Error-Bounded Graph Construction for Semi-supervised Manifold Learning*

Christopher T. Symons
Oak Ridge National Laboratory
Oak Ridge, Tennessee
symonsct@ornl.gov

ABSTRACT

Graphs are commonly used in semi-supervised learning to represent a manifold on which the data reside in a high-dimensional ambient space. The graph can then be utilized in different ways, typically via the Laplacian of the graph, in order to leverage associations among the unlabeled data to improve learning. One common way to leverage the graph Laplacian is as a regularization term, where models that would disagree with the graph are penalized. More often the spectrum of the graph Laplacian is used to find a lower dimensional embedding in which neighboring relations encoded via the graph are preserved. Most manifold-based methods of semi-supervised learning depend upon geometric structure in the ambient feature space in order to construct a graph whose edges encode similarity that should be useful in selecting a model. A critical assumption is that some standard measure of similarity applied to the ambient space can be used to construct a graph that is error-free or of low error, meaning that examples (i.e., vertices) from distinct classes are not connected. However, this assumption often precludes the use of such methods in noisy or complex feature spaces, even though such spaces often arise in problems that can most benefit from structure that might be uncovered within the unlabeled data. This paper presents a method of graph construction for manifold-based semi-supervised learning that respects the manifold assumptions underlying these methods and bounds the error on the graph itself, which then permits bounds on the overall generalization error of the learning algorithms without relying on assumptions that do not hold in many modern problem domains.

CCS CONCEPTS

• **Theory of computation** → **Semi-supervised learning**; • **Computing methodologies** → *Spectral methods*; *Regularization*;

KEYWORDS

learning theory, manifold learning, semi-supervised learning, graphs

ACM Reference Format:
Christopher T. Symons. 2018. Error-Bounded Graph Construction for Semi-supervised Manifold Learning. In *Proceedings of KDD Workshop on Mining and Learning with Graphs (MLG '18).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In most applications that are likely to benefit from semi-supervised learning [6], unlabeled data samples are plentiful, while there are limits on the ability to obtain labeled data, or very significant costs involved. The hope is that the unlabeled data can be leveraged in a manner that reduces the sample complexity of the learning problem. While there are many cases where semi-supervised learning has been applied successfully, there are still many instances where its application can lead to worse results than supervised learning alone. Thus, understanding the conditions under which unlabeled data can be used to confidently reduce error in machine learning remains a significant problem in many application areas.

Many modern data analysis problems are high-dimensional, and the dimensionality continues to grow as our ability to measure phenomena increases in both scope and granularity. These high dimensional spaces create new problems for data analysis due to the *curse of dimensionality* [7]. However, it is often the case that the degrees of freedom along which the data points vary are much less than the input dimensionality. Manifold learning [15] takes advantage of this phenomenon to find a low dimensional space that allows for better understanding or utilization of the data, and it plays a prominent role in many pattern recognition and machine learning algorithms.

In particular, the manifold assumption plays a prominent role in many semi-supervised machine learning algorithms. Most often graphs are used to represent these manifolds. However, constructing a graph that represents a good manifold is difficult in noisy data representations, particularly if there is little labeled data to focus the analysis. In this context, better methods of graph construction are necessary in order to apply semi-supervised learning more robustly. Moreover, more complete theoretical analyses that include an analysis of the graph are required in order to ensure that the usage of unlabeled data through the graph is not likely to hurt the performance of the learning algorithms. This paper presents a method of graph construction that respects the manifold learning assumptions, while simultaneously providing error bounds on the graph itself, which leads to overall error bounds on many graph-based semi-supervised learning algorithms.

MLG '18, 2018

https://doi.org/10.1145/nnnnnnn.nnnnnnn

2 BACKGROUND

Graph-based methods of semi-supervised learning [23] typically use a graph consisting of nodes that represent examples and edges that are meant to indicate some kind of similarity that should preserved or respected when choosing a final model in whatever hypothesis space is being utilized. Traditionally, these graphs, which are meant to encode the information that one hopes to leverage using the unlabeled points, have been constructed without regard to the target learning problem (just as one would expect in a standard unsupervised step). However, there are many applications where use of the unlabeled data is hampered by the fact that there is no known similarity metric that is guaranteed to have low error with respect to the target learning problem. In fact, in many modern data analysis problems, the target problem is a high-dimensional learning problem in which many of the features are of unknown utility. Therefore, without utilizing information about the target problem, it is impossible to use these features to construct an unsupervised similarity metric that is guaranteed to have low error.

