# Lecture 10:

# Naïve Bayes

## Week of February 13, 2023


University California, Berkeley

Machine Learning Algorithms

MSSE 277B, 3 Units

Spring 2023

Prof. Teresa Head-Gordon

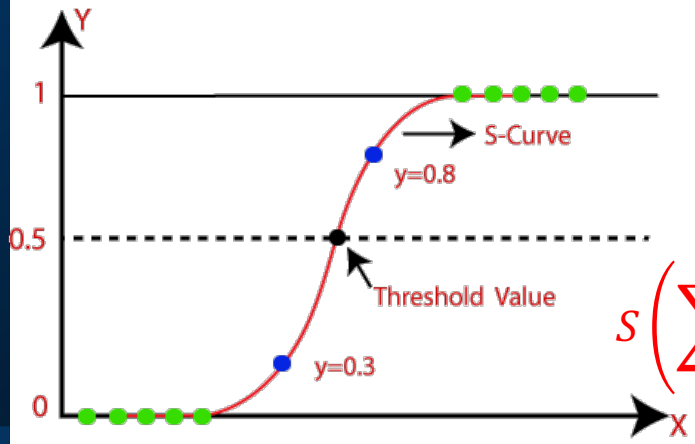
Departments of Chemistry, Bioengineering, Chemical and Biomolecular Engineering

# Summary Previous Lecture

Logistic regression is a discriminative classification method, which models the posterior probability $P(y_k|x_1, \ldots x_p)$ directly, i.e. the most probable classification given the data.

For binary classification we defined a likelihood function,

$$L(\{w\}|y, x) = \prod_{\mu}^{N} \left[ S\left( \sum w_j x_{j\mu} \right) \right]^{y_\mu} \left[ 1 - S\left( \sum w_j x_{j\mu} \right) \right]^{1-y_\mu}$$

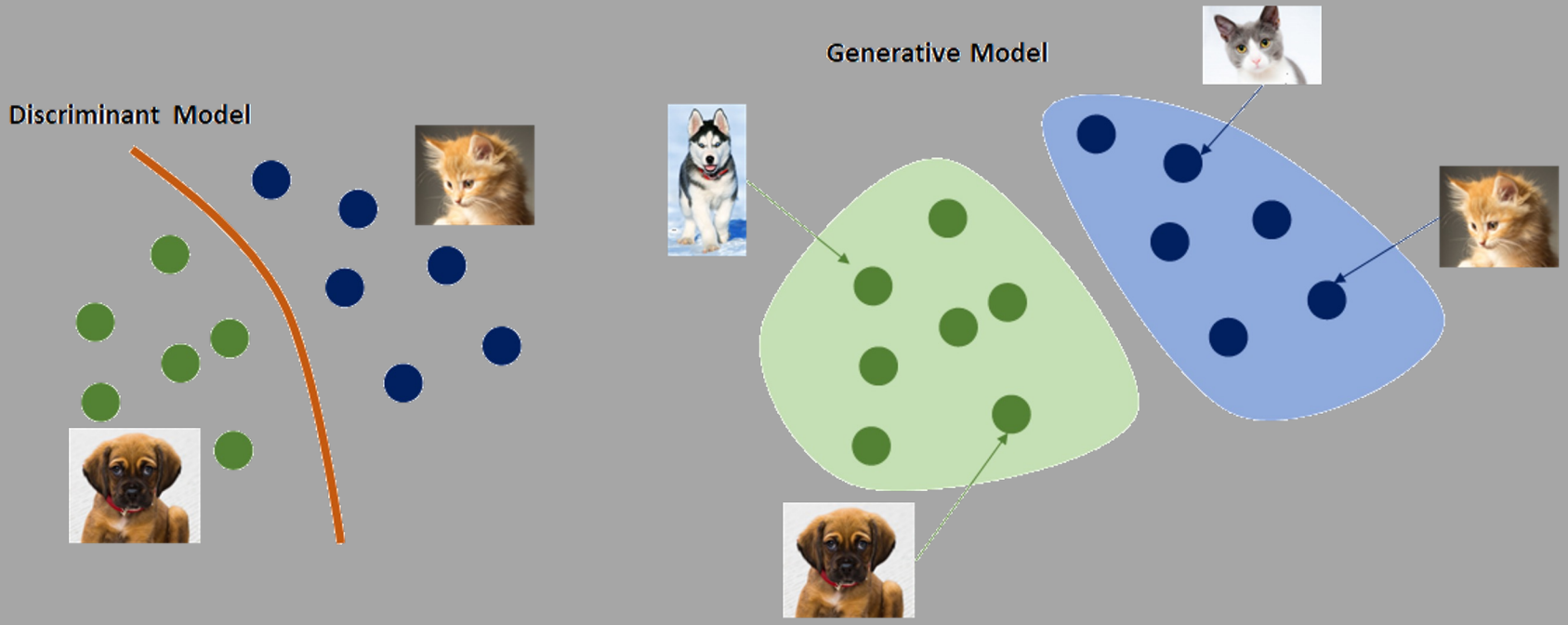



$$S\left( \sum w_j x_{j\mu} \right) = \frac{1}{1 + exp\left(-\sum w_j x_{j\mu}\right)}$$

which we maximize (but in practice must solve numerically) to determine the regression coefficients {w}

