Word Sense Disambiguation Using Label Propagation Based Semi-Supervised Learning

Zheng-Yu Niu, Dong-Hong Ji
Institute for Infocomm Research
21 Heng Mui Keng Terrace
119613 Singapore
{zniu, dhji}@i2r.a-star.edu.sg

Chew Lim Tan
Department of Computer Science
National University of Singapore
3 Science Drive 2
117543 Singapore
tancl@comp.nus.edu.sg

Abstract

Shortage of manually sense-tagged data is an obstacle to supervised word sense disambiguation methods. In this paper we investigate a label propagation based semi-supervised learning algorithm for WSD, which combines labeled and unlabeled data in learning process to fully realize a global consistency assumption: similar examples should have similar labels. Our experimental results on benchmark corpora indicate that it consistently outperforms SVM when only very few labeled examples are available, and its performance is also better than monolingual bootstrapping, and comparable to bilingual bootstrapping.

1 Introduction

In this paper, we address the problem of word sense disambiguation (WSD), which is to assign an appropriate sense to an occurrence of a word in a given context. Many methods have been proposed to deal with this problem, including supervised learning algorithms (Leacock et al., 1998), semi-supervised learning algorithms (Yarowsky, 1995), and unsupervised learning algorithms (Schütze, 1998).

Supervised sense disambiguation has been very successful, but it requires a lot of manually sense-tagged data and can not utilize raw unannotated data that can be cheaply acquired. Fully unsupervised methods do not need the definition of senses and manually sense-tagged data, but their sense clustering results can not be directly used in many NLP tasks since there is no sense tag for each instance in clusters. Considering both the availability of a large amount of unlabelled data and direct use of word senses, semi-supervised learning methods have received great attention recently.

Semi-supervised methods for WSD are characterized in terms of exploiting unlabeled data in learning procedure with the requirement of predefined sense inventory for target words. They roughly fall into three categories according to what is used for supervision in learning process: (1) using external resources, e.g., thesaurus or lexicons, to disambiguate word senses or automatically generate sense-tagged corpus, (Lesk, 1986; Lin, 1997; McCarthy et al., 2004; Seo et al., 2004; Yarowsky, 1992), (2) exploiting the differences between mapping of words to senses in different languages by the use of bilingual corpora (e.g. parallel corpora or untagged monolingual corpora in two languages) (Brown et al., 1991; Dagan and Itai, 1994; Diab and Resnik, 2002; Li and Li, 2004; Ng et al., 2003), (3) bootstrapping sense-tagged seed examples to overcome the bottleneck of acquisition of large sense-tagged data (Hearst, 1991; Karov and Edelman, 1998; Mihalcea, 2004; Park et al., 2000; Yarowsky, 1995).

As a commonly used semi-supervised learning method for WSD, bootstrapping algorithm works by iteratively classifying unlabeled examples and adding confidently classified examples into labeled dataset using a model learned from augmented labeled dataset in previous iteration. It can be found that the affinity information among unlabeled examples is not fully explored in this bootstrapping process. Bootstrapping is based on a local consistency assumption: examples close to labeled examples within same class will have same labels, which is also the assumption underlying many supervised learning algorithms, such as kNN.

Recently a promising family of semi-supervised learning algorithms are introduced, which can effectively combine unlabeled data with labeled data

in learning process by exploiting cluster structure in data (Belkin and Niyogi, 2002; Blum et al., 2004; Chapelle et al., 1991; Szummer and Jaakkola, 2001; Zhu and Ghahramani, 2002; Zhu et al., 2003). Here we investigate a label propagation based semi-supervised learning algorithm (LP algorithm) (Zhu and Ghahramani, 2002) for WSD, which works by representing labeled and unlabeled examples as vertices in a connected graph, then iteratively propagating label information from any vertex to nearby vertices through weighted edges, finally inferring the labels of unlabeled examples after this propagation process converges.

Compared with bootstrapping, LP algorithm is based on a global consistency assumption. Intuitively, if there is at least one labeled example in each cluster that consists of similar examples, then unlabeled examples will have the same labels as labeled examples in the same cluster by propagating the label information of any example to nearby examples according to their proximity.

This paper is organized as follows. First, we will formulate WSD problem in the context of semi-supervised learning in section 2. Then in section 3 we will describe LP algorithm and discuss the difference between a supervised learning algorithm (SVM), bootstrapping algorithm and LP algorithm. Section 4 will provide experimental results of LP algorithm on widely used benchmark corpora. Finally we will conclude our work and suggest possible improvement in section 5.

