Naïve Bayes:

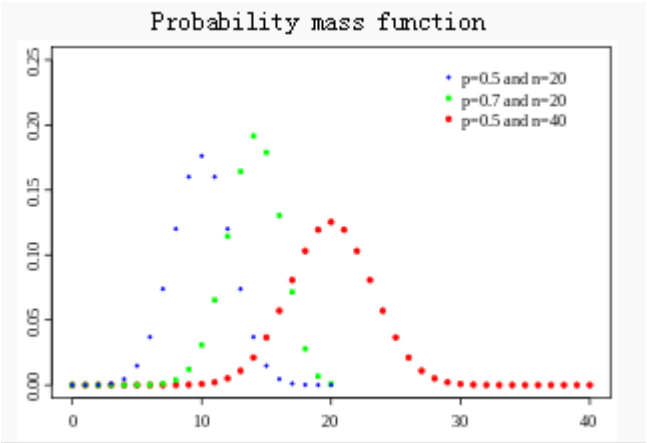
Random variable: is a variable whose value is subject to variations due to chance. A random variable can take on a set of possible different values (similarly to other mathematical variables), each with an associated probability.

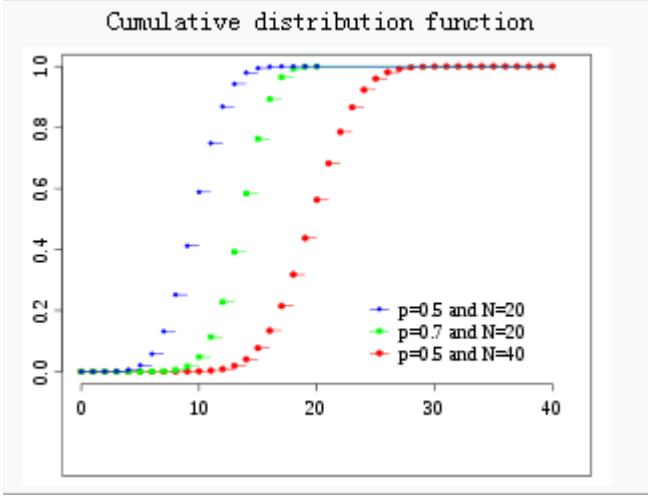
Bernoulli distribution: The Bernoulli distribution is a special case of the binomial distribution with n = 1.
Binomial distribution:

$$f(k;n,p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$$

for $k = 0, 1, 2, \ldots, n$, where

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$





Categorical distribution:
Multinomial distribution:

$$f(x_1,\ldots,x_k;n,p_1,\ldots,p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$

$$= \begin{cases} \dfrac{n!}{x_1!\cdots x_k!}p_1^{x_1}\cdots p_k^{x_k}, & \text{when } \sum_{i=1}^k x_i = n \\ 0 & \text{otherwise,} \end{cases}$$

Naïve Bayes classifiers: are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.
Naïve Bayes equation:

$$\hat{y} = \underset{k\in\{1,\ldots,K\}}{\operatorname{argmax}}\ p(C_k)\prod_{i=1}^{n} p(x_i|C_k).$$

Naïve Bayes assumption:
To estimate the parameters for a feature's distribution, one must assume a distribution or generate nonparametric models for the features from the training set. The assumptions on distributions of features are called the event model of the Naive Bayes classifier. For discrete features like the ones encountered in document classification (include spam filtering), multinomial and Bernoulli distributions are popular. These assumptions lead to two distinct models, which are often confused.
For each y, Naïve Bayes assumes x follow a specific distribution.
For instance, Gaussian Naïve Bayes
When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

which conforms with the definition of generative model. Generative model pay more attention to P(x|y).
我觉得这就符合了 generative model 的意义，这些针对不同的 y 的不同的分布就是对不同的 y 的描述。就是 generative model 里的 P(x|y)。

Naïve Bayes Assumption:

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

Each observed variable is assumed to be independent of each other variable given the class

Generative model vs discriminative (conditional) model:

A **generative algorithm** models how the data was generated in order to categorize a signal. It asks the question: based on my generation assumptions, which category is most likely to generate this signal?

