COS 424: Interacting with Data

Lecturer: Dave Blei | Lecture #12
Scribe: Joe Wenjie Jiang | Mar 15, 2007

Some questions and answers regarding the comments in the last lecture. Which learning algorithms do current spam filters use? The answer is not available since it is business confidential. Generally, Naive Bayes is not utilized, while logistic regression and SVM boosting may be good candidates.

1 Review of Naive Bayes for Text Classification

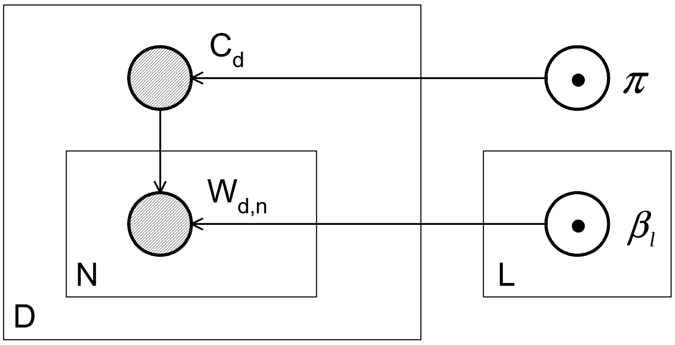

Figure 1: Naive Bayes model for text classification

Here shows the highlight of the concepts. Figure 1 depicts the Naive Bayes model for text classification. We have D documents, each containing N words. The document is labelled a class c_d, which is generated by a distribution π. There are L different classes, each of which possesses a word distribution β_l on a vocabulary of size V. In this graphical model, the joint distribution of a document can be written as:

$$P(c, w_{1:N}|\pi, \beta) = P(c|\pi) \prod_{n=1}^{N} P(w_n|c, \beta) \tag{1}$$

Note that the Naive Bayes assumes that the words are independent given the document's class, which is not true in reality. For instance, if the word "house" appears in a document, it is very likely that "mortgage" will also appear. The generative process of a document follows a multi-nomial distribution:

$$c \sim \mathrm{Multi-nomial}\ (\pi) \tag{2}$$

For each word w_n, it is generated by a multi-nomial distribution given its specific document class c:

$$w_n|c \sim \mathrm{Multi-nomial}\ (\beta_c) \tag{3}$$

We classify these documents based on its posterior distribution, e.g.,

$$\hat{c} = argmax_c\ P(c|w_{1:N}, \pi, \beta) \tag{4}$$

which is the same as choosing the classes that maximize the joint distribution:

$$\hat{c} = argmax_c \; P(c|\pi) \prod_n P(w_n|c, \beta) \tag{5}$$

Therefore, we can write down the log-likelihood as

$$L(\pi, \beta; D) = \sum_{d=1}^{D} \sum_{l=1}^{L} c_d^l \log \pi_l + \sum_{d=1}^{D} \sum_{l=1}^{L} c_d^l \sum_{n=1}^{N} \sum_{v=1}^{V} w_{d,n}^v \log \beta_{l,v} \tag{6}$$

Note that the first term of this log-likelihood function counts the number of documents we saw for each class, and the second term shows the word counts of each document given its class. To produce an MLE of $P(\pi, \beta; D)$, we are lucky to be able to separate the problem into $L + 1$ different MLEs. As shown in Equation (7)

$$\begin{aligned} \hat{\pi_l} &= \sum_{d=1}^{D} c_d^l / D \\ \hat{\beta_{l,v}} &= \sum_{d=1}^{D} c_d^l \sum_{n=1}^{N} w_{d,n}^v / \sum_{d=1}^{D} c_d^l N, \end{aligned} \tag{7}$$

we obtain an estimation of the parameters π and β. The explanation is rather straight-forward: the probability of a document being class l is the number of documents we saw of class l, divided by the total number of documents, and the probability of a word v appearing in a document of class l is the number of times we saw the word v in all documents of class l, divided by the total number of words in documents of class l. Essentially, the estimation is interpreted as the relative frequency.