2 Problem Setup

Let $X = \{x_i\}_{i=1}^n$ be a set of contexts of occurrences of an ambiguous word w, where x_i represents the context of the i-th occurrence, and n is the total number of this word's occurrences. Let $S = \{s_j\}_{j=1}^c$ denote the sense tag set of w. The first l examples $x_g (1 \leq g \leq l)$ are labeled as y_g ($y_g \in S$) and other u ($l+u = n$) examples $x_h (l+1 \leq h \leq n)$ are unlabeled. The goal is to predict the sense of w in context x_h by the use of label information of x_g and similarity information among examples in X.

The cluster structure in X can be represented as a connected graph, where each vertex corresponds to an example, and the edge between any two examples x_i and x_j is weighted so that the closer the vertices

in some distance measure, the larger the weight associated with this edge. The weights are defined as follows: $W_{ij} = exp(-\frac{d_{ij}^2}{\sigma^2})$ if $i \neq j$ and $W_{ii} = 0$ ($1 \leq i, j \leq n$), where d_{ij} is the distance (ex. Euclidean distance) between x_i and x_j, and σ is used to control the weight W_{ij}.

3 Semi-supervised Learning Method

3.1 Label Propagation Algorithm

In LP algorithm (Zhu and Ghahramani, 2002), label information of any vertex in a graph is propagated to nearby vertices through weighted edges until a global stable stage is achieved. Larger edge weights allow labels to travel through easier. Thus the closer the examples, more likely they have similar labels (the global consistency assumption).

In label propagation process, the soft label of each initial labeled example is clamped in each iteration to replenish label sources from these labeled data. Thus the labeled data act like sources to push out labels through unlabeled data. With this push from labeled examples, the class boundaries will be pushed through edges with large weights and settle in gaps along edges with small weights. If the data structure fits the classification goal, then LP algorithm can use these unlabeled data to help learning classification plane.

Let $Y^0 \in N^{n \times c}$ represent initial soft labels attached to vertices, where $Y_{ij}^0 = 1$ if y_i is s_j and 0 otherwise. Let Y_L^0 be the top l rows of Y^0 and Y_U^0 be the remaining u rows. Y_L^0 is consistent with the labeling in labeled data, and the initialization of Y_U^0 can be arbitrary.

Optimally we expect that the value of W_{ij} across different classes is as small as possible and the value of W_{ij} within same class is as large as possible. This will make label propagation to stay within same class. In later experiments, we set σ as the average distance between labeled examples from different classes.

Define $n \times n$ probability transition matrix $T_{ij} = P(j \rightarrow i) = \frac{W_{ij}}{\sum_{k=1}^n W_{kj}}$, where T_{ij} is the probability to jump from example x_j to example x_i.

Compute the row-normalized matrix $\overline{T}$ by $\overline{T}_{ij} = T_{ij} / \sum_{k=1}^n T_{ik}$. This normalization is to maintain the class probability interpretation of Y.
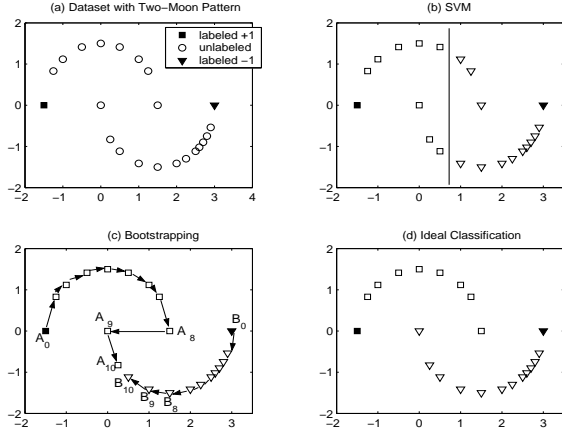
396

Figure 1: Classification result on two-moon pattern dataset. (a) Two-moon pattern dataset with two labeled points, (b) classification result by SVM, (c) labeling procedure of bootstrapping algorithm, (d) ideal classification.

Then LP algorithm is defined as follows:

1. Initially set t=0, where t is iteration index;

2. Propagate the label by $Y^{t+1} = \overline{T}Y^t$;

3. Clamp labeled data by replacing the top l row of Y^{t+1} with Y_L^0. Repeat from step 2 until Y^t converges;

4. Assign $x_h(l + 1 \leq h \leq n)$ with a label $s_{\hat{j}}$, where $\hat{j} = argmax_j Y_{hj}$.

This algorithm has been shown to converge to a unique solution, which is $\widehat{Y}_U = \lim_{t \to \infty} Y_U^t = (I - \overline{T}_{uu})^{-1}\overline{T}_{ul}Y_L^0$ (Zhu and Ghahramani, 2002). We can see that this solution can be obtained without iteration and the initialization of Y_U^0 is not important, since Y_U^0 does not affect the estimation of $\widehat{Y}_U$. I is $u \times u$ identity matrix. $\overline{T}_{uu}$ and $\overline{T}_{ul}$ are acquired by splitting matrix $\overline{T}$ after the l-th row and the l-th column into 4 sub-matrices.