A simple perceptron can also do logistic regression through back-propagation of weights and using a nonlinear activation function that classifies output. Multilayer ANN's with combinations of linear and non-linear activation functions can do non-linear regression

# Discriminative vs. Generative Statistical Models




Purpose of Today's Lecture: There are a number of statistical models that can do classification. These get classified as Discriminative vs. Generative models

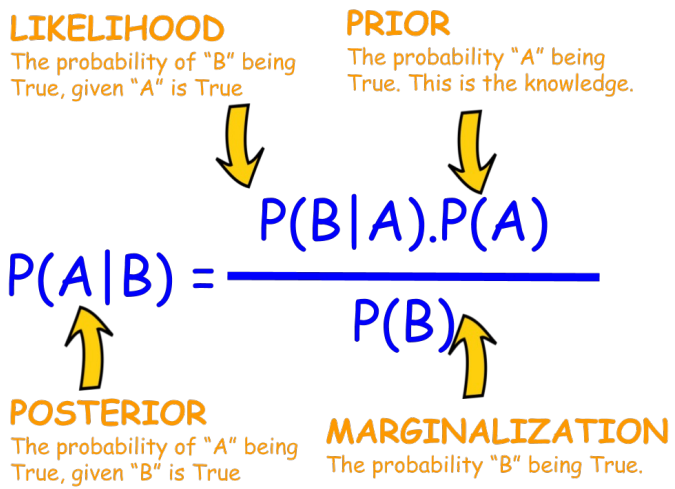
A discriminative model (like logistic regression) directly models the posterior probability of $P(y_k|x_1, \ldots x_p)$, i.e. by learning the input $\{x\}$ to classification mapping $y_k$ by minimizing the maximum likelihood error.

A generative model first considers the joint distribution of the feature $\{x\}$ and target $y_k$, $P(x_1, \ldots x_p|y_k)$, and then *predicts* the posterior probability $P(y_k|x_1, \ldots x_p)$ (by picking its maximum value) by using Baye's Rule

# Bayes Rule

But we first have to introduce Bayes rule!

$$P(y_k|x_1, \dots x_p) = \frac{P(x_1, \dots x_p|y_k)P(y_k)}{P(x_1, \dots x_p)}$$




where

- $P(y_k|x_1, \dots x_p)$ is a posterior probability of a class $y_k$ given the features $\{x_1, \dots x_p\}$. This is what we modeled in logistic regression

- $P(x_1, \dots x_p|y_k)$ is likelihood of the features given the class, k. This is what we will model in Naïve Bayes!

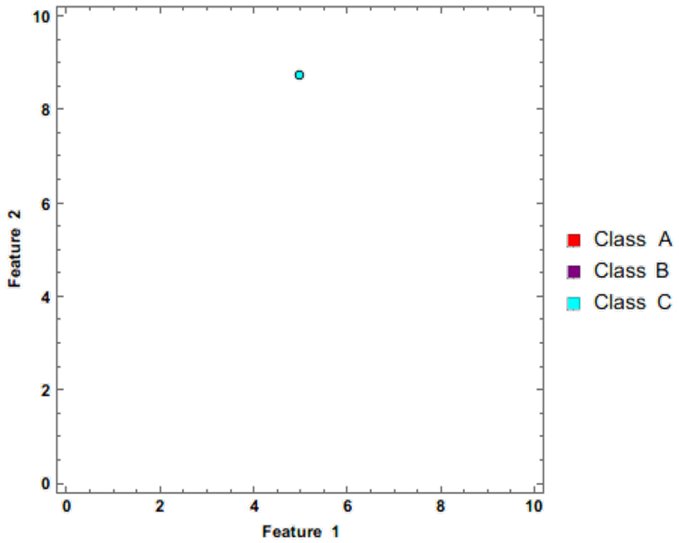
- $P(y_k)$ and $P(x_1, \dots x_p)$ are the probabilities of observing the class and features independently of each other; these are known as the marginal probabilities or priors

- $P(y_k)$ is simple to calculate; it is just the proportion of the data-set that falls in the $k$ class.

- $P(x_1, \ldots x_p|y_k)$ is more difficult to compute. In order to simplify its computation, we make the assumption that $x_1, \ldots x_p$ are independent given $y_k$

$$P(x_1, \ldots x_p|y_k) = P(x_1|y_k) * P(x_2|y_k) * * * P(x_p|y_k).$$

This assumption is the "naïve" Bayes classifier.

