bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Gaussian Process Based Heteroscedastic Noise Modeling for Tumor Mutation Burden Prediction from Whole Slide Images

Sunho Park*, Hongming Xu and Tae Hyun Hwang

Department of Quantitative Health Sciences, Lerner Research Institute, Cleveland clinic, 9500 Euclid Ave, Cleveland, OH 44195, USA

*Corresponding author. Email: parks@ccf.org

Tumor mutation burden (TMB) is a quantitative measurement of how many mutations present in tumor cells from a patient tumor as assessed by next-generation sequencing (NGS) technology. High TMB is used as a predictive biomarker to select patients that likely respond to immunotherapy in many cancer types, thus it is critical to accurately measure TMB for cancer patients who need to receive the immunotherapy. Recent studies showed that image features from histopathology whole slide images can be used to predict genetic features (e.g., mutation status) or clinical outcome of cancer patients. In this study, we develop a computational method to predict the TMB level from cancer patients' histopathology whole slide images. The prediction problem is formulated as multiple instance learning (MIL) because a whole slide image (a bag) has to be divided into multiple image blocks (instances) due to computational reasons but a single label is available only to an entire whole slide image not to each image block. In particular, we propose a novel heteroscedastic noise model for MIL based on the framework of Gaussian process (GP), where the noise variance is assumed to be a latent function of image level features. This noise variance can encode the confidence in predicting the TMB level from each training image and make the method to put

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

different levels of effort to classify images according to how difficult each image can be correctly classified. The method tries to fit an easier image well while it does not put much effort in classifying a harder (ambiguous) image correctly. Expectation and propagation (EP) is employed to efficiently infer our model and to find the optimal hyper-parameters. We have demonstrated from synthetic and real-world data sets that our method outperforms on TMB prediction from whole slide images base-line methods, including a special case of our method that does not include the heteroscedastic noise modeling and multiple instance ordinal regression (MIOR) that is one of few algorithms to solve ordinal regression in the MIL setting.

1 Introduction

Tumor mutation burden (TMB) measures the number of mutations found in a patient's tumor assessed by next-generation sequencing (NGS) technology. Recent studies have shown that TMB is associated with better response to immunotherapy in many cancer types including lung and bladder cancer, thus can be used as a predictive biomarker for immunotherapy (*3, 10*). One key challenge is that not all patients would have adequate tumor tissue or will be able to undergo a tumor biopsy to generate NGS-based genomic testing to measure TMB. Although a blood-based TMB (e.g., liquid biopsies) recently becomes available, this approach also poises many technical challenges to measure TMB accurately. On the other hand, histopathology whole slide images used by pathologists to diagnose and assess tumor stage for cancer patients are widely available for the most cancer patients. Recent studies have shown that computational histopathology whole slide images (hereafter whole slide images) could be used to predict tumor subtype, clinical outcome, or certain mutation status (*6, 25, 31*). The fundamental assumption of these studies is that the morphology present in whole slides images is reflecting genetic aberrations driving cancer; therefore the algorithm utilizing morphological image features can reversely predict genetic status,

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

clinical outcome, etc.

Our central hypothesize of this study is that the morphological image features generated from whole slide images could be used to predict TMB status. However, there are several difficulties to implement efficient TMB prediction methods from whole slide images. The main challenge is that the relation between TMB and morphology on images has not been fully explored. There has been no research to address this relation clearly yet. The second challenge is related to computational feasibility. Whole slide images are typically very huge: the size of either dimension of images is typically larger than 100,000 pixels and the file size of an image can be from hundreds of megabytes to several gigabytes. One possible solution to the computational difficulty is to divide an image into multiple non-overlapping image blocks. Then predictive or analysis methods are separately applied to each image block in an image, and then the final results can be obtained by aggregating the results from all the image blocks (*6, 25*). On the other hands, in (*31*) feature vectors from all the image blocks in an image are first combined into a single feature vector which is then fed into a SVM classifier as the input with the image's label.

There has been few research to predict genetic features from histopathology images. Most research is designed to predict presence of certain mutations, e.g., EGFR in lung cancer (*6*), from images. As mentioned earlier, due to computational issues, the methods divide an image into multiple blocks and train predictive models on the images blocks (the target label of an image block is assumed to be the same as that of the image to which the block belongs) (*6, 25*). We can consider the same approach to predict TMB from whole side images. However, although this approach can increase the number of labeled training samples, it may degrade the predictive performance of the algorithms because image blocks that might be irrelevant to prediction are also treated as individual training samples and directly affecting the decision boundary. In fact, TMB prediction can be formulated as multiple instance learning (MIL) in which a label is assigned to a set (bag) of instances, not to each instance. In our problem, an image can be considered a bag and image blocks in an image instances. Furthermore the aforementioned methods, where all the image blocks in an image are assumed to have the same label as the image, corresponds

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

to the naive MIL method. It is known that methods that fully account the MIL assumptions typically outperform the naive MIL method.

We develop a prediction method of TMB from whole slide images, where a novel noise model for MIL is proposed based on the framework of Gaussian process (GP). We formulate the prediction problem of TMB as ordinal regression (i.e., predicting TMB low $\preceq$ intermediate $\preceq$ high) and assume a latent function mapping from the input feature space to the intervals on the real line, modeled by a GP. Then, the mean value of the function values evaluated at all image blocks in an image is compared to the corresponding label in a noise (likelihood) model where the noise variance is also modeled by another GP (heteroscedastic noisy modeling) based on image level features that are calculated from feature vectors in an image. This noise variance encodes the confidence that our method has regarding image samples. The method more relies on image samples that are easier (having the higher confidence level or the lower noise variance) but less relies on image samples that are hard (having the higher noise variance). In other words, the method ties to fit easier image samples well while it does not put much effort in classifying harder (ambiguous) images correctly. Finally, expectation propagation (EP) is employed to efficiently train our model and find the optimal hyper-parameters. We have demonstrated from a synthetic and real-world data sets the usefulness of our heteroscedastic noise modeling. Our contribution can be summarized as follows

- To our best knowledge our work is the first attempt to solve the prediction of TMB from whole slide images in the framework of multiple instance learning (MIL).
- We propose a heteroscedastic noise model for MIL, which can encode the confidence in predicting TMB from an image.

2 Related Work

MIL, first introduced in (*7*), is a variation of supervised learning where a (positive or negative) label is assigned not to an individual instance but to a bag that consists of a set of instances. The standard MIL assumption (*7, 28*) states that a bag is positive if and only if there is at least one positive instance

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

in the bag while a negative bag should not contain any positive instances. This assumption can be implemented by maximum operator (*1, 17, 19*). Assuming that each instance in the ith bag has a latent label $y_{ij} \in \{0, 1\}$, the standard MIL assumption states that the bag's label can be given by

$$y_i = \begin{cases} -1 & \text{iif } \sum_j y_{ij} = 0, \\ 1 & \text{Otherwise.} \end{cases} \tag{1}$$

Furthermore, it can be further simplified using maximum operator as mentioned earlier, i.e., $y_i = \max_j \{y_{ij}\}$. In the standard MIL assumption, we can easily see that only instances in a bag with positive label can affect the class label of the bag. On the other hand, there have been several alternatives to the standard MIL assumption (please refer (*8*)). One of them is the collective assumption (*8, 32*), where all instances in a bag are assumed to equally contribute the bag's label. Our prediction method is based on the collective assumption to connect image blocks (instances) in a patient's whole slide image (a bag) to the patient's TMB level, which will be explained in the next section.

Most of MIL algorithms are designed to solve binary classification problem (*11, 18, 19*). There have been proposed very few MIL methods for ordinal regression problems (*29*) as it is not straightforward to directly extend the standard MIL assumption to multi-class classification or ordinal regression problems. On the other hand, Gaussian process is a powerful Bayesian non-parametric method for nonlinear function estimation problem (*23*). It has been successfully applied to ordinal regression problems as well (*5*). In this study, we propose an efficient ordinal regression method for MIL based on the framework of GP. Compared to the method in (*29*), which gives point estimate solutions, our method could be more attractive because our method is a Bayesian approach, more robust to overfitting, and provides a way to find optimal hyper-parameters by maximizing marginal likelihood (or evidence).

Recently, deep learning based methods have been extensively applied to analyze whole slide images (*6, 16, 25*). These methods could be more desirable than traditional machine learning methods because the performance of traditional machine learning methods rely on the quality of manually designed input features. On the other hand deep learning based methods can learn feature representations directly from images. Moreover, there have been proposed deep learning based method designed for MIL, e.g., (*17*).

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

However, deep learning based method might not be directly applicable to TMB prediction problems. The main difficulties are related to computational issues caused by the nature of whole slide images, i.e., small sample size but a huge size of images. For example, TCGA bladder cancer data used in our experiments contains just hundreds of whole slide images, but either dimension of each image is at least 100,000 pixels. Each image can have hundreds or thousands image blocks even if we focus on only tumorous regions in an image. It becomes more problematic if we train a more complex deep learning model with more many parameters to be tuned. In this study, thus we pursue the traditional machine learning rather than deep learning approaches. We first calculate features from image blocks in a image and build a predictive model to aggregate the results from all the image blocks to obtain the image's final label.