2.1 Graph construction

Many popular forms of semi-supervised learning rely on graph-based methods for regularization or spectral dimensionality reduction. Some common forms of nonlinear dimensionality reduction include Isomap [21], Locally-Linear Embedding (LLE) [17], Laplacian Eigenmaps (LEM) [3], Diffusion Maps (DM) [16], Semidefinite Embedding [22], etc. A more comprehensive list can be found in [15]. A central construct in many of these methods is the graph Laplacian. A graph is constructed to represent a manifold (or densely populated region of interest in the ambient space), and the graph Laplacian facilitates the discovery of a low-dimensional space that is smooth with respect to this graph. In semi-supervised learning the goal is to augment learning through the use of unlabeled samples, so the unlabeled data is used to find a low-dimensional space on which learning via the labels can be more effective.

As emphasized in [10], normalized-output algorithms, such as LLE, LEM, and DM, do not handle noise well, and should not be applied arbitrarily, and there is a need for improvements that are robust. The authors in [8] recognize the potential problem of having data that resides on multiple manifolds and offer some methods for applying semi-supervised manifold learning in such cases. However, while the methodology adds some robustness in such multi-manifold cases, the manifold representation is made without the use of the labels so that high levels of noise can still hide the manifolds. Furthermore, it is quite possible that many of these discoverable manifolds are not relevant to the target problem.

There is a general recognition that the labels can be used as constraints when building the graph, but in semi-supervised learning, such an approach would touch only a small portion of the graph. In [5], the authors allows labeled points to act as hubs, such that addition of edges that link to labeled points are given priority. In [24], a low-rank representation is used where the coefficients between labeled points are zero when they belong to different classes. In [9], a dissimilarity measure is used to alter the graph. However, the method either requires ground truth dissimilarity, in which the only parts of the graph that are affected are those for which labels are available, or dissimilarity based on some domain specific features that are manually constructed to enforce disparity.

Recent work intended to leverage the target problem (i.e., the labels) to influence the graph construction [5, 20, 24] recognize the fact that there may be multiple manifolds, some of which are not ideal for discrimination in the target learning problem or that finding a manifold in noisy data requires additional information. However, none of these previous methods can supply an error-bound on the graph itself. While it might at first seem that providing an error bound on the graph is straightforward, there are other more subtle aspects to constructing the graph that turn out to be vitally important in manifold learning. When constructing a graph to represent a manifold, it is important to make sure that the graph respects a local notion of similarity, both in terms of respecting the notion of a geodesic distance that matters in the graph, and in terms of respecting the fact that for most learning problems there are subclasses within any class, and even if these subclasses lie in close proximity to each other in the ambient feature space (which is not guaranteed at all), if one attempts to treat them equally in a graph it may make it impossible to find a manifold that is good for learning (i.e., that results in a model that generalizes well).

This paper explores a form of graph construction based on Kleinberg's Stochastic Discrimination algorithm. We provide error bounds on the learning algorithm that include bounds on the graph, and we demonstrate the ability of this method to substantially improve performance on a very challenging semi-supervised learning task in a noisy application.

3 STOCHASTIC DISCRIMINATION

Stochastic Discrimination (SD) [11, 12, 14] is an ensemble method of classifier construction, in which so-called *weak* classifiers are combined to make a higher-level model. SD differs in significant ways from standard methods of combining classifiers. Most notably, it is well known for its ability to support complex models without overfitting.