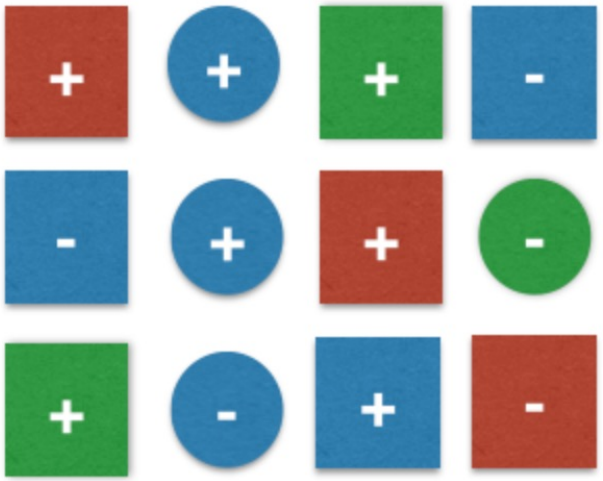



Calculating individual $P(x_j|y_k)$ terms will depend on what distribution your features follow; how good the resulting model is whether descriptors are independent!

# Naïve Bayes

The key ingredient of a generative model like Naïve Bayes is to now model each $P(x_j|y_k)$.

# Naïve Bayes Training




Bayes' Rule:

$$P(y_k|x_1, \dots x_p) \propto P(x_1, \dots x_p|y_k)P(y_k)$$

$$y_k \in (+, -)$$
$$x_1 \in (blue, green, red, yellow)$$
$$x_2 \in (square, circle)$$

Easy to get $P(y_1) = \frac{7}{12}$, $P(y_2) = \frac{5}{12}$

Not easy to get $P(x_1, \dots x_p|y_k)$ from the data unless we make assumptions/

But for Naïve Bayes of conditional independence, it is easy

$$P(blue, square\ |y_1) = P(blue\ |y_1)\ P(square\ |y_1) = \frac{3}{7} * \frac{5}{7}$$

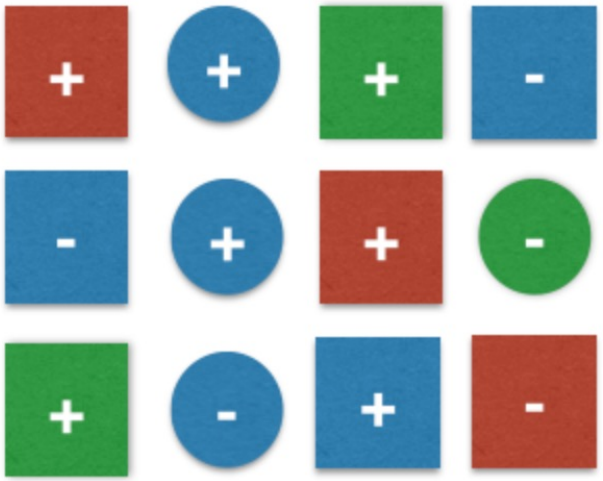
$$P(blue, square\ |y_2) = P(blue\ |y_2)\ P(square\ |y_2) = \frac{3}{5} * \frac{3}{5}$$

$$P(y_1|blue, square) = \left(\frac{3}{7} * \frac{5}{7}\right) * \frac{7}{12} = 0.18$$

$$P(y_2|blue, square) = \left(\frac{3}{5} * \frac{3}{5}\right) * \frac{5}{12} = 0.15$$

There is little explicit training in Naive Bayes compared to ANN since individual PDFs can be done quickly and deterministically (i.e. just frequency in this case)

# Naïve Bayes Classification




Naive Bayes produces classification in a very simple manner; simply pick the classification with the largest probability given the data point's features (blue,square).

This is referred to as the Maximum *A Posteriori* decision rule. This is because for Bayes rule, we require both P(B|A) and P(A) terms, which are the likelihood and prior terms, respectively, to define posterior probability

$$\boldsymbol{P(y_2|blue, square) = \left(\frac{3}{5} * \frac{3}{5}\right) * \frac{5}{12} = 0.15}$$

$$\boldsymbol{P(y_1|blue, square) = \left(\frac{3}{7} * \frac{5}{7}\right) * \frac{7}{12} = 0.18}$$

The Maximum Likelihood Error and Maximum *A Posteriori* differ in the inclusion of the prior

$$P(y_k|x_1, \dots x_p) \propto P(x_1, \dots x_p|y_k)P(y_k)$$

I.e. the likelihood is now weighted due to some model of the prior.

If we use a uniform prior we have MLE.

# Lecture 10 (Part 2):

## Naïve Bayes

## Week of February 14, 2022


University California, Berkeley

Machine Learning Algorithms

MSSE 277B, 3 Units

Spring 2022

Prof. Teresa Head-Gordon

Departments of Chemistry, Bioengineering, Chemical and Biomolecular Engineering

# Summary Previous Lecture

Logistic regression is a discriminative classification method, which models the posterior probability $P(y_k|x_1, \ldots x_p)$ directly, i.e. the most probable classification given the data.

