Blockwise Supervised Inference on Large Graphs

Kai Yu KAI.YU@SIEMENS.COM
Information and Communication, Corporate Technology, Siemens AG, Munich, Germany

Shipeng Yu SPYU@DBS.INFORMATIK.UNI-MUENCHEN.DE
Institute for Computer Science, University of Munich, Munich, Germany

Volker Tresp VOLKER.TRESP@SIEMENS.COM
Information and Communication, Corporate Technology, Siemens AG, Munich, Germany

Abstract

In this paper we consider supervised learning on large-scale graphs, which is highly demanding in terms of time and memory costs. We demonstrate that, if a graph has a bipartite structure that contains a small set of nodes separating the remaining from each other, the inference can be equivalently done over an induced graph connecting only the separators. Since each separator influences a certain neighborhood, the method essentially explores the block structure of graphs to improve the scalability. In the next step, instead of identifying the bipartite structure in a given graph, which is often difficult, we propose to construct a set of separators via two methods, one is adjacency matrix factorization and the other is mixture models, which both naturally ends up with a bipartite graph and meanwhile preserves the original data structure. Finally we report results of experiments on a toy problem and an intrusion detection problem.

1. Introduction

Recent years have seen considerable interests in graphs built on the pairwise relationships (e.g. similarity) of data objects. Though those relationships are rather local, the diffusion of information along with a graph often induces a global structure of the data distribution. One important area of exploring this global structure is to make use of unlabeled data in supervised learning, which is called semi-supervised learning.

In many supervised learning problems, collecting measurements is expensive, while vast amounts of unlabeled (or unmeasured) data are often readily available. These unlabeled data often offer some additional information, which is the situation where semi-supervised learning is useful. The graph based on the similarities of labeled and unlabeled data offers an elegant way to explore the additional information. Examples of recent work in this direction include Markov random walks (Szummer & Jaakkola, 2002), cluster kernels (Chapelle et al., 2003), regularization on graphs (Belkin & Niyogi, 2003; Zhu et al., 2003; Zhou et al., 2004), and directed graphs (Zhou et al., 2005).

In this paper we consider supervised learning on a large graph that corresponds to a large-size of data set. Since the number of unlabeled data is often very large, the situation is very common in real-world applications but highly demanding in terms of time and memory costs. Previous approaches rarely considered the scalability issue and typically handled at most thousands of data points. In this paper we take two steps to solve the problem. First, we demonstrate that, if a graph has a bipartite structure that contains a small set of nodes separating the remaining from each other, the inference can be equivalently done over an induced smaller graph connecting only the separators. Since each separator influences a certain neighborhood, the method essentially explores the block structure of graphs to improve the efficiency; Second, instead of identifying the bipartite structure in a given graph, which is often very difficult, we propose to construct a set of separators via two alternative ways, one is symmetric nonnegative factorization of the adjacency matrix and the other is mixture modeling, which both naturally ends up with a bipartite graph and meanwhile preserves the original data

Appearing in *Proc. of the 22st ICML Workshop on Learning with Partially Classified Training Data*, Bonn, Germany, August 2005. Copyright 2005 by the author(s).

structure. Once such a bipartite graph is constructed, supervised inference can be done efficiently with the computational cost $O(m^3)$, much smaller than $O(n^3)$ in the original graph. The benefits are more notable if various learning problems are frequently run on the same set of data, since the factorization needs to be done only once. Finally we report encouraging results of experiments on a toy problem and an intrusion detection problem.

The rest of this paper is organized as the following. In Sec. 2 we introduce the notion of harmonic functions. In Sec. 3 we describe an efficient learning algorithm on graphs which have a small set of separators to make up a bipartite structure. Then we explain how to produce a bipartite graph by factorization of adjacency matrix in Sec. 4. Finally we report the empirical study in Sec. 6 and conclude in Sec. 7.

