› Course Shortcuts

› Student Lounge

› Q&A

› Estimating Distributions from Data

⌄ Classification with the Naive Bayes Algorithm

- Module Introduction: Classification with the Naive Bayes Algorithm
- Limitations of High Dimensional Space in MLE
- MLE Curse of Dimensionality
- Formalize the Curse of Dimensionality
- Capturing All Possibilities in d Dimensions
- Naive Bayes Assumption
- Data Sets Where the Naive Bayes Assumption Holds
- Naive Bayes Classifier
- Derivation of Naive Bayes Classifier
- Naive Bayes Cheat Sheet
- Determine Probability With Categorical Naive Bayes
- Categorical Naive Bayes Classifiers
- Naive Bayes in Action
- Module Wrap-up: Classification with the Naive Bayes Algorithm

› Building a Baby Name Classifying System

› Course Resources

Categorical Naive Bayes Classifiers

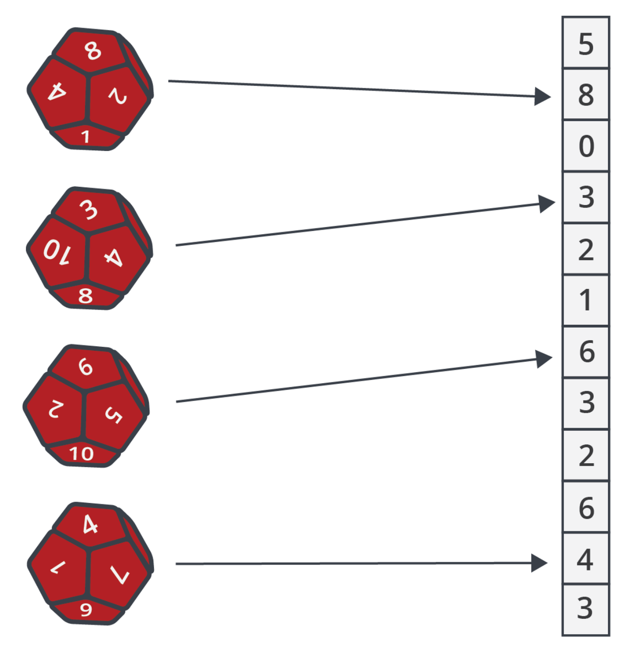
You've seen how to calculate $P(y|\mathbf{x})$ using the Naive Bayes assumption and Bayes' Rule to calculate $P(\mathbf{x}|y)$. However, depending on the types of features in your data, the way you calculate $P(\mathbf{x}|y)$ may differ. In this section, you will examine Naive Bayes when applied to categorical features, where each feature may fall into K_α categories.

Below is a visualization of how categorical Naive Bayes works. For d dimensional data, imagine there exist d independent dice that represent each feature. We assume training samples were generated by rolling one die after another, where there are K_α possible values for each roll. The value in dimension α corresponds to the outcome that was rolled with the α^{th} die.

☆ Key Points

For d dimensional data, imagine there exist d independent dice that represent each feature.

We assume training samples were generated by rolling one die after another, where there are K_α possible values for each roll.



Let's begin by defining categorical Naive Bayes features more formally:

$$[\mathbf{x}]_\alpha \in \{f_1, f_2, \cdots, f_{K_\alpha}\}$$

Per the expression above, the α^{th} feature is drawn from a set with K_α elements. Each feature α falls into one of K_α categories. (Note that binary features are just a specific case of this, where $K_\alpha = 2$.) For example, you might have medical data where one feature could be "Does the patient have hypertension?" and the answer is binary (yes=1, no=0). In that case, we would have $[\mathbf{x}]_\alpha \in \{0,1\}$.

Now we can begin to construct the model, $\mathrm{P}(x_\alpha|y)$:

$$\mathrm{P}(x_\alpha = j|y = c) = [\theta_{jc}]_\alpha$$

and

$$\sum_{j=1}^{K_\alpha} [\theta_{jc}]_\alpha = 1$$

In the equation above, $[\theta_{jc}]_\alpha$ is shortcut notation denoting the probability that feature α has the value j, given that the label is c . The second expression is simply a constraint that indicates that x_α must fall into one of the categories $\{1,\dots,K_\alpha\}$ i.e. the probabilities must sum to 1.

Now, we must estimate $[\theta_{jc}]_\alpha$ itself to complete our model:

$$[\hat{\theta}_{jc}]_\alpha = \frac{\sum_{i=1}^{n} I(y_i=c)I(x_{i\alpha}=j)+l}{\sum_{i=1}^{n} I(y_i=c)+lK_\alpha}$$

where $x_{i\alpha} = [\mathbf{x}_i]_\alpha$, l is a smoothing parameter, and I is an indicator function.

To train the Naive Bayes classifier, first you must estimate θ_{jc} for all j and c and store them in the respective conditional probability tables (CPT). Also note that you can set the smoothing parameter to different values in order to use different estimation techniques:

- $l = 0$ is maximum likelihood estimation (MLE)
- $l > 0$ is maximum a posteriori (MAP)
- $l = +1$ is Laplace ("plus one") smoothing

In other words, without the l hallucinated samples, this formula means the probability that feature α takes on value j given that the label is c is:

$$[\hat{\theta}_{jc}]_\alpha = \frac{\#\text{ of samples with label } c \text{ that have feature } \alpha \text{ with value } j}{\#\text{ of samples with label } c}$$

Essentially, the categorical feature model associates a special die with each feature and label. Data is generated by first choosing the label (e.g. "healthy person"), which comes with a set of d dice, one for each dimension. The generator rolls each die, and fills in the feature value with the outcome of the die roll. So if there are C possible labels and d dimensions we are estimating dC dice from the data, but per the example only d of them are rolled. 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 to approximate the real world.

We also need to estimate the probability of the class label independent of the labels: $\hat{\pi}_c = \mathrm{P}(y=c)$. This denotes the probability that a sample is of label c without knowing anything about its features (e.g. out of the whole population, how many people have a particular illness, and how many don't). As this is just a simple one dimensional estimation problem, we can use MLE and compute the fraction of samples with label c out of the total set.

Putting all this together, we can formulate our Naive Bayes predictions for categorical features using Bayes Rule:

$$\arg\max_y \mathrm{P}(y=c|\vec{x}) \propto \arg\max_y \mathrm{P}(y=c)\prod_{\alpha=1}^{d}\mathrm{P}(x_\alpha=j|y=c) = \arg\max_y \hat{\pi}_c \prod_{\alpha=1}^{d}[\hat{\theta}_{jc}]_\alpha$$