

# Aligned Graph Classification with Regularized Logistic Regression

Brian Quanz, Jun Huan  
Information and Telecommunication Technology Center  
Department of Electrical Engineering and Computer Science  
University of Kansas  
Lawrence, Kansas, 66045  
{bquanz,jhuan}@ku.edu

## Abstract

Data with intrinsic feature relationships are becoming abundant in many applications including bioinformatics and sensor network analysis. In this paper we consider a classification problem where there is a fixed and known binary relation defined on the features of a set of multivariate random variables. We formalize such a problem as an aligned graph classification problem. By incorporating this feature relationship in the learning process we aim to obtain improved classification performance over conventional learning that does not consider the additional information of the feature relationship. To incorporate the feature relationship, we extend logistic regression and use a regularization term that includes the normalized Laplacian of the graph, similar to the  $L_2$  regularization, deriving a modified optimization problem and solution. We demonstrate the effectiveness of our method and compare it to other methods using simulated and real data sets.

## 1 Introduction

Consider a  $p$ -dimensional multivariate random variable  $X = (x_1, x_2, \dots, x_p) \in \mathbb{R}^p$  where there are some known relationships for the features in  $X$ . We investigate the problem of performing effective supervised learning to build accurate classification models for mapping such random variables to class labels, based on observed samples and the relation of the features.

Data with intrinsic feature relationships are becoming abundant in many application domains such as bioinformatics, sensor networks, and social networks among others. For instance, in pathway-based microarray classification, a biological network contains a set of genes, taking values based on their expression levels, and there is a known binary relation of genes: the pathway topology [15, 21]. In this case the goal of the data analysis is to use the expression data to predict a measurable outcome, such as the presence or absence of a disease. In sensor networks, there has been a burgeoning interest in incorporating sensors in everyday life to

monitor the environment, supply information, and ensure security. At a given time point regarding the state of the full sensor network, the features are the readings of the sensors, and we usually know the topology or the physical location of the sensors in relation to each other. The goal of the analysis is to detect events of interest based on the collective values of the sensors in the network.

Exploring the relationship between features is not new. Recently in structured feature selection, supervised learning algorithms have been explored for data sets where features have some natural “structure” relationships [27, 28, 29, 30, 31]. For example Yuan and Lin explored the situation where features may be naturally partitioned into groups and studied the regression problem of grouped features using a technique called grouped Lasso [28]. Another possible type of structure relationship of features is a hierarchical relation (i.e. a directed acyclic graph defined on features) and that has been explored in [27, 30]. In [29], both group structure and hierarchical relation have been studied in a unified framework. Recently Kim and Xing assumed that all the features fit into a linear chain (e.g. genes in a chromosome) and have studied regression problems for such data sets [13]. All these studies, however, do not consider the general case where a general undirected graph is defined to capture the structure relationship of features for classification and regression.

In this paper we extend previous work on structured feature selection and investigate the new classification problem where features of a data set have a natural graph relationship. We assume such relationships are known and fixed among all instances of the data set. We call such a problem an *aligned graph classification problem* where we may use a graph to model a datum, vertices represent features, edges represent binary relation between features, and vertex and edge set remains the same across a set of samples. Specifically we formalize our classification problem below.

**Problem Statement: the Aligned Graph Classification Problem.** Given a random variable  $X = (x_1, x_2, \dots, x_p) \in \mathbb{R}^p$ , a graph  $G$  is a *feature relationship graph* of  $X$  if the vertex set of  $G$  is the  $p$  features. Given a set of  $n$  observations  $\{(X_i, y_i)\}$ ,  $X_i \in \mathcal{X} \subset \mathbb{R}^p$ ,  $y_i \in \mathcal{Y} = \{1, 2, \dots, K\}$ ,  $K \in \mathbb{N}$ ,  $i \in [1, n]$ , and a feature relationship graph, the *aligned graph classification problem* is to build a classification model  $f: \mathcal{X} \rightarrow \mathcal{Y}$  to assign class labels to unseen random variables in  $\mathcal{X}$  to minimize expected loss. To simplify discussion, from here on, we restrict  $\mathcal{Y} = \{1, 2\}$  to the binary class case, 0-1 loss function (i.e. 1 if  $y = f(x)$  and 0 otherwise), and undirected feature relationship graphs. Furthermore, we restrict the feature relationship graph structure to be fixed across the set of observations. In other words, the relationship between features is fixed and thus the edges defined between features are fixed for the aligned graphs, each graph will have the same set of edges but possibly different, but aligned, vertex labels, given by the value the random variable takes for that observation.

One way to perform aligned graph classification is to simply use traditional supervised classification algorithms that do not consider the fixed graph structure represented by the feature relationships. By incorporating the graph structure information along with the vertex labels (feature values) in the classification model construction the aim is to improve predictive performance over methods that only consider the feature values for a given observation. Another approach for aligned graph classification that might be considered is to use graph kernel functions for classification [10]. Graph kernels map a set of data to a high dimensional Hilbert space without explicitly computing the coordinates of the data. Coupled with kernel machines such as support vector machines, graph kernel methods can be used for tasks include classification [26], regression [6] and feature extraction through principle component analysis [23]. The adoption of existing graph kernels for aligned graphs, however, is not straightforward for two major reasons: (i) most current graph kernels assume discrete node labels and aligned graphs have numeric node labels and (ii) most current graph kernels measure the difference of graph structures while the graph structures do not change in the aligned graph data.

In this paper instead of exploring graph kernel methods, we adopt the framework of logistic regression and extend the work from numeric data to data with an intrinsic graph structure using regularization. Logistic regression is a popular statistical method for classification that works by modeling conditional probability distributions using a log-linear model and identifying parameters that maximize the log likelihood of the data,

and has been successfully applied to many problems [9, 16]. Comparing to other classification algorithms, logistic regression has the benefits of probabilistic outputs - the probability of a label is returned as opposed to only a discrete class label - and a straight-forward generalization from the binary classification case to the multi-class case. In addition, logistic regression tolerates missing values in data [17]. Many improvements have been proposed and the two most significant ones are (i) adding regularization to the objective function and (ii) applying logistic regression in a kernel space. Incorporating a regularization term that penalizes the square of the  $L_2$  norm of the parameters has been seen to improve the predictive performance of the method particularly for high-dimensional and highly-correlated data [3], following the same idea as ridge regression [11] in which, by penalizing the  $L_2$  norm of the parameters, reduced generalization error can be achieved by shrinking the prediction variance at the cost of increasing bias.

Here, we extend the  $L_2$  regularized logistic regression with a straight-forward modification of the objective function that allows the model learning to be regularized with respect to the graph structure. The basic idea is to force the parameters to vary smoothly over the graph, the idea being quite similar to recent work in semi-supervised learning. The structure of a similarity graph is incorporated in the learning framework in the form of the Laplacian of the graph; the Laplacian of the graph is used in unsupervised (e.g. [25]) and transductive and semi-supervised learning (e.g. [1, 32]) when such a similarity structure exists between the data samples. We pursue a similar idea; to improve prediction we incorporate additional information in the form of the graph structure relating the variables and enforce a smooth parameter variation over the graph structure for the variables by means of regularization. The idea should be of particular interest when less labeled information is available, i.e. for small sample data sets or data sets where the ratio of the number of samples to the dimensionality of the data is small.