2. Preliminaries

In this section we first review the notion of *harmonic functions* on general undirected graphs. Let $G(\mathbf{V}, \mathbf{E})$ be a graph with vertices $\mathbf{V}$ and edges $\mathbf{E}$, where $[i, j] \in \mathbf{E}$ denotes an edge from vertex v_i to v_j. An adjacency matrix $\mathbf{W} \in \mathbb{R}_+^{n\times n}$ describes the strengths of connections between vertices, satisfying $w_{i,j} = w_{j,i}$, $w_{i,j} \geq 0$, and $w_{i,j} = 0$ if $[i, j] \notin \mathbf{E}$. Let $d_i = \sum_j w_{i,j}$ be the degree of vertex v_i, and $\mathbf{D}$ a diagonal matrix with $(\mathbf{D})_{i,i} = d_i$. Then the combinatorial Laplacian is defined as $\triangle = \mathbf{D} - \mathbf{W}$, which operates on real-valued functions $\mathcal{H} = \{f : \mathbf{V} \to \mathbb{R}\}$ and yields

$$(\triangle f)_i = d_i f_i - \sum_j w_{i,j} f_j, \quad i = 1, \ldots, n.$$

$f \in \mathcal{H}$ is said to be a harmonic function if

$$\triangle f = (\mathbf{D} - \mathbf{W})f = 0. \tag{1}$$

The notion of harmonic functions is related to the potential theory (Kellogg, 1969) and important in many phenomena in physics. One can think G as an electrical circuit, f as voltages on vertices and $w_{i,j} = 1/R_{i,j}$ as the inverse resistance of edge $[i, j]$, based on Ohm's law the current entering v_j from v_i is $I_{i,j} = (f_i - f_j)/R_{i,j} = w_{i,j}(f_i - f_j)$. Kirchoff's current law says that the sum of incoming currents at any vertex in a circuit must equal to zero, namely

$$\sum_j I_{i,j} = \sum_j w_{i,j}(f_i - f_j) = (\triangle f)_i = 0.$$

In above situation we say that f is harmonic at vertex v_i. The property of harmonic functions has been explored in (Zhu et al., 2003) for supervised learning on general undirected graphs.

3. Inference on Bipartite Graphs

Now we consider a special type of graphs, called bipartite graph, which contains a set of vertices $\mathbf{Z}$ that separate the remaining vertices $\mathbf{V}$ from each other. The separators $\mathbf{Z}$ reflect the block structure of graphs. Our work is inspired by an intuition that block-wise algorithms should be faster than vertex-wise algorithms if the size of $\mathbf{Z}$ is small.

Formally, let $G(\mathbf{V}, \mathbf{Z}, \mathbf{E})$ be a bipartite graph with two vertex sets $\mathbf{V} = \{v_i\}_{i=1}^n$ and $\mathbf{Z} = \{z_k\}_{k=1}^m$, and edges $[i, k] \in \mathbf{E}$ connecting v_i to z_k. Note that there is no intra connections within $\mathbf{V}$ or $\mathbf{Z}$. Let $\mathbf{A} \in \mathbb{R}_+^{n\times m}$ be the adjacency matrix such that $a_{i,k}$ describes the strength of the connection between v_i and z_k, satisfying $a_{i,k} = a_{k,i}$, $a_{i,k} \geq 0$, and $a_{i,k} = 0$ if $[i, k] \notin \mathbf{E}$. Let $d_i^v = \sum_k a_{i,k}$ be the degree of vertex v_i, $d_k^z = \sum_i a_{i,k}$ the degree of vertex z_k, and $\mathbf{D}_v$ and $\mathbf{D}_z$ the corresponding diagonal matrices.

Let $\mathcal{H}_v = \{f : \mathbf{V} \to \mathbb{R}\}$ and $\mathcal{H}_z = \{g : \mathbf{Z} \to \mathbb{R}\}$ be the two spaces of real-valued functions. Based on the harmonic property that function values on each vertex equals to the weighted average of function values on neighbor vertices, if f and g are both harmonic there are relationships

$$f = \mathbf{D}_v^{-1}\mathbf{A}g \quad \text{and} \quad g = \mathbf{D}_z^{-1}\mathbf{A}^\top f.$$

Combining the two equations, it is not difficult to have the forms similar to Eq. (1)

$$(\mathbf{D}_v - \mathbf{A}\mathbf{D}_z^{-1}\mathbf{A}^\top)f = 0$$
$$(\mathbf{D}_z - \mathbf{A}^\top\mathbf{D}_v^{-1}\mathbf{A})g = 0$$

