Variational Bayesian Multiple Instance Learning with Gaussian Processes

Manuel Haußmann[1] Fred A. Hamprecht[1] Melih Kandemir[1,2*]
[1]HCI/IWR, Heidelberg University [2]Özyeğin University
{manuel.haussmann, fred.hamprecht}@iwr.uni-heidelberg.de
melih.kandemir@ozyegin.edu.tr

Abstract

Gaussian Processes (GPs) are effective Bayesian predictors. We here show for the first time that instance labels of a GP classifier can be inferred in the multiple instance learning (MIL) setting using variational Bayes. We achieve this via a new construction of the bag likelihood that assumes a large value if the instance predictions obey the MIL constraints and a small value otherwise. This construction lets us derive the update rules for the variational parameters analytically, assuring both scalable learning and fast convergence. We observe this model to improve the state of the art in instance label prediction from bag-level supervision in the 20 Newsgroups benchmark, as well as in Barrett's cancer tumor localization from histopathology tissue microarray images. Furthermore, we introduce a novel pipeline for weakly supervised object detection naturally complemented with our model, which improves the state of the art on the PASCAL VOC 2007 and 2012 data sets. Last but not least, the performance of our model can be further boosted up using mixed supervision: a combination of weak (bag) and strong (instance) labels.

1. Introduction

Recent years have seen a tremendous increase in our ability to collect ever larger data sets automatically at ever decreasing costs. This has further widened the gulf between our data collection and labeling capacities. Weakly supervised learning has emerged as an active area of machine learning to bridge this gap. It targets learning effective predictors from minimal annotator effort. Among the many weakly supervised learning approaches, multiple instance learning (MIL) [1] stands out as an excellent match to computer vision. MIL assumes that the training data is partitioned into groups of instances, called *bags*, and labels are available only at the level of entire groups. A bag is given a positive label if at least one of its instances contain the target pattern, and a negative label if none of its instances contain it. The difficulty of this setting arises from the fact that the labels of individual instances in positive bags are not known at training time. The MIL model, hence, needs to account for this missing information. MIL has been shown to be greatly beneficial in image categorization [3, 7, 27].

Gaussian Processes (GPs) [33] are commanding much attention of the machine learning community due to their high potential in supervised learning. They are able to fit complex non-linear decision boundaries thanks to their inherent kernelization. The high expressive power of GPs can also be understood from their proven equivalence to a multilayer perceptron with infinitely many hidden neurons [32]. The probabilistic nature of GPs allows them to handle uncertainty in a principled manner [10].

MIL is a supervised learning task with missing instance labels. The uncertainty in these latent variables makes GP modeling a natural fit. Even so, there has only been limited work on GP-based approaches to MIL models to date. Kim and Torre [23] were the first to use GPs in the MIL setting, by applying the softmax approximation to a Bernoulli likelihood and performing inference using Laplace's method. This approach suffers from two limitations: i) it does not scale to large data sets due to the inversion of one Hessian matrix per bag in each iteration, ii) the learned posterior is not accurate since both softmax and Laplace's method are not tight approximations of the true modeling assumptions. Recent work by Kandemir *et al*. [21] alleviates these issues by relaxing the MIL assumption (by allowing a fraction of positive predictions in negative bags) and performing variational inference. Although this approach yields a very accurate bag-level predictor, it cannot predict instance labels as the MIL assumption is violated during training.

We introduce the first adaptation of GPs to MIL that affords variational inference and instance label prediction. Furthermore, our construction allows learning the variational parameters by closed-form updates, resulting in fast convergence. We achieve this tractability by a new bag-level likelihood formulation that ensures consistency be-

[*] The main part of this work has been done while the author was with Heidelberg Collaboratory for Image Processing (HCI), Heidelberg University.

tween instance-level and bag-level predictions of the model. We further extend this model to a large-margin setting, which forces the immediate neighborhood of the decision boundary to remain sparse. We observe this extension to perform better in tasks with high overlap between classes.

Our model improves the state of the art in three applications: (i) categorization of postings in the 20 Newsgroups data set, a standard benchmark for instance label prediction with MIL, (ii) detection of Barrett's cancer tumors from histopathology tissue microarrays, and (iii) object detection from natural images in the PASCAL VOC 2007 and 2012 data sets. We outperform existing approaches in PASCAL VOC thanks to a novel processing pipeline where our GP-based MIL model takes its place naturally. The source code of our model is publicly available[1].

2. Related Work

Among existing GP-based models [21, 23], none has targeted the instance label prediction problem. However, there does exist a whole range of alternative approaches. Liu *et al.* [30] use a k-nearest-neighbour based approach, referred to as the *voting framework* (VF/VFr). Li *et al.* [28] and Wang *et al.* [44] use different SVM based models. Kandemir and Hamprecht [20] combine a MIL likelihood with two Dirichlet Process mixture models, one per class (DP-MIL). Kotzias *et al.* [24] introduce the *Group-Instance Cost Function* (GICF), a special objective function to encourage smoothness between the instance labels and a method that relies on convolutional neural networks (CNNs) for text data to get higher level features for the instances.

Object detection (predicting the object type and locating its bounding box) can be interpreted as a MIL problem when existence of the target pattern is known only at the image level. Treating each image as a bag and patches extracted from that image as its instances fits exactly into the MIL setup. In the fully-supervised setting, object detection has seen huge improvements in recent years with the work by Girshick *et al.* [12, 13] on using CNNs to create higher order features for region proposals. These region proposals are typically generated by off-the-shelf algorithms, such as Selective Search [42], EdgeBoxes [49], Binarized normed gradients [6] or AttractioNet [11]. More recent work treats the region proposal generation as an integral part of the pipeline that can be trained end-to-end [34, 35].

There has been a growing interest in weakly supervised object detecion in the recent years. The seminal work of Cinbis *et al.* [7] get region proposals from selective search, compute CNN and Fisher vectors on top of them, and use a multi-fold MIL approach for final prediction. Wang *et al.* [43] use probabilistic Latent Semantic Analysis to learn latent categories for their region proposals, which they obtain again via selective search and represent by higher order CNN features from a pretrained network. Bilen *et al.*'s WSDDN [3] follows the fast R-CNN approach more closely, modifying its architecture to two streams, one for detection, the other for classification. Kantorov *et al.* [22] extend this two-stream approach by exploiting the context around region proposals. Following the observation that weakly supervised localization algorithms often have more difficulty with smaller objects than larger ones, Shi and Ferrari [37] use a curriculum learning [2] approach that iteratively sorts its proposed objects by size and learned weights. Similarly to our approach, they rely on a neural network mostly to generate high level features and train a classifier—in their case an SVM—on the output. Li *et al.* [27] also split the training process into two steps, focusing first on image level classification and then adapting their net progressively to detection. We discuss how our suggested pipeline relates to the existing weakly supervised detection approaches in Section 3.3.

2.1. Notation

The data set $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^N$, consisting of N instances $x_n \in \mathbb{R}^d$ and their unobserved binary labels $y_n \in \{0, 1\}$, is partitioned into B non-overlapping bags, with label $T_b \in \{0, 1\}$ for each bag b. We denote with $\{y_i\}_b := \{y_i | i \in \text{Bag } b\}$ the instance labels in bag b. The MIL assumption is then that $T_b = \max\{y_i\}_b$, *i.e.* the bag label is positive if at least one instance label is positive and zero otherwise. $\{y_i\}_{b-n}$ is the collection of all labels in bag b except for instance label y_n.

$\mathcal{N}(\cdot|\mu, \Sigma)$ and $\mathcal{B}er(\cdot|\pi)$ denote the Normal and Bernoulli distribution, respectively. $\langle X \rangle_{p(X)}$ is the expectation of X with respect to distribution $p(X)$ shortened to $\langle X \rangle$. The Gram matrix between two data sets $X = \{x_1, ..., x_N\} \in \mathbb{R}^{N \times d}$ and $Z = \{z_1, ..., z_M\} \in \mathbb{R}^{M \times d}$ is denoted as $K_{XZ} \in \mathbb{R}^{N \times M}$, with $(K_{XZ})_{ij} = k(x_i, z_j)$. We use the RBF kernel function $k(x_i, z_j) = \exp(-(x_i - z_j)^\top (x_i - z_j)/2l^2)$ throughout the experiments and keep the length-scale l fixed to $\sqrt{d}$. Finally, $\text{diag}(\cdot)$ returns a square diagonal matrix with the values of the input vector on the diagonal (or the diagonal values for a matrix input).

3. Variational Bayes for Gaussian Processes under MIL

For a given data set $X = \{x_1, ..., x_N\}$ and corresponding labels $y = \{y_1, ..., y_N\}$, GP classification [33] is given by

$$f|X \sim \mathcal{N}(f|0, K_{XX}), \tag{1}$$

$$y|f \sim \prod_{n=1}^N \mathcal{B}er(y_n|\sigma(f_n)). \tag{2}$$

[1] https://github.com/manuelhaussmann/vgpmil

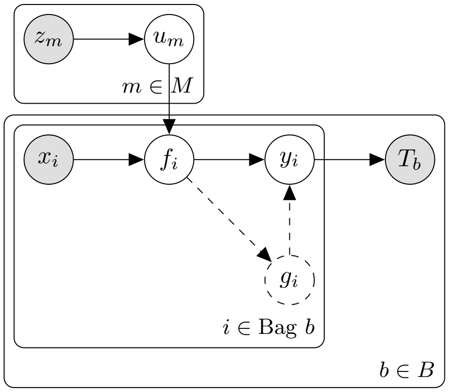


Figure 1: Plate diagram for VGPMIL. Observed variables are represented by grey circles and latent variables by white circles. Dashed arrows and circles show the additional interactions introduced by the LM-VGPMIL extension.

