Naïve Bayes Classifier

COMP4901K and MATH4824B
Yangqiu Song

Suppose we have a set of documents $d_1, \ldots, d_N$, with associated labels $y_1, \ldots, y_N$. Each document d_i is a sequence of word tokens $w_{d_i,1}, \ldots, w_{d_i,L_{d_i}}$. We also build a vocabulary $\{x_1, \ldots, x_V\}$ with V word types.

We use italic to represent scalars (e.g., a, x, y) and boldface to represent vectors and matrices (e.g., $\mathbf{x}$, $\mathbf{y}$, $\mathbf{A}$).

1 Document and Label Representation

For each document d, we count the term-frequency of each word x_j: $c_d(x_j)$. Similarly, for d_i, we denote the term-frequency of x_j in d_i as $c_{d_i}(x_j)$. Then for each document, we can build a vector representation

$$\mathbf{x}_i = [c_{d_i}(x_1), \ldots, c_{d_i}(x_V)]^\top \doteq [\mathbf{x}_i^{(1)}, \ldots, \mathbf{x}_i^{(V)}]^\top \in \mathbb{R}^V.$$

We can build a matrix for all training documents as: $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times V}$ where each document is a row in the matrix.

For each label $y_i \in \{1, \ldots, K\}$, we also convert it into a matrix representation:

$$\mathbf{y}_i = [0, \ldots, 1_{y_i = k}, \ldots, 0]^\top \in \mathbb{R}^K.$$

Then we can build a label matrix $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^\top \in \mathbb{R}^{N \times K}$.

2 Naïve Bayes Classifier

A classifier in general predicts the label probability given the evidence of features:

$$P(y|\mathbf{x}). \tag{1}$$

2.1 Naïve Bayes Assumption

A naïve Bayes classifier makes the assumption that, given the label, all features are *conditionally independent*:

$$P(\mathbf{x}|y) = \prod_j^V P(\mathbf{x}^{(j)}|y) \tag{2}$$

where $\mathbf{x} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(V)}]^\top \in \mathbb{R}^V$.

By using Bayes formula:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{3}$$

we rewrite the classifier as:

$$\begin{aligned} P(y|\mathbf{x}) &= \frac{P(\mathbf{x}|y)P(y)}{P(\mathbf{x})} \\ &= \frac{P(y)\prod_j P(\mathbf{x}^{(j)}|y)}{P(\mathbf{x})}. \end{aligned} \tag{4}$$

After the simplification of the classification probability, we can see that, the complexity is reduced to $O(\prod_j |\mathbf{x}^{(j)}| \cdot K + K)$ where $|\mathbf{x}^{(j)}|$ is the number of values that $\mathbf{x}^{(j)}$ can take. If $\mathbf{x}^{(j)}$ is occurrence of a word, the complexity is $VK + K$.

2.2 Naïve Bayes For Texts

Suppose we are given a document $d = w_1, \ldots, w_{M_d}$. We make use of the naïve Bayes assumption over words so that given the document label, all observations of words are conditionally independent:

$$P(d|y) = \prod_n^{M_d} P(w_n|y). \tag{5}$$

Since words can be duplicated in a document, we can convert the above equation into word type based probabilities:

$$P(d|y) = \prod_j^V P(x_j|y)^{c_d(x_j)} = \prod_j^V P(x_j|y)^{\mathbf{x}^{(j)}}. \tag{6}$$

2.3 Simplified Notations for Parameters

We define $\pi_k \doteq P(y = k)$ and use $\boldsymbol{\pi} = [\pi_1, \ldots, \pi_K]^\top$ to store all $P(y)$. Here we have $\sum_k \pi_k = 1$.

We also define $\theta_{k,j} \doteq P(x_j|y = k)$ and use

$$\boldsymbol{\Theta} = \begin{bmatrix} \theta_{1,1} & \ldots & \theta_{1,V} \\ \vdots & \ddots & \vdots \\ \theta_{K,1} & \ldots & \theta_{K,V} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta}_1^\top \\ \vdots \\ \boldsymbol{\theta}_K^\top \end{bmatrix}$$

to store all $P(x|y)$. Here we have $\sum_{j=1}^V \theta_{k,j} = 1$.