Then we define two combinatorial Laplacians $\triangle_v : \mathcal{H}_v \to \mathbb{R}_+^n$ and $\triangle_z : \mathcal{H}_z \to \mathbb{R}_+^m$ respectively as

$$\triangle_v = \mathbf{D}_v - \mathbf{A}\mathbf{D}_z^{-1}\mathbf{A}^\top \tag{2}$$
$$\triangle_z = \mathbf{D}_z - \mathbf{A}^\top\mathbf{D}_v^{-1}\mathbf{A} \tag{3}$$

3.1. Vertexwise Inference

Now we consider the semi-supervised problem on bipartite graphs. Without loss of generality, we assume that f is partially observed and want to estimate the whole function f. The measurements are referred as boundary conditions in the potential theory. To state formally, let $\mathbf{y}_l$ be the measurements of f on a subset $\mathbf{V}_l = \{v_i\}_{i=1}^{n_l}$, and $\mathbf{V}_u = \mathbf{V} - \mathbf{V}_l$ the set of unmeasured vertices. The boundary condition is *one side* since there is no observations on g.

It is usually impossible to directly probe into a physical process and exactly obtain the target quantity.

To underlie this, we attach to each $v_i \in \mathbf{V}_l$ an extra *boundary vertex* $b_i \in \mathbf{B}$, where the measuring truly takes place, and assign a weight a_0 to the edge connecting b_i and v_i. Intuitively a_0 is large if the $\mathbf{y}_l$ are precise. Then the harmonic property for f should be

$$f_i = \begin{cases} \frac{1}{d_i^v + a_0}\left(\sum_k a_{i,k} g_k + a_0 y_i\right), & \text{if } v_i \in \mathbf{V}_l \\ \frac{1}{d_i^v} \sum_k a_{i,k} g_k, & \text{if } v_i \in \mathbf{V}_u \end{cases} \tag{4}$$

The above function can be easily justified again by the Kirchoff's laws, where b_i are the instruments measuring the voltages on v_i, a_0 stands for the inverse of some resistance caused by, for example, the contact resistance between b_i and v_i, and y_i are the voltage values read on b_i. The harmonic property for g remains the same, but can be written differently

$$g_k = \frac{1}{d_k^u} \sum_i a_{i,k} f_i \tag{5}$$

Insert Eq. (5) into Eq. (4), one obtains

$$f_i = \begin{cases} \frac{1}{d_i^v + a_0}\left(\mathbf{a}_i \mathbf{D}_z^{-1} \mathbf{A}^\top f + a_0 y_i\right), & \text{if } v_i \in \mathbf{V}_l \\ \frac{1}{d_i^v} \mathbf{a}_i \mathbf{D}_z^{-1} \mathbf{A}^\top f, & \text{if } v_i \in \mathbf{V}_u \end{cases} \tag{6}$$

where $\mathbf{a}_i$ is the i-th row vector of $\mathbf{A}$.

Theorem 3.1. *Eq. (6) is the sufficient and necessary condition of $f^* \in \mathcal{H}_v$, which solves the variational problem*

$$f^* = \arg\min_f \|\mathbf{f}_l - \mathbf{y}_l\|^2 + \lambda f^\top \triangle_v f \tag{7}$$

where $\mathbf{f}_l$ are f's values on $\mathbf{V}_l$, and $\lambda = \frac{1}{a_0}$.