This model places a GP prior over the decision margins f, squeezes them by a logistic sigmoid function[2] to the unit interval, and feeds the outcome to a Bernoulli mass function as the mean parameter. The sign of f determines the class and its magnitude how confident the prediction is. This full GP model is limited to small data sets due to the necessity of inverting the kernel matrix K_{XX} at the cost of $\mathcal{O}(N^3)$. This cost is alleviated by sparsifying the GP following the *fully independent training conditional (FITC)* approximation [39], which introduces a set of inducing points $Z = \{z_1, ..., z_M\}$ and corresponding output u, mirroring the relation between X and f, so that u and f are jointly normal distributed

$$f|X, Z, u \sim \mathcal{N}(f|K_{XZ}K_{ZZ}^{-1}u, \mathcal{K}), \quad u|Z \sim \mathcal{N}(0, K_{ZZ}),$$

where $\mathcal{K} := \operatorname{diag}(K_{XX} - K_{XZ}K_{ZZ}^{-1}K_{ZX})$. FITC reduces the cost to $\mathcal{O}(M^2N)$, where $M \ll N$ is a design parameter.

In the MIL setting, we have only bag-level labels. GPMIL of Kim and Torre [23] adapts the GP classifier to this setting by $p(T_b|\{f_i\}_b) = \mathcal{B}er(T_b|\sigma(\max\{f_i\}_b))$. Approximating the indifferentiable $\max$ with softmax: $\max\{f_i\}_b \approx \log\left(\sum_i \exp(f_i)\right)$, they propose

$$f|X \sim \mathcal{N}(f|0, K_{XX}),$$
$$T|f \sim \prod_{b=1}^{B} 1\Big/\Big(1 + \Big(\sum_{i \in \text{Bag } b} e^{f_i}\Big)^{-T_b}\Big).$$

They infer this model using Laplace's method, which unfortunately involves calculating and inverting an $N \times N$ Hessian matrix in each iteration. Even though this matrix is block-diagonal, it consists of $N_b \times N_b$ non-zero blocks for each bag of size N_b, limiting scalability. Furthermore, the prediction performance suffers from two coarse approximations: i) softmax, which diverges from the exact $\max$ when the values are evenly distributed, ii) Laplace's method, which approximates a potentially multimodal posterior with a single mode.

We take an alternative approach and directly represent the (latent) binary instance labels. One of our core contributions is the following parametrization of the bag label likelihood

$$p(T_b|\{y_i\}_b) = \left(\tfrac{H}{H+1}\right)^{G_b}\left(\tfrac{1}{H+1}\right)^{1-G_b} = \tfrac{H^{G_b}}{H+1}, \tag{3}$$

with $G_b := T_b \max\{y_i\}_b + (1 - T_b)(1 - \max\{y_i\}_b)$ and a positive constant H. G_b equals one if the MIL constraint ($T_b = \max\{y_i\}_b$) is fulfilled and zero otherwise. To those states Equation 3 assigns a high probability and acts as a *noisy* version of the MIL assumption, with the level of noise being controlled by H, becoming exact as H approaches infinity. A reasonably large H (*e.g.* 100) works well in practice. Sparsifying the GP prior for scalability, the model is

$$u|Z \sim \mathcal{N}(u|0, K_{ZZ}), \tag{4}$$
$$f|X, Z, u \sim \mathcal{N}\left(f|K_{XZ}K_{ZZ}^{-1}u, \mathcal{K}\right), \tag{5}$$
$$y|f \sim \prod_{n=1}^{N} \mathcal{B}er\big(y_n|\sigma(f_n)\big), \tag{6}$$
$$T|y \sim \prod_{b=1}^{B} \frac{H^{G_b}}{H+1}, \tag{7}$$

the plate diagram of which is illustrated in Figure 1. We refer to this model as *Variational Gaussian Process Multiple Instance Learning (VGPMIL)*. We will show in the next subsection that this model can be trained efficiently with closed form updates using variational inference, avoiding the necessity of gradient descent and hence the need for tuning a learning rate.

3.1. Inference

Using variational inference, we aim to approximate the intractable posterior $p(y, f, u|T, X)$ by a variational distribution $Q = q(u)p(f|u)\prod_n q(y_n)$ (with simplified notation $q(y_n) := q_n(y_n)$). That is, we introduce variational distributions over u and the instance labels y_n. Consequently, the inference problem is reformulated as the following optimization problem

$$\arg\min_{Q} \mathrm{KL}\big(Q||p(y, f, u|T, X, Z)\big). \tag{8}$$

This KL divergence can be rearranged as

$$\log p(T|X) = \mathrm{KL}\big(Q||p(y, f, u|T, X, Z)\big) + \langle \log p(y, f, u, T|X)\rangle - \langle \log Q\rangle,$$

where the last two terms are jointly known as the evidence lower bound (ELBO). Since $\mathrm{KL}(q||p) \geq 0$, $\forall q$ and the

[2] $\sigma(a) = 1/(1 + e^{-a})$

marginal likelihood $p(T|X)$ forms an upper bound that is independent of Q, minimizing the KL divergence is equivalent to maximizing the ELBO. This minimization can be achieved by updating each of the factors of Q to its optimum, keeping the others fixed. One can show that (see *e.g.* [4] for details) the optimal update for a factor $\tilde{q}$ of Q is[3]

$$\log \tilde{q} \leftarrow \langle \log p(T, y, f, u) \rangle_{Q \setminus \tilde{q}} + \text{const}, \tag{9}$$

where $\langle \cdot \rangle_{Q \setminus \tilde{q}}$ refers to the expectation of $\log p$ with respect to the variational distribution Q except for $\tilde{q}$, the current factor being updated, and the constant term can be determined by calculating the normalizing constant for $\tilde{q}$. In our case this translates to finding updates for $q(u)$ and $q(y_n)$. Note that the factorization of the variational distribution is the only assumption we make regarding these factors. The actual distributional forms they eventually take are fully determined by the update rules.

We handle the intractable combination of the Bernoulli mass and the sigmoid function in its mean parameter in Equation 6 with the Jaakkola bound [17]

$$\sigma(x) \geq \sigma(\xi) \exp\left(\frac{x - \xi}{2} - \lambda(\xi)(x^2 - \xi^2) \right), \tag{10}$$

where $\lambda(\xi) = \frac{1}{2\xi}\left(\sigma(\xi) - \frac{1}{2}\right)$. This bound has been used for variational inference of GPs for the first time by Kandemir *et al.* [21] and gives us

$$\mathcal{B}er(y_n|\sigma(f_n)) \geq \exp\left(-\tfrac{f_n + \xi_n}{2} - \lambda(\xi_n)(f_n^2 - \xi_n^2) \right) \\ \exp(y_n f_n)\sigma(\xi_n),$$

introducing a new variational parameter ξ_n for each y_n.

Updating $q(u)$. Equation 9 gives us $q(u) = \mathcal{N}(u|m, S)$

$$\text{with} \quad S \leftarrow \left(K_{ZZ}^{-1} K_{ZX} \Lambda K_{XZ} K_{ZZ}^{-1} + K_{ZZ}^{-1} \right)^{-1} \tag{11}$$

$$m \leftarrow S K_{ZZ}^{-1} K_{ZX} \left(\langle y \rangle - \tfrac{1}{2} \right) \tag{12}$$

where $\Lambda := 2\text{diag}\big((\lambda(\xi_1), ..., \lambda(\xi_N))\big)$. The update for the variational parameters is $\xi_n^2 \leftarrow \langle f_n^2 \rangle$, *i.e.*

$$\xi_n^2 \leftarrow K_{x_n Z} K_{ZZ}^{-1} (m m^\top + S) K_{ZZ}^{-1} K_{Z x_n} + \mathcal{K}_{nn},$$

where $\mathcal{K}_{nn}$ refers to the variance of $p(f|u)$ in Equation 5.

