ABSTRACT

Boosting is a very successful classification algorithm that produces a linear combination of “weak” classifiers (a.k.a. base learners) to obtain high quality classification models. In this paper we propose a new boosting algorithm where base learners have structure relationships in the functional space. Though such relationships are generic, our work is particularly motivated by the emerging topic of pattern based classification for semi-structured data including graphs. Towards an efficient incorporation of the structure information, we have designed a general model where we use an undirected graph to capture the relationship of subgraph-based base learners. In our method, we combine both $L_1$ norm and Laplacian based $L_2$ norm penalty with Logit loss function of Logit Boost. In this approach, we enforce model sparsity and smoothness in the functional space spanned by the basis functions. We have derived efficient optimization algorithms based on coordinate decent for the new boosting formulation and theoretically prove that it exhibits a natural grouping effect for nearby spatial or overlapping features. Using comprehensive experimental study, we have demonstrated the effectiveness of the proposed learning methods.

Categories and Subject Descriptors
H.2.8 [Database Management]: Database Applications-Data Mining

General Terms
Algorithms, Experimentation

Keywords
$L_1$ Regularization, Graph Classification, Semi-Structured Data, Boosting, Feature Selection

1. INTRODUCTION

Boosting is a very successful classification algorithm that produces a linear combination of “weak” classifiers (a.k.a. base learners) to obtain high quality classification models [7, 9, 34, 35]. Recently, the boosting algorithm has been successfully extended to tasks such as multi-class classification [23], multi-label classification [39], cost sensitive learning [26], semi-supervised learning [43], manifold learning [24], classification with missing-value [12], and transfer learning [3] among others.

In this paper we propose a new boosting algorithm where base learners have structure relationships in the functional space. Our work is particularly motivated by the emerging topic of pattern based classification for semi-structure data including graphs [16, 31, 36, 38, 40]. For example, Kudo et al. [20] recently applied boosting to graph classification using subgraphs as base learners and showed the connection of graph boosting to support vector machine with the R-convolution kernel. Nowozin et al. [28, 32] combined subgraph mining and graph boosting for classifying graphs representing images.

Though graph boosting has demonstrated promising results, the limitations of the current algorithms are that they totally ignore the structure relationships among subgraph base learners and hence may not provide the optimal results for graph classification. We illustrate the point with the following example:

Consider the three labeled graphs $G_1$, $G_2$, $G_3$ and three subgraph features $F_1$, $F_2$, $F_3$ shown in Figure 1. Suppose that the class labels for graphs $G_1$, $G_2$, $G_3$ are $Y = [1, 1, -1]^T$. We may construct three base learners $h_1(G)$, $h_2(G)$ and $h_3(G)$ in the format $h_i(G) = 1$ if $F_i \subseteq G$ and $h_i(G) = -1$ otherwise ($i \in \{1, 2, 3\}$). These decision rules are derived based on a majority voting of subgraph coverage on positively and negatively labeled graphs.

Figure 1: Three subgraph features in three graphs. Dashed edge means that the two nodes are connected by a path with varying length >1.
Considering a boosting algorithm that iteratively selects base classifiers to build ensemble models, since \( h_1 \) is perfectly correlated with class labels as evaluated on the three training samples, \( h_2 \) will be selected first. \( h_2 \) and \( h_3 \) produce the same prediction for all the graphs in the training data set and hence may be perceived to have the same discriminative power. This is not true in this example. Subgraph \( F_1 \) and \( F_2 \) occur in every positive graph sample and are clustered with a consistent relative spatial position. \( F_1 \) occurs in every graph, but in contrast to \( F_1 \) and \( F_2 \), it has quite different spatial distribution as compared to \( F_1 \) and \( F_2 \) and hence we consider \( F_2 \) as a spurious pattern. Once \( F_1 \) is selected, we argue that we should select \( F_2 \) rather than \( F_1 \) to build more stable and interpretable classification models. However, current boosting methods are not designed to perform such model selection since the structure relationships of base learners are not considered in any case.

The spatial relationship is special cases of possible relationships of base learners. Another example is the partial overlapping relationship. We call the possible information regarding to the relationships of base learners as structure relationships. Here we hypothesize that the structure relationship of subgraph features carries important information regarding the importance of the base learners in boosting. Towards an efficient incorporation of such information, we design a general model where we use an undirected graph to capture the relationship of subgraph-based base learners. We combined \( L_1 \) norm and Laplacian based \( L_2 \) norm penalty with Logit less function of Logit Boost [9]. In this approach, we enforce model sparsity and smoothness in the functional space spanned by the basis functions. We derive efficient optimization algorithms based on coordinate descent for the new boosting formulation and theoretically prove that it exhibits a natural grouping effect for nearby spatial or overlapping features. Using comprehensive experimental study and comparing with the state-of-the-art, we have demonstrated the effectiveness of the proposed learning method.

