Chapter 2
The Naïve Bayes Model in the Context of Word Sense Disambiguation

Abstract This chapter discusses the Naïve Bayes model strictly in the context of word sense disambiguation. The theoretical model is presented and its implementation is discussed. Special attention is paid to parameter estimation and to feature selection, the two main issues of the model's implementation. The EM algorithm is recommended as suitable for parameter estimation in the case of unsupervised WSD. Feature selection will be surveyed in the following chapters.

Keywords Bayesian classification · Expectation-Maximization algorithm · Naïve Bayes classifier

2.1 Introduction

The classical approach to WSD that relies on an underlying Naïve Bayes model represents an important theoretical approach in statistical language processing: Bayesian classification (Gale et al. 1992). The idea of the Bayes classifier (in the context of WSD) is that it looks at the words around an ambiguous word in a large context window. Each content word contributes potentially useful information about which sense of the ambiguous word is likely to be used with it. The classifier does no feature selection. Instead it combines the evidence from all features. The mentioned classifier (Gale et al. 1992) is an instance of a particular kind of Bayes classifier, the Naïve Bayes classifier.

Naïve Bayes is widely used due to its efficiency and its ability to combine evidence from a large number of features. It is applicable if the state of the world that we base our classification on is described as a series of attributes. In our case, we describe the context of the ambiguous word in terms of the words that occur in the context.

The Naïve Bayes assumption is that the attributes used for description are all conditionally independent, an assumption having two main consequences. The first is that all the structure and linear ordering of words within the context are ignored,

F. T. Hristea, *The Naïve Bayes Model for Unsupervised Word Sense Disambiguation*, SpringerBriefs in Statistics, DOI: 10.1007/978-3-642-33693-5_2, © The Author(s) 2013

leading to a so-called “bag of words model”.[1] The other is that the presence of one word in the bag is independent of another, which is clearly not true in the case of natural language. However, in spite of these simplifying assumptions, as noted in (Manning and Schütze 1999), this model has been proven to be quite effective when put into practice. This is not surprising when viewing the Bayesian model from a cognitive perspective, which is an adequate one in the case of a problem concerning natural language processing. And when taking into consideration that, as noted in (Eberhardt and Danks 2011), “without an account of the rationality of the observed input-output relation, the computational level models provide a summary of the observed data, but no rational explanation for the behaviour”.

2.2 The Probability Model of the Corpus and the Bayes Classifier

In order to formalize the described model, we shall present the probability structure of the corpus $\mathscr{C}$. The following *notations* will be used: w is the word to be disambiguated (target word); $s_1, \ldots, s_K$ are possible senses for w; $c_1, \ldots, c_I$ are contexts of w in a corpus $\mathscr{C}$; $v_1, \ldots, v_J$ are words used as contextual features for the disambiguation of w.

Let us note that the contextual features could be some attributes (morphological, syntactical, etc.), or they could be actual “neighboring” content words of the target word. The contextual features occur in a fixed position near w, in a window of fixed length, centered or not on w. In what follows, a window of size n will denote taking into consideration n content words to the left and n content words to the right of the target word, whenever possible. The total number of words taken into consideration for disambiguation will therefore be $2n + 1$. When not enough features are available, the entire sentence in which the target word occurs will represent the context window.

The probability structure of the corpus is based on one main assumption: *the contexts* $\{c_i, i\}$ *in the corpus* $\mathscr{C}$ *are independent*. Hence, the likelihood of $\mathscr{C}$ is given by the product

$$P(\mathscr{C}) = \prod_{i=1}^{I} P(c_i)$$

Let us note that this is a quite natural assumption, as the contexts are not connected, they occur at significant lags in $\mathscr{C}$.

On considering the possible senses of each context, one gets

$$P(\mathscr{C}) = \prod_{i=1}^{I} \sum_{k=1}^{K} P(s_k) \cdot P(c_i \mid s_k)$$

[1] A bag is similar to a set, only it allows repetition.