In summary, our contributions in this paper are

- We formalized the aligned graph classification problem for data set where features have a natural structure relationship.
- We extended the logistic regression to include the normalized graph Laplacian, incorporating the Laplacian in the regularization term. We showed that this results in a simple modification to the original logistic regression solution and update using the efficient newton-raphson approach for finding the zeros of the gradient.
- We developed an approach to incorporate the graph

Laplacian regularization in *kernel logistic regression*, which uses a basis expansion to allow non-linear functions of the variables, similar to support vector machines.

- We performed a comprehensive experimental evaluation, showed that Laplacian regularized logistic regression is an effective method for incorporating the graph structure in the prediction problem, evaluated these methods on synthetic and real world data sets and compared the performance of the methods to competing methods including support vector machines and unregularized logistic regression.

The rest of the paper is organized in the following way. Section 2 discusses related work. Section 3 presents background information and detailed discussion of our algorithms. Section 4 presents the experimental study of our algorithms as compared to competing methods. Finally we give a short conclusion and a discussion of the future work.

## 2 Related Work

We use logistic regression as our framework for building classification models for aligned graph classification; logistic regression has also been used extensively for scientific data analysis. For example, sparse logistic regression was proposed to perform gene selection in [24], a partial least squares with penalized logistic regression algorithm was proposed for high-dimensional, small-sample problems in [7], and in [16] logistic regression is used for feature selection. The approach of [24] has been recently improved in [2] using Bayesian regularization, and applied to the problem of cancer classification, and an  $L_2$  penalized logistic regression method for classification was proposed in [31].

In bioinformatics research there has recently been much interest in using computational methods to associate groups of genes such as groups defined by biological pathways (graphs) with a clinical outcome such as a disease. For example, a statistical method for determining if a group of genes is significantly related to a clinical outcome by calculating a p-value for the group was proposed in [8]. Another statistical test, the Multi-dimensional Cluster Misclassification test (MCM-test), was proposed in [15] for associating pathways with disease outcomes by modeling expression values for a group of genes as fuzzy sets for each outcome and using the membership of the genes in the fuzzy sets to determine significance. For the similar problem of selecting significant pathways and performing classification, a random forest approach was proposed in [20]. For the problem of detecting gene-gene interaction, an  $L_2$  regularized lo-

gistic regression method was proposed in [21].

Our work is different from existing work in that we use a general graph to capture relationship between features. In our method we consider a graph as a manifold and we factor in the graph topology using graph Laplacian as a regularization factor. Hence the key insight is that the conditional probability distribution, as evaluated in the logistic regression, varies smoothly along the manifold representing a graph.

## 3 Methodology

**3.1 Background and Notations.** A *graph*  $G$  is described by a finite set of nodes  $V$  and a finite set of edges  $E \subset V \times V$ . In most applications, a graph is labeled, where labels are drawn from a label set  $\lambda$ . A labeling function  $\lambda : V \cup E \rightarrow \Sigma$  assigns labels to nodes and edges. In *node-labeled graphs*, labels are assigned to nodes only and in *fully-labeled graphs*, labels are assigned to nodes and edges. In this paper, we consider node labeled graphs only since nodes represent features for a sample.

Following convention, we denote a graph as a quadruple  $G = (V, E, \Sigma, \lambda)$  where  $V, E, \Sigma, \lambda$  are explained before. We represent a graph with  $n$  nodes using its adjacency matrix  $\xi = (\xi_{i,j})_{i,j=1}^n$  where  $\xi_{i,j} = 1$  if there exists an edge incident on nodes  $i$  and  $j$  in  $G$ , and zero otherwise. We use capital letters, such as  $G$ , for a single graph,  $V[G]$  for the node set of  $G$  and  $E[G]$  for the edge set of  $G$ , and upper case calligraphic letters, such as  $\mathcal{G} = G_1, G_2, \dots, G_n$ , for a set of  $n$  graphs.

Two graphs  $G, G'$  are *aligned* if there exists a 1-1 mapping  $\varphi : V[G] \rightarrow V[G']$  such that  $(u, v) \in E[G]$  if and only if  $(\varphi(u), \varphi(v)) \in E[G']$ . Clearly the aligned relation is (i) reflective, (ii) symmetric, and (iii) transitive and hence an equivalence relation. A group of graphs is *aligned* if the graphs in the group are pair-wise aligned.

**EXAMPLE 3.1.** In Figure 1 we show three graphs defined on 4 features  $\{x_1, x_2, x_3, x_4\}$  with a star topology. Clearly the three graphs are aligned since they have the same topology. We view each graph as an instance of a 4-dimensional variable  $X_i = (x_{i_1}, x_{i_2}, x_{i_3}, x_{i_4}) \in \mathbb{R}^4$ ,  $i \in [1, 3]$  with a binary relation defined on the 4 features.

**3.2 Logistic Regression.** Before we introduce regularized logistic regression, we briefly overview basic logistic regression [9]. Logistic regression fits a sigmoid function,  $P(Y = 1 | \vec{X} = \vec{x}; \vec{\beta}) = \frac{1}{1 + e^{-\vec{\beta}^T \vec{x}}} = \frac{e^{\vec{\beta}^T \vec{x}}}{e^{\vec{\beta}^T \vec{x}} + 1}$ , representing the probability the class label takes value 1 given the data sample has values  $\vec{x}$  and the parameters are  $\vec{\beta}$ , to the training data, here we use  $\vec{x}$  to de-

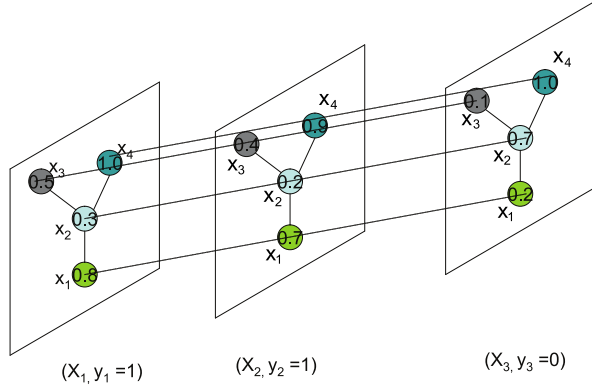


Figure 1: Three aligned graphs

note a data vector with an additional feature value of 1 concatenated to the beginning for convenience (to incorporate the intercept). Using the training data we find the parameters  $\vec{\beta}$  that best fit the data, and can then use the sigmoid function to map any future data vector to a value in  $[0, 1]$ . The fitting is achieved by maximizing the log-likelihood of the data (which we will denote as  $\ell(\vec{\beta})$ , as it is a function of the parameters  $\vec{\beta}$ ),  $\sum_{i=1}^N \{y_i \log(P(Y = 1 | \vec{X} = \vec{x}_i; \vec{\beta})) + (1 - y_i) \log(1 - P(Y = 1 | \vec{X} = \vec{x}_i; \vec{\beta}))\}$ , which can be expressed as:

$$(3.1) \quad \ell(\vec{\beta}) = \sum_{i=1}^N \{y_i \vec{\beta}^T \vec{x}_i - \log(1 + e^{\vec{\beta}^T \vec{x}_i})\}$$

