Naive Bayes: Estimating $\mathbf{P}(\mathbf{x} \mid y)$

Cornell CS 4/5780 — Spring 2022

Case #1: Categorical features

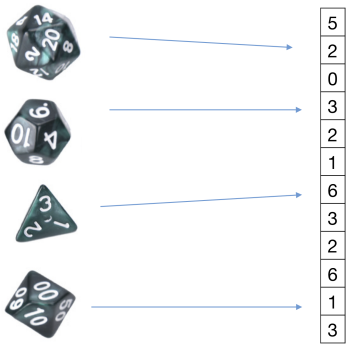


Illustration of categorical NB. For d dimensional data, there exist d independent dice for each class. Each feature has one die per class. We assume training samples were generated by rolling one die after another. The value in dimension i corresponds to the outcome that was rolled with the i^{th} die.

Feature Assumption: the αth feature lies in a finite set of K_α categories $x_\alpha \in \mathcal{X}_\alpha = \{f_1, f_2, \cdots, f_{K_\alpha}\}$. That is, $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_d$. Note that the case of binary features is just a specific case of this, where $K_\alpha = 2$. An example of such a setting may be personal data where a feature could be *marital status* (single/married) or *gender* or *state of residence* (Alabama/Alaska/Arizona/et cetera).

Model Assumption: We model $\mathbf{P}(x_\alpha \mid y)$ as a categorical distribution.

$$\mathbf{P}(x_\alpha = j \mid y = c) = \theta_{j,c,\alpha} \text{ where } \theta_{j,c,\alpha} \geq 0 \text{ and } \sum_{j \in \mathcal{X}_\alpha} \theta_{j,c,\alpha} = 1 \text{ for all } \alpha$$

where $\theta_{j,c,\alpha}$ denotes the probability of feature α having the value j, given that the label is c. Here the constraints ensure that this is a probability distribution.

Parameter estimation:

$$\hat{\theta}_{j,c,\alpha} = \frac{|\{(x,y) \in \mathcal{D} \mid y = c \text{ and } x_\alpha = j\}| + l}{|\{(x,y) \in \mathcal{D} \mid y = c\}| + lK_\alpha} = \frac{\sum_{i=1}^n I(y_i = c) I(x_{i,\alpha} = j) + l}{\sum_{i=1}^n I(y_i = c) + lK_\alpha},$$

where $x_{i,\alpha}$ is the αth feature of the ith example in the training set $\mathcal{D}$ and l is a smoothing parameter. By setting $l = 0$ we get an MLE estimator, and $l > 0$ leads to MAP. Setting $l = +1$ is a technique called *Laplace smoothing*.

The generative model that we are assuming is that the data was generated by first choosing the label (e.g. *"healthy person"*). That label comes with a set of d "dice", one for each dimension. The generator picks each die, tosses it (independently) and fills in the feature value with the outcome of the coin toss. So if there are C possible labels and d dimensions we are estimating $d \times C$ "dice" from the data. However, per data point only d dice are tossed (one for each dimension). Die α (for any label) has K_α possible "sides". Of course this is not how the data is generated in reality — but it is a modeling assumption that we make.

Prediction: The prediction made by the categorical Naive bayes classifier is

$$h(\mathbf{x}) = \arg\max_{c \in \mathcal{Y}} \mathbf{P}(y = c \mid \mathbf{x}) = \arg\max_{c \in \mathcal{Y}} \hat{\pi}_c \prod_{\alpha=1}^d \hat{\theta}_{\mathbf{x}_\alpha,c,\alpha} \text{ where } \hat{\pi}_c = \frac{\sum_{i=1}^n I(y_i = c)}{n}.$$

Case #2: Multinomial Naive Bayes

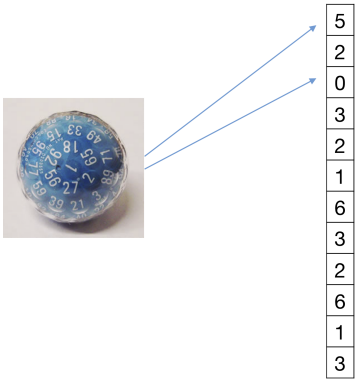


Illustration of multinomial NB. There are only as many dice as classes. Each die has d sides. The value of the αth feature shows how many times this particular side was rolled.

