Active Sampling for Graph-Aware Classification

Dimitris Berberidis and Georgios B. Giannakis
Dept. of ECE and Digital Tech. Center, University of Minnesota Minneapolis, MN 55455, USA

***Abstract*—The present work deals with data-adaptive active sampling of graph nodes representing training data for binary classification. The graph may be given or constructed using similarity measures among nodal features. Leveraging the graph for classification builds on the premise that labels over neighboring nodes are correlated according to a categorical Markov random field (MRF). This model is further relaxed to a Gaussian (G)MRF with labels taking continuous values, an approximation that not only mitigates the combinatorial complexity of the categorical model, but also offers optimal unbiased soft predictors of the unlabeled nodes. The proposed sampling strategy is based on querying the node whose label disclosure is expected to inflict the largest expected mean-square deviation on the GMRF, a strategy which subsumes the existing variance-minimization-based sampling method. A simple yet effective heuristic is also introduced for increasing the exploration capabilities, and reducing bias of the resultant estimator, by taking into account the confidence on the model label predictions. The novel sampling strategy is based on quantities that are readily available without the need for model retraining, rendering it scalable to large graphs. Numerical tests using synthetic and real data demonstrate that the proposed methods achieve accuracy that is comparable or superior to the state-of-the-art even at reduced runtime.**

***Index Terms*—Active learning, classification, graphs, expected change.**

I. Introduction

Active learning or adaptive sampling has recently gained popularity for various applications ranging from bioinformatics [1] to distributed signal classification and estimation [2]. It yields markedly improved classification accuracy over passive or random sampling when the number of training labels is fixed [3]–[5]. Moreover, it can be particularly appealing when unlabeled data (instances) are readily available, but obtaining training labels is expensive. For instance, a classifier trained to predict the presence of cancer based on certain protein attributes requires labels that involve costly and time-consuming medical examinations (see, e.g. [1]). Arguably, active sampling is expected to outperform random sampling when (un)labeled instances are correlated. Such a case emerges with graph-aware classification, where each instance is denoted by a node, while edges capture correlation among nodes. Although graphs may arise naturally in certain applications (e.g. social and citation networks), they can in general be constructed from any set of instances using proper similarity measures; see e.g., [6], [7]. In a nutshell, *graph-aware* classification boils down to propagating the information from labeled nodes to unlabeled ones through edges of neighboring nodes; thus, predictions of unlabeled nodes are performed jointly using the entire graph structure, see, e.g. [8]. As a result, classification on graphs is inherently semi-supervised and thus conducive to active learning.

Work was supported by NSF 1514056 and 1500713, and NIH 1R01GM104975-01. E-mails: {bermp001,georgios}@umn.edu

Prior art in graph-based active learning is either non-adaptive or adaptive. The former includes the *non-adaptive* design-of-experiments-type methods, where sampling strategies are designed *offline* depending only on the graph structure, based on ensemble optimality criteria such as variance minimization [9], the error upper bound minimization [10]; and the Σ-optimality metric [11]. The adaptive approaches select samples online and jointly with the classification process, taking into account both graph structure as well as previously obtained labels. Such *data-adaptive* methods give rise to sampling schemes that are not optimal on average, but adapt to a given realization of labels on the graph. Adaptive methods include the Bayesian risk minimization [12], the information gain maximization [13], as well as the manifold preserving method of [14]; see also [15]–[17]. Finally, related works deal with selective sampling of nodes that arrive sequentially in a gradually augmented graph [18]–[20], as well as active sampling to infer the graph structure [21], [22].

The present work develops an adaptive pool based active learning algorithm for graph-aware classification. The proposed sampling strategy relies on querying the node that is expected to inflict the largest change on the underlying label correlation model. Albeit in different context, a related criterion was adopted for semantic segmentation of images [23], and for regression of Gaussian processes [24]. Our approach here advocates a novel metric of expected model change based on the mean-square deviation (MSD) of the GMRF. A simple yet effective heuristic is also introduced for improving the exploration capabilities and reducing the bias of the resultant classifier, by taking into account the confidence on the model label predictions. The rest of the paper is organized as follows. Section II states the problem and the GMRF model adopted to approximate the marginal distributions of the unknown categorical node labels. Section III develops our MSD-based active sampling scheme and bias reduction heuristic. Finally, Section IV presents numerical tests on real and synthetic datasets.