, by setting the gradient,  $\frac{\partial \ell(\vec{\beta})}{\partial \vec{\beta}} = \sum_{i=1}^n \{\vec{x}_i (y_i - P(Y = 1 | \vec{X} = \vec{x}_i; \vec{\beta}))\}$ , equal to  $\vec{0}$ . We then find the zeros using an iterative process, the Newton-Raphson algorithm, which requires taking the second derivative of the log-likelihood. We express the derivative and second derivative of the log-likelihood in matrix form so that the update becomes:

$$(3.2) \quad \vec{\beta}^{new} = \vec{\beta}^{old} - \left( \frac{\partial^2 \ell(\vec{\beta}^{old})}{\partial \vec{\beta} \partial \vec{\beta}^T} \right)^{-1} \frac{\partial \ell(\vec{\beta}^{old})}{\partial \vec{\beta}}$$

which is:

$$(3.3) \quad \vec{\beta}^{new} = \vec{\beta}^{old} - (X^T W X)^{-1} X^T (\vec{y} - \vec{p})$$

where  $\vec{p}$  is a column vector with  $p_i = P(Y = 1 | \vec{X} = \vec{x}_i; \vec{\beta}^{old})$ , and  $W = \text{diag}(p) * \text{diag}(\vec{1} - p)$ , where  $\text{diag}(p)$  signifies a diagonal matrix with diagonal entries  $W_{ii} = p_i$  and all other entries set to 0, and  $\vec{1}$  is a column vector of ones, with dimension  $N$ . With the new beta calculated with equation 3.3, the probabilities are recalculated ( $p$  and  $W$  updated), and the process repeats until convergence, measured by the entries of

$W$  becoming close to 0 or by the change in  $\vec{\beta}$  becoming close to 0, using some small threshold value.

Thus for each data vector, we learn a set of parameters  $\vec{\beta}$ , and can then map each data vector to a probability of class label. We can threshold the output from the logistic regression at 0.5 to obtain the predicted class.

**3.3 Laplacian-Norm Regularized Logistic Regression.** Here we incorporate graph Laplacian as a regularization term in the logistic regression. Before we talk about regularized logistic regression, we define graph Laplacian and normalized graph Laplacian.

For an undirected graph  $G$  with the adjacency matrix  $\xi$ , the *Laplacian*  $L$  of  $G$  is:

$$(3.4) \quad L = D - \xi;$$

Where  $D$  is the density matrix of  $\xi$ , defined as  $D = (d_{i,j})_{i,j=1}^n$  where

$$d_{i,j} = \begin{cases} \sum_{k=1}^n \xi_{i,k} & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}$$

The *normalized Laplacian* is  $\mathcal{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$ .

Incorporating the normalized graph Laplacian norm as a regularization term in the logistic regression actually results in a simple modification to the original logistic regression solution. Furthermore, substituting the identity matrix for the normalized Laplacian  $\mathcal{L}$  results in logistic regression with the ridge penalty (the square of the  $L_2$  norm of  $\beta$ ), since  $\vec{\beta}^T I \vec{\beta} = \vec{\beta}^T \vec{\beta}$ .

The new objective function becomes:

$$(3.5) \quad g(\vec{\beta}) = \sum_{i=1}^N \{y_i \vec{\beta}^T \vec{x}_i - \log(1 + e^{\vec{\beta}^T \vec{x}_i})\} - \frac{1}{2} \lambda \vec{\beta}^T \mathcal{L} \vec{\beta}$$

The new gradient is given by:

$$(3.6) \quad \frac{\partial g(\vec{\beta})}{\partial \vec{\beta}} = X^T (\vec{y} - \vec{p}) - \lambda \mathcal{L} \vec{\beta}$$

The new hessian is given by:

$$(3.7) \quad \frac{\partial^2 g(\vec{\beta})}{\partial \vec{\beta} \partial \vec{\beta}^T} = -X^T W X - \lambda \mathcal{L}$$

And the new newton-raphson update is given by:

$$(3.8) \quad \vec{\beta}^{new} = \vec{\beta}^{old} - (X^T W X + \lambda \mathcal{L})^{-1} (X^T (\vec{y} - \vec{p}) - \lambda \mathcal{L} \vec{\beta}^{old})$$

### 3.4 Graph Regularized Kernel Logistic Regression.

Kernel logistic regression works by introducing a basis expansion so that  $f(\vec{x})$  in  $P(Y = 1|\vec{X} = \vec{x}; \vec{\beta}) = \frac{1}{1+e^{-f(\vec{x})}}$ , previously equal to  $\vec{\beta}^T \vec{x}$  is now equal to  $\alpha_0 + \sum_{i=1}^N \alpha_i K(\vec{x}, \vec{x}_i)$  where  $K(\cdot, \cdot)$  is a kernel function implicitly defining a Hilbert space and a feature mapping. In order to keep our Laplacian-regularization framework intact, we define a second method. Since the parameters are translated to the feature space, i.e. from  $\vec{\beta}$  varying over the  $p$  features (vertices) in the input feature space to  $\vec{\alpha}$  varying over the  $n$  features in the kernel space, the original constraints on the graph structure are lost for the parameters  $\alpha$ . Thus, in order to include the Laplacian regularization in the kernel space it is necessary to translate the graph structure from the input feature space to the kernel feature space. Essentially we want to define a new weighted graph structure between the  $n$ -samples such that the similarity function between two samples is regularized by the original graph structure (the original graph Laplacian in our framework). This is a similar idea to semi-supervised learning where we define an underlying similarity graph from the data. Here we want the graph created to impose similarity based on the closeness for matching vertices and the smoothness over the vertices.

In order to derive a similarity graph to regularize the alpha parameters, we estimate a sample similarity function that itself is regularized by the Laplacian of the original graph. We start with an edge of weight 1 between each training sample with the same label, of weight 0 (no edge) otherwise, a rough graph with connections between all samples of the same class. To incorporate the original graph structure, we train a logistic regression model to predict probabilities of link connections that is regularized by the original graph Laplacian. To do this we use a similarity measure (in the form of a Gaussian kernel function) between each pair of aligned vertices in the original graph, and fit a set of logistic regression parameters, using the Laplacian regularization. This translates the binary edge existence function to a weight that is regularized by the original graph structure, in effect smoothing the similarity function over the original graph structure.

To select the vertex-wise similarity parameter (width of the Gaussian) and the regularization parameter,  $\lambda$ , one option is to perform a cross-validation grid search with the training data, enforcing only that the thresholded output correctly predicts the link. In this way, the values can still vary smoothly. However, the number of samples in this case becomes  $(n^2 - n)/2$  (for  $n$  training samples), since each pair of training samples becomes a new training sample for the edge prediction function, so performing the multiple iterations with this

higher sample size set can be time consuming. As an alternative, we only perform the logistic regression once by setting  $\sigma$  equal to the standard deviation for each feature and using a high  $\lambda$  value to strongly enforce the regularization term (two times the number of new training samples), avoiding the lengthy grid search process.

In this way we can achieve our goal of creating a new graph structure in the kernel feature space that is still regularized by the original graph structure in the input feature space. Figure 2 shows a comparison of the rough, original similarity matrix to the derived similarity matrix for 90 training samples from a synthetic data set. The original structure can still be seen in the regressed similarity matrix (e.g. the cross shape) but this structure is softened (regularized).