3.2 Comparison between SVM, Bootstrapping and LP

For WSD, SVM is one of the state of the art supervised learning algorithms (Mihalcea et al., 2004), while bootstrapping is one of the state of the art semi-supervised learning algorithms (Li and Li, 2004; Yarowsky, 1995). For comparing LP with SVM and bootstrapping, let us consider a dataset with two-moon pattern shown in Figure 1(a). The upper moon consists of 9 points, while the lower moon consists of 13 points. There is only one labeled point in each moon, and other 20 points are un-
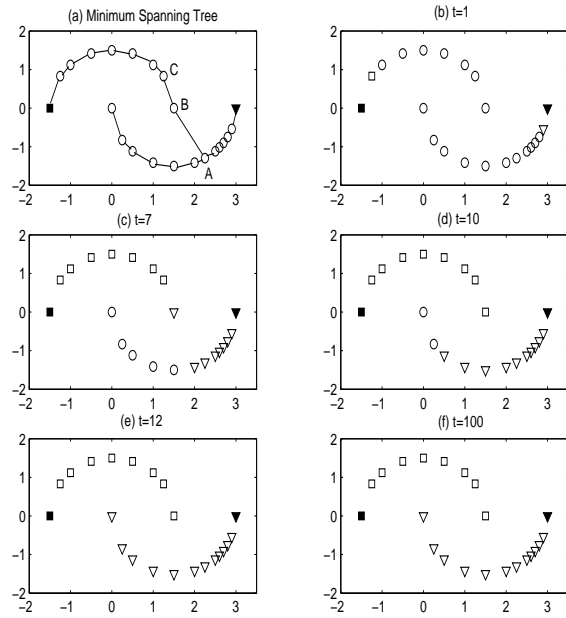

Figure 2: Classification result of LP on two-moon pattern dataset. (a) Minimum spanning tree of this dataset. The convergence process of LP algorithm with t varying from 1 to 100 is shown from (b) to (f).

labeled. The distance metric is Euclidian distance. We can see that the points in one moon should be more similar to each other than the points across the moons.

Figure 1(b) shows the classification result of SVM. Vertical line denotes classification hyperplane, which has the maximum separating margin with respect to the labeled points in two classes. We can see that SVM does not work well when labeled data can not reveal the structure (two moon pattern) in each class. The reason is that the classification hyperplane was learned only from labeled data. In other words, the coherent structure (two-moon pattern) in unlabeled data was not explored when inferring class boundary.

Figure 1(c) shows bootstrapping procedure using kNN (k=1) as base classifier with user-specified parameter $b = 1$ (the number of added examples from unlabeled data into classified data for each class in each iteration). Termination condition is that the distance between labeled and unlabeled points is more than inter-class distance (the distance between A_0 and B_0). Each arrow in Figure 1(c) represents one classification operation in each iteration for each class. After eight iterations, $A_1 \sim A_8$ were tagged

as $+1$, and $B_1 \sim B_8$ were tagged as -1, while $A_9 \sim A_{10}$ and $B_9 \sim B_{10}$ were still untagged. Then at the ninth iteration, A_9 was tagged as $+1$ since the label of A_9 was determined only by labeled points in kNN model: A_9 is closer to any point in $\{A_0 \sim A_8\}$ than to any point in $\{B_0 \sim B_8\}$, regardless of the intrinsic structure in data: $A_9 \sim A_{10}$ and $B_9 \sim B_{10}$ are closer to points in lower moon than to points in upper moon. In other words, bootstrapping method uses the unlabeled data under a local consistency based strategy. This is the reason that two points A_9 and A_{10} are misclassified (shown in Figure 1(c)).

From above analysis we see that both SVM and bootstrapping are based on a local consistency assumption.

Finally we ran LP on a connected graph-minimum spanning tree generated for this dataset, shown in Figure 2(a). A, B, C represent three points, and the edge $A - B$ connects the two moons. Figure 2(b)- 2(f) shows the convergence process of LP with t increasing from 1 to 100. When $t = 1$, label information of labeled data was pushed to only nearby points. After seven iteration steps ($t = 7$), point B in upper moon was misclassified as -1 since it first received label information from point A through the edge connecting two moons. After another three iteration steps (t=10), this misclassified point was re-tagged as $+1$. The reason of this self-correcting behavior is that with the push of label information from nearby points, the value of $Y_{B,+1}$ became higher than $Y_{B,-1}$. In other words, the weight of edge $B - C$ is larger than that of edge $B - A$, which makes it easier for $+1$ label of point C to travel to point B. Finally, when $t \geq 12$ LP converged to a fixed point, which achieved the ideal classification result.

