Introduction to Data Science

NAÏVE BAYES

BRIAN D'ALESSANDRO

NAÏVE BAYES

BAYES RULE

In classification we want to estimate: $P(y|x_1, x_2, \ldots, x_m)$

If we remember Bayes rule: $$posterior = \frac{prior\ x\ likelihood}{evidence}$$

So mathematically:

$$P(y|x_1, x_2, \ldots, x_m) = \frac{P(y)P(x_1, x_2, \ldots, x_m|y)}{P(x_1, x_2, \ldots, x_m)}$$

How many parameters must we estimate for the above model?

THE NAÏVE PART

We make one assumption that simplifies our estimation problem tremendously. This is that each feature is independent of each other, condition on *Y=y.* Mathematically, this translates to:

$$P(x_i|y, x_j) = P(x_i|y)$$

We now apply this conditional independence assumption to the joint distribution of X given Y=y.

$$P(x_1, x_2, \ldots, x_m|y) = P(x_1|y)P(x_2|y) \ldots P(x_m|y) = \prod_{i=1}^{m} P(x_i|y)$$

Why do we make this assumption?

CONSTRUCTING A CLASSIFIER

We start with the fact that $P(x_1, x_2, \ldots, x_m)$ is constant given the data (and also not dependent on the class value).

Which leads us to this relation for each value of *Y=y:*

$$P(y|x_1, x_2, \ldots, x_m) \propto P(y) \prod_{i=1}^{m} P(x_i|y)$$

We then add a decision rule, which is to choose the value of y that is the most probable. This leads to the following maximum a posteriori decision rule:

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}}\ P(y) \prod_{i=1}^{m} P(x_i|y)$$

NB AS A LINEAR MODEL

We can position NB as another type of linear model. The previous optimization problem can be expressed as:

$$\hat{y} = \mathbb{I}\left(\frac{P(y=1|x_1,x_2,\ldots,x_m)}{P(y=0|x_1,x_2,\ldots,x_m)} > 1\right)$$

And using Bayes rule again, the log of the quantity in the above indicator function can be written as:

$$f(x) = ln\,\frac{P(y=1)}{P(y=0)} + \sum_{i=1}^{m} ln\,\frac{P(x_i|y=1)}{P(x_i|y=0)} = \alpha + \beta \cdot x$$

Which makes our NB estimator:

$$\hat{y} = \mathbb{I}(f(x) > 0)$$

VARIATIONS OF NB

Multinomial Naïve Bayes

In MNB we assume each document is characterized by a set of words/tokens (here called x_i, and that each word/token has a weight (usually frequency of occurrence or tf-idf for the document).

The posterior distribution is given by $P(y|doc) \propto P(y)P(doc|y) \propto p(y) \prod_{x_i \in doc} P(x_i|y)^{freq_i}$

$P(x_i|y)$ is a smoothed maximum likelihood estimate, where N_{yi} is the total count of a word given y, N_y is the document count of all words given y, α is a smoothing parameter (usually=1), n is count of all distinct words.

$$P(x_i|y) = \frac{N_{yi}+\alpha}{N_y+\alpha n}$$

We can use this to define a decision rule:

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}}\ log[P(y)P(doc|y)] = \underset{y \in Y}{\operatorname{argmax}}\ [log(P(y)) + \sum_{x_i \in doc} w_i\ log(\frac{N_{yi}+\alpha}{N_y+\alpha n})]$$

With the weight w_i being either the frequency or the tf-idf score of the word/token in the given document.

References: http://scikit-learn.org/stable/modules/naive_bayes.html,
"Tackling the Poor Assumptions of Naive Bayes Text Classifiers", Rennie et al, ICML-2003

VARIATIONS OF NB

Bernoulli Naïve Bayes

In BNB, each word/token x_i is a binary indicator, where w_i indicates that x_i is in a given document. The posterior distribution is then given by:

$$P(y|doc) \propto P(y)P(doc|y) = p(y) \prod_i P(x_i|y)^{w_i} \, (1 - P(x_i|y))^{1-w_i}$$

In this scenario we explicitly account for the fact that a given x_i is not in a particular document, and $P(x_i|y)$ is again the smoothed maximum likelihood estimate that word x_i appears in a given document given y. I.e., # of documents containing x_i in the class over the # of documents in the class.

Our decision function is then.

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}} \, [log(P(y)) + \sum_i w_i \, log(P(x_i)) + (1 - w_i) \, log(1 - P(x_i))]$$

References: http://scikit-learn.org/stable/modules/naive_bayes.html,
"Naive Bayes and Text Classification I", Sebastian Raschka

VARIATIONS OF NB

Gaussian Naïve Bayes

GNB is the standard formulation for continuously valued features. Again we use the following decision rule:

$$\hat{y} = \underset{y \in Y}{\operatorname{argmax}}\ P(y) \prod_{i=1}^{m} P(x_i|y)$$

We make the assumption that X_i is distributed as a normal random variable:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_{iy}^2}}\ exp(\frac{-(x_i-\mu_{iy})^2}{\sigma_{iy}^2})$$

Where: $\mu_{iy} = E[X_i|y]$ $\sigma_{iy}^2 = E[(X_i - \mu_{iy})^2|y]$

References: http://www.cs.cmu.edu/~tom/mlbook/NBayesLogReg.pdf
Generative and Discriminative Classifiers: Naïve Bayes and Logistic Regression

NOTES/ADVANTAGES

Naïve Bayes fast and scalable

- estimates each parameter separately
- can handle sparse data and is easily parallelized
- No numeric optimization – just counting (easy in a query language)

Good classifier, bad as a generative model

- if Naïve assumption holds, NB is a Bayes optimal classifier
- Naïve assumption never holds, so P(Y|X) tends to be biased towards 0 or 1
- Nonetheless, works well under 0/1 loss

Sparsity Can Hurt You

- P(x|y) needs to be smoothed, usually with beta-prior on the binomial distribution (LaPlace smoothing)

Class Imbalance

- Binary decision rule works best when classes are roughly equal
- Will always pick dominant class in practice if high skew exists
- Can use posterior estimate of P(Y|X) (or log) to rank and then use ROC to define optimal cut-off point.