Updating $q(y_n)$. For the updates for the variational distributions of the instance labels, we need to deal with the $\max\{y_i\}_b$ operator, which is neither differentiable nor does it allow analytical updates. In order to update an individual instance y_n, we decompose the max as

$$\max\{y_i\}_b = y_n + \max\{y_i\}_{b-n} - y_n \max\{y_i\}_{b-n}, \tag{13}$$

which holds since $y_i \in \{0, 1\}$. Following Equation 9 again, we get the update rule $q(y_n) = \mathcal{B}er(y_n|\pi_n)$ with[4]

$$\pi_n \leftarrow \sigma\Big(\langle f_n \rangle + \log H \cdot \big(2T_b + \langle \max\{y_i\}_{b-n} \rangle \\ - 2T_b \langle \max\{y_i\}_{b-n} \rangle - 1\big) \Big), \tag{14}$$

where $\langle f_n \rangle = \langle f_n \rangle_{p(f|u)q(u)} = K_{x_n Z} K_{ZZ}^{-1} m$. The first term in the sigmoid contains the information of the current instance given by the sparse GP, while the second consists of two factors. The first applies a penalty of $\log H$, the size and sign of which is controlled by the second factor, that checks the MIL constraint. Consider the two possible cases

$$T_b = 1: \quad \pi_n \leftarrow \sigma\Big(\langle f_n \rangle + \log H \cdot \big(1 - \langle \max\{y_i\}_{b-n} \rangle\big) \Big),$$

$$T_b = 0: \quad \pi_n \leftarrow \sigma\Big(\langle f_n \rangle + \log H \cdot \big(\langle \max\{y_i\}_{b-n} \rangle - 1\big) \Big).$$

By approximating $\langle \max\{y_i\}_{b-n} \rangle$ with $\max\{\langle y_i \rangle\}_{b-n}$, the model picks the largest expected value of the instance labels in bag b if y_n were not part of that bag. For a positive bag, if the model predicts existence of at least one positive instance in this bag, it will drag the second term towards zero, indicating that the MIL constraint is fulfilled. Hence, the expected label for y_n depends only on its local evidence $\langle f_n \rangle$. If, however, the other instances are all predicted to be negative, the model uses the global (bag-level) evidence and $\log H$ will push instance y_n strongly towards the positive side, overwhelming local evidence. On the other hand, for a negative bag, $\max\{\langle y_i \rangle\}_{b-n}$ will be close to zero, giving $\log H$ a negative sign. This time the model forces π_n towards zero and to once more satisfy the MIL constraint.

Mixed strong/weak supervision. A nice property of our model is that it can directly combine weak and strong supervision by simply fixing the corresponding variational distributions for these instances. As it is already known for negative bags that all their instances are negative, full supervision is always given for negative bag instances in the MIL setting. However, as shown in Equation 14, full supervision is helpful for positive bags. Furthermore, mixed supervision is a more ecologically valid scenario for many real-world applications, as we can always support a weakly supervised data set by a small portion of fully supervised observations with acceptable additional effort. A plausible MIL model should benefit maximally from this support.

3.2. The large margin version

Although GPs are flexible learners, they have an inherent regularization mechanism making the model immune to overfitting [36]. Nevertheless, extremely high clutter

[3] We omit the conditioning on X, Z from the notation.

[4] See the Supplement for detailed derivations of the update rules and the decomposition in Equation 13.

of classes in challenging applications could destabilize the model and confuse it by the noisy variations around the decision margin. To overcome this, we introduce further margin control similarly to that of SVMs. It forces the model to favour solutions that keeps the close neighborhood of the margin as empty as possible. We achieve this by replacing the simple Bernoulli distribution for instance labels $y_n|f_n \sim \mathcal{B}er(y_n|\sigma(f_n))$ (Equation 6) by

$$g_n|f_n \sim \mathcal{B}er\big(g_n|\sigma(C(|f_n| - V))\big), \tag{15}$$

$$y_n|f_n, g_n \sim \mathcal{B}er\big(y_n|\sigma(f_n g_n)\big). \tag{16}$$

The gating distribution introduced in Equation 15 determines how confident the prediction on y_n is. The parameters V and C tune the degree of regularization we prefer to employ on the margin. C determines how strongly the model will penalize margin violations in a similar spirit to the C parameter of SVMs, by regulating how close the sigmoid becomes to a step function. By shifting the sigmoid V controls the margin we wish to enforce. A shift with $V = 2$ can be interpreted as requiring the model to be 88% sure of the instance prediction.[5] Its output g_n forces the model in Equation 16 to refrain from making decisions if a prediction is dangerously close to the decision boundary. g_n serves as a gatekeeper that decides whether to let f_n pass to Equation 16. If the model is certain enough, $\mathcal{B}er(y_n|\sigma(f_n g_n)) \approx \mathcal{B}er(y_n|\sigma(f_n))$ as in the main model, otherwise $\mathcal{B}er(y_n|\sigma(f_n g_n)) \approx \mathcal{B}er(y_n|0.5)$. This way, the model forces uncertain probabilities towards the decision boundary, discarding them from the active set, as a prediction on the decision boundary is effectively ignored in a probabilistic model. This creates a large margin around the boundary. After the modifications discussed above, the large-margin variant of our model becomes (see Figure 1)

$$u \sim \mathcal{N}(u|0, K_{ZZ}), \tag{17}$$

$$f|u \sim \mathcal{N}(f|K_{XZ}K_{ZZ}^{-1}u, \mathcal{K}), \tag{18}$$

$$g|f \sim \prod_{n=1}^{N} \mathcal{B}er\big(g_n|\sigma(C(|f_n| - V))\big), \tag{19}$$

$$y|f, g \sim \prod_{n=1}^{N} \mathcal{B}er\big(y_n|\sigma(f_n g_n)\big), \tag{20}$$

$$T|y \sim \prod_{b=1}^{B} \frac{H^{G_b}}{H+1}, \tag{21}$$

which we refer to as the *Large-Margin VGPMIL (LM-VGPMIL)*. We enforce large margins only during training. We predict the instance labels on test bags for both of our models identically to the basic GP classifier.

Given the structural similarity to the VGPMIL model, it can be learned with closed-form updates via variational inference as well. Our variational distribution in this case is given by $Q = q(u)p(f|u)\prod_n q(y_n)q(g_n)$. The update rules for each $q(\cdot)$ are given in the Supplement[6]. Figure 2 shows the large margin effect for the Barrett's cancer data set discussed later in the experiments, where the LM-VGPMIL improves two percentage points on the VGPMIL.

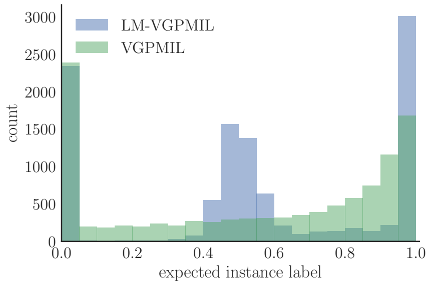


Figure 2: Visualization of the difference between LM-VGPMIL and the standard VGPMIL. Histogram of the expected instance labels for both models after training on the Barrett's cancer data set. While VGPMIL has a bimodal structure with two clusters around zero and one, the LM-VGPMIL has a trimodal structure, pushing uncertain instances towards 0.5, hence effectively eliminating them.

While there exist prior approaches to Bayesian large-margin learning [15, 26], ours is the first one to apply this idea to the MIL setting, which we believe to be valuable for the computer vision community.

3.3. Object detection with VGPMIL

Apart from its methodological novelty, our VGPMIL can serve as an essential building block of a weakly supervised object detection pipeline, which consists of three standard modules: (i) a region proposal generator, (ii) a feature extractor, and (iii) a classifier (See Figure 3). For the fully supervised setting, R-CNN [13] achieved a drastic improvement in prediction performance using CNNs as feature extractors. The follow-up work [12, 18] brought additional improvements by joining the latter two modules (and combining computations). An end-to-end trained version of R-CNN [35] outperformed all its predecessors.

The recent trend in weakly supervised object detection follows adaptations of the same three-module pipeline. Cinbis *et al.* [7] cascade modules as separate processing steps,

[5] Visualizations of C and V's effect can be found in the Supplement.

[6] To retain closed-form updates, we need to use the approximation $|f_n| \approx (2\langle y_n\rangle - 1)f_n$. The factor $(2\langle y_n\rangle - 1)$—the expected instance label rescaled to $[-1, 1]$—ensures the positivity of the whole expression as the signs of the two factors should agree. This introduces mutual dependence between g and y, resulting in a directed cyclic graphical model [40]. Variational inference applied to this setting is analogous to loopy belief propagation. Hence, variational parameter updates no longer guarantee a non-decreasing ELBO. However, our experimental results show that this does not harm performance in practice.

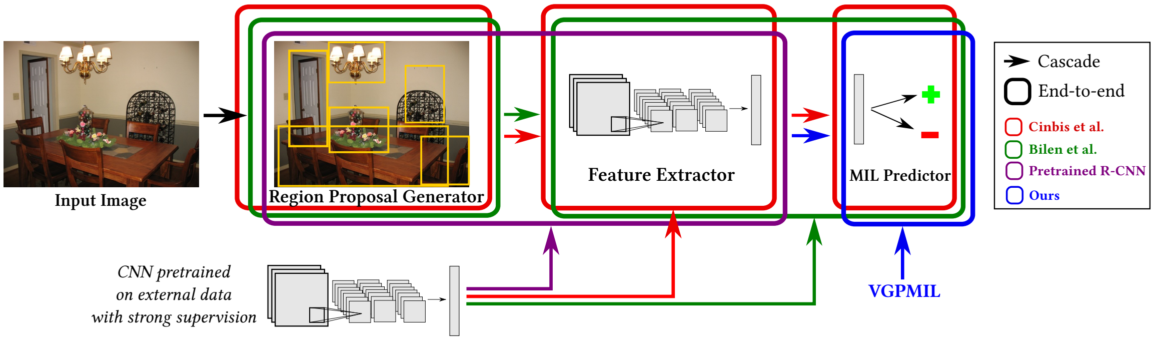


Figure 3: General weakly supervised detection pipeline and how different methods fit into it. Colored boxes group the end-to-end trained parts of the pipeline, while colored arrows indicate disjoint processing steps.

