Naïve Bayes

MIT 15.097 Course Notes

Cynthia Rudin

Thanks to Şeyda Ertekin
Credit: Ng, Mitchell

The Naïve Bayes algorithm comes from a generative model. There is an important distinction between generative and discriminative models. In all cases, we want to predict the label y, given x, that is, we want $P(Y = y|X = x)$. Throughout the paper, we'll remember that the probability distribution for measure P is over an unknown distribution over $\mathcal{X} \times \mathcal{Y}$.

Naïve Bayes Generative Model	Estimate $P(X = x\|Y = y)$ and $P(Y = y)$ and use Bayes rule to get $P(Y = y\|X = x)$
Discriminative Model	Directly estimate $P(Y = y\|X = x)$

Most of the top 10 classification algorithms are discriminative (K-NN, CART, C4.5, SVM, AdaBoost).

For Naïve Bayes, we make an assumption that if we know the class label y, then we know the mechanism (the random process) of how x is generated.

Naïve Bayes is great for very high dimensional problems because it makes a very strong assumption. Very high dimensional problems suffer from the curse of dimensionality – it's difficult to understand what's going on in a high dimensional space without tons of data.

Example: Constructing a spam filter. Each example is an email, each dimension "j" of vector $\mathbf{x}$ represents the presence of a word.

$$\mathbf{x} = \begin{array}{c} \\ \\ \\ \\ \\ \\ \\ \end{array}\begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix} \begin{array}{c} \text{a} \\ \text{aardvark} \\ \text{aardwolf} \\ \vdots \\ \text{buy} \\ \vdots \\ \text{zyxt} \end{array}$$

This $\mathbf{x}$ represents an email containing the words "a" and "buy", but not "aardvark" or "zyxt". The size of the vocabulary could be ∼50,000 words, so we are in a 50,000 dimensional space.

Naïve Bayes makes the assumption that the $x^{(j)}$'s are conditionally independent given y. Say $y = 1$ means spam email, word 2,087 is "buy", and word 39,831 is "price." Naïve Bayes assumes that if $y = 1$ (it's spam), then knowing $x^{(2,087)} = 1$ (email contains "buy") won't effect your belief about $x^{(39,381)}$ (email contains "price").

Note: This does not mean $x^{(2,087)}$ and $x^{(39,831)}$ are independent, that is,

$$P(X^{(2,087)} = x^{(2,087)}) = P(X^{(2,087)} = x^{(2,087)}|X^{(39,831)} = x^{(39,831)}).$$

It only means they are conditionally independent given y. Using the definition of conditional probability recursively,

$$\begin{aligned} &P(X^{(1)} = x^{(1)}, \dots, X^{(50,000)} = x^{(50,000)}|Y = y) = \\ &\quad P(X^{(1)} = x^{(1)}|Y = y)P(X^{(2)} = x^{(2)}|Y = y, X^{(1)} = x^{(1)}) \\ &\quad\; P(X^{(3)} = x^{(3)}|Y = y, X^{(1)} = x^{(1)}, X^{(2)} = x^{(2)}) \\ &\quad\quad \dots P(X^{(50,000)} = x^{(50,000)}|Y = y, X^{(1)} = x^{(1)}, \dots, X^{(49,999)} = x^{(49,999)}). \end{aligned}$$

The independence assumption gives:

$$\begin{aligned} &P(X^{(1)} = x^{(1)}, \dots, X^{(n)} = x^{(n)}|Y = y) \\ &= P(X^{(1)} = x^{(1)}|Y = y)P(X^{(2)} = x^{(2)}|Y = y) \dots P(X^{(n)} = x^{(n)}|Y = y) \\ &= \prod_{j=1}^{n} P(X^{(j)} = x^{(j)}|Y = y). \end{aligned} \tag{1}$$

Bayes rule says

$$P(Y=y|X^{(1)}=x^{(1)},...,X^{(n)}=x^{(n)})=\frac{P(Y=y)P(X^{(1)}=x^{(1)},...,X^{(n)}=x^{(n)}|Y=y)}{P(X^{(1)}=x^{(1)},...,X^{(n)}=x^{(n)})}$$

so plugging in (1), we have

$$P(Y=y|X^{(1)}=x^{(1)},...,X^{(n)}=x^{(n)})=\frac{P(Y=y)\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}|Y=y)}{P(X^{(1)}=x^{(1)},...,X^{(n)}=x^{(n)})}$$