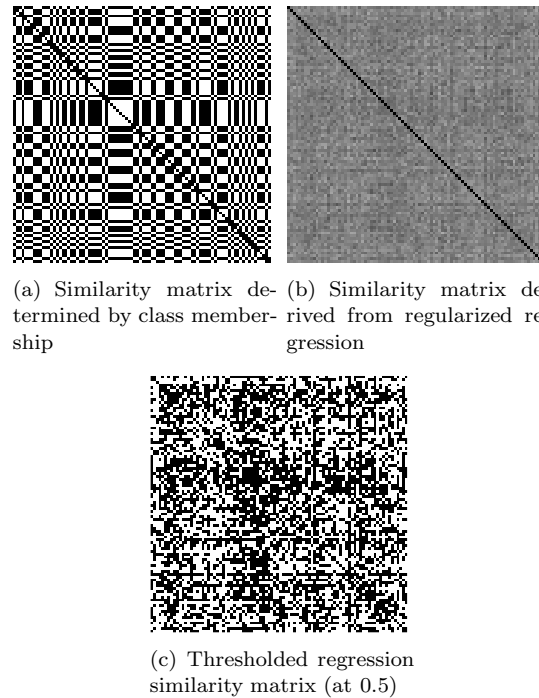


Figure 2: Regularized similarity graph for 90 samples of synthetic data

### 3.5 Regularized Local Logistic Regression.

Since the regularized kernel logistic regression method described in the previous section is time-consuming to perform in full, we explore another kernel logistic regression method for learning nonlinear class boundaries as an alternative, local logistic regression. The motivation is that often we may desire a model that does not find a global fit to the data, but rather a local fit, similar to the nearest neighbor method and local linear regression method. In this case local logistic regression can

be used. Local logistic regression results from a simple modification to the original logistic regression formulation; each sample is weighted by how close it is to the input test sample using some smoothed distance function such as the Gaussian kernel, when the model is fitted. This is described by the following weighting of the likelihood ( $L$ ) equation:  $L = \prod_{i=1}^N P(Y = y_i | \bar{X} = \bar{x}_i; \bar{\beta})^{\gamma_i}$ , with  $\gamma_i = e^{-\frac{\|\bar{x}_i - \bar{x}_t\|^2}{2\sigma^2}}$  for test input  $\bar{x}_t$ , which translates into multiplying each term in the log-likelihood by its sample weight. The Laplacian regularized version is the same as for regular logistic regression, except for weighting samples in the likelihood term of the objective function. The new update equations result by modifying equations 3.3 and 3.8 so that  $W_{ii} = p_i \gamma_i$  and  $\bar{y} - \bar{p}$  is scaled by the weights ( $diag(\bar{\gamma})(\bar{y} - \bar{p})$ ). Here increasing the kernel width  $\sigma$  results in moving closer to the global solution.

In the subsequent discussion for simplicity, we refer to the logistic regression method as “LR”, the Laplacian-regularized logistic regression method as “LREG”, the  $L_2$  norm regularized logistic regression method (with  $\mathcal{L}$  equal to the identity matrix) as “L2”, the kernel logistic regression as “KLR”. Similarly, we refer to the unregularized local logistic regression method as “LOC\_LR”, the  $L_2$  norm regularized local logistic regression method as “LOC\_L2” and the Laplacian-regularized local logistic regression method as “LOC\_LREG”.

## 4 Experimental Evaluation

### 4.1 Data

**4.1.1 Synthetic Data.** We generated synthetic test data for an undirected graph with 19 vertices described by the 4 arbitrary created pathways shown in figures 3(a) - 3(d), which specify the binary relationships between the given variables. For our tests we assume all we know is the existence of a relationship between the variables and form the corresponding undirected graph and 19x19 adjacency matrix. To generate data, the graph class is labeled 1 if at least 2 pathways “produce” (take value) 1, otherwise it is 0. A pathway “produces” 1 if all the node values along any path from a start node (at the left) to an end node of the path are greater than 0.5, otherwise it produces 0. Examples are given in figures 3(e) and 3(f). We indicate a path with all values greater than 0.5 in Figure 3(e) by small arrows. In Figure 3(f) we show a broken path since node (3) has value 0.3 which is less than 0.5. Thus the pathway in Figure 3(e) “produces” a label 1 and the pathway in Figure 3(f) “produces” a label 0. To generate data we randomly generate values for all the nodes in the

range  $[0, 1]$  and test the graph outcome. We generate 100 samples, and continue replacing samples with label 0 until half have label 1.

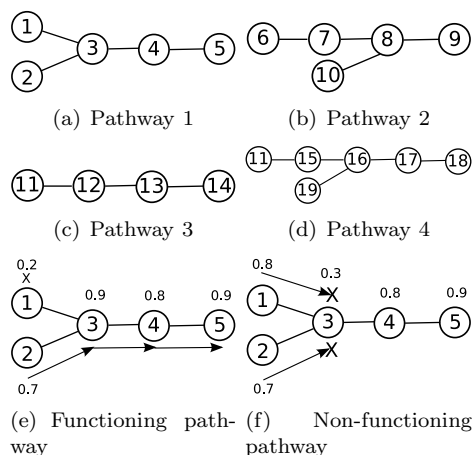


Figure 3: Artificial pathways used to generate test data

**4.1.2 Real World Data.** Next, we consider microarray gene expression data classification: given a set of samples of gene expression values and the associated class labels (e.g. disease or no disease), learn a classification model to predict the label of a test sample using its gene expression values as features. We can view the microarray classification task as an aligned graph classification task by considering the biological pathway structures associated with the genes. Here each pathway related to the outcome of interest is represented by an undirected graph with vertices as genes and edges representing the existence of relations between the genes such as protein-protein interactions resulting in activation or phosphorylation. To obtain the aligned graph structures for our experiments, we extract pathway graphs from a standard source of biological pathway information, the internet-accessible KEGG pathway database [12].

Since incorporating pathway structure in the learning process for pathways that are not related to the outcome of interest would not be expected to improve performance, and to avoid testing every pathway, we first perform external pathway selection. Determining which pathways are related to a particular outcome could be performed separately by any number of methods, e.g. searching through scientific literature for known related pathways, or using a computational statistical test tool; we use a readily-available method provided as a pre-built statistical package, the *global test* [8] method which tests if a group of variables are significantly related to an outcome of interest (the idea of incorporating grouped

variable selection into our Laplacian regularized framework is an area of future work). We use global test with the pathway gene expression data paired with the outcome labels to obtain a top candidate list of pathways from the KEGG database; the pathway structures of the selected pathways form the aligned graphs used for evaluating our algorithms.