Method	mAP
GPMIL [20]	0.40
VF [30]	0.59
VFr [30]	0.67
DPMIL [20]	0.70
GICF [24]	0.71
VGPMIL (ours)	0.65
VGPMIL & kPCA (ours)	0.72
LM-VGPMIL (ours)	**0.73**

Table 1: Instance label prediction scores on the 20 Newsgroups data set. See Supplement for more detailed results.

representing the weakly-supervised counterpart of the plain R-CNN. Bilen *et al.* [3] achieve better results by joining feature extraction and MIL classification via a CNN, while keeping region proposal generation separate.

A main weakness of off-the-shelf region proposal generators is that the priority score they assign to proposals does not retrieve target patterns with high recall. Bilen *et al.* solve this issue by an internal scoring mechanism and training a committee of CNNs. Li *et al.* [27] assign a heuristic score to region proposals and train one single CNN interchangeably, which lets them achieve similar performance levels with much less computational effort.

From prior work, we deduce three observations: i) end-to-end training improves performance, ii) scoring region proposals with high recall is of critical importance, iii) all three weakly supervised object detection methods build one module on a CNN pretrained on supervised external data. We merge all these three lessons learned and construct a pipeline orthogonal to previous work. We perform region proposal generation and feature extraction jointly and recruit a pretrained CNN for this task. This translates precisely to feeding images into the region proposal network part of a Faster R-CNN trained on another data set and using the feature map of the fully-connected layer assigned to each region proposal as its feature vector. Finally, we feed these feature vectors into a powerful and scalable MIL predictor for final detection. Our VGPMIL serves as such a predictor. It is powerful because it can learn as complex decision boundaries as a multilayer perceptron with infinite number of neurons [32]. It is scalable because its update rules scale linearly with the training set size. Figure 3 visualizes how our approach differs from two seminal works in the way it composes the modules.

4. Experiments

We evaluate our models in three settings: (i) the 20 Newsgroups data set introduced by [48], (ii) the Barrett's cancer data set introduced by [19], (iii) the PASCAL VOC 2007 & 2012 data sets [9] with the goal of object detection. The third setting is meant to illustrate how our model can be an essential part in a pipeline that solves a mainstream computer vision application with unprecedented success. We chose the 20 Newsgroups data set as a standard benchmark for instance label prediction with MIL. The Barrett's cancer data set is an interesting medical image analysis application showing evidence that our findings can generalize across application fields.

4.1. 20 Newsgroups data set

We evaluate our model on the 20 Newsgroups corpus as introduced in [48]. It contains 20 data sets, each consisting of 100 bags (50 positive and 50 negative). Each bag contains around 40 instances of posts from 20 different topics. Each instance is one post represented by the 200 top TF-IDF features. Despite not being a computer vision application, this data set is informative since it is curated to test the extreme case in MIL where positive bags contain very few ($\approx 3\%$) positive instances. Consequently, it has been widely used as a standard benchmark for weakly supervised instance predictors.

Following previous work [20, 24, 30], we report results

Method	**Accuracy (in %)**	**F-Score**
GPMIL [20]	65.8	0.54
DPMIL [20]	71.8	0.74
VGPMIL (ours)	75.1	0.76
LM-VGPMIL (ours)	**77.3**	**0.77**

Table 2: Performance scores for localization of Barrett's cancer tumors from histopathology tissue microarrays.

on ten times 10-fold cross validation using the readily available splits. We report the results in Table 1. As the classes in this data set are poorly separated, LM-VGPMIL can show its strength over VGPMIL. This can be seen from the fact that VGPMIL comes much closer to the performance of LM-VGPMIL when the input is preprocessed with Kernel PCA, as done in [20]. Yet, LM-VGPMIL is still one percentage point ahead, benefiting from end-to-end learning of the non-linear class separation and classification stages.

4.2. Barrett's cancer data set

The second testbed for our model is localization of Barrett's cancer from histopathology tissue microarray images. This is an interesting application field for weakly supervised learning methods, since annotations for histopathology images can only be provided by expert pathologists. Because such annotations are extremely expensive, any tool to alleviate the pathologist's effort would be greatly valuable for the field. We pursue our experiments on the data set kindly supplied by the authors of [20], which consists of 210 tissue slides (143 cancerous, 67 healthy) containing 14353 pixel patches/instances[7], each represented by a 738 dimensional feature vector[8]. We choose the inducing point count $M = 50$ and initialize them to k-means centroids, following prior art [16, 21]. We split the training instances based on their bag labels and apply k-means to both classes separately, choosing $k = 25$. We construct the inducing point set by concatenating the centroids found for both classes. Table 2 reports the average accuracies and F1-Scores after four-fold cross-validation repeated five times. Both versions of our model perform markedly better than both the baseline GPMIL and the previous state of the art DPMIL, with the large-margin version of the model improving another two percentage points on VGPMIL.

4.3. PASCAL VOC

The PASCAL VOC 2007 data set consists of 9963 images containing objects from 20 classes split into a training/validation (trainval) set of 5011 images and a test set of 4952 images, with VOC 2012 being roughly twice as large.

Method	**VOC 2007**		**VOC 2012**	
	mAP	**CorLoc**	**mAP**	**CorLoc**
Cinbis *et al.* [7]	30.2	54.2	–	–
Teh *et al.* [41]	34.5	64.6	–	–
Kantorov *et al.* [22]	36.3	55.1	35.3	54.8
Shi and Ferrari [37]	37.2	64.7	–	–
Bilen and Vedaldi [3]	39.3	58.0	–	–
Li *et al.* [27]	39.5	52.4	–	–
VGPMIL (ours)	**46.1**	**66.0**	34.6	58.3
LM-VGPMIL (ours)	43.1	62.5	**37.8**	**60.8**

Table 3: Performance scores on PASCAL VOC data sets. *mAP* is reported on the test set, *CorLoc* on the trainval set. See Supplement for more detailed results.

For simplicity, we train a separate model for each of the classes following a one-versus-all approach[9]. We use the region proposal network part of a Faster R-CNN as both the region proposal generator and feature extractor. The network architecture is based on the VGG-16 [38] and is pretrained on the MS-COCO data set [29] using a Caffe based implementation[10].

We treat each image as a bag and the top-ranking 50 region proposals for each region as the instances, giving us for example on the PASCAL VOC 2007 data set 250550 training instances and 247600 test instances. We reduce the input dimensionality from 4096 to 500 via principal component analysis (PCA). Similarly to the previous experiments, we use a set of 50 inducing points fitted via k-means for the VGPMIL variants and keep them fixed throughout training. We train our models for 20 iterations and apply non-maximum suppression on the predictions. We set $C = 2$ and $V = 2$ for LM-VGPMIL as prior guesses. Fitting these values to data with cross validation or gradient descent could only improve the results further. We report our results in Table 3 along with a comparison to the state of the art. Following earlier work, we report both Correct Localization (CorLoc) [8] on the trainval split and mean average precision (mAP) on the test split.

4.3.1 Mixed supervision

Our model extends easily to mixed supervision: Part of the data is fully supervised and the (usually larger) rest contains only image level labels. This is a relatively less studied setup for mainstream computer vision tasks, one rare exception being Cinbis *et al.* [7]. One pitfall of the MIL

[7] They report 14303 instances due probably to a typographical error.

[8] See [19] for a detailed description of the features.

[9] VGPMIL can also be easily extended for the multiclass case, *e.g.* by replacing the Bernoulli distribution in (6) with a multinomial distribution, lower bounding it with the bound on the softmax introduced by [5]. However, we have observed this not to improve the results in PASCAL VOC 2007. Hence we report only the simpler binary output version.

[10] https://github.com/rbgirshick/py-faster-rcnn

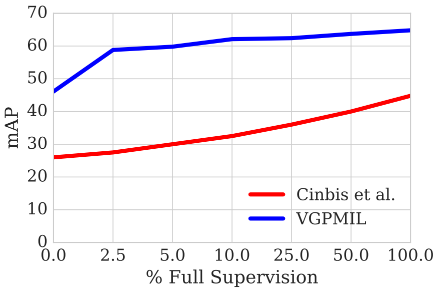


Figure 4: Object detection results for mixed supervision. The plot shows the change of detection performance on PASCAL VOC 2007 as the percentage of fully supervised bags (*i.e*. instance-level supervision) increases, while the others are still only weakly labeled at bag level. The scores for Cinbis *et al*. [7] are estimated from their Figure 9.

setup is that a model is never exposed to a precise example of the target pattern, which is prone to ambiguities (e.g. are we searching for an aeroplane or a wing?). Even a tiny amount of full supervision could solve this problem with ignorable annotation overhead. Furthermore, full supervision allows us to group inducing points into more discriminative groups. As Figure 4 shows, a small portion of strong supervision is sufficient to boost up performance.