II. Modeling and Problem Statement

Consider a connected undirected graph $\mathcal{G}=\{\mathcal{V},\mathcal{E}\}$, where $\mathcal{V}$ is the set of N nodes, and $\mathcal{E}$ contains the edges that are also represented by the $N\times N$ weighted adjacency matrix $\mathbf{W}$ whose $(i,j)-$th entry denotes the weight of the edge that connects nodes v_i and v_j. Let us further suppose that a binary label $y_i\in\{-1,1\}$ is associated with each node v_i. The weighted binary labeled graph can either be given, or, it can be inferred from a set of N data points $\{\mathbf{x}_i,y_i\}_{i=1}^N$ such that each node of the graph corresponds to a data point. Matrix $\mathbf{W}$ can be obtained from the feature vectors $\{\mathbf{x}_i\}_{i=1}^N$ using different similarity measures. For example, one may use the radial basis function $w_{i,j}=\exp\left(-\|\mathbf{x}_i-\mathbf{x}_j\|_2^2/\sigma^2\right)$ that

assigns large edge weights to pairs of points that are neighbors in Euclidean space, or the Pearson correlation coefficients $w_{i,j}=\langle \mathbf{x}_i,\mathbf{x}_j\rangle/(\|\mathbf{x}_i\|_2\|\mathbf{x}_j\|_2)$. If $w_{i,j}\neq 0\ \forall i,j$, the resulting graph will be fully connected, but one may obtain a more structured graph by rounding small weights to 0.

Having embedded the data on a graph, semi-supervised learning amounts to propagating an observed subset of labels to the rest of the network. Thus, upon observing $\{y_i\}_{i\in\mathcal{L}}$ where $\mathcal{L}\subseteq\mathcal{V}$, henceforth collected in the $|\mathcal{L}|\times 1$ vector $\mathbf{y}_\mathcal{L}$, the goal is to infer the labels of the unlabeled nodes $\{y_i\}_{i\in\mathcal{U}}$ concatenated in the vector $\mathbf{y}_\mathcal{U}$, where $\mathcal{U}:=\mathcal{V}/\mathcal{L}$. Let us consider labels as random variables that follow an unknown joint distribution $(y_1,y_2,\ldots,y_N)\sim p(y_1,y_2,\ldots,y_N)$, or $\mathbf{y}\sim p(\mathbf{y})$ for brevity.

For the purpose of inferring unobserved from observed labels, it would suffice if the posterior $p\left(\mathbf{y}_\mathcal{U}|\mathbf{y}_\mathcal{L}\right)$ were available; then, $\mathbf{y}_\mathcal{U}$ could be obtained as a combination of labels that maximizes $p\left(\mathbf{y}_\mathcal{U}|\mathbf{y}_\mathcal{L}\right)$. Moreover, obtaining the marginal $p(y_i|\mathbf{y}_\mathcal{L})$ of each unlabeled node i is often of interest, especially in the present greedy active sampling approach. To this end, it is well documented that MRFs are suitable for modeling probability mass functions over undirected graphs using the generic form, see e.g., [12]

$$p(\mathbf{y}):=\frac{1}{Z_\beta}\exp(-\frac{\beta}{2}\Phi(\mathbf{y})) \tag{1a}$$

where the "partition function" Z_β ensures that (1a) integrates to 1, β is a scalar that controls the smoothness of $p(\mathbf{y})$, and $\Phi(\mathbf{y})$ is the so termed "energy" of a given configuration of labels

$$\Phi(\mathbf{y}):=\sum_{i,j\in\mathcal{V}} w_{i,j}\left(y_i-y_j\right)^2=\mathbf{y}^T\mathbf{L}\mathbf{y} \tag{1b}$$

that captures the graph-induced label dependencies through the graph Laplacian matrix $\mathbf{L}:=\mathbf{D}-\mathbf{W}$ with $\mathbf{D}:=\mathrm{diag}(\mathbf{W}\mathbf{1})$. This categorical MRF model in (1a) naturally incorporates the known graph structure (through $\mathbf{L}$) in the label distribution by assuming label configurations where nearby labels (large edge weights) are similar, and have lower energy as well as higher likelihood. Still, finding the joint and marginal posteriors using (1a) and (1b) remains a task of *exponential complexity* since $\mathbf{y}_\mathcal{U}\in\{-1,1\}^{|\mathcal{U}|}$. To deal with this challenge, less complex continuous-valued models are well motivated for a scalable approximation of the marginal posteriors. This prompts our next step to allow for continuous-valued label configurations $\boldsymbol{\psi}_\mathcal{U}\in\mathbb{R}^{|\mathcal{U}|}$ that are modeled by a GMRF.