If feature values represent counts (not categories) we need to use a different model. E.g. in the text document categorization, feature value $x_\alpha = j$ means that in this particular document $\mathbf{x}$ the αth word in my dictionary appears j times. Let us consider the example of spam filtering. Imagine the αth word is indicative of being "spam". Then if $x_\alpha = 10$ means that this email is likely spam (as word α appears 10 times in it). And another email with $x'_\alpha = 20$ should be even more likely to be spam (as the spammy word appears twice as often). With categorical features this is not guaranteed. It could be that the training set does not contain any email that contain word α exactly 20 times. In this case you would simply get the hallucinated smoothing values for both spam and not-spam — and the signal is lost. We need a model that incorporates our knowledge that features are counts — this will help us during estimation (you don't have to see a training email with exactly the same number of word occurrences) and during inference/testing (as you will obtain these monotonicities that one might expect). The multinomial distribution does exactly that.

Feature assumption: the features lie in $x_\alpha \in \mathbb{N}$ under the constratint that $\sum_{\alpha=1}^d x_\alpha = m$ for some $m \in \mathbb{N}$, the total feature count. Usually, each feature x_α represents a count of the number of types something occurred in a sequence and m is the length of the sequence. An example of this could be the count of a specific word α in a document of length m, where d is the size of the vocabulary.

Model assumption: For multinomial Naive Bayes, we use the parameter estimates Use the multinomial distribution

$$\mathbf{P}(\mathbf{x} \mid m, y = c) = \frac{m!}{x_1! \cdot x_2! \cdot \cdots \cdot x_d!} \prod_{\alpha=1}^d (\theta_{\alpha c})^{x_\alpha} \text{ where } \theta_{\alpha c} \geq 0 \text{ and } \sum_{\alpha=1}^d \theta_{\alpha c} = 1$$

where $\theta_{\alpha c}$ represents the probability of selecting α in any particular element of the sequence, conditioned on the class being c. So, we can use this to generate a spam email, i.e., a document $\mathbf{x}$ of class $y = \text{spam}$ by picking m words independently at random from the vocabulary of d words using $\mathbf{P}(\mathbf{x} \mid y = \text{spam})$. Note that *this is not exactly satisfying the Naive Bayes assumption* on the features x_α. Rather, it corresponds to making the naive Bayes assumption on the members of the underlying sequence, where the features are occurrence counts for items within this sequence. To see that it doesn't satisfy the Naive Bayes assumption on the features, observe that if we know the counts for all but one of the features, we'd know the count for the last feature (since they must sum to m); this wouldn't be true if the features were independent.

Parameter estimation:

$$\hat{\theta}_{\alpha c} = \frac{\sum_{i=1}^n I(y_i = c) x_{i\alpha} + l}{\sum_{i=1}^n I(y_i = c) m_i + l \cdot d} \text{ where } m_i = \sum_{\beta=1}^d x_{i\beta} \qquad \left(\text{e.g. } \hat{\theta}_{\alpha c} = \frac{\#\text{ of times word } \alpha \text{ appears in all spam emails}}{\#\text{ of words in all spam emails combined}}\right).$$

Here, m_i denotes the number of words in document i. Again, l is the smoothing parameter. The numerator sums up all counts for feature x_α and the denominator sums up all counts of all features across all data points.

