A Comparison of Graph Construction and Learning Algorithms for Graph-Based Phonetic Classification

Yuzong Liu and Katrin Kirchhoff
{yzliu,katrin}@ee.washington.edu

Speech, Signal and Language Interpretation Lab
Department of Electrical Engineering, University of Washington
Seattle WA, 98195-2500

UWEE Technical Report
Number UWEETR-2012-0000
February 2012

Department of Electrical Engineering
University of Washington
Box 352500
Seattle, Washington 98195-2500
PHN: (206) 543-2150
FAX: (206) 543-3842
URL: http://www.ee.washington.edu

A Comparison of Graph Construction and Learning Algorithms for Graph-Based Phonetic Classification

Yuzong Liu and Katrin Kirchhoff
{yzliu,katrin}@ee.washington.edu

Speech, Signal and Language Interpretation Lab
Department of Electrical Engineering, University of Washington
Seattle WA, 98195-2500

University of Washington, Dept. of EE, UWEETR-2012-0000
February 2012

Abstract

Graph-based semi-supervised learning (SSL) algorithms have been widely applied in large-scale machine learning. In this work, we show different graph-based SSL methods (modified adsorption, measure propagation, and prior-based measure propagation) and compare them to the standard label propagation algorithm on a phonetic classification task. In addition, we compare 4 different ways of constructing the phonetic graph: graph construction based on acoustic features vs. first-pass classifier outputs, in combination with either standard k-nearest neighbor search, or mutual k-nearest neighbor search. The best results are obtained with first-pass classifier outputs, and prior-based measure propagation.

1 Introduction

Current research on acoustic modeling in speech processing focuses mainly on the development of improved classifiers (e.g. deep models [11, 21], large-margin classifiers [14, 6], nearest-neighbor classifiers [9], etc.) and alternative training criteria [13, 23, 7]. Another line of research that has recently been shown to be successful in speech processing is graph-based learning (GBL) [26, 25]. Graph-based SSL is a transductive, discriminative learning procedure that differs from all other approaches to acoustic-phonetic classification currently in use in that it explicitly tries to maximize the consistency of the classification output over both the labeled and the unlabeled data. Rather than exploiting similarities between training and test samples only, GBL also takes into consideration the similarities between different unlabeled samples in order to arrive at the final classification output. In this sense, GBL differs fundamentally from classification approaches that rely exclusively on models of the training set (supplemented by adaptation algorithms to better fit the test data) and which ignore classification decisions for other test samples. Several graph-based learning procedures have been proposed for acoustic classification tasks [26, 17, 1]; however, they have not been compared on the same data set and classification task, with identical experimental parameters. The present study has two goals: (a) we compare several state-of-the-art graph-based learning procedures (label propagation, modified adsorption [20], and measure propagation [17]) on the same task under identical experimental conditions in order to determine the best procedure; (b) we propose a novel extension to measure propagation by introducing a prior-based regularization term, where the prior information is derived from a first-pass classifiers. Finally we also compare two different graph construction algorithms, standard k-nearest neighbor vs. mutual k-nearest neighbors. We show that among the graph-based learning procedures, measure propagation and modified adsorption perform comparably while both individually outperform the older label propagation algorithm by [26]. Our new prior-regularized measure propagation method achieves the best results.

2 Graph-Based Learning Algorithms

In graph-based learning algorithms, the training and test data are jointly represented as a graph $G = (V, E, W)$, where V is a set of vertices, each of which represents a data sample, $E = V \times V$ is a set of edges, and W is a similarity measure between these two samples. An edge e_{ij} connects vertices v_i, v_j and is labeled with a similarity value w_{ij}. The graph is constructed over labeled and unlabeled data jointly. GBL was developed as a semi-supervised learning method where a subset of the data points $(1, ..., l)$ is labeled and the rest is unlabeled. Various learning algorithms can then be applied to the graph to infer labeles for the unlabeled points.