4 Experiments and Results

4.1 Experiment Design

For empirical comparison with SVM and bootstrapping, we evaluated LP on widely used benchmark corpora - "interest", "line" [1] and the data in English lexical sample task of SENSEVAL-3 (including all 57 English words) [2].

---

[1] Available at http://www.d.umn.edu/~tpederse/data.html
[2] Available at http://www.senseval.org/senseval3

Table 1: The upper two tables summarize accuracies (averaged over 20 trials) and paired t-test results of SVM and LP on SENSEVAL-3 corpus with percentage of training set increasing from 1% to 100%. The lower table lists the official result of baseline (using most frequent sense heuristics) and top 3 systems in ELS task of SENSEVAL-3.

Percentage	SVM	LP_{cosine}	LP_{JS}
1%	24.9±2.7%	27.5±1.1%	28.1±1.1%
10%	53.4±1.1%	54.4±1.2%	54.9±1.1%
25%	62.3±0.7%	62.3±0.7%	63.3±0.9%
50%	66.6±0.5%	65.7±0.5%	66.9±0.6%
75%	68.7±0.4%	67.3±0.4%	68.7±0.3%
100%	69.7%	68.4%	70.3%

Percentage	SVM vs. LP_{cosine}		SVM vs. LP_{JS}	
	p-value	Sign.	p-value	Sign.
1%	8.7e-004	$\ll$	8.5e-005	$\ll$
10%	1.9e-006	$\ll$	1.0e-008	$\ll$
25%	9.2e-001	$\sim$	3.0e-006	$\ll$
50%	1.9e-006	$\gg$	6.2e-002	$\sim$
75%	7.4e-013	$\gg$	7.1e-001	$\sim$
100%	-	-	-	-

Systems	Baseline	htsa3	IRST-Kernels	nusels
Accuracy	55.2%	72.9%	72.6%	72.4%

We used three types of features to capture contextual information: part-of-speech of neighboring words with position information, unordered single words in topical context, and local collocations (as same as the feature set used in (Lee and Ng, 2002) except that we did not use syntactic relations). For SVM, we did not perform feature selection on SENSEVAL-3 data since feature selection deteriorates its performance (Lee and Ng, 2002). When running LP on the three datasets, we removed the features with occurrence frequency (counted in both training set and test set) less than 3 times.

We investigated two distance measures for LP: cosine similarity and Jensen-Shannon (JS) divergence (Lin, 1991).

For the three datasets, we constructed connected graphs following (Zhu et al., 2003): two instances u, v will be connected by an edge if u is among v's k nearest neighbors, or if v is among u's k nearest neighbors as measured by cosine or JS distance measure. For "interest" and "line" corpora, k is 10 (following (Zhu et al., 2003)), while for SENSEVAL-3 data, k is 5 since the size of dataset for each word in SENSEVAL-3 is much less than that of "interest" and "line" datasets.

4.2 Experiment 1: LP vs. SVM

In this experiment, we evaluated LP and SVM [3] on the data of English lexical sample task in SENSEVAL-3. We used l examples from training set as labeled data, and the remaining training examples and all the test examples as unlabeled data. For each labeled set size l, we performed 20 trials. In each trial, we randomly sampled l labeled examples for each word from training set. If any sense was absent from the sampled labeled set, we redid the sampling. We conducted experiments with different values of l, including $1\% \times N_{w,train}$, $10\% \times N_{w,train}$, $25\% \times N_{w,train}$, $50\% \times N_{w,train}$, $75\% \times N_{w,train}$, $100\% \times N_{w,train}$ ($N_{w,train}$ is the number of examples in training set of word w). SVM and LP were evaluated using accuracy [4] (fine-grained score) on test set of SENSEVAL-3.

We conducted paired t-test on the accuracy figures for each value of l. Paired t-test is not run when percentage$= 100\%$, since there is only one paired accuracy figure. Paired t-test is usually used to estimate the difference in means between normal populations based on a set of random paired observations. $\{\ll, \gg\}$, $\{<, >\}$, and $\sim$ correspond to p-value ≤ 0.01, $(0.01, 0.05]$, and > 0.05 respectively. $\ll$ (or $\gg$) means that the performance of LP is significantly better (or significantly worse) than SVM. $<$ (or $>$) means that the performance of LP is better (or worse) than SVM. $\sim$ means that the performance of LP is almost as same as SVM.