Heteroscedastic noise modeling, which assumes the noise variance to vary across the input space, provides a more realistic noise model. This idea has been widely used for Gaussian process regression (*9, 20, 21, 27*). However, in (*15*) heteroscedastic noise modeling is used for binary classification, where the noise variance is modeled as a function of privileged information that is available only during training time. The authors interpret that this latent function for the noise variance can encode the confidence about each training sample. They also have showed that this modeling improves the performance over a standard GP classifier. Our heteroscedastic modeling is quite related to this approach. We also use the heteroscedastic modeling to encode the confidence in predicting an image sample (i.e., a set of image blocks) by our prediction method in the MIL setting. However, in contrast to their method, in our method the noise variance is assumed to be a function defined on image level features that are calculated from all the image blocks in an image so that our heteroscedastic modeling is also available for test data.

3 Proposed Method

We propose a heteroscedastic noise model for MIL based on the framework of Gaussian process, where a latent scoring function (assumed to be follow a GP) maps from the input feature space to the intervals on the real line, and the noise variance is a latent function (assumed to be follow another GP) defined

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

on image level features that are calculated from all the feature vectors in an image. We also propose an efficient EP algorithm to approximate the posterior distributions to learn the model hyper-parameters.

3.1 Data preparation

Due to computational feasibility, each whole slide image is divided into a set of image blocks. As discussed earlier, we calculate a feature vector from each image block in an image. The whole workflow of extracting image feature vectors from a whole slide image is shown in Figure 1. For detailed image processing and feature extracting steps, please refer our experimental results section (Section 4.2.1).

Let us assume that we are given a data set composed of N whole slide images. As mentioned earlier, we calculate a feature vector from each image block in an image. For example, let $\boldsymbol{x}_{ij} \in \mathbb{R}^D$ denote as a feature vector calculated from the jth block in the ith image, and then $\mathcal{X}_i = \{\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, ..., \boldsymbol{x}_{iN_i}\}$ represents a set of feature vectors of all N_i blocks in the image. From the training data, we will have $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_N\}$ and $\boldsymbol{y} = [y_1, y_2, ..., y_N]^\top$. Here, $y_i \in \mathcal{Y}$ is a target label assigned to the ith image, where $\mathcal{Y} = \{C_1, C_2, .., C_K\}$ are discrete labels in the ordinal relation, i.e., $C_1 \preceq C_2 \preceq ... \preceq C_K$ (K is set to 3 in our TBM prediction problem). Our goal is to learn a latent function that associates a set of feature vectors of an image, $\mathcal{X}_i$, with the corresponding ordinal label y_i.

3.2 Model specification

We first assume a latent function that maps from the input feature space to intervals on the real line, i.e., $f : \mathbb{R}^D \mapsto \mathbb{R}$ and the real line is divided into non-overlapping intervals with thresholds $-\infty < b_1 < b_2 < ... < b_{|\mathcal{Y}|-1} < \infty$. For example, the function value at $\boldsymbol{x}_{ij}$ in the ith image, $f(\boldsymbol{x}_{ij})$, is assumed to be in the interval, $b_{y_i-1} < t \leq b_{y_i}$ (the end points, b_{y_i-1} and b_{y_i}, are determined by the image's target label y_i). We further assume that the function $f(\cdot)$ follows a Gaussian process:

$$f(\boldsymbol{x}_{ij}) \sim \mathcal{GP}(0, \kappa_f(\boldsymbol{x}_{ij}, \cdot)), \tag{2}$$

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

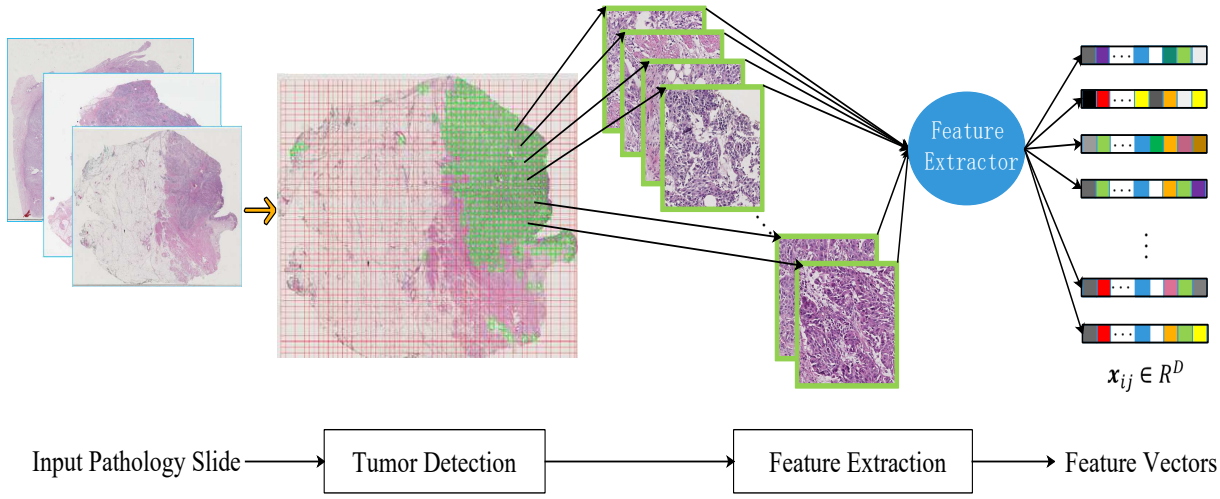

Figure 1: Feature extraction workflow. We divide a whole slid image into a set of image blocks and calculate a feature vector from each image block. The detailed image processing and feature extracting steps are explained in Section 4.2.1.

where $\kappa_f(\cdot,\cdot)$ is a covariance function. In our paper, we use an automatic relevance determination (ARD) SE covariance function

$$\kappa_f(\boldsymbol{x}_{ij}, \boldsymbol{x}_{ik}) = s_f^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D}\frac{1}{\ell_d^2}\Big([\boldsymbol{x}_{ij}]_d - [\boldsymbol{x}_{ik}]_d\Big)^2\right), \tag{3}$$

where s_f is a magnitude parameter and $\ell_d > 0$ is a scale length parameter which determines the relevancy of the dth input feature to the prediction. Letting $f_{ij} \triangleq f(\boldsymbol{x}_{ij})$ and $\boldsymbol{f}$ be a vector containing all function values evaluated at all image blocks in training data, i.e., $\boldsymbol{f} = [f_{11}, f_{12}, ..., f_{ij-1}, f_{ij}, f_{ij+1}, ..., f_{NN_i-1} f_{NN_i}]$, the prior distribution over the latent variable $\boldsymbol{f}$ can be given by $p(\boldsymbol{f}) = \mathcal{N}(0, \boldsymbol{K}_{ff})$, where $\boldsymbol{K}_{ff} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|} = \kappa_f(\mathcal{X}, \mathcal{X})$. However, since the length of $\boldsymbol{f}$ $(= |\mathcal{X}|)$ can be high, we employ sparse approximation (PITC) (*22*) to reduce computational complexity. Let us define $\boldsymbol{u} \in \mathbb{R}^M$ as inducing variables which are function values evaluated at inducing points $\boldsymbol{Z} \triangleq \{\boldsymbol{z}_1, ..., \boldsymbol{z}_M\}$, where $\boldsymbol{z}_m \in \mathbb{R}^D$ $(u_m \triangleq f(\boldsymbol{z}_m))$, the conditional distribution of $\boldsymbol{f}$ given $\boldsymbol{u}$ is defined as follows

$$p_{\text{PITC}}(\boldsymbol{f}|\boldsymbol{u}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{u}, \text{bd}[\boldsymbol{K}_{ff} - \boldsymbol{K}_{fu}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{fu}^{\top}]) \tag{4}$$

where $\boldsymbol{K}_{fu} = \kappa_f(\mathcal{X}, \boldsymbol{Z})$ and $\boldsymbol{K}_{uu}$ is a covariance matrix for the inducing variables, i.e., $\boldsymbol{K}_{uu} = \kappa_f(\boldsymbol{Z}, \boldsymbol{Z}) \in \mathbb{R}^{M\times M}$. Please note that the operator $bd[\boldsymbol{M}]$ forms a block-diagonal matrix from the matrix $\boldsymbol{M}$ where

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

each diagonal block is calculated from each image, i.e., $\left(\boldsymbol{K}_{f_i f_i} - \boldsymbol{K}_{f_i u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{f_i u}^{\top}\right) \in \mathbb{R}^{N_i \times N_i}$, where $\boldsymbol{f}_i = [f_{i1}, f_{i2}, ..., f_{iN}]^{\top} \in \mathbb{R}^{N_i}$, $\boldsymbol{K}_{f_i f_i} = \kappa_f(\mathcal{X}_i, \mathcal{X}_i)$ and $\boldsymbol{K}_{fu} = \kappa_f(\mathcal{X}_i, \boldsymbol{Z})$. In other words, the approximation is enabled by removing the dependency between any pairs of function values that are not in the same image.