4.4. Discussion

Both VGPMIL and LM-VGPMIL improve the state of the art in three data sets and LM-VGPMIL also on PASCAL VOC 2012. While LM-VGPMIL reaches highest performance scores in 20 Newsgroups, Barrett's cancer and VOC 2012 data sets, it lags three percentage points behind VGPMIL in VOC 2007. We speculate that this is due to the structure of the data. In VOC 2007, the pretrained deep neural net we used for joint region proposal generation and feature extraction is apparently able to achieve a high level of class separation. Hence, regularizing the margin further does not bring any benefit in this particular case. In contrast, the additional parameters introduced into LM-VGPMIL lead to a performance drop. On the other hand, the Barrett's cancer and 20 Newsgroups data sets exhibit a high level of class clutter, calling for models with robust predictions at regions nearby the decision boundary. The same holds true for the larger VOC 2012 data. In these cases, LM-VGPMIL improves on the plain VGPMIL. Consequently, VGPMIL catches up with LM-VGPMIL when the input data is preprocessed with Kernel PCA, which performs dimensionality reduction on the Hilbert space, leveraging class separation. We provide t-SNE [31] visualizations of Barrett's cancer and PASCAL VOC 2007 data sets in the Supplement Figure 1 to illustrate the difference between the levels of class clutter in these two applications.

Our object detection pipeline improves the state of the art in the PASCAL VOC data sets due to two reasons: i) better region proposals represented with more expressive features thanks to the deep network of [35] which is trained on external data to perform region proposal generation and object detection jointly, ii) a powerful MIL algorithm that can benefit maximally from the output of this network. Unfortunately, it is not trivial to measure individual contributions of these two factors to performance, since existing MIL classifiers are either tailored to a pipeline taking raw region proposals as input [3, 7, 27] or do not scale up to hundreds of thousands of data points [1, 14, 23] or are not able to predict instance labels [21, 25, 45]. To evaluate the influence of the proposal quality, we trained the model by Li *et al*. [27], using their public implementation and default parameters, on our generated region proposals for VOC 2007. This gave a performance of 31.9 mAP on the test set, indicating that the proposals alone are not sufficient. Since existing MIL classifiers cannot evaluate both the influence of the regions and their features jointly, we adapted the DMIL model by [47] to allow for instance prediction to replace the (LM-)VGPMIL part of the pipeline. This approach achieves 36.7 mAP, which puts it in the performance range of recent methods, yet clearly below ours. This indicates that while the CNN part of the pipeline is very powerful, a strong MIL method as the second part is still necessary for state of the art results.

By virtue of closed-form update rules, both VGPMIL and LM-VGPMIL can be trained without requiring to fine tune a learning rate. Furthermore, they both follow steep learning curves and converge to their eventual prediction scores within 20 iterations (See Supplement Figure 2).

5. Conclusion

We made GPMIL efficiently and scalably trainable by variational inference with closed-form updates. We reported experiments on three different applications where our model improves the state of the art. We also demonstrated that our model extends naturally to the mixed supervision setting, allowing the model to profit simultaneously from bag-level and instance-level annotations.

Our model achieves a performance jump on the PASCAL VOC detection tasks thanks to the end-to-end pretrained region proposal generator and feature extractor, also to its effective combination with our proposed VGPMIL model. Inspired by how performance evolved from plain R-CNN to Faster R-CNN, an interesting future direction is end-to-end training of VGPMIL together with the preceding blocks. This can be achieved using the recent Deep Kernel Learning [46] approach if its scalability bottleneck across input dimensionality can be resolved.

References

[1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, 2003. 1, 8

[2] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, 2009. 2

[3] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 1, 2, 6, 7, 8

[4] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. 4

[5] G. Bouchard. Efficient bounds for the softmax function and applications to approximate inference in hybrid models. In *NIPS 2007 Workshop for Approximate Bayesian Inference in Continuous/hybrid Systems*, 2007. 7

[6] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014. 2

[7] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 1, 2, 5, 7, 8

[8] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *International Journal of Computer Vision*, 100(3), 2012. 7

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2), 2010. 6

[10] Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553), 2015. 1

[11] S. Gidaris and N. Komodakis. Attend refine repeat: Active box proposal generation via in-out localization. *BMVC*, 2016. 2

[12] R. Girshick. Fast R-CNN. In *ICCV*, 2015. 2, 5

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2, 5

[14] Y. Han, Q. Tao, and J. Wang. Avoiding false positive in multi-instance learning. In *NIPS*, 2010. 8

[15] R. Henao, X. Yuan, and L. Carin. Bayesian nonlinear support vector machines and discriminative factor modeling. In *NIPS*, 2014. 5

[16] J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian processes for big data. In *UAI*, 2013. 7

[17] T. S. Jaakkola and M. I. Jordan. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10(1), 2000. 4

[18] H. Kaiming, Z. Xiangyu, R. Shaoqing, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014. 5

[19] M. Kandemir, A. Feuchtinger, A. Walch, and F. A. Hamprecht. Digital pathology: multiple instance learning can detect barrett's cancer. In *ISBI*, 2014. 6, 7

[20] M. Kandemir and F. A. Hamprecht. Instance label prediction by Dirichlet process multiple instance learning. In *UAI*, 2014. 2, 6, 7

[21] M. Kandemir, M. Haußmann, F. Diego, K. Rajamani, J. van der Laak, and F. A. Hamprecht. Variational weakly supervised Gaussian processes. In *BMVC*, 2016. 1, 2, 4, 7, 8

[22] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016. 2, 7

[23] M. Kim and F. Torre. Gaussian processes multiple instance learning. In *ICML*, 2010. 1, 2, 3, 8

[24] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth. From group to individual labels using deep features. In *SIGKDD*. ACM, 2015. 2, 6

[25] G. Krummenacher, C. Ong, and J. Buhmann. Ellipsoidal multiple instance learning. In *ICML*, 2013. 8

[26] N. Lawrence and M. Jordan. Semi-supervised learning via Gaussian processes. In *NIPS*, 2004. 5

[27] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang. Weakly supervised object localization with progressive domain adaptation. In *CVPR*, 2016. 1, 2, 6, 7, 8

[28] Y.-F. Li, J. T. Kwok, I. W. Tsang, and Z.-H. Zhou. A convex method for locating regions of interest with multi-instance learning. In *ECML/PKDD*, 2009. 2

[29] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, 2014. 7

[30] G. Liu, J. Wu, and Z.-H. Zhou. Key instance detection in multi-instance learning. *ACML*, 2012. 2, 6

[31] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2008. 8

[32] R. M. Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 1996. 1, 6

[33] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. 1, 2

[34] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CVPR*, 2016. 2

[35] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 5, 8

[36] M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian Process classification. *Journal of Machine Learning Research*, 3(Oct), 2002. 4

[37] M. Shi and V. Ferrari. Weakly supervised object localization using size estimates. In *ECCV*, 2016. 2, 7

[38] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 7

[39] E. Snelson and Z. Ghahramani. Sparse gaussian processes using pseudo-inputs. In *NIPS*, 2005. 3

[40] P. Spirtes. Directed cyclic graphical representations of feedback models. In *UAI*, 1995. 5

[41] E. W. Teh, M. Rochan, and Y. Wang. Attention networks for weakly supervised object localization. In *BMVC*, 2016. 7

[42] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2), 2013. 2

[43] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *ECCV*, 2014. 2

[44] X. Wang, Z. Zhu, C. Yao, and X. Bai. Relaxed multiple-instance SVM with application to object discovery. In *ICCV*, 2015. 2

[45] X.-S. Wei, J. Wu, and Z.-H. Zhou. Scalable multi-instance learning. In *ICDM*, 2014. 8

[46] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. Xing. Deep kernel learning. In *AISTATS*, 2015. 8

[47] J. Wu, Y. Yu, C. Huang, and K. Yu. Deep multiple instance learning for image classification and auto-annotation. In *CVPR*, 2015. 8

[48] Z.-H. Zhou, Y.-Y. Sun, and Y.-F. Li. Multi-instance learning by treating instances as non-iid samples. In *ICML*. ACM, 2009. 6

[49] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 2

Supplementary Material for the Paper: ”Variational Bayesian Multiple Instance Learning with Gaussian Processes”

Manuel Haußmann[1] Fred A. Hamprecht[1] Melih Kandemir[1,2*]
[1]HCI/IWR, Heidelberg University [2]Özyeğin University
{manuel.haussmann, fred.hamprecht}@iwr.uni-heidelberg.de
melih.kandemir@ozyegin.edu.tr

1. Updates for VGPMIL

As we derive the updates of the variational distributions, we drop all independent terms from the equations and use the notation $a \overset{c}{=} b$ to mean $a = b$ up to an additive constant.