We believe the new formalization is applicable to a variety of boosting applications where (i) base learners have a known structure relationship and (ii) the optimal ensemble of base learner functions is sparse in the functional space. The proposed method can be naturally extended to other semi-structured data such as sequences and trees where patterns such as frequent subsequences and frequent subtrees are widely used for classification [21].

1.1 Related Work

Subgraph based supervised learning on graphs has recently attracted extensive research interest [16, 31, 36, 38, 40]. For example, Yan et al. [40] proposed Leap algorithm with two concepts: structural leap search and frequency descending to reduce search space and mine informative patterns faster than previous methods. However, LEAP only considers individual pattern rather than a set of patterns [16]. Moreover, the discriminate power of a pattern is evaluated entirely on the occurrence information of the pattern and misses interaction among patterns. gPLS [31] applies partial least square regression to graph mining and performs feature selection and classifier construction simultaneously, but the model interpretability is low due to the use of latent variables [16]. In addition, the structure relationships among features is neglected. COM [16] is a newly proposed method that mines co-occurrence rules. COM is prone to giving high number of false positives and fails to consider the structure information among features as well.

Recently, a significant amount of progress has been made on developing supervised learning algorithms for feature selection from data with structured features [4, 15, 18, 22, 33, 37, 41]. In these models, features may be naturally partitioned into groups [4, 15, 41] or ordered in some meaningful way, such as a chain [18, 37], a tree [42] or a graph [22, 33]. These approaches demonstrate the importance of incorporating prior structure information among features to build highly accurate and interpretable models. However, all these algorithms handles vector data and hence are not applicable to graphs.

Though subgraph based feature selection on graph data has been studied for a long time, none of the existing methods considers the structure relationships among subgraph features and hence may not provide the optimal results for graph classification. The objective of this paper is to incorporate the structural information on features into learning and build a more accurate and interpretable graph boosting model.

The rest of the paper is organized as following. In Section 2, we provide some preliminaries on Boosting. In Section 3 we show our details of our methodology. In Section 4 we apply the algorithm and theory developed in Section 3 to graph data and graph classification. In Section 5 we present the experimental study of our algorithm, followed by a conclusion and a discussion of the future work.

2. PRELIMINARIES

We use the following notations throughout the rest of the paper. We use lowercase letters to represent scalar values, lower-case letters with an arrow to represent vectors (e.g. \( \mathbf{\beta} \)), uppercase letters to represent matrices, \( \{\lambda, \lambda_1, \lambda_2, \ldots\} \) to represent Lagrange multiplier, and uppercase calligraphic letters to represent sets. Unless state otherwise, all vectors in this paper are column vectors.

Given training instances \( T = \{x_i, y_i\}_{i=1}^n \) where \( y_i \in \{-1,+1\} \), \( x_i \in X \), we construct a set of base learners \( \mathcal{H} = \{h_j : X \mapsto \{-1,+1\}, j = 1 \cdots p\} \). In this paper, we do not assume any type of \( X \); it may be a vector space, or simply a set. The objective of boosting is to train a composite binary classifier with weight vector \( \mathbf{\beta} \) taking the form of

\[
\hat{h}_\mathbf{\beta}(x_i) = \text{sign}(\sum_{j=1}^p \beta_j h_j(x_i))
\]

such that the following empirical loss function \( \ell(X, y, \mathbf{\beta}) \) is minimized.

\[
\mathcal{L}(X, y, \mathbf{\beta}) = \sum_{i=1}^n \ell(y_i, h_\mathbf{\beta}(x_i))
\]

where \( \ell \) is a loss function.

AdaBoost [8] takes the exponential loss function:

\[
\ell(y_i, h_\mathbf{\beta}(x_i)) = \exp(-y_i \sum_{j=1}^p \beta_j h_j(x_i))
\]

and LogitBoost [9] takes the logit loss function:

\[
\ell(y_i, h_\mathbf{\beta}(x_i)) = \log(1 + \exp(-y_i \sum_{j=1}^p \beta_j h_j(x_i)))
\]
framework. However, they neglect the structure information among base learners. We consider a simple yet effective modification to Logit Boost [9] that incorporates a composite penalty with $L_1$ and $L_2$ regularization encoding the structural information among base learners on the weight vector, which is detailed in the following section.

3. Boosting with Structure Information in the Functional Space

We capture the structure relationships among base learners as an undirected graph $G$, whose nodes correspond to the set of $p$ base learners. Edges in the graph $G$ are weighted, with $w_{i,j}$ indicating the “closeness” between the two features and 0 indicating that the two features have no relationship. We call the graph $G$ “feature graph” and explore approaches for building a feature graph in Section 4.2.