We then propose a novel likelihood (noise) model for multiple instance learning using heteroscedastic noisy modeling, where the noise variance varies across image samples. We first need to link all the function values $\{f_{ij}\}_{j=1}^{N_i}$ from a image to the corresponding target label y_i. We will use their mean value to aggregate them into a single value, $\overline{f}_i$, and design a likelihood function comparing that value and the corresponding target label:

$$\overline{f}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} f_{ij}. \tag{5}$$

In fact, this definition can be considered as an implementation of the collective multi-instance assumption (*8, 12, 18, 32*) (where all the instances in a bag are equally contribute to the target label), rather than the standard multi-instance assumption, i.e., taking the maximum value among the instances in a bag. We hypothesis that the collective multi-instance assumption (5) is more appropriate to TBM prediction than the standard multi-instance assumption is because genetic alteration might affect the overall scanned tumor regions and thus just a single image block cannot be a dominant representative of all the others, from which we predict the TMB level of a patient sample. In addition, in contrast to the standard multi-instance assumption (using max operation), the definition (5) leads a more efficient inference algorithm which is discussed in the sequel (Section 3.3). The likelihood function could be simply defined as (please recall our definition of the latent function $f(\cdot)$):

$$\widetilde{p}(y_i|\overline{f}_i) = \begin{cases} 1 & \text{if } b_{y_i-1} < \overline{f}_i \leq b_{y_i} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Due to noise from inputs or target labels, however, we consider a noisy version of the ideal (noise free) likelihood function (6):

$$p(y_i|\overline{f}_i) = \int \widetilde{p}(y_i = k|\overline{f}_i + \epsilon)\mathcal{N}(\epsilon|0, \sigma^2)d\epsilon \tag{7}$$

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

where we have assumed that the estimator $\overline{f}_i$ is contaminated by a Gaussian noise with zero mean and unknown variance σ^2.

We further extend the likelihood function (7) to the case where the noise variance σ^2 can vary across image samples, leading to a heteroscedastic noise model. We might have different levels of confidence in predicting TMB from different images. For example, we would have high confidence in prediction if the patterns across all the image blocks in an image are consistent. However, our confidence in prediction would be low if there is high diversity in the patterns across all the image blocks. Thus the noise variance in (7) can represent the level of this confidence in prediction a set of image blocks in the MIL setting. As shown in Figure 2, our main idea is to make our method put different levels of effort to classify each image correctly according to the confidence level. In other words, we make our method try to fit easier image samples well but not to put much effort in classifying harder (ambiguous) images correctly. We further assume that the noise variance can be determined from the distribution of each feature among the image blocks in an image. We define an image level feature vector $\boldsymbol{x}_i^*$ for each image, which contains statistics of each feature in an image:

$$\boldsymbol{x}_i^* = [\mathrm{mean}(\mathcal{X}_i)^\top, \mathrm{std}(\mathcal{X}_i)^\top, \mathrm{skew}(\mathcal{X}_i)^\top, \mathrm{kurt}(\mathcal{X}_i)^\top]^\top \tag{8}$$

where the mean, standard deviation, skewness and kurtosis of each feature are calculated separately and thus $\boldsymbol{x}_i^* \in \mathbb{R}^{4D}$. Assuming the noise variance at the ith image be a function of the statistical features $\boldsymbol{x}_i^*$, our heteroscedastic noise model can be defined as

$$p(y_i|\overline{f}_i, \boldsymbol{x}_i^*) = \int \widetilde{p}(y_i|\overline{f}_i + \epsilon)\mathcal{N}(\epsilon|0, \sigma^2(\boldsymbol{x}_i^*))d\epsilon. \tag{9}$$

Finally we assume that the noise variance $\sigma^2(\boldsymbol{x}_i^*) = \exp\{g(\boldsymbol{x}_i^*)\}$ is modeled by another Gaussian process:

$$g(\boldsymbol{x}_i^*) \quad \sim \quad \mathcal{GP}(0, \kappa_g(\boldsymbol{x}_i^*, \cdot)) \tag{10}$$

where κ_g is also assumed to be a ARD SE covariance function. The statistical feature $\boldsymbol{x}_i^*$ can be considered similar to the privileged information in (*15*). However, unlike the setting in (*15*), the statistical features can be always calculated from a test image.

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

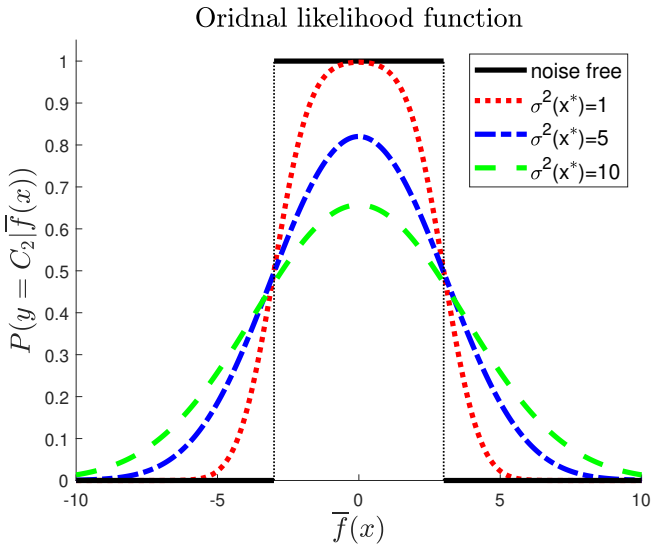

Figure 2: The likelihood function (9) with different levels of the noise variance. As the noise variance increases, the likelihood is reduced in overall over $\overline{f}_i$. Controlling this noise variance, we can make our prediction model put different levels of effort to classify each image correctly.

As a summary, our predication model of TMB from images with the heteroscedastic noise modeling is given as follows:

$$p(y_i|\overline{f}_i, g_i) = \Phi\left(\frac{b_{y_i} - \overline{f}_i}{\exp\{g(\boldsymbol{x}_i^*)\}}\right) - \Phi\left(\frac{b_{y_i-1} - \overline{f}_i}{\exp\{g(\boldsymbol{x}_i^*)\}}\right) \tag{11}$$

$$p(\overline{f}_i|\boldsymbol{u}) = \mathcal{N}(\overline{f}_i|m_i, \upsilon_i), \tag{12}$$

$$p(\boldsymbol{u}|\boldsymbol{Z}) = \mathcal{N}(\boldsymbol{u}|0, \boldsymbol{K}_{uu}), \quad p(\boldsymbol{g}|\boldsymbol{X}^*) = \mathcal{N}(\boldsymbol{g}|0, \boldsymbol{K}_g) \tag{13}$$

where

$$m_i = \frac{1}{N_i}\mathbf{1}_{N_i}^\top\left(\boldsymbol{K}_{f_i u}\boldsymbol{K}_{uu}^{-1}\boldsymbol{u}\right), \tag{14}$$

$$\upsilon_i = \frac{1}{N_i^2}\mathbf{1}_{N_i}^\top\left(\boldsymbol{K}_{f_i f_i} - \boldsymbol{K}_{uf_i}^\top\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf_i}\right)\mathbf{1}_{N_i}, \tag{15}$$

$g_i \triangleq g(\boldsymbol{x}_i^*)$ and $\boldsymbol{g} = [g_1, g_2, ..., g_N]^\top$. In fact, we have included the variable $\overline{f}_i$ in the model for readers' easy and clear understanding, but it can be easily marginalized out from the model.

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

3.3 Inference with EP

The inference of the model involves calculating the posterior distribution over the latent variables, $\boldsymbol{u}$ and $\boldsymbol{g}$, and estimating the hyper-parameters (e.g., scale length parameters in the ARD covariances). The posterior over $\boldsymbol{u}$ and $\boldsymbol{g}$ can be calculate as follows:

$$\begin{aligned} p(\boldsymbol{u},\boldsymbol{g}|\boldsymbol{Y}) &= \frac{\int p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{g})p_{\text{PITC}}(\boldsymbol{f}|\boldsymbol{u})d\boldsymbol{f}p(\boldsymbol{u}|\boldsymbol{Z})p(\boldsymbol{g}|\boldsymbol{X}^*)}{p(\boldsymbol{y}|\boldsymbol{Z},\boldsymbol{X}^*)} \\ &= \frac{1}{p(\boldsymbol{y}|\boldsymbol{Z},\boldsymbol{X}^*)}\left(\prod_{i=1}^{N} t_i(\boldsymbol{u},\boldsymbol{g})\right)p(\boldsymbol{u})p(\boldsymbol{g}) \end{aligned} \tag{16}$$

where

$$p(\boldsymbol{y}|\boldsymbol{f},\boldsymbol{g}) = \prod_{i=1}^{N} p(y_i|\overline{f}_i, g_i), \tag{17}$$

$$\begin{aligned} t_i(\boldsymbol{u},g_i) &= \int p(y_i|\overline{f}_i,g_i)\mathcal{N}(\overline{f}_i|m_i,\upsilon_i)d\overline{f}_i \\ &= \Phi\left(\frac{b_{y_i}-m_i}{\sqrt{\upsilon_i+\exp\{g_i\}}}\right)-\Phi\left(\frac{b_{y_i-1}-m_i}{\sqrt{\upsilon_i+\exp\{g_i\}}}\right), \end{aligned} \tag{18}$$

and the marginal likelihood $p(\boldsymbol{y}|\boldsymbol{Z},\boldsymbol{X}^*)$ ($\triangleq \mathcal{Z}$) can be used to find the optimal hyper-parameters ($\boldsymbol{\vartheta}$) by maximizing $\mathcal{Z}$ w.r.t. $\boldsymbol{\vartheta}$.