Updating $q(u)$**.** Equation 9 gives us as the optimal update for $q(u)$

$$\log q(u) \overset{c}{=} \langle \log p(y|f) \rangle + \log p(u).$$

To derive an analytical solution we need to apply the Jaakkola bound on the first term

$$\begin{aligned} \langle \log p(y|f) \rangle &= \Big\langle \sum_n \log p(y_n|f_n) \Big\rangle \\ &\geq \sum_n \langle y_n \rangle \langle f_n \rangle - \frac{\langle f_n \rangle + \xi_n}{2} \\ &\quad - \lambda(\xi_n) \left(\langle f_n^2 \rangle - \xi_n^2 \right) \\ &\overset{c}{=} \sum_n \langle f_n \rangle \left(\langle y_n \rangle - \tfrac{1}{2} \right) - \lambda(\xi_n) \langle f_n^2 \rangle \\ &= \langle f \rangle^\top \left(\langle y \rangle - \tfrac{1}{2} \right) - \tfrac{1}{2} \langle f^\top \Lambda f \rangle, \end{aligned}$$

where $\Lambda = 2\mathrm{diag}\big((\lambda(\xi_1), ..., \lambda(\xi_N))\big)$. The complete expression then becomes[1]

$$\begin{aligned} \log q(u) &\overset{c}{=} \langle f \rangle^\top \left(\langle y \rangle - \tfrac{1}{2} \right) - \tfrac{1}{2} \langle f^\top \Lambda f \rangle - \tfrac{1}{2} u^\top K_{ZZ}^{-1} u \\ &\overset{c}{=} u^\top K_{ZZ}^{-1} K_{ZX} \left(\langle y \rangle - \tfrac{1}{2} \right) - \tfrac{1}{2} u^\top K_{ZZ}^{-1} u \\ &\quad - \tfrac{1}{2} u^\top K_{ZZ}^{-1} K_{ZX} \Lambda K_{XZ} K_{ZZ}^{-1} u, \end{aligned}$$

and we get the desired $q(u) = \mathcal{N}(u|m, S)$ with

$$\begin{aligned} S &\leftarrow \left(K_{ZZ}^{-1} K_{ZX} \Lambda K_{XZ} K_{ZZ}^{-1} + K_{ZZ}^{-1} \right)^{-1}, \\ m &\leftarrow S K_{ZZ}^{-1} K_{ZX} \left(\langle y \rangle - \tfrac{1}{2} \right). \end{aligned}$$

The optimal update for ξ_n^2 can be derived by taking the derivative of the Jaakkola bound.

The max **decomposition (Equation 13).** The variational inference updates for $q(y_n)$ in the paper rely on our ability to separate instance y_n from the others in bag b ($\{y_i\}_{b-n}$) in the expression $\max\{y_i\}_b$. To achieve this we introduced the following decomposition for the max over binary y_i

$$\max\{y_i\}_b = y_n + \max\{y_i\}_{b-n} - y_n \max\{y_i\}_{b-n}.$$

To proof this equation we use the noisy OR function, *i.e.* that for a set of binary values $\{y_1, ..., y_N\} \in \{0,1\}^N$ we have

$$\max\{y_1, ..., y_N\} = 1 - \prod_{i=1}^N (1 - y_i).$$

With this, we get

$$\begin{aligned} \max\{y_i\}_b &= 1 - \prod_i (1 - y_i) = 1 - (1 - y_n) \prod_{i \neq n} (1 - y_i) \\ &= 1 - \prod_{i \neq n} (1 - y_i) + y_n - y_n + y_n \prod_{i \neq n} (1 - y_i) \\ &= \max\{y_i\}_{b-n} + y_n - y_n \Big(1 - \prod_{i \neq n} (1 - y_i) \Big) \\ &= y_n + \max\{y_i\}_{b-n} - y_n \max\{y_i\}_{b-n}. \end{aligned}$$

Updating $q(y_n)$**.** Starting from Equation 9 we have, assuming instance n to be in bag b, that

$$\begin{aligned} \log q(y_n) &\overset{c}{=} \langle \log p(T_b|\{y_i\}_b) + \log p(y_n|f_n) \rangle \\ &\overset{c}{=} \langle G_b \rangle \log H + \langle \log p(y_n|f_n) \rangle. \end{aligned}$$

The second term evaluates to

$$\langle \log p(y_n|f_n) \rangle = y_n \langle f_n \rangle + \mathrm{const}, \qquad (*)$$

[*] The main part of this work has been done while the author was with Heidelberg Collaboratory for Image Processing (HCI), Heidelberg University

[1] Where we use the trace trick to evaluate the expectation $\langle f^\top \Lambda f \rangle$. That is $\langle f^\top \Lambda f \rangle = \langle \mathrm{tr}(\Lambda f f^\top) \rangle = \mathrm{tr}(\Lambda \langle f f^\top \rangle)$.

while the first term can be extended as

$$\begin{aligned}\log H \cdot \langle G_b \rangle &= \log H \cdot \Big\langle T_b \max\{y_i\}_b \\ &\qquad + (1 - T_b)(1 - \max\{y_i\}_b) \Big\rangle \\ &\overset{c}{=} \log H \cdot \big(2T_b \langle \max\{y_i\}_b \rangle - \langle \max\{y_i\}_b \rangle \big).\end{aligned}$$

In order to separate the instance y_n from the other instances in bag b, we use the $\max$ decomposition from Equation 13. This gives

$$\begin{aligned}\log H \cdot \langle G_b \rangle &\overset{c}{=} \log H \cdot \Big(2T_b\big(y_n + \langle \max\{y_i\}_{b-n} \rangle \\ &\qquad - y_n \langle \max\{y_i\}_{b-n} \rangle\big) - y_n \\ &\qquad - \langle \max\{y_i\}_{b-n} \rangle + y_n \langle \max\{y_i\}_{b-n} \rangle \Big) \\ &\overset{c}{=} \log H \cdot \Big(2T_b\big(y_n - y_n \langle \max\{y_i\}_{b-n} \rangle\big) \\ &\qquad - y_n + y_n \langle \max\{y_i\}_{b-n} \rangle \Big) \\ &\overset{c}{=} y_n \log H \cdot \Big(2T_b - 2T_b \langle \max\{y_i\}_{b-n} \rangle \\ &\qquad + \langle \max\{y_i\}_{b-n} \rangle - 1 \Big). \qquad (**)\end{aligned}$$

We combine $(*)$ and $(**)$ to arrive at the desired update of $q(y_n) = \mathcal{B}er(y_n|\pi_n)$ with

$$\begin{aligned}\pi_n \leftarrow \sigma\Big(&\langle f_n \rangle + \log H \cdot \big(2T_b + \langle \max\{y_i\}_{b-n} \rangle \\ &- 2T_b \langle \max\{y_i\}_{b-n} \rangle - 1\big)\Big).\end{aligned}$$

2. Updates for LM-VGPMIL

Our variational distribution for the LM-VGPMIL model is given by $Q = q(u)p(f|u)\prod_n q(y_n)q(g_n)$. However, as mentioned in the main text, to still be able to perform closed form updates, we need to approximate $|f_n|$. We approximate it by $(2y_n - 1)f_n$, where the first factor is the (latent) instance label transformed to $\{\pm 1\}$ to agree with the sign of f_n. Note that this expression poses the problem that we create a circular/loopy structure where y and g both depend on each other, resulting in a directed cyclical graphical model [10]. Unfortunately this means that we lose the guarantees provided by variational inference—a non-decreasing ELBO in each iteration—but our experiments show that the model is still able to learn in practice.

Since we now have a second Bernoulli distribution, we need to apply the bound on the sigmoid twice. We will denote the variational parameters introduced for each instance by the necessary bounds on the Bernoulli distribution in Equation 19 by ξ_n and those for Equation 20 by ϕ_n. Apart from this second bound the updates are derived completely analogously to their non-margin relatives and we will only state the final updates.

Updating $q(u)$. As in our main model Equation 9 gives us the optimal update of $q(u)$ to be

$$\begin{aligned}q(u) &= \mathcal{N}(u|m, S) \qquad \text{with} \\ S &\leftarrow \left(K_{ZZ}^{-1} K_{ZX} \Lambda K_{XZ} K_{ZZ}^{-1} + K_{ZZ}^{-1}\right)^{-1}, \\ m_n &\leftarrow S K_{ZZ}^{-1} K_{Zx_n} \Big(C(2\langle y_n \rangle - 1)\left(\langle g_n \rangle - \tfrac{1}{2}\right) \\ &\qquad + 2C^2 \lambda(\xi_n)(2\langle y_n \rangle - 1)V \\ &\qquad + \langle g_n \rangle \left(\langle y_n \rangle - \tfrac{1}{2}\right)\Big),\end{aligned}$$

where $\Lambda = 2\text{diag}\left(\left(..., \lambda(\xi_n)C^2 + \lambda(\phi_n)\langle g_n \rangle, ...\right)\right)$.

Updating $q(g_n)$ and $q(y_n)$. For $q(g_n)$ we get

$$\begin{aligned}q(g_n) &= \mathcal{B}er(g_n|\tau_n) \qquad \text{with} \\ \tau_n &\leftarrow \sigma\Big(C\big(\langle(2\langle y_n \rangle - 1)f_n\rangle - V\big) \\ &\qquad + \langle f_n \rangle \left(\langle y_n \rangle - \tfrac{1}{2}\right) - \lambda(\phi_n)\langle f_n^2 \rangle\Big).\end{aligned}$$

And for $q(y_n)$ we have

$$\begin{aligned}q(y_n) &= \mathcal{B}er(y_n|\pi_n) \qquad \text{with} \\ \pi_n &\leftarrow \sigma\Big(2\langle g_n \rangle C \langle f_n \rangle - C\langle f_n \rangle + 4\lambda(\xi_n)C^2 \langle f_n \rangle V \\ &\qquad + \langle f_n \rangle \langle g_n \rangle + \log H \Big(2T_b - 2T_b \langle \max\{y_i\}_{b-n} \rangle \\ &\qquad - 1 + \langle \max\{y_i\}_{b-n} \rangle \Big)\Big),\end{aligned}$$

assuming instance n is in bag b.

