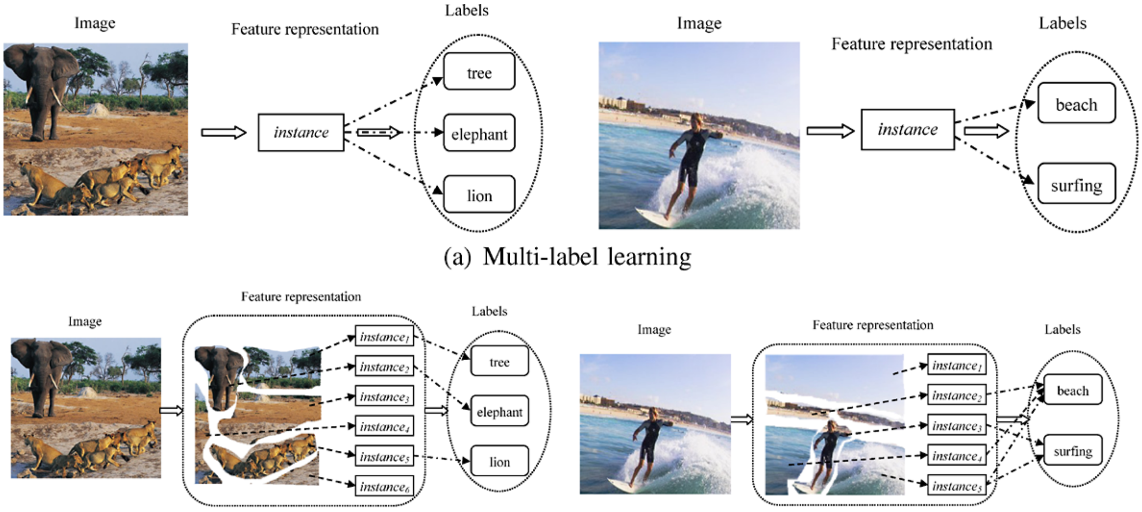
Z. Wang
e-mail: wangzl@dlut.edu.cn

1 Introduction

Nowadays, multi-label learning (ML) problems have attracted more and more attention in machine learning field due to their extensive applications in text categorization (McCallum 1999; Kazawa et al. 2005), scene classification (Boutell et al. 2004), functional genomics (Barutcuoglu et al. 2006), music categorization (Wieczorkowska et al. 2006), and so on. As shown by the examples in Fig. 1(a), multi-label learning studies the problems where a real-world object is associated with a number of class labels, i.e., the object has different semantic meanings simultaneously if it is viewed from different aspects. A straightforward approach solving multi-label problem is to transform it into one or more single-label problems. This can be realized by regarding every possible combination of labels as a 'meta-label' (Boutell et al. 2004; Diplaris et al. 2005) or considering the prediction of each label as an independent binary classification problem (Yang 1999). However, the first strategy is infeasible in most cases because the number of meta-labels will increase substantially and the samples of such meta-label are usually sparse; the second one has limitations as well because it neglects the correlations among the labels. Another approach is to consider the problem as a one-to-many mapping and design special learning algorithms for it. In Elisseeff and Weston (2002), a multi-label support vector machine was developed by defining a specific cost function and the corresponding margin. Some other multi-label support vector machines also were developed by Boutell et al. (2004) and Godbole and Sarawagi (2004). Zhang and Zhou (2006) employed neural networks for multi-label learning by defining a new error function to capture the characteristics of multi-label problem. Many other multi-label learning algorithms were developed as well, such as multi-label version of C4.5 decision tree (Clare and King 2001), multi-label k-nearest neighbor classifier (Zhang and Zhou 2007a), parametric mixture model (Ueda and Saito 2003), and boosting (Schapire and Singer 2000). Noting that although these studies on multi-label learning assume that an instance can be associated with multiple valid labels, one-to-many mapping is not a proper mathematical function and it may be the major difficulty in dealing with multi-label problems (Zhou et al. 2012). Moreover, in many ML problems, different labels are often tied to the different parts of the object, thus, developing classifiers based on the whole object would incur too much noise and harm the performance.

Recently, a multi-instance multi-label learning (MIML) framework was proposed by Zhou et al. (2007, 2012) for learning with ambiguous objects, in which an object is described by multiple feature vectors named instances and associated with multiple class labels. Compared with the traditional multi-label learning framework, MIML is more reasonable for dealing with the ML problems since it enables us to explore the inner causality between the object and its labels. In other words, it offers a possibility for understanding why a concerned object has the certain class labels, e.g., the object on the left part of Fig. 1(b) has label 'tree' because it contains *instance*$_1$, while label 'elephant' is caused by *instance*$_2$. The following two crucial problems should be considered for dealing with ML problems by using the MIML framework.

The first one is how to model the relationships between instances and labels of a sample. Although the MIML framework offers a possibility for understanding why a concerned object has the certain labels and a correct model of these connections may be helpful for making an accurate prediction, because different labels of an object usually arise from different instances and the way how the instances trigger labels of object usually may be different in different problems, for example, each label of the object on the right part of Fig. 1(b) is collectively determined by multiple instances, in contrast, every label of the object on the left part is only tied to a key instance, how to correctly model the connections between instances and labels is a crucial and challenging problem for MIML.



(a) Multi-label learning

(b) Multi-instance multi-label learning

Fig. 1 Examples of multi-label learning and multi-instance multi-label learning

The second one is how to exploit the relationship among labels. For many multi-label learning problems, different labels usually have strong correlations which also are the important information for improving the accuracy of prediction algorithms. For example, in the place recognition problem, the places with near locations usually appear in the same visual image and in contrast the places located at different areas scarcely appear in the same image. So, if an image is simultaneously labeled as two places located at different areas, we should reconsider this conclusion. Taking the subcellular localization prediction problem of eukaryotic proteins (Chou and Shen 2010) as another example, it can be seen that almost all the proteins of cyanelle and hydrogenosome have only one location. If a predictor can obtain this information from the training data set, it will avoid some wrong prediction results such as "a query protein belongs to cyanelle and hydrogenosome simultaneously". Moreover, for the problems with a large number of labels and small amount of training samples, the correlation information among labels will be very important to complement the lack of training samples. Thus, effective exploitation of the correlations among labels is also crucial for the success of a MIML algorithm.

Although many MIML algorithms have been proposed by employing neural network, maximum margin method, regularization method and so on, almost all of them pay attention to only one of aforementioned problems and few can consider them simultaneously. As shown by Zhou et al. (2012), the main reason may be that these models will become very complex and difficult to be solved if these two problems are considered simultaneously. In order to consider the two aforementioned problems simultaneously, an innovative MIML algorithm is proposed by using Gaussian process in this paper. The basic idea is to define a latent function with Gaussian process prior in the instance space for every label, and then output the probabilities over different labels for each sample based on the latent function values of its instances. In this algorithm, the correlations among the labels can be identified by a covariance matrix of these latent functions and automatically inferred by maximizing the marginal likelihood of the covariance matrix; the connections between instances and labels of a sample can be exploited by defining a new likelihood function and we can employ different likelihood functions to capture various connections.

The paper is organized as follows: In Sect. 2, we illustrate the formal definition of MIML and review previous work in this area. The proposed MIML algorithm is then presented in Sect. 3 and tested on several multi-label learning problems in Sect. 4. The conclusion is given in Sect. 5.

2 Related works

As stated in the above section, a sample is described by multiple instances and associated with multiple class labels in the MIML framework. Formally, suppose $\mathcal{X} = \mathbb{R}^d$ is the domain of instances and $\mathcal{Y} = \{1, 2, \dots, Q\}$ is the set of class labels. Given a training set $S = \{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$, where $X_i \subseteq \mathcal{X}$ called bag is a set of instances $\{x_{ij} | x_{ij} \in \mathcal{X}, j = 1, 2, \dots, n_i\}$, and $Y_i \subseteq \mathcal{Y}$ is the label set $\{y_{ik} | y_{ik} \in \mathcal{Y}, k = 1, 2, \dots, l_i\}$ associated with X_i, here, n_i denotes the number of instances in X_i and l_i denotes the number of labels in Y_i. For notational convenience, Y_i usually is represented by a vector $[y_{i1}, y_{i2}, \dots, y_{iQ}]^{\mathrm{T}}$, in which $y_{is} = 1$ denotes that label s is a proper label of bag X_i, otherwise $y_{is} = -1$. The task of MIML is to learn a function $h : 2^{\mathcal{X}} \to 2^{\mathcal{Y}}$ from S which can predict a set of labels for any unseen sample. It is interesting to compare MIML with multi-instance learning framework and multi-label learning framework. Multi-label learning considers the ambiguity in the label space; Multi-instance learning considers the ambiguity in the instance space; While MIML studies the ambiguities in both the instance and label space simultaneously. It can be seen that MIML is also different from the framework of learning from candidate labeling sets (Cour et al. 2009; Jie and Orabona 2010) where each sample is supplied with multiple potential labels, only one of them is correct. Since many real objects are inherited with input ambiguity as well as output ambiguity, MIML is more natural and convenient for learning with such objects.