We used the following three data sets for our experimental study:

- Diabetes Data:** The first microarray data set we include is a microarray data set related to diabetes, obtained from [19] (available online at <http://www.broad.mit.edu/mpg/oxphos/>). The data set contains the gene expression values of 22,280 genes for 44 different subjects, 17 with type 2 diabetes (DM2), 17 with normal glucose tolerance (NGT) and 10 with impaired glucose tolerance (IGT). As in [15], we use only the samples of subjects with type 2 diabetes and those with normal glucose tolerance, resulting in a total of 34 samples. We use the global test method to estimate related pathways; we select all pathways found to be related to the diabetes outcome by the global test method with a significance p-value of less than 0.1 and keep those that have an associated graph structure, resulting in the 14 pathways shown in table 1. In evaluating the aligned graph classification methods, their performance on the Insulin Signaling Pathway is of particular interest, since aside from the global test results, we would expect this pathway to be related to the diabetes disease, and as such can be more confident that the pathway is related to the outcome in this case.
- Breast Cancer Data:** The next data set we use is a microarray gene expression data set for human breast cancer samples [5]; in this case there are 118 breast tumor samples and we select the “alive at endpoint” factor as the class label, resulting in 41 positive samples and 77 negative samples. We once again use global test to select related pathways, however since only 3 pathways were found with p-value less than 0.13, we select the pathways with graphs from the top 20, resulting in 14 pathways.
- Yeast Data:** The final data set is a microarray data set for yeast [18, 22]; here the gene expression values are measured across 18 independent samples of (*Saccharomyces cerevisiae*) yeast cultures, and the goal is to classify whether or not a sample was grown with irradiation (6 samples are labeled as Irradiated, I, and 12 as Not Irradiated, NI). Since the data set was much smaller (around 6,000

Table 1: Estimated related pathways found with global test (p-value < 0.1) for the Diabetes data set

| Index | Pathway   | Genes | P-value |
|-------|---|-------|---------|
| 1     | Insulin signaling pathway                           | 250   | 0.0673  |
| 2     | mTOR signaling pathway                              | 90    | 0.0229  |
| 3     | Biosynthesis of steroids                            | 42    | 0.0577  |
| 4     | Oxidative phosphorylation                           | 153   | 0.0384  |
| 5     | Alanine and aspartate metabolism                    | 44    | 0.0264  |
| 6     | Phenylalanine, tyrosine and tryptophan biosynthesis | 14    | 0.0497  |
| 7     | Glycosphingolipid biosynthesis - lactoseries        | 15    | 0.0931  |
| 8     | Glycosphingolipid biosynthesis - globoseries        | 23    | 0.0839  |
| 9     | Lipoic acid metabolism                              | 2     | 0.0379  |
| 10    | Terpenoid biosynthesis                              | 12    | 0.0337  |
| 11    | Nitrogen metabolism                                 | 39    | 0.0969  |
| 12    | Alkaloid biosynthesis I                             | 7     | 0.0500  |
| 13    | PPAR signaling pathway                              | 118   | 0.0755  |
| 14    | SNARE interactions in vesicular transport           | 65    | 0.0263  |

genes), we obtained results for all pathways we were able to make graphs for, a total of 94 pathways. In addition we applied pre-processing to handle missing values by replacing feature values with the average value for that feature if at least 80% were not missing, otherwise we removed the feature.

**4.2 Evaluation Criteria.** We use several approaches to evaluate the performance of the graph classification methods. For the synthetic data we perform 100 trial iterations using a hold-out approach, generating a new sample set from the given graph and using a fixed fraction of the 100 samples for the training data and the remainder for testing, taking the average and standard deviation of the performance criteria across the trials. For the diabetes data set, we average the performance across 30 iterations of ten-fold cross-validation [14], and for the breast cancer data set, 30 iterations of five-fold cross-validation, since there are more samples. For the yeast data, we estimated performance using two approaches, due to the small data set size and imbalance of labels. For the first approach, we generate 50 training and test sets by generating all 50 unique partitions of the positive class such that at least 2 samples from the positive class (I) are in each set, and randomly partition the data from negative class (NI) so that the training set always has 10 samples. The other approach we used was bootstrap sampling, the “.632+” bootstrap estimator (see [9] for more details), using 100 bootstrap data sets.

For all the experiments, we estimate the accuracy and performance for our new Laplacian regularized logistic regression method (LREG) and compare it to five other methods, which only use the feature values of the

graphs: previous logistic regression methods, including unregularized logistic regression (LR),  $L_2$  norm regularized logistic regression (L2), and kernel logistic regression (KLR), and support vector machine methods which include a linear kernel support vector machine (SVM\_LIN) and a Gaussian radial-basis function (RBF) kernel support vector machine (SVM\_RBF) (see, e.g., [9] for more information about these common classifiers). In addition, for our synthetic experiments and for the key diabetes pathway, we include results for the Laplacian regularized local logistic regression (LOC\_LREG) along with the unregularized local logistic regression (LOC\_LR) and an  $L_2$  norm regularized local logistic regression (LOC\_L2). We implemented the logistic regression methods in Matlab and used a Matlab toolbox implementation for the support-vector methods. To select parameters for all aligned graph classification models where needed (specifically  $\lambda$  for the various regularized logistic regression methods,  $\sigma$  for the kernel logistic regression methods and RBF SVM method, and  $C$  for the SVM methods), we perform a cross-validation grid search with the training data using a course-to-fine grid approach as in LibSVM [4].

In addition to accuracy, we include three other common performance criteria as described in the following list:

1. Accuracy (ACC):  $\frac{TP+TN}{TP+TN+FP+FN}$
2. Matthews Correlation Coefficient (MCC):  $\frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$
3. Sensitivity (SEN):  $\frac{TP}{TP+FN}$
4. Specificity (SPE):  $\frac{TN}{TN+FP}$

In this description, FP denotes “false positive,” a negative instance that was classified as positive, TP denotes “true positive,” a positive instance that was classified as positive, TN denotes “true negative,” a negative instance that was classified as negative, and FN denotes “false negative,” a positive instance that was classified as negative.

Additionally, since the average accuracy of one method may be better than another, but the standard deviation could be too high to distinguish if the method performed better consistently across test iterations, we perform a paired  $t$ -test at the five percent level between the 100 test accuracies for each method, to determine if a method’s higher accuracy can be considered statistically significant. For the real-world data sets with the cross-validation, the  $t$ -test is across the number of iterations, 30.

**4.3 Synthetic Data Classification Results.** The first set of results shows the performance criteria averaged over 100 iterations of a 60% hold-out, so that for each iteration, 100 samples were generated from which 40 samples were randomly selected for training, 60 for testing (the samples were selected so that at least one-third of each class was present). These results are shown in table 2, with the best method for each criteria shown in bold (results for the local logistic regression methods are not included in this table to save space, but are shown in figure 4).

Table 2: Results on synthetic test data for aligned graph classification methods

|     | LREG         | L2    | SVM_LIN | LR    | SVM_RBF | KLR          |
|-----|--------------|-------|---------|-------|---------|--------------|
| ACC | <b>0.767</b> | 0.732 | 0.722   | 0.666 | 0.726   | 0.716        |
| std | 0.060        | 0.062 | 0.060   | 0.066 | 0.067   | 0.061        |
| MCC | <b>0.541</b> | 0.469 | 0.448   | 0.338 | 0.460   | 0.470        |
| std | 0.119        | 0.125 | 0.121   | 0.133 | 0.135   | 0.110        |
| SEN | 0.789        | 0.747 | 0.739   | 0.677 | 0.716   | <b>0.890</b> |
| std | 0.094        | 0.100 | 0.095   | 0.107 | 0.117   | 0.086        |
| SPE | <b>0.746</b> | 0.716 | 0.704   | 0.656 | 0.737   | 0.542        |
| std | 0.092        | 0.096 | 0.099   | 0.111 | 0.104   | 0.147        |

We performed a paired  $t$ -test at the five percent level on the accuracies obtained from the 100 runs, and found that the LREG method is performs significantly better in terms of accuracy (the null hypothesis of same mean of distribution is rejected) than all of the other methods. Similarly, all the regularized methods are found to perform significantly better than the unregularized logistic regression (LR). These results are shown in table 3, in which a significance was found using the paired  $t$ -test between the method in each row and column, a 1 indicating a significant difference with a positive 1 indicating the method in the row had a higher average accuracy than the method in the column and a negative 1 lower, and a 0 representing that the null hypothesis could not be rejected.