We incorporate the priori domain knowledge by adding a Tikhonov regularization factor $\frac{1}{2} \sum_{i,j} w_{i,j}(\beta_i - \beta_j)^2$ in a convex fitness function $\ell(\mathbf{X}; \mathbf{y}; \mathbf{\beta})$ to enforce that the feature coefficients vary smoothly for neighboring features. The Tikhonov regularization factor can be conveniently written in matrix format $\mathbf{\beta}^T \mathbf{L} \mathbf{\beta}$ where $\mathbf{L}$ is the Laplacian of $G$ given by: $\mathbf{L} = \mathbf{D} - \mathbf{W}$. $\mathbf{W}$ is the $p \times p$ edge weight matrix $W = (w_{i,j})_{i,j=1}^p$, and $\mathbf{D}$ is the density matrix of $\mathbf{W}$, defined as $\mathbf{D} = (d_{i,i})_{i,j=1}^p$ where $d_{i,j} = \left\{ \begin{array}{ll} 1 - w_{i,j}/d_{i,i} & \text{if } i = j \\ -w_{i,j}/\sqrt{d_{i,i}d_{j,j}} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{array} \right.$

(4)

Tikhonov regularization does not lead to the sparsity of the model. To obtain a sparse solution, we add the $L_1$ norm of $\mathbf{\beta}$ to the convex function $\ell(\mathbf{X}; \mathbf{y}; \mathbf{\beta})$. Specifically, we seek to identify a vector $\mathbf{\beta}$ that minimizes the following loss function:

$$g(\mathbf{X}, \mathbf{y}; \mathbf{\beta}) = \ell(\mathbf{X}, \mathbf{y}; \mathbf{\beta}) + \lambda_1 ||\mathbf{\beta}||_1 + \frac{1}{2} \lambda_2 \mathbf{\beta}^T \mathbf{L} \mathbf{\beta}$$

(5)

where $\lambda_1 > 0$, $\lambda_2 > 0$, $||.||_1$ is $L_1$ norm. In our implementation, we use the logit loss [9]:

$$\ell(\mathbf{X}, \mathbf{y}; \mathbf{\beta}) = \sum_{i=1}^n \log(1 + \exp(-y_i \sum_{j=1}^p \beta_j h_j(x_i)))$$

(6)

The major challenge in fitting the model described in Equation (5,6) to data is to estimate the parameter $\mathbf{\beta}$ efficiently and accurately. In the following subsection, we provide the optimization algorithm.

3.1 Optimization Algorithm

We discuss the optimization algorithm for Equation (5) bellow. We first show that the structurally regularized boosting with logit loss function can be interpreted as an additive logistic regression with the same regularization in the functional space spanned by base learners. We then provide the optimization algorithm based on coordinated decent to solve the equivalent regularized logistic regression towards the base learners. For simplicity, let $F(x) = \sum_j \beta_j h_j(x)$ be the decision function on the sample x. For a fixed training data set, we denote all the predicted labels for the training data using functions in $H$ as an $n \times p$ matrix $\mathbf{H}$, where $n$ is the sample size, $p$ is the number of base learners. $H_{l,i} = h_i(x_l)$ is the label given by base learner $h_i \in \mathbf{H}$ on the training sample $x_l$. We call $\mathbf{H}$ “object-prediction” matrix. We use $H_l$ to denote ith row of object-prediction matrix $\mathbf{H}$ (the predictions of all the base learners on the sample $x_l$), and $H_{l,j}$ to represent jth column of $\mathbf{H}$ (the predictions of $h_j$ on the training data).

We use the following Lemma to show that the minimizer of the expected loss function $J(F) = E(\log(1 + \exp(-y F(x))))$ is the symmetric logistic transform of $P(y = 1|x)$.

**Lemma 3.1.** $E(\log(1 + \exp(-y F(x))))$ is minimized at $F(x) = \log\left(\frac{P(y=1|x)}{P(y=-1|x)}\right)$. Hence $P(y = 1|x) = \frac{1}{1+e^{-F(x)}}$ and $P(y = -1|x) = \frac{1}{1+e^{F(x)}}$.

**Proof.** Since E imposes expectation over the joint distribution of $y$ and $x$, we have $E(\log(1 + \exp(-y F(x)))) = P(y = 1|x) \log(1 + \exp(-y F(x))) + P(y = -1|x) \log(1 + \exp(y F(x)))$. Then it is sufficient to minimize $J(F)$ by computing the first derivative with respect to $F(x)$:

$$\frac{\partial J(F)}{\partial F(x)} = \frac{P(y=1|x)e^{-F(x)}}{1 + e^{-2F(x)}} + \frac{P(y=-1|x)e^{F(x)}}{1 + e^{2F(x)}}$$