A. GMRF relaxation

Consider approximating the binary field $\mathbf{y}\in\{-1,1\}^{|\mathcal{U}|}$ that is distributed according to (1a) with the continuous-valued $\boldsymbol{\psi}\sim\mathcal{N}(\mathbf{0},\mathbf{C})$, where the covariance matrix satisfies $\mathbf{C}^{-1}=\mathbf{L}$. When the inverse of the rank deficient $\mathbf{L}$ is needed, the inverse of $\mathbf{L}+\delta\mathbf{I}$ will be used instead, with $\delta\ll 1$. Label propagation under this relaxed GMRF model becomes readily available in closed form. In fact, vector $\boldsymbol{\psi}_{\mathcal{U}|\mathcal{L}}$ of unlabeled nodes conditioned on the labeled ones is follows

$$\boldsymbol{\psi}_{\mathcal{U}|\mathcal{L}}\sim\mathcal{N}(\boldsymbol{\mu}_{\mathcal{U}|\mathcal{L}},\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}) \tag{2}$$

where $\mathbf{L}_{\mathcal{U}\mathcal{U}}$ is the part of the graph Laplacian that corresponds to unlabeled nodes in the partitioning

$$\mathbf{L}=\begin{bmatrix}\mathbf{L}_{\mathcal{U}\mathcal{U}} & \mathbf{L}_{\mathcal{U}\mathcal{L}}\\ \mathbf{L}_{\mathcal{L}\mathcal{U}} & \mathbf{L}_{\mathcal{L}\mathcal{L}}\end{bmatrix}. \tag{3}$$

Conditional expectation of the unobserved field given the observed $\boldsymbol{\psi}_\mathcal{L}$ is given by

$$\boldsymbol{\mu}_{\mathcal{U}|\mathcal{L}}=\mathbf{C}_{\mathcal{U}\mathcal{L}}\mathbf{C}_{\mathcal{L}\mathcal{L}}^{-1}\boldsymbol{\psi}_\mathcal{L}=-\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\boldsymbol{\psi}_\mathcal{L} \tag{4}$$

where the first equality is the conditional expectation of jointly Gaussian zero-mean vectors (see e.g., [25, p. 382]), while the second equality can be established using the partitioned form of the matrix factors in the identity $\mathbf{L}\mathbf{C}=\mathbf{I}$. When binary labels $\mathbf{y}_\mathcal{L}$ are obtained, they can be treated as measurements of the continuous field ($\boldsymbol{\psi}_\mathcal{L}:=\mathbf{y}_\mathcal{L}$), and combined with (4) to yield

$$\boldsymbol{\mu}_{\mathcal{U}|\mathcal{L}}=-\mathbf{L}_{\mathcal{U}\mathcal{U}}^{-1}\mathbf{L}_{\mathcal{U}\mathcal{L}}\mathbf{y}_\mathcal{L} \tag{5}$$

Interestingly, the mean of the Gaussian field may serve as an approximation of the marginal posteriors of the unknown labels, that is

$$\begin{aligned}p(y_i=1|\mathbf{y}_\mathcal{L})&=\frac{1}{2}\left(\mathbb{E}\left[\left[\mathbf{y}_{\mathcal{U}|\mathcal{L}}\right]_i\right]+1\right)\approx\frac{1}{2}\left(\mathbb{E}\left[\left[\boldsymbol{\psi}_{\mathcal{U}|\mathcal{L}}\right]_i\right]+1\right)\\&:=\frac{1}{2}\left(\left[\boldsymbol{\mu}_{\mathcal{U}|\mathcal{L}}\right]_i+1\right)\end{aligned} \tag{6}$$

where the first equality follows because the expectation of a Bernouli random variable equals its probability. Consequently, given the approximation of $p(y_i|\mathbf{y}_\mathcal{L})$ in (6), and the uninformative prior $p(y_i=1)=0.5\ \forall i\in\mathcal{V}$, the maximum a posteriori (MAP) estimate of y_i, which in the Gaussian case here reduces to the minimum distance decision rule, is

$$\hat{y}_i=\begin{cases}1 & \left[\boldsymbol{\mu}_{\mathcal{U}|\mathcal{L}}\right]_i>0\\ -1 & \text{else}\end{cases},\quad \forall i\in\mathcal{U} \tag{7}$$

thus completing the propagation of the observed $\mathbf{y}_\mathcal{L}$ to the unlabeled nodes of the graph. It is worth stressing at this point, that as the set of labeled samples changes, so does the dimensionality of the conditional mean in (5), along with the "auto-" and "cross-" Laplacian sub-matrices that enable label propagation via (5). Two remarks are now in order.