Updating ξ_n, ϕ_n. Lastly, the updates for the newly introduced variational parameters are the following:

$$\begin{aligned}\xi_n^2 &\leftarrow C^2\left(\langle f_n^2 \rangle - 2(2\langle y_n \rangle - 1)\langle f_n \rangle V + V^2\right) \\ \phi_n^2 &\leftarrow \langle f_n^2 \rangle \langle g_n \rangle.\end{aligned}$$

3. Tables & Figures

Tables 1-5 on the following pages report the class based performances on the PASCAL VOC and Newsgroup data sets. Figure 1 gives t-SNE visualizations, Figure 2 the learning rates for the three smaller data sets and Figure 3, 4 visualize the influence of the additional hyperparameters C, V for the large margin version of the model.

Model	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv	avg
Cinbis *et al.* [2]	67.1	66.1	49.8	34.5	23.3	68.9	**83.5**	44.1	27.7	71.8	**49.0**	48.0	65.2	79.3	37.4	42.9	65.2	51.9	62.8	46.2	54.2
Teh *et al.* [11]	**84.0**	64.6	70.0	**62.4**	25.8	**80.7**	73.9	**71.5**	35.7	81.6	46.5	71.3	79.1	78.8	**56.7**	34.3	69.8	56.7	77.0	**72.7**	64.6
Kantorov *et al.* [4]	83.3	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
Shi and Ferrari [9]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	64.7
Bilen and Vedaldi [1]	68.9	**68.7**	65.2	42.5	40.6	72.6	75.2	53.7	29.7	68.1	33.5	45.6	65.9	**86.1**	27.5	44.9	76.0	**62.4**	66.3	66.8	58.0
Li *et al.* [6]	78.2	67.1	61.8	38.1	36.1	61.8	78.8	55.2	28.5	68.8	18.5	49.2	64.1	73.5	21.4	47.4	64.6	22.3	60.9	52.3	52.4
VGPMIL (ours)	81.5	62.1	68.5	45.3	**68.0**	80.1	58.3	64.7	**54.8**	**84.4**	34.5	71.0	**80.5**	63.7	53.2	**52.7**	**82.3**	**62.4**	**80.1**	71.1	**66.0**
LM-VGPMIL (ours)	78.6	63.0	**71.2**	44.2	65.2	76.3	63.4	63.2	49.0	75.2	17.5	**72.7**	74.6	58.4	52.4	48.6	70.8	55.9	78.9	70.3	62.5

Table 1: PASCAL VOC2007 CorLoc. Results on the trainval set, separated over all classes.

Model	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv	avg
Cinbis *et al.* [2]	39.3	43.0	28.8	20.4	8.0	45.5	47.9	22.1	8.4	33.5	23.6	29.2	38.5	47.9	20.3	20.0	35.8	30.8	41.0	20.1	30.2
Teh *et al.* [11]	48.8	45.9	37.4	**26.9**	9.2	50.7	43.4	43.6	10.6	35.9	27.0	38.6	48.5	43.8	24.7	12.1	29.0	23.2	48.8	41.9	34.5
Kantorov *et al.* [4]	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	**49.2**	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
Shi and Ferrari [9]	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	37.2
Bilen and Vedaldi [1]	46.4	**58.3**	35.5	25.9	14.0	**66.7**	53.0	39.2	8.9	41.8	26.6	38.6	44.7	59.0	10.8	17.3	40.7	**49.6**	56.9	50.8	39.3
Li *et al.* [6]	54.5	47.4	41.3	20.8	17.7	51.9	**63.5**	**46.1**	21.8	57.1	22.1	34.4	50.5	**61.8**	16.2	**29.9**	40.7	15.9	55.3	40.2	39.5
VGPMIL (ours)	**57.9**	45.8	44.2	23.6	**54.8**	54.7	39.3	40.4	**34.9**	**63.7**	19.0	**53.4**	**61.4**	50.1	**31.1**	27.3	**63.0**	45.0	**59.1**	**52.9**	**46.1**
LM-VGPMIL (ours)	51.9	47.1	**46.4**	17.9	52.3	48.4	41.6	38.8	31.8	59.6	7.4	53.2	53.8	47.5	30.6	28.4	56.2	41.3	56.9	51.9	43.1

Table 2: PASCAL VOC2007 Detection. Results on the test set, separated over all classes.

Model	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv	avg
Kantorov *et al.* [4]	**78.3**	**70.8**	52.5	34.7	36.6	80.0	58.7	38.6	27.7	71.2	**32.3**	48.7	76.2	**77.4**	16.0	48.4	69.9	**47.5**	66.9	**62.9**	54.8
VGPMIL (ours)	59.6	50.9	68.4	**42.7**	**57.2**	75.8	66.3	**64.4**	**47.2**	74.6	19.0	67.8	78.2	66.3	56.7	45.4	75.1	33.3	62.9	54.6	58.3
LM-VGPMIL (ours)	67.9	70.7	**72.7**	35.6	47.6	**81.9**	**72.7**	57.2	36.2	**75.6**	11.3	**69.9**	**84.4**	76.8	**66.0**	**51.0**	**80.3**	30.8	**72.1**	54.8	**60.8**

Table 3: PASCAL VOC2012 CorLoc. Results on the trainval set, separated over all classes.

Model	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	persn	plant	sheep	sofa	train	tv	avg
Kantorov *et al.* [4]	**64.0**	**54.9**	36.4	8.1	12.6	53.1	40.5	28.4	6.6	35.3	**34.4**	**49.1**	42.6	**62.4**	19.8	15.2	27.0	**33.1**	33.0	**50.0**	35.3
VGPMIL† (ours)	29.4	30.9	37.2	**15.4**	**35.4**	47.4	42.5	**41.4**	**22.6**	53.3	5.1	43.4	54.7	42.7	35.9	20.1	54.0	17.1	40.0	23.8	34.6
LM-VGPMIL‡ (ours)	39.4	49.7	**42.1**	11.3	13.8	**55.4**	**48.5**	35.7	13.4	**57.3**	2.7	48.3	**63.7**	56.0	**50.5**	**25.2**	**61.1**	13.1	**46.3**	22.8	**37.8**

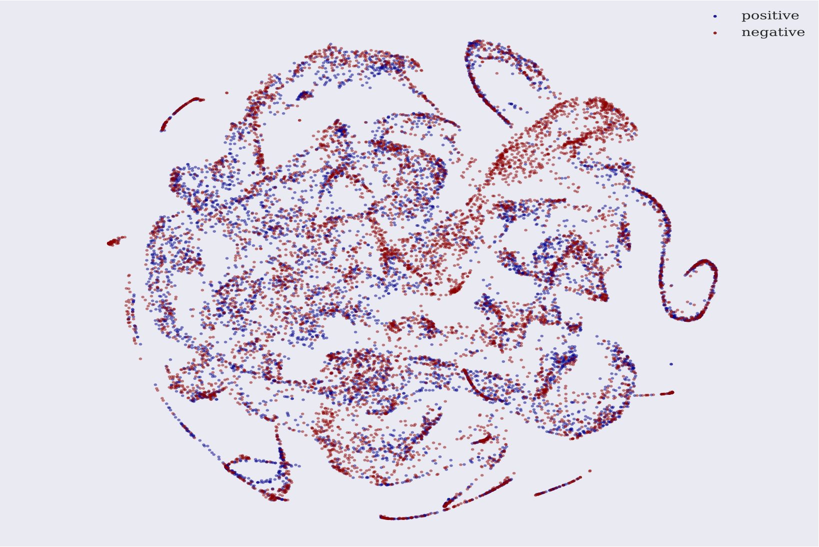
Table 4: PASCAL VOC2012 Detection. Results on the test set, separated over all classes.

†http://host.robots.ox.ac.uk:8080/anonymous/SO7RUO.html

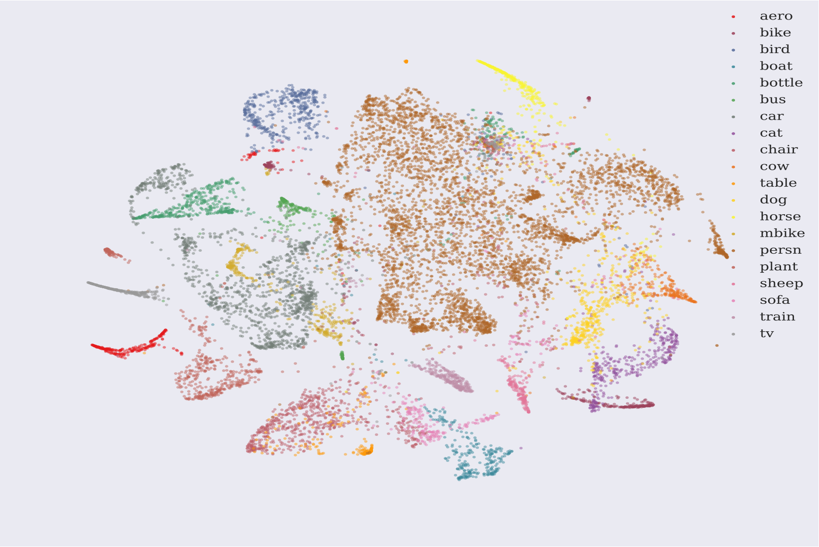
‡http://host.robots.ox.ac.uk:8080/anonymous/9Q2KP6.html