Proof. Let

$$f = \begin{bmatrix} \mathbf{f}_l \\ \mathbf{f}_u \end{bmatrix} \text{ and } \triangle_v = \begin{bmatrix} \triangle_v^{ll} & \triangle_v^{lu} \\ \triangle_v^{ul} & \triangle_v^{uu} \end{bmatrix}.$$

The partial derivative of the cost in Eq. (7) with respect to f gives

$$\mathbf{f}_l - \mathbf{y}_l + \lambda \triangle_v^{ll} \mathbf{f}_l + \lambda \triangle_v^{lu} \mathbf{f}_u = 0 \tag{8}$$

$$\triangle_v^{ul} \mathbf{f}_l + \triangle_v^{uu} \mathbf{f}_u = 0 \tag{9}$$

where Eq. (8) easily drives the first part of Eq. (6) if $\lambda = \frac{1}{a_0}$ and Eq. (9) gives the second part. Thus Eq. (6) is the necessary condition. The convexity of the optimization problem Eq. (7) further suggests that Eq. (6) is also the sufficient condition. □

Theorem 3.1 indicates that the harmonic function under the one-side boundary condition is the solution to a *regularized regression problem* with the regularizer $f^\top \triangle_v f$ and square loss. The regularization ensures f to be sufficiently smooth with respect to the bipartite graph. Based on Eq. (8) and Eq. (9), the estimated function is given by

$$f^* = \begin{bmatrix} I + \lambda \triangle_v^{ll} & \lambda \triangle_v^{lu} \\ \lambda \triangle_v^{ul} & \lambda \triangle_v^{uu} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y}_l \\ 0 \end{bmatrix} \tag{10}$$

3.2. Blockwise Inference

In case $m \ll n$, it is much more efficient to deal with g than f. Since each separator z_k influences a certain neighborhood of vertices v_i, we essentially explore the block structure of graphs. Inserting Eq. (4) into Eq. (5), we obtain

$$g = \mathbf{D}_z^{-1} \mathbf{A}^\top \begin{bmatrix} a_0 \mathbf{I} + \mathbf{D}_v^{ll} & 0 \\ 0 & \mathbf{D}_v^{uu} \end{bmatrix}^{-1} \mathbf{A} g + \mathbf{D}_z^{-1} \mathbf{g}' \tag{11}$$

where $\mathbf{D}_v^{ll}$ and $\mathbf{D}_v^{uu}$ are respectively the blocks in $\mathbf{D}_v$ corresponding to $\mathbf{V}_l$ and $\mathbf{V}_u$, and

$$\mathbf{g}' = \mathbf{A}_l^\top \left[a_0 \mathbf{I} + \mathbf{D}_v^{ll}\right]^{-1} a_0 \mathbf{y}_l \tag{12}$$

Eq. (11) can be rewritten as

$$(\mathbf{D}_z - \mathbf{A}'_z) g = \mathbf{g}' \tag{13}$$

where

$$\mathbf{A}'_z = \mathbf{A}^\top \begin{bmatrix} a_0 \mathbf{I} + \mathbf{D}_v^{ll} & 0 \\ 0 & \mathbf{D}_v^{uu} \end{bmatrix}^{-1} \mathbf{A} \tag{14}$$

Finally we get the estimate of g as

$$g^* = (\mathbf{D}_z - \mathbf{A}'_z)^{-1} \mathbf{g}' \tag{15}$$

Note that it is easy to check that $(\mathbf{D}_z - \mathbf{A}'_z)$ is positive definite, thus its inverse exists. The the estimate of f is given by

$$f^* = \mathbf{D}_v^{-1} \mathbf{A} g^* \tag{16}$$

Compared with the solution Eq. (10), the computation here is much faster when $m \ll n$, since we only need to invert a much smaller matrix.

4. Adjacency Matrix Factorization

In Section 3 we discussed how to make inference on the bipartite graph $G(\mathbf{V}, \mathbf{Z}, \mathbf{E})$. In this section we investigate how to construct such a bipartite graph based on a general undirected graph.

4.1. Symmetric Nonnegative Matrix Factorization

Let us follow the notation in Section 2 with graph $G(\mathbf{V},\mathbf{E})$ and adjacency matrix $\mathbf{W} \in \mathbb{R}_+^{n\times n}$. In the graph factorization problem, we aim to seek m ($m < n$) *latent nodes* $\{z_k\}_{k=1}^m = \mathbf{U}$ and build the bipartite graph with adjacency matrix $\mathbf{A} \in \mathbb{R}_+^{n\times m}$, which approximate the adjacency matrix $\mathbf{W}$ defined between pairs (v_i, v_j):

$$\mathbf{w}_{ij} \approx \sum_{k=1}^{m} \frac{a_{ik}a_{jk}}{d_k^z} = \mathbf{A}\mathbf{D}_z^{-1}\mathbf{A}^\top,$$

where $d_k^z = \sum_{l=1}^n a_{lk}$ sum over the kth column of $\mathbf{A}$. The idea behind this formulation is that the $n \times m$ matrix $\mathbf{A}$ accounts for the adjacency relationships given in $\mathbf{W}$. For a concrete example, if the nodes in $\mathbf{V}$ are text documents, latent nodes z could be the latent topics that underlie the graph and connect documents.