Selection of the weak learners is guided very strictly by three overarching principles: *generalization*, *uniformity*, and *enrichment*. In order for a model to have the ability to generalize, weak models must cover enough space to capture points outside of the training data. In other words, the weak models must apply to test points, such that standard generalization assumptions apply. A weak model must also have at least some discriminatory power, even if its error rate is close to fifty percent. Thus, an enriched model is one that contains a greater fraction of the labeled points from one class than from the other class. This does not simply mean that it has more of one class than the other; rather it means that the percentage of all points of class 1 covered by the weak model is greater than the percentage of all class 2 points that it covers. The amount of enrichment that one requires a new weak learner to have can be set using a parameter, β, that defines the minimum difference between the coverage percentages of the two classes. Finally, the algorithm seeks to ensure uniformity of coverage of points. This uniformity is class-specific, such that we won't add a new model to the ensemble even if it is enriched, unless the average coverage of points of each class is less than the average coverage for that class so far (plus some constant λ).

The following definitions are taken from [14]:

Definition 3.1. **Enrichment**: A subset of the feature space (i.e., a weak classfier) $\mathcal{M}$ of F is said to be enriched with respect to classes C_1 and C_2 if
$inf\{|Pr(M|C_1) - Pr(M|C_2)| \; |M \in \mathcal{M}\} > 0$

Definition 3.2. **Uniformity**: A subset of the feature space (i.e., a weak classfier) $\mathcal{M}$ of F is said to be uniform with respect to classes C_1 and C_2 if for every point, p, in either C_1 or C_2, and every nonempty subset of $\mathcal{M}$ of the form $\mathcal{M}_{x,y}, Pr_F(p \in M | M \in \mathcal{M}_{x,y})$ is equal to x if p is a member of C_1, and is equal to y if p is a member of C_2.

The idea is to generate an ensemble model consisting of a wide variety of subsets of the feature space, such that those subsets (classifiers) cover points evenly, cover enough space to provide generalization ability, and cover a disproportionate number of points from one class, C_1, or the other, C_2.

To turn these ensembles into classifiers, a new test point is evaluated based on the subsets that cover it, as well as those that do not. Each subset contributes the following score to the point:

$$X_{(C_1,C_2)}(p,S) = \left(\frac{1_S(p) - Pr(S|C_2)}{Pr(S|C_1) - Pr(S|C_2)} \right), \tag{1}$$

where 1_S is the indicator function of the set S. In other words $1_S(p) \mapsto 1$ for points $p \in S$, and $1_S \mapsto 0$ for points $p \notin S$.

Then, points are classified using the following sum:

$$Y^t_{(C_1,C_2)} = \frac{\left(\sum_{k=1}^{t} X^k_{(C_1,C_2)} \right)}{t}, \tag{2}$$

where t is the number of classifiers.

It is possible in this fashion to cause points of class C_1 to have a mean of 1.0 and points of class C_2 to have a mean of 0. New points can then be classified as belonging to class C_1 when $Y^t_{(C_1,C_2)} > 0.5$. By the Central Limit Theorem, as t approaches infinity, the variance of the probability density function of points of class C_1 and the variance of the probability density function of points of class C_2 both approach zero. This fact helps explain the resistance to overfitting as the number of weak classifiers is increased.

The SD algorithm as described in [13, 14] can be implemented in a variety of ways. In particular, the performance is heavily dependent upon how the stream of weak classifiers is generated and which classifiers are retained. One significant choice we make in our implementation is that we grow weak classifiers around neighboring points rather than choosing the expansion points at random. The motivation for this is the fact that we want to use SD to generate a psuedometric that is both target-based and local-proximity-based. This is because manifold learning is based on the assumption that geodesic proximity should relate to smooth changes in the feature space.

4 GRAPH CONSTRUCTION USING SD

The characteristics we want to require in our graph construction approach are resistance to overfitting, since we won't have much labeled data, and lack of correlation between ensemble members to ensure a globally applicable psuedometric. Among ensemble classification methods, there are two prominent approaches that have both of these properties. The first, is the well-known AdaBoost algorithm [18, 19]. The second is Stochastic Discrimination [11, 13, 14]. AdaBoost ensures that ensemble-members' errors are not correlated by adding more weight in the next round to examples on which the current ensemble makes mistakes or is unsure. While AdaBoost modifies the training set from the example point of view, SD modifies it from the feature point of view. Thus, SD is a natural fit for what we want to do, while AdaBoost is not.