A **discriminative algorithm** does not care about how the data was generated, it simply categorizes a given signal.

Let's say you have input data x and you want to classify the data into labels y. A generative model learns the **joint** probability distribution `p(x,y)` and a discriminative model learns the **conditional** probability distribution `p(y|x)` - which you should read as *"the probability of y given x"*.

Here's a really simple example. Suppose you have the following data in the form (x,y):

```
(1,0), (1,0), (2,0), (2, 1)
```

`p(x,y)` is

```
     y=0   y=1
    -----------
x=1 | 1/2   0
x=2 | 1/4   1/4
```

`p(y|x)` is

```
     y=0   y=1
    -----------
x=1 | 1     0
x=2 | 1/2   1/2
```

If you take a few minutes to stare at those two matrices, you will understand the difference between the two probability distributions.

The distribution `p(y|x)` is the natural distribution for classifying a given example `x` into a class `y`, which is why algorithms that model this directly are called discriminative algorithms. Generative algorithms model `p(x,y)`, which can be tranformed into `p(y|x)` by applying Bayes rule and then used for classification. However, the distribution `p(x,y)` can also be used for other purposes. For example you could use `p(x,y)` to *generate* likely `(x,y)` pairs.

From the description above you might be thinking that generative models are more generally useful and therefore better, but it's not as simple as that. This paper is a very popular reference on the subject of discriminative vs. generative classifiers, but it's pretty heavy going. The overall gist is that discriminative models generally outperform generative models in classification tasks.

http://stackoverflow.com/questions/879432/what-is-the-difference-between-a-generative-and-discriminative-algorithm

A good example:

> Imagine your task is to classify a speech to a language:
>
> you can do it either by:
>
> **1)** Learning each language and then classifying it using the knowledge you just gained
>
> OR
>
> **2)** Determining the difference in the linguistic models without learning the languages and then classifying the speech.
>
> the first one is the **Generative** Approach and the second one is the **Discriminative** approach.

Naïve Bayes likelihood:
Smoothing: Laplace smoothing

$$p(x_i|y_j) = \frac{c(x_i, y_j) + \alpha}{c(y_j) + \alpha s_i}$$

where $s_i = |Val(x_i)|$, $Val(x_i)$ denotes the range of the random variable x_i

Why we need smoothing? When there is no such data in training set, the probability of P(x|y) = 0. When we multiply it with other P(x|y), the final result would be 0.
We need sum of P(x|y) is still 1 after smoothing!

Deriving Maximum likelihood estimation for Naïve Bayes (why we use count):
It is the same as why we use count in hmm. Usually this issue can be explained by using MLE.

$$\frac{\partial L}{\partial \theta_1} = \sum_{k=1}^{m} \left(\frac{y^{(k)}}{\theta_1} + \frac{1 - y^{(k)}}{1 - \theta_1}(-1) \right) = 0$$

$$\theta_1 = \frac{1}{m} \sum_{k}^{m} y^{(k)} = \frac{\text{total number of class 1}}{\text{total number of instances}}$$

$$\frac{\partial L}{\partial \theta_{i,1}} = \sum_{k=1}^{m} \left(y^{(k)} \left(\frac{x_i^{(k)}}{\theta_{i,1}} + \frac{1 - x_i^{(k)}}{1 - \theta_{i,1}}(-1) \right) \right) = 0$$

$$\theta_{i,1} = \frac{\sum_k^m x_i^{(k)} y^{(k)}}{\sum_k^m y^{(k)}} = \frac{\text{total number of class 1 instances with } x_i = 1}{\text{total number of class 1 instances}}$$