Mathematically, we are solving the following optimization problem:

$$\min_{\mathbf{A}\in\mathbb{R}_+^{n\times m}} \quad l(\mathbf{W}, \mathbf{A}\mathbf{D}_z^{-1}\mathbf{A}^\top) \tag{17}$$

$$s.t. \quad h_{ik} \geq 0, \mathbf{D}_z = \mathrm{diag}(d_1^z, \ldots, d_m^z), d_k^z = \sum_{i=1}^n h_{ik},$$

where $l(\cdot,\cdot)$ defines a distance function for two matrices, e.g., $l(\mathbf{P},\mathbf{Q}) = \|\mathbf{P}-\mathbf{Q}\|_F^2$. Denote $\mathbf{H} = \mathbf{A}\mathbf{D}_z^{-1/2}$ as the *adjacency factor* of graph G, we can change Eq. (17) into the following equivalent problem:

$$\min_{\mathbf{H}\in\mathbb{R}_+^{n\times m}} \quad l(\mathbf{W}, \mathbf{H}\mathbf{H}^\top) \tag{18}$$

$$s.t. \quad h_{ik} \geq 0.$$

This is seen as a nonnegative matrix factorization (NMF) problem (Lee & Seung, 2000), and is not convex in $\mathbf{H}$. Various numerical methods can be used here to find a local minima for this problem, and in this paper we focus on a special gradient descent method for two kinds of distance functions:

Theorem 4.1. *(i) If the distance in Eq. (18) is Frobenius norm $l(\mathbf{P},\mathbf{Q}) = \|\mathbf{P}-\mathbf{Q}\|_F^2$, the distance is non-increasing under the update rule*

$$\tilde{h}_{ik} = h_{ik}\frac{(\mathbf{W}\mathbf{H})_{ik}}{(\mathbf{H}\mathbf{H}^\top\mathbf{H})_{ik}}. \tag{19}$$

(ii) If the distance in Eq. (18) is divergence $l(\mathbf{P},\mathbf{Q}) = \sum_{ij}\left(p_{ij}\log\frac{p_{ij}}{q_{ij}} - p_{ij} + q_{ij}\right)$, the distance is non-increasing under the update rule

$$\tilde{h}_{ik} = \frac{h_{ik}}{\sum_j h_{jk}} \sum_j \frac{w_{ij}}{(\mathbf{H}\mathbf{H}^\top)_{ij}} h_{jk}. \tag{20}$$

In both cases, the distance is invariant under the update if and only if $\mathbf{H}$ is at a stationary point of the distance.

The theorem provides a method of symmetric nonnegative matrix factorization (SNMF), which can be derived via a modification to NMF in (Lee & Seung, 2000). For both objective functions, the update is a gradient descent method with a specific step size. It can be easily checked that the update is unity when $\mathbf{W} = \mathbf{H}\mathbf{H}^\top$. Due to nonnegative constraints, a graph is decomposed to a set of additive components, which represents clusters or blocks on the graph.

After $\mathbf{H}$ is obtained, adjacency matrix $\mathbf{A}$ can be easily calculated as $\mathbf{A} = \mathbf{H}\mathbf{\Lambda}$, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_m)$ and $\lambda_k = \sum_{i=1}^n h_{ik}$. This is easily checked by equating $\sum_{i=1}^n h_{ik} = \sum_{i=1}^n h_{ik}/\sqrt{d_k^z} = \sqrt{d_k^z}$ since $\mathbf{H} = \mathbf{A}\mathbf{D}^{-1/2}$.

