Graph-Based Semi-supervised Learning for Phone and Segment Classification

Yuzong Liu, Katrin Kirchhoff

Department of Electrical Engineering, University of Washington, Seattle, WA, USA

yzliu@ee.washington.edu, katrin@ee.washington.edu

Abstract

This paper presents several novel contributions to the emerging framework of graph-based semi-supervised learning for speech processing. First, we apply graph-based learning to variable-length segments rather than to the fixed-length vector representations that have been used previously. As part of this work we compare various graph-based learners, and we utilize an efficient feature selection technique for high-dimensional feature spaces that alleviates computational costs and improves the performance of graph-based learners. Finally, we present a method to improve regularization during the learning process. Experimental evaluation on the TIMIT frame and segment classification tasks demonstrates that the graph-based classifiers outperform standard baseline classifiers; furthermore, we find that the best learning algorithms are those that can incorporate prior knowledge.

1. Introduction

In recent years semi-supervised learning (SSL) approaches, which learn jointly from labeled and unlabeled data, have been applied to many different areas in machine learning and pattern recognition. The main motivation for semi-supervised learning is typically the sparseness of labeled training data. In speech processing, training data can be gathered quite easily, although it can be costly and error-prone to annotate large amounts of data, especially at the phonetic level. An SSL framework that has recently been successfully applied to speech processing is semi-supervised graph-based learning (GBL) [1]. This approach jointly models the training and test data as a graph whose nodes represent data samples and whose edges are labeled with non-negative values representing the similarity of samples. A learning algorithm is then run on the graph that infers a label assignment for all data samples jointly, using information from the labeled training points and the smoothness constraints imposed by the graph. The label assignment must respect the inherent clustering properties expressed by the graph: highly similar data points are encouraged to receive the same labels, whereas dissimilar data points are more likely to receive different labels. Such graph-based learners utilize not only similarities between training and test points, but also *similarities between different test points*. The latter is a unique property that is not being exploited by conventional acoustic modeling approaches in speech processing systems. It was shown in [2] that this property may be beneficial not only when labeled training data is sparse, but also when the test set has a different distribution than the training data since the GBL learner can enforce a form of adaptation to the test data.

Since then several GBL algorithms have been tested on the task of frame-based phone classification. In [2], label propagation (LP) was used for vowel classification. In [3] a graph-based algorithm called measure propagation (MP) was developed and applied to the TIMIT database as well as to the Switchboard corpus [4]. In [5], another algorithm, modified adsorption (MAD) was also tested for frame-based classification on the TIMIT database. Most recently, [6] proposed a graph-based learning procedure integrating confidence measures. All of them have reported improvements over supervised learners, and GBL is emerging as a viable paradigm in research on alternative acoustic modeling techniques. However, the various studies used different data sets or different preprocessing methods, making it difficult to compare and evaluate different GBL algorithms. More importantly, all previous experiments have focused on *frame-based classification*; however, speech processing systems need to handle units (phones, words) that are inherently of variable length. Thus, the next step towards integrating GBL into realistic speech processing applications is to devise ways of handling variable-length units, which is the focus of this paper. In addition, our paper makes two other contributions: we present a different regularization method during learning that incorporates prior information, and we compare four different learning algorithms on two different tasks under the same experimental conditions. Our results show that those learners perform best that are able to exploit prior information about the label distribution.

2. Graph-Based Learning Algorithms

We follow the conventional notation by defining two sets in semi-supervised learning. The first set $D_L = \{(\mathbf{x}_i, y_i)\}_{i=1}^{l}$ contains the labeled data, and the second set $D_U = \{\mathbf{x}_i\}_{i=l+1}^{n}$ with $n = l + u$ contains the unlabeled data. Each data point lies in a d-dimensional vector space $\mathbb{R}^d$. The goal of GBL algorithms is to infer the label of D_U via an undirected weighted graph $\mathcal{G} = (V, E, W)$ where V are the data points in D_L and D_U and E are the undirected edges on the graph, weighted by $w_{ij} \in \mathbf{W}$.