The generality of MIML inevitably makes it much difficult to address and only a few literatures are available up to present. MIMLBOOST and MIMLSVM are the earliest MIML algorithms, which were proposed by Zhou and Zhang (2007) for scene classification based on a simple degeneration strategy. In Nguyen (2010), a new SVM approach named SISL-MIML was developed by using an improved degeneration strategy. Considering that the degeneration methods may lose useful information encoded in the training data during the reduction process, another algorithm called M3MIML was proposed by Zhang and Zhou (2008) based on a maximum margin method, which directly exploits the connections between instances and labels. In Zhang and Wang (2009), RBF neural network was adopted to learn from MIML samples. A probabilistic generative model called Dirichlet-Bernoulli alignment (DBA) was proposed by Yang et al. (2009) for MIML. Since the performance of DBA may deteriorate exponentially as the number of classes increases and can not be used to the task with a large number of classes, in 2010, they improved DBA to a hybrid generative/discriminative learning model (Yang et al. 2010) for the problem of automatic image annotation. A common characteristic of these approaches is that they rarely consider the correlations among the labels. In Zhou et al. (2012), a D-MIMLSVM algorithm was presented, which tackles MIML problems directly in a regularization framework. The algorithm defines an objective function which balances the loss between the labels and predictions on the bags as well as on the constituent instances. It also considers the correlations among the labels associated to the same sample. Unfortunately, as shown in the discussion section of their paper, the algorithm is established under the assumption that all the class labels share the same commonness, which over-simplifies the real scenario. In fact, in real applications it is rare that all class labels share the same commonness; it is more typical that some labels share commonness, but the commonness shared by different label subsets may be different. For example, label ‘surfing’ may share commonness ‘water’ with label ‘beach’, and label ‘elephant’ may share commonness ‘animal’ with label ‘lion’, but maybe ‘surfing’, ‘beach’, ‘elephant’ and ‘lion’ share nothing together. Although they also extend the model to exploit the problem that different pairs of labels share different commonness, the new model is difficult to be solved since it involves too many variables. In addition to the design of various MIML algorithms, some other progresses on MIML, such as theoretical exploration to the learnability of MIML (Wang and Zhou 2012), metric learning from MIML data (Jin et al. 2009), multi-label learning by instance differentiation (Zhang and Zhou 2007b) as well as applications of MIML in bioinformatics (Li et al. 2012), image classification (Zhou and Zhang 2007; Zha et al. 2008), visual mobile robot navigation (He et al. 2012), have been made.

Although the existing MIML algorithms may achieve better performance than the traditional multi-label learning algorithms, most of them only focus on the correspondence between instances and labels, and the correlations among the labels are rarely considered. In the next section, Gaussian process is used to establish a new MIML algorithm which exploits not only the connections between instances and labels but also the correlations among labels.

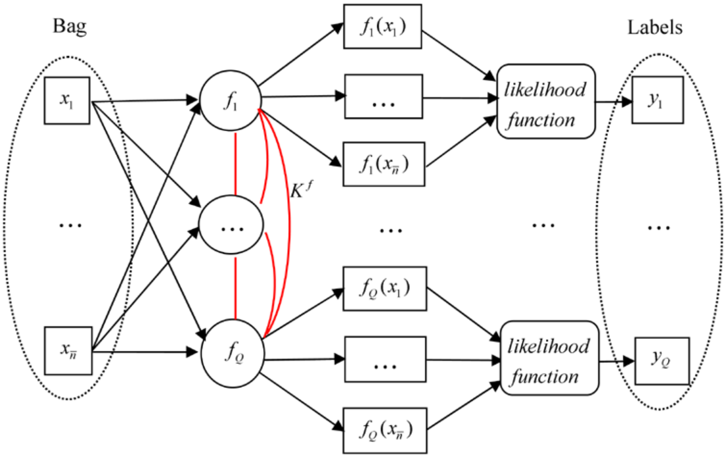


Fig. 2 A graphic representation of the proposed MIML algorithm

3 The proposed algorithm

Gaussian process has been widely used for traditional supervised learning (Williams and Rasmussen 1996; Lawrence and Platt 2004; Williams and Barber 1998; He et al. 2011), a detailed introduction about this area can be found in Rasmussen and Williams (2006). Explicit probabilistic formulation is an important advantage of Gaussian process models over other non-Bayesian models. This also provides the ability to infer model parameters such as those control the kernel shape and the noise level. In this section, a probabilistic kernel algorithm is established by using Gaussian process for MIML. To represent our uncertainty over class labels for a sample $X = \{x_1, x_2, \dots, x_{\bar{n}}\}$, a better method is to output a probability for each label. As shown in Fig. 2, the main idea of the proposed model is to assume an unobservable latent function f_s for every label y_s on the instance space $\mathbb{R}^d$, $s = 1, 2, \dots, Q$, and then the probability that sample X belongs to label y_s can be determined by the values $\{f_s(x_1), f_s(x_2), \dots, f_s(x_{\bar{n}})\}$ of function f_s on the bag $\{x_1, x_2, \dots, x_{\bar{n}}\}$. In this model, the correlations among the labels are identified by a covariance matrix K^f between the latent functions $\{f_1, f_2, \dots, f_Q\}$, it will be seen that K^f can be obtained by maximizing a marginal likelihood; the connection between the instances and each label can be exploited by defining a likelihood function $p(y_s | f_s(x_1), f_s(x_2), \dots, f_s(x_{\bar{n}}))$, for example, we can define the likelihood as $p(y_s = 1 | f_s(x_1), f_s(x_2), \dots, f_s(x_{\bar{n}})) = \max_j p(y_s = 1 | f_s(x_j))$ for the standard multi-instance assumption which states that the label of a bag is positive if and only if it contains at least one positive instance. Noting that although the proposed model also consists of three layers from input to output with one hidden layer just like the MIML-RBF (Zhang and Wang 2009) and DBA model (Yang et al. 2009), they may be very different because our model also introduces a direct relationship of the hidden variables. The details of the proposed algorithm are shown as follows.

3.1 Gaussian process prior

The basic idea behind Gaussian process prediction is to place a Gaussian process prior over the latent functions. Inspired by Bonilla et al. (2008), in which the same GP prior is placed for a multi-task regression problem, we approach MIML problem by placing a GP prior with zero mean over the latent functions $\{f_s|s = 1, 2, \dots, Q\}$, i.e.,

$$\langle f_l(x), f_s(x')\rangle = K^f_{ls} \cdot k(x, x'), \quad l, s = 1, 2, \dots, Q \tag{1}$$

where $K^f = (K^f_{ls})_{Q\times Q}$ is a positive semi-definite matrix that specifies the correlations among labels, so that the observation of one class can affect the prediction on another class. An important property of this model is that the correlations among labels can be deduced directly by a covariance matrix K^f. $k(x, x')$ is a covariance function over instances x and x'. Two covariance functions are used in this study, i.e.,

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{\delta^2}} \tag{2}$$

$$k(x, x') = x \cdot (x')^{\mathrm{T}} \tag{3}$$

We assume that all the parameters have been given except matrix K^f. For notational convenience, let $Y = [y_{11}, \dots, y_{n1}, \dots, y_{1s}, \dots, y_{ns}, \dots, y_{1Q}, \dots, y_{nQ}]^{\mathrm{T}}$, $D = \{x_{ij}|i = 1, \dots, n, j = 1, \dots, n_i\}$ be the set including all the instances in training set, $f_{ijs} = f_s(x_{ij})$ be shorthand for the value of the latent function, $F_{is} = [f_{i1s}, f_{i2s}, \dots, f_{in_is}]^{\mathrm{T}}$ be the values of latent function f_s corresponding to bag X_i, and $F = [F^{\mathrm{T}}_{11}, \dots, F^{\mathrm{T}}_{n1}, \dots, F^{\mathrm{T}}_{1s}, \dots, F^{\mathrm{T}}_{ns}, \dots, F^{\mathrm{T}}_{1Q}, \dots, F^{\mathrm{T}}_{nQ}]^{\mathrm{T}}$. So, the joint distribution

$$p(F|D, K^f) = \mathcal{N}(F|\mathbf{0}, K^f \otimes K) \tag{4}$$

is a Gaussian distribution with zero mean and covariance matrix $K^f \otimes K$, where $\otimes$ denotes the Kronecker product, the element of K is $k(x, x')$, $x, x' \in D$. The joint prior over F and $F_* = [F^{\mathrm{T}}_{*1}, F^{\mathrm{T}}_{*2}, \dots, F^{\mathrm{T}}_{*Q}]^{\mathrm{T}}$ is

$$p(F_*, F|D, X_*, K^f) = \mathcal{N}\left(\begin{bmatrix} F_* \\ F \end{bmatrix}\Big|\mathbf{0}, \begin{bmatrix} K^f \otimes K_{**} & K^f \otimes K^T_* \\ K^f \otimes K_* & K^f \otimes K \end{bmatrix}\right) \tag{5}$$

where, $F_{*s} = [f_{*1s}, f_{*2s}, \dots, f_{*n_*s}]^{\mathrm{T}}$, $f_{*js} = f_s(x_{*j})$, the elements of K_{**} are $k(x_{*i}, x_{*j})$, the elements of K_* are $k(x, x')$, $x \in D$, $x' \in X_*$. Thus, the conditional prior

$$p(F_*|F, D, X_*, K^f) = \mathcal{N}\left(F_*|(I^f \otimes K^{\mathrm{T}}_* K^{-1})F, K^f \otimes (K_{**} - K^{\mathrm{T}}_* K^{-1} K_*)\right) \tag{6}$$

may be deduced analytically, where I^f is an identity matrix with the same order as K^f.

3.2 Joint likelihood

The joint likelihood, denoted as $p(Y|F)$, is the joint probability of observing class labels given the latent functions. Generally, the class labels are assumed to be independent variables given the latent functions. Thus, the joint likelihood $p(Y|F)$ can be evaluated as a product of the likelihoods on individual observation, that is

$$p(Y|F) = \prod_{i=1}^{n}\prod_{s=1}^{Q} p(y_{is}|F_{is}) \tag{7}$$

Because the way in which instances trigger labels may be different in different problems, in different learning tasks, it is difficult to know which assumption is the fittest. Just as the different assumptions on the relationship between instance-labels and bag-labels in multi-instance learning (Maron and Lozano-Pérez 1998; Zhou 2004), different likelihood functions can be defined to deal with various problems. In this paper, we will first present the algorithm under the collective multi-instance assumption, and then modify it to obey the standard multi-instance assumption. Note that a more detailed description about different multi-instance assumptions can be found in the work of Foulds and Frank (2010).

The collective assumption states that all the instances contribute equally and independently to a bag's label. Thus, the likelihood $p(y_{is}=1|F_{is})$ can be intuitively defined as

$$p(y_{is}=1|F_{is})=sig\left(\frac{1}{n_i}\sum_{j=1}^{n_i}f_{ijs}\right) \tag{8}$$

where, $sig(\cdot)$ is a sigmoid function and logistic function $\sigma(t)=\frac{1}{1+e^{-t}}$ is used in this study. As the probability of the two classes must sum to 1, we have $p(y_{is}=-1|F_{is})=1-p(y_{is}=1|F_{is})$. So, for logistic function, likelihood $p(y_{is}|F_{is})$ can be written as

$$p(y_{is}|F_{is})=\sigma\left(\frac{y_{is}}{n_i}\sum_{j=1}^{n_i}f_{ijs}\right) \tag{9}$$

by using the property $\sigma(-t)=1-\sigma(t)$.