Data Set	LM-VGPMIL (ours)	VGPMIL (ours)	GICF [5]	DPMIL [3]	VF [7]	VFr [7]	GPMIL [3]
alt.atheism	**0.70**	0.67	–	0.67	–	–	0.44
comp.graphics	**0.79**	0.54	–	**0.79**	–	–	0.49
comp.os.ms-windows.misc	**0.52**	0.38	–	0.51	–	–	0.36
comp.sys.ibm.pc.hardware	**0.70**	0.55	–	0.67	–	–	0.35
comp.sys.mac.hardware	**0.79**	0.66	–	0.76	–	–	0.54
comp.windows.x	0.69	0.64	–	**0.73**	–	–	0.36
misc.forsale	**0.54**	0.48	–	0.45	–	–	0.33
rec.autos	0.71	0.46	–	**0.76**	–	–	0.38
rec.motorcycles	**0.76**	0.69	–	0.69	–	–	0.46
rec.sport.baseball	**0.76**	**0.76**	–	0.74	–	–	0.38
rec.sport.hockey	**0.94**	**0.94**	–	0.91	–	–	0.43
sci.crypt	0.82	**0.88**	–	0.68	–	–	0.31
sci.electronics	**0.92**	0.80	–	0.90	–	–	0.71
sci.med	**0.73**	0.65	–	**0.73**	–	–	0.32
sci.space	**0.74**	0.69	–	0.70	–	–	0.32
soc.religion.christian	**0.73**	0.69	–	0.72	–	–	0.45
talk.politics.guns	**0.72**	0.67	–	0.64	–	–	0.38
talk.politics.mideast	**0.87**	0.84	–	0.80	–	–	0.46
talk.politics.misc	**0.64**	0.61	–	0.60	–	–	0.29
talk.religion.misc	0.49	0.45	–	**0.51**	–	–	0.32
Average	**0.73**	0.65	0.71	0.70	0.67	0.59	0.40

Table 5: 20 Newsgroups database. Results for each data set.

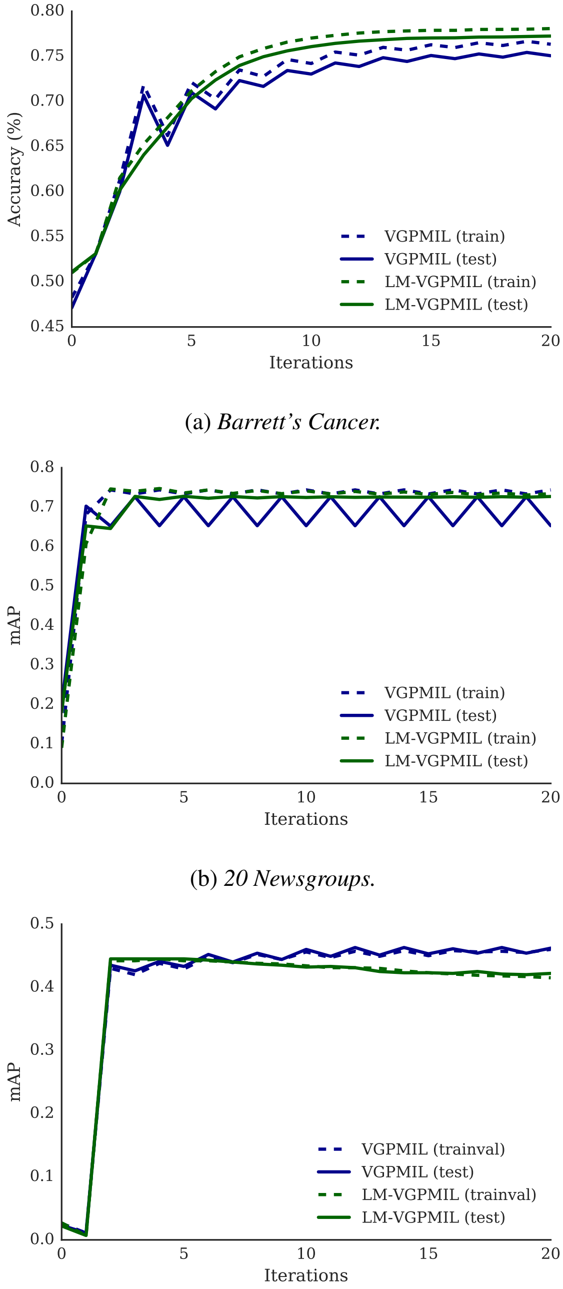


(a) *t-SNE on Barrett's cancer data set.*



(b) *t-SNE on PASCAL VOC 2007.*

Figure 1: t-SNE [8] visualization. For the Barrett's cancer data set all of the instances coloured according to the correct instance labels are plotted. For PASCAL VOC 2007 we plotted the (for the model unavailable) ground truth instances.

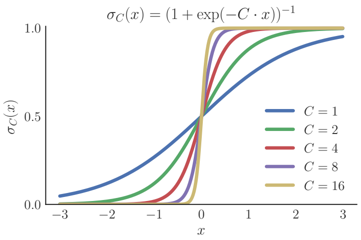


(a) *Barrett's Cancer.*

(b) *20 Newsgroups.*

(c) *PASCAL VOC 2007.*

Figure 2: **Learning Curves.** Both models have very steep learning curves and tend to come very close to their final performance after just a couple of iterations. While the regularization introduced in the LM-VGPMIL leads to smoother learning curves in all three data sets it's overall benefit varies between them. While it clearly outperforms VGPMIL in the Barrett's Cancer data set, for the 20 Newsgroups data base the curves are a lot closer. Both models come close to their final performance after just three iterations, but while the large margin model can deal with the heavy imbalance in the bags and their cluttered features, the base model keeps oscillating (as stated in the main text, this behaviour can be alleviated by further preprocessing the features *e.g.* with a kernel PCA). Figure (c) shows a case where the additional regularization actually hurts the predictive performance. While LM-VGPMIL initially reaches a higher performance than the VGPMIL model, its performance decreases with further iterations, whereas VGPMIL improves further.



(a) Effect of C

V	$\sigma(V)$
0	0.5
0.5	0.62
1	0.73
1.5	0.82
2	0.88
2.5	0.92
3	0.95

(b) Effect of V

Figure 3: Influence of C and V on the logistic sigmoid. With larger values of C, the sigmoid approximates the step function more closely and penalizes margin violations more heavily. Similarly, V shifts the sigmoid along the x-axis.

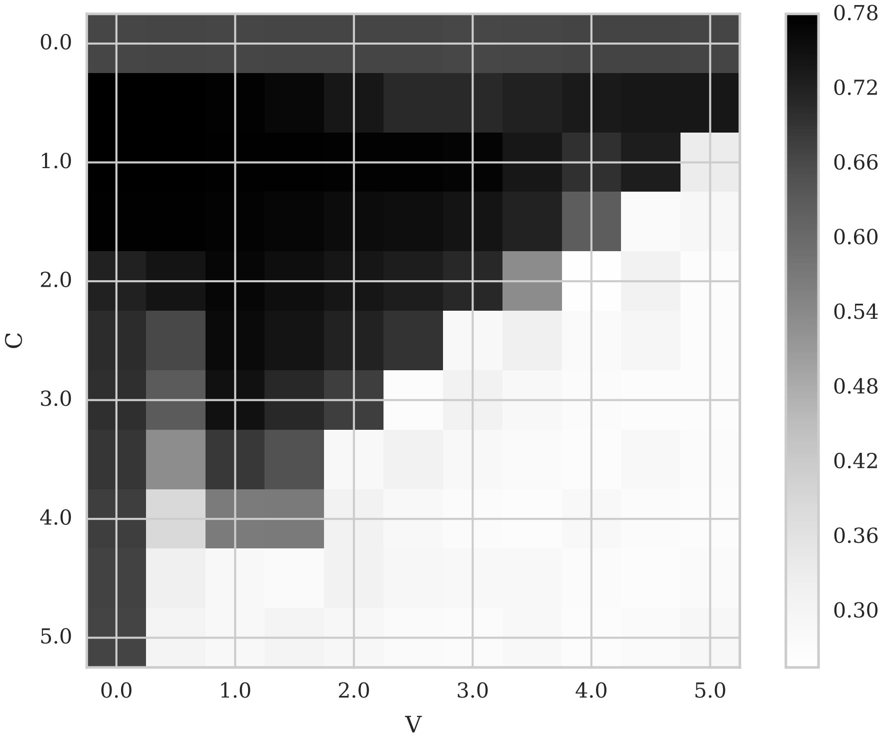


Figure 4: Influence of C and V in the LM-VGPMIL model trained on the Barrett's Cancer dataset.

References

[1] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, 2016. 3

[2] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016. 3

[3] M. Kandemir and F. A. Hamprecht. Instance label prediction by Dirichlet process multiple instance learning. In *UAI*, 2014. 4

[4] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, 2016. 3

[5] D. Kotzias, M. Denil, N. De Freitas, and P. Smyth. From group to individual labels using deep features. In *SIGKDD*. ACM, 2015. 4

[6] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang. Weakly supervised object localization with progressive domain adaptation. In *CVPR*, 2016. 3

[7] G. Liu, J. Wu, and Z.-H. Zhou. Key instance detection in multi-instance learning. *ACML*, 2012. 4

[8] L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2008. 5

[9] M. Shi and V. Ferrari. Weakly supervised object localization using size estimates. In *ECCV*, 2016. 3

[10] P. Spirtes. Directed cyclic graphical representations of feedback models. In *UAI*, 1995. 2

[11] E. W. Teh, M. Rochan, and Y. Wang. Attention networks for weakly supervised object localization. In *BMVC*, 2016. 3