The result follows by setting the derivative to zero. \Box

With Lemma 3.1, the structurally regularized boosting can be interpreted as logistic regression with the same regularization function. Let $y^* = (y + 1)/2$, taking values of 0, 1, and parameterize the binomial probabilities by $P(y = 1|x) = p(x) = \frac{1}{1+e^{-F(x)}}$, it is sufficient to derive that the logit loss function is equivalent to negative binomial log-likelihood:

$$l_0(y^*, p(x)) = \left[ y^* \log(p(x)) + (1 - y^*) \log(1 - p(x)) \right]$$

$$= \log(1 + \exp(-y F(x)))$$

(7)

By plug $p(x)$ into (7), we can reduce (7) to $l_0(y^*, p(x)) = \log(1 + e^{F(x)}) - y^* F(x)$. Now we rewrite (5) in terms of negative binomial log-likelihood with $y^*$:

$$g(\mathbf{X}, \mathbf{y}; \mathbf{\beta}) = \sum_{i=1}^n \log(1 + \exp(\sum_j \beta_j h_j(x_i))) - y_i \sum_j \beta_j h_j(x_i) + \lambda_1 ||\mathbf{\beta}||_1 + \frac{1}{2} \lambda_2 \mathbf{\beta}^T \mathbf{L} \mathbf{\beta}$$

(8)

After transforming logit loss to negative binomial log-likelihood, we followed the general framework of coordinated decent algorithm proposed in [10] recently proposed by Friedman et al. for $L_1$ norm regularized logistic regression. Their approach relies on the connection between the Newton’s method for optimizing logistic regression and the least square formulation. The Newton’s method amounts to using Taylor expansion, up to a quadratic function, to approximate the logit function. In this way, applying Newton’s method can be viewed as solving a series of least squares problem (also called iterative reweighted least squares fitting [10]).

Applying Taylor’s expansion at current estimate $\mathbf{\beta}$ to negative log-likelihood function (7), we have the reweighted least square problem:

$$l_0(\mathbf{\beta}) = -\sum_{i=1}^n w_i (z_i - H_i \mathbf{\beta})^2 + C(\mathbf{\beta})$$

(9)
where $z_i = H_i \hat{\beta} + (\hat{y}_i - \hat{p}(x_i))/(\hat{p}(x_i)(1 - \hat{p}(x_i)))$, $w_i = \hat{p}(x_i)(1 - \hat{p}(x_i))$ and $C(\hat{\beta})$ is a constant.

In the remaining discussion, we show an extension of Friedman’s work to solve a reweighted least square fitting (9) with Laplacian weighted $L_2$ and $L_1$ norm regularization. To handle the new mixture penalty, we derive a modified coordinate descent scheme in Lemma 3.2 extending the work presented in [10].

**Lemma 3.2.** Suppose that the data set contains $n$ observations and $p$ predictors, with the response vector $Y = (y_1, \ldots, y_n)^T$ and the data matrix $X = (x_{11}, \ldots, x_{n1})$. We also assume that the predictors are standardized and the response is centered so that for all $j$, $\sum_{i=1}^n x_{ij} = 0$, $\sum_{i=1}^n x_{ij} = 1$ and $\sum_{i=1}^n y_i = 0$. The Lagrange form of the network constrained objective function (with least squares fitness function) is:

$$L(\lambda_1, \lambda_2, \hat{\beta}) = \frac{1}{2}(Y - X\hat{\beta})^T(Y - X\hat{\beta}) + \frac{1}{2}\lambda_2 \beta^T L\beta + \lambda_1 ||\beta||_1$$  \hspace{1cm} (10)

The coordinate-wise update has the form for each $\beta_j$: $\hat{\beta}_j = S(\sum_{i=1}^n x_{ij}(y_i - \hat{y}_i)) - \lambda_2 \sum_{k \neq j} L_{jk} \hat{\beta}_k + \lambda_1 / (1 + \lambda_2 L_{jj})$ where $\hat{y}_i = \sum_{k \neq j} x_{ik} \hat{\beta}_k \beta_i / \sum_{k \neq j} L_{jk} \hat{\beta}_k + \lambda_1$ is the fitted response value excluding the contribution from $x_{ij}$ and $S(z, \gamma) = \text{sgn}(z)(|z| - \gamma)_+$ is the soft thresholding operator where:

$$\text{sgn}(z)(|z| - \gamma)_+ = \begin{cases} 
  z - \gamma & \text{if } z > 0 \text{ and } \gamma < |z| \\
  z + \gamma & \text{if } z < 0 \text{ and } \gamma < |z| \\
  0 & \text{if } \gamma \geq |z|
\end{cases}$$