3.3 Posterior distribution

By using Bayes's rule, the posterior distribution over F for a given K^f becomes

$$p\big(F|D,Y,K^f\big)=\frac{p(Y|F)p(F|D,K^f)}{p(Y|D,K^f)} \tag{10}$$

where,

$$p\big(Y|D,K^f\big)=\int p(Y|F)p\big(F|D,K^f\big)dF \tag{11}$$

denotes the marginal likelihood for the parameter matrix K^f.

Note that the posterior $p(F|D,Y,K^f)$ is a non-Gaussian distribution and can not be computed analytically. In the traditional Gaussian process classification algorithms, a popular idea is to approximate the posterior by using a tractable Gaussian distribution, and many approaches such as Laplace approximation (LA), expectation propagation (EP) and Kullback-Leibler divergence minimization (KL) have been proposed. We can use any of these approaches to approximate the posterior $p(F|D,Y,K^f)$ of this paper. The major difference among them may be that the computational complexity and the performance are different. A more comprehensive overview of these algorithms can be found in Nickisch and Rasmussen (2008). Because it may spend more shorter computing time than others, in this paper, the Laplace's method is utilized to obtain a Gaussian approximation

$$q\big(F|D,Y,K^f\big)=\mathcal{N}\big(F|\hat{F},A^{-1}\big) \tag{12}$$

of $p(F|D,Y,K^f)$, where $\hat{F}=\arg\max_F p(F|D,Y,K^f)$, $A=-\nabla\nabla\log p(F|D,Y,K^f)|_{F=\hat{F}}$ is the Hessian matrix of the negative log posterior distribution at $\hat{F}$.

It can be seen that the marginal likelihood $p(Y|D,K^f)$ is independent of F, so we only need to consider the un-normalized posterior when maximizing (10) with regard to F. By taking the logarithm and introducing (4), (7), we obtain

$$\begin{aligned}\psi(F)&\triangleq\log p(Y|F)+\log p\big(F|D,K^f\big)\\&=\log p(Y|F)-\frac{1}{2}F^{\mathrm{T}}\big(K^f\otimes K\big)^{-1}F-\frac{1}{2}\log\big|K^f\otimes K\big|-\frac{Q}{2}\log 2\pi\left(\sum_{i=1}^{n}n_i\right)\end{aligned} \tag{13}$$

Then, by differentiating (13) with regard to F, we obtain

$$\nabla\psi(F) = \nabla \log p(Y|F) - \big(K^f \otimes K\big)^{-1} F \tag{14}$$

$$\nabla\nabla\psi(F) = \nabla\nabla \log p(Y|F) - \big(K^f \otimes K\big)^{-1} \tag{15}$$

Taking $\frac{\partial^2 \log p(Y|F)}{\partial f_{ijs}\partial f_{ils}} = -\exp(\frac{y_{is}}{n_i}\sum_{j=1}^{n_i} f_{ijs})/(n_i + n_i \exp(\frac{y_{is}}{n_i}\sum_{j=1}^{n_i} f_{ijs}))^2$ and $\frac{\partial \log p(Y|F)}{\partial f_{ijs}} = y_{is}/(n_i + n_i \exp(\frac{y_{is}}{n_i}\sum_{j=1}^{n_i} f_{ijs}))$ into $\nabla\nabla \log p(Y|F)$ and $\nabla \log p(Y|F)$ respectively, we obtain,

$$\nabla \log p(Y|F) = \big(I^f \otimes \operatorname{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\}\big) d \tag{16}$$

$$\begin{aligned} W &\triangleq -\nabla\nabla \log p(Y|F) \\ &= \big(I^f \otimes \operatorname{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\}\big) W_0 \big(I^f \otimes \operatorname{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\}\big)^{\mathrm{T}} \end{aligned} \tag{17}$$

where, $d = [d_{11}, d_{21}, \ldots, d_{n1}, \ldots, d_{1Q}, d_{2Q}, \ldots, d_{nQ}]^{\mathrm{T}}$, $d_{is} = y_{is}/(n_i + n_i \exp(\frac{y_{is}}{n_i}\sum_{j=1}^{n_i} f_{ijs}))$, $W_0 = \operatorname{diag}\{w_{11}, w_{21}, \ldots, w_{n1}, \ldots, w_{1Q}, w_{2Q}, \ldots, w_{nQ}\}$, $w_{is} = \exp(\frac{y_{is}}{n_i}\sum_{j=1}^{n_i} f_{ijs})/(n_i + n_i \exp(\frac{y_{is}}{n_i}\sum_{j=1}^{n_i} f_{ijs}))^2$, $\mathbf{1_{n_i}}$ is an n_i dimensional column vector of ones, $\operatorname{diag}\{A_1, A_2, \ldots, A_n\}$ denotes a block matrix in which the ith main diagonal block is A_i and the off-diagonal blocks are zero matrices, $\exp(\cdot)$ is the exponential function. It can be seen from the negative definite Hessian matrix in (15) that $\psi(F)$ is concave and the equation $\nabla\psi(F) = 0$ has a unique solution $\hat{F}$. Thus, (18) can be obtained by using (14)

$$\hat{F} = \big(K^f \otimes K\big) \nabla \log p(Y|\hat{F}) \tag{18}$$

Since (18) is a non-linear equation and can not be solved analytically, Newton's method is used to solve it in this paper. And the iterative formula is

$$F_{new} = F - (\nabla\nabla\psi)^{-1}\nabla\psi = \big(W + \big(K^f \otimes K\big)^{-1}\big)^{-1}\big(\nabla \log p(Y|F) + WF\big) \tag{19}$$

By using matrix inversion formula,

$$\big(Z + UPV^{\mathrm{T}}\big)^{-1} = Z^{-1} - Z^{-1}U\big(P^{-1} + V^{\mathrm{T}}Z^{-1}U\big)^{-1}V^{\mathrm{T}}Z^{-1} \tag{20}$$

formula (19) can be expressed as

$$\begin{aligned} F_{new} = {} & \big(K^f \otimes \big(K \operatorname{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\}\big)\big)\big(I - W_0^{1/2} B^{-1} W_0^{1/2} \big(K^f \otimes K_{sum}\big)\big) \\ & \times \big(W_0 \big(I^f \otimes \operatorname{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\}\big)^{\mathrm{T}} F + d\big) \end{aligned} \tag{21}$$

where, $K_{sum} = (\operatorname{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\})^{\mathrm{T}} K (\operatorname{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\})$, I is an identity matrix, $B = I + W_0^{1/2}(K^f \otimes K_{sum})W_0^{1/2}$.

At last, the matrix A may be obtained by substituting $\hat{F}$ into the negative Hessian matrix in (15),

$$A = -\nabla\nabla\psi(F)|_{F=\hat{F}} = \big(W + \big(K^f \otimes K\big)^{-1}\big)\big|_{F=\hat{F}} \tag{22}$$

By using Gaussian approximation (12) of posterior $p(F|D, Y, K^f)$, the distribution of the latent variables F_* can be simplified as a Gaussian distribution:

$$\begin{aligned} p\big(F_*|D, Y, X_*, K^f\big) &\simeq \int p\big(F_*|F, D, X_*, K^f\big) q\big(F|D, Y, K^f\big) dF \\ &= \mathcal{N}\big(F_*|\overline{K}\hat{F}, K^f \otimes \big(K_{**} - K_*^{\mathrm{T}} K^{-1} K_*\big) + \overline{K} A^{-1} \overline{K}^{\mathrm{T}}\big) \end{aligned} \tag{23}$$

where, $\overline{K} = I^f \otimes K_*^{\mathrm{T}} K^{-1}$. And then by utilizing (18) and (20), formula (23) can be transformed into another form:

$$p\big(F_*|D, Y, X_*, K^f\big) = \mathcal{N}\big(F_*|\overline{K}_*\hat{d}, K^f \otimes K_{**} - \overline{K}_*\hat{W}_0^{1/2}\hat{B}^{-1}\hat{W}_0^{1/2}\overline{K}_*^{\mathrm{T}}\big) \tag{24}$$

where, $\overline{K}_* = K^f \otimes (K_*^{\mathrm{T}}\mathrm{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \dots, \mathbf{1}_{n_n}\})$, $\hat{d} = d|_{F=\hat{F}}$, $\hat{W}_0 = W_0|_{F=\hat{F}}$, $\hat{B} = B|_{F=\hat{F}}$.

3.4 Prediction

Let $p(y_{*s} = 1|D, Y, X_*, K^f)$ denote the probability that label s is a proper label of bag X_*, $s = 1, 2, \dots, Q$, it can be predicted by averaging out F_{*s}, i.e.,

$$p\big(y_{*s} = 1|D, Y, X_*, K^f\big) = \int \sigma\left(\frac{1}{n_*}\sum_{j=1}^{n_*} f_{*js}\right) p\big(F_{*s}|D, Y, X_*, K^f\big) dF_{*s} \tag{25}$$

Because of the non-linear form of the logistic function $\sigma(\cdot)$, the predictive probability (25) can not be computed analytically. Thus, we need to resort to sampling methods or analytical approximations to compute these integrals. In this paper, Monte Carlo sampling method is used to compute (25).