While both of the distance functions in Theorem 4.1 can be applied, we prefer the divergence function because in Eq. (20) we only need to sum over all nonzero terms of $\mathbf{w}_{ij}$ for a new update. This is extremely efficient if $\mathbf{W}$ is sparse, and the time complexity of Eq. (20) is $\mathcal{O}(m^2 nL)$ with L the number of nonzero entries in $\mathbf{W}$.

5. Inference with Mixture Models

Interestingly, a mixture density model has a natural bipartite structure, which implies a combination of the results from Sec. 3 and Sec. 4 with mixture models, and offers a principled way to handle new data points inductively .

Let $\mathbf{X} = \{x_i\}_{i=1}^n$ be the set of samples that are generated from a mixture density

$$p(x) = \sum_{k=1}^{m} \pi_k p(x|\theta_k) \tag{21}$$

where π_k are the probability mass of components c_k, satisfying $\sum_k \pi_k = 1$, θ_k are the parameters of c_k, and $p(x|\theta_k)$ are the conditional density of a random sample x given c_k.

To build the connection to bipartite graphs, we treat samples x_i as vertices v_i, components c_k as vertices z_k, and the adjacency matrix $\mathbf{A}$ as

$$a_{i,k} = p(x_i, c_k) = p(x_i|\theta_k)\pi_k \tag{22}$$

A mixture model also induces a similarity measure between data objects. To see this, let $w_{i,j}$ between v_i and v_j be the marginalized kernel (Tsuda et al., 2002),

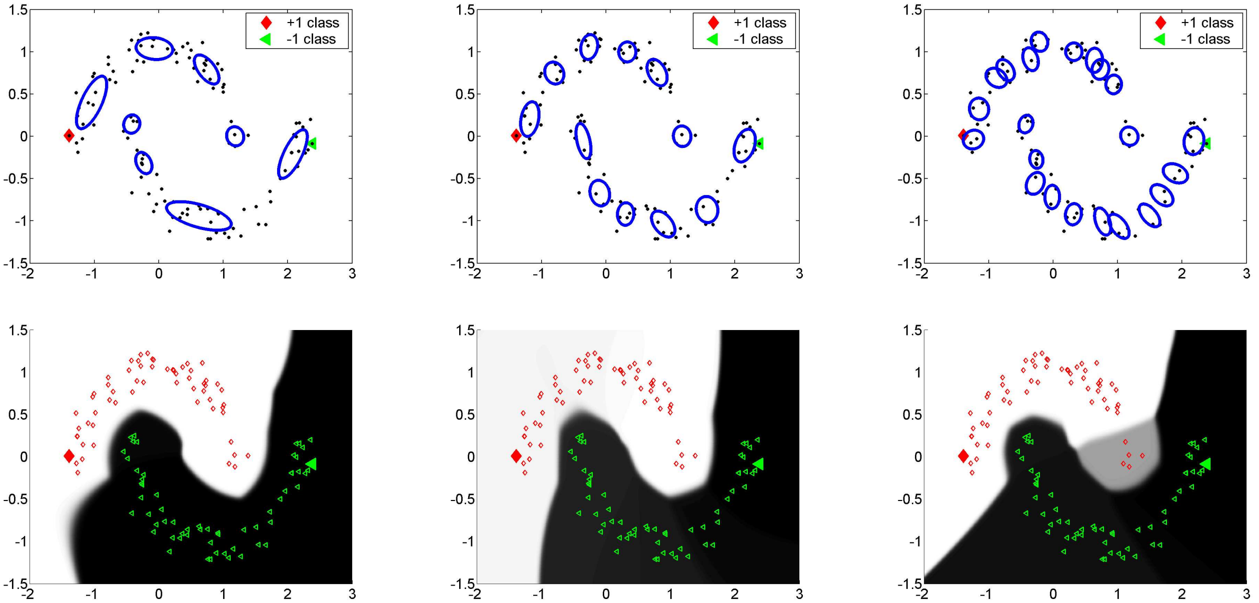

Figure 1. Semi-supervised learning on the two-moon data: mixture models (upper rows) and learned functions (lower rows). Three results are obtained with 8 (left), 12 (middle) and 22 (right) components. For each Gaussian component the covariance structure is plotted. The learned functions show the classification decisions on the test points (with colors and markers) and function values in the whole input space (with gray levels).

