Pacific Association for Computational Linguistics

LEARNING OF WORD SENSE DISAMBIGUATION RULES BY BELIEF NETWORKS

HIROYUKI SHINNOU AND SHUYA ABE AND MINORU SASAKI

Department of Systems Engineering, Ibaraki University

Department of Systems Engineering, Ibaraki University (Research Institute of Systems Planning, Inc. from 2003-04)

Department of Computer and Information Sciences, Ibaraki University

This paper uses Belief Networks (BN) to solve word sense disambiguation (WSD) problems. For classification problems, the Naive Bayes (NB) is widely used because it generates high performance rules regardless of the simplicity of the model. We use a little more complex model than the NB to get better rules, that is the BN. In the experiments, we attacked Japanese Dictionary Task in SENSEVAL2 and evaluated the BN by comparing it with the NB. One of the features of our BN is that unlabeled data is available in learning. Here, we report on an experiment in which unlabeled data was used in learning.

Key words: Belief Network, Naive Bayes, Word sense disambiguation, Japanese Dictionary Task, unlabeled data

1. INTRODUCTION

In this paper, we apply Belief Networks (BN) to word sense disambiguation (WSD) problems.

Many problems in natural language processing can be converted into classification problems and solved by an inductive learning method. This strategy has been very successful. One of these inductive learning methods, the Naive Bayes (NB) is widely used because it generates high performance rules regardless of the simplicity of the model. By assuming that any two features are independent, the NB can compute actual $P(\texttt{class}|\texttt{instance})$. However, this assumption is not applicable to many real problems. Therefore, a model embedded with a dependency relation may be better than the NB model. To test this hypothesis, we examined the Belief Network[Russell and Norvig1995].

The Belief Network uses a directed acyclic graph (DAG) to represent the model. Thus, the BN can handle more complex models than the NB. A node of a graph has a conditional probability table (CPT). The learning of BN means constructing CPTs of each node. For test instance, the classification can be made using a message-passing algorithm called the Junction Tree algorithm[Huang and Darwiche1996].

In the experiments, we used the BN to solve Japanese Dictionary Tasks in SENSEVAL2 [Shirai2003]. We evaluated the BN by comparing it with the NB. One of the features of our BN is that unlabeled data is available in learning. Here, we report on an experiment in which unlabeled data was used in learning.

2. WSD BY THE BELIEF NETWORK

2.1. WSD by the NB

In a classification problem, let $C = \{c_1, c_2, \cdots, c_m\}$ be a set of classes. An instance x is represented as a feature list: $x = (f_1, f_2, \cdots, f_n)$. We can solve the classification problem by estimating the conditional probability $P(c|x)$. Actually, the class c_x of x, is given by $c_x = arg\max_{c \in C} P(c|x)$.

Bayes theorem shows that $P(c|x) = \frac{P(c)P(x|c)}{P(x)}$. As a result, we get

$$c_x = arg\max_{c \in C} P(c)P(x|c).$$

In the above equation, $P(c)$ is estimated easily; the question is how to estimate $P(x|c)$. Naive Bayes models assume the following:

$$P(x|c) = \prod_{i=1}^{n} P(f_i|c).$$

The estimation of $P(f_i|c)$ is easy, so we can estimate $P(x|c)$ [Mitchell1997].

In this paper, we use four attributes (e1 to e4) for WSD. Suppose that the target word is w_i which is the i-th word in the sentence.

e1: the word w_{i-1}
e2: the word w_{i+1}
e3: two content words to the left of w_i
e4: two content words to the right of w_i

2.2. The BN model extended from the NB model

A Belief Network is a graph to represent the dependence between variables, and gives a concise specification of the joint probability distribution. The graph of the belief network assumes the following:

- A set of random variables makes up the nodes of the network.
- A set of arrows connects pairs of nodes. The arrow from node X to node Y intuitively means that X has a direct influence on Y.
- Each node has a conditional probability table.
- The graph has no directed cycles, that is, it is a directed acyclic graph (DAG).

The network can be regarded as a model to solve the problem. In the case of the NB, that model can be represented by a network like that in Figure 1. Note that there are no links between nodes e_i and e_j. This is because the NB assumes that any two features are independent. In this paper, we use a little complex model represented by the network shown in Figure 2. In this model, two arrows, that is, the arrows from e_3 to e_1 and from e_4 and e_2, are added into the NB model. The model in Figure 2 is more suitable than the NB model because e_i and e_j are not independent in the real world.

Furthermore, the network in Figure 2 satisfies the above conditions required by a graph of the belief network. That is, the network in Figure 2 represents a model of the belief network.

2.3. Use of unlabeled data

In inductive learning, a large amount of labeled data is not generally available because it is too expensive. Therefore we have to estimate a target probability with relatively little labeled data. However, as with the BN, this kind of estimation is not so desirable. Therefore, in this paper, we use unlabeled data to estimate more precise probabilities.

Suppose there is an arrow from node A to node B. If nodes A and B are irrelevant to a class, we do not need labeled data to estimate the probability $P(B|A)$.