A model with independent features (usually known as the Naïve Bayes model) assumes that the contextual features are conditionally independent. That is,

$$P\left(c_i \mid s_k\right) = \prod_{v_j \ in \ c_i} P\left(v_j \mid s_k\right) = \prod_{j=1}^{J} \left(P\left(v_j \mid s_k\right)\right)^{\left|v_j \ in \ c_i\right|},$$

where by $\left|v_j \ in \ c_i\right|$ we denote the number of occurrences of feature v_j in context c_i. Then, the likelihood of the corpus $\mathscr{C}$ is

$$P\left(\mathscr{C}\right) = \prod_{i=1}^{I} \sum_{k=1}^{K} P\left(s_k\right) \prod_{j=1}^{J} \left(P\left(v_j \mid s_k\right)\right)^{\left|v_j \ in \ c_i\right|}$$

The parameters of the probability model with independent features are

$$\left\{P\left(s_k\right), k = 1, ..., K \ \text{ and } \ P\left(v_j \mid s_k\right), \quad j = 1, ..., J, k = 1, ..., K\right\}$$

Notation:

- $P\left(s_k\right) = \alpha_k,\ k = 1, ..., K,\ \alpha_k \geq 0$ for all k, $\sum_{k=1}^{K} \alpha_k = 1$
- $P\left(v_j \mid s_k\right) = \theta_{kj},\ k = 1, ..., K,\ j = 1, ..., J,\ \theta_{kj} \geq 0$ for all k and j, $\sum_{j=1}^{J} \theta_{kj} = 1$ for all $k = 1, ..., K$

With this notation, the likelihood of the corpus $\mathscr{C}$ can be written as

$$P\left(\mathscr{C}\right) = \prod_{i=1}^{I} \sum_{k=1}^{K} \alpha_k \prod_{j=1}^{J} \left(\theta_{kj}\right)^{\left|v_j \ in \ c_i\right|}$$

The well known Bayes classifier involves the a posteriori probabilities of the senses, calculated by the Bayes formula for a specified context c,

$$P\left(s_k \mid c\right) = \frac{P\left(s_k\right) \cdot P\left(c \mid s_k\right)}{\sum\limits_{k=1}^{K} P\left(s_k\right) \cdot P\left(c \mid s_k\right)} = \frac{P\left(s_k\right) \cdot P\left(c \mid s_k\right)}{P\left(c\right)},$$

with the denominator independent of senses.

The Bayes classifier chooses the sense s' for which the a posteriori probability is maximal (sometimes called the Maximum A Posteriori classifier)

$$s' = \underset{k=1,...,K}{\arg\max}\ P\left(s_k \mid c\right)$$

Taking into account the previous Bayes formula, one can define the Bayes classifier by the equivalent formula

$$s' = \underset{k=1,...,K}{\arg\max} \left(\log P\left(s_k\right) + \log P\left(c \mid s_k\right)\right)$$

Of course, when implementing a Bayes classifier, one has to estimate the parameters first.

2.3 Parameter Estimation

Parameter estimation is performed by the Maximum Likelihood method, for the available corpus $\mathscr{C}$. That is, one has to solve the optimization problem

$$\max\left(\log P\left(\mathscr{C}\right) \mid \left\{P\left(s_k\right), k = 1, ..., K \text{ and } P\left(v_j \mid s_k\right), \quad j = 1, ..., J,\ k = 1, ..., K\right\}\right)$$

For the Naïve Bayes model, the problem can be written as

$$\max\left(\sum_{i=1}^{I} \log\left(\sum_{k=1}^{K} \alpha_k \prod_{j=1}^{J} \left(\theta_{kj}\right)^{\left|v_j \ in \ c_i\right|}\right)\right) \tag{2.1}$$

with the constraints

$$\left.\begin{array}{l} \displaystyle\sum_{k=1}^{K} \alpha_k = 1 \\ \displaystyle\sum_{j=1}^{J} \theta_{kj} = 1 \end{array}\right. \quad \text{for all } k = 1, ..., K$$

For *supervised disambiguation*, where an annotated training corpus is available, the parameters are simply estimated by the corresponding frequencies:

$$\widehat{\theta}_{kj} = \frac{\left|occurrences\ of\ v_j \ \ in\ a\ context\ of\ sense\ s_k\right|}{\displaystyle\sum_{j=1}^{J} \left|occurrences\ of\ v_j \ \ in\ a\ context\ of\ sense\ s_k\right|},$$

$$k = 1, ..., K;\ j = 1, ..., J$$

$$\widehat{\alpha}_k = \frac{\left|occurrences\ of\ sense\ s_k\ in\ \mathscr{C}\right|}{\left|occurrences\ of\ w\ in\ \mathscr{C}\right|}, \quad k = 1, ..., K$$

For *unsupervised disambiguation*, where no annotated training corpus is available, the maximum likelihood estimates of the parameters are constructed by means of the Expectation-Maximization (EM) algorithm.