2 The Theoretical Foundation of Smoothing

2.1 Smoothing

The problem of Naive Bayes is that the estimation of $\beta_{l,v}$ is zero if we never saw this word in the training data. In the last lecture, we introduced smoothing to alleviate the inaccuracy. The basic idea of smoothing is to add a constant term in both the numerator and the denominator, to "smooth" the estimation when some data is not in the training set. The smoothed estimation of $\beta_{l,v}$ is

$$\hat{\beta_{l,v}} = \frac{\sum_{d=1}^{D} c_d^l \sum_{n=1}^{N} w_{d,n}^v + \alpha}{\sum_{d=1}^{D} c_d^l N + V\alpha} \tag{8}$$

and α is some parameter. Smoothing can greatly improve the accuracy of Naive Bayes from 75% to 97%.

Next we are going to explore the probability model of smoothing. Figure 2 shows the graphical model of smoothing. Note that in the original model, β is a parameter (which is represented as a circle with a dot). Now β becomes a random variable with a parameter α, which decides the distribution of β. That is, instead of estimating β merely from the data sets, we consider β which is determined by the training set as well as a hyper-parameter α. We assume β follows a prior distribution $p(\beta|\alpha)$ before we see any data, given a parameter α. Following up we amend this probability by looking at its posterior distribution $p(\beta|\alpha, w_{1:N})$, e.g., what is the best β given our observation in the data?

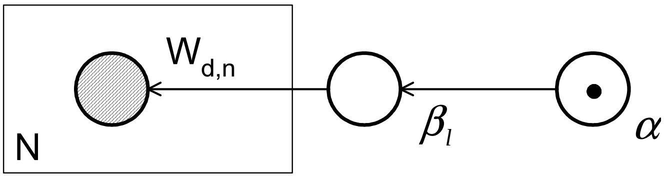

Figure 2: Graphical model for smoothing

2.2 Dirichlet Distribution

We say β lives on the simplex if

$$\beta_v > 0, \sum_v \beta_v = 1 \tag{9}$$

To put it another way, a vector that satisfies (9) is on the simplex. Next we move on to define a distribution over the space of a simplex. Dirichlet (Dir(α)) is a family of continuous multivariate probability distributions over a simplex, parameterized by the vector α of positive values. Formally, let $\vec{\alpha}$ be a positive V-vector. The probability density function of the Dirichlet distribution is defined as

$$p(\beta|\alpha) = Z(\alpha) \cdot \beta_1^{\alpha_1 - 1}\beta_2^{\alpha_2 - 1} \cdots \beta_V^{\alpha_V - 1} \tag{10}$$

wherein $Z(\alpha)$ is a factor that normalizes the pdf such that it integrates to one. We first take a look at the Dirichlet distribution over a 2-simplex, e.g., $V = 2$. The domain of β is defined on $\{\beta | \beta \in \mathbb{R}_+^2, \beta_1 + \beta_2 = 1\}$. Figure 3 shows the shape of a family of Dirichlet distributions in 2D (Note that in 2D this is also called the beta distribution). The mean of a random variable with a Dirichlet distribution can be readily shown:

$$E\big[\beta_v|\alpha\big] = \frac{\alpha_v}{\sum_{v'} \alpha_{v'}} \tag{11}$$

After defining the prior Dirichlet distribution $p(\beta|\alpha)$, we ask, what is the posterior distribution given the data we observed? We have,

$$\begin{aligned} p(\beta|w_{1:N}, \alpha) &= \frac{p(\beta|\alpha) \cdot p(w_{1:N}|\beta)}{p(w_{1:N}|\alpha)} \\ &\propto p(\beta|\alpha) \cdot p(w_{1:N}|\beta) \\ &= Z(\alpha) \prod_{v=1}^{V} \beta_v^{\alpha_v - 1} \prod_{n=1}^{N} \beta_{w_n} \\ &= Z(\alpha) \prod_{v=1}^{V} \beta_v^{\alpha_v - 1 + K_v} \\ &\propto \prod_{v=1}^{V} \beta_v^{\alpha_v - 1 + K_v} \end{aligned} \tag{12}$$