Table 3: Paired  $t$ -test results on synthetic test data across 100 iterations, between each pair of methods. A positive 1 indicates the method in the row performed significantly better on average than the method in the column, a negative 1, worse, and a 0 that the difference in performance of the two methods was not statistically significant according to the  $t$ -test at the 5% level.

|         | LREG | L2 | SVM_LIN | LR | SVM_RBF | KLR |
|---------|------|----|---------|----|---------|-----|
| LREG    | 0    | +1 | +1      | +1 | +1      | +1  |
| L2      | -1   | 0  | +1      | +1 | 0       | +1  |
| SVM_LIN | -1   | -1 | 0       | +1 | 0       | 0   |
| LR      | -1   | -1 | -1      | 0  | -1      | -1  |
| SVM_RBF | -1   | 0  | 0       | +1 | 0       | 0   |
| KLR     | -1   | -1 | 0       | +1 | 0       | 0   |

The next set of results, figure 4, shows the relationship between accuracy and the size of the training set used, obtained by running the experiments with each hold-out percentage (100 iterations as before). As can be seen the Laplacian regularized method (LREG) outperforms the others consistently, but the performance gain is greatest with smaller training sample size. While the other methods converge to a lower value at the smallest training sample size tested (10 training samples), the Laplacian regularized method maintains a 5 percent higher accuracy. We also included results for the local logistic regression methods, for the first 4 training set sizes. Here we see that the  $L_2$  regularized local logistic regression (LOC\_L2) is a significant improvement over the unregularized local logistic regression (LOC\_LR), and that the Laplacian regularized local logistic regression (LOC\_LREG) significantly outperforms both. For small samples, regular Laplacian regularized logistic regression (LREG) outperforms LOC\_LREG, which in turn outperforms the other methods, but with increasing sample size the LOC\_LREG method achieves comparable performance. While in general, the results obtained for the local logistic regression method using a nonlinear similarity function were worse than the methods with linear models, the results were not far off. We included these results to show the plausibility of using the Laplacian regularized local logistic regression to incorporate aligned graph structure for those cases where a nonlinear boundary is desired or known to exist.

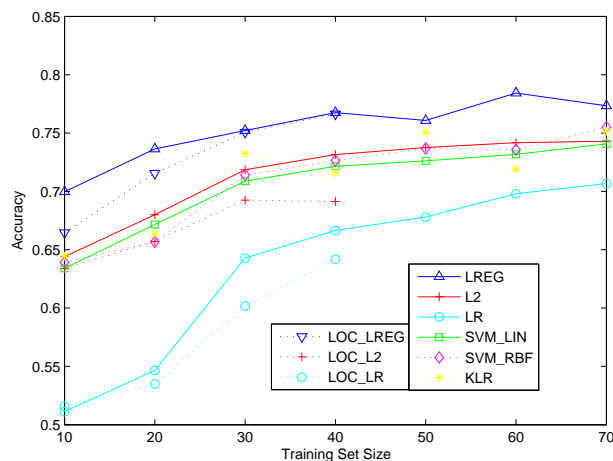


Figure 4: Average Accuracy vs. Training Set Size for Synthetic Data

Figure 5 shows the variation of the accuracy of the Laplacian regularized logistic regression method (LREG) with respect to the regularization weight,  $\lambda$ , obtained by averaging over 100 iterations as before with

a training set size of 40. We also include the results for the  $L_2$  regularized logistic regression (L2) for comparison as well as the constant result for unregularized logistic regression as a baseline. From the results we see that the LREG method's performance varies in a similar way to the L2 method's performance with respect to the regularization parameter for this experiment, and additionally that in this case it is safer to overestimate the value of the regularization parameter than underestimate, since accuracy increases steadily until about  $\lambda = 2^4$  at which point it remains close to the highest value reached.

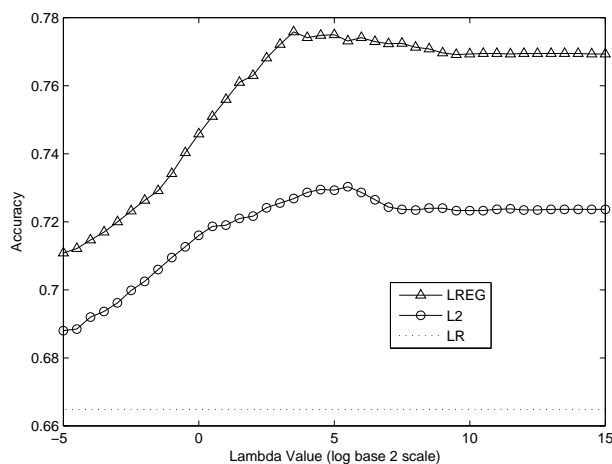


Figure 5: Average Accuracy vs. Regularization Parameter for Synthetic Data

#### 4.4 Real-World Data Classification Results.

For the real-world data classification results, we show the results for each pathway, i.e. by treating the set of data for each pathway as an aligned graph classification problem. Thus, for example, for data with 14 pathways we in effect have 14 data sets. For the diabetes data, we performed 30 iterations of ten-fold cross-validation to estimate the performance of each method for each pathway. The results of each method for each pathway are shown in figure 6, in which each point on the x-axis represents a pathway, and each point on the y-axis the average accuracy.

For the 14 pathways, the Laplacian regularized method (LREG) performed significantly better than the rest for 2 of the pathways, as did the linear SVM (SVM\_LIN); the other methods did not perform significantly better than the rest of the methods for any of the pathways, except for the kernel logistic regression method (KLR) for 1 pathway. Furthermore, the only pathways for which the LREG method performed the worst were those for which all the methods had 50

percent accuracy or worse.

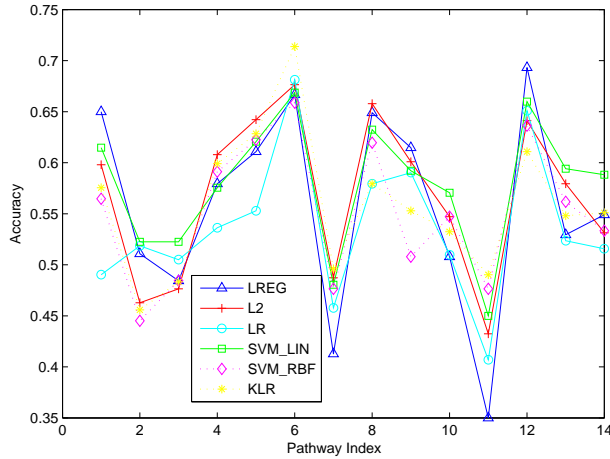


Figure 6: Average Accuracy vs. Pathway Index for Diabetes Data

We suspect one reason the Laplacian regularized method did not perform significantly better on all pathways is that many pathways are likely unrelated to the disease outcome, or some of the genes in a given pathway are related, but as a part of a different pathway instead of the given pathway, in which case the Laplacian regularized method would not be expected to improve the performance. Thus we take a closer look at the Insulin Signaling Pathway which we reason is one pathway that is more likely to be related to the diabetes disease outcome. For this pathway we also include results from the local logistic regression methods. The results for the Insulin Signaling Pathway are shown in table 4, the best score for each criteria is shown in bold. For this pathway, the Laplacian regularized logistic regression (LREG) performed the best for all criteria. We also see that for this pathway the Laplacian regularized local logistic regression outperformed the other kernel methods, and for each method adding regularization improved the performance. By performing paired  $t$ -tests as with the synthetic data, we see that the improvement from the LREG method was statistically significant (table 5).