Prediction: The prediction made by the Multinomial Naive Bayes classifier is

$$h(x) = \arg\max_{c \in \mathcal{Y}} \mathbf{P}(y = c \mid \mathbf{x}) = \arg\max_{c \in \mathcal{Y}} \hat{\pi}_c \prod_{\alpha=1}^{d} \left(\hat{\theta}_{\alpha c}\right)^{x_\alpha}.$$

Case #3: Continuous features (Gaussian Naive Bayes)

Feature assumption: The features for Gaussian Naive Bayes are real numbers $x_\alpha \in \mathbb{R}$.

Model assumption: Use the Gaussian distribution

$$\mathbf{P}(x_\alpha \mid y = c) = \mathcal{N}\left(\mu_{\alpha c}, \sigma^2_{\alpha c}\right) = \frac{1}{\sqrt{2\pi}\sigma_{\alpha c}} e^{-\frac{1}{2}\left(\frac{x_\alpha - \mu_{\alpha c}}{\sigma_{\alpha c}}\right)^2}$$

where $\mu_{\alpha c}$ is the mean of the distribution and $\sigma^2_{\alpha c}$ is its variance. Note that the model specified above is based on our assumption about the data — that each feature x_α comes from a class-conditional Gaussian distribution. The full distribution of the whole feature vector is *also* Gaussian, with $\mathbf{P}(\mathbf{x}|y) \sim \mathcal{N}(\mu_y, \Sigma_y)$, where Σ_y is a diagonal covariance matrix with $[\Sigma_y]_{\alpha,\alpha} = \sigma^2_{\alpha,y}$.

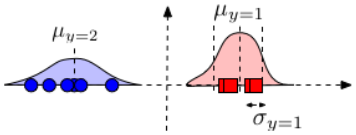


Illustration of Gaussian NB. Each class conditional feature distribution $\mathbf{P}(x_\alpha|y)$ is assumed to originate from an independent Gaussian distribution with its own mean $\mu_{\alpha,y}$ and variance $\sigma^2_{\alpha,y}$.

Parameter estimation: As always, we estimate the parameters of the distributions for each dimension and class independently. Gaussian distributions only have two parameters, the mean and variance. The mean $\mu_{\alpha,y}$ is estimated by the average feature value of dimension α from all samples with label y. The (squared) standard deviation is simply the variance of this estimate.

$$\mu_{\alpha c} = \frac{1}{n_c}\sum_{i=1}^{n} I(y_i = c) x_{i\alpha} \qquad \text{and} \qquad \sigma^2_{\alpha c} = \frac{1}{n_c}\sum_{i=1}^{n} I(y_i = c)(x_{i\alpha} - \mu_{\alpha c})^2 \qquad \text{where } n_c = \sum_{i=1}^{n} I(y_i = c)$$

What is the classification rule for Gaussian Naive Bayes? How much can you simplify the expression?

Naive Bayes is a linear classifier

1. Suppose that $y_i \in \{-1, +1\}$ and features are multinomial We can show that

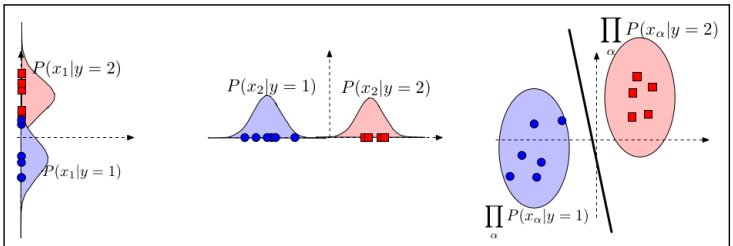
$$h(\mathbf{x}) = \underset{y}{\text{argmax}}\ \mathbf{P}(y) \prod_{\alpha-1}^{d} \mathbf{P}(x_\alpha \mid y) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b)$$

That is, $\mathbf{w}^\top \mathbf{x} + b > 0 \iff h(\mathbf{x}) = +1$. As before, we define $\mathbf{P}(x_\alpha|y = +1) \propto \theta_{\alpha+}^{x_\alpha}$ and $\mathbf{P}(y = +1) = \pi_+$:

$$[\mathbf{w}]_\alpha = \log(\theta_{\alpha+}) - \log(\theta_{\alpha-})$$
$$b = \log(\pi_+) - \log(\pi_-)$$