Naïve Bayes Likelihood

- Likelihood: $p(D|\theta) = \prod_{k=1}^{|D|} p(y^{(k)}, x_1^{(k)}, .., x_n^{(k)})$
- Typically we will work with the log-likelihood

$$\begin{aligned}
\log p(D|\theta) &= \log \prod_{k=1}^{m} p(y^{(k)}, x_1^{(k)}, .., x_n^{(k)}) \\
&= \sum_{k=1}^{m} \log p(y^{(k)}, x_1^{(k)}, .., x_n^{(k)}) \\
&= \sum_{k=1}^{m} \log p(x_1^{(k)}, .., x_n^{(k)}|y^{(k)}) p(y^{(k)}) \\
&= \sum_{k=1}^{m} \log p(x_1^{(k)}, .., x_n^{(k)}|y^{(k)}) + \log p(y^{(k)}) \\
&= \sum_{k=1}^{m} \log \prod_{i=1}^{n} p(x_i^{(k)}|y^{(k)}) + \log p(y^{(k)}) \\
&= \sum_{k=1}^{m} \sum_{i=1}^{n} \log p(x_i^{(k)}|y^{(k)}) + \log p(y^{(k)})
\end{aligned}$$

Deriving ML Estimates for Naïve Bayes

Wlg, consider *binary* classification. Let θ_1 denote $P(y = 1)$; $\theta_{i,1}$ denote $p(x_i = 1|y = 1)$ and $\theta_{i,0}$ denote $p(x_i = 1|y = 0)$, then:

$$\begin{aligned}
\log p(D|\theta) &= \sum_{k=1}^{m} \sum_{i=1}^{n} \log p(x_i^{(k)}|y^{(k)}) + \log p(y^{(k)}) \\
&= \sum_{k=1}^{m} [y^{(k)} \log \theta_1 + (1 - y^{(k)}) \log(1 - \theta_1) \\
&+ \sum_{i=1}^{n} y^{(k)} \left(x_i^{(k)} \log \theta_{i,1} + (1 - x_i^{(k)}) \log(1 - \theta_{i,1}) \right) \\
&+ \sum_{i=1}^{n} (1 - y^{(k)}) \left(x_i^{(k)} \log \theta_{i,0} + (1 - x_i^{(k)}) \log(1 - \theta_{i,0}) \right)]
\end{aligned}$$

Document classification:

The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar

and even word order but keeping multiplicity. Recently, the bag-of-words model has also been used for computer vision.

Different types of Naïve Bayes:
(Bernoulli Naïve Bayes) multivariate Bernoulli event model:
最简单的例子只有两个 y， y=1, y=0，相当于一个 y 里面有 N 个 Bernoulli 分布，N=bag of words 的数量。
In each y, Bernoulli Naïve Bayes follows a multivariate Bernoulli distribution.

Multinomial Naïve Bayes:
最简单的例子只有两个 y，y=1, y=0，相当于一个 y 里面有 1 个 Multinomial 分布，这个 Multinomial 里面是(X1…XN, P1…PN)， N=bag of words 的数量。
In each y, Multinomial Naïve Bays follows a multinomial distribution.

Multinomial model does not take into account *negative evidence*

Bernoulli Naïve Bayes takes into account negative evidence!

所以 smooth 时候对 Bernoulli Bayes 分母只用加 2 就保证了概率相加为 1，而对于 Multinomial Bayes 需要分母加上 bag of words 的 range。

- Laplace smoothing

$$p(x_i|y_j) = \frac{c(x_i, y_j) + 1}{c(y_j) + 2}$$

- Generalized Laplace smoothing

$$p(x_i|y_j) = \frac{c(x_i, y_j) + \alpha}{c(y_j) + \alpha s_i}$$

where $s_i = |Val(x_i)|$, $Val(x_i)$ denotes the range of the random variable x_i

Gaussian Naïve Bayes for continuous data: When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution.

https://en.wikipedia.org/wiki/Naive_Bayes_classifier

TF-IDF (term frequency inverse document frequency):

The log frequency weight of term t in d is defined as follows

$$w_{t,d} = \begin{cases} 1 + \log_{10} \mathrm{tf}_{t,d} & \text{if } \mathrm{tf}_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$$

idf weight

- df_t is the document frequency, the number of documents that *t* occurs in.
- df_t is an inverse measure of the informativeness of term *t*.
- We define the idf weight of term t as follows:

$$\mathrm{idf}_t = \log_{10} \frac{N}{\mathrm{df}_t}$$

(*N* is the number of documents in the collection.)