wherein K_v is the number of appearance of word v in the training data. Equation (12) reveals the truth that the posterior distribution given the observed data is still a Dirichlet distribution $\text{Dir}(\vec{\alpha} + \vec{K})$, where $\vec{K}$ is a vector of word counts! So $p(\beta|w_{1:N}, \alpha)$ is a Dirichlet distribution with parameter $\vec{\alpha} + \vec{K}$. This nice property is called *conjugacy*. We can further write down the expectation given the posterior distribution,

$$E\big[\beta|w_{1:N}, \alpha\big] = \frac{K_v + \alpha_v}{\sum_{v'} K_{v'} + \alpha} \tag{13}$$

When $\alpha_v = \alpha$, e.g., all the components have the same value, the expectation simplifies to

$$E[\beta|w_{1:N}, \alpha] = \frac{K_v + \alpha}{\sum_{v'}(K_{v'} + \alpha)} = \frac{K_v + \alpha}{N + V \cdot \alpha} \tag{14}$$

which looks like what smoothing does with a smoothing parameter α. Essentially, the probabilistic interpretation of smoothing is more clear: when we have little data in the training set, we assume that the data we did not see and what we saw are uniformly distributed on a simplex in prior.

How to choose a good parameter α is important. Generally speaking there are two approaches: (i) MLE fund with a method called "empirical Bayes" (ii) cross validation. When $\alpha = 1$, it is called the Laplace smoothing. When $\alpha = 0.5$, it is called Jeffery's prior. However, the purely Bayesian approach towards smoothing argues that we choose α before we actually see the data set.

3 Preliminary of EM Algorithm

In the next two lectures, we are going to explore the case when the random variable c_d becomes an empty circle, e.g., c_d becomes a hidden variable. So we do not observe the class of our data any more and we have no idea which document comes from which class. The only observed data is the documents $\{\vec{w_d}\}_{d=1}^{D}$. So we are introducing the idea of clustering into our graphical models. It has got a name *mixture model*, or *model-based clustering*. In the case of text classification, it is modeled as a mixture of multi-nomials. The estimation includes two parameters: (i) π, which is the distribution over possible groups, and the data turns out to be a mixture of groups with different proportions. (ii) β, which is the per group word distribution, and the data of each group is a mixture of its components. The observed data is $w_{d,n}$. The conditional probability density function can be written as

$$p(w_{1:N}|\beta, \pi) = \sum_{l=1}^{L} p(c = l|\pi) \prod_{n=1}^{N} p(w_n|\beta, c = l) \tag{15}$$

The corresponding log-likelihood is

$$\begin{aligned} L(\beta, \pi, D) &= \log \prod_{d=1}^{D} p(\vec{w_d}|\pi, \beta) \\ &= \sum_{d=1}^{D} \log \sum_{l} p(c = l|\pi) \prod_{n=1}^{N} p(w_n|c = l, \beta) \end{aligned} \tag{16}$$

We have no analytical solutions for this MLE since there are summations in the log. One option is to perform numerical optimization. An alternative is to leverage the iterative method to solve this optimization problem, e.g., EM algorithm. Actually EM algorithm is a powerful machinery to solve arbitrary graphical models with any hidden variables.

Before we formally introduce the EM algorithm, let us first recall what K-mean algorithm does. The K-mean algorithm repeatedly performs the following two steps:

- Partition the data according to current cluster centers.
- Estimate new cluster centers based on the partition.

The EM algorithm possesses the similar idea. The question is, suppose we know the grouping of the data, which is the best way to estimate the parameters? Use Naive Bayes! Therefore, we present the high level description of the EM algorithm as:

- Suppose we have current estimates of $\pi^{(t)}, \beta^{(t)}$
- Replace c_d with $E\left[c_d|w_d, \pi^{(t)}, \beta^{(t)}\right]$

In the next lecture, we are going to see how the expectation is calculated.