3.5 Learning parameter matrix K^f

The Bayesian framework described above is conditional on the kernel parameter matrix K^f. In this section, we wish to learn the parameter matrix K^f by maximizing the marginal likelihood in (11), in other words, maximizing the agreement between observed data and the model. Since the integral is intractable, one way to achieve this is to provide a lower bound of the marginal likelihood (11). A popular lower bound is the one obtained by Seeger (2003) utilizing Jensen's inequality and a Gaussian approximation of the posterior $p(F|D, Y, K^f)$. That is,

$$\begin{aligned} \log p\big(Y|D, K^f\big) \geq & \int q\big(F|D, Y, K^f\big) \log p(Y|F) dF \\ & + \int q\big(F|D, Y, K^f\big) \log p\big(F|D, K^f\big) dF \\ & - \int q\big(F|D, Y, K^f\big) \log q\big(F|D, Y, K^f\big) dF \\ =: & \log Z \end{aligned} \tag{26}$$

Therefore, the parameter matrix K^f can be obtained by maximizing $\log Z$. Because the parameters $\hat{F}$ and A of $q(F|D, Y, K^f)$ are also the non-linear functions of K^f, it is difficult to maximize $\log Z$ directly. Inspired by Kim and Ghahramani (2006), an EM-like algorithm is used to solve this problem. The algorithm is divided into two steps: in the E-step, we compute the values of $\hat{F}$ and A by using (21) and (22) given the parameter matrix K^f, and in the M-step, the lower bound $\log Z$ is maximized with regard to K^f given the values of $\hat{F}$ and A obtained in the E-step. The E-step and M-step are alternated until convergence. Since the first and third terms of $\log Z$ are independent of the parameter matrix K^f given the values of $\hat{F}$ and A, we only need to maximize $\int q(F|D, Y, K^f) \log p(F|D, K^f) dF$ in the M-step. By expanding it, we obtain

$$\begin{aligned}
&\int q\big(F|D,Y,K^f\big)\log p\big(F|D,K^f\big)dF\\
&\quad=-\frac{1}{2}\Bigg(\hat{F}^{\mathrm{T}}\big(K^f\otimes K\big)^{-1}\hat{F}+Q\log 2\pi\sum_{i=1}^{n}n_i+Q\log|K|\\
&\qquad+\sum_{i=1}^{n}n_i\log\big|K^f\big|+\mathrm{tr}\big(A^{-1}\big(K^f\otimes K\big)^{-1}\big)\Bigg)\\
&\quad=: G\big(K^f|\hat{F},A\big)
\end{aligned}\tag{27}$$

For distinguishing different K^f, we denote the K^f related to $\hat{F}$ and A as K^f_{old}, i.e., $\hat{F}=(K^f_{old}\otimes K)\nabla\log p(Y|\hat{F})$, $A=(W+(K^f_{old}\otimes K)^{-1})|_{F=\hat{F}}$. Then, by using matrix inversion formula (20), $G(K^f|\hat{F},A)$ can be expressed as

$$\begin{aligned}
G\big(K^f|\hat{F},A\big)=-\frac{1}{2}\Bigg(&Q\log|K|+\Bigg(\sum_{i=1}^{n}n_i\Bigg)\big(Q\log 2\pi+\log\big|K^f\big|+\mathrm{tr}\big(K^f_{old}\big(K^f\big)^{-1}\big)\big)\\
&-\mathrm{tr}\big(\big(\hat{W}_0^{1/2}\hat{B}^{-1}\hat{W}_0^{1/2}-\hat{d}\hat{d}^{\mathrm{T}}\big)\big(\big(K^f_{old}\big(K^f\big)^{-1}K^f_{old}\big)\otimes K_{sum}\big)\big)\Bigg)
\end{aligned}\tag{28}$$

where tr$(\cdot)$ denotes the trace of a matrix. Differentiating (28) with regard to K^f,

$$\begin{aligned}
&\nabla G\big(K^f|\hat{F},A\big)\\
&\quad=-\frac{1}{2}\Bigg(\Bigg(\sum_{i=1}^{n}n_i\Bigg)\big(K^f\big)^{-1}-\Bigg(\sum_{i=1}^{n}n_i\Bigg)\big(K^f\big)^{-1}K^f_{old}\big(K^f\big)^{-1}\\
&\qquad+\big(K^f\big)^{-1}K^f_{old}HK^f_{old}\big(K^f\big)^{-1}\Bigg)
\end{aligned}\tag{29}$$

where H is a square matrix of order Q. The element H_{ij} of H can be computed by

$$H_{ij}=\mathrm{tr}(C_{ij}K_{sum})\tag{30}$$

where, $C_{ij}, i, j=1,\dots,Q$ are square matrixes of order n by which $\hat{W}_0^{1/2}\hat{B}^{-1}\hat{W}_0^{1/2}-\hat{d}\hat{d}^{\mathrm{T}}$ is expressed with block , that is

$$\begin{pmatrix}C_{11}&C_{12}&\cdots&C_{1Q}\\C_{21}&C_{22}&\cdots&C_{2Q}\\\vdots&\vdots&\ddots&\vdots\\C_{Q1}&C_{Q2}&\cdots&C_{QQ}\end{pmatrix}=\hat{W}_0^{1/2}\hat{B}^{-1}\hat{W}_0^{1/2}-\hat{d}\hat{d}^{\mathrm{T}}\tag{31}$$

By setting $\nabla G(K^f|\hat{F},A)$ to zero, we have

$$K^f=K^f_{old}-K^f_{old}HK^f_{old}/\Bigg(\sum_{i=1}^{n}n_i\Bigg)\tag{32}$$

Thus, in the M-step, we only need update K^f with (32). Till now, the whole MIML algorithm has been presented. Note that a significant problem with Gaussian process prediction is that it needs to invert the matrix $B=I+W_0^{1/2}(K^f\otimes K_{sum})W_0^{1/2}$, which is prohibitive on modern workstations for large problems. We will deal with this problem in the next section.

3.6 Approximation of matrix K

The issue of dealing with large data set has been studied in many Gaussian process literatures, and a more detailed overview can be found in Rasmussen and Williams (2006). In this paper, we reduce the computational complexity of inverting the $Qn \times Qn$ matrix B by approximating K_{sum} in the form $K_{sum} \approx PP^{\mathrm{T}}$, here, P is an $n \times m$ matrix, $m \ll n$. Notice that, by representing K_{sum} with P, $W_0^{1/2}B^{-1}W_0^{1/2}$ can be expressed as

$$W_0^{1/2}B^{-1}W_0^{1/2} = W_0 - W_0(L \otimes P)B_1^{-1}\big(L^{\mathrm{T}} \otimes P^{\mathrm{T}}\big)W_0 \tag{33}$$

where, $B_1 = I + (L^{\mathrm{T}} \otimes P^{\mathrm{T}})W_0(L \otimes P)$, $K^f = LL^{\mathrm{T}}$. Thus, the problem is transformed into the inversion of a $Qm \times Qm$ matrix. For the problem with moderate n, we can consider reduced-rank approximations to K_{sum}. In this paper, the optimal reduced-rank approximation $U_m \varLambda_m U_m^{\mathrm{T}}$ of K_{sum} with respect to the Frobenius norm is used, where $\varLambda_m$ is the diagonal matrix of the leading m eigenvalues of K_{sum} and U_m is the matrix of the corresponding eigenvectors. Thus, $P = U_m \varLambda_m^{1/2}$. Unfortunately, this is limited for the problem with large n because the eigendecomposition needs $O(n^3)$ operations. However, the Nyström method (Williams and Seeger 2001) can be used to compute an approximation of K_{sum}. This approximation is obtained by randomly choosing m rows/columns of K_{sum}, and then setting $P = K_{nm}K_{mm}^{-1/2}$, where K_{nm} is an $n \times m$ block of the original matrix K_{sum}. In the next section, we will give more details of the implementation of the proposed algorithm.

3.7 Implementation of the proposed algorithm

In this section, care is taken to minimize the computational cost and to avoid numerically unstable computations. Because the mean of F_{is}, not F_{is}, is used in the whole algorithm, we can rewrite the Newton iteration given in (21) as

$$F_{sum} = \big(K^f \otimes K_{sum}\big)\big(I - W_0^{1/2}B^{-1}W_0^{1/2}\big(K^f \otimes K_{sum}\big)\big)(W_0F_{sum} + d) \tag{34}$$

where, F_{sum} denotes the vector composed of the sum of F_{is}, i.e. $F_{sum} = I^f \otimes (\mathrm{diag}\{\mathbf{1}_{n_1}, \mathbf{1}_{n_2}, \ldots, \mathbf{1}_{n_n}\})^{\mathrm{T}}F$. By substituting (33) into (34), we obtain

$$F_{sum} = \big(K^f \otimes PP^{\mathrm{T}}\big)a_5 \tag{35}$$

where, $a_5 = a_1 - a_2 + W_0(L \otimes P)a_4, a_4 = B_1^{-1}a_3, a_3 = (L^{\mathrm{T}} \otimes P^{\mathrm{T}})a_2, a_2 = W_0(K^f \otimes PP^{\mathrm{T}})a_1, a_1 = W_0F_{sum} + d$. For minimizing the cost of computing B_1^{-1}, formula $a_4 = B_1^{-1}a_3$ is transformed into a minimum problem $a_4 = \arg\min_x \frac{1}{2}(B_1x - a_3)^{\mathrm{T}}(B_1x - a_3)$, which is solved by using conjugate gradient method. Similarly, we also can rewrite formula (24) and (31). Note that the main reason for using the symmetric positive definite matrix B_1 widely is that it is well-conditioned for many covariance functions. Moreover, because K_{sum}, not K, is used in the whole algorithm, we only need to store kernel matrix K_{sum} in the system. The flowchart of the algorithm is outlined in Algorithm 1. In this paper, the thresholds ε_0 and ε are set to be 0.01 and 2×10^{-6}, respectively; m (the number of columns of P) is determined by the cumulative percentage of $\varLambda_m$, and detailed discussions about m can be found in the second paragraph of Sect. 4.1.

It can be seen from the flowchart that the computational complexity for training the model is mainly dominated by solving the minimum problem in the step 3.2 and computing the inverse of B in the step 4. Thus, in the training stage, the computational complexity of the model is about $O(lQ^3nm^2)$, where l is the number of iterations of the EM-like algorithm for solving the model. Notice that the computational complexity does not depend on the instance number. This is because the algorithm only relates to the kernel matrix K_{sum} rather than K in the training stage. In the testing stage, the computational complexity is dominated by computing the term $\overline{K}_* \hat{W}_0^{1/2} \hat{B}^{-1} \hat{W}_0^{1/2} \overline{K}_*^{\mathrm{T}}$ in formula (24) which needs about $O(Q^2 n^2 \bar{n})$ operations, here, $\bar{n}$ denotes the average number of instances.