- idf_t is a measure of the informativeness of the term.
- [log N/df_t] instead of [N/df_t] to "dampen" the effect of idf
- Note that we use the log transformation for both term frequency and document frequency.

tf-idf weighting

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = (1 + \log \mathrm{tf}_{t,d}) \cdot \log \frac{N}{\mathrm{df}_t}$$

- tf-weight
- idf-weight
- Best known weighting scheme in information retrieval
- Note: the "-" in tf-idf is a hyphen, not a minus sign!
- Alternative names: tf.idf, tf x idf

Generative vs Discriminative: Naïve Bayes vs Logistic regression

通常来说 discriminative model 效果更好，asymptotic error 更小，然而 generative model 接近 asymptotic error 速度更快
Training data 数据量越小用 generative model 更好？

$$p(\mathbf{x}|C_k) = \frac{(\sum_i x_i)!}{\prod_i x_i!} \prod_i {p_{ki}}^{x_i}$$

The multinomial naive Bayes classifier becomes a linear classifier when expressed in log-space:[2]

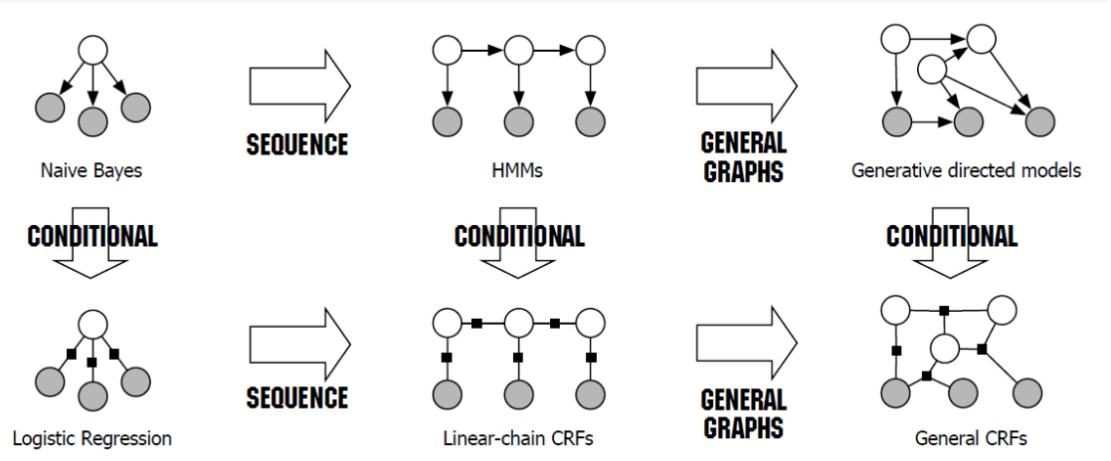
$$\begin{aligned}\log p(C_k|\mathbf{x}) &\propto \log\left(p(C_k)\prod_{i=1}^{n}{p_{ki}}^{x_i}\right)\\ &= \log p(C_k) + \sum_{i=1}^{n} x_i \cdot \log p_{ki}\\ &= b + \mathbf{w}_k^\top \mathbf{x}\end{aligned}$$

where $b = \log p(C_k)$ and $w_{ki} = \log p_{ki}$.

这样看起来 bayes 和 regression 一样，他的 weight 是相对应的 P(x|y)。

A *joint* model gives probabilities P(*d*,*c*) and tries to maximize this joint likelihood.

- It turns out to be trivial to choose weights: just relative frequencies.



For discriminative model, we model conditional probability, P(Y|X) directly, while for generative model we need to model joint probability, P(Y, X).
In discriminative model we can see the boundary clearly, while in generative model we can't.
Discriminative vs generative:
http://stats.stackexchange.com/questions/12421/generative-vs-discriminative
Logistic regression vs Naïve Bayes
http://www.quora.com/What-is-the-difference-between-logistic-regression-and-Naive-Bayes
假如出现 correlated feature 因为 bayes 里假设了相互无关，这就出现了很大问题。
A brief Maxent Tutorial