then we have

$$\begin{aligned} w_{i,j} = p(v_i, v_j) &= \sum_k p(x_i|\theta_k)p(x_j|\theta_k)\pi_k \qquad (23) \\ &= \sum_k \frac{p(x_i, c_k)p(x_j, c_k)}{\pi_k} = (\mathbf{A}\mathbf{D}_z^{-1}\mathbf{A}^\top)_{i,j} \end{aligned}$$

where $(\mathbf{D}_z)_k = d_k^z = \pi_k$ are the marginal probabilities of c_k. Furthermore, based on Eq. (21) it is not difficult to see that the degree of vertex v_i corresponds to the density of x_i

$$d_i^v = p(x_i) = \sum_k a_{i,k} = \sum_j w_{i,j}. \tag{24}$$

The connections between mixture models and bipartite graphs can also be explained from the *random walk* point of view, where the conditional probabilities correspond to the transition probabilities.

The messages obtained from the above analysis are mainly two folds. First, a mixture model can be treated as a bipartite graph. Then the learning algorithms built on bipartite graphs in Sec. 3 can be directly applied on mixture models, which actually explores the structure of the input density in supervised learning; Second, a bipartite graph can be parameterized into a mixture model. Since the posterior probabilities $p(c_k|x_i) = w_{i,k}/d_i^v$ are given, the whole graph factorization can be seen as a E-step of an EM algorithm. At the next, we just need to perform an M step to estimate the parameters θ_k of the mixture model. In practice, one can either directly estimate a mixture model based on labeled and unlabeled data, or first factorize the similarity graph of data and then fit a mixture model. We found that the second choice works better when the dimensionality of data is high.

As an advantage of mixture models, an inductive learning algorithm can be derived. For any new test point x, its joint probability with each component c_k can be easily computed. While for a bipartite graph, computing $w(\mathbf{x}, c_k)$ requires an iterative procedure. Based on Eq. (16) the prediction is given by

$$f^*(x) = \frac{1}{p(x)} \sum_k p(x, c_k)g^*(c_k) = \sum_k p(c_k|x)g^*(c_k) \tag{25}$$

6. Experiments

6.1. The Two-Moon Problem

We test the proposed algorithm on the two-moon data set (Zhou et al., 2004). Mixture of gaussian models are trained to model the density distribution of input data. We used the variational Bayesian methods described in (Yu et al., 2005), the freedom of birthing new compo-

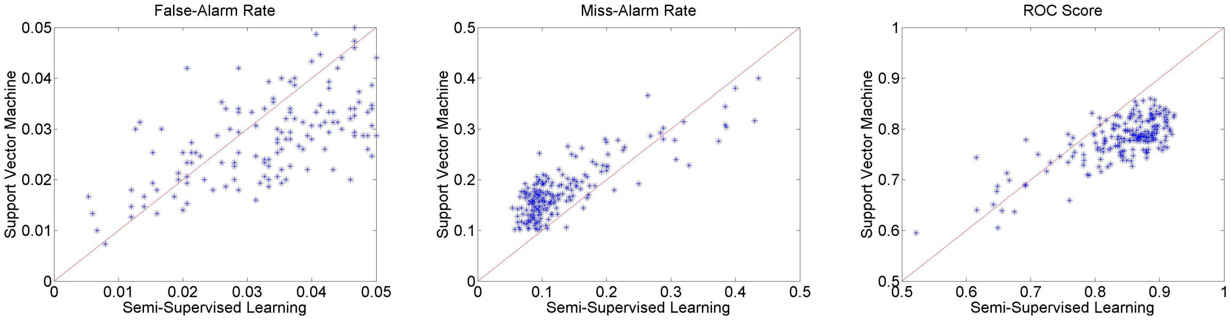

Figure 2. 200 random repeats of intrusion detection with DARPA data. Each repeat corresponds a spot.