Remark 1. The method discussed here for label propagation (cf. (5)) is related with the one reported in [12]. The main differences are: i) we perform oprimal soft label propagation by minimizing the mean-square prediction error of unlabeled from labeled samples; and ii) our model approximates $\{-1,1\}$ labels with a *zero-mean* Gaussian field, while the model in [12] approximates $\{0,1\}$ labels also with a zero-mean Gaussian field (instead of one centered at 0.5). Apparently, [12] treats the two classes differently since it exhibits a bias towards class 0; thus, simply denoting class 0 as class 1 yields different marginal posteriors and classification results. In contrast, our model is bias-free and treats the two classes equally.

B. Active sampling with GMRFs

In passive learning, the set $\mathcal{L}$ of labeled nodes is either chosen at random, or, it is determined a priori. In our pool based active learning setup, the learner can examine a set of instances (nodes in graph-aware classification), and can choose which instances to label. Given its cardinality $|\mathcal{L}|$, the exponentially complex task of selecting $\mathcal{L}$ can be approximately tackled by greedily sampling one node per iteration t with index

$$k_t=\arg\max_{i\in\mathcal{U}^{t-1}} U(v_i,\mathcal{L}^{t-1}) \tag{8}$$

where $U(v,\mathcal{L}^{t-1})$ is a utility function that evaluates how informative node v is while taking into account information already available in $\mathcal{L}^{t-1}$. Upon acquiring the label y_{k_t}, it can

Algorithm 1 Active Graph Sampling Algorithm

Input: Adjacency matrix $\mathbf{W}$, $\delta \ll 1$
Initialize: $\mathcal{U}^0 = \mathcal{V}$, $\mathcal{L}^0 = \emptyset$, $\boldsymbol{\mu} = \mathbf{0}, \mathbf{G}_0 = (\mathbf{L} + \delta \mathbf{I})^{-1}$
First query is chosen at random
for $t = 1 : T$ **do**
  Scan $\mathcal{U}^{t-1}$ to find best query node v_{k_t} as in (8)
  Obtain label y_{k_t} of v_{k_t}
  Update the GMRF mean as in (9)
  Update $\mathbf{G}_t$ as in (10)
  $\mathcal{U}^t = \mathcal{U}^{t-1}/\{k_t\}$, $\mathcal{L}^t = \mathcal{L}^{t-1} \cup \{k_t\}$
end for
Predict remaining unlabeled nodes as in (7)

be shown that instead of re-solving (5), the GMRF mean is updated recursively using the "dongle node" trick in [12] as

$$\boldsymbol{\mu}^{+y_{k_t}}_{\mathcal{U}^{t-1}|\mathcal{L}^{t-1}} = \boldsymbol{\mu}_{\mathcal{U}^{t-1}|\mathcal{L}^{t-1}} + \frac{1}{g_{k_t k_t}} (y_{k_t} - [\boldsymbol{\mu}_{\mathcal{U}^{t-1}|\mathcal{L}^{t-1}}]_{k_t}) \mathbf{g}_{k_t} \tag{9}$$

where $\boldsymbol{\mu}^{+y_{k_t}}_{\mathcal{U}^{t-1}|\mathcal{L}^{t-1}}$ is the conditional mean of the unlabeled nodes when node v_{k_t} is assigned label y_{k_t} (thus "gravitating" the GMRF mean $[\boldsymbol{\mu}_{\mathcal{U}^{t-1}|\mathcal{L}^{t-1}}]_{k_t}$ toward its replacement y_{k_t}); vector $\mathbf{g}_{k_t} := [\mathbf{L}^{-1}_{\mathcal{U}^{t-1}\mathcal{U}^{t-1}}]_{:k_t}$ and scalar $g_{k_t k_t} := [\mathbf{L}^{-1}_{\mathcal{U}^{t-1}\mathcal{U}^{t-1}}]_{k_t k_t}$ are the k_t-th column and diagonal entry of the Laplacian inverse, respectively. Using Shur's lemma it can be shown that the inverse Laplacian $\mathbf{G}_t^{-k_t}$ when the k_t-th node is removed from the unlabeled sub-graph can be efficiently updated from $\mathbf{G}_t = \mathbf{L}^{-1}_{\mathcal{U}^t\mathcal{U}^t}$ as [11]

$$\begin{bmatrix} \mathbf{G}_t^{-k_t} & \mathbf{0} \\ \mathbf{0}^T & 0 \end{bmatrix} = \mathbf{G}_t - \frac{1}{g_{k_t k_t}} \mathbf{g}_{k_t} \mathbf{g}_{k_t}^T \tag{10}$$

which requires only $\mathcal{O}(|\mathcal{U}|^2)$ computations instead of $\mathcal{O}(|\mathcal{U}|^3)$ for matrix inversion. Alternatively, one may obtain $\mathbf{G}_t^{-k_t}$ by applying the matrix inversion lemma employed by the RLS-like solver in [12]. The resultant greedy active sampling scheme for graphs is summarized in Algorithm 1.

