EC999: Naive Bayes Classifiers

Thiemo Fetzer

University of Warwick

February 6, 2017

Generative versus Discriminative Models

We need to obtain estimates of $\hat{P}(Y = y|X)$ for each value that y can take and then, following the Maximum A Posteriori Decision rule, which minimizes overall test error, assign a label such that

$$\hat{Y} = argmax_{y \in \mathcal{C}} \hat{P}(Y = y|X)$$

Bayes Rule says, that you can write $P(Y|X) = \frac{P(Y \cap X)}{P(X)} = \frac{P(X|Y)P(Y)}{P(X)}$

1. Logistic Regression and KNN are **discriminative model**, which directly computes $P(Y|X)$ ✓
2. Naive Bayes is a **generative model**, which computes $P(Y|X)$ indirectly, exploiting the factorization due to Bayes Theorem, i.e. $P(X|Y)P(Y)$

Naive Bayes

- We introduce the Naive Bayes classifier in the context of classification, our applications will focus on text features.
- This is a setting where it is most powerful, since were there are many features (i.e. X has many columns, many words to consider), but any given feature only has a small effect on $P(Y|X)$.
- We begin with a simple motivating example to illustrate Bayes rule.

Bayes Law allows us to rewrite the classification problem as:

$$\hat{Y} = argmax_{y \in \mathcal{C}} \hat{P}(Y = y|X) = argmax_{y \in \mathcal{C}} \frac{\hat{P}(X|y)\hat{P}(y)}{\hat{P}(X)}$$

Plan

Naive Bayes Classifier

The Classification Problem and Bayes Rule

- The classification problem is still the same, i.e.

$$\hat{Y} = argmax_{y \in \mathcal{C}} \hat{P}(Y = y|X) = argmax_{y \in \mathcal{C}} \frac{\hat{P}(X|y)\hat{P}(y)}{\hat{P}(X)}$$

- Note that the denominator does not change for different classes $\hat{P}(X)$ is constant for each value in $\mathcal{C}$, so we can just drop the denominator.

$$\hat{Y} = argmax_{y \in \mathcal{C}} \hat{P}(Y = y|X) = argmax_{y \in \mathcal{C}} \hat{P}(X|y)\hat{P}(y)$$

- In reality, we have many features in X.

The "Naive" Bayes Classifier

- Typically you have many features X, i.e. X has many columns
- Suppose you can write $X = (x_1, ..., x_p)$, then we can write:

$$\hat{Y} = argmax_{y \in \mathcal{C}} P(x_1, ..., x_p | y) P(y)$$

- Very difficult to estimate joint probability $\hat{P}(x_1, ..., x_p | y)$, so we assume

Assumption (*Naive Bayes Assumption*)

The distribution of features x_i, x_j within a class Y is independent from one another.

- The simplifying assumption allows us to write

$$P(x_1, ..., x_p | Y = y) = \prod_{i=1}^{p} P(x_i | y)$$

The "Naive" Bayes Classifier

- The Naive Bayes Classifier assigns lables such that

$$\hat{Y} = argmax_{y \in \mathcal{C}} P(y) \prod_{i=1}^{p} P(x_i|y)$$

- This is equivalent to

$$\hat{Y} = argmax_{y \in \mathcal{C}} log(P(y)) + \sum_{i=1}^{p} log(P(x_i|y))$$

- This is still a linear classifier: it uses a linear combination of the inputs to make a classification decision.

An Example "Naive" Bayes Classifier

- You are asked to build a predictive model, based on a set of features, whether a car is likely to be stolen.
- You intend to use this information to target resources towards policing.
- What are our features here? X has dimensions 9×3
- Note that all features are binary - so the *presence (or absence)* of a feature may tell you something about the underlying probability.

Example No.	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No

An Example "Naive" Bayes Classifier

- How would we classifiy a new observation $\mathbf{x_i}$ of a Red Domestic SUV?

$$\mathbf{x_i} = (Red, SUV, Domestic)$$

- The Naive Bayes assumption allows us to factorize the joint distribution as:

$$P(x_1, ..., x_p|y) = \prod_{i=1}^{p} P(x_i|y)$$

- I.e. we need to estimate for $y_i = Yes$:

$$P(Yes), P(Red|Yes), P(SUV|Yes), P(Domestic|Yes)$$

- Similarly, we need to estimate for $y_i = No$:

$$P(No), P(Red|No), P(SUV|No), and P(Domestic|No)$$

Estimating Parameters Using Training Data

Prior probability $P(Yes) = \frac{4}{9}$

Since all features are binary, the class conditional probabilities are easy to compute given the training data