We demonstrated this with a binary Bernoulli likelihood function, which we maximize (but in practice must solve numerically) to determine the regression coefficients

A simple perceptron can also do logistic regression through back-propagation of weights and using a nonlinear activation function that classifies output. Multilayer ANN's with combinations of linear and non-linear activation functions can do non-linear regression

(Naïve) Bayes is a generative classification model that instead models $P(x_1, \ldots x_p|y_k)$ as the likelihood function from the the {y,x} data. It uses Baye's theorem to get posterior probability

$$P(y_k|x_1, \ldots x_p) = \frac{P(x_1, \ldots x_p|y_k)P(y_k)}{P(x_1, \ldots x_p)}$$

*under the assumption* that the likelihood function has uncorrelated inputs

$$P(x_1, \ldots x_p|y_k) = P(x_1|y_k) * P(x_2|y_k) * * * P(x_p|y_k)$$

# Naïve Bayes and other Generative Models

Purpose of Today's Lecture: Naïve Bayes is an alternative and easily solvable model for the posterior probability using maximum a posteriori. It can choose different PDFs (we showed multinomial in previous Lecture) for the likelihood function of the uncorrelated PDFs

Bernoulli Naive Bayes: This is similar to the multinomial naive Bayes but the predictors are boolean variables. The parameters that we use to predict the class variable take up only values yes-no, right-left etc.

Gaussian Naive Bayes: When the predictors take up a continuous value and are not discrete, we assume that these values are sampled from a gaussian distribution.

Maximum Entropy: Now we are back to discriminative models. Naïve Bayes works under the assumption that the likelihood function has uncorrelated features. By lifting this assumption we can get to a new cost function - the maximum entropy!

# Naïve Bayes

Assuming conditional independence of the explanatory variables or features, then we get Naïve Bayes for arbitrary PDF

$$P(y_k|x_1, \ldots x_p) = \frac{P(y_k)\prod_j P(x_j|y_k)}{\sum_i [P(y_i)\prod_j P(x_j|y_i)]}$$

And classification rule for $\{x_i\}$ is maximum *a posteriori* (ignore denominator)

$$y_k \rightarrow \max_{y_k} P(y_k)\prod_j P(x_j|y_k)$$

Assuming a model PDF, there are parameters of that model that can be maximized given the labeled data {x,y}. I.e. determine $\theta$ that maximizes the likelihood, or log likelihood

$$\max_\theta \mathrm{L}(y_k|x_j,\theta) \rightarrow \max_\theta \; P(y_k|\theta)\prod_j P(x_j|\theta, y_k)$$

$$\max_\theta \log\mathrm{L}(y_k|x_j,\theta) \rightarrow \max_\theta \; \log P(y_k|\theta) + \sum_j \log P(x_j|\theta, y_k)$$

Lets assume the prior $P(y_k|\theta)$ is just the frequency of class (like multinomial), and thus focus on the PDF models for $P(x_j|\theta, y_k) \equiv P(x_{jk}|\theta)$ next

# Bernoulli Naïve Bayes

Bernoulli Naive Bayes only takes binary values of features, i.e. x={0,1}. Take the prediction of analysis of words in a text. In cases where counting the word frequency (like our previous multinomial example) is less important, Bernoulli checks whether or not a word appears in a document at all.

$$p(x|y) = \theta^x(1-\theta)^{1-x}$$

The Bernoulli algorithm explicitly penalizes the non-occurrence of a feature

| Features | | | Outcome |
|---|---|---|---|
| pH>7 | Water soluble | low MW | Drug Bioavailable |
| Yes | No | No | Fail |
| Yes | No | Yes | Hit |
| No | Yes | Yes | Fail |
| No | Yes | No | Hit |
| Yes | Yes | Yes | Hit |

P(y): P(Hit)=3/5; P(Fail)=2/5

P(pH>7|Hit)=2/3; P(pH>7|Fail)=1/3
P(pH<7|Hit)=1/2; P(pH>7|Fail)=1/2
P(water soluble|Hit)=2/3; P(water soluble|Fail)=1/3
P(water insoluble|Hit)=1/2; P(water insoluble|Fail)=1/2
P(lowMW|Hit)=2/3; P(lowMW|Fail)=1/3
P(highMW|Hit)=1/2; P(highMW|Fail)=1/2

$$p(hit|X) = p(X\{pH > 7, watersoluble, low\ MW\}\,|\,y = hit)p(Y) = \left(\frac{2}{3} * \frac{2}{3} * \frac{2}{3}\right) * \frac{3}{5}$$

$$p(fail|X) = p(X\{pH > 7, watersoluble, low\ MW\}\,|\,y = fail)p(Y) = \left(\frac{1}{3} * \frac{1}{3} * \frac{1}{3}\right) * \frac{2}{5}$$