In summary, designing proper sampling strategies amounts to judiciously selecting the appropriate $U(v, \mathcal{L}^{t-1})$ in (8). In this context, we leverage the properties of GMRFs, in order to develop a novel data-adaptive active learning scheme that does not require retraining.

III. Expected model change

Here, we introduce a novel utility function based that relies on expected model change. From a high-level vantage point, the idea is to identify and sample nodes of the graph that are expected to have the greatest impact on the available GMRF model of the unknown labels. Thus, contrary to the expected error reduction and entropy minimization approaches that actively sample with the goal of increasing the "confidence" on the model, our focus is on effecting maximum perturbation of the model with each node sampled. The intuition is that by sampling nodes with large impact, one may take faster steps towards a model of satisfactory accuracy. Specifically, we propose sampling by maximizing the mean-square deviation (MSD) that a new sample is expected to inflict on the GMRF. The MSD between two rv's X_1 and X_2 is

$$\begin{aligned} \mathrm{MSD}(X_1, X_2) &:= \int (X_1 - X_2)^2 f(X_1, X_2) dX_1 dX_2 \\ &= \mathbb{E}\Big[(X_1 - X_2)^2\Big]. \end{aligned}$$

Our utility function is the expected MSD between $\boldsymbol{\psi}_{\mathcal{U}}$ and $\boldsymbol{\psi}^{+y_i}_{\mathcal{U}}$ before and after obtaining y_i; that is,

$$\begin{aligned} U_{MSD}(v_i, \mathcal{L}) &= \mathbb{E}_{y_i|\mathbf{y}_\mathcal{L}}\big[\mathrm{MSD}(\boldsymbol{\psi}^{+y_i}_{\mathcal{U}}, \boldsymbol{\psi}_{\mathcal{U}})\big] \\ &= \sum_{y_i \in \{-1,1\}} p(y_i|\mathbf{y}_\mathcal{L}) \big[\mathrm{MSD}(\boldsymbol{\psi}^{+y_i}_{\mathcal{U}}, \boldsymbol{\psi}_{\mathcal{U}})\big] \\ &\approx \frac{1}{2}(\mu_i + 1) \mathrm{MSD}(\boldsymbol{\psi}^{+y_i=1}_{\mathcal{U}}, \boldsymbol{\psi}_{\mathcal{U}}) \\ &\quad + \left[1 - \frac{1}{2}(\mu_i + 1)\right] \mathrm{MSD}(\boldsymbol{\psi}^{+y_i=-1}_{\mathcal{U}}, \boldsymbol{\psi}_{\mathcal{U}}) \end{aligned} \tag{11}$$

where

$$\begin{aligned} \mathrm{MSD}(\boldsymbol{\psi}^{+y_i}_{\mathcal{U}}, \boldsymbol{\psi}_{\mathcal{U}}) &:= \mathbb{E}\big[\|\boldsymbol{\psi}^{+y_i}_{\mathcal{U}} - \boldsymbol{\psi}_{\mathcal{U}}\|^2\big] \\ &= 2\mathrm{tr}(\mathbf{L}^{-1}_{\mathcal{U}\mathcal{U}}) + \|\boldsymbol{\mu}^{+y_i}_{\mathcal{U}|\mathcal{L}} - \boldsymbol{\mu}_{\mathcal{U}|\mathcal{L}}\|_2^2 \\ &\propto \frac{1}{g_{ii}^2}(y_i - \mu_i)^2 \|\mathbf{g}_i\|_2^2 \end{aligned} \tag{12}$$

where the term $2\mathrm{tr}(\mathbf{L}^{-1}_{\mathcal{U}\mathcal{U}})$ is ignored since it does not depend on y_i, and the final expression of (12) is obtained using (9); the proof of (12) is omitted due to space limitation. Finally, substituting (12) into (11) yields the following closed-form expression of the MSD-based utility score function

$$U_{MSD}(v_i, \mathcal{L}) \propto (1 - \mu_i^2) \frac{\|\mathbf{g}_i\|_2^2}{g_{ii}^2}. \tag{13}$$

It is worth mentioning that the MSD-based method subsumes the variance minimization method in [9]. This becomes apparent upon recalling that the variance-minimization utility score functions is given by $U_{VM}(v_i) := \|\mathbf{g}_i\|_2^2 / g_{ii}$. Then, further inspection reveals that the metrics are related by