The essential assumption of graph-based learning is that the data is characterized by an underlying a low-dimensional manifold, i.e. data points close to each other on the manifold tend to have the same label. Such graph-based SSL algorithms [5], [4], [19], [24], [27] can be usually formulated as minimizing an quadratic function based on the graph that is regularized by both the consistency with training labels, and the smoothness over the manifold as represented by the graph. We next present three different graph-based learning algorithms in detail: label propagation, which has become a de facto baseline in graph-based learning research, modified adsorption, and measure propagation.

2.1 Label Propagation

We follow the conventional notation by defining a data set $\mathcal{X} = \{x_1, \cdots, x_l, x_{l+1}, \cdots, x_n\} \in \mathbf{R}^d$ and a set of labels $\mathcal{Y} = (Y_L, Y_U)$, where $Y_L = \{y_1, y_2, \cdots, y_l\}$ is known. Each y_i is a m-dimensional vector where each entry k indicates the probability of y_i belonging to class k. Here l is the number of labeled data points, u is the number of unlabeled data points, and $n = l + u$ is the total number of data points. Y_L contains labels that correspond to the labeled portion of the data. The goal is to infer $\hat{\mathcal{Y}} = (Y_L, \hat{Y}_U)$, which contains labels for the unlabeled portion. Label propagation (LP) iteratively propagates the information from the labeled data on a graph according to Algorithm 1.

x

Algorithm 1 Label Propagation Algorithm

1. Construct a similarity matrix $\mathbf{W}$ (self-weight $\leftarrow 0$)
2. Compute the transition matrix $\mathbf{T}$: $T_{ij} = \frac{w_{ij}}{\sum_{k=1}^{n} w_{kj}}$, and initialize $\hat{Y} = Y_L$
3. Label propagation by $\hat{Y} \leftarrow T\hat{Y}$
4. Normalization for rows in $\hat{Y}$
5. Fix the labeled portion in $\hat{Y}$ and repeat from step 3 until convergence

Label propagation minimizes the following function:

$$\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}||\hat{y}_i - \hat{y}_j||^2 \text{subject to} \hat{y}_i = y_i, \forall i = 1, \cdots, l \tag{1}$$

As one of the first graph-based SSL algorithms label propagation has had a significant impact on subsequent work and is still used as a de facto baseline in many works on graph-based learning. It enjoys many properties such as a closed-form solution and a natural extension to multi-class classification. However it has several limitations:

1. The labeled portion in $\hat{Y}$ has to be fixed at every iteration. However it is highly possible that there are incorrectly labeled data in the training set. It would be preferable to allow initial labels to be changed if the evidence is strong enough.

2. It does not utilize complete probability distributions on labeled vertices .

2.2 Modified Adsorption

Modified adsorption [20] is derived from the adsorption algorithm, which has been used e.g. in video suggestion for YouTube [3]. Unlike traditional random walk approaches [19], modified adsorption represents a controlled random walk in which each vertex v has three alternative actions: injection, continuation, and abandon, with probabilities $p_v^{inj}, p_v^{conj}, p_v^{abdn}$, respectively. Injection refers to the termination of a random walk and uses prior knowledge about the vertex (labeled points in the training data); continuation refers to the normal continuation of the random walk according to the transition matrix of the graph, and abandon refers to abandoning a random walk and defaulting to a

dummy label. Note that instead of a dummy label, a different label can be used instead, such as that predicted by a different or first-pass classifier (this is the approach used in our experiments described below). Thus, this procedure is able to incorporate prior information.

Define a label matrix $Y \in \mathbf{R}^{n\times(m+1)}$, where Y_{ij} stands for the j-th label entry of vertex i, which is given at the training stage. Note that this label matrix is augmented to have $m+1$ entries for each data point. The very last entry corresponds to the 'dummy' label. Also define a predicted label matrix $\hat{Y} \in \mathbf{R}^{n\times(m+1)}$, and a default matrix $R \in \mathbf{R}^{n\times(m+1)}$, where each $R_i = \begin{bmatrix} \mathbf{0} \in \mathbf{R}^m \\ 1 \end{bmatrix}$ is a $(m+1)$-dimensional vector. The modified adsorption algorithm constructs an objective function based on the following three aspects: 1) the output is close to the given prior knowledge; 2) the manifold is smooth; 3) the output is a dummy proxy if the first two terms are not favored. The objective function to be minimized is the following:

$$\sum_i [\mu_1 \sum_k p_i^{inj}(Y_{ik} - \hat{Y}_{ik})^2 + \mu_2 \sum_{j\in\mathcal{N}(i)} \sum_k p_i^{cont} w_{ij}(\hat{Y}_{ik} - \hat{Y}_{jk})^2 + \mu_3 \sum_k p_i^{abdn}(\hat{Y}_{ik} - R_{ik})^2] \tag{2}$$

where i and k range over all nodes. Equation 2 can be further modified to formulate a convex optimization problem that can readily be solved by Jacobi iteration. The values of $p_v^{inj}, p_v^{conj}, p_v^{abdn}$ can be computed by tuning a hyperparameter β, according to [20]. The hyperparameter β in modified adsorption discounts the high-degree vertices, which are often unreliable.

2.3 Measure Propagation

Measure propagation [17] is a novel graph-based SSL algorithm based on minimizing a Kullback-Leibler divergence (KLD) based loss. Instead of minimizing a quadratic objective function, measure propagation uses a probability measure in the objective function. For completeness, let $\mathcal{X} = \{x_1, \cdots, x_l, x_{l+1}, \cdots, x_n\} \in \mathbf{R}^d$, where l is the number of labeled data, u is the number of unlabeled data, and $n = l + u$ is the total number of data. Define two sets of probability measures $r_i \in \mathbf{R}^m, i = 1, \cdots, l$, and $p_i \in \mathbf{R}^m, i = 1, \cdots, n$, where r_i is the given probability distribution of vertex i for the training portion of the data, and p_i is the predicted probability distribution for vertex i. Both r_i and p_i live on the m-dimensional probability simplex. In the general case, since the labels of training data are already given, r_i is a 'one-hot' vector with one single entry in the given class position. The optimization criterion is formulated as follows:

$$\text{minimize} \qquad C(p) = \sum_{i=1}^{l} D_{KL}(r_i||p_i) + \mu \sum_{i=1}^{n} \sum_{j\in\mathcal{N}(i)} w_{ij} D_{KL}(p_i||p_j) - \nu \sum_{i=1}^{n} H(p_i) \tag{3}$$

This objective function also consists of three criteria, similar to the modified adsorption algorithm: 1) the first term ensures that the predicted probability distribution matches the given distribution on labeled vertices as closely as possible; 2) the second term stands for the smoothness on the manifold (i.e. close points should have a smaller KL divergence); and 3) the third term encourages high-entropy output (if the first two terms are not preferred, the output is then expected to have a uniform distribution). Although $C(p)$ is convex with respect to p_i, there does not exist a closed-form solution. Interior-point methods or methods of multipliers can be used to solve the problem but the computation is expensive.

In [17], alternating minimization (AM) has been proposed to solve the optimization problem. To see a detailed procedure to derive the update equation and proof of convergence for AM, the reader is to referred to a technical report [16].

2.4 Prior-Based Measure Propagation

Measure propagation has many advantages: it enables parallelization and it can handle large-scale dataset. Measure propagation has shown promising results on large-scale datasets such as WebKB [15] and Switchboard [18]. Another notable attribute in measure propagation is its ability to incorporate prior knowledge. In Equation 3, we see the last term encourages the entropy of the probability distribution associated with a vertex to approach the maximum. Note that the entropy of $p \in \mathbf{R}^m$ can be written as

$$H(p) = -\sum_y p(y) \log p(y) = \log m - D_{KL}(p||u) \tag{4}$$

where u is a uniform distribution. By substituting the uniform distribution u with a prior distribution $\tilde{p}$, we instead encourage the model to produce output distributions close to the prior distributions. The objective function of "prior-based" measure propagation can be written as:

$$\text{minimize} \qquad C'(p) = \sum_{i=1}^{l} D_{KL}(r_i||p_i) + \mu \sum_{i=1}^{n} \sum_{j \in \mathcal{N}(i)} w_{ij} D_{KL}(p_i||p_j) + \nu \sum_{i=1}^{n} D_{KL}(p_i||\tilde{p}_i) \tag{5}$$

where r_i, p_i and $\tilde{p}_i$ live on the m-dimensional probability simplex. r_i is the given probability distribution of vertex i for the training portion of the data, p_i is the predicted probability distribution of vertex i, $\tilde{p}_i$ is the prior distribution for vertex i. Again, this convex optimization problem can be solved using alternating minimization.

3 Graph Construction for Acoustic Phone Classification

Graph construction is particularly important in graph-based learning. Since the graph is represented as $G = \{V, E, W\}$, it is critical to define a good similarity measure. In automatic speech recognition, it is conventional to represent speech data as a feature vector x of Mel Frequency Cepstral Coefficients (MFCCs) or Perceptual Linear Predictive (PLP) features. Euclidean, Mahalanobis or cosine distance is then used as a distance measure between feature vectors x_i, x_j, i.e.

$$d_e(x_i, x_j) = ||x_i - x_j||_2 \tag{6}$$

$$d_m(x_i, x_j) = \sqrt{(x_i - x_j)^T \Sigma^{-1} (x_i - x_j)} \tag{7}$$

where $\Sigma \succeq 0$ is a positive definite matrix. The resulting distance values are passed through an RBF kernel

$$w_{ij} = \exp(-\gamma d(x_i, x_j)^2) \tag{8}$$

or a Gaussian kernel to obtain similarity values.

An alternative to this conventional way of building a similarity graph using MFCC features, is to use the probability distribution output from a first-pass classifier. The output from the first-pass classifier is considered a new feature space, possibly allowing different types of similarity measures. In [2] the output from a multi-layer perceptron (MLP) was used. The MLP was trained on a small portion of the original training data, and the predicted probability distributions were used as new features. The advantage of employing a first-pass classifier is that it removes noise in the original acoustic feature space. In order to compute similarity in the new transformed feature space, Kullback-Leibler divergence (KL divergence) can be naturally applied here. Given two probability distrubution p, q, where p, q live on a m-dimensional probability simplex, the similarity between the two probability distribution can be expressed as the Jenson-Shannon divergence:

$$d_{JS}(p, q) = \frac{d_{KL}(p, m) + d_{KL}(q, m)}{2} \tag{9}$$

where $d_{KL}(p, q) = \sum_i p_i \log \frac{p_i}{q_i}$ is the KL-divergence between two probability distribution, and m is the equal-weight interpolation of a and b, i.e. $m_{[i]} = (p_{[i]} + q_{[i]})/2$. The resulting distance is converted to similarity by using a Gaussian kernel. Another advantage is that it allows us to condense the training data. Rather than using the predicted probability distributions for all training data points, we can establish one training vector for each class, by assigning it the optimal 'one-hot' probability vector (i.e. each class seed is a 48-dimensional vector, with only one entry 1 at the position of its class).

The resulting similarity matrix W of the graph is a dense matrix, and for large datasets it is reasonable to prune it to a more sparse matrix. The most common approach is to convert the dense matrix W to a sparse matrix via k-NN selection. Recent study [12] shows that k-NN graphs have hubs (i.e. vertices with high degree), which typically provide little information. In order to reduce to the effects of high degree hubs, mutual k-NN graphs can be used instead. In our work, for each unlabeled vertex, we choose k nearest neighbors in the labeled portion on the graph, and another k (mutual) nearest neighbors in the unlabeled portions. In Jenson-Shannon divergence, we increase the largest weights between an unlabeled vertex and seed if it is below a certain threshold and augment weights from other seeds to it proportionally. In order to avoid the scenario that there exists isolated cliques on a graph, we use a maximum spanning tree and add the minimum number of edges to the graph, as in [12].