Suppose that we have estimates of $\hat{\beta}_j$ for $l \neq j$ and we wish to partially optimize the objective function with respect to $\beta_j$. We would like to compute the gradient at $\beta_j = \beta_j$, which only exists if $\hat{\beta}_j \neq 0$. If $\hat{\beta}_j > 0$, then the gradient for equation 10 is given by

$$\frac{\partial L(\lambda_1, \lambda_2, \hat{\beta})}{\partial \beta_j} = - \sum_{i=1}^n x_{ij}(y_i - \hat{y}_i) - \lambda_2 \sum_{k \neq j} L_{jk} \hat{\beta}_k + \lambda_1 / (1 + \lambda_2 L_{jj})$$

Since $X$ is standardized, by setting 11 to 10, we obtain $\hat{\beta}_j = \sum_{i=1}^n x_{ij}(y_i - \hat{y}_i) - \lambda_2 \sum_{k \neq j} L_{jk} \hat{\beta}_k - \lambda_1 / (1 + \lambda_2 L_{jj})$. A similar closed form exists for $\hat{\beta}_j < 0$. Combining two cases we will get Lemma 3.2.

We notice that our solution is not constrained in $L_1$ and $L_2$ penalty, but can be extended to $L_\infty$, which recently attracted research interest [4], since $L_\infty$ norm is differentiable everywhere except singular points ($\hat{\beta} = 0$) [45].

We summarize what is discussed previously in the algorithm called LPGB. Given the training data $T = \{X, \hat{y}\}$, the $n$ by $p$ object-prediction matrix $H = \{h_{ij}, \{h_{ij}(x_i)\}\}$ constructed from base learners, regularization parameters $\lambda_1$, $\lambda_2$ and convergence parameter $\epsilon$, our algorithm iteratively solves (8). Here we transform $\hat{y}$ to $\hat{y}$ using 0/1 to present the outcome and $p(x) = P(y = 1|x) = P(y = 1|x) = 1/(1 + \exp(-\sum_{i=1}^p \beta_i h_i(x)))$.

As evaluated in our experimental study in Section 5, the regularized LPGB algorithm usually has better classification performance and are insensitive to outliers and class label noises, comparing to the unregularized gBoosting [20]. We believe that these advantages are contributed to the capability of LPGB to select clustered base learners in the functional space. We call this phenomenon the “grouping effect” and we provide theorems to explain the “grouping effect” below. Our proof is similar to that presented in [22] where we consider a simple case of two base learners that are linked. We show that the related $L_2$ regularization ensures that the difference of the estimated coefficients have an upper bound based on the sample size and the regularization coefficients.

### 3.2 Grouping Effect

We derive an upper bound of the difference of coefficients between two neighboring features. Motivated from a similar proof in [22] where a linear regression framework with $L_1$ and $L_2$ regularization, we study the special case in which only two features are connected to each other in the feature graph.

**Theorem 3.3.** Given training data $T = \{x_i, y_i\}_{i=1}^n$ where $x_i \in X$ and fixed scalars $\lambda_1$, $\lambda_2$ and let $\hat{\beta}(\lambda_1, \lambda_2)$ be the optimal solution to (8), suppose that $\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2) > 0$, and the two features $F_i$ and $F_j$ are only linked to each other on the feature graph. Define $D_{\lambda_1, \lambda_2}(i, j) = |\hat{\beta}_i(\lambda_1, \lambda_2) - \hat{\beta}_j(\lambda_1, \lambda_2)|$, then $D_{\lambda_1, \lambda_2}(i, j) \leq 2n/\lambda_2$.

**Proof:** Since $\hat{\beta}(\lambda_1, \lambda_2)$ is the optimal solution to (8), $\hat{\beta}(\lambda_1, \lambda_2)$ satisfies $\frac{\partial^2 L(\hat{\beta}, y_i, \lambda_1, \lambda_2)}{\partial \hat{\beta}_i \partial \hat{\beta}_j} \{\hat{\beta}_i(\lambda_1, \lambda_2) = 0 \text{ if } \hat{\beta}_k(\lambda_1, \lambda_2) \neq 0 \text{. More specifically, for } \hat{\beta}_i \text{ and } \hat{\beta}_j, \text{ we have}

$$-H_i^T (\hat{y} - \hat{p}(X)) + \lambda_1 \text{sgn}(\hat{\beta}_i) + \lambda_2 \hat{\beta}_i$$

$$-\lambda_2 \sum_{u \neq i} \frac{w_{u,i}}{\sqrt{d_{u,i} d_{i,i}}} = 0 \hspace{1cm} (12)$$

$$-H_j^T (\hat{y} - \hat{p}(X)) + \lambda_1 \text{sgn}(\hat{\beta}_j) + \lambda_2 \hat{\beta}_j$$

$$-\lambda_2 \sum_{v \neq j} \frac{w_{v,j}}{\sqrt{d_{v,j} d_{j,j}}} = 0 \hspace{1cm} (13)$$

where $\hat{p}(X) = 1/(1 + \exp(-H \hat{\beta}))$, $\hat{y} = (\hat{y} + \hat{1})/2$ and $H$ is the object-prediction matrix. Subtracting (12) from (13) and taking the absolute value with the assumption that $d_{i,i} = d_{j,j} = w_{i,i}$ and $\text{sgn}(\hat{\beta}_i) = \text{sgn}(\hat{\beta}_j)$ gives

$$|\hat{\beta}_i - \hat{\beta}_j| = \frac{|H_i^T - H_j^T| |\hat{y} - \hat{p}(X)|}{\lambda_2} \hspace{1cm} (14)$$