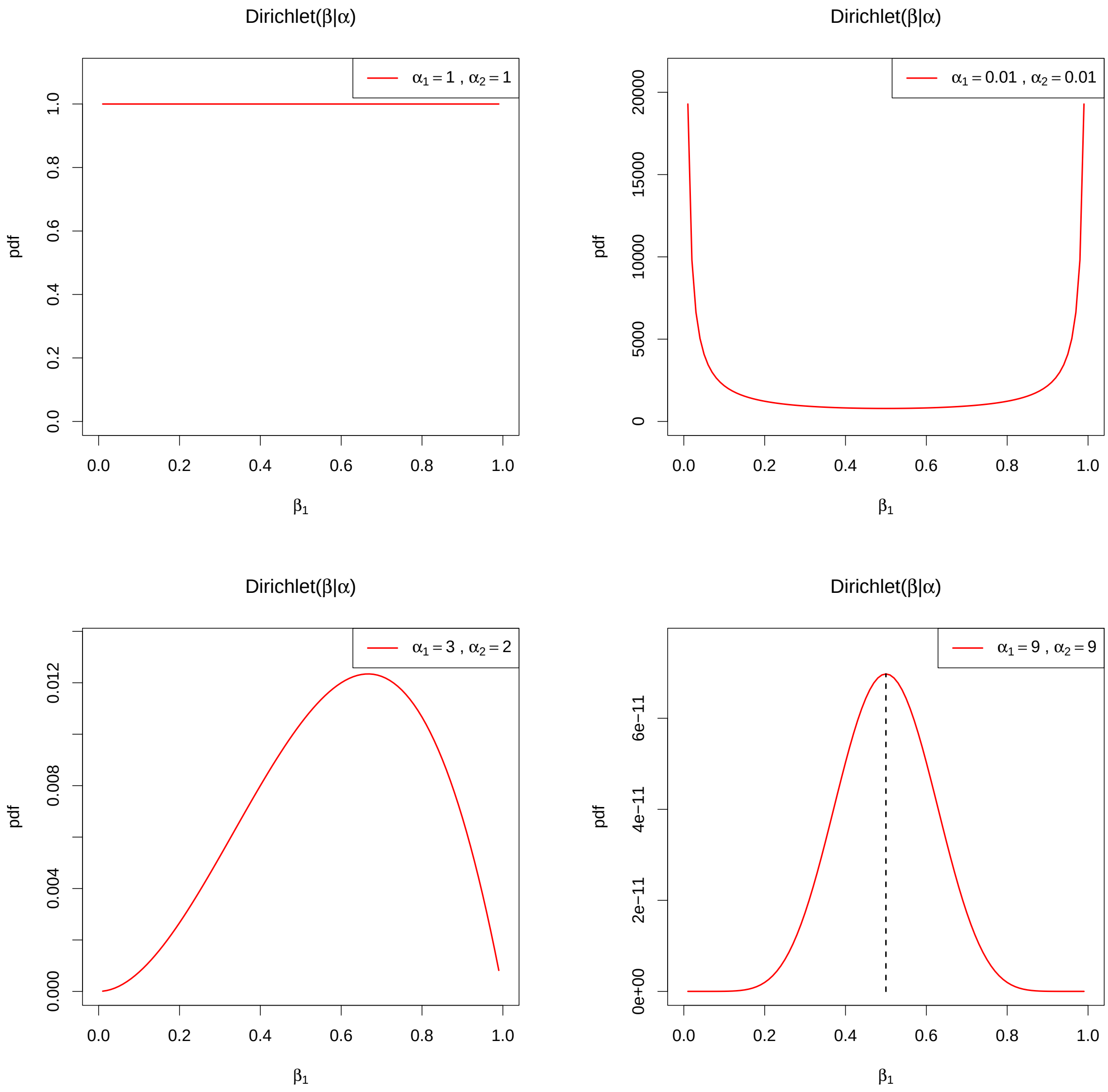

Figure 3: A family of Dirichlet distributions