We use expectation and propagation (EP) to approximate the posterior distribution (16) as the calculations involved in (16) are intractable. EP enables analytical computation of the posterior distribution by approximating each non-Gaussian likelihood $t_i(\boldsymbol{u},g_i)$ with a non-normalized Gaussian factor $\widetilde{t}_i(\boldsymbol{u},g_i)$, defined as follows

$$\widetilde{t}_i(\boldsymbol{u},g_i) = \tilde{s}_i \exp\left\{-\frac{\tilde{\tau}_i^u}{2}\boldsymbol{u}^\top(\boldsymbol{\omega}_i\boldsymbol{\omega}_i^\top)\boldsymbol{u}+\tilde{\mu}_i^u\boldsymbol{u}^\top\boldsymbol{\omega}_i\right\}\exp\left\{-\frac{\tilde{\tau}_i^g}{2}\left(g_i-\tilde{\mu}_i^g\right)^2\right\}, \tag{19}$$

where $\boldsymbol{\omega}_i = \frac{1}{N_i}\left(\boldsymbol{K}_{uu}^{-1}\boldsymbol{K}_{uf_i}\mathbf{1}_{N_i}\right)$. Then the approximate posterior distribution over $\boldsymbol{u}$ and $\boldsymbol{g}$, $q(\boldsymbol{u},\boldsymbol{g})$, can be calculated as follows

$$\begin{aligned} q(\boldsymbol{u},\boldsymbol{g}) &= \frac{1}{\mathcal{Z}_{\text{EP}}}\left(\prod_{i=1}^{N}\tilde{t}_i(\boldsymbol{u},g_i)\right)p(\boldsymbol{u})p(\boldsymbol{g}) \\ &= \mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}^u,\boldsymbol{\Sigma}^u)\mathcal{N}(\boldsymbol{g}|\boldsymbol{\mu}^g,\boldsymbol{\Sigma}^g) \end{aligned} \tag{20}$$

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

where

$$\boldsymbol{\Sigma}^u = \left(\boldsymbol{K}_{uu}^{-1} + \boldsymbol{\Omega}\boldsymbol{T}^u\boldsymbol{\Omega}^\top\right)^{-1}, \tag{21}$$

$$\boldsymbol{\mu}^u = \boldsymbol{\Sigma}^u\boldsymbol{\Omega}\tilde{\boldsymbol{\mu}}^u, \tag{22}$$

$$\boldsymbol{\Sigma}^g = \left(\boldsymbol{\Gamma}^{-1} + \boldsymbol{T}^g\right)^{-1}, \tag{23}$$

$$\boldsymbol{\mu}^g = \boldsymbol{\Sigma}^g\boldsymbol{T}^g\tilde{\boldsymbol{\mu}}^g, \tag{24}$$

and $\mathcal{Z}_{\text{EP}}$ is the EP approximation to the marginal likelihood $\mathcal{Z}$.

In the standard EP, each factor $\widetilde{t}_i$ ($\triangleq \widetilde{t}_i(\boldsymbol{u}, \boldsymbol{g})$) is iteratively updated until convergence in a way that it becomes the most accurate in regions of high posterior probability (*2*). The update rule is as follows. *i*) We first calculate the cavity distribution $q_{\setminus i}$ by removing $\widetilde{t}_i$ from the approximate posterior distribution q ($\triangleq q(\boldsymbol{u}, \boldsymbol{g})$), i.e., $q_{\setminus i} \propto q/\tilde{t}_i$. *ii*) Then we find a new distribution $\widehat{q}$ that minimizes the Kullback–Leibler divergence between $\mathcal{Z}_i^{-1} q_{\setminus i} t_i(\boldsymbol{u}, g_i)$ and q, i.e., $\widehat{q} = \arg\min_q \text{KL}\big[\mathcal{Z}_i^{-1} q_{\setminus i} t_i(\boldsymbol{u}, g_i) || q\big]$, where $\mathcal{Z}_i$ is a normalization constant. Since q is assumed to be Gaussian, this minimization can be done by matching the expected sufficient statistics (mean and covariance) of q with those of the distribution $\mathcal{Z}_i^{-1} q_{\setminus i} t_i(\boldsymbol{u}, g_i)$. These expectation can be calculated using the derivative of $\log \mathcal{Z}_i$ w.r.t. the natural parameter of $q_{\setminus i}$ (*26*). However, due to our heteroscedastic noise modeling, $\log \mathcal{Z}_i$ and its derivations cannot be calculated in closed form. However, fortunately, all the intractable calculations involve just an one-dimensional numerical integration (w.r.t. g_i) which can be efficiently computed in different ways, e.g., use Gauss-Hermite quadrature. *iii*) Finally the factor $\widetilde{t}_i$ can be updated by $\widetilde{t}_i \propto \widehat{q}/q_{\setminus i}$. At convergence, the approximate posterior distribution can be calculated using all the factors $\{\widetilde{t}_i\}$ and the prior distributions as in (20). In addition, the approximate marginal likelihood also can be calculated from these factors:

$$\log \mathcal{Z}_{\text{EP}} = \left(h^u(\boldsymbol{\theta}^u) - h^u(\boldsymbol{\theta}^u_{\text{prior}})\right) + \left(h^g(\boldsymbol{\theta}^g) - h^g(\boldsymbol{\theta}^g_{\text{prior}})\right) + \sum_{i=1}^N \log \widetilde{\mathcal{Z}}_i \tag{25}$$

where

$$\log \widetilde{\mathcal{Z}}_i = \log \mathcal{Z}_i + \left(h^u(\boldsymbol{\theta}^u_{\setminus i}) - h^u(\boldsymbol{\theta}^u)\right) + \left(\boldsymbol{\theta}^g_{\setminus i}) - h^g(\boldsymbol{\theta}^g)\right)$$

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

where $\boldsymbol{\theta}^u$, $\boldsymbol{\theta}^u_{\setminus i}$ and $\boldsymbol{\theta}^u_{\text{prior}}$ are the natural parameters of the marginal distributions, q, $q_{\setminus i}$ and $p(\boldsymbol{U}|\boldsymbol{Z})$, respectively ($q$ is assumed to be factorized into two independent distributions of $\boldsymbol{u}$ and $\boldsymbol{g}$, and so is $q_{\setminus i}$); and $h^u(\boldsymbol{\theta}^u)$ is the log partition function of a Gaussian distribution with natural parameter $\boldsymbol{\theta}^u$ (*26*). We can further optimize the hyper-parameters ($\boldsymbol{\vartheta}$) using the gradient of the log approximate marginal likelihood $\log Z_{EP}$ w.r.t $\boldsymbol{\vartheta}$. Please see Appendix for the detailed information about the EP algorithm for our model.

For computational efficiency, we follow the approach proposed (a variant of the standard EP algorithm) in (*14*). Instead of waiting until the EP updating converges, not only the approximate factors $\{\widetilde{t}_i\}$ but also the hyper-parameters $\boldsymbol{\vartheta}$ are updated at the same time. We use the same (batch) gradient ascent method in (*14*) to update the hyper-parameters, where each parameter has its own learning rate which is adaptively determined according to the sign change of the gradient between two consecutive iterations.

Finally, the predictive distribution of a new test image sample (e.g., a bag $\mathcal{X}_o$ with $\boldsymbol{x}^*_o$) can be approximately calculated using the posterior distribution $q(\boldsymbol{u}, \boldsymbol{g})$ obtained from the EP algorithm:

$$\begin{aligned} & p(y_o = k|\mathcal{X}_o, \boldsymbol{x}^*_o, \boldsymbol{Z}) \\ \approx & \int p(y_o = k|\overline{f}_o, g(\boldsymbol{x}^*_o))p(\overline{f}_o|\boldsymbol{u})p(g(\boldsymbol{x}^*_o)|\boldsymbol{g})q(\boldsymbol{u}, \boldsymbol{g})d\boldsymbol{u}d\boldsymbol{g} \end{aligned} \tag{26}$$

where the distribution $p(\overline{f}_o|\boldsymbol{u})$ is given by

$$p(\overline{f}_o|\boldsymbol{u}) = \mathcal{N}(\overline{f}_o|m_o, \nu_o) \tag{27}$$

where it mean and the variance are as follows

$$m_o = \frac{1}{N_o}\mathbf{1}^\top_{N_i}\big(\boldsymbol{K}_{f_o u}\boldsymbol{K}^{-1}_{uu}\big)\boldsymbol{u} = \boldsymbol{\omega}^\top_o \boldsymbol{u}, \tag{28}$$

$$\nu_o = \frac{1}{N_o^2}\mathbf{1}^\top_{N_o}\Big(\boldsymbol{K}_{f_o f_o} - \boldsymbol{K}^\top_{uf_o}\boldsymbol{K}^{-1}_{uu}\boldsymbol{K}_{uf_o}\Big)\mathbf{1}_{N_o}, . \tag{29}$$

where $\boldsymbol{K}_{f_o f_o} = \kappa_f(\mathcal{X}_o, \mathcal{X}_o)$, $\boldsymbol{K}_{uf_o} = \kappa_f(\boldsymbol{Z}, \mathcal{X}_o)$ and $\mathbf{1}_{N_o} \in \mathbb{R}^{N_o}$ is a length N_o (the number of image

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

blocks in the new image) of one vector. As a result, the predictive distribution can be calculated as

$$p(y_o = k|\mathcal{X}_o, \boldsymbol{x}_o^*, \boldsymbol{Z}) \approx \int \left(\Phi\Big(\frac{b_k - a_o}{\sqrt{d_o + \exp\{g(\boldsymbol{x}_o^*)\}}}\Big) - \Phi\Big(\frac{b_{k-1} - a_o}{\sqrt{d_o + \exp\{g(\boldsymbol{x}_o^*)\}}}\Big) \right) \mathcal{N}(g(\boldsymbol{x}_o^*)|\mu_o^g, \sigma_o^g) dg(\boldsymbol{x}_o^*). \quad (30)$$

where

$$a_o = \boldsymbol{\omega}_0^\top \boldsymbol{\mu}^u \quad (31)$$

$$d_o = \nu_0 + \boldsymbol{\omega}_0^\top \boldsymbol{\Sigma}^u \boldsymbol{\omega}_0 \quad (32)$$

$$\mu_o^g = (\boldsymbol{k}_o^*)^\top (\boldsymbol{K}^*)^{-1} \boldsymbol{\mu}^g \quad (33)$$

$$\sigma_o^g = \kappa_g(\boldsymbol{x}_o^*, \boldsymbol{x}_o^*) - (\boldsymbol{k}_o^*)^\top (\boldsymbol{K}^*)^{-1} \boldsymbol{k}_o^* + (\boldsymbol{k}_o^*)^\top (\boldsymbol{K}^*)^{-1} \boldsymbol{\Sigma}^g (\boldsymbol{K}^*)^{-1} \boldsymbol{k}_o^* \quad (34)$$

where $\boldsymbol{k}_o^* = [\kappa_g(\boldsymbol{x}_o^*, \boldsymbol{x}_1^*), \kappa_g(\boldsymbol{x}_o^*, \boldsymbol{x}_2^*), ..., \kappa_g(\boldsymbol{x}_o^*, \boldsymbol{x}_N^*)]^\top$. Note that this integral also does not have a close form solution. We can use again Gaussian quadrature to numerically calculate this integration.