For the unsupervised case, the optimization problem (2.1) can be solved only by iterative methods. The Expectation-Maximization algorithm (Dempster et al. 1977) is a very successful iterative method, known as very well fitted for models with missing data.

Each iteration of the algorithm involves two steps:

- estimation of the missing data by the conditional expectation method (E-step)
- estimation of the parameters by maximization of the likelihood function for complete data (M-step)

The E-step calculates the conditional expectations given the current parameter values, and the M-step produces new, more precise parameter values. The two steps alternate until the parameter estimates in iteration $r + 1$ and r differ by less than a threshold ε.

The EM algorithm is guaranteed to increase the likelihood $\log P(\mathscr{C})$ in each step. Therefore, two stopping criteria for the algorithm could be considered: (1) Stop when the likelihood $\log P(\mathscr{C})$ is no longer increasing significantly; (2) Stop when parameter estimates in two consecutive iterations no longer differ significantly.

Further on, we present the EM algorithm for solving the optimization problem (2.1).

The available data, called *incomplete data*, are given by the corpus $\mathscr{C}$. The *missing data* are the senses of the ambiguous words, hence they must be modeled by some random variables

$$h_{ik} = \begin{cases} 1, \text{ context } c_i \text{ generates sense } s_k \\ 0, \qquad\qquad \text{otherwise} \end{cases}, \quad i = 1, ..., I; \; k = 1, ..., K$$

The *complete data* consist of incomplete and missing data, and the corresponding likelihood of the corpus $\mathscr{C}$ becomes

$$P_{\text{complete}}(\mathscr{C}) = \prod_{i=1}^{I} \prod_{k=1}^{K} \left(\alpha_k \prod_{j=1}^{J} \left(\theta_{kj}\right)^{\left|v_j \; in \; c_i\right|} \right)^{h_{ik}}$$

Hence, the log-likelihood for complete data is

$$\log P_{\text{complete}}(\mathscr{C}) = \sum_{i=1}^{I} \sum_{k=1}^{K} h_{ik} \left(\log \alpha_k + \sum_{j=1}^{J} \left|v_j \; in \; c_i\right| \cdot \log \theta_{kj} \right)$$

Each M-step of the algorithm solves the maximization problem

$$\max \left(\sum_{i=1}^{I} \sum_{k=1}^{K} h_{ik} \left(\log \alpha_k + \sum_{j=1}^{J} \left|v_j \; in \; c_i\right| \cdot \log \theta_{kj} \right) \right) \tag{2.2}$$

with the constraints

$$\begin{array}{l} \sum_{k=1}^{K} \alpha_k = 1 \\ \sum_{j=1}^{J} \theta_{kj} = 1 \end{array} \quad \text{for all } k = 1, ..., K$$

For simplicity, we denote the vector of parameters by

$$\psi = (\alpha_1, ..., \alpha_K, \theta_{11}, ..., \theta_{KJ})$$

and notice that the number of independent components (parameters) is $(K - 1) + (KJ - K) = KJ - 1$.

The EM algorithm starts with a random initialization of the parameters, denoted by

$$\psi^{(0)} = \left(\alpha_1^{(0)}, ..., \alpha_K^{(0)}, \theta_{11}^{(0)}, ..., \theta_{KJ}^{(0)}\right)$$

The *iteration* $(r + 1)$ consists in the following two steps:

The E-step computes the missing data, based on the model parameters estimated at iteration r, as follows:

$$h_{ik}^{(r)} = P_{\psi^{(r)}} (h_{ik} = 1 \mid \mathscr{C}),$$

$$h_{ik}^{(r)} = \frac{\alpha_k^{(r)} \cdot \prod_{j=1}^{J} \left(\theta_{kj}^{(r)}\right)^{|v_j \ in \ c_i|}}{\sum_{k=1}^{K} \alpha_k^{(r)} \cdot \prod_{j=1}^{J} \left(\theta_{kj}^{(r)}\right)^{|v_j \ in \ c_i|}}, \quad i = 1, ..., I; \ k = 1, ..., K$$