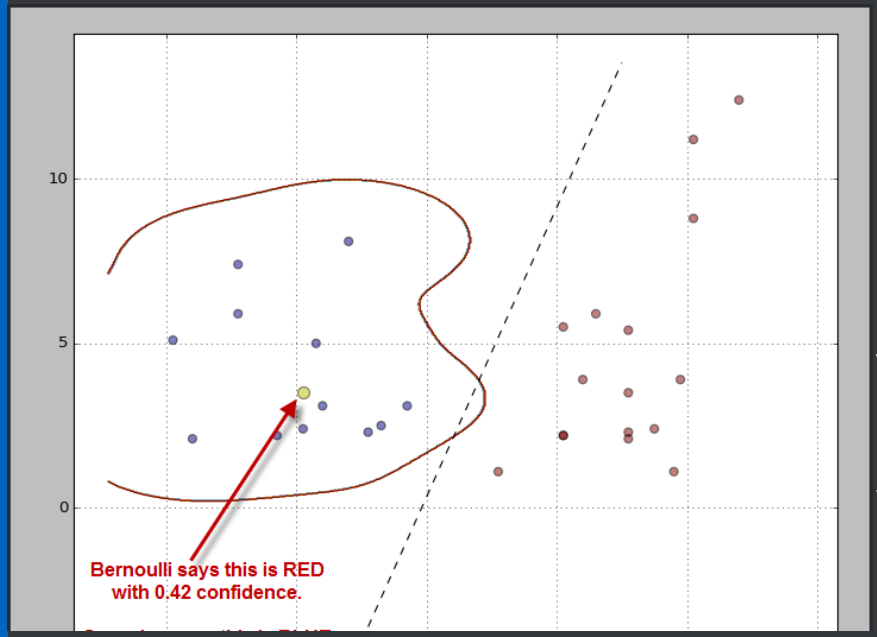
# Bernoulli Naïve Bayes

Find Bernoulli PDF parameters $\theta$ that best fit the data (index $\mu$) for each $j$ likelihood function

$$\mathrm{L}(x_{jk}\,|\theta) = \prod_{\mu} \theta^{x_j^{\mu}} (1-\theta)^{1-x_j^{\mu}}$$

Each explanatory variable $x_j$ is assumed to be binary-valued, and data has N examples for each $j$ feature in category $k$

$$\mathrm{L}(x_{jk}\,|\theta) = \theta^{x_j^1}(1-\theta)^{1-x_j^1} \ldots \theta^{x_j^N}(1-\theta)^{1-x_j^N}$$




rewrite above

$$\mathrm{L}(x_{jk}|\theta) = \theta^{\sum_{\mu}^{N} x_j^{\mu}} (1-\theta)^{1-\sum_{\mu}^{N} x_j^{\mu}}$$

$$\mathrm{L}(x_{jk}|\theta) = \theta^{m}(1-\theta)^{N-m}$$

Where $m$ is just the sum of $x$=1 in category $k$, and $N$ is the number of training data examples, where $\mathrm{N}-m$ are the $x$=0. Given the log likelihood

$$\log\mathrm{L}(x_{jk}|\theta) = m\log\theta + (N-m)\log(1-\theta)$$

# Bernoulli Naïve Bayes

$$\log \mathrm{L}(x_{jk}|\theta) = m \log \theta + (N - m) \log(1 - \theta)$$

Lets determine $\theta$ that maximizes the log likelihood

$$\frac{\partial log \mathrm{L}(x_{jk}|\theta)}{\partial \theta} = 0$$

$$\frac{m}{\theta} - \frac{(N - m)}{(1 - \theta)} = 0$$

$$\theta = \frac{m}{N}$$

This is simply the fraction of examples, for a given class, that contain the particular feature. This is what we did in our chemical example

Note that this differs from multinomial Naïve Baye's rule in that the multinomial variant would simply ignore a non-occurring feature. I.e. if it has never seen that feature – it won't be able to classify

Because it explicitly penalizes the non-occurrence of a feature that is an indicator for class, this is a distinct advantage of Bernoulli Naïve Bayes.

Still suffers from assumption of independence among features

# Gaussian Naïve Bayes

Gaussian Naive Bayes is a variant of Naive Bayes that follows Gaussian normal distribution and supports continuous data.

A Gaussian PDF (same as linear regression), but now we are modeling each $P(x_{jk}|\theta)$

$$\mathrm{L}(x_{jk}|\theta = m_{jk}, \sigma_{jk}) = \prod_{\mu} \frac{1}{\sqrt{2\pi\sigma_{jk}^2}} exp\left(-\frac{\left(x_{jk}^{\mu} - m_{jk}\right)^2}{2\sigma_{jk}^2}\right)$$

Now each explanatory variable $x_{jk}$ is assumed to be continuous, and the model PDF parameters are mean and variance for that feature $j$,category $k$.