Then we can rewrite

$$\begin{aligned} P(y = k|d) &= \frac{P(d|y)P(y)}{P(\mathbf{x})} \\ &= \frac{P(y) \prod_j^V P(x_j|y)^{c_d(x_j)}}{P(\mathbf{x})} \\ &= \frac{\pi_k \prod_j^V \theta_{k,j}^{c_d(x_j)}}{P(\mathbf{x})}. \end{aligned} \tag{7}$$

The prediction of label y is given by taking the argument of this function:

$$\begin{aligned} \hat{y} &= \arg\max_y \frac{P(y) \prod_j^V P(x_j|y)^{c_d(x_j)}}{P(\mathbf{x})} \\ &= \arg\max_y P(y) \prod_j^V P(x_j|y)^{c_d(x_j)} \\ &= \arg\max_y \log P(y) + \sum_j c_d(x_j) \log P(\mathbf{x}_j|y) \\ &= \arg\max_k \log \pi_k + \sum_j c_d(x_j) \log \theta_{k,j} \\ &= \arg\max_k \log \pi_k + \mathbf{x}^\top \log \boldsymbol{\theta}_k \end{aligned} \tag{8}$$

We denote $\boldsymbol{\pi}$ and $\boldsymbol{\Theta}$ as parameters of a naïve Bayes classifier.

2.4 Learning

We estimate the parameters based on the observations we sampled from the true distribution (consider the Urn Model). Given a training set $\{(d_1, y_1), \ldots, (d_N, y_N)\}$, we maximize the joint (log) likelihood of the training set:

$$\begin{aligned}
\mathcal{L} &= \log \prod_i^N P(d_i, y_i | \boldsymbol{\pi}, \boldsymbol{\Theta}) \\
&= \sum_i^N \log P(d_i, y_i | \boldsymbol{\pi}, \boldsymbol{\Theta}) \\
&= \sum_i^N \log \pi_{y_i} + \mathbf{x}_i^\top \log \boldsymbol{\theta}_{y_i}. \qquad (9)
\end{aligned}$$

We can maximize this log-likelihood by optimizing the following problem:

$$\begin{aligned}
&\max_{\boldsymbol{\pi}, \boldsymbol{\Theta}} \mathcal{L} \\
&s.t. \sum_k^K \pi_k = 1 \quad \text{and} \quad \forall k, \ \sum_{j=1}^V \theta_{k,j} = 1. \qquad (10)
\end{aligned}$$

It is easy to solve it using Lagrange multipliers.[1] Taking π_k as an example, we optimize

$$\mathcal{L}(\boldsymbol{\pi}) = \sum_i^N \log \pi_{y_i} + \lambda(\sum_k^K \pi_k - 1). \qquad (11)$$

Taking partial derivatives w.r.t. π_k and setting to zero, we have:

$$\frac{\partial \mathcal{L}(\boldsymbol{\pi})}{\partial \pi_k} = \sum_i^N \frac{I_{y_i=k}}{\pi_k} + \lambda = 0. \qquad (12)$$

Then we have

$$\pi_k = -\frac{\sum_i^N I_{y_i=k}}{\lambda}. \qquad (13)$$

Substituting it into $\sum_k^K \pi_k = 1$ we have:

$$\lambda = -\sum_k^K \sum_i^N I_{y_i=k} = -N. \qquad (14)$$

So we have

$$\pi_k = \frac{\sum_i^N I_{y_i=k}}{N} \qquad (15)$$

[1]The optimization part is out of the scope of this class.

where $I_{(\cdot)}$ is an indicator function: $I_{true} = 1$ and $I_{false} = 0$.

Similarly, we have

$$\theta_{k,j} = \frac{\sum_i^N I_{y_i=k} c_i(x_j)}{\sum_j^V \sum_i^N I_{y_i=k} c_i(x_j)}. \tag{16}$$

The maximum likelihood solution is intuitive: count the class frequency in the training set, and the word frequency within each class.

3 Implementations

By using the matrix representation of $\mathbf{X}$, $\mathbf{Y}$, $\boldsymbol{\pi}$ and $\boldsymbol{\Theta}$, we have:

$$\boldsymbol{\pi}^\top = \text{normalize}(\text{rowsum}(\mathbf{Y})) \tag{17}$$

and

$$\boldsymbol{\Theta} = \text{normalize each row}(\mathbf{Y}^\top \mathbf{X}) \tag{18}$$

4 Naïve Bayes as a Linear Classifier

Let's consider a binary classification where $y \in \{0, 1\}$. Our classification rule with argmax is equal to log odds ratio:

$$f(\mathbf{x}) = \log \frac{P(y = 1|\mathbf{x})}{P(y = 0|\mathbf{x})} \tag{19}$$

$$= \log P(y = 1|\mathbf{x}) - \log P(y = 0|\mathbf{x}) \tag{20}$$

$$= (\log \pi_1 - \log \pi_0) + \mathbf{x}^\top (\log \boldsymbol{\theta}_1 - \log \boldsymbol{\theta}_0). \tag{21}$$

The decision rule is to classify $\mathbf{x}$ to 1 if $f(\mathbf{x}) > 0$ and 0 otherwise. This is a linear function in $\mathbf{x}$. Naïve Bayes classifier induces a linear decision boundary in feature space of $\mathcal{X}$.