As in the more general random subspace method used in [20], we can use Stochastic Discrimination to generate a task-relevant psuedometric. Kleinberg discusses the point that weak classifiers generated via SD are not classifiers in the traditional sense of the word. Similarly, we don't want classifiers in the traditional sense of the word either. Just as SD depends on the weak learners being error prone, we do too. In other words, if each SD learner was highly accurate, we would likely be joining together many points that, while sharing the same class, are not close in the sense of where they lie on the manifold that we want to discover.

Owing to the over-fitting resistance of SD, we are able to benefit from additional *weak classifiers* as they help lower the overall bound on the generalization error without any real risk of hurting performance. In addition, because the weak classifiers are error prone, we can obtain a fine-grained pseudometric simply by using the scores (Equation 1) generated by the weak classifiers covering any given two points. This pseudometric captures local proximity due to the way we create our stream of classifiers, and any bound on the generalization error that applies to the SD algorithm applies to our graph edges as well, with a sufficient number of weak classifiers and a sufficient number of unlabeled points.

The graph construction algorithm is shown in Algorithm 1.

Algorithm 1 SD Graph Construction

Input: data $\{(x_i, y_i)\}_{i=1}^{l}$, $\{x_i\}_{i=l+1}^{n}$, $numNeighbors := k$, $numWeakClassifiers := c$, β, λ, $minPoints := q$
Train SD classifier: $SD(c, \beta, \lambda, q)$
for $u = 1$ to n **do**
  for $v = 1$ to n **do**

$$w_{u,v} = \sum_{i=1}^{c} \begin{cases} \frac{1_{S_i}(u) - Pr(S_i|C_2)}{Pr(S_i|C_1) - Pr(S_i|C_2)}, & \text{if } 1_{S_i}(u) = 1_{S_i}(v) \\ 0, & \text{otherwise} \end{cases}$$

  end for
end for
Retain k nearest neighbors; create Laplacian matrix $L(u, v)$

5 THEORETICAL ANALYSIS

The framework described in this section can be considered to rely on the notion of compatibility, χ, as described in [1, 2]. The notion of compatibility is based on finding a model that has a low *unlabeled error rate*. In the case of a graph regularizer, this can indicate that the function being learned *agrees with the graph* and would not label two connected nodes with different class labels. Of course, if the graph incorrectly connects examples from different classes, then the target function itself does not have an unlabeled error rate of zero, even if some hypotheses do.

In [1], various sample complexity bounds are provided. In some cases an assumption is made that the target function's unlabeled error rate is low (essentially zero), and in other cases the bounds depend on the unlabeled error of c^*, the true target function. For example, Theorem 2.3.2 provides a sample complexity bound in the realizable case ($c^* \in C$) that depends upon the unlabeled error of the target, c^*. A graph constructed over noisy samples is likely to have many "errors." Therefore, the first assumption is too simplistic for many real-world situations. Using unlabeled data alone, the target function's unlabeled error cannot be bounded at all, since it is entirely possible that similarity in the ambient feature space does not reflect similarity in terms of the target concept at all. In other words, the number of mistakes in the notion of compatibility itself (the graph) cannot be bound while ignoring all information concerning the target concept. Although labeled and unlabeled error are of different types, it should still be possible to use supervised-learning bounds on generalization error to provide a bound on the unlabeled error rate of c^*, meaning that use of label information in the construction of the graph can bound this error with respect to the target, allowing bounds to be applied to the overall sample complexity.

We will use the definition of a *notion of compatibility* given in Definition 2.2.1 from [1], which we restate here, in slightly modified form, for completeness.

Definition 5.1. A notion of compatibility is a function $\chi : C \times X \mapsto [0, 1]$ where we define $\chi(f, D) = \mathbf{E}_{x\sim D}[\chi(f, x)]$. Given a sample S, we define $\chi(f, S)$ to be the empirical average of χ over the sample.

C is the *hypothesis space* from which the hypothesis, f, is chosen, and X is the *instance space*, from which the distribution, D, is drawn. For our purposes, we will actually assume that $\chi(f, D) = \mathbf{E}_{x\sim D}[\chi(f, x_i, x_j)]$, so we are looking at the expectation over pairs of examples (edges in the graph). Note that the *unlabeled error rate* is simply a measure of incompatibility between a hypothesis, f, and the distribution, D; i.e. $1 - \chi(f, D)$, or $1 - \chi(f, S)$, for a given sample.