We determine $m_{jk}$ and $\sigma_{jk}$ from training data using maximum log likelihood

$$\log\mathrm{L}(x_{jk}|\theta = m_{jk}, \sigma_{jk}) = \sum_{\mu}^{N} -log\sqrt{2\pi\sigma_{jk}^2} - \frac{\left(x_{jk}^{\mu} - m_{jk}\right)^2}{2\sigma_{jk}^2}$$

$$\log L\left(x_{jk}\middle|\theta = m_{jk}, \sigma_{jk}\right) = \sum_{\mu}^{N} -log\sqrt{2\pi\sigma_{jk}^2} - \frac{\left(x_{jk}^{\mu} - m_{jk}\right)^2}{2\sigma_{jk}^2}$$

PDF parameter mean $m_{jk}$

$$\frac{\partial \log L\left(x_{jk}\middle|m_{jk}\right)}{\partial m_{jk}} = 0 = \sum_{\mu}^{N} \frac{2\left(x_{jk}^{\mu} - m_{jk}\right)}{2\sigma_{jk}^2}$$

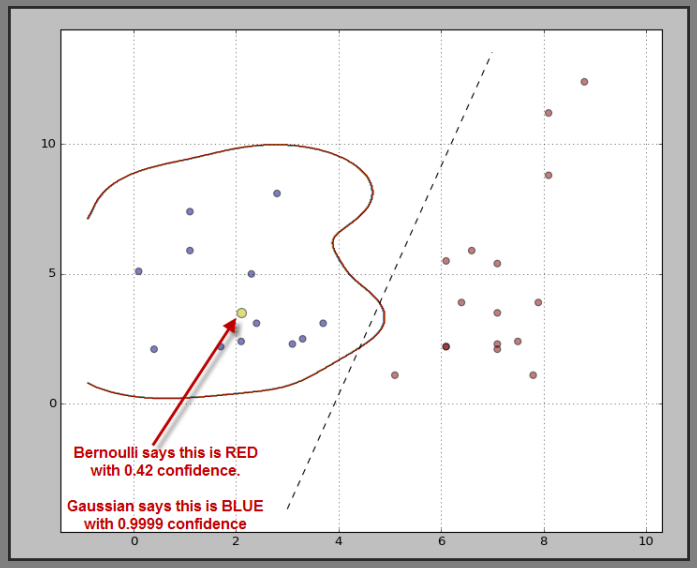
$$m_{jk} = \frac{1}{N}\sum_{\mu}^{N} x_{jk}^{\mu}$$

PDF parameter variance; let $\sigma^2 = z$

$$\frac{\partial \log L\left(x_{jk}\middle|z_{jk}\right)}{\partial z} = \sum_{\mu}^{N} -\frac{1}{z_{jk}} + \frac{2\left(x_{jk}^{\mu} - m_{jk}\right)^2}{2{z_{jk}}^3} = 0$$

$${z_{jk}}^2 = \frac{1}{N}\sum_{\mu}^{N}\left(x_{jk}^{\mu} - m_{jk}\right)^2$$

Gaussian Naïve Bayes is for continuous data. It makes stronger modeling assumption than logistic regression because of feature independence; if true, is best model for large N.




# Gaussian Naïve Bayes

Logistic regression is less sensitive to incorrect modeling assumptions, and if features correlated, Logistic regression beats Gaussian Naïve Bayes

Given 2 features weight and height. What is the probability for a given $(h_j, w_j)$ feature that is an adult or child?

$$P(h_j|c) = \frac{1}{\sqrt{2\pi\sigma_{hc}^2}} exp\left(-\frac{(h_j - m_{hc})^2}{2\sigma_{hc}^2}\right)$$

$$P(w_j|c) = \frac{1}{\sqrt{2\pi\sigma_{wc}^2}} exp\left(-\frac{(w_j - m_{wc})^2}{2\sigma_{wc}^2}\right)$$

$$P(h_j|a) = \frac{1}{\sqrt{2\pi\sigma_{ha}^2}} exp\left(-\frac{(h_j - m_{ha})^2}{2\sigma_{ha}^2}\right)$$

$$P(w_j|a) = \frac{1}{\sqrt{2\pi\sigma_{wa}^2}} exp\left(-\frac{(w_j - m_{wa})^2}{2\sigma_{wa}^2}\right)$$