The M-step solves the maximization problem (2.2) and computes $\alpha_k^{(r+1)}$ and $\theta_{kj}^{(r+1)}$as follows:

$$\alpha_k^{(r+1)} = \frac{1}{I} \sum_{i=1}^{I} h_{ik}^{(r)}, \quad k = 1, ..., K$$

$$\theta_{kj}^{(r+1)} = \frac{\sum_{i=1}^{I} \left|v_j \ in \ c_i\right| \cdot h_{ik}^{(r)}}{\sum_{j=1}^{J} \sum_{i=1}^{I} \left|v_j \ in \ c_i\right| \cdot h_{ik}^{(r)}}, \quad k = 1, ..., K; \ j = 1, ..., J$$

The stopping criterion for the algorithm is “Stop when parameter estimates in two consecutive iterations no longer differ significantly”. That is, stop when

$$\left\| \psi^{(r+1)} - \psi^{(r)} \right\| < \varepsilon,$$

namely

$$\sum_{k=1}^{K} \left(\alpha_k^{(r+1)} - \alpha_k^{(r)} \right)^2 + \sum_{k=1}^{K} \sum_{j=1}^{J} \left(\theta_{kj}^{(r+1)} - \theta_{kj}^{(r)} \right) < \varepsilon$$

It is well known that the EM iterations $\left(\psi^{(r)}\right)_r$ converge to the Maximum Likelihood Estimate $\widehat{\psi} = \left(\widehat{\alpha}_1, ..., \widehat{\alpha}_K, \widehat{\theta}_{11}, ..., \widehat{\theta}_{KJ}\right)$.

Once the parameters of the model have been estimated, we can disambiguate contexts of w by computing the probability of each of the senses based on features v_j occurring in the context c. Making the Naïve Bayes assumption and using the Bayes decision rule, we can decide s' if

$$s' = \underset{k=1,...,K}{\arg\max} \left(\log \widehat{\alpha}_k + \sum_{j=1}^{J} \left| v_j \; in \; c \right| \cdot \log \widehat{\theta}_{kj} \right)$$

Our choice of recommending usage of the EM algorithm for parameter estimation in the case of unsupervised WSD with an underlying Naïve Bayes model is based on the fact that this algorithm has proven itself to be not only a successful iterative method, but also one which fits well to models with missing data. However, our choice is based on previously existing discussions and reported disambiguation results as well. The EM algorithm has equally been used for parameter estimation (together with Gibbs sampling), relatively to an underlying Naïve Bayes model, in (Pedersen and Bruce 1998), to the results of which the accuracies obtained by other disambiguation methods (see Chaps. 3 and 5) have constantly been compared. These are disambiguation accuracies resulted when feeding knowledge of completely different natures to the Naïve Bayes model, as a result of using various different ways of performing feature selection (see Chaps. 3–5). The EM algorithm has equally been used with a Naïve Bayes model in (Gale et al. 1995), in order to distinguish city names from people's names. An accuracy percentage in the mid-nineties, with respect to *Dixon*, a name found to be quite ambiguous, was reported.

References

Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B **39**(1), 1–38 (1977)

Eberhardt, F., Danks, D.: Confirmation in the cognitive sciences: the problematic case of Bayesian models. Mind. Mach. **21**, 389–410 (2011)

Gale, W., Church, K., Yarowsky, D.: A method for disambiguating word senses in a large corpus. Comput. Humanit. **26**(5–6), 415–439 (1992)

Gale, W.A., Church, K.W., Yarowsky, D.: Discrimination decisions for 100,000-dimensional space. Ann. Oper. Res. **55**(2), 323–344 (1995)

Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing. The MIT Press, Cambridge (1999)

Pedersen, T., Bruce, R.: Knowledge lean word-sense disambiguation. In: Proceedings of the 15th National Conference on Artificial Intelligence, pp. 800–805. Madison, Wisconsin (1998)