Algorithm 1 (the flowchart of proposed algorithm)

```
Input: $D, Y, K_{sum}, X_*, m, \varepsilon_0, \varepsilon$
1 Initializing $F_{sum}$ and $K^f$ as a vector of ones and an identical matrix, respectively;
2 Approximating $K_{sum}$ in the form $K_{sum} \approx PP^{\mathrm{T}}$;
3 E-step: given $K^f$, updating $F_{sum}$ by using formula (35);
  3.1 $a_1 = W_0 F_{sum} + d, a_2 = W_0(K^f \otimes PP^{\mathrm{T}})a_1, a_3 = (L^{\mathrm{T}} \otimes P^{\mathrm{T}})a_2$;
  3.2 $a_4 = \arg\min_x \frac{1}{2}(B_1 x - a_3)^{\mathrm{T}}(B_1 x - a_3)$;
  3.3 $a_5 = a_1 - a_2 + W_0(L \otimes P)a_4$, updating $F_{sum} = (K^f \otimes PP^{\mathrm{T}})a_5, obj = -\frac{1}{2}a_5^{\mathrm{T}} F_{sum} + \log p(Y|F_{sum})$;
  3.4 If the difference of $obj$ between two subloops is smaller than a threshold $\varepsilon_0$, then go to 4, else go to 3.1;
4 M-step: given $F_{sum}$, updating $K^f$ by using (32);
5 If the difference of $K^f$ between two loops is smaller than a threshold $\varepsilon$ (i.e., $\|K_1^f - K_2^f\|/\|K_1^f\| < \varepsilon$, where $K_1^f$ and $K_2^f$ denote the values of $K^f$ obtained in the two loops), then output $F_{sum}, K^f$, else go to 3;
6 Computing the mean and variance of $F_*|D, Y, X_*, K^f$ by using (24);
7 Predicting labels of $X_*$ with (25).
```

3.8 Extension to the standard multi-instance assumption

Through the efforts of above sections, we have shown the proposed algorithm under the collective multi-instance assumption. In this section, the algorithm will be modified to deal with the problems satisfying the standard multi-instance assumption. Being different with the collective assumption, the standard multi-instance assumption states that a bag is positive if and only if it contains at least one positive instance. Thus, for the standard assumption, the likelihood $p(y_{is} = 1|F_{is})$ (8) can be redefined as

$$p(y_{is} = 1|F_{is}) = \max_j p(y_{is} = 1|f_{ijs}) = \max_j \sigma(f_{ijs}) = \sigma\left(\max_j f_{ijs}\right) \tag{36}$$

and the likelihood function $p(y_{is}|F_{is})$ (9) can be written as

$$p(y_{is}|F_{is}) = \sigma\left(y_{is} \max_j f_{ijs}\right) \tag{37}$$

Since function $\max(\cdot)$ is non-differentiable, the aggregate function (also known as exponential penalty function) can be used as an approximation of it, i.e.,

$$\max_j f_{ijs} \approx \frac{1}{v} \log\left(\sum_{j=1}^{n_i} \exp(v f_{ijs})\right) \tag{38}$$

where, $v > 0$ is a control parameter. It can be seen that

$$\max_j f_{ijs} = \lim_{v \to \infty} \frac{1}{v} \log\left(\sum_{j=1}^{n_i} \exp(v f_{ijs})\right) \tag{39}$$

Substituting (38) into (37), we can obtain

$$p(y_{is}|F_{is}) \approx \frac{1}{1 + (\sum_{j=1}^{n_i} \exp(v f_{ijs}))^{-y_{is}/v}} \tag{40}$$

which is an infinitely differentiable function.

Thus, formulas (16) and (17) may be approximately updated as below,

$$\nabla \log p(Y|F) \approx Ed, \qquad W \triangleq -\nabla\nabla \log p(Y|F) \approx EW_0E^{\mathrm{T}} \tag{41}$$

where, $d = [d_{11}, d_{21}, \dots, d_{n1}, \dots, d_{1Q}, d_{2Q}, \dots, d_{nQ}]^{\mathrm{T}}$, $d_{is} = y_{is}/(1 + \exp(y_{is} \max_j f_{ijs}))$, $W_0 = \operatorname{diag}\{w_{11}, w_{21}, \dots, w_{n1}, \dots, w_{1Q}, w_{2Q}, \dots, w_{nQ}\}$, $w_{is} = \exp(y_{is} \max_j f_{ijs})/(1 + \exp(y_{is} \max_j f_{ijs}))^2$, $E = \operatorname{diag}\{e_{11}, e_{21}, \dots, e_{n1}, \dots, e_{1Q}, e_{2Q}, \dots, e_{nQ}\}$, e_{is} is an n_i dimensional column vector with one on its j_0th element and zero elsewhere, $j_0 = \arg\max_j f_{ijs}$.

The iterative formula (21) can be expressed as

$$F_{new} = \bigl(K^f \otimes K\bigr)E\bigl(I - W_0^{1/2}B^{-1}W_0^{1/2}E^{\mathrm{T}}\bigl(K^f \otimes K\bigr)E\bigr)\bigl(d + W_0E^{\mathrm{T}}F\bigr) \tag{42}$$

where $B = I + W_0^{1/2}E^{\mathrm{T}}(K^f \otimes K)EW_0^{1/2}$. And the formula (32) for computing K^f is

$$K^f = K^f_{old} - K^f_{old}HK^f_{old}/\left(\sum_{i=1}^{n} n_i\right) \tag{43}$$

where the (i, j)th element of H is $H_{ij} = \operatorname{tr}(C_{ij}e_j^{\mathrm{T}}Ke_i)$, $e_i = \operatorname{diag}\{e_{1i}, e_{2i}, \dots, e_{ni}\}$.

At last, the distribution (24) of the latent variable F_* and the predictive probability (25) can be respectively written as

$$\begin{aligned} p\bigl(F_*|D, Y, X_*, K^f\bigr) = {}& \mathcal{N}\bigl(F_*|\bigl(K^f \otimes K_*^{\mathrm{T}}\bigr)E\hat{d}, K^f \otimes K_{**} \\ & - \bigl(K^f \otimes K_*^{\mathrm{T}}\bigr)E\hat{W}_0^{\frac{1}{2}}\hat{B}^{-1}\hat{W}_0^{\frac{1}{2}}E^{\mathrm{T}}\bigl(K^f \otimes K_*\bigr)\bigr) \end{aligned} \tag{44}$$

and

$$p\bigl(y_{*s} = 1|D, Y, X_*, K^f\bigr) = \int \sigma\Bigl(\max_j f_{*js}\Bigr)p\bigl(F_{*s}|D, Y, X_*, K^f\bigr)dF_{*s} \tag{45}$$

Until now we have successfully modified the proposed algorithm to satisfy the standard multi-instance assumption. It is also necessary to estimate the computational complexity of it. It can be seen that the main difference from the original algorithm is that the term $E^{\mathrm{T}}(K^f \otimes K)E$ is used in the modified algorithm instead of K_{sum}. Since E is a sparse matrix and each of its elements is 0 or 1, the term $E^{\mathrm{T}}(K^f \otimes K)E$ is indeed a submatrix of $K^f \otimes K$ which has the same size as $K^f \otimes K_{sum}$. Thus, the computational complexity of the modified algorithm would be the same as the original one.

4 Experiments

In order to obtain a more comprehensive understanding about the proposed algorithm, we test it on both multi-label learning problems and multi-instance learning problems. Since the performance evaluation of multi-label learning algorithm is much more complicated than single-label learning one, the following multi-label evaluation metrics proposed by Schapire and Singer (2000) are used in this paper: (a) *Average precision*: computes the average fraction of labels ranked above a particular label $y \in Y$ which actually is in Y. (b) *Coverage*: evaluates how far one needs to go in the list of labels to cover all the relevant labels of a sample. (c) *Hamming loss*: evaluates how many times an object-label pair is misclassified, i.e., a proper label is missed or a wrong label is predicted. (d) *One—error*: determines how many times the top-ranked label is not in the set of proper labels of a sample. (e) *Ranking loss*: evaluates the average fraction of label pairs that not correctly ordered for a sample. Due to page limit, we only give some simple descriptions of these metrics, and the detailed definitions of these metrics can be found in Schapire and Singer (2000). In order to make the smaller value to indicate the better performance of the algorithm for all evaluation metrics, 1—*Average precision* is used in all the experiments.

4.1 Multi-label learning problems

In this section, the proposed algorithm satisfying the collective multi-instance assumption is validated on two MIML data sets respectively come from a multi-label scene classification problem and a multi-label text categorization problem. As shown in the second row in Table 1, the scene classification data set which is proposed by Zhou and Zhang (2007) contains 2000 natural scene images belonging to five classes. Each image is represented as a bag of nine 15-dimensional instances generated by the SBN method (Maron and AL 1998). And over 22% images belong to multiple classes simultaneously and the average number of labels per image is 1.24. The text categorization data set is a subset of the widely studied Reuters-21578 collection. Some information of this data set is listed in the third row in Table 1. In this data set, each document is represented as a bag of instances, where each instance is obtained by splitting the document into several passages by using overlapping windows of maximal 50 words. The seven most frequent categories are considered. After removing the documents that do not have labels and randomly removing some documents which have only one label, the data set consists of 2000 documents and about 15% samples have multiple labels. More detailed information of this data set can be found in Zhou et al. (2012) or Zhang and Zhou (2008). In order to evaluate the relative performance of the proposed algorithm, it is also compared with three existing MIML algorithms named MIMLRBF (Zhang and Wang 2009), MIMLSVM (Zhou and Zhang 2007) and MIML-kNN (Zhang 2010).