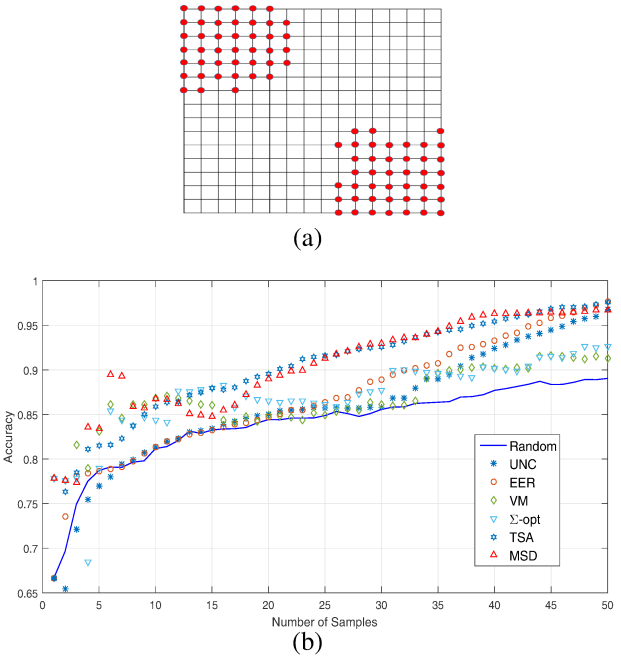
$$U_{MSD}(v_i) \propto \frac{1}{g_{ii}} (1 - \mu_i^2) U_{VM}(v_i).$$

In fact, U_{MSD} may be interpreted as a *data-driven* version of U_{VM} that is enhanced with the uncertainty term $(1 - \mu_i^2)$. On the one hand, $U_{\Sigma-opt}$ and U_{VM} are design-of-experiments-type methods that rely on ensemble criteria and offer *offline* sampling schemes more suitable for applications where the set $\mathcal{L}$ of nodes may *only* be labeled as a *batch*. On the other hand, U_{MSD} is an adaptive sampling scheme that adjusts to the *specific realization* of labels, and is expected to outperform batch alternatives in general. Note also that the results presented here for binary classification can be easily generalized to multiple classes. Nevertheless, before proceeding to numerical tests, an important modification is proposed in the ensuing section in order to deal with the challenge of *bias* that is inherent to all data-adaptive sampling schemes.

A. Incorporating model confidence

In the present section, we introduce a bias-reduction heuristic that is better tailored to the sampling strategy at hand. The main idea is to obtain expected change using a different set of probabilities than the ones provided by the model (cf. (6)). Specifically, we average over label predictions that are closer to an "non-informative" prior early on, and gradually converge to the ones provided by the trained model as our confidence on the latter increases. Thus, instead of taking the expectation in (11) over $p(y_i|\mathbf{y}_L)$, we instead use

$$\check{p}(y_i|\mathbf{y}_L; \alpha_t) = \alpha_t \pi(y_i) + (1 - \alpha_t) p(y_i|\mathbf{y}_L) \tag{14}$$



(a) (b)

Fig. 1. (a) Rectangular grid synthetic graph with two separate class 1 regions. (b) Test results for synthetic grid.

where $0 \leq \alpha_t \leq 1$ is a constant that quantifies the confidence on the current estimate of the posterior. If no prior is available, one may simply use $\pi(y_i = 1) = \pi(y_i = -1) = 1/2$. The resultant modified MSD utility function is

$$U_{MSD}(v_i, \mathcal{L}, a_t) \propto \left[0.5a_t + (1-a_t)(1-\mu_i^2)\right] \frac{\|\mathbf{g}_i\|_2^2}{g_{ii}^2} \qquad (15)$$