Now consider the worst case, for any $x \in X$, $h_i(x) \neq h_j(x)$, also $|\hat{y} - \hat{p}(X)| \leq 1$, hence $D_{\lambda_1, \lambda_2}(i, j) = |\hat{\beta}_i - \hat{\beta}_j| \leq \frac{2n}{\lambda_2}$.
The upper bound of $D_{\lambda_1, \lambda_2}(i, j)$ provides two insights of our method: 1) smoothness: the coefficients of neighboring base learners are close to each other due to the $L_2$ norm regularized feature graph Laplacian penalty term. 2) Grouping effect: Once a base learner is selected, its spatially neighboring base learners will be more likely selected. Thus our boosting algorithm can select groups of spatially neighboring base learners.

4. APPLICATION TO GRAPH DATA

We show how to apply the LPGB algorithm to graph classification below.

4.1 Base Learner Construction

In our model, we use frequent subgraphs as features and construct base learners (decision stamps) from these features. Given training data $\{X, \tilde{y}\}$ and a set of frequent subgraphs, the decision stamp classifier for subgraph $F_i$ is given by:

$$h_i(x) = \begin{cases} \tilde{y} & \text{if } F_i \subseteq x \\ -\tilde{y} & \text{otherwise} \end{cases}$$

The prediction $\hat{y}$ for $F_i$ given training data $X$ is found by:

$$\hat{y} = \arg\max_{y \in \{\pm 1\}} \sum_{i=1}^{n} y_i h_i(x_j)$$

This criteria is to perform a majority voting to obtain prediction of the decision stamp based on the percentage of positive (or negative) graphs where the feature occurs. gBoosting [20] uses a similar strategy to construct base learners.

4.2 Feature Graph Construction

One challenge of processing graph data is that there is no natural approach to define the structure relationship of base learners. We notice a few recent studies that are moving towards the direction of defining the relationship among features in graphs and sets. For example, in the recently defined graph Graphlet Spectrum kernel [19], the spatial relationship of graph feature (called graphlets) are explored in an algebraic framework for measuring the structure similarity of graph adjacency matrices. In addition, recently developed association net uses a graph model to represent a set of association rules [29]. However, these work could not be directly applied in our current framework since the graphlet spectrum method models the spatial relationship of graphlet in an implicit approach and the association rule net only explore the overlapping relationship of features.

Here we adopted our previous work [5, 6] to construct feature graphs. In [5], we formalize a concept which we called "feature consistency map". A feature consistency map is a undirected graph in which each node represents a feature and each edge encodes the spatial consistency relationship between two features. We measure the minimum distance between two features using the average shortest path connecting a node in one feature to a node in the other feature. We compute the variance of the minimal distance between the occurrences of the two subgraphs in the training data. If the variance is bellow a threshold, we consider the two features are in a consistent spatial relationship. In our experiment study, we adopt the feature consistency map as an approach to construct a feature graph.

In addition, we also explored the possibility of evaluating the structure-overlapping relationship of features as did in [6]. Towards that end, we compute a kernel function for the set of features. A graph kernel function is a positive semi-definite function that maps graphs to a Hilbert space in order to evaluate the similarity of graphs in the space. Many kernel functions have been designed for graphs and we use the random walk based Marginalized Graph kernel function [17] to compute the kernel function for the set of subgraph features. We convert such kernel matrix to a feature graph where nodes are features and edges are labeled with the inner product (as evaluated with a graph kernel function) of the two features. To avoid a complete connected graph, we use a threshold. If the inner product between two features is less than a threshold, we set the weight of the edge to zero (and hence canceling the edge). The aforementioned approach provides another way to construct a feature graph.

5. EXPERIMENTAL STUDY

We have performed a rigorous evaluation of our algorithm in terms of modeling accuracy and feature selection performance using 6 Protein structure data sets, obtained from [16]. We implemented a prototype of our method in Matlab. We have compared our method with state-of-the-art methods including Support Vector Machine Recursive Feature Elimination (SVM-RFE) [11], gBoosting [20], graph partial least square regression (gPLS) [31], graph classification based Pattern Co-occurrence (COM) [16]. We obtained the SVM-RFE executable along with the spider machine learning toolbox from http://www.kyb.tuebingen.mpg.de/bs/people/spider/. For gBoosting, we use the gboost toolbox [30]. We obtained gPLS and COM directly from the original authors of the methods. All the experiments were conducted on a PC with a 2.8Ghz duo core CPU and 3GB memory.