In order to speed up the computation of the proposed algorithm, as shown in Sect. 3.6, we need to approximate the kernel matrix K_{sum} with an $n \times m$ matrix P, i.e., $K_{sum} \approx PP^{\mathrm{T}}$. Although the computational complexity is greatly reduced when $m \ll n$, the quality of the solution may be not guaranteed. Thus, it is necessary to analysis the influence of m on the algorithm. Figure 3 shows the relationships between m and different aspects of the algorithm on the scene classification data set when the Gaussian kernel (2) with $\delta = 1$ is used. In this experiment, the data set is randomly partitioned in half to form a training set and a testing set. We repeat the experiment for 10 random splits, and report the mean of the results obtained over 10 different testing sets. Figure 3(a) illustrates the relationships between m and the relative errors of latent variables F and K^f. It can be seen that the errors of F and K^f reduce evidently when m changes from 5 to 100 and then gradually tend to 0 in the remaining increasing phase of m. The relative error is defined as follows: taking F as an example, let F_0 be the truth value of F obtained by using K_{sum}, and F_1 be an approximate value obtained by using PP^{T}, the relative error of F_1 is $\|F_1 - F_0\|/\|F_0\|$. Figure 3(b) depicts how the algorithm performs on the data set under different m. Just like what we have foreseen, the performance of the proposed algorithm will not be guaranteed if m is very small. Fortunately, as can be seen from the figure, the performance will not change significantly as long as m is greater than 50 which still is far smaller than 1000. Moreover, in the case of $50 < m < 150$, although the errors of F and K^f are still great, the performance is already on par with the best one. That is because the prediction results of the algorithm are mainly dependent on the sign of F rather than its numerical value. Figure 3(c) presents the influence of m on the computing time of the algorithm, which is in accord with the computational complexity of the algorithm achieved in Sect. 3.7, i.e., the computational complexity of the

Table 1 Characteristics of the MIML data sets

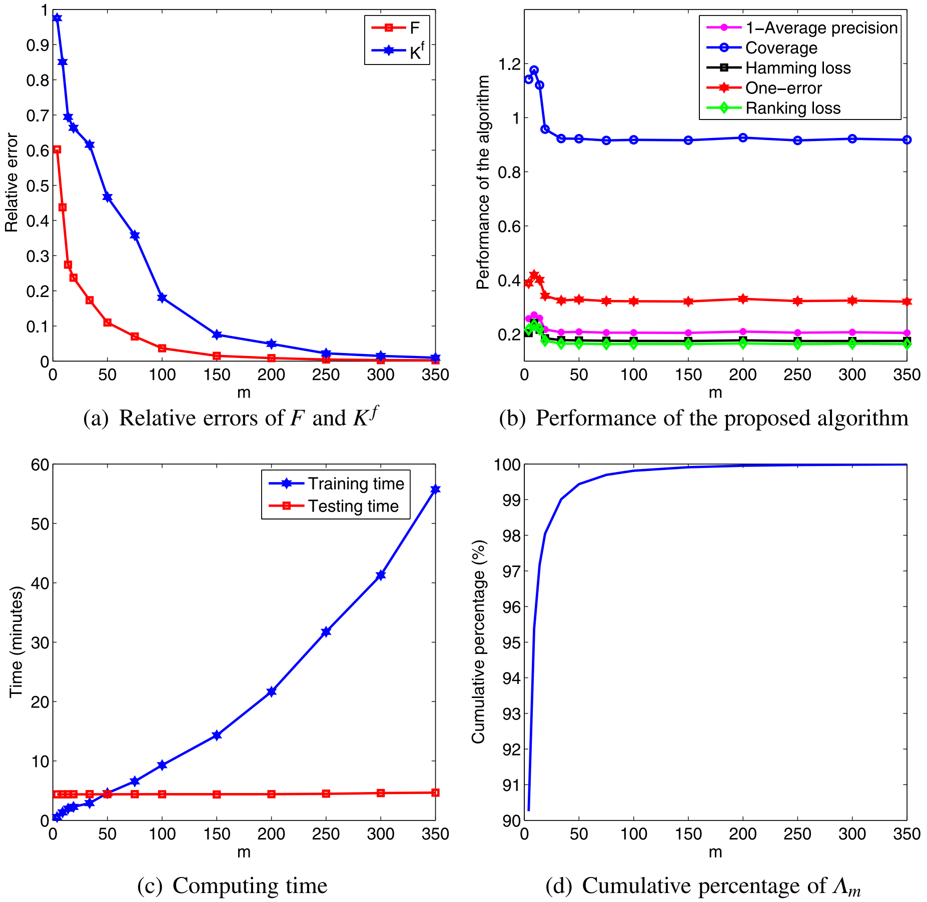
Data sets	Domains	Number of bags	Number of classes	Number of features	Average number of labels	Instances per bag		
						Min	Max	Mean
Scene	Vision	2000	5	15	1.24	9	9	9
Reuters	Text	2000	7	243	1.15	2	26	3.56

Table 2 The experimental results of the compared algorithms (mean ± std.) on the scene classification data set

Evaluation metric	Algorithms				
	The proposed algorithm		MIMLRBF	MIMLSVM	MIML-kNN
	Normal	Special			
1-Average precision	**0.183** ± **0.018**	0.225 ± 0.027	0.216 ± 0.017	0.238 ± 0.019	0.210 ± 0.017
Coverage	**0.857** ± **0.063**	1.002 ± 0.088	1.018 ± 0.059	1.100 ± 0.093	0.943 ± 0.046
Hamming loss	**0.165** ± **0.010**	0.186 ± 0.007	0.183 ± 0.013	0.192 ± 0.011	0.173 ± 0.010
One-error	**0.286** ± **0.032**	0.346 ± 0.047	0.329 ± 0.032	0.361 ± 0.028	0.331 ± 0.034
Ranking loss	**0.146** ± **0.015**	0.181 ± 0.019	0.184 ± 0.016	0.205 ± 0.020	0.169 ± 0.013

model is proportional to m^2 in the training stage but irrelevant to m in the testing stage. Considering that the relationship between m and the performance of the algorithm may change as the training set or number n of samples changes and it is difficult to determine a proper m in practice, the cumulative percentage of Λ_m is introduced to determine m. Let Λ be the diagonal matrix consisting of the eigenvalues of K_{sum}, the cumulative percentage of Λ_m is defined as $\mathrm{tr}(\Lambda_m)/\mathrm{tr}(\Lambda)$. The relationship between m and the cumulative percentage of Λ_m is depicted in Fig. 3(d). It can be seen that the shape of the curve in Fig. 3(d) is just opposite to the ones in Fig. 3(b). We can deduce that the significant improvement of the performance may be caused by the significant increase of the cumulative percentage in the initial increasing phase of m. For validating this supposition, we analysis the relationship between the performance and the cumulative percentage on different data sets and using different kernels with various parameters. We found that the performance of the proposed algorithm gradually gets better when the cumulative percentage tends to 100% and will not significantly change as long as the cumulative percentage is greater than 99.5%. In contrast, there is no unified rule to follow in the relationship between the performance of the algorithm and m. Thus, in the practical applications, m can be obtained based on the condition $\mathrm{tr}(\Lambda_m)/\mathrm{tr}(\Lambda) > 0.995$. Note that the m obtained through above condition is still far smaller than the size of training set in most situations. Taking the experiment of Fig. 3 as an example, the cumulative percentage already reaches to 99.5% when m is 50.

Tables 2 and 3 summarize the experimental results of each compared algorithm obtained by using the 10-fold cross-validation on the scene and Reuters data sets respectively, where the best result on each metric is shown in bold face. For the kernel-based algorithms including MIMLSVM and the proposed one, the Gaussian kernel (2) and linear kernel (3) are used on the scene and Reuters data sets respectively. For a fair comparison, the 2-fold cross validation is performed on the training data set for each algorithm to select the optimal parameter. It is evident from these tables that the proposed algorithm achieves superior performance to the other existing algorithms in terms of all metrics. In addition, the MIMLSVM algorithm performs apparently worse on the both data sets, which may be an experimental evidence



(a) Relative errors of F and K^f

(b) Performance of the proposed algorithm

(c) Computing time

(d) Cumulative percentage of Λ_m

Fig. 3 The influence of m on the proposed algorithm

Table 3 The experimental results of the compared algorithms (mean ± std.) on the Reuters data set

Evaluation metric	Algorithms				
	The proposed algorithm		MIMLRBF	MIMLSVM	MIML-kNN
	Normal	Special			
1-Average precision	**0.029** ± **0.009**	0.033 ± 0.011	0.037 ± 0.008	0.039 ± 0.017	0.046 ± 0.013
Coverage	**0.252** ± **0.037**	0.273 ± 0.039	0.283 ± 0.032	0.298 ± 0.061	0.341 ± 0.067
Hamming loss	**0.030** ± **0.003**	0.033 ± 0.005	0.034 ± 0.005	0.033 ± 0.008	0.039 ± 0.006
One-error	**0.046** ± **0.015**	0.052 ± 0.017	0.059 ± 0.012	0.061 ± 0.026	0.069 ± 0.022
Ranking loss	**0.015** ± **0.005**	0.018 ± 0.006	0.019 ± 0.005	0.022 ± 0.009	0.027 ± 0.008

of the supposition that degeneration methods may lose information during the degeneration process.

As described in Sect. 3, compared with the existing MIML algorithms, a main contribution of the proposed algorithm is that the correlations among labels are exploited by using

Table 4 The computing time of each compared algorithm on both the scene classification and Reuters data sets (minutes)

Computing time		Algorithms				
		The proposed algorithm		MIMLRBF	MIMLSVM	MIML-kNN
		Normal	Special			
Scene	Training time	133.43 ± 5.22	0.49 ± 0.02	4.68 ± 0.07	4.92 ± 0.09	5.76 ± 0.01
	Testing time	1.63 ± 0.01	1.63 ± 0.02	0.33 ± 0.01	0.26 ± 0.01	6.03 ± 0.01
Reuters	Training time	42.55 ± 1.37	2.38 ± 0.17	1.51 ± 0.02	2.0 ± 0.08	2.78 ± 0.06
	Testing time	0.41 ± 0.01	0.43 ± 0.01	0.1 ± 0.01	0.06 ± 0.01	2.34 ± 0.06

a covariance matrix. In order to further investigate whether the superior performance of the proposed algorithm benefits by considering the correlations among labels, the third column 'the proposed algorithm (special)' of each table presents the experimental result of the proposed algorithm when the covariance matrix K^f is set to be the identity matrix, i.e., the labels are considered as mutually independent ones. It can be seen that the covariance matrix has important influence on the performance of the algorithm. In other words, as what we expect, the correlations among the labels are the important information for improving the performance of the MIML algorithms. It is interesting that the performance of the proposed algorithm (special) also is competitive compared with the existing algorithms.