$$P(X|c) = P(h_j|c)P(w_j|c)$$

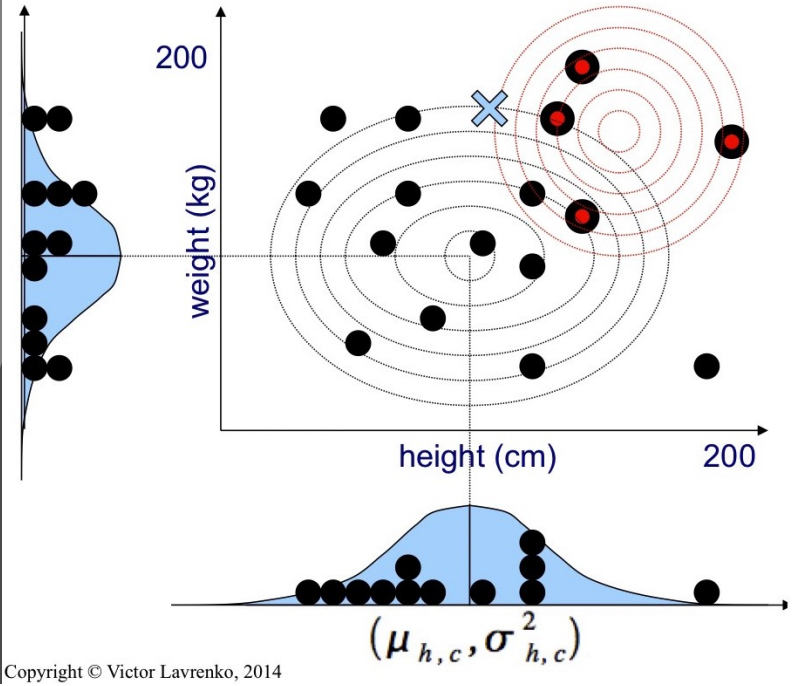
Baye's

$$P(c|X) = P(X|c)P(c)$$

with normalization to define probability

$$P(c|X) = \frac{P(X|c)P(c)}{P(X|c)P(c) + P(X|a)P(a)}$$

# Gaussian Naïve Bayes



Given 2 features weight and height. What is the probability for a given $(h_j, w_j)$ feature that is an adult or child?

$$P(h_j|c) = \frac{1}{\sqrt{2\pi\sigma_{hc}^2}} exp\left(-\frac{(h_j - m_{hc})^2}{2\sigma_{hc}^2}\right)$$

$$P(w_j|c) = \frac{1}{\sqrt{2\pi\sigma_{wc}^2}} exp\left(-\frac{(w_j - m_{wc})^2}{2\sigma_{wc}^2}\right)$$

$$P(h_j|a) = \frac{1}{\sqrt{2\pi\sigma_{ha}^2}} exp\left(-\frac{(h_j - m_{ha})^2}{2\sigma_{ha}^2}\right)$$

$$P(w_j|a) = \frac{1}{\sqrt{2\pi\sigma_{wa}^2}} exp\left(-\frac{(w_j - m_{wa})^2}{2\sigma_{wa}^2}\right)$$

$$P(X|a) = P(h_j|a)P(w_j|a)$$

Baye's

$$P(a|X) = P(X|a)P(a)$$

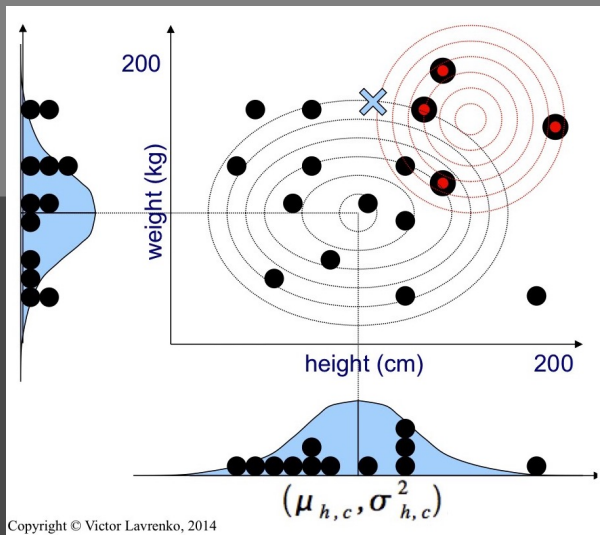
Maximum a postieori

$$P(a|X) > P(c|X)?$$

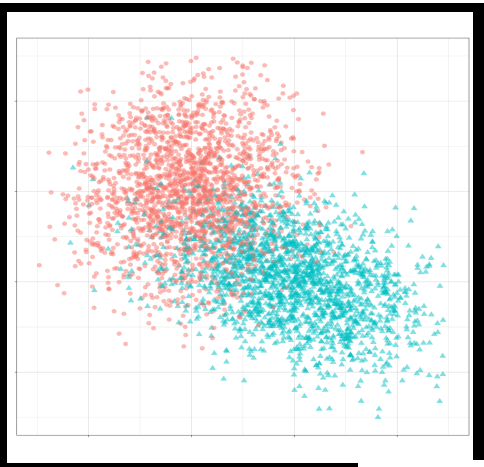
with normalization to define probability