Now, consider Theorem 2.3.2 from [1], which we restate verbatim here for completeness.

Theorem 5.2. *If $c^* \in C$ and $err_{unl}(c*) = t$, then m_u unlabeled examples and m_l labeled examples are sufficient to learn to error ϵ with probability $1 - \delta$, for*

$$m_u = \frac{2}{\epsilon^2}\left[ln|C| + ln\frac{4}{\delta}\right] \text{ and } m_l = \frac{1}{\epsilon}\left[ln|C_{D,\chi}(t + 2\epsilon)| + ln\frac{2}{\delta}\right].$$

In particular, with probability at least $1 - \delta$, the $f \in C$ that optimizes $\hat{err}_{unl}(f)$ subject to $\hat{err}(f) = 0$ has $err(f) \leq \epsilon$.

Alternatively, given the above number of unlabeled examples m_u, for any number of labeled examples m_l, with probability at least $1-\delta$, the $f \in C$ that optimizes $\hat{err}_{unl}(f)$ subject to $\hat{err}(f) = 0$ has

$$err(f) \leq \frac{1}{m_l}\left[ln|C_{D,\chi}(err_{unl}(C^*) + 2\epsilon)| + ln\frac{2}{\delta}\right].$$

Next, we need the following definition from [13]:

Definition 5.3. An m-class supervised learning problem presented as two finite sequences $\mathbf{E} = (E_1, E_2, \ldots, E_m)$ and $\mathbf{T} = (T_1, T_2, \ldots, T_m)$ of classes in a finite feature space (intuitively, all examples and the training examples, respectively), is said to be solvable if there exists a collection $\mathbf{M}$ of subsets of the feature space such that $\mathbf{T}$ is $\mathbf{M}$-representative of $\mathbf{E}$, and such that $\mathbf{M}$ is $\mathbf{T}$-enriched and $\mathbf{T}$-uniform.

Note that *enrichment* and *uniformity* are as defined above.

Now, consider Theorem 1 from [13], which we also restate essentially verbatim here for completeness.

Theorem 5.4. *There exists an algorithm $\mathcal{A}$ with the following property: given any solvable problem, $\mathbf{E}$, $\mathbf{T}$, in supervised learning, if $\mathbf{M}$ is a collection of subsets of the feature space, such that $\mathbf{T}$ is $\mathbf{M}$-representative of $\mathbf{E}$, and if $\mathbf{M}$ is $\mathbf{T}$-enriched and $\mathbf{T}$-uniform, then given any desired upper bound u on the error rate, $\mathcal{A}$ will output, within time proporational to $\frac{1}{u}$ and inversely proportional to the square of $e(\mathbf{T}, \mathbf{M})$ (the $\mathbf{T}$-enrichment degree of $\mathbf{M}$), a classifier whose expected error rate on $\mathbf{E}$ is less than u.*

The algorithm $\mathcal{A}$ builds classifiers by sampling, with replacement, from the set $\mathbf{M}$, and then combining the "weak classifiers" in the resulting samples. We reduce n-class problems to n-many two-class problems; given a training pair (T_1, T_2) for any such two-class problem, a sample S of size t produces the classifier which assigns any given example q to class 1 if

$$\frac{1}{t}\sum_{S\in\mathbf{S}} \frac{1_S(q) - Pr(S|T_2)}{Pr(S|T_1) - Pr(S|T_2)} > 0.5, \tag{3}$$

(where $1_S(q)$ is the indicator function of the set S).

Note that the phrase $\mathbf{M}$-representative in the above theorem, just means that the set of all examples, E_i, of class i is indistinguishable from the set of training examples, T_i, for that class when using the sets in $\mathbf{M}$.

Now, we can combine the two theorems, by building a Stochastic Discrimination graph using Algorithm 1, such that vertices concur with the SD classifier, which allows us to bound the error on edges; i.e. the *unlabeled error rate*. If we can use Theorem 5.4 to impose an unlabeled error rate on our semi-supervised algorithm, then the unlabeled error rate, t, of the target function in Theorem 5.2 can be defined, and thus we can bound the generalization error in terms of the number of unlabeled examples m_u and the number of labeled examples m_l. Note that we are considering this in the context of binary classification for simplicity.