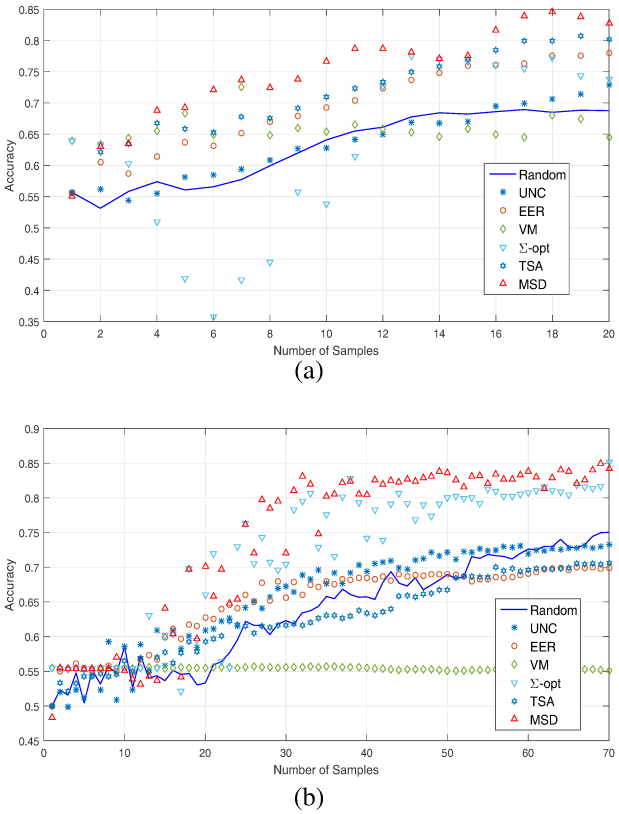
where a_t tunes the sensitivity of the sampling process to the uncertainty metric $(1-\mu_i^2)$. As more samples become available, the confidence that the current estimate of the posterior is close to the true distribution may increase. Thus, instead of using a constant α, one may use a sequence $\{\alpha_t\}_{t=1}^T$, where t is the iteration index, T the total number of samples, and a_t is *inversely proportional* to t.

IV. Numerical Tests

Here we report numerical tests to assess performance of the proposed method in terms of prediction accuracy $(|\mathcal{U}|^{-1} \sum_{i \in \mathcal{U}} \mathbf{1}_{\{\hat{y}_i = y_i\}})$ as a function of the number of nodes sampled by the GMRF-based active learning algorithms (cf. Algorithm 1). We compare the proposed MSD-based method with the variance minimization (VM) method [9], $\Sigma-$optimality method [11], expected error minimization (EER) [12], and two-step approximation method (TSA) [15]. Furthermore, we compare with a simple uncertainty sampling (UNC) scheme that samples the node with smallest difference between the predicted label probabilities, which is equivalent to using the scoring function $U_{UNC}(v_i, \mathcal{L}) := -|\mu_i|$. Finally, all methods are compared to the predictions that are given by a passive learning method based on random sampling.

A. Synthetic Experiments

The synthetic tests here are similar to those in [15]. First, a 10×10 rectangular grid plotted in Fig. 1a was adopted as a simple graph, where each node is connected to four neighboring nodes, except for the boundary nodes. Red dots correspond to nodes belonging to class 1, and uncolored intersections correspond to nodes belonging to class -1. To make the classification task more challenging, the class 1 region was separated into two 3×3 squares (upper left and lower right) and additional class 1 nodes were added w.p. 0.5 along the dividing lines. Plotted in Fig. 1b is the accuracy-vs-number of samples performance averaged over 50 Monte Carlo runs. As expected, most algorithms perform better than random sampling. In addition, one observes that purely exploratory non-adaptive methods (VM and $\Sigma-$optimality) enjoy relatively high accuracy for a small number of samples, but are eventually surpassed by adaptive methods. It can also be observed that the novel MSD method with $a_t = t^{-1/2}$ performs equally well to the state-of-the-art TSA method. Interestingly, it does so while using a much simpler criterion that avoids model retraining, and therefore requires shorter runtime.



(a) (b)

Fig. 2. Online classification tests on (a) Coloncancer dataset, and (b) Australian dataset.

B. Real Datasets

Real binary classification datasets taken from [26] and [27] were used for further testing of the proposed methods. First, each entry of the feature vectors was normalized such that it lies between -1 and 1. Then, a graph was constructed using the Pearson correlations among pairs of normalized feature vectors as weights of the adjacency matrix $\mathbf{W}$; thresholding was also applied to negative and small weights leading to sparse adjacency matrices. Plotted in Figs. 2a and 2b are the results of the numerical tests, where it is seen that the performance of the proposed low-complexity MSD-based scheme is comparable or superior to that of competing alternatives.

REFERENCES

[1] S. A Danziger, R. Baronio, L. Ho, L. Hall, K. Salmon, G. W. Hatfield, P. Kaiser, and R. H. Lathrop, "Predicting positive p53 cancer rescue regions using most informative positive (MIP) active learning," *PLOS Comput. Biol.*, vol. 5, no. 9, Sept. 2009.

[2] J. Haupt, R. M. Castro, and R. Nowak, "Distilled sensing: Adaptive sampling for sparse detection and estimation," *IEEE Trans. Info. Theory*, vol. 57, no. 9, pp. 6222–6235, 2011.

[3] Y. Fu, X. Zhu, and B. Li, "A survey on instance selection for active learning," *J. of Knowl. and Infor. Systems*, vol. 35, no. 2, pp. 249–283, 2013.