In general in our experiments, the linear logistic regression methods, LR, LREG, and L2 had comparable training time to the support-vector machine methods, and were in many cases faster. However the kernel-based logistic regression methods, KLR and LOC\_LR, LOC\_L2, and LOC\_LREG usually took longer to train, KLR due to calculating the basis expansions and a slower convergence of Newton’s method, and the local logistic regression took longer since the regression process had to be repeated for each test point, since the weights  $\gamma_i$  assigned in the optimization were based on

Table 4: Results on diabetes data for aligned graph classification methods for the Insulin Signaling Pathway

|            | LREG         | L2    | SVM_LIN | LR     | SVM_RBF | KLR   |
|------------|--------------|-------|---------|--------|---------|-------|
| <b>ACC</b> | <b>0.650</b> | 0.598 | 0.615   | 0.490  | 0.565   | 0.575 |
| std        | 0.056        | 0.052 | 0.035   | 0.068  | 0.062   | 0.053 |
| <b>MCC</b> | <b>0.301</b> | 0.197 | 0.230   | -0.019 | 0.130   | 0.151 |
| std        | 0.112        | 0.104 | 0.070   | 0.140  | 0.124   | 0.106 |
| <b>SEN</b> | <b>0.633</b> | 0.588 | 0.584   | 0.463  | 0.565   | 0.584 |
| std        | 0.081        | 0.056 | 0.046   | 0.087  | 0.068   | 0.064 |
| <b>SPE</b> | <b>0.667</b> | 0.608 | 0.645   | 0.518  | 0.565   | 0.567 |
| std        | 0.052        | 0.071 | 0.036   | 0.113  | 0.080   | 0.065 |

|             | LOC_LREG | LOC_L2 | LOC_LR |
|-------------|----------|--------|--------|
| <b>ACC</b>  | 0.590    | 0.540  | 0.509  |
| std         | 0.046    | 0.056  | 0.075  |
| <b>MCC</b>  | 0.180    | 0.081  | 0.017  |
| std         | 0.093    | 0.113  | 0.156  |
| <b>SENS</b> | 0.588    | 0.551  | 0.483  |
| std         | 0.073    | 0.084  | 0.131  |
| <b>SPEC</b> | 0.591    | 0.529  | 0.536  |
| std         | 0.060    | 0.076  | 0.121  |

Table 5: Paired  $t$ -test results on diabetes test data across 30 iterations, between each pair of methods. A positive 1 indicates the method in the row performed significantly better on average than the method in the column, a negative 1, worse, and a 0 that the difference in performance of the two methods was not statistically significant according to the  $t$ -test at the 5% level.

|                | LREG | L2 | SVM_LIN | LR | SVM_RBF | KLR |
|----------------|------|----|---------|----|---------|-----|
| <b>LREG</b>    | 0    | +1 | +1      | +1 | +1      | +1  |
| <b>L2</b>      | -1   | 0  | -1      | +1 | +1      | +1  |
| <b>SVM_LIN</b> | -1   | +1 | 0       | +1 | +1      | +1  |
| <b>LR</b>      | -1   | -1 | -1      | 0  | -1      | -1  |
| <b>SVM_RBF</b> | -1   | -1 | -1      | +1 | 0       | 0   |
| <b>KLR</b>     | -1   | -1 | -1      | +1 | 0       | 0   |

the kernel similarity of the tests point to the training points. Thus due to time constraints, we do not include results for these kernel-based methods for all data sets.

Next, we show the results for the breast cancer data in the same graph form as the diabetes data in figure 7. In general the less regularized logistic regression such as  $L_2$  regularized logistic regression performs as well as unregularized logistic regression; the Laplacian regularized logistic regression did not outperform all of the other classifiers for any pathway. We suspect that, since the pathways themselves are not known for certain, the relation to the known pathways to the disease may not be strong and hence regularization does not help too much. To test the hypothesis, we checked the global test matches and identified that none of the pathways have  $p$ -value less than 0.05 and only the first three had  $p$ -value less than 0.10.

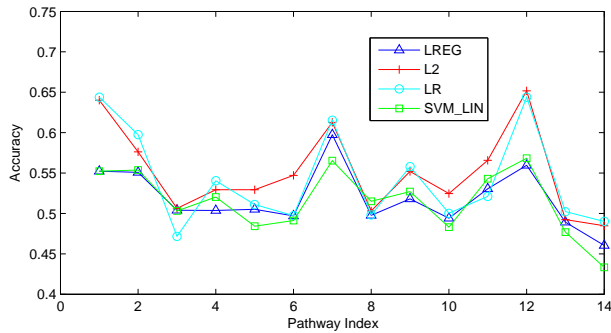


Figure 7: Average Accuracy vs. Pathway Index for Breast Cancer Data

Finally we show the results for the 94 pathways of the yeast data for the 50 partition estimate (training set size 10) in figure 8 and the “.632+” bootstrap estimate (training sets of size 18) in figure 9, with the pathway number on the x-axis and the estimated accuracy on the y-axis. The results are similar to the diabetes results, the best performing method varies for each pathway. The Laplacian regularized logistic regression only obtains significantly improved performance for a few of the pathways. However, we might expect this since it is likely only a few of the pathways are directly related to the outcome of interest. In this case, however, we have no ground truth available for which pathways are truly related, and the methods performed similarly on the top pathways selected by global test, though even this test we would expect to be less accurate with such few samples.

## 5 Conclusion

Data with intrinsic graph topology are becoming abundant in many applications including bioinformatics and sensor network analysis. We call such data aligned graphs and in this paper, we investigate a new problem of classification on aligned graphs. We have extended the  $L_2$  regularized logistic regression to aligned graph classification. Our experimental study demonstrates the utility of the methods in synthetic and real data sets. In the future, we will investigate dynamic graph structure, where we allow small amount of graph topology changes, in the Laplacian based logistic regression framework.

## 6 Acknowledgement

This work has been partially supported by the Kansas IDeA Network for Biomedical Research Excellence (NIH/NCCR award #P20 RR016475) and by the Office of Naval Research (award number N00014-07-1-1042).