Theorem 5.5. *If $c^* \in C$ and we define $err_{unl}(c*)$ to be $1 - \chi(f, S)$, where $\chi(f, S) = \boldsymbol{E}_{x\sim S}[\chi(f, x_i, x_j)]$ and S represents pairs of samples defined by a graph constructed using Stochastic Discrimination with expected error $< t$, then $err_{unl}(c*) \leq 2t - 2t^2$ and m_u unlabeled examples and m_l labeled examples are sufficient to learn to error ϵ with probability $1 - \delta$, for*

$$m_u = \frac{2}{\epsilon^2}\left[ln|C| + ln\frac{4}{\delta}\right] \text{ and } m_l = \frac{1}{\epsilon}\left[ln|C_{D,\chi}(2t - 2t^2 + 2\epsilon)| + ln\frac{2}{\delta}\right].$$

In particular, with probability at least $1 - \delta$, the $f \in C$ that optimizes $\hat{err}_{unl}(f)$ subject to $\hat{err}(f) = 0$ has $err(f) \leq \epsilon$.

Proof. Recall that we defined the *unlabeled error rate*, $err_{unl}(c*)$, over pairs of samples, and that these pairs were selected (joined) according to the SD algorithm with expected error $< t$, which is possible by Theorem 5.4 from [13]. Then, in the binary case, $err_{unl}(c*)$ depends on the number of pairs having only one vertex misclassified, since if both vertices are misclassified, then it does not increase $err_{unl}(c*)$. Therefore, it follows that if the error on the

individual vertices is $< t$, then the error on the pairs is $err_{unl}(c*) \leq (1-t)t + t(1-t) = 2t - 2t^2$. The remainder of the proof follows directly from Theorom 5.2 from [1]. □

6 EXPERIMENTAL RESULTS

The Brain-Computer Interface (BCI) problem described in [6] is a particularly challenging problem for semi-supervised learning algorithms. It represents the type of noisy, high-dimensional problem that modern machine learning is being asked to solve more and more frequently. The task is to discriminate between electroencephalography (EEG) recordings in which a human subject was concentrating on moving their right or left hand. The supervised baseline obtained using an SVM in [6] had an error rate of 34.31% and an AUC score of 71.17%.

The experiments in Table 1 were conducted using the Laplacian Regularized Least Squares (LapRLS) algorithm described in [4]. The algorithm uses the Laplacian matrix of a graph constructed using ($k = 6$) nearest neighbors. The results use the 12 data splits from the benchmark set in [6], where the LapRLS performed the best out of all methods in the benchmark. We see that our approach can improve even these results. The SD-LapRLS was built with $\beta = 0.05$, $\lambda = 5$, and using 1000 weak classifiers. The difference between the standard and SD algorithms lies only in the graph construction method, where the second result with the standard LapRLS uses cosine similarity to compute neighbors.

7 DISCUSSION

In this paper, we have begun to explore methods for constructing overall error bounds on semi-supervised manifold learning methods. Graph construction methods that utilize the labeled data while respecting the manifold assumption have the potential to make such methods much more broadly applicable, especially in noisy or complex data domains. The number of variations possible in the implementation of the Stochastic Discrimination algorithm provide flexibility to choose a method that is catered to graph construction for manifold representation, but there are still many areas to explore in optimizing such a graph construction method. Ideally, more theoretical analysis can be developed to help guide such implementation choices.

8 ACKNOWLEDGEMENTS

Research sponsored by the Laboratory Directed Research and Development Program of Oak Ridge National Laboratory, managed by UT-Battelle, LLC, for the U. S. Department of Energy. This manuscript has been authored by a contractor of the U.S. Government under Contract No. DE-AC05-00OR22725. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