5.1 Data sets

To evaluate our methods, we utilized 6 protein-structure graph data sets that were originally studied in [16]. Each data set is a set of geometric graphs representing a set of three-dimensional protein structures. Nodes in such graphs represent amino acids in a protein structure and are labeled with the amino acid type. Edges represent the pairwise Euclidian distance of amino acids (defined between $C_{\alpha}$ atoms) and are labeled with the discretized distances.

Graphs in the data sets are labeled. Positive samples are sampled from a selected protein family. Negative samples are randomly sampled from the Protein Data Bank. On average a graph contains 250 nodes and 1600 edges. Protein-structure graphs are much larger than chemical-structure graphs, which usually contain about hundreds of nodes and thousands of edges, and contain much large number of patterns. Working with protein structure graphs are hence more challenging for constructing sparse predictive models.

In Table 1, we summarize the characteristics of the 6 protein-structure graph data sets. For each data set, we list the data set index, the related protein family ID in the SCOP database [27], the description of the protein family, the number of positive samples and the number of negative samples. See [16] for a comprehensive description of the data collection process.
max the feature consistency map, as investigated in [5]. We fix nel [17]. The second approach, LPGBCMP, is to construct tion for the subgraphs, utilizing the the Marginalized ker-

The first approach, LPGBK, is to construct a kernel func-
tion of base learners (i.e. subgraphs). We perform feature selection using SVM, SVM_RFE, and LibSVM [1] with linear kernel to construct the best model. We use 5-fold cross validation in the training data set to select important parameter C for SVM.

For COM, we set tp = 0.3 and tn = 0 as proposed in [16], where tp is the minimal positive frequency for a classification rule and tn represents the maximal negative frequency permitted. For gPLS, we use min_sup = 0.3 and examine the combinations of n = {2, 4, 8, 16} and δ = {2, 4, 8, 16} for optimal setting. For gBoosting, we also set min_sup = 0.3 and search the optimal parameter µ (misclassification cost) in the range of [0.04, 0.06, ..., 0.18, 0.20]. All the parameter selection are based on another 5-fold cross validation on the training data only.

For our own methods, we utilize two approaches to model the spatial correlation of base learners (i.e. subgraphs). The first approach, LPGBK, is to construct a kernel function for the subgraphs, utilizing the the Marginalized kernel [17]. The second approach, LPGBCMP, is to construct the feature consistency map, as investigated in [5]. We fix max_var = 1 for feature consistency map building threshold and δ = 0.25 for overlapping threshold. Empirical study shows that there is no significant change if we change these two parameters within a wide range. Further details of the two spatial correlation computation methods can be found in [5, 6].

Below we summarize the model construction and model evaluation.

Model Construction. For each data set, we partition the data set into 5-folds to perform 5-fold cross-validation (CV) with 4 folds for training and 1 fold for testing. We use another 5-fold CV on the training data set to select the optimal parameters for each method. We then generate a single model from the entire training set with the selected parameters and apply the model to the testing data set for prediction.

Model Comparison. For model comparison, we collect the sensitivity (TP/(TP+FN)), specificity (TN/(TP+FP)) and accuracy ((TP+TN)/S) of the trained model, where TP stands for true positive, FP stands for false positive, TN stands for true negative, FN stands for false negative, and S stands for the total number of samples. All the values reported are collected from the testing data set only and are averaged across 10 replicates of the 5-fold cross validation in a total of 50 experiments.

5.3 Classification Performance

In this subsection, we show the performance of our methods compared with SVM-RFE, gPLS, gBoosting and COM. The accuracy is shown in Fig 2. Since the standard deviation is around 2%-5% for all these methods, we do not list it here.

![Figure 2: Accuracy comparison of on 6 data sets.](image)

In Fig 2, we observe that the accuracy of all these methods has the same trend with different data sets. gBoosting and gPLS have comparable performance in the 6 data sets. SVM_RFE outputs gBoosting, gPLS, and COM in three out of six data sets and have comparable performance for the rest. Comparing two versions of our methods, LPGBCMP outperforms LPGBK on all data sets. In fact LPGBCMP performs best among all the evaluated data sets though the margin may be small for 3 data sets when compared with SVM_RFE.