4 Numerical experimental results

We investigated the performance of our prediction model on synthetic and real-world data sets. We compared our methods with the multiclass SVM method on whole slide images proposed in (*31*) (hereafter MSVM) and multiple instance ordinal regression (MIOR) (*29*). In fact, although MSVM is not a MIL method, we included this method in our experiments because the method is also based on whole slide images and the way the method combines multiple feature vectors in an image into a single vector is related to our noise modeling. MSVM also calculates the statistics of each feature, such as mean, standard deviation, skewness and kurtosis, among image blocks in an image and use these values as a new feature vector to the image which will be fed to the classifier (note that this multiclass classifier is defined by binary decomposition method, e.g., all-pair comparisons). In other words, the method uses image level features $\boldsymbol{x}_i^*$ as input to the classifier. Regarding MIOR, there have been proposed very few MIL methods for ordinal regression problems. As its name suggests MIOR solves ordinal regression in the MIL setting. Note that, it is not straightforward to directly extend the standard MIL assumption

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

to multi-class or ordinal regression problems. Thus, the method proposes its own strategy how to link multiple instances in a bag to the corresponding label. For each bag, only the instance which is nearest to the center between two corresponding hyperplanes given by the class label is selected to be compared with the label (*29*). All methods, including ours, were implemented in Matlab (version info: Matlab R2018a). For MSVM, we used LibSVM (*4*) as a base binary classifier (we only considered the linear kernel for the SVM because the nonlinear kernel, i.e., a rbf kernel, showed worse performance).

4.1 Synthetic data

With this synthetic data set, we tested the effectiveness of our heteroscedastic noise modeling. To do this, we considered the reduced version of our method (the Gaussian process based MIL prediction model but without the heteroscedastic noise modeling), i.e. , we implemented the same Gaussian process prediction model with the likelihood function with a fixed noise variance (7), instead of the original likelihood function (9). We refer to this reduced version of our method as **GPMIL** and call our method with the heteroscedastic noise modeling by **GPMIL-HN**. Both methods were trained based on the frame of EP approach. For the case of training GPMIL, the EP algorithm becomes more simplified because it involves neither the additional Gaussian process which models the noise variance from image level features $\{\boldsymbol{x}_i^*\}$ nor any numerical integration, e.g., Gauss-Hermite quadrature, as in the case of learning GPMIL-HN.

To evaluate the performance of the methods, we calculated the multi-class prediction accuracy, which is given by

$$\frac{1}{N_t}\sum_{i=1}^{N_t} 1[Y_i = \hat{Y}_i], \tag{35}$$

where N_t is the number of the total test samples, $\hat{Y}_i$ is the estimated label for the ith sample and the operator 1[expr] is 1 if the expression, expr, is true or 0 otherwise. To compare the performance of the methods, we used random K-fold cross-validations where the data points (bags in our case) are randomly divided into K-folds and 1 fold is used to test the prediction performance of the model and the rest folds are used to train the model. For MSVM and MIOR, the training data (bags in the $K-1$ folds) are

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

again randomly divided into training and validation sets to select the optimal hyper-parameters, e.g., the regularization constant. However, for our methods all the training data can be used to train the model and to optimize the hyper-parameters, such as ARD coefficients and inducing points, because our methods provide approximate marginal likelihood from which we can optimize these hyper-parameters.

4.1.1 Dataset

We generated a 2 dimensional three class MIL data set, where each point (instances) in a sample (bag) is generated from one of three Gaussian distributions or the mixture of them. Please see Figure 3, where a square with different colors represents the mean of each Gaussian distribution (mean: $C_1 = [-3, -3]^\top$, $C_2 = [-0.5, -3]^\top$, $C_3 = [3, 3]^\top$ and covariance matrix: $C_1 = [1, 0.2;\ 0.2, 1]$, $C_2 = [1, 0.5;\ 0.5, 1]$, $C_3 = [1, 0;\ 0, 1]$). Note that these mean vectors and covariance matrices were chosen arbitrarily. We generated 100 bags from each class and made each bag to have 100 instances. We assumed that there were two types of samples: easy and hard samples. For the case of easy samples, all instances in a bag were sampled from a single Gaussian distribution corresponding to the label to which the bag belongs. On the other hand, for the case of hard samples, instances in a bag were sampled from a mixture of the three Gaussian distributions with the mixing coefficients $[0.4, 0.3, 0.3]$. We made sure that the mixing coefficient for the corresponding Gaussian distribution to the class to which a bag belongs was always the largest, i.e., it was set to 0.4, so that the majority of instances in a bag still came from the corresponding class. Finally we introduce the noise rate which is the portion of hard sample in the total samples, i.e., if the noise rate is set to 0.5, then it means that 50 bags among 100 bags in each class are hard samples.

4.1.2 Performance evaluation

We evaluated the performance of the methods with increasing the noise rate. We repeated random 5-fold cross validation 5 times at each noise rate and report the results in Figure (4). Each line represents the mean performance of each method and each error bar is the standard deviation calculated from the repetition at each noise level. We first note from the figure that MSVM is not able to handle the presence

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

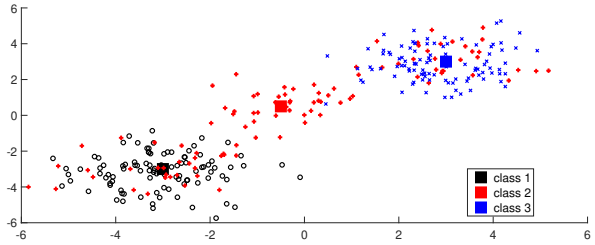

Figure 3: 2-D 3 class synthetic data. Each square in different colors represents the mean vector of each Gaussian distribution (each class) and the symbols, (x, o, +) are instances in class 1, 2 and 3, respectively. Here, we show just three samples (bags) from each class as examples: i.e., two easy samples, black (class 1) and blue (class 3), and one hard sample (red). The instances in the red bag are spread across the areas where samples from the other classes mainly reside.

of noise in the MIL setting. Its performance is almost the same as that of a random classifier. The figure also showed that MIOR worked better than MSVM but its performance decreased as the noise rate increased. It might indicate that MIOR is sensitive to noise. It might be because the method selects only a single instance from each bag to train the model. It is more likely that an irrelevant instance to a bag can be selected as a training example in the presence of highly noisy data. However, our methods, both GPMIL-HL and GPMIL, showed almost stable performance across all the range of the noise rates. It might be because our methods are based on the collective MIL assumption which use all instances in a bag to train the model, unlike the assumption used in MIOR or the standard MIL assumption. We also confirmed that there is a gain caused by our heteroscedastic noise modeling from the comparison between GPMIL-HN and GPMIL (without the heteroscedastic noise modeling).

4.2 TMB prediction

4.2.1 Dataset

A cohort of 386 bladder cancer patients (and corresponding clinical information) with 457 diagnostic H&E stained whole slide images were downloaded from TCGA data portal. Among 386 patient samples, 24 patients were excluded due to poor quality of their images, such as undesired pen marks, torn tissues or other artifacts. According to the percentile of total number single nucleotide variants (*24*), the remaining

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

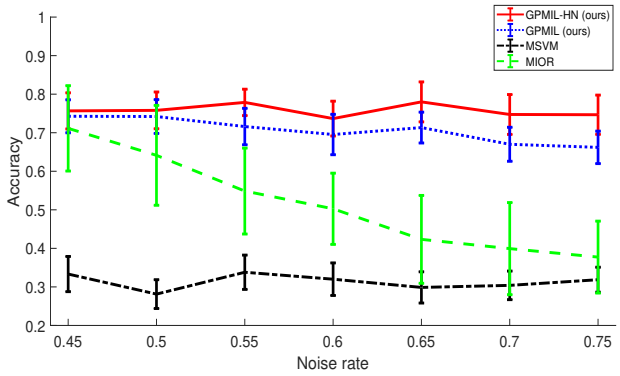

Figure 4: The accuracy of each method evaluated with increasing the noise rate. Each line and error bar represent the mean performance and the corresponding standard deviation, respectively, of each method evaluated by 5-repetition of random 5-fold cross validation.

362 patient samples had been categorized into 3 groups: 119 low TMB patients, 123 medium TMB patients and 120 high TMB patients. Although a small number of patients includes more than 1 diagnostic slides, for simplicity this study only uses the first diagnostic slide (i.e., with DX1 suffix) of those patients. Therefore, the dataset for evaluating TMB prediction consists of 362 different patients of whole slides images, with each image scanned at 20× or 40× magnification and having a size from hundreds of megabytes to several gigabytes.