For a new test instance, called $\mathbf{x}_{\text{test}}$, we want to choose the most probable value of y, that is

$$\begin{aligned} y_{NB} &\in \arg\max_{\tilde{y}} \frac{P(Y=\tilde{y})\prod_j P(X^{(1)}=x^{(1)}_{\text{test}},...,X^{(n)}=x^{(n)}_{\text{test}}|Y=\tilde{y})}{P(X^{(1)}=x^{(1)}_{\text{test}},...,X^{(n)}=x^{(n)}_{\text{test}})} \\ &= \arg\max_{\tilde{y}} P(Y=\tilde{y})\prod_{j=1}^{n}P(X^{(j)}=x^{(j)}|Y=\tilde{y}). \end{aligned}$$

So now, we just need $P(Y=\tilde{y})$ for each possible $\tilde{y}$, and $P(X^{(j)}=x^{(j)}_{\text{test}}|Y=\tilde{y})$ for each j and $\tilde{y}$. Of course we can't compute those. Let's use the empirical probability estimates:

$$\hat{P}(Y=\tilde{y})=\frac{\sum_i \mathbb{1}_{[y_i=\tilde{y}]}}{m}=\text{ fraction of data where the label is } \tilde{y}$$

$$\hat{P}(X^{(j)}=x^{(j)}_{\text{test}}|Y=\tilde{y})=\frac{\sum_i \mathbb{1}_{\left[x_i^{(j)}=x^{(j)}_{\text{test}},y_i=\tilde{y}\right]}}{\sum_i \mathbb{1}_{[y_i=\tilde{y}]}}=\text{Conf}(Y=\tilde{y}\rightarrow X^{(j)}=x^{(j)}_{\text{test}}).$$

That's the simplest version of Naïve Bayes:

$$y_{NB}\in\arg\max_{\tilde{y}}\hat{P}(Y=\tilde{y})\prod_{j=1}^{n}\hat{P}(X^{(j)}=x^{(j)}_{\text{test}}|Y=\tilde{y}).$$

There could potentially be a problem that most of the conditional probabilities are 0 because the dimensionality of the data is very high compared to the amount of data. This causes a problem because if even one $\hat{P}(X^{(j)}=x^{(j)}_{\text{test}}|Y=\tilde{y})$ is zero then the whole right side is zero. In other words, if no training examples from class "spam" have the word "tomato," we'd never classify a test example containing the word "tomato" as spam!

To avoid this, we (sort of) set the probabilities to a small positive value when there are no data. In particular, we use a "Bayesian shrinkage estimate" of $P(X^{(j)} = x^{(j)}_{\text{test}} | Y = \tilde{y})$ where we add some hallucinated examples. There are K hallucinated examples spread evenly over the possible values of $X^{(j)}$. K is the number of distinct values of $X^{(j)}$. The probabilities are pulled toward $1/K$. So, now we replace:

$$\hat{P}(X^{(j)} = x^{(j)}_{\text{test}} | Y = \tilde{y}) = \frac{\sum_i \mathbb{1}_{\left[x_i^{(j)} = x^{(j)}_{\text{test}}, y_i = \tilde{y}\right]} + 1}{\sum_i \mathbb{1}_{[y_i = \tilde{y}]} + K}$$

$$\hat{P}(Y = \tilde{y}) = \frac{\sum_i \mathbb{1}_{[y_i = \tilde{y}]} + 1}{m + K}$$

This is called Laplace smoothing. The smoothing for $\hat{P}(Y = \tilde{y})$ is probably unnecessary and has little to no effect.

Naïve Bayes is not necessarily the best algorithm, but is a good first thing to try, and performs surprisingly well given its simplicity!

There are extensions to continuous data and other variations too.

PPT Slides

15.097 Prediction: Machine Learning and Statistics
Spring 2012