REFERENCES

[1] Maria-Florina Balcan. 2008. *New Theoretical Frameworks for Machine Learning.* PhD Thesis.
[2] Maria-Florina Balcan and Avrim Blum. 2006. *An Augmented PAC Model for Semi-Supervised Learning.* MIT Press, Cambridge, MA.
[3] Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15, 6 (2003), 1373–1396.
[4] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. *Journal of Machine Learning Research* 7 (2006), 2399–2434.
[5] L. Berton and A. d A. Lopes. 2014. Graph Construction Based on Labeled Instances for Semi-supervised Learning. In *2014 22nd International Conference on Pattern Recognition*. 2477–2482. https://doi.org/10.1109/ICPR.2014.428
[6] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2006. *Semi-Supervised Learning.* MIT Press, Cambridge, MA.
[7] D. L. Donoho. 2000. High-Dimensional Data Analysis: The Curses and Blessings of Dimensionality. Aide-Memoire of a Lecture at AMS Conference on Math Challenges of the 21st Century. (2000).
[8] Andrew Goldberg, Xiaojin Zhu, Aarti Singh, Zhiting Xu, and Robert Nowak. 2009. Multi-Manifold Semi-Supervised Learning. In *12th International Conference on Artificial Intelligence and Statistics (AISTATS).*
[9] Andrew B. Goldberg, Xiaojin Zhu, and Stephen Wright. 2007. Dissimilarity in Graph-Based Semi-Supervised Classification. In *the Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS).*
[10] Yair Goldberg, Alon Zakai, Dan Kushnir, and Ya'acov Ritov. 2008. Manifold Learning: The Price of Normalization. *Journal of Machine Learning Research* 9 (2008), 1909–1939.
[11] A. Irle and J. Kauschke. 2011. On Kleinberg's Stochastic Discrimination Procedure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33, 7 (2011), 1482–1486.
[12] E. M. Kleinberg. 1990. Stochastic Discrimination. *Annals of Mathematics and Artificial Intelligence* 1 (1990), 207–239.
[13] E. M. Kleinberg. 2000. A Mathematically Rigorous Foundation for Supervised Learning. *Lecture Notes in Computer Science* 1857 (2000).
[14] E. M. Kleinberg. 2000. On the Algorithmic Implementation of Stochastic Discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 5 (2000), 473–490.
[15] Tong Lin and Hongbin Zha. 2008. Riemannian Manifold Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30, 5 (2008), 796–809.
[16] Boaz Nadler, Stephane Lafon, and Ronald R. Coifman. 2005. Diffusion Maps, Spectral Clustering and Eigenfunctions of Fokker-Planck Operators. In *Advances in Neural Information Processing (NIPS).*
[17] Sam T. Roweis and Lawrence K. Saul. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science* 290 (2000).
[18] Robert E. Schapire and Yoav Freund. 2012. *Boosting: Foundations and Algorithms.* MIT Press.
[19] Robert E. Schapire, Yoav Freund, Peter Barlett, and Wee Sun Lee. 1997. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML '97).* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 322–330. http://dl.acm.org/citation.cfm?id=645526.657129
[20] Christopher T. Symons, Ranga R. Vatsavai, Goo Jun, and Itamar Arel. 2012. Bias Selection Using Task-Targeted Random Subspaces for Robust Application of Graph-Based Semi-Supervised Learning. In *11th International Conference on Machine Learning and Applications.*
[21] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. 2000. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science* 290 (2000).
[22] K. Q. Weinberger and Lawrence K. Saul. 2006. Unsupervised learning of image manifolds by semidefinite programming. *International Journal of Computer Vision* 70, 1 (2006), 77–90.
[23] Jun Zhu, Eric P. Xing, and Bo Zhang. 2008. Laplace Maximum Margin Markov Networks. In *Proceedings of the 25th International Conference on Machine Learning (ICML).*
[24] L. Zhuang, Z. Zhou, S. Gao, J. Yin, Z. Lin, and Y. Ma. 2017. Label Information Guided Graph Construction for Semi-Supervised Learning. *IEEE Transactions on Image Processing* 26, 9 (Sept. 2017), 4182–4192. https://doi.org/10.1109/TIP.2017.2703120

Table 1: Average error of Laplacian RLS classifiers on BCI data.

Model Building Condtions	Average Error	AUC
Standard LapRLS*	0.3136	0.7483
Standard LapRLS	0.3244	0.7431
SD-LapRLS	**0.2750**	**0.7894**

*LapRLS results in [6], obtained using model selection, a normalized graph Laplacian, and an RBF base kernel; which was the best result among all 11 semi-supervised algorithms tested in the benchmark.