4.2.2 Image processing pipeline

Since the whole slide image has a large volume size and is difficult to be processed directly, we use a multi-resolution analysis pipeline to extract features from the image. Fig. illustrates our image processing pipeline, which includes two main modules: tumor detection and feature extraction. In the tumor detection module, the whole slide image is processed at a coarse level (e.g., 5× in this study). First, the whole slide image is divided into a number of non-overlapping 256×256 image blocks (see overlapped red and green blocks in the image). Next, local binary patterns (LBP) texture features are computed from each image block. The blocks that belong to tumor regions (see green blocks in the image) are then detected by using a trained SVM classifier (*30*). In the feature extraction module, the detected tumor

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

blocks are analyzed at a fine level (e.g., 10× in this study). To extract fine level features, we make use of transfer learning on the Resnet50 model (*13*), which was pre-trained on approximately 1.28 million image from ImageNet dataset. We replace the output layer of Resnet50 model with a global average pooling layer, and hence the Resnet50 model finally outputs a 2048 dimensional feature vector for each input tumor block. Finally the dimensional of input features was further reduced to 50 by principal component analysis (PCA) since these first 50 principal components explain 97% of variance.

4.2.3 Performance evaluation

We evaluated the performance of each method again with random 5-fold cross-validation. We repeated each experiment 5 times. Beside the prediction accuracy, we computed F_1 measure, which is the Harmonic mean of precision and recall, i.e., $\mathrm{F}_1 = \frac{2 \cdot \text{presion} \cdot \text{recall}}{\text{presion} + \text{recall}}$ of each class under the assumption that the class is supposed to be the positive class and the other classes negative. Calculating F_1 measures, we can examine the performance of each method with respect to a specific class.

Table 1 shows the multi-class prediction accuracy of each method. Although, the overall performance of all the method is not that impressive, comparing GPMIL-HN and GPMIL, we can see a clear improvement due to the heteroscedastic noise modeling. In fact, GPMIL did not work at all for the TMB prediction as the method classified all images as *intermediate* class as shown in Table 2. Our method also showed better performance than MSVM and MIOR. In particular, MIOR also did not work for the TMB prediction. It classified most all test images as the same class 20 times during the experiments, where the total number of evaluations was 25 (= 5 repetition × 5 folds). Please see the results of MIOR in Table 2. As a result, we confirmed from the TMB prediction on the real-world data set (TCGA bladder cancer whole slide images) that the usefulness of our heteroscedastic noise modeling.

5 Conclusion

In this study we develop a computational prediction method of the TMB level of patients from whole slide images, where the prediction problem is formulated as a MIL problem and the heteroscedastic noise

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Table 1: The multi-class prediction accuracy. We report the mean performance of each method. The number in the parentheses represents the standard deviation.

Method	Accuracy
GPMIL-HN(ours)	**0.429** (0.055)
GPMIL (ours)	0.339 (0.018)
MSVM	0.346 (0.053)
MIOR	0.341 (0.022)

Table 2: The F1 measures. We report the mean performance of each method and its standard deviation in the parentheses. The symbol '-' means that the value is non-available. This is happen when the method does not classify any test image samples as the corresponding class in any-fold experiment during 5-fold cross-validation (5 repetition).

Method	(TMB) Low	Intermediate	High
GPMIL-HN (ours)	0.429 (0.084)	0.466 (0.044)	0.364 (0.097)
GPMIL (ours)	-	0.504 (0.017)	-
MSVM	0.196 (0.098)	0.391 (0.062)	0.405 (0.090)
MIOR	-	-	-

model is used to model the noise variance from image level features. In particular, this noise variance encodes the confidence that our method has regarding image samples. The method trusts image samples that are easier (having the higher confidence level or the lower noise variance) but less relies on image samples that are hard (having the lower confidence level or the higher noise variance). Then the posterior distributions and the model hyper-parameters are efficiently learned by the EP algorithm. We finally have demonstrated from the synthetic and TCGA bladder cancer whole slide image data sets the usefulness of our heteroscedastic noise modeling over the other baseline methods.

In comparison with MIOR, which is also an ordinal regression method in the MIL setting, our method has several advantages. As we have seen from the experimental results on the synthetic data, MIOR is more sensitive to noisy instances presented in a bag than our method is. It would be mainly because of its selective assumption, i.e., only the instance which is nearest to the center between two hyperplanes given by the class label is selected to train the model (*29*). In addition, our method is a Bayesian approach and thus would be more robust against overfitting than MIOR (which is implemented by regularized loss minimization) is. Furthermore, the marginal likelihood allows us to use more complex covariance

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

functions, corresponding to the kernel function used in in MIOR. With using of ARD covariance functions we can automatically reduce the influence of features that are irrelevant to the prediction task. We also can optimize the parameters of in ARD covariance functions. This advanced adjustment is not available with MIOR.

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Appendix

Experiment setting The hyper-parameters in our method include inducing points $\boldsymbol{Z}$ and model-parameters for the covariance functions, κ_f and κ_g. In all our experiments, the number of inducing points, M, was set to 100. Each inducing point was optimized by maximizing the marginal likelihood in the framework of EP. They were initialized by randomly selected feature vectors from the training data. In addition, we set b_1 and b_2 to 0 and 2 respectively. In fact, these values also can be optimized. However, we did not find significant improvement from optimizing of these values. For MSVM, the regularized constant was selected among $[2^{-3}; 2^{-2}; 2^{-1}; 2^0; 2^1; 2^2; 2^3; 10; 2^4]$ by cross-validation. For MIOR there are two regularized constants. These values were also chosen by CV. The search space was 2-dimensional grid space and the possible candidates for each parameter were set to the same set, $[2^{-2}; 2^{-1}; 2^0; 2^1; 2^2; 2^3]$.

Detailed derivations The detailed steps for the EP in Section 3.3 are as follows.

i) The cavity distribution $q_{\backslash i} \propto q/\tilde{t}_i$ can be calculated as

$$q_{\backslash i} = \mathcal{N}\Big(\boldsymbol{u}\big|\boldsymbol{\mu}^u_{\backslash i}, \boldsymbol{\Sigma}^u_{\backslash i}\Big)\mathcal{N}\Big(g_i\big|\mu^g_{\backslash i}, \sigma^g_{\backslash i}\Big), \tag{36}$$

where

$$\begin{aligned}
\boldsymbol{\Sigma}^u_{\backslash i} &= \Big(\big(\boldsymbol{\Sigma}^u\big)^{-1} - \tilde{\tau}^u_i\big(\boldsymbol{\omega}_i\boldsymbol{\omega}_i^\top\big)\Big)^{-1} = \boldsymbol{\Sigma}^u + \Big((\tilde{\tau}^u_i)^{-1} - \boldsymbol{\omega}_i^\top\boldsymbol{\Sigma}^u\boldsymbol{\omega}_i\Big)^{-1}\boldsymbol{\Sigma}^u\boldsymbol{\omega}_i\boldsymbol{\omega}_i^\top\boldsymbol{\Sigma}^u, && (37)\\
\boldsymbol{\mu}^u_{\backslash i} &= \boldsymbol{\Sigma}^u_{\backslash i}\Big(\big(\boldsymbol{\Sigma}^u\big)^{-1}\boldsymbol{\mu}^u - \tilde{\mu}^u_i\boldsymbol{\omega}_i\Big) = \boldsymbol{\mu}^u + \Big(\tilde{\tau}^u_i\big(\boldsymbol{\omega}_i^\top\boldsymbol{\mu}^u\big) - \tilde{\mu}^u_i\Big)\boldsymbol{\Sigma}^u_{\backslash i}\boldsymbol{\omega}_i, && (38)\\
\sigma^g_{\backslash i} &= \Big((\sigma^g_i)^{-1} - \tilde{\tau}^g_i\Big)^{-1}, && (39)\\
\mu^g_{\backslash i} &= \sigma^g_{\backslash i}\Big((\sigma^g_i)^{-1}\mu^g_i - \tilde{\tau}^g_i\tilde{\mu}^g_i\Big). && (40)
\end{aligned}$$

ii) We now compute the distribution $\widehat{q}$:

$$\widehat{q} = \mathcal{N}\Big(\boldsymbol{u}\big|\widehat{\boldsymbol{\mu}}^u, \widehat{\boldsymbol{\Sigma}}^u\Big)\mathcal{N}\Big(g_i\big|\widehat{\mu}^g_i, \widehat{\sigma}^g_i\Big). \tag{41}$$