In our model, we apply unlabeled data to the arrow from the node e3 to the node e1 and the arrow from the node e4 to the node e2.

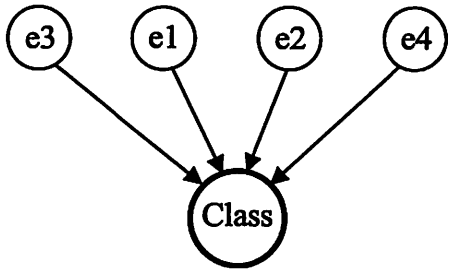

FIGURE 1. Naive Bayes model

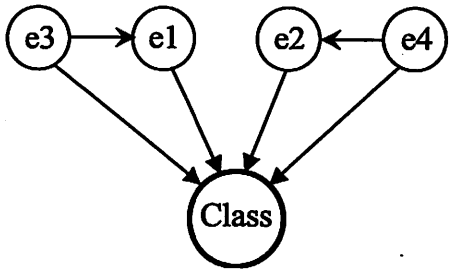

FIGURE 2. Belief Network model

3. EXPERIMENTS

To confirm the effectiveness of the BN model, we applied it to the Japanese Dictionary Task of SENSEVAL2[Shirai2003].

The Japanese Dictionary Task is a set of commonplace WSD problems using 50 nouns and 50 verbs as evaluation words. These words are selected so as to balance the difficulty of WSD. The average number of labeled instances is 177.4 for nouns, 172.7 for verbs. The number of test instances for each evaluation word is 100, so the total number of test instances is 5000 each for noun and verb evaluation words. Unlabeled data consisted of 7585.5 instances per noun and 6571.9 instances per verb evaluation word on average that were taken from Mainichi newspaper articles for 1995. Word segmentations were provided by RWC.

The results are shown in Table 1. In the table, NB and BN mean the NB model and the BN model respectively, and BN+ means the BN model using unlabeled data. The table shows that the BN model was better than the NB model, and the use of unlabeled data had an adverse effect.

TABLE 1. Result of the experiment

	NB	BN	BN+
Noun	75.85%	76.00%	75.74%
Verb	76.77%	76.85%	76.77%

4. DISCUSSION

In this section, we briefly discuss the reason why we could not improve the precision by using unlabeled data.

A value of probabilistic variables in our model is a word. Let M and N be such probabilistic variables, and m and n be values (i.e. words) of M and N, respectively. To estimate the conditional probability $P(M|N)$, we need huge amounts of (m,n) pair data, but we cannot use such huge data in actual.

In this study, we did not consider unseen features, which are not in training data. If a test instance had an unseen feature, we ignored that feature. For this reason, the BN was almost unaffected by the arrow from e3 to e1 (called arrow-31) and the arrow from e4 to e2 (called arrow-42), which are exactly the difference from the NB. As a result, the BN produced a similar result as the NB. When the test instance happened to be similar to a training instance, the BN was influenced by the arrow-31 and the arrow-42. However, in this case, the BN was affect like example based methods, so it was a little better than the NB.

With unlabeled data, there was greater influence from the arrow-31 and the arrow-42. This is because the (m,n) pair data in training data increased. However, the $P(M=m|N=n)$ was not reliable because the (m,n) pair data were still small. Furthermore, we did not have the label for (m,n) unlike the original BN. As a result, the inference using $P(M=m|N=n)$ had an adverse effect.

To apply the BN to real word problems, we should consider unseen features. In theory, we can use an EM algorithm. Recently, the Bound and Collapse method[Ramoni and Sebastiani1998] and Maximum Entropy method[Cowell1999] have been proposed for this problem.

Ignorance of unseen features was our fault. However, we can surely use unlabeled data to learn CPTs of nodes e3 and e4. In future, we will approach this problem by referring to the above studies.

5. CONCLUSION

In this paper, we applied Belief Networks to word sense disambiguation problems.

To evaluate our model, we used the Japanese Dictionary Task of SENSEVAL2. The experiments showed that our model was better than the Naive Bayes model. Furthermore, we tried to use unlabeled data to estimate some probabilities, but could not improve the model for the reason given in the discussion.

In the future, we will investigate how to use unlabeled data in the BN model.

REFERENCES

[Cowell1999] Robert G. Cowell. 1999. Parameter learning from incomplete data for Bayesian networks. In *http://www.city.ac.uk/actstat/pub/stat-20.pdf.*

[Huang and Darwiche1996] Cecil Huang and Adnan Darwiche. 1996. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263.

[Mitchell1997] Tom Mitchell. 1997. *Machine Learning.* McGraw-Hill Companies.

[Ramoni and Sebastiani1998] M. Ramoni and P. Sebastiani. 1998. Parameter estimation in Bayesian networks form incomplete databases. *Intelligent Data Analysis Journal*, 2(1).

[Russell and Norvig1995] Stuart Russell and Peter Norvig. 1995. *Artificial Intelligence A Modern Approach.* Prentice-Hall, Inc.

[Shirai2003] Kiyoaki Shirai. 2003. SENSEVAL-2 Japanese Dictionary Task (in Japanese). *Natural Language Processing*, 10(3):3–24.