2.1. Previous methods

Label propagation (LP) [7], iteratively propagates the information from the labeled data on the graph $\mathcal{G}$. Label propagation minimizes the following function:

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

where $\hat{y}$ is a predicted label and y is the true label. Two more recent GBL algorithms are modified adsorption [8] and measure propagation [3]. MAD represents a controlled random walk. Unlike traditional random walks, each vertex v has three alternative actions, injection, continuation, and abandon, associated with probabilities $p_v^{inj}, p_v^{conj}, p_v^{abdn}$, respectively. Injection refers to the termination of a random walk and use 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. The objective function is

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

where $Y \in \mathbf{R}^{n\times(m+1)}$ is the matrix of known labels; Y_{ij} stands for the j-th label entry of vertex i. Note that this label matrix is augmented to have $m+1$ entries for each data point. The very last entry corresponds to a 'dummy' label. $\hat{Y} \in \mathbf{R}^{n\times(m+1)}$, is the matrix of predicted labels, and $R \in \mathbf{R}^{n\times(m+1)}$, is a default matrix where each $R_i = \begin{bmatrix}\mathbf{0} \in \mathbf{R}^m \\ 1\end{bmatrix}$ is a $(m+1)$-dimensional vector. The 'dummy' label provides a way of incorporating prior knowledge, which allows us to default to e.g. the a priori most frequent label when the learner is uncertain.

Measure propagation minimizes the following objective function:

$$\begin{aligned}&\sum_{i=1}^{l} D_{KL}(r_i||p_i) \qquad (3)\\ &+\mu \sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)} w_{ij}D_{KL}(p_i||p_j) - \nu\sum_{i=1}^{n} H(p_i)\end{aligned}$$

where p is the predicted probability distribution over the classes, r is the true (reference) distributions, N is the graph neighborhood of node i, KL is the Kullback-Leibler divergence, and H is the entropy. Thus, the first term ensures that the predicted probability distribution matches the true distribution on labeled vertices as closely as possible; 2) the second term stands for the smoothness of the label assignment enforced by the graph (i.e. the class probability distributions on neighboring points in the graph should have a smaller KL divergence); and 3) the third term encourages high-entropy output. Minimization of this function can be done by alternating minimization.

2.2. Prior-Regularized Measure Propagation

We propose to extend MP as follows: The entropy of $p \in \mathbf{R}^m$ in Equation 3 can be written as $H(p) = -\sum_y p(y)\log p(y) = \log m - D_{KL}(p||u)$, where u is a uniform distribution. By substituting the uniform distribution u with a prior distribution $\tilde{p}$, we can instead encourage the model to produce output distributions close to the a prior distributions, if that information is available. The objective function of this "prior-based' measure propagation" (pMP) can be written as:

$$\begin{aligned}&\sum_{i=1}^{l} D_{KL}(r_i||p_i) + \mu\sum_{i=1}^{n}\sum_{j\in\mathcal{N}(i)} w_{ij}D_{KL}(p_i||p_j)\\ &+\nu\sum_{i=1}^{n} D_{KL}(p_i||\tilde{p}_i) \qquad (4)\end{aligned}$$

3. GBL with Variable-Length Segments

GBL was developed for fixed-length feature vectors, and all previous applications in speech processing have focused on frame-level classification, where all vectors are of the same length. The next logical steps towards integrating GBL into fully-fledged speech processing applications are the classification and recognition of variable-length segments – in the former task, the time segmentation is given and the label needs to be predicted; in the latter, both time stamps and labels need to be inferred. Here we focus on segment classification, which requires either length normalization of the acoustic input or a similarity measure that is able to handle variable-length input. Similar to [9] we initially compared input length normalization techniques in combination with standard similarity or distance measure (such as cosine similarity and Euclidean distance) against using a variable-length kernel (dynamic time alignment kernels and the Fisher kernel) for the similarity measure $\mathbf{W}$. Initial results indicated that using the Fisher kernel [10] is by far the superior method and we describe it here in detail: The Fisher score for a sample (segment) X is defined as

$$U_X = \nabla_\theta log P(X|\theta) \qquad (5)$$

where θ are the parameters of some trained (usually generative) model, like a HMM or GMM; thus, the Fisher transformation converts X into a fixed-length vector of derivatives that depends on the dimensionality of