Table 1 reports the average accuracies and paired t-test results of SVM and LP with different sizes of labled data. It also lists the official results of baseline method and top 3 systems in ELS task of SENSEVAL-3.

From Table 1, we see that with small labeled dataset (percentage of labeled data $\leq 10\%$), LP performs significantly better than SVM. When the percentage of labeled data increases from 50% to 75%, the performance of LP_{JS} and SVM become almost same, while LP_{cosine} performs significantly worse than SVM.

---

[3] we used linear SVM^{light}, available at http://svmlight.joachims.org/.

[4] If there are multiple sense tags for an instance in training set or test set, then only the first tag is considered as correct answer. Furthermore, if the answer of the instance in test set is "U", then this instance will be removed from test set.

Table 2: Accuracies from (Li and Li, 2004) and average accuracies of LP with $c \times b$ labeled examples on "interest" and "line" corpora. Major is a baseline method in which they always choose the most frequent sense. MB-D denotes monolingual bootstrapping with decision list as base classifier, MB-B represents monolingual bootstrapping with ensemble of Naive Bayes as base classifier, and BB is bilingual bootstrapping with ensemble of Naive Bayes as base classifier.

Ambiguous words	Accuracies from (Li and Li, 2004)			
	Major	MB-D	MB-B	BB
interest	54.6%	54.7%	69.3%	75.5%
line	53.5%	55.6%	54.1%	62.7%

Ambiguous words	Our results		
	#labeled examples	LP_{cosine}	LP_{JS}
interest	4×15=60	80.2±2.0%	79.8±2.0%
line	6×15=90	60.3±4.5%	59.4±3.9%

4.3 Experiment 2: LP vs. Bootstrapping

Li and Li (2004) used "interest" and "line" corpora as test data. For the word "interest", they used its four major senses. For comparison with their results, we took reduced "interest" corpus (constructed by retaining four major senses) and complete "line" corpus as evaluation data. In their algorithm, c is the number of senses of ambiguous word, and b ($b = 15$) is the number of examples added into classified data for each class in each iteration of bootstrapping. $c \times b$ can be considered as the size of initial labeled data in their bootstrapping algorithm. We ran LP with 20 trials on reduced "interest" corpus and complete "line" corpus. In each trial, we randomly sampled b labeled examples for each sense of "interest" or "line" as labeled data. The rest served as both unlabeled data and test data.

Table 2 summarizes the average accuracies of LP on the two corpora. It also lists the accuracies of monolingual bootstrapping algorithm (MB), bilingual bootstrapping algorithm (BB) on "interest" and "line" corpora. We can see that LP performs much better than MB-D and MB-B on both "interest" and "line" corpora, while the performance of LP is comparable to BB on these two corpora.

4.4 An Example: Word "use"

For investigating the reason for LP to outperform SVM and monolingual bootstrapping, we used the data of word "use" in English lexical sample task of SENSEVAL-3 as an example (totally 26 examples in training set and 14 examples in test set). For data
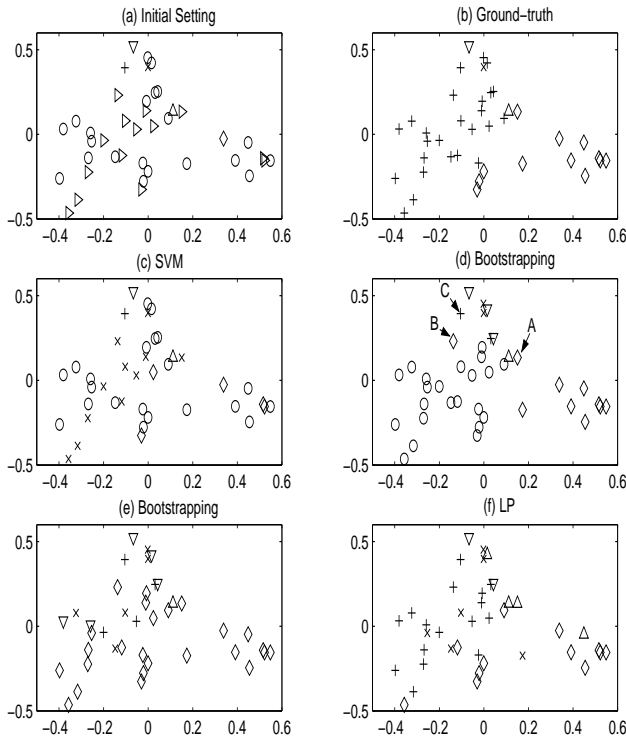
Figure 3: Comparison of sense disambiguation results between SVM, monolingual bootstrapping and LP on word "use". (a) only one labeled example for each sense of word "use" as training data before sense disambiguation (○ and ▷ denote the unlabeled examples in SENSEVAL-3 training set and test set respectively, and other five symbols (+, ×, △, ◇, and ▽) represent the labeled examples with different sense tags sampled from SENSEVAL-3 training set.), (b) ground-truth result, (c) classification result on SENSEVAL-3 test set by SVM (accuracy= $\frac{3}{14}$ = 21.4%), (d) classified data after bootstrapping, (e) classification result on SENSEVAL-3 training set and test set by 1NN (accuracy= $\frac{6}{14}$ = 42.9%), (f) classification result on SENSEVAL-3 training set and test set by LP (accuracy= $\frac{10}{14}$ = 71.4%).