It is well known that the efficiency also is an important factor for investigating the practicality of an algorithm. Table 4 reports the training and testing time consumed by each compared algorithm on both data sets. These results are based on the experiments conducted on a 2.7 GHz PC with 2 GB RAM. It can be seen that the proposed algorithm (normal) spends more time than the others in the training stage. That is mainly because the EM-like is of linear rates of convergence and we only initialize K^f with an identical matrix. In practice, we can speed up the algorithm by initializing K^f with the value obtained in a subset of the training set. Certainly, we will try to solve the model by using an approach with superlinear rates of convergence in the future. In the testing stage, the efficiency of the proposed algorithm is slightly worse than MIMLRBF and MIMLSVM while superior to MIML-kNN.

4.2 Multi-instance learning problems

In addition to the correlations among the labels, another motivation of the paper is to model the connections between the instances and labels. In Sect. 3.8, we also extend the proposed algorithm to obey the standard multi-instance assumption. In this section, we will conduct experiments on several multi-instance learning problems to justify the usefulness of modeling different instance-label correspondences. Note that 'collective' and 'standard' are used to distinguish the algorithms satisfying the collective multi-instance assumption and the standard multi-instance assumption, respectively.

We first validate the proposed algorithm on a text categorization problem approximately satisfying the standard multi-instance assumption. It includes twenty data sets respectively derived from 20 Newsgroups corpus by Zhou et al. (2009) in order to test the performance of their miGraph algorithm. For each of the twenty data sets, 50 positive and 50 negative bags are generated. Each instance is a post represented by the top 200 TFIDF features. Each positive bag contains 3% posts randomly drawn from the target category and 97% posts randomly and uniformly drawn from the other categories. All instances in negative bags are randomly and uniformly drawn from the other categories. In Table 5, column 2

Table 5 The average accuracies of the compared algorithms (mean ± std.) on the data set proposed by Zhou et al. (2009)

Data sets	Algorithms				
	# inst	MI-Kernel	miGraph	The proposed algorithm	
				Standard	Collective
alt.atheism	54.4	60.2 ± 3.9	65.5 ± 4.0	**84.5 ± 2.5**	47.4 ± 4.4
comp.graphics	30.9	47.0 ± 3.3	77.8 ± 1.6	**83.8 ± 1.9**	51.0 ± 0.0
comp.os.ms-windows.misc	51.8	51.0 ± 5.2	63.1 ± 1.5	**67.4 ± 6.3**	48.0 ± 4.0
comp.sys.ibm.pc.hardware	48.3	46.9 ± 3.6	59.5 ± 2.7	**77.8 ± 2.5**	50.4 ± 1.7
comp.sys.mac.hardware	44.7	44.5 ± 3.2	61.7 ± 4.8	**79.6 ± 1.9**	49.0 ± 3.9
comp.windows.x	31.1	50.8 ± 4.3	69.8 ± 2.1	**79.5 ± 3.0**	50.2 ± 0.8
misc.forsale	53.1	51.8 ± 2.5	55.2 ± 2.7	**71.5 ± 1.7**	50.8 ± 3.3
rec.autos	34.6	52.9 ± 3.3	72.0 ± 3.7	**79.2 ± 1.5**	49.6 ± 2.8
rec.motorcycles	47.3	50.6 ± 3.5	64.0 ± 2.8	**82.0 ± 1.0**	50.0 ± 2.5
rec.sport.baseball	33.6	51.7 ± 2.8	64.7 ± 3.1	**85.2 ± 0.8**	48.2 ± 1.8
rec.sport.hockey	19.8	51.3 ± 3.4	85.0 ± 2.5	**90.0 ± 1.7**	49.2 ± 1.3
sci.crypt	42.8	56.3 ± 3.6	69.6 ± 2.1	**77.8 ± 1.9**	48.8 ± 2.3
sci.electronics	31.9	50.6 ± 2.0	87.1 ± 1.7	**91.6 ± 0.5**	53.0 ± 0.0
sci.med	30.5	50.6 ± 1.9	62.1 ± 3.9	**84.2 ± 0.8**	49.8 ± 3.4
sci.space	36.6	54.7 ± 2.5	75.7 ± 3.4	**80.4 ± 1.8**	49.5 ± 1.9
sci.religion.christian	46.8	49.2 ± 3.4	59.0 ± 4.7	**82.0 ± 1.4**	45.5 ± 1.7
talk.politics.guns	35.6	47.7 ± 3.8	58.5 ± 6.0	**75.4 ± 2.3**	48.0 ± 4.1
talk.politics.mideast	33.8	55.9 ± 2.8	73.6 ± 2.6	**80.2 ± 1.5**	48.3 ± 3.0
talk.politics.misc	47.9	51.5 ± 3.7	70.4 ± 3.6	**70.5 ± 2.2**	55.0 ± 6.3
talk.religion.misc	46.1	55.4 ± 4.3	63.3 ± 3.5	**76.0 ± 1.7**	50.0 ± 1.2

lists the average number of instances per bag for each data set. The average accuracy (%) with standard deviation of each compared algorithm is presented in the other columns of Table 5, where the best result on each data set is shown in bold face. The accuracies of miGraph and MI-Kernel are taken from Zhou et al. (2009). For all these methods, the ten-times 10-fold cross validation is run on each data set; moreover, the Gaussian RBF Kernel is used and the parameters are determined through cross validation on the training set. It is obviously that the performance of the proposed algorithm (standard) is superior to those of other algorithms. In addition, the performance of the proposed algorithm (collective) is not competitive. The main reason may be that there is only about 3% positive instances in each positive bag but the proposed algorithm (collective) is based on the assumption that bag's label is collectively determined by all its instances.

Image categorization has been formulated and tackled as a MIL problem in the past work. The 1000-Image and 2000-Image data sets have been used in many literatures such as Chen and Wang (2004), Chen et al. (2006) and Zhou et al. (2009) to test the performance of their algorithms. In this section, these data sets are used to further evaluate the performance of the proposed algorithm. The 1000-Image and 2000-Image data sets consist of ten and twenty categories of COREL images, respectively, where each category contains 100 images. Each image is regarded as a bag and its segmented regions are regarded as instances. Each instance is characterized by a 9-dimensional feature vector in several aspects such as the color, texture, and shape properties. Being different with the above text categorization problem, the

Table 6 The average accuracies of the compared algorithms on the data set proposed by Chen and Wang (2004)

Algorithms	1000-Image Data Set	2000-Image Data Set
The proposed algorithm (collective)	**84.6: [83.6, 85.5]**	**73.0: [71.8, 74.2]**
The proposed algorithm (standard)	70.0: [67.5, 72.5]	53.2: [48.6, 57.8]
MIGraph	83.5: [81.2, 85.7]	72.1: [71.0, 73.2]
miGraph	81.4: [80.2, 82.6]	70.5: [68.7, 72.3]
MI-Kernel	81.8: [80.1, 83.6]	72.0: [71.2, 72.8]
MILES	82.6: [81.4, 83.7]	68.7: [67.3, 70.1]
DD-SVM	81.5: [78.5, 84.5]	67.5: [66.1, 68.9]
MI-SVM	74.7: [74.1, 75.3]	54.6: [53.1, 56.1]
k-means-SVM	69.8: [67.9, 71.7]	52.3: [51.6, 52.9]

class labels of the sample in these data sets are usually determined by the collective property of multiple regions. Taking the label 'skiing' as an example, it should include the regions of snow, people, and perhaps a steep slope or mountain in the image. Since the proposed algorithm will be compared with other existing MIL algorithms, we adopt the same experimental routine as what has been used in Chen et al. (2006) and Zhou et al. (2009). Images within each category are randomly partitioned in half to form a training and a testing set. Each experiment is repeated for 5 random splits. Since the problem is multi-class, one-against-the rest strategy is used by the proposed algorithm. The average accuracy with 95% confidence intervals obtained on 5 different testing sets is shown in Table 6. In order to evaluate the relative performance of the proposed algorithms, the results of some other MIL algorithms reported in Chen et al. (2006) and Zhou et al. (2009) are also shown in the table. As can be seen from Table 6, the proposed algorithm can achieve competitive performance on both 1000-Image and 2000-Image data sets. Moreover, in contrast to the results of Table 5, the performance of the proposed algorithm (standard) is relatively poor. It also is consistent with the instance-label correspondence of the image data sets.

Thus, besides to the superior performance of the proposed algorithm, it also can be seen from the quite different results of the proposed algorithm in the above two experiments that an important condition for achieving better performance in a certain problem is that the algorithm can efficiently capture the connections between the instances and the labels contained in the problem.

5 Conclusion

Considering that the existing MIML algorithms can not efficiently exploit the advantages of the MIML representation, a novel MIML algorithm was proposed by employing Gaussian process in this paper. Through supposing a latent function for every label over instance space, the connections between instances and class labels can be exploited by defining different likelihoods and the correlations among labels can be identified by a covariance matrix of the latent functions which may be obtained by maximizing a marginal likelihood function. Experimental results show that the proposed algorithm outperforms the existing ones. It is well known that the ways in which instances trigger labels are different in different problems. In the future work, we will try to define the likelihood $p(y_{is} = 1|F_{is})$ based on other assumptions such that the algorithm can be used to deal with more practical problems. Moreover, since the computational complexity of the algorithm in the training stage is still prohibited for the problems with larger number of labels, to improve the computational complexity of the algorithm also is a major focus in the future research.