θ. When several models are involved their individual Fisher score vectors are stacked to form the complete score space:

$$U'_X = ((U_X^{\theta_1})^\intercal, (U_X^{\theta_2})^\intercal, ..., (U_X^{\theta_n})^\intercal)^\intercal \qquad (6)$$

The Fisher kernel is then defined as

$$K(X_i, X_j) = U'_X I^{-1} U_Y \qquad (7)$$

where I is the covariance matrix of the Fisher score vectors, also called the Fisher information matrix. For computational reasons, I is often omitted, or it is approximated by its diagonal. Depending on the number of models and parameters in the generative models, the Fisher kernel vector can be extremely high-dimensional, and many dimensions may be redundant. We adopt the feature selection framework based on submodular functions as proposed in [11] that can be easily extended to very high-dimensional spaces. Compared to traditional feature selection approaches based on the scores of mutual information between features and target labels, the submodular feature selection yields more diverse features and reduce the redundancy within features.

4. Experiments and Results

Our experiments are performed on the TIMIT database [12]. We use the standard core test set of 192 sentences, the training set without the *sa* sentences (3,686 sentences), and a development set of 210 sentences. For training we use the standard phone set of 48 phones, which is collapsed into 39 phones for evaluation (using the mapping suggested in [12]). Glottal stop segments are excluded. The total number of segments for the training, development and test sets, respectively, are 121385, 7416, and 6589. This corresponds to frame counts of 1044671, 63679, and 57908. We establish four different training conditions using 10%, 30%, 50% and 100% of the training data, respectively, to investigate the performance of our algorithms on varying amounts of training data. The data was preprocessed by extracting 39-dimensional feature vectors (consisting of 12 MFCC coefficients, 1 energy coefficient, deltas, and delta-deltas) every 10ms with a window of 25ms. Speaker-dependent mean and variance normalization was applied.

4.1. Comparison of LP, MP, MAD, and pMP

Our initial goal was to determine which of the four learning algorithms presented above (LP, MP, MAD, and pMP) performs best under identical conditions. To this end we first built GBL systems for frame-based classification. In line with [5] we utilize a first-pass supervised classifier and construct the graph on its output, i.e. phone probability distributions over the 48 phone classes. Our first-pass classifier is a 3-layer MLP that takes as input a window of 9 acoustic frames. 2000 hidden units are used, and the output function is the softmax function. For each training condition the MLP only receives the same training data as the subsequent graph-based learner, i.e. 10%, 30%, 50% or 100%. We then use the outputs from this classifier to build the data graph according to the procedure detailed in [5]. The MLP achieves frame error rates between 65.94% and 72.45%, depending on the amount of training data used (Table 1). This matches results of similar baselines reported in the literature (e.g.,[13]). We are aware that the currently best-performing TIMIT systems, especially so-called deep models (e.g.,[14, 15]) achieve higher accuracy; however, the purpose of this paper is not to beat the currently best models, but to determine best practices and the best learning algorithms within the GBL framework. These can then later be combined with more powerful baseline classifiers. In fact, the information exploited by graph-based learner is orthogonal to information exploited by standard acoustic modeling. The latter utilizes the information from the test data on the manifold structure.