4 Experiment Setup

We use the TIMIT corpus for our experiments. For training we use the standard training set without the *sa* sentences (3686 sentences), the core test set of 192 sentences, and a development set of 210 sentences. These correspond to 1044671, 57908, and 63679 frames, respectively. We use a set of 48 phone classes for training (collapsed down from the original TIMIT phone set according to the mapping described in [10]). For evaluation, the 48 classes are further collapsed into 39 classes according to [10].

Our task is *frame-level phone classification*, i.e. results are measured at the frame level and no knowledge of time boundaries is used in our experiments. Feature vectors were extracted from the acoustic data every 10ms, with a 25ms Hamming window and a pre-emphasis of $\alpha = 0.97$. The feature vectors consist of 39 MFCC coefficients (i.e. 12 MFCC features plus energy, plus 13 delta features and 13 delta-delta features). Each frame is further concatenated with the preceding 4 frames and the subsequent 4 frames to form a 351-dimensional feature vector. Speaker-based mean and variance normalization of the feature vectors was performed.

We compare two different ways of constructing the data graph. The first, based on solely on the acoustic feature representation, computes the Euclidean distance between MFCC vectors. The second method investigates the effect of employing a first-pass classifier and using the resulting probability feature space as a representation. To this end we use a multi-layer perceptron, with an input layer of 351 nodes, a hidden layer of 2000 nodes, and an output layer of 48 nodes. The output function at the output layer is the softmax function; training is accomplished by back-propagation. The performance of the MLP is validated on a held-out portion of 10% of the training data and training is stopped when the frame-level error rate on the held-out data starts to increase. We then use the Jensen-Shannon divergence between the 48-dimensional output probability vectors as a distance measures.

The respective distance measures are used by a kdtree procedure to find the k nearest neighbors, both from the training set and from the test set, for any given data point in the test set. After the nearest neighbors have been identified, we use an RBF kernel to convert the distances to similarities. We tune the hyperparameter of the RBF kernel according to the procedure detailed in [8]. The graph is then constructed using the nearest neighbors identified by the kdtree algorithm and the similarities as edge weights.

In a contrastive experiment, we apply the mutual k-NN method with maximum spanning tree [12] for graph construction ($k = 10$), which has been shown to often lead to superior graphs.

In order to evaluate the effects of the number of labeled data points we randomly choose $10\%, 30\%, 50\%$ training data, as well as utilizing the whole training data to generate different graphs.

For the modified adsorption algorithm, we perform a grid search over the following parameters: $\mu_1 = 1, \mu_2 = \{1, 10, 100, 1000\}, \mu_3 = \{0.001, 0.01, 0.1, 1\}, \beta = \{1, 2, 5, 10, 50\}$; for measure propagation we choose the parameters μ, ν from following set: {1e-8,1e-6,1e-4,1e-3,1e-2,1e-1,1,10,100}; for prior-based measure propagation, we choose the parameters μ, ν from the following set: {1e-6,1e-4,1e-3,1e-2,1e-1,1}. The search is based on the performance on the development set.

5 Results

We show the results of phone classification on the core test set of TIMIT in Tables 1, and 2. We compare four graph-based SSL algorithms (label propagation (LP), modified adsorption (MAD), measure propagation (MP) and prior-based measure propagation (pMP)), two feature representations, (MFCC features vs. multilayer perceptron output), and two graph construction techniques (kNN graph (kNN), and mutual kNN graph with maximum spanning tree (mkNN+MST)). Note that MAD and pMP are not applicable to graphs constructed from the MFCC features directly, as no first-pass classifier is involved in these cases. The baseline system is the supervised MLP classifier. In the tables below, Accuracies that are significantly better than the baseline at a level of $p = 0.05$ or lower are shown in **boldface**.