To better understand the accuracy differences, we plot the average sensitivity and average specificity of all methods in Fig 3. It is clear that COM provides the best sensitivity among the majority of data sets. COM utilizes a rule-based classification algorithm where it classifies a graph sample as positive if a co-occurrence pattern-rule is satisfied. This algorithm is not specific enough, as compared to other methods (shown in the right panel of Fig 3). Interestingly enough, all boosting based methods, including gBoosting, LPGBCMP, and LPGBK, have very high specificity comparing to the rest of the methods. Overall, the regularized boosting methods such as LPGBCMP and LPGBK seem to have a good compromise between specificity and sensitivity. This observation provides experimental evidence supporting our hypothesis that structure information among base classifier should be considered in order to build a highly accurate predictive model for semi-structure data such as graphs.

5.4 Grouping Selection Effect and Stable Spatial Distribution

To evaluate the capability of the LPGBCMP algorithm for selecting grouped base learners, we visualize the spatial distribution of selected base learners in original graphs. By
ranking the base learners by the learned coefficients, we select the top three features for LPBCMP and gBoosting. We plot the embedding of the three subgraphs in two proteins: protein 1EGI and protein 1H8U belonging to the same protein family in Fig 5. We rotate the protein structures and highlight the occurrence of the features with circles for a better demonstration.

In Fig 5, the upper row shows the spatial distribution of the top three features for LPBCMP in two proteins and the lower row shows the distribution for the top three features from gBoosting. Each column uses the same protein for demonstration. From Fig 5, we observe that $F_1$, $F_2$ and $F_3$ from LPBCMP have a consistent spatial distribution on the two proteins. $F_1$ and $F_2$ are clustered and both are close to $F_3$. In contrast, features from gBoosting do not have a stable spatial distribution in the two proteins. The observation supports our claim that our method can select grouped features with stable spatial distribution among the graph data.

### 5.5 Method Robustness

A common concern with boosting is that the method is usually sensitive to outliers and errors in the training data set due to the exponential loss function. We use logit loss function that is less sensitive to outliers. However, as claimed in [25], any convex loss function may degenerate to random guess with a certain level of random classification noise. $L_2$ regularization in linear regression has been shown to stabilize the learning function [13]. In our algorithm design, we used the Laplacian based $L_2$ regularization and this may reduce the boosting algorithms’ sensitivity to outliers and random classification noise. To test the robustness of our method experimentally, we singled out the $P_4$ data set and performed 5 fold cross validation with class label errors. In particular, for each fold, we change certain percentage of the class labels in the training data, train a model with changed training data, and apply trained model to normal test data.

In Fig 4, we report the average accuracy with error rate ranging from 0% to 25% for LPBCMP, gPLS, gBoosting, SVM_RFE and COM. From Fig 4, a clear trend is that the accuracy of all methods decreases as more errors are introduced in the training data set. There is a sharp deceasing from 0 to 5% for SVM_RFE and COM. The regularized boosting method remains the best over all the settings, even though the performance gain is not significant. From the test, we conclude that LPBCMP is at least as sensitive (if not less) to noises as other classifiers including SVM and partial least square based methods.

In addition, we evaluate the robustness of the regularized boosting algorithm by changing different parameter values. Among the parameters that may affect the performance of the regularized boosting algorithm, we test the parameter $\text{max}_{\text{var}}$, which is used to derive the feature consistency map. With a large value of $\text{max}_{\text{var}}$, the edge number of feature consistency map increases and with smaller value of $\text{max}_{\text{var}}$, the edge number of feature consistency map decreases.

Fig 6 indicates average accuracy on 5 fold cross validation for each value of threshold $\text{max}_{\text{var}}$ from 0.5 to 8. From the result, we observe that the accuracy remains stable within a relatively wide range of threshold and the best accuracy can be obtained around 1 to 2. Furthermore, the relation-
ship between the performance and parameter $\text{max}_\text{var}$ is revealed. When $\text{max}_\text{var}$ is quite small, the structure information among features is ignored and our method will degenerate to regular logit boosting with elastic net regularization [44]; when $\text{max}_\text{var}$ is large, the feature graph will be a complete graph and our method may possibly introduce less discriminative features hence undermine the performance.

Overall, the regularized boosting method is effective and achieves good accuracy within a wide range parameters and a certain number of outliers.

6. CONCLUSIONS

In this paper, we presented a novel boosting algorithm that considered the structure relationship of base learners in the functional space. We model the structure relationship as an undirected graph and incorporate such information by introducing a $L_2$ norm regularized graph Laplacian to standard boosting formalization. Though the new algorithm may be applied to many applications, we specifically focus on constructing supervised graph learning models in this paper. Using a comprehensive experimental study with protein structure graphs and comparing with current state-of-the-art, we demonstrate that the new algorithm selects clustered features with stable spatial relationship, and achieves better predictive performance. In the future we plan to extend the current work to constructing boosting algorithms for other types of data such as trees and sequences, where the base learners have a structure relationship in the functional space.

Acknowledgments

This work has been supported by an Office of Naval Research award (N00014-07-1-1042) and the National Science Foundation under Grant No. 0845951.

7. REFERENCES