If we use the above to do classification, we can compute for $\mathbf{w}^\top \cdot \mathbf{x} + b$

Simplifying this further leads to



Naive Bayes leads to a linear decision boundary in many common cases. Illustrated here is the case where $\mathbf{P}(x_\alpha|y)$ is Gaussian and where $\sigma_{\alpha,c}$ is identical for all c (but can differ across dimensions α). The boundary of the ellipsoids indicate regions of equal probabilities $\mathbf{P}(\mathbf{x}|y)$. The red decision line indicates the decision boundary where $\mathbf{P}(y = 1|\mathbf{x}) = \mathbf{P}(y = 2|\mathbf{x})$.

$$\begin{aligned}
\mathbf{w}^\top \mathbf{x} + b > 0 &\iff \sum_{\alpha=1}^{d} [\mathbf{x}]_\alpha \overbrace{(\log(\theta_{\alpha+}) - \log(\theta_{\alpha-}))}^{[\mathbf{w}]_\alpha} + \overbrace{\log(\pi_+) - \log(\pi_-)}^{b} > 0 && \text{(Plugging in definition of } \mathbf{w}, b.\text{)} \\
&\iff \exp\left(\sum_{\alpha=1}^{d} [\mathbf{x}]_\alpha (\log(\theta_{\alpha+}) - \log(\theta_{\alpha-})) + \log(\pi_+) - \log(\pi_-)\right) > 1 && \text{(exponentiating both sides)} \\
&\iff \prod_{\alpha=1}^{d} \frac{\exp\left(\log \theta_{\alpha+}^{[\mathbf{x}]_\alpha} + \log(\pi_+)\right)}{\exp\left(\log \theta_{\alpha-}^{[\mathbf{x}]_\alpha} + \log(\pi_-)\right)} > 1 && \text{Because } a\log(b) = \log(b^a) \text{ and } \exp(a-b) = \frac{e^a}{e^b} \text{ operations} \\
&\iff \prod_{\alpha=1}^{d} \frac{\theta_{\alpha+}^{[\mathbf{x}]_\alpha} \pi_+}{\theta_{\alpha-}^{[\mathbf{x}]_\alpha} \pi_-} > 1 && \text{Because } \exp(\log(a)) = a \text{ and } e^{a+b} = e^a e^b \\
&\iff \frac{\prod_{\alpha=1}^{d} \mathbf{P}([\mathbf{x}]_\alpha|Y = +1)\pi_+}{\prod_{\alpha=1}^{d} \mathbf{P}([\mathbf{x}]_\alpha|Y = -1)\pi_-} > 1 && \text{Because } \mathbf{P}([\mathbf{x}]_\alpha|Y = -1) = \theta_{\alpha-}^{[\mathbf{x}]_\alpha} \\
&\iff \frac{\mathbf{P}(\mathbf{x}|Y = +1)\pi_+}{\mathbf{P}(\mathbf{x}|Y = -1)\pi_-} > 1 && \text{By the naive Bayes assumption.} \\
&\iff \frac{\mathbf{P}(Y = +1|\mathbf{x})}{\mathbf{P}(Y = -1|\mathbf{x})} > 1 && \text{By Bayes rule (the denominator } \mathbf{P}(\mathbf{x}) \text{ cancels out, and } \pi_+ = \mathbf{P}(Y = +1).\text{)} \\
&\iff \mathbf{P}(Y = +1|\mathbf{x}) > \mathbf{P}(Y = -1|\mathbf{x}) \\
&\iff \underset{y}{\text{argmax}}\, \mathbf{P}(Y = y|\mathbf{x}) = +1 && \text{i.e. the point } \mathbf{x} \text{ lies on the positive side iff Naive Bayes predicts } +1
\end{aligned}$$