nents in data fitting is dependent on a hyper parameter. Instead of optimizing the hyper-parameters using the evidence framework in (Yu et al., 2005), we set different values to see the sensitivity of semi-supervised learning with respect to the number of mixture components. The results are shown in Fig. 1, where for each class one label is given. Three results are obtained with 8 (left), 12 (middle) and 22 (right) components. For each Gaussian component the covariance structure is plotted. The learned functions show the classification decisions on the test points (with colors and markers) and function values in the whole input space (with gray levels). In all the cases classification results are quite good, indicating that the number of components m is not very critical to the performance. Actually as m increases, the model is approaching to the graph-based algorithms like (Zhu et al., 2003) and (Zhou et al., 2004). Interestingly, in the last case where $m = 22$, a gray area indicates that a small group of data points are somewhat different from others.

6.2. Intrusion Detection

In the last experiment, we test on an intrusion detection problem based on the KDDCup 1999 data set. The data set consist of connection record data collected in 1998 DARPA IDS (intrusion detection systems) evaluation. The data applied here was from (Pavel et al., 2004) and contains 50,000 data points with 500 attacks that belong to 37 intrusion types. We are only interested in finding the attacks thus the problem becomes a binary classification task. In each run, we randomly pick up $l \in [37, 200]$ intrusions, at least one for each class, and select 20,000 unlabeled points. Usually semi-supervised learning algorithms do not scale to such a big sample size. Our algorithm is trained on the selected labeled and unlabeled data and used to predict all the unlabeled points. The prediction accuracy is evaluated by false-alarm rate, miss-alarm rate, and ROC score. The experiments are repeated by 200 times and spot plots are shown in Fig. 2. Based on the false-alarm rate the semi-supervised algorithm performs similar as support vector machines, while in terms of miss-alarm rate and ROC score our algorithm performs much better than support vector machines.

7. Conclusion and Future Work

In this paper we proposed a blockwise supervised learning approach on undirected graphs, which explores the bipartite structure of graphs. Compared to previous algorithms on graphs, the proposed algorithm simulates the information diffusion over blocks instead of vertices and is thus much more scalable to large graphs. We also proposed two approaches to construct bipartite graphs for capturing the distribution structure of data, one is to approximate an arbitrary undirected graph via nonnegative adjacency matrix factorization, and the other is mixture density modeling. The algorithm was initially tested on a toy problem and an intrusion detection data. In the near future, more empirical study should be done. Also, it is desired to investigate how to choose the size of clusters in order to obtain a good bipartite approximation.

Acknowledgments

Thanks to Dr. DengYong Zhou for the fruitful discussion when the first author was visiting Max-Plank-Institute at Tuebingen. Also, thanks for the reviewers for constructive comments.

References

Belkin, M., & Niyogi, P. (2003). Using manifold structure for partially labeled classification. *Advances in Neural Information Processing Systems 15*. MIT Press.

Chapelle, O., Weston, J., & Schölkopf, B. (2003). Cluster kernels for semi-supervised learning. *Advances* *in Neural Information Processing Systems 15*. MIT Press.

Kellogg, O. D. (1969). *Foundations of potential theory*. Dover Publications.

Lee, D. D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems 13* (pp. 556–562).

Pavel, L., Christin, S., & Igor, K. (2004). Intrusion detection in unlabeled data with quarter-sphere support-vector machines. *DIMVA'2004*.

Szummer, M., & Jaakkola, T. (2002). Partially labeled classification with markov random walks. *Advances in Neural Information Processing Systems 14*. MIT Press.

Tsuda, K., Kin, T., & Asai, K. (2002). Marginalized kernels for biological sequences. *Bioinformatics*, 268s–275s.

Yu, S., Yu, K., & Tresp, V. (2005). *Variational bayesian dirichlet-multinomial allocation for mixture of exponential family distributions*. manuscript.

Zhou, D., Bousquet, O., Lal, T., Weston, J., & Schölkopf, B. (2004). Learning with local and global consistency. *Advances in Neural Information Processing Systems 16*. MIT Press.

Zhou, D., Schlkopf, B., & Hofmann, T. (2005). Semi-supervised learning on directed graphs. *Advances in Neural Information Processing Systems 17*.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using gaussian fields and harmonic functions. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*.