To obtain the weights for the graph edges, we compute the Jensen-Shannon divergence between the predicted phone probability distributions for pairs of samples. Divergence values are then converted to similarity values by a Gaussian kernel. A fast nearest neighbour search procedure (kd-trees) is used to select the training and test data points within a given radius r around each test point. ($r = 7.0$ for selecting training points, $r = 0.02$ for test points). A separate graph is built for each of the four training condition. The graphs have around 7M nodes each. For MP and pMP, since the Jensen-Shannon divergence between a given unlabeled vertex and all the seeds can be very small (i.e. $< 10^{-10}$), we set the largest weight to 0.1 and scale up the other weights accordingly. For the prior distribution in the last term of the pMP objective function (see Eq. 4) we use the predicted probability distributions from the MLP. For MAD, in the case where the highest-scoring prediction is the "dummy" label, we also use the label predicted by the MLP as a "backoff" label. Thus, both MAD and pMP incorporate prior information from the first-pass classifier. The various coefficients in the objective functions for the graph-based learners are optimized on the development set. Test set results are shown in Table 1.

We see that the best GBL procedures are those that incorporate prior information, viz. MAD and pMP. The best results are obtained by pMP; improvements over the supervised baseline range between 1.28% and 1.82% absolute.

4.2. Segment classification

For the segment classification task, we require a stochastic baseline classifier. For this purpose we trained a simple monophone HMM system using the GMTK toolkit [16], again using only as much training data as for the GBL step. We use 16 Gaussian mixture components with

	Amount of training data			
System	10%	30%	50%	100%
MLP	65.94	69.24	70.84	72.45
LP	65.47	69.24	70.44	71.46
MP	65.48	69.24	70.44	71.46
MAD	**66.53**	**70.25**	71.60	**73.01**
pMP	**67.22**	**71.06**	**72.46**	**73.75**

Table 1: Accuracy rates (%) for frame-based phone classification for the baseline (MLP) and various graph-based learners. Bold-face numbers are significant ($p < 0.05$) improvements over the baseline.

	Amount of training data			
System	10%	30%	50%	100%
HMM	66.81	68.09	67.92	68.02
LP	62.01	67.26	68.72	69.43
MP	63.32	67.49	69.01	**70.45**
MAD	67.23	69.42	**69.62**	**70.21**
pMP	**70.45**	**70.71**	**70.89**	**71.15**

Table 2: Accuracy rates (%) for segment classification, baseline system (HMM) and the various graph-based learners. Bold-face numbers are significant ($p < 0.05$) improvements over the baseline.

diagonal covariance matrices per phone class throughout all training conditions. Performance of the HMM saturates at 30% of the training data; for the higher percentage cases, better performance might be achieved by using more mixture components; however this would lead to even higher-dimensional Fisher score spaces, as explained below.

In order to compute the Fisher kernel we take the derivatives of all Gaussian components and stack them into one single vector per segment. The resulting dimensionality of the Fisher kernel vector is 182017, making the pairwise similarity computation too costly. To reduce the huge dimensionality, we first prune the original feature set by eliminating each feature i in the initial feature set F for which its mutual information with the classes, $MI(f_i;C)$, is less than a threshold τ (set to 0.01 in our case). We then use the more computationally expensive submodular feature selection method in [11] on the remaining feature set (73978 features), and reduce the dimensionality to 800 final features. The dot product in Equation 7 is then computed on these vectors. We do not use the Fisher information matrix to avoid computation; a simple normalization is performed by $\tilde{K}_{ij} = \frac{K_{ij}}{\sqrt{K_{ii}K_{jj}}}$. Using this similarity measure, 10-nearest neighbor graphs were constructed for all of the different training conditions. The same learners as before were used; hyperparameters were re-tuned on the development set. We obtain the prior distribution for our pMP method as follows: each phone model of 48 HMMs, the Viterbi score (represented in log-likelihood) per frame is computed. Then the probability is derived by marginalizing the sum of likelihoods. Test set results are shown in Table 2. Again, the pMP method shows the best results. Improvements over the supervised baseline classifier is 2.62% and 3.64% absolute. We did not compare against the full Fisher vectors since the computation would have been too costly; however, on other acoustic processing tasks [11], the submodular feature selection task turned out to work better than using the full feature vectors or less powerful feature selection techniques. The results show that GBL is also effective for variable-length segments, with improvements being slightly larger than for the frame-based case.