visualization, we conducted unsupervised nonlinear dimensionality reduction[5] on these 40 feature vectors with 210 dimensions. Figure 3 (a) shows the dimensionality reduced vectors in two-dimensional space. We randomly sampled only one labeled example for each sense of word "use" as labeled data. The remaining data in training set and test set served as unlabeled data for bootstrapping and LP. All of these three algorithms are evaluated using accuracy on test set.

From Figure 3 (c) we can see that SVM misclassi-

___

[5]We used $Isomap$ to perform dimensionality reduction by computing two-dimensional, 39-nearest-neighbor-preserving embedding of 210-dimensional input. $Isomap$ is available at http://isomap.stanford.edu/.

fied many examples from class + into class × since using only features occurring in training set can not reveal the intrinsic structure in full dataset.

For comparison, we implemented monolingual bootstrapping with kNN (k=1) as base classifier. The parameter b is set as 1. Only b unlabeled examples nearest to labeled examples and with the distance less than $d_{inter-class}$ (the minimum distance between labeled examples with different sense tags) will be added into classified data in each iteration till no such unlabeled examples can be found. Firstly we ran this monolingual bootstrapping on this dataset to augment initial labeled data. The resulting classified data is shown in Figure 3 (d). Then a 1NN model was learned on this classified data and we used this model to perform classification on the remaining unlabeled data. Figure 3 (e) reports the final classification result by this 1NN model. We can see that bootstrapping does not perform well since it is susceptible to small noise in dataset. For example, in Figure 3 (d), the unlabeled example B [6] happened to be closest to labeled example A, then 1NN model tagged example B with label ◇. But the correct label of B should be + as shown in Figure 3 (b). This error caused misclassification of other unlabeled examples that should have label +.

In LP, the label information of example C can travel to B through unlabeled data. Then example A will compete with C and other unlabeled examples around B when determining the label of B. In other words, the labels of unlabeled examples are determined not only by nearby labeled examples, but also by nearby unlabeled examples. Using this classification strategy achieves better performance than the local consistency based strategy adopted by SVM and bootstrapping.

4.5 Experiment 3: LP_{cosine} vs. LP_{JS}

Table 3 summarizes the performance comparison between LP_{cosine} and LP_{JS} on three datasets. We can see that on SENSEVAL-3 corpus, LP_{JS} per-

___

[6]In the two-dimensional space, example B is not the closest example to A. The reason is that: (1) A is not close to most of nearby examples around B, and B is not close to most of nearby examples around A; (2) we used $Isomap$ to maximally preserve the neighborhood information between any example and all other examples, which caused the loss of neighborhood information between a few example pairs for obtaining a globally optimal solution.

| | LP_{cosine} vs. LP_{JS} | |
Data	p-value	Significance
SENSEVAL-3 (1%)	1.1e-003	$\ll$
SENSEVAL-3 (10%)	8.9e-005	$\ll$
SENSEVAL-3 (25%)	9.0e-009	$\ll$
SENSEVAL-3 (50%)	3.2e-010	$\ll$
SENSEVAL-3 (75%)	7.7e-013	$\ll$
SENSEVAL-3 (100%)	-	-
interest	3.3e-002	$>$
line	8.1e-002	$\sim$

| | $H(D)$ | $H(W)$ | $H(Y_U)$ |
Data	cos. vs. JS	cos. vs. JS	cos. vs. JS
SENSEVAL-3 (1%)	$> (\sqrt{})$	$> (\sqrt{})$	$< (\times)$
SENSEVAL-3 (10%)	$< (\times)$	$> (\sqrt{})$	$< (\times)$
SENSEVAL-3 (25%)	$< (\times)$	$> (\sqrt{})$	$< (\times)$
SENSEVAL-3 (50%)	$> (\sqrt{})$	$> (\sqrt{})$	$> (\sqrt{})$
SENSEVAL-3 (75%)	$> (\sqrt{})$	$> (\sqrt{})$	$> (\sqrt{})$
SENSEVAL-3 (100%)	$< (\circ)$	$> (\circ)$	$< (\circ)$
interest	$< (\sqrt{})$	$> (\times)$	$< (\sqrt{})$
line	$> (\circ)$	$> (\circ)$	$> (\circ)$

forms significantly better than LP_{cosine}, but their performance is almost comparable on "interest" and "line" corpora. This observation motivates us to automatically select a distance measure that will boost the performance of LP on a given dataset.