Stolen?	Color	Type	Origin
Yes	$\hat{P}(Red \mid Yes)$	$\hat{P}(Sports \mid Yes)$	$\hat{P}(Domestic \mid Yes)$
No	$\hat{P}(Red \mid No)$	$\hat{P}(Sports \mid No)$	$\hat{P}(Domestic \mid No)$

We estimate these looking at the training data as simple ratios
$\hat{P}(Red \mid Yes) = \frac{\text{Number of stolen red cars}}{\text{Number of stolen cars}} = \frac{2}{4}$.
You can fill out this table as

Stolen?	Color	Type	Origin
Yes	$\frac{2}{4}$	$\frac{3}{4}$	$\frac{2}{4}$
No	$\frac{2}{5}$	$\frac{2}{5}$	$\frac{3}{5}$

Computing the Naive Bayes Scores

For our new data point $\mathbf{x_i} = (Red, SUV, Domestic)$, we need to compute

$$\hat{Y} = argmax_{y \in \mathcal{C}} P(Y) \prod_{i=1}^{p} P(x_i|y)$$

For $y_i = Yes$:

$$\hat{P}(Yes)\hat{P}(Red|Yes)\hat{P}(SUV|Yes)\hat{P}(Domestic|Yes) = \frac{4}{9}\frac{2}{4}(1-\frac{3}{4})\frac{2}{4} = 0.027$$

For $y_i = No$:

$$\hat{P}(No)\hat{P}(Red|No)\hat{P}(SUV|No)\hat{P}(Domestic|No) = \frac{5}{9}\frac{2}{5}(1-\frac{2}{5})\frac{3}{5} = 0.08$$

So we would classify this instance $\mathbf{x_i}$ as *not stolen*.
Why dont the probabilities add up to 1?

Evaluating the Naive Bayes assumption

- In general, we can not directly *test* the Naive Bayes assumption of class conditional independence of individual features.
- However, we can look for evidence in the population data on whether features appear as independent.
- How do we do that? The Naive Bayes assumption states that

$$P(Red, SUV|Yes) = P(Red|Yes)P(SUV|Yes)$$

- We can see whether this holds approximately in the training data. $P(Red, SUV|Yes) = \frac{\text{No. of Stolen Red SUVs}}{\text{No. of stolen}} = \frac{0}{4} = 0$
- Versus $P(Red|Yes)P(SUV|Yes) = \frac{2}{4} \times (1 - \frac{3}{4}) \neq 0$
- This is **NOT** a statistical test, but suggests that the Naive Bayes assumption does not hold.

Some implicit assumptions made

- We treated the individual features x_j as a sequence of *Bernoulli distributed* random variables.
- This highlights why Naive Bayes is called a **generative** model, since we model the underlying probability distributions of the individual features contained in the data matrix $\mathbf{X}$.
- We estimate $P(Red|Yes)$ using $\frac{\text{No. of Stolen Red}}{\text{No of stolen}}$, this is actually a *maximum likelihood estimator* for the population probability $P(Red|Yes)$ of a sequence of bernoulli distributed random varibales.
- Why? Suppose you have a sequence of iid coin tosses, the joint likelihood of observing such a sequence of length n, $\mathbf{x} = (x_1, ..., x_n)$, where $x_j = 1$ if head occures, can be written as:

$$\mathcal{L}(p) = \prod_{j=1}^{n} p^{x_j}(1-p)^{1-x_j}$$

Some implicit assumptions made

- Taking logs,

$$\log \mathcal{L}(p) = \sum_{j=1}^{n} x_j log(p) + (1 - x_j) log((1 - p))$$

- a maximum likelihood estimate of $\hat{p}$ is satisfies a FOC

$$\sum_j \frac{x_j}{p} - \frac{1 - x_j}{1 - p} = 0$$

- This is solved by $\hat{p} = \frac{\sum x_j}{n}$.
- So our intuitive choice for the estimator of the class conditional probabilities etc is actually theoretically well founded, but *only* if our features follow a bernoulli distribution.
- In reality, the p features in our data matrix $\mathbf{X}$ could come from different generating functions (i.e. some may be Bernoulli, Multinomial, Poisson, Normal, ...etc)

Naive Bayes is very powerful...

- Naive Bayes is a classification method that tends to be used for discretely distributed data, and mainly, for text - we present the Bernoulli and Multinomial language models in the next section.
- The next section introduces the idea of representing text as data for economists and political scientist to work with, and presents an example of a Naive Bayes classifier applied to text data.
- Most often Naive Bayes classifiers are used to work with text, such as spam filters, sentiment categorization, ... and many other use cases.
- ...