For the graphs built on MFCC features, we see a significant reduction in accuracy compared to the supervised baseline, indicating that graph-based classification from the raw MFCC features lacks robustness. Whereas LP performs slightly worse under the mkNN+mst graph construction scheme, MP benefits from symmetrized graphs. For the graphs built on MLP features, we observe markedly higher accuracies overall, indicating that the strategy of using a first-pass classifier to produce more stable representations is successful. We see significant improvements from MAD and pMP. Both of these algorithms utilize prior information: MAD backs off to the label predicted by the first-pass MLP whenever the dummy label is the highest-scoring label in the MAD output. pMP uses the distribution over phones predicted by the first-pass classifier in the regularization term. It appears that those methods work best that are

	Amount of training data			
System	10%	30%	50%	100%
MLP	65.94	69.24	70.84	72.45
LP - kNN	53.97	57.02	57.84	59.15
MP - kNN	54.26	57.25	58.08	59.51
LP - mKNN	53.16	56.65	57.55	59.05
MP - mkNN	54.56	57.91	58.81	60.28

Table 1: Accuracy rates for frame-based phone classification for the baseline (MLP) and various graph-based learners, based on MFCC feature representation using the simple kNN vs. the mutual kNN (mkNN) graph.

	Amount of training data			
System	10%	30%	50%	100%
MLP	65.94	69.24	70.84	72.45
LP - kNN	65.47	69.24	70.44	71.46
MP - kNN	65.48	69.24	70.44	71.46
MAD - kNN	**66.53**	**70.25**	71.60	**73.01**
pMP - kNN	**67.22**	**71.06**	**72.46**	**73.75**
LP - mkNN	65.47	69.24	70.44	71.46
MP - mkNN	65.47	69.24	70.44	71.46
MAD - mkNN	65.93	69.57	71.08	72.47
pMP - mkNN	**67.23**	**71.06**	**72.43**	**73.76**

Table 2: Accuracy rates for frame-based phone classification for the baseline (MLP) and various graph-based learners, based on the NN output representation using kNN vs. mkNN graph construction.

able to utilize prior information. The mkNN+mst graph construction procedure does not seem to be very useful here, degrading MAD slightly and producing almost no change for the other classifiers.

Another way of analyzing how this prior information helps is to compare the weights of regularization constants μ and ν (in log-10 representation) of MP and pMP in Table 3. The table indicates that we penalize less on the third term (in ν) in Equation 3 and Equation 5, since we encourage vertices to adopt probability distributions close to the prior distribution.

6 Discussion

In a graph-based learning framework, a suitable graph representation is particular important. Given a 'good' graph, different inference algorithms often do not lead to dramatically different results. Graph construction requires us to define a good similarity measure As have shown, the use of first-pass classifier outputs leads to more robust representations and enables us to use similarity measures in probability space, resulting in better graphs. Instead of using a MLP as a first-pass classifier, we have many alternative options such as deep belief nets [22] that are more powerful.

Frame-level phone classification is only the first step towards ASR-style acoustic modeling. Our future goal is to integrate graph-based learning into fully-fledged ASR. To this end will develop similarity measures for variable-length phone segments, incremental graph construction and inference, and better methods for scalability to large data sets.

7 Conclusion

In this work, we have investigated recent graph-based SSL algorithms to phonetic classification. We have also studied the graph construction procedures as well as different representation for acoustic features. The result shows that modified adsorption and measure propagation outperform the baseline label propagation, and that using probability distribution representation in graph-based learning framework improves the result significantly. We conclude that by incorporating the output from a weakly trained first-pass classifier as priori-information, priori-based measure

graph	kNN		mkNN+MST	
method	MP	pMP	MP	pMP
10p	(-1,2)	(-6,-6)	(0,0)	(-6,-6)
30p	(1,2)	(-6,-6)	(0,0)	(-6,-6)
50p	(0,2)	(-6,-6)	(0,0)	(-6,-6)
full	(1,2)	(-6,-6)	(2,2)	(-6,-6)

Table 3: Weights of the Regularization Constants

propagation significantly outperforms the other three algorithms under different setups. In the future, we will move on to the segment-level for phone classification and recognition. We will further incorporate the power of stochastic modeling into the graph-based learning framework.

8 Acknowledgments

This work is supported by NSF grant IIS-0812435.

References

[1] A. Alexandrescu and K. Kirchhoff. Graph-based learning for phonetic classification. In *Proceedings of ASRU*, 2007.