Cross-validation on labeled data is not feasible due to the setting of semi-supervised learning ($l \ll u$). In (Zhu and Ghahramani, 2002; Zhu et al., 2003), they suggested a label entropy criterion $H(Y_U)$ for model selection, where Y is the label matrix learned by their semi-supervised algorithms. The intuition behind their method is that good parameters should result in confident labeling. Entropy on matrix W ($H(W)$) is a commonly used measure for unsupervised feature selection (Dash and Liu, 2000), which can be considered here. Another possible criterion for model selection is to measure the entropy of $c \times c$ inter-class distance matrix D calculated on labeled data (denoted as $H(D)$), where $D_{i,j}$ represents the average distance between the i-th class and the j-th class. We will investigate three criteria, $H(D)$, $H(W)$ and $H(Y_U)$, for model selection. The distance measure can be automatically

selected by minimizing the average value of function $H(D)$, $H(W)$ or $H(Y_U)$ over 20 trials.

Let Q be the $M \times N$ matrix. Function $H(Q)$ can measure the entropy of matrix Q, which is defined as (Dash and Liu, 2000):

$$S_{i,j} = \exp\left(-\alpha * Q_{i,j}\right), \qquad (1)$$

$$H(Q) = -\sum_{i=1}^{M}\sum_{j=1}^{N}\left(S_{i,j}\log S_{i,j} + (1 - S_{i,j})\log(1 - S_{i,j})\right), \qquad (2)$$

where α is positive constant. The possible value of α is $-\frac{\ln 0.5}{\bar{I}}$, where $\bar{I} = \frac{1}{MN}\sum_{i,j} Q_{i,j}$. S is introduced for normalization of matrix Q. For SENSEVAL-3 data, we calculated an overall average score of $H(Q)$ by $\sum_w \frac{N_{w,test}}{\sum_w N_{w,test}} H(Q_w)$. $N_{w,test}$ is the number of examples in test set of word w. $H(D)$, $H(W)$ and $H(Y_U)$ can be obtained by replacing Q with D, W and Y_U respectively.

Table 3 reports the automatic prediction results of these three criteria.

From Table 3, we can see that using $H(W)$ can consistently select the optimal distance measure when the performance gap between LP_{cosine} and LP_{JS} is very large (denoted by $\ll$ or $\gg$). But $H(D)$ and $H(Y_U)$ fail to find the optimal distance measure when only very few labeled examples are available (percentage of labeled data $\leq 10\%$).

$H(W)$ measures the separability of matrix W. Higher value of $H(W)$ means that distance measure decreases the separability of examples in full dataset. Then the boundary between clusters is obscured, which makes it difficult for LP to locate this boundary. Therefore higher value of $H(W)$ results in worse performance of LP.

When labeled dataset is small, the distances between classes can not be reliably estimated, which results in unreliable indication of the separability of examples in full dataset. This is the reason that $H(D)$ performs poorly on SENSEVAL-3 corpus when the percentage of labeled data is less than 25%.

For $H(Y_U)$, small labeled dataset can not reveal intrinsic structure in data, which may bias the estimation of Y_U. Then labeling confidence ($H(Y_U)$) can not properly indicate the performance of LP. This may interpret the poor performance of $H(Y_U)$ on SENSEVAL-3 data when percentage $\leq 25\%$.

5 Conclusion

In this paper we have investigated a label propagation based semi-supervised learning algorithm for WSD, which fully realizes a global consistency assumption: similar examples should have similar labels. In learning process, the labels of unlabeled examples are determined not only by nearby labeled examples, but also by nearby unlabeled examples. Compared with semi-supervised WSD methods in the first and second categories, our corpus based method does not need external resources, including WordNet, bilingual lexicon, aligned parallel corpora. Our analysis and experimental results demonstrate the potential of this cluster assumption based algorithm. It achieves better performance than SVM when only very few labeled examples are available, and its performance is also better than monolingual bootstrapping and comparable to bilingual bootstrapping. Finally we suggest an entropy based method to automatically identify a distance measure that can boost the performance of LP algorithm on a given dataset.