Appendix: Derivation of formula (41)

By differentiating $\log p(Y|F)$ with regard to f_{ijs} based on (40), we obtain

$$\frac{\partial \log p(Y|F)}{\partial f_{ijs}} = \frac{y_{is}}{(1 + (\sum_{j_0} \exp(v f_{ij_0 s}))^{y_{is}/v})(1 + \sum_{j_0 \neq j} \exp(v(f_{ij_0 s} - f_{ijs})))} \tag{46}$$

$$-\frac{\partial^2 \log p(Y|F)}{\partial f_{i_1 j_1 s_1} \partial f_{ijs}} = \begin{cases} 0, & (i,s) \neq (i_1, s_1) \\ p_1(F), & (i,j,s) = (i_1, j_1, s_1) \\ p_2(F), & \text{other} \end{cases} \tag{47}$$

where,

$$p_1(F) = \frac{(\sum_{j_0} \exp(v f_{ij_0 s}))^{y_{is}/v} - y_{is} v (1 + (\sum_{j_0} \exp(v f_{ij_0 s}))^{y_{is}/v})(\sum_{j_0 \neq j} \exp(v(f_{ij_0 s} - f_{ijs})))}{(1 + (\sum_{j_0} \exp(v f_{ij_0 s}))^{y_{is}/v})^2 (1 + \sum_{j_0 \neq j} \exp(v(f_{ij_0 s} - f_{ijs})))^2}$$

$$p_2(F) = \frac{(\sum_{j_0} \exp(v f_{ij_0 s}))^{y_{is}/v}}{(1 + (\sum_{j_0} \exp(v f_{ij_0 s}))^{y_{is}/v})^2 (1 + \sum_{j_0 \neq j} \exp(v(f_{ij_0 s} - f_{ijs})))(1 + \sum_{j_0 \neq j_1} \exp(v(f_{ij_0 s} - f_{ij_1 s})))} + \frac{y_{is} v \exp(v(f_{ij_1 s} - f_{ijs}))}{(1 + (\sum_{j_0} \exp(v f_{ij_0 s}))^{y_{is}/v})(1 + \sum_{j_0 \neq j} \exp(v(f_{ij_0 s} - f_{ijs})))^2}$$

Suppose that for each (i,s) there exists unique $j_{max} \in \{1, 2, \ldots, n_i\}$ such that $f_{ij_{max}s} = \max_j f_{ijs}$, formulas (46) and (47) may be approximatively written as

$$\frac{\partial \log p(Y|F)}{\partial f_{ijs}} \approx \begin{cases} \frac{y_{is}}{1+\exp(y_{is} \max_{j_0} f_{ij_0 s})}, & j = \arg\max_{j_0} f_{ij_0 s} \\ 0, & \text{other} \end{cases} \tag{48}$$

and

$$-\frac{\partial^2 \log p(Y|F)}{\partial f_{i_1 j_1 s_1} \partial f_{ijs}} \approx \begin{cases} \frac{\exp(y_{is} \max_{j_0} f_{ij_0 s})}{(1+\exp(y_{is} \max_{j_0} f_{ij_0 s}))^2}, & (i,j,s) = (i_1, j_1, s_1) \text{ and } j = \arg\max_{j_0} f_{ij_0 s} \\ 0, & \text{other} \end{cases} \tag{49}$$

By using (48) and (49), we can obtain (41) directly.

References

Barutcuoglu, Z., Schapire, R. E., & Troyanskaya, O. G. (2006). Hierarchical multi-label prediction of gene function. *Bioinformatics*, *22*(7), 830–836.

Bonilla, E. V., Chai, K. M. A., & Williams, C. K. I. (2008). Multi-task Gaussian process prediction. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Boutell, M. R., Luo, J. B., Shen, X. P., & Brown, C. M. (2004). Learning multi-label scene classification. *Pattern Recognition*, *37*(9), 1757–1771.

Chen, Y. X., & Wang, J. Z. (2004). Image categorization by learning and reasoning with regions. *Journal of Machine Learning Research*, *5*, 913–939.

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

Chou, K. C., & Shen, H. B. (2010). A new method for predicting the subcellular localization of eukaryotic proteins with both single and multiple sites: Euk-mPLoc 2.0. *PLoS ONE*, *5*(3), e9931.

Clare, A., & King, R. D. (2001). Knowledge discovery in multi-label phenotype data. In *Proceedings of the 5th European conference on principles of data mining and knowledge discovery*, Freiburg, Germany (pp. 42–53).

Cour, T., Sapp, B., Jordan, C., & Taskar, B. (2009). Learning from ambiguously labeled images. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 919–926).

Diplaris, S., Tsoumakas, G., Mitkas, P. A., & Vlahavas, I. (2005). Protein classification with multiple algorithms. In *10th Panhellenic conference on informatics*.

Elisseeff, A., & Weston, J. (2002). A kernel method for multi-labeled classification. In *Advances in neural information processing systems* (pp. 681–687). Cambridge: MIT Press.

Foulds, J., & Frank, E. (2010). A review of multi-instance learning assumptions. *Knowledge Engineering Review*, *25*(1), 1–25.

Godbole, S., & Sarawagi, S. (2004). Discriminative methods for multi-labeled classification. In *The 8th Pacific-Asia conference on knowledge discovery and data mining*, Sydney, Australia.

He, J. J., Gu, H., & Jiang, S. R. (2011). Twin Gaussian processes for binary classification. In *Proceedings of the 11th IEEE international conference on data mining*, Vancouver, Canada (pp. 1074–1079).

He, J. J., Gu, H., & Wang, Z. L. (2012). Multi-instance multi-label learning based on Gaussian process with application to visual mobile robot navigation. *Information Sciences*, *190*, 162–177.

Jie, L., & Orabona, F. (2010). Learning from candidate labeling sets. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Jin, R., Wang, S. J., & Zhou, Z. H. (2009). Learning a distance metric from multi-instance multi-label data. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, Miami, FL (pp. 896–902).

Kazawa, H., Izumitani, T., Taira, H., & Maeda, E. (2005). Maximal margin labeling for multi-topic text categorization. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Kim, H. C., & Ghahramani, Z. B. (2006). Bayesian Gaussian process classification with EM-EP algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *28*(12), 1948–1959.

Lawrence, N. D., & Platt, J. C. (2004). Learning to learn with the informative vector machine. In *Proceedings of the 21st international conference on machine learning* (pp. 512–519).

Li, Y. X., Ji, S. W., Kumar, S., Ye, J. P., & Zhou, Z. H. (2012). Drosophila gene expression pattern annotation through multi-instance multi-label learning. *ACM/IEEE Transactions on Computational Biology and Bioinformatics*, *9*(1), 98–112.

Maron, O., & Lozano-Pérez, T. (1998). A framework for multiple-instance learning. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Maron, O., & Ratan, A. L. (1998). Multiple-instance learning for natural scene classification. In *Proceeding of the 15th international conference on machine learning*, Madison, WI (pp. 341–349).

McCallum, A. K. (1999). Multi-label text classification with a mixture model trained by EM. In *Working notes of the AAAI'99 workshop on text learning*, Orlando, FL.

Nguyen, N. (2010). A new svm approach to multi-instance multi-label learning. In *Proceedings of the 10th IEEE international conference on data mining*, Sydney, Australia (pp. 384–392).

Nickisch, H., & Rasmussen, C. E. (2008). Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, *9*, 2035–2078.

Rasmussen, C. E., & Williams, K. I. (2006). *Gaussian process for machine learning*. Cambridge: MIT Press.

Schapire, R. E., & Singer, Y. (2000). BoosTexter: a boosting-based system for text categorization. *Machine Learning*, *39*(2–3), 135–168.

Seeger, M. (2003). *Bayesian Gaussian process models: PAC-Bayesian generalization error bounds and sparse approximations*. PhD thesis, University of Edinburgh.

Ueda, N., & Saito, K. (2003). Parametric mixture models for multi-labeled text. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Wang, W., & Zhou, Z. H. (2012). Learnability of multi-instance multi-label learning. *Chinese Science Bulletin*, in press.

Wieczorkowska, A., Synak, P., & Ras, Z. W. (2006). Multi-label classification of emotions in music. In *International conference on intelligent information processing and web mining*.

Williams, C. K. I., & Barber, D. (1998). Bayesian classification with Gaussian process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(12), 1342–1351.

Williams, C. K. I., & Rasmussen, C. E. (1996). Gaussian processes for regression. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Williams, C. K. I., & Seeger, M. (2001). Using the Nyström method to speed up kernel machines. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Yang, S. H., Zha, H. Y., & Hu, B. G. (2009). Dirichlet-Bernoulli alignment: a generative model for multi-class multi-label multi-instance corpora. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Yang, S. H., Bian, J., & Zha, H. Y. (2010). Hybrid generative/discriminative learning for automatic image annotation. In *Proceedings of the 26th conference on uncertainty in artificial intelligence*.

Yang, Y. M. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval*, *1*(1–2), 69–90.

Zha, Z. J., Hua, X. S., Mei, T., Wang, J. D., Qi, G. J., & Wang, Z. F. (2008). Joint multi-label multi-instance learning for image classification. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*, Anchorage, AK (pp. 1–8).

Zhang, M. L. (2010). A k-nearest neighbor based multi-instance multi-label learning algorithm. In *The 22nd international conference on tools with artificial intelligence* (pp. 207–212).

Zhang, M. L., & Wang, Z. J. (2009). MIMLRBF: RBF neural networks for multi-instance multi-label learning. *Neurocomputing*, *72*(16–18), 3951–3956.

Zhang, M. L., & Zhou, Z. H. (2006). Multi-label neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, *18*(10), 1338–1351.

Zhang, M. L., & Zhou, Z. H. (2007a). ML-kNN: a lazy learning approach to multi-label learning. *Pattern Recognition*, *40*(7), 2038–2048.

Zhang, M. L., & Zhou, Z. H. (2007b). Multi-label learning by instance differentiation. In *Proceedings of the 22nd AAAI conference on artificial intelligence*, Vancouver, Canada (pp. 669–674).

Zhang, M. L., & Zhou, Z. H. (2008). M3MIML: a maximum margin method for multi-instance multi-label learning. In *Proceedings of the 8th IEEE international conference on data mining*, Pisa, Italy (pp. 688–697).

Zhou, Z. H. (2004). *Multi-instance learning: a survey* (Technical Report). AI Lab, Department of Computer Science and Technology, Nanjing University, China.

Zhou, Z. H., & Zhang, M. L. (2007). Multi-instance multi-label learning with application to scene classification. In *Advances in neural information processing systems*. Cambridge: MIT Press.

Zhou, Z. H., Sun, Y. Y., & Li, Y. F. (2009). Multi-instance learning by treating instances as non-i.i.d. samples. In *Proceedings of the 26th international conference on machine learning* (pp. 1249–1256).

Zhou, Z. H., Zhang, M. L., Huang, S.J., & Li, Y. F. (2012). Multi-instance multi-label learning. *Artificial Intelligence 176*(1), 2291–2320.