Naïve Bayes Classifier

Lecturer: Xiaojin Zhu   *jerryzhu@cs.wisc.edu*

We are given a set of documents $x_1, \ldots, x_n$, with the associated class labels $y_1, \ldots, y_n$. We want to learn a model that will predict the label y for any future document x. This task is known as classification. Naive Bayes is one classification method.

1 Naive Bayes Classifier

Let each document be represented by $x = (c_1, \ldots, c_v)^\top$ the word count vector, otherwise known as *bag of word* representation. We assume within each class y, the probability of a document follows the multinomial distribution with parameter θ_y:

$$p(x|y) \propto \prod_{w=1}^{v} \theta_{yw}^{c_w}. \tag{1}$$

The log likelihood is

$$\log p(x|y) = x^\top \log \theta_y + const. \tag{2}$$

Note different classes have different θ_y's. Also note that the multinomial distribution assume *conditional independence* of feature dimensions $1, \ldots, v$ given the class y. We know this is not true in reality, and more sophisticated models would assume otherwise. For this reason, such assumption on independence of features is known as the *naïve Bayes* assumption.

If we know $p(x|y)$ and $p(y)$ for all classes, classification is done via the Bayes rule:

$$
\begin{aligned}
y^* &= \underset{y}{\arg\max}\, p(y|x) \tag{3}\\
&= \underset{y}{\arg\max}\, \frac{p(x|y)p(y)}{p(x)} \tag{4}\\
&= \underset{y}{\arg\max}\, p(x|y)p(y) \tag{5}\\
&= \underset{y}{\arg\max}\, x^\top \log \theta_y + \log p(y), \tag{6}
\end{aligned}
$$

The process of computing the conditional distribution $p(y|x)$ of the unknown variable (y) given observed variables (x) is called *inference*. Making classification predictions given $p(x|y), p(y)$, and x is called *inference*.

Where do we get $p(x|y)$ and $p(y)$? These are the parameters of the model, and we learn them from the training set. Given a training set $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, *training* or *parameter learning* involves finding the best parameters $\Theta = \{\pi, \theta_1, \ldots, \theta_C\}$. Our complete model is $p(y = j) = \pi_j$, and $p(x|y = j) = \text{Mult}(x; \theta_j) \propto \prod_{w=1}^{V} \theta_{jw}^{x_w}$. For simplicity we use the MLE here, but MAP is common too. We maximize the joint (log)

likelihood of the training set:

$$\ell \;=\; \log p((x,y)_{1:n}|\Theta) \;;\; \text{hide } \Theta \text{ below} \tag{7}$$

$$=\; \log \prod_{i=1}^{n} p(x_i, y_i) \tag{8}$$

$$=\; \sum_{i=1}^{n} \log p(x_i, y_i) \tag{9}$$

$$=\; \sum_{i=1}^{n} \log p(y_i) + \log p(x_i|y_i). \tag{10}$$

We can formulate this as a constrained optimization problem,

$$\max_{\Theta} \qquad \ell \tag{11}$$

$$\text{s.t.} \quad \sum_{j=1}^{C} \pi_j = 1, \; C \text{ is the number of classes} \tag{12}$$

$$\sum_{w=1}^{v} \theta_{jw} = 1, \forall j = 1 \ldots C. \tag{13}$$

It is easy to solve it using Lagrange multipliers and arrive at

$$\pi_j \;=\; \frac{\sum_{i=1}^{n}[y_i = j]}{n} \tag{14}$$

$$\theta_{jw} \;=\; \frac{\sum_{i:y_i=j} x_{iw}}{\sum_{i:y_i=j} \sum_{u=1}^{V} x_{iu}}. \tag{15}$$

These MLEs are intuitive: they are the class frequency in the training set, and the word frequency within each class.

Note that the concepts of inference and parameter learning described above are fairly general. The only special thing is the naïve Bayes assumption (i.e., unigram language model for $p(x|y)$) which assumes conditional independence of features. This makes it a Naïve Bayes classifier.

1.1 Naive Bayes as a Linear Classifier

Consider binary classification where $y = 0$ or 1. Our classification rule with $\arg\max$ can equivalently be expressed with log odds ratio

$$f(x) \;=\; \log \frac{p(y=1|x)}{p(y=0|x)} \tag{16}$$

$$=\; \log p(y=1|x) - \log p(y=0|x) \tag{17}$$

$$=\; (\log \theta_1 - \log \theta_0)^{\top} x + (\log p(y=1) - \log p(y=0)). \tag{18}$$

The decision rule is to classify x with $y = 1$ if $f(x) > 0$, and $y = 0$ otherwise. Note for given parameters, this is a *linear function* in x. That is to say, the Naive Bayes classifier induces a linear decision boundary in feature space $\mathcal{X}$. The boundary takes the form of a hyperplane, defined by $f(x) = 0$.

1.2 Naive Bayes as a Generative Model

A generative model is a probabilistic model which describe the full generation process of the data, i.e. the joint probability $p(x, y)$. Our Naive Bayes model consists of $p(y)$ and $p(x|y)$, and does just that: One can generate data (x, y) by first sample $y \sim p(y)$, and then sample word counts from the multinomial $p(x|y)$.

There is another family of models known as discriminative models, which do not model $p(x, y)$. Instead, they directly model the conditional $p(y|x)$, which is directly related to classification. We will see our first discriminative model when we discuss logistic regression.

1.3 Naive Bayes as a Special Case of Bayes Networks

A *Bayes Network* is a directed graph that represent a family of probability distributions. This is covered in detail in [cB] Chapter 8.1, 8.2. Outline:

- nodes: each node is a random variable. We have one y node, and v x_w nodes.

- directed edges: No directed cycles allowed, i.e. must be a DAG. For naive Bayes, from y to x_w.

- meaning: the joint probability on all nodes $s_{1:K}$ is factorized in a particularly form

$$p(s) = \prod_{i=1}^{K} p(s_i|\mathrm{pa}(s_i)), \tag{19}$$

where $\mathrm{pa}(s_i)$ are the parents of s_i. For naive Bayes, $p(x_{1:v}, y) = p(y) \prod_{i=1}^{v} p(x_i|y)$.

- observed nodes: nodes with known values, e.g. $x_{1:v}$. Shaded.

- plate: a lazy way to duplicate the node (and associated edges) multiple times. Our $x_{1:v}$ can be condensed into a plate.