It has been shown that one sense per discourse property can improve the performance of bootstrapping algorithm (Li and Li, 2004; Yarowsky, 1995). This heuristics can be integrated into LP algorithm by setting weight $W_{i,j} = 1$ if the i-th and j-th instances are in the same discourse.

In the future we may extend the evaluation of LP algorithm and related cluster assumption based algorithms using more benchmark data for WSD. Another direction is to use feature clustering technique to deal with data sparseness and noisy feature problem.

References

Belkin, M., & Niyogi, P.. 2002. Using Manifold Structure for Partially Labeled Classification. *NIPS 15*.

Blum, A., Lafferty, J., Rwebangira, R., & Reddy, R.. 2004. Semi-Supervised Learning Using Randomized Mincuts. *ICML-2004*.

Brown P., Stephen, D.P., Vincent, D.P., & Robert, Mercer.. 1991. Word Sense Disambiguation Using Statistical Methods. *ACL-1991*.

Chapelle, O., Weston, J., & Schölkopf, B. 2002. Cluster Kernels for Semi-supervised Learning. *NIPS 15*.

Dagan, I. & Itai A.. 1994. Word Sense Disambiguation Using A Second Language Monolingual Corpus. *Computational Linguistics*, Vol. 20(4), pp. 563-596.

Dash, M., & Liu, H.. 2000. Feature Selection for Clustering. *PAKDD*(pp. 110–121).

Diab, M., & Resnik. P.. 2002. An Unsupervised Method for Word Sense Tagging Using Parallel Corpora. *ACL-2002*(pp. 255–262).

Hearst, M.. 1991. Noun Homograph Disambiguation using Local Context in Large Text Corpora. *Proceedings of the 7th Annual Conference of the UW Centre for the New OED and Text Research: Using Corpora*, 24:1, 1–41.

Karov, Y. & Edelman, S.. 1998. Similarity-Based Word Sense Disambiguation. *Computational Linguistics*, 24(1): 41-59.

Leacock, C., Miller, G.A. & Chodorow, M.. 1998. Using Corpus Statistics and WordNet Relations for Sense Identification. *Computational Linguistics*, 24:1, 147–165.

Lee, Y.K. & Ng, H.T.. 2002. An Empirical Evaluation of Knowledge Sources and Learning Algorithms for Word Sense Disambiguation. *EMNLP-2002*, (pp. 41-48).

Lesk M.. 1986. Automated Word Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. *Proceedings of the ACM SIGDOC Conference.*

Li, H. & Li, C.. 2004. Word Translation Disambiguation Using Bilingual Bootstrapping. *Computational Linguistics*, 30(1), 1-22.

Lin, D.K.. 1997. Using Syntactic Dependency as Local Context to Resolve Word Sense Ambiguity. *ACL-1997.*

Lin, J. 1991. Divergence Measures Based on the Shannon Entropy. *IEEE Transactions on Information Theory*, 37:1, 145–150.

McCarthy, D., Koeling, R., Weeds, J., & Carroll, J.. 2004. Finding Predominant Word Senses in Untagged Text. *ACL-2004.*

Mihalcea R.. 2004. Co-training and Self-training for Word Sense Disambiguation. *CoNLL-2004.*

Mihalcea R., Chklovski, T., & Kilgariff, A.. 2004. The SENSEVAL-3 English Lexical Sample Task. *SENSEVAL-2004.*

Ng, H.T., Wang, B., & Chan, Y.S.. 2003. Exploiting Parallel Texts for Word Sense Disambiguation: An Empirical Study. *ACL-2003*, pp. 455-462.

Park, S.B., Zhang, B.T., & Kim, Y.T.. 2000. Word Sense Disambiguation by Learning from Unlabeled Data. *ACL-2000.*

Schütze, H.. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24:1, 97–123.

Seo, H.C., Chung, H.J., Rim, H.C., Myaeng. S.H., & Kim, S.H.. 2004. Unsupervised Word Sense Disambiguation Using WordNet Relatives. *Computer, Speech and Language*, 18:3, 253–273.

Szummer, M., & Jaakkola, T.. 2001. Partially Labeled Classification with Markov Random Walks. *NIPS 14*.

Yarowsky, D.. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. *ACL-1995*, pp. 189-196.

Yarowsky, D.. 1992. Word Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora. *COLING-1992*, pp. 454-460.

Zhu, X. & Ghahramani, Z.. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. *CMU CALD tech report CMU-CALD-02-107.*

Zhu, X., Ghahramani, Z., & Lafferty, J.. 2003. Semi-Supervised Learning Using Gaussian Fields and Harmonic Functions. *ICML-2003.*