[2] Andrei Alexandrescu and Katrin Kirchhoff. Data-driven graph construction for semi-supervised graph-based learning in NLP. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 204–211, Rochester, New York, April 2007. Association for Computational Linguistics.

[3] Shumeet Baluja, Rohan Seth, D. Sivakumar, Yushi Jing, Jay Yagnik, Shankar Kumar, Deepak Ravichandran, and Mohamed Aly. Video suggestion and discovery for youtube: taking random walks through the view graph. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 895–904, New York, NY, USA, 2008. ACM.

[4] Mikhail Belkin and Partha Niyogi. Semi-supervised learning on riemannian manifolds. *Machine Learning*, 56(1-3):209–239, 2004.

[5] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. In Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, pages 193–216. MIT Press, 2006.

[6] G. Heigold, T. Deselaers, R. Schlüter, and H. Ney. Modified MMI/MPE: a direct evaluation of the margin in speech recognition. In *Proceedings of ICML*, pages 384–391, 2008.

[7] Joseph Keshet, Chih-Chieh Cheng, Mark Stoehr, and David McAllester. Direct error rate minimization of hidden markov models. In *Proceedings of Interspeech*, 2011.

[8] Katrin Kirchhoff and Andrei Alexandrescu. Phonetic classification using controlled random walks. In *INTERSPEECH*, pages 2389–2392, 2011.

[9] J. Labiak and K. Livescu. Nearest neighbor classifiers with learned distances for phonetic frame classification. In *Proceedings of Interspeech*, 2011.

[10] K.F. Lee and H.W. Hon. Speaker-independent phone recognition using Hidden Markov Models. *IEEE Trans. ASSP*, 37:1641–1648, 1989.

[11] A. Mohamed, G. Dahl, and G. Hinton. Deep belief networks for phone recognition. In *Proceedings of NIPS*, 2009.

[12] Kohei Ozaki, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. Using the mutual k-nearest neighbor graphs for semi-supervised classification of natural language data. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, CoNLL '11, pages 154–162, 2011.

[13] D. Povey and P.C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *Proceedings of ICASSP*, 2002.

[14] F. Sha and L. Saul. Large margin Gaussian mixture modeling for phonetic classification and recognition. In *Proceedings of ICASSP*, pages 265–268, 2006.

[15] Amar Subramanya and Jeff Bilmes. Soft-supervised learning for text classification. In *Empirical Methods in Natural Language Processing (EMNLP)*, Honolulu, Hawaii, October 2008.

[16] Amar Subramanya and Jeff Bilmes. Semi-supervised learning with measure propagation. Technical Report UWEETR-2010-0004, University of Washington, Seattle, 2010.

[17] Amar Subramanya and Jeff A. Bilmes. Entropic graph regularization in non-parametric semi-supervised classification. In *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2009.

[18] Amar Subramanya and Jeff A. Bilmes. The semi-supervised switchboard transcription project. In *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Brighton, UK, September 2009.

[19] Martin Szummer and Tommi Jaakkola. Partially labeled classification with markov random walks. In T. Dietterich et al., editor, *Advances in Neural Information Processing Systems 14*. MIT Press, 2001.

[20] Partha Pratim Talukdar and Koby Crammer. New regularized algorithms for transductive learning. In *ECML/PKDD (2)*, pages 442–457, 2009.

[21] D. Yu and L. Deng. Deeps-structured hidden conditional random fields for phonetic recognition. In *Proceedings of Interspeech*, 2010.

[22] Dong Yu and Li Deng. Deep learning and its applications to signal and information processing. *IEEE Signal Processing Magazine*, 28(January):145–150, 2011.

[23] J. Zheng and A. Stolcke. Improved discriminative training using phone lattices. In *Proceedings of Eurospeech*, 2005.

[24] Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[25] Xiaojin Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[26] Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

[27] Xiaojin Zhu, Zoubin Ghahramani, and John Lafferty. Semi-supervised learning using gaussian fields and harmonic functions. In *IN ICML*, pages 912–919, 2003.