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

To do this, we first need to calculate $\mathcal{Z}_i$, i.e., the normalization constant of $t(\boldsymbol{u}, g_i)q_{\backslash i}$,:

$$\begin{aligned}\mathcal{Z}_i &= \int t_i(\boldsymbol{u}, g_i)\mathcal{N}\Big(\boldsymbol{u}|\boldsymbol{\mu}^u_{\backslash i}, \boldsymbol{\Sigma}^u_{\backslash i}\Big)\mathcal{N}\Big(g_i|\mu^g_{\backslash i}, \sigma^g_{\backslash i}\Big)dg_i \\ &= \int \left(\Phi\Big(\frac{b_{y_i} - a_i}{\sqrt{d_i + \exp\{g_i\}}}\Big) - \Phi\Big(\frac{b_{y_i-1} - a_i}{\sqrt{d_i + \exp\{g_i\}}}\Big)\right)\mathcal{N}\Big(g_i|\mu^g_{\backslash i}, \sigma^g_{\backslash i}\Big)dg_i, \qquad (42)\end{aligned}$$

where

$$a_i = \frac{1}{N_i}\mathbf{1}^\top_{N_i}\boldsymbol{K}_{f_i u}\boldsymbol{K}^{-1}_{uu}\boldsymbol{\mu}^u_{\backslash i} = \boldsymbol{\omega}^\top_i\boldsymbol{\mu}^u_{\backslash i}, \qquad (43)$$

$$\begin{aligned}d_i &= \frac{1}{N_i^2}\mathbf{1}^\top_{N_i}\Big(\boldsymbol{K}_{f_i,f_i} - \boldsymbol{K}^\top_{u,f_i}\boldsymbol{K}^{-1}_{uu}\boldsymbol{K}_{u,f_i}\Big)\mathbf{1}_{N_i} + \boldsymbol{\omega}^\top_i\boldsymbol{\Sigma}^u_{\backslash i}\boldsymbol{\omega}_i \\ &= \nu_i + \boldsymbol{\omega}^\top_i\boldsymbol{\Sigma}^u_{\backslash i}\boldsymbol{\omega}_i. \qquad (44)\end{aligned}$$

Note that (42) does not admit a close-form solution and thus we need numerical approximation methods, e.g., Gauss-Hermite quadrature. Then, the distribution $\widehat{q}$ can be computed from the derivative of $\log \mathcal{Z}_i$ with respect to the parameters of $q_{\backslash i}$. The mean ($\widehat{\boldsymbol{\mu}}^u$) and covariance matrix ($\widehat{\boldsymbol{\Sigma}}^u$) of the distribution $\widehat{q}$ is given by

$$\widehat{\boldsymbol{\mu}}^u = \boldsymbol{\mu}^u_{\backslash i} + \boldsymbol{\Sigma}^u_{\backslash i}\frac{\partial \log \mathcal{Z}_i}{\partial \boldsymbol{\mu}^u_{\backslash i}} = \boldsymbol{\mu}^u_{\backslash i} - \rho_i\boldsymbol{\Sigma}^u_{\backslash i}\boldsymbol{\omega}_i \qquad (45)$$

$$\begin{aligned}\widehat{\boldsymbol{\Sigma}}^u &= \boldsymbol{\Sigma}^u_{\backslash i} - \boldsymbol{\Sigma}^u_{\backslash i}\left(\Big(\frac{\partial \log \mathcal{Z}_i}{\partial \boldsymbol{\mu}^u_{\backslash i}}\Big)\Big(\frac{\partial \log \mathcal{Z}_i}{\partial \boldsymbol{\mu}^u_{\backslash i}}\Big)^\top - 2\frac{\partial \log \mathcal{Z}_i}{\partial \boldsymbol{\Sigma}^u_{\backslash i}}\right)\boldsymbol{\Sigma}^u_{\backslash i} \\ &= \boldsymbol{\Sigma}^u_{\backslash i} - (\rho_i^2 + \varphi_i)\boldsymbol{\Sigma}^u_{\backslash i}\boldsymbol{\omega}_i\boldsymbol{\omega}^\top_i\boldsymbol{\Sigma}^u_{\backslash i} \qquad (46)\end{aligned}$$

where

$$\rho_i = \frac{1}{\mathcal{Z}_i}\int\Big(\mathcal{N}(b_{y_i}|a_i, \tilde{d}^g_i) - \mathcal{N}(b_{y_i-1}|a_i, \tilde{d}^g_i)\Big)\mathcal{N}(g_i|\mu^g_{\backslash i}, \sigma^g_{\backslash i})dg_i \qquad (47)$$

$$\varphi_i = \frac{1}{\mathcal{Z}_i}\int\Big(\frac{b_{yi} - a_i}{\tilde{d}^g_i}\mathcal{N}(b_{y_i}|a_i, \tilde{d}^g_i) - \frac{b_{yi-1} - a_i}{\tilde{d}^g_i}\mathcal{N}(b_{y_i-1}|a_i, \tilde{d}^g_i)\Big)\mathcal{N}(g_i|\mu^g_{\backslash i}, \sigma^g_{\backslash i})dg_i \qquad (48)$$

where $\tilde{d}^g_i = d_i + \exp\{g_i\}$. Then, $\widehat{\mu}^g_i$ and $\widehat{\sigma}^g_i$ are computed by

$$\widehat{\mu}^g_i = E_{\frac{1}{\mathcal{Z}_i}t'_i(g_i)q_{\backslash i}(g_i)}[g_i] \qquad (49)$$

$$\widehat{\sigma}^g_i = E_{\frac{1}{\mathcal{Z}_i}t'_i(g_i)q_{\backslash i}(g_i)}[g_i^2] - (\widehat{\mu}^g_i)^2, \qquad (50)$$

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

Where $t_i^{'}(g_i) = \int t_i(\boldsymbol{u}, g_i)\mathcal{N}(\boldsymbol{u}|\boldsymbol{\mu}^u_{\setminus i}, \boldsymbol{\Sigma}^u_{\setminus i})d\boldsymbol{u}$ and these computations can be done by numerical methods, e.g., Gauss-Hermite quadrature.

iii) The factor $\widetilde{t}_i$ can be updated by $\widetilde{t}_i \propto \widehat{q}/q_{\setminus i}$.

$$\begin{aligned}\widetilde{t}_i(\boldsymbol{u}, g_i) &= \tilde{s}_i \exp\left\{-\frac{\tilde{\tau}_i^u}{2}\boldsymbol{u}^\top(\boldsymbol{\omega}_i\boldsymbol{\omega}_i^\top)\boldsymbol{u} + \tilde{\mu}_i^u\boldsymbol{u}^\top\boldsymbol{\omega}_i\right\}\cdot \\ &\quad \exp\left\{-\frac{\tilde{\tau}_i^g}{2}\left(g_i - \tilde{\mu}_i^g\right)^2\right\}, \end{aligned} \tag{51}$$

For the latent variable $\boldsymbol{u}$, the updated factor is given by

$$\left(\tilde{\boldsymbol{\Sigma}}_i^{\,u}\right)^{-1} = (\widehat{\boldsymbol{\Sigma}}^u)^{-1} - (\boldsymbol{\Sigma}^u_{\setminus i})^{-1} = \tilde{\tau}_i^u\boldsymbol{\omega}_i\boldsymbol{\omega}_i^\top, \tag{52}$$

$$\tilde{\boldsymbol{\mu}}_i^{\,u} = (\widehat{\boldsymbol{\Sigma}}^u)^{-1}\widehat{\boldsymbol{\mu}}^u - (\boldsymbol{\Sigma}^u_{\setminus i})^{-1}\boldsymbol{\mu}^u_{\setminus i} = \tilde{\mu}_i^u\boldsymbol{\omega}_i, \tag{53}$$

where

$$\tilde{\tau}_i^u = \left((\rho_i^2 + \varphi_i)^{-1} - \boldsymbol{\omega}_i\boldsymbol{\Sigma}^u_{\setminus i}\boldsymbol{\omega}_i^\top\right)^{-1}, \tag{54}$$

$$\tilde{\mu}_i^u = -\rho_i + \tilde{\tau}_i^u a_i - \rho_i\tilde{\tau}_i^u\left(\boldsymbol{\omega}_i^\top\boldsymbol{\Sigma}^u_{\setminus i}\boldsymbol{\omega}_i\right). \tag{55}$$

For the latent variable g_i, the updated factor is given by

$$\tilde{\tau}_i^g = (\widehat{\sigma}_i^g)^{-1} - (\sigma^g_{\setminus i})^{-1}, \tag{56}$$

$$\tilde{\mu}_i^g = (\tilde{\tau}_i^g)^{-1}\left((\widehat{\sigma}_i^g)^{-1}\widehat{\mu}_i^g - (\sigma^g_{\setminus i})^{-1}\mu^g_{\setminus i}\right) \tag{57}$$

The approximate marginal likelihood $\mathcal{Z}_{\mathrm{EP}} = p(\boldsymbol{y}|\boldsymbol{Z}, \boldsymbol{X}^*)$ can be calculated as

$$\mathcal{Z}_{\mathrm{EP}} = \left(h^u(\boldsymbol{\theta}^u) - h^u(\boldsymbol{\theta}^u_{\mathrm{prior}})\right) + \left(h^g(\boldsymbol{\theta}^g) - h^g(\boldsymbol{\theta}^g_{\mathrm{prior}})\right) + \sum_{i=1}^N \log\widetilde{\mathcal{Z}}_i \tag{58}$$

where

$$\log\widetilde{\mathcal{Z}}_i = \log\mathcal{Z}_i + \left(h^u(\boldsymbol{\theta}^u_{\setminus i}) - h^u(\boldsymbol{\theta}^u)\right) + \left(h^g\left(\mu^g_{\setminus i}, \sigma^g_{\setminus i}\right) - h^g\left(\mu_i^g, \sigma_i^g\right)\right) \tag{59}$$

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

With the definition of the log-normalizer of a Gaussian distribution, $\mathcal{Z}_{\text{EP}}$ can be computed as

$$\begin{aligned}\log \mathcal{Z}_{\text{EP}} &= \frac{1}{2}\Big(\log|\boldsymbol{\Sigma}^u| + \big((\boldsymbol{\mu}^u)^\top(\boldsymbol{\Sigma}^u)^{-1}\boldsymbol{\mu}^u\big) - \log|\boldsymbol{K}_{uu}|\Big) \\ &+ \frac{1}{2}\Big(\log|\boldsymbol{\Sigma}^g| + \big((\boldsymbol{\mu}^g)^\top(\boldsymbol{\Sigma}^g)^{-1}\boldsymbol{\mu}^g\big) - \log|\boldsymbol{K}^*|\Big) \\ &+ \sum_{i=1}^{N}\log\widetilde{\mathcal{Z}}_i. \end{aligned} \tag{60}$$