[4] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proc. of Conf. on Empirical Methods in Nat. Lang. Processing*, Waikiki, Honolulu, Oct. 2008.

[5] D. A. Cohn, Z. Ghahramani, and M. I Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, no. 1, pp. 129–145, 1996.

[6] G. V. Karanikolas, G. B. Giannakis, K. Slavakis, and R. M. Leahy, "Multi-kernel based nonlinear models for connectivity identification of brain networks," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Proc.*, Shanghai, China, March 2016.

[7] B. Baingana Y. Shen and G. B. Giannakis, "Kernel-based structural equation models for topology identification of directed networks," *IEEE Trans. Sig. Proc.*, vol. 65, no. 10, pp. 2503–2516, May 2017.

[8] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. of Intl. Conf. on Machine Learning*, Washington DC, Aug. 2003.

[9] M. Ji and J. Han, "A variance minimization criterion to active learning on graphs.," in *Intl. Conf. on Artif. Intel. and Stat.*, La Palma, Canary Islands, April 2012.

[10] Q. Gu and J. Han, "Towards active learning on graphs: An error bound minimization approach," in *Proc. of Intl. Conf. on Data Mining*, Brussels, Belgium, Dec. 2012.

[11] Y. Ma, R. Garnett, and J. Schneider, "$\sigma-$optimality for active learning on Gaussian random fields," in *Proc. of Adv. in Neural Inf. Proc. Systems*, Lake Tahoe, Dec. 2013.

[12] X. Zhu, J. Lafferty, and Z. Ghahramani, "Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. of Intl. Conf. on Machine Learning*, Washington DC, Aug. 2003.

[13] J. Long, J. Yin, W. Zhao, and E. Zhu, "Graph-based active learning based on label propagation," in *Intl. Conf. on Modeling Decisions for Artif. Intel.*, Catalonia, Spain, Oct. 2008.

[14] J. Zhou and S. Sun, "Active learning of Gaussian processes with manifold-preserving graph reduction," *J. of Neural Computing and Applications*, vol. 25, no. 7-8, pp. 1615–1625, 2014.

[15] K.-S. Jun and R. Nowak, "Graph-based active learning: A new look at expected error minimization," *arXiv preprint arXiv:1609.00845*, 2016.

[16] E. E. Gad, A. Gadde, A. S. Avestimehr, and A. Ortega, "Active learning on weighted graphs using adaptive and non-adaptive approaches," in *Proc. of Intl. Conf. on Acoustics, Speech and Signal Proc.*, Shanghai, China, March 2016.

[17] N. Cesa-Bianchi, C. Gentile, F. Vitale, and G. Zappella, "Active learning on trees and graphs," *arXiv preprint arXiv:1301.5112*, 2013.

[18] Q. Gu, C. Aggarwal, J. Liu, and J. Han, "Selective sampling on graphs for classification," in *Proc. of Intl. Conf. on Knowledge, Discovery and Data Mining*, Chicago, IL, Aug. 2013.

[19] K. Fujii and H. Kashima, "Budgeted stream-based active learning via adaptive submodular maximization," in *Proc. of Adv. in Neural Inf. Proc. Systems*, Barcelona, Spain, Dec. 2016.

[20] H. Su, Z. Yin, T. Kanade, and S. Huh, "Active sample selection and correction propagation on a gradually-augmented graph," in *Proc. of Conf. on Computer Vision and Pattern Recognition*, Boston, MA, June 2015, pp. 1975–1983.

[21] J. J. Pfeiffer III, J. Neville, and P. N. Bennett, "Active sampling of networks," in *Proc. of Intl. Work. on Mining and Learning with Graphs*, Edinburgh, Scotland, July 2012.

[22] A. Hauser and P. Bühlmann, "Two optimal strategies for active learning of causal models from interventions," in *Proc. of Europ. Work. on Prob. Graph. Models*, Granada, Spain, Sept. 2012.

[23] A. Vezhnevets, J. M. Buhmann, and V. Ferrari, "Active learning for semantic segmentation with expected change," in *Proc. of Conf. on Comp. Vision and Pattern Recog.*, Portland, OR, 2013.

[24] A. Freytag, E. Rodner, and J. Denzler, "Selecting influential examples: Active learning with expected model output changes," in *Proc of Europ. Conf. on Comp. Vision*, Zurich, Switzerland, Sept. 2014.

[25] S. M. Kay, *Fundamentals of Statistical Signal Processing, Vol. I: Estimation Theory*, Englewood Cliffs: Prentice Hall PTR, 1993.

[26] "https://archive.ics.uci.edu/ml/index.html," .

[27] "https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/binary.html," .