5. Discussion and Future Work

In the frame-based experiments the MAD and pMP algorithms outperform both the baseline and the LP and MP algorithms, indicating that those learners work best that can utilize prior information. The best results are those obtained by our proposed pMP algorithm, which outperforms the supervised baseline even in the 100% case. This indicates that GBL may be useful not only in cases where training data is sparse but also as a general way of adapting classifiers to the test data.

There are several implications of this study for speech processing. First, we anticipate that it will be possible to combine GBL with other first-pass classifiers, such as deep neural networks, with similar results since the type of information exploited by GBL (information from the test data) is orthogonal to information exploited by standard supervised classifiers. For example, a deep models could be used instead of an MLP for the experiments in table 1, or to generate the term $\tilde{p}_i$ in Equation 4 for better regularization. Second, there are several points at which GBL could be applied in a speech recognition system: during acoustic modeling for first pass recognition, when rescoring word hypothesis lattices, or when generating phone lattices for adaptation. For example, for the purpose of word lattice rescoring, one could generate a labeled set of word segments through forced alignment of the training transcriptions, and the word hypothesis lattices for the test data would form the unlabeled data set. Scores obtained by a graph based learner run jointly on these two sets could then be re-integrated into the lattices for rescoring. These experiments will be part of our future work. Additional work will be carried out on developing more scalable versions of this framework, especially with respect to efficient graph construction and score space computation.

6. References

[1] X. Zhu, “Semi-supervised learning literature survey,” Computer Sciences, University of Wisconsin-Madison, Tech. Rep. 1530, 2005, http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.

[2] A. Alexandrescu and K. Kirchhoff, “Graph-based learning for phonetic classification,” in *Proceedings of ASRU*, 2007.

[3] A. Subramanya and J. Bilmes, “Entropic graph regularization in non-parametric semi-supervised classification,” in *Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2009.

[4] ——, “The semi-supervised Switchboard transcription project,” in *Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH)*, Brighton, UK, September 2009.

[5] K. Kirchhoff and A. Alexandrescu, “Phonetic classification using controlled random walks,” in *Proceedings of Interspeech*, 2011.

[6] M. Orbach and K. Crammer, “Transductive phoneme classification using local scaling and confidence,” in *Proceedings of IEEE 27th Convention of Electric and Electronics Engineers in Israel*, 2012.

[7] X. Zhu and Z. Ghahramani, “Learning from labeled and unlabeled data with label propagation,” CMU-CALD-02, Tech. Rep., 2002. [Online]. Available: citeseer.ist.psu.edu/article/zhu02learning.html

[8] P. Talukdar and K. Crammer, “New regularized algorithms for transductive learning,” in *Proceedings of ECML-PKDD*, 2009, pp. 442–457.

[9] A. Temko, E. Monte, and C. Nadeu, “Comparison of sequence discriminant support vector machines for acoustic event classification,” in *Proceedings of ICASSP*, 2006.

[10] T. Jaakkola and D. Haussler, “Exploiting generative models in discriminative classifiers,” in *Advances in Neural Information Processing Systems 11*, 1999a, p. 487493.

[11] Y. Liu, K. Wei, K. Kirchhoff, Y. Song, and J. Bilmes, “Submodular feature selection for high-dimensional acoustic score spaces,” in *Proceedings of ICASSP*, 2013.

[12] K. Lee and H. Hon, “Speaker-independent phone recognition using Hidden Markov Models,” *IEEE Trans. ASSP*, vol. 37, pp. 1641–1648, 1989.

[13] J. Labiak and K. Livescu, “Nearest neighbors with learned distances for phonetic frame classification,” in *Proceedings of Interspeech*, 2011.

[14] A. Mohamed, G. Dahl, and G. Hinton, “Deep belief networks for phone recognition,” in *Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications*, 2009.

[15] D. Yu and L. Deng, “Deep hidden conditional random fields for phonetic recognition,” in *Proceedings of Interspeech*, 2010.

[16] J. Bilmes, “Dynamic graphical models,” *IEEE Signal Processing Magazine*, vol. 27(6), p. 2942, 2010.