The gradient of $\mathcal{Z}_{\text{EP}}$ with respect to a hyper parameter, ϑ, in the covariance function κ_f can be computed as

$$\frac{\partial \log \mathcal{Z}_q}{\partial \vartheta} = \Big(\boldsymbol{\eta}^u - \boldsymbol{\eta}^u_{\text{prior}}\Big)^\top \frac{\boldsymbol{\theta}^u_{\text{prior}}}{\partial \vartheta} + \sum_{i=1}^{N}\frac{\partial \log \mathcal{Z}_i}{\partial \vartheta} \tag{61}$$

where

$$\Big(\boldsymbol{\eta}^u - \boldsymbol{\eta}^u_{\text{prior}}\Big)^\top \frac{\boldsymbol{\theta}^u}{\partial \vartheta} = -\frac{1}{2}\text{tr}\left\{\Big(\boldsymbol{K}_{uu}^{-1} - \boldsymbol{K}_{uu}^{-1}\Big(\boldsymbol{\Sigma}^u + \boldsymbol{\mu}^u(\boldsymbol{\mu}^u)^\top\Big)\boldsymbol{K}_{uu}^{-1}\Big)^\top \frac{\boldsymbol{\theta}^u}{\partial \vartheta}\right\}, \tag{62}$$

and

$$\begin{aligned}\frac{\partial \log \mathcal{Z}_i}{\partial \vartheta} &= \rho_i\left[\text{tr}\left\{\Big(\frac{\partial a_i}{\partial \boldsymbol{K}_{f_i u}}\Big)^\top \frac{\partial \boldsymbol{K}_{f_i u}}{\partial \vartheta}\right\} + \text{tr}\left\{\Big(\frac{\partial a_i}{\partial \boldsymbol{K}_{uu}}\Big)^\top \frac{\partial \boldsymbol{K}_{uu}}{\partial \vartheta}\right\}\right] \\ &+ \varphi_i\left[\text{tr}\left\{\Big(\frac{\partial d_i}{\partial \boldsymbol{K}_{f_i u}}\Big)^\top \frac{\partial \boldsymbol{K}_{f_i u}}{\partial \vartheta}\right\} + \text{tr}\left\{\Big(\frac{\partial d_i}{\partial \boldsymbol{K}_{uu}}\Big)^\top \frac{\partial \boldsymbol{K}_{uu}}{\partial \vartheta}\right\}\right]\end{aligned} \tag{63}$$

The gradient of $\mathcal{Z}_{\text{EP}}$ with respect to a hyper parameter in the covariance function κ_g, can be further simplified (since $\frac{\partial \log \mathcal{Z}_i}{\partial \vartheta} = 0$, $\forall$i in this case):

$$\begin{aligned}\frac{\partial \log \mathcal{Z}_g}{\partial \vartheta} &= \Big(\boldsymbol{\eta}^g - \boldsymbol{\eta}^g_{\text{prior}}\Big)^\top \frac{\boldsymbol{\theta}^g_{\text{prior}}}{\partial \vartheta} \\ &= -\frac{1}{2}\text{tr}\left\{\Big((\boldsymbol{K}^*)^{-1} - (\boldsymbol{K}^*)^{-1}\Big(\boldsymbol{\Sigma}^g + \boldsymbol{\mu}^g(\boldsymbol{\mu}^g)^\top\Big)(\boldsymbol{K}^*)^{-1}\Big)^\top \frac{\boldsymbol{\theta}^u}{\partial \vartheta}\right\}\end{aligned} \tag{64}$$

References and Notes

1. Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. Support vector machines for multiple-instance learning. In *Proceedings of the 15th International Conference on Neural Informa-*

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

tion Processing Systems, Advances in Neural Information Processing Systems (NIPS), pages 577–584, 2002.

2. Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006.

3. R. L. Brown, S. D. nad Warren, E. A. Gibb, S. D. Martin, J. J. Spinelli, B. H. Nelson, and R. A. Holt. Neo-antigens predicted by tumor genome meta-analysis correlate with increased patient survival. *Genome research*, 25:743–50, 2014.

4. C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machines, 2001.

5. Wei Chu and Zoubin Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019.

6. Nicolas Coudray, Paolo Santiago Ocampo, Theodore Sakellaropoulos, Navneet Narula, Matija Snuderl, David Fenyö, Andre L. Moreira, and Narges Razavian.

7. Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1):31 – 71, 1997.

8. James Foulds and Eibe Frank. A review of multi-instance learning assumptions. *The Knowledge Engineering Review*, 25(1):1–25, 2010.

9. Paul W. Goldberg, Christopher K. I. Williams, and Christopher Bishop. Regression with input-dependent noise: A gaussian process treatment, 1998.

10. Aaron M. Goodman, Shumei Kato, Lyudmila Bazhenova, Sandip P. Patel, Garrett M. Frampton, Vincent Miller, Philip J. Stephens, Gregory A. Daniels, and Razelle Kurzrock. Tumor mutational burden as an independent predictor of response to immunotherapy in diverse cancers. *Molecular Cancer Therapeutics*, 16(11):2598–2608, 2017.

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

11. M. Haußmann, F. A. Hamprecht, and M. Kandemir. Variational bayesian multiple instance learning with gaussian processes. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 810–819, 2017.

12. Jianjun He, Hong Gu, and Zhelong Wang. Bayesian multi-instance multi-label learning using gaussian process prior. *Machine Learning*, 88(1):273–295, Jul 2012.

13. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

14. Daniel Hernandez-Lobato and Jose Miguel Hernandez-Lobato. Scalable gaussian process classification via expectation propagation. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 51 of *Proceedings of Machine Learning Research*, pages 168–176, Cadiz, Spain, May 2016. PMLR.

15. Daniel Hernández-lobato, Viktoriia Sharmanska, Kristian Kersting, Christoph H Lampert, and Novi Quadrianto. Mind the nuisance: Gaussian process classification using privileged noise. In *Advances in Neural Information Processing Systems 27*.

16. Le Hou, Dimitris Samaras, Tahsin M. Kurç, Yi Gao, James E. Davis, and Joel H. Saltz. Patch-based convolutional neural network for whole slide tissue image classification. *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2424–2433, 2016.

17. Maximilian Ilse, Jakub M. Tomczak, and Max Welling. Attention-based deep multiple instance learning. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2018.

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

18. Melih Kandemir, Manuel Haussmann, Ferran Diego, Kumar Rajamani, Jeroen Van Der Laak, and Fred Hamprecht. Variational weakly supervised gaussian processes. In *British Machine Vision Conference*, York, UK, 2016.

19. Minyoung Kim and Fernando Torre. Multiple instance learning via gaussian processes. *Data Mining and Knowledge Discovery*, 28(4):1078–1106, 2014.

20. Miguel Lázaro-Gredilla and Michalis K. Titsias. Variational heteroscedastic gaussian process regression. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 841–848, 2011.

21. N. Quadrianto, K. Kersting, M. D. Reid, T. S. Caetano, and W. L. Buntine. Kernel conditional quantile estimation via reduction revisited. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 938–943, 2009.

22. Joaquin Quiñonero Candela and Carl Edward Rasmussen. A unifying view of sparse approximate gaussian process regression. *J. Mach. Learn. Res.*, 6:1939–1959, December 2005.

23. Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

24. A Gordon Robertson, Jaegil Kim, Hikmat Al-Ahmadie, Joaquim Bellmunt, Guangwu Guo, Andrew D Cherniack, Toshinori Hinoue, Peter W Laird, Katherine A Hoadley, Rehan Akbani, et al. Comprehensive molecular characterization of muscle-invasive bladder cancer. *Cell*, 171(3):540–556, 2017.

25. Andrew J Schaumberg, Mark A Rubin, and Thomas J Fuchs. H&e-stained whole slide image deep learning predicts spop mutation state in prostate cancer. *bioRxiv*, 2018.

26. Matthias Seeger. Expectation propagation for exponential families. Technical report, 2005.

bioRxiv preprint doi: https://doi.org/10.1101/554261; this version posted February 18, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder. All rights reserved. No reuse allowed without permission.

27. Ville Tolvanen, Pasi Jylänki, and Aki Vehtari. Expectation propagation for nonstationary heteroscedastic gaussian process regression. *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2014.

28. Nils Weidmann, Eibe Frank, and Bernhard Pfahringer. A two-level learning method for generalized multi-instance problems. In *Proceedings of the European Conference on Machine Learning (ECML)*, pages 468–479, Berlin, Heidelberg, 2003.

29. Y. Xiao, B. Liu, and Z. Hao. Multiple-instance ordinal regression. *IEEE Transactions on Neural Networks and Learning Systems*, 29(9):4398–4413, 2018.

30. Hongming Xu, Cheng Lu, Richard Berendt, Naresh Jha, and Mrinal Mandal. Automated analysis and classification of melanocytic tumor on skin whole slide images. *Computerized Medical Imaging and Graphics*, 66:124–134, 2018.

31. Hongming Xu, Sunho Park, and Tae Hyun Hwang. Automatic classification of prostate cancer gleason scores from digitized whole slide tissue biopsies. In *17th International Workshop on Data Mining in Bioinformatics*, 2018.

32. Xin Xu. Statistical learning in multiple instance problems. Master's thesis, University of Waikato, Hamilton, New Zealand, 2003.