## References

- [1] R. Ando and T. Zhang. Learning on Graph with Laplacian Regularization. In *Advances in Neural Information Processing Systems: Proceedings of the 2006 Conference*. MIT Press, 2007.
- [2] G. Cawley and N. Talbot. Gene selection in cancer classification using sparse logistic regression with Bayesian regularization. *Bioinformatics*, 22(19):2348, 2006.
- [3] S. L. Cessie and J. C. V. Houwelingen. Ridge estimators in logistic regression. *Applied Statistics*, 41(1):191–201, 1992.
- [4] C. Chang and C. Lin. Libsvm: a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [5] K. Chin, S. DeVries, J. Fridlyand, P. Spellman, R. Roydasgupta, W. Kuo, A. Lapuk, R. Neve, Z. Qian, T. Ryder, F. Chen, H. Feiler, T. Tokuyasu, C. Kingsley, S. Dairkee, Z. Meng, K. Chew, D. Pinkel, A. Jain, B. Ljung, L. Esserman, D. Albertson, F. Waldman, and J. Gray. Genomic and transcriptional aberrations linked to breast cancer pathophysiology. *Cancer Cell*, 10(6):529–541, 2006.
- [6] R. Collobert and S. Bengio. Svmtorch: Support vector machines for large-scale regression problems. *Journal of Machine Learning Research*, 21, 2001.
- [7] G. Fort and S. Lambert-Lacroix. Classification using partial least squares with penalized logistic regression. *Bioinformatics*, 21(7):1104–1111, 2005.
- [8] J. Goeman, S. van de Geer, F. de Kort, and H. van Houwelingen. A global test for groups of genes: testing association with a clinical outcome, 2004.
- [9] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, 2001.
- [10] D. Haussler. Convolution kernels on discrete structures. *Technical Report UCSC-CRL099-10*, Computer Science Department, UC Santa Cruz, 1999.
- [11] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.
- [12] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Research*, 28(1):27, 2000.
- [13] S. Kim and E. P. Xing. Structured feature selection in high-dimensional space via block regularized regression. In *Proceedings of the 24th International Conference on Conference on Uncertainty in Artificial Intelligence*, 2008.
- [14] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the fourteenth international joint conference on artificial intelligence*, pages 1137–1143. Morgan Kaufmann, 1995.
- [15] L. Liang, V. Mandal, Y. Lu, and D. Kumar. Mcm-test: a fuzzy-set-theory-based approach to differential analysis of gene pathways. *BMC Bioinformatics*, 9(Suppl 6):S16, 2008.
- [16] J. Liao and K. Chin. Logistic regression for disease classification using microarray data: model selection in a large p and small n case. *Bioinformatics*, 23(15):1945, 2007.
- [17] X. Liao, H. Li, and L. Carin. Quadratically gated mixture of experts for incomplete data classification. In *Proceedings of the 24th International Conference on Machine Learning*, 2007.
- [18] G. Mercier, N. Berthault, J. Mary, J. Peyre, A. Antoniadis, J.-P. Comet, A. Cornuejols, C. Froidevaux, , and M. Dutreix. Biological detection of low radiation doses by combining results of two microarray analysis methods. *Nucleic Acids Research*, 32(1):e12, 2004.
- [19] V. Mootha, C. Lindgren, K. Eriksson, A. Subramanian, S. Sihag, J. Lehar, P. Puigserver, E. Carlsson, M. Ridderstraale, E. Laurila, et al. PGC-1  $\alpha$ -responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nature Genetics*, 34(3):267–273, 2003.
- [20] H. Pang, A. Lin, M. Holford, B. Enerson, B. Lu, M. Lawton, E. Floyd, and H. Zhao. Pathway analysis using random forests classification and regression. *Bioinformatics*, 22(16):2028, 2006.
- [21] M. Y. Park and T. Hastie. Penalized logistic regression for detecting gene interactions. *Biostatistics*, 9(1):30–50, 2008.
- [22] F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot, and J.-P. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8:R35, 2007.
- [23] B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352, 1999.

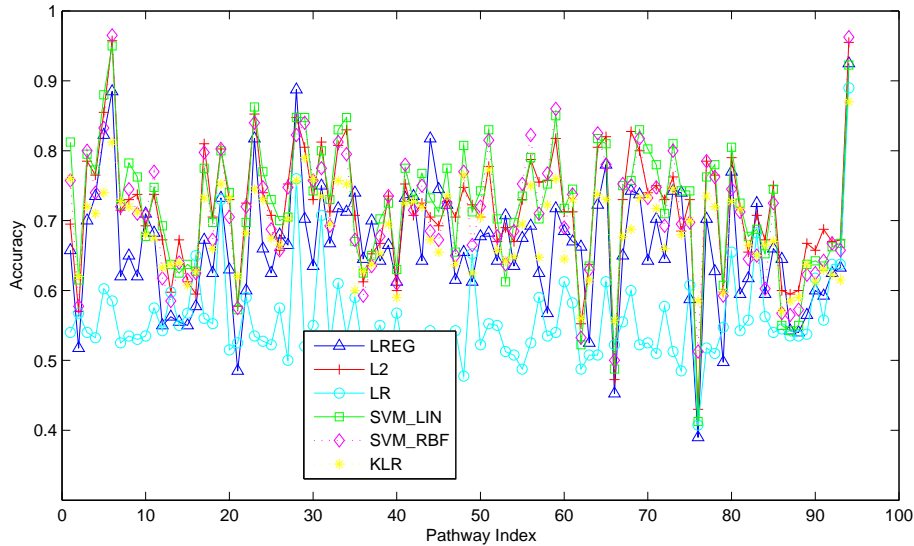


Figure 8: Average Accuracy vs. Pathway Index for Yeast Data: Partitioning Estimate

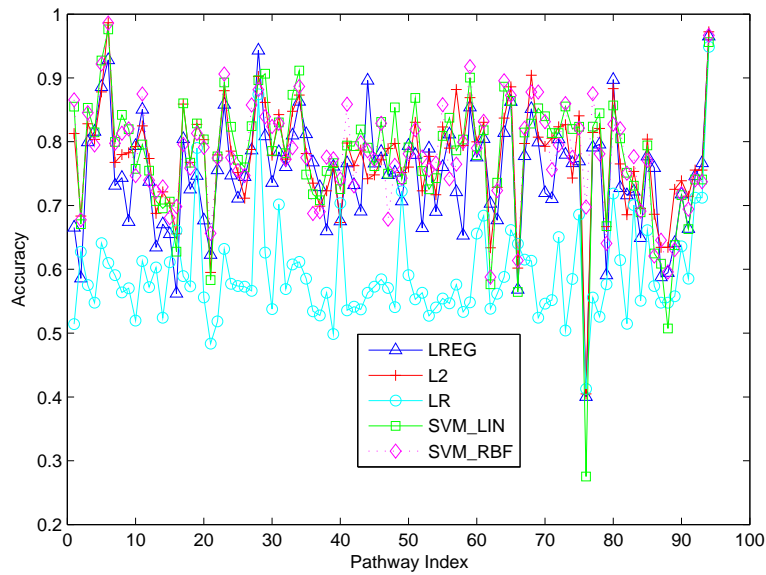


Figure 9: Average Accuracy vs. Pathway Index for Yeast Data: Bootstrap Estimate

- [24] S. Shevade and S. Keerthi. A simple and efficient algorithm for gene selection using sparse logistic regression, 2003.
- [25] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 888–905, 2000.
- [26] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.
- [27] S. Wu, H. Zou, and M. Yuan. Structural variable selection in support vector machines. *The Electronic Journal of Statistics*, 2:103–117, 2008.
- [28] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1)(1-2):49–67, 2006.
- [29] P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. Technical report, Department of Statistics, University of California, Berkeley, 2006.
- [30] N. Zhou and J. Zhu. Group variable selection via a hierarchical lasso and its oracle property. Technical report, Department of Statistics, University of Michigan, 2007.
- [31] J. Zhu and T. Hastie. Classification of gene microarrays by penalized logistic regression. *Biostatistics*, 5(3):427–443, 2004.
- [32] X. Zhu, J. Kandola, J. Lafferty, and Z. Ghahramani. Graph kernels by spectral transforms. In O. Chapelle, B. Scholkopf, and A. Zien, editors, *Semi-Supervised Learning*. The MIT Press, Cambridge, MA, 2006.