$$P(a|X) = \frac{P(X|a)P(a)}{P(X|c)P(c) + P(X|a)P(a)}$$

# Gaussian Naïve Bayes

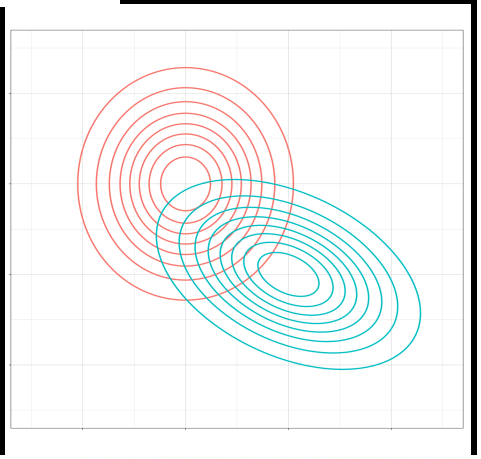


# Gaussian Naïve Bayes



Then classification rule given $\{x_{new}\}$ is *maximum a posteriori* to get

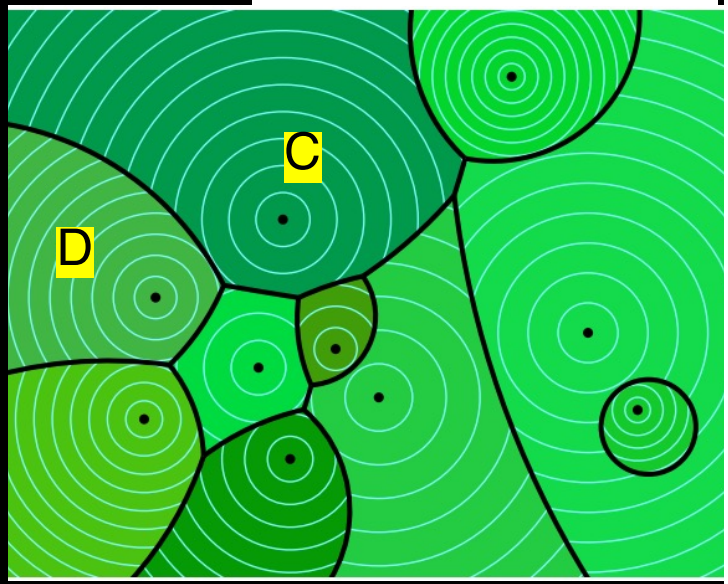
$$y_{new} \propto \max_{y_k} \sum_{\mu}^{N} -\frac{\left(x_{j,new}^{\mu} - m_j\right)^2}{2\sigma_j^2}$$



Suppose there are only 2 classes C, D; the optimal decision boundary is formed where the contours of the class-conditional densities intersect – because this is where the classes' discriminant functions are equal that determine the shape of these contours.

The decision boundary is quadratic in x if $\sigma_C^2 \neq \sigma_D^2$

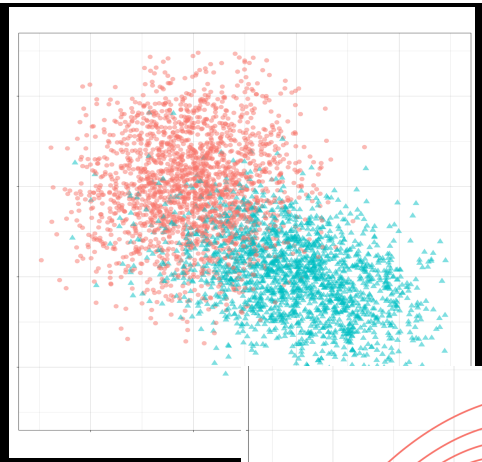
$$y_C - y_D = -\frac{\left(x_j^{\mu} - m_{jC}\right)^2}{2\sigma_{jC}^2} + \frac{\left(x_j^{\mu} - m_{jD}\right)^2}{2\sigma_{jD}^2} > 0$$

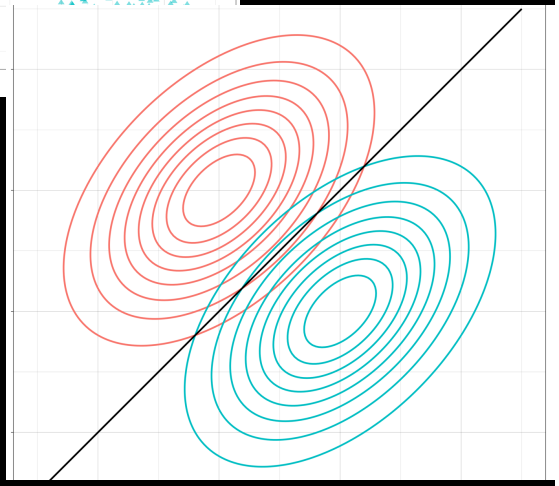


# Gaussian Naïve Bayes

Then classification rule given $\{x_{new}\}$ is *maximum a posteriori* to get

$$y_{new} \propto \max_{y_k} \sum_{\mu}^{N} - \frac{\left(x_{j,new}^{\mu} - m_j\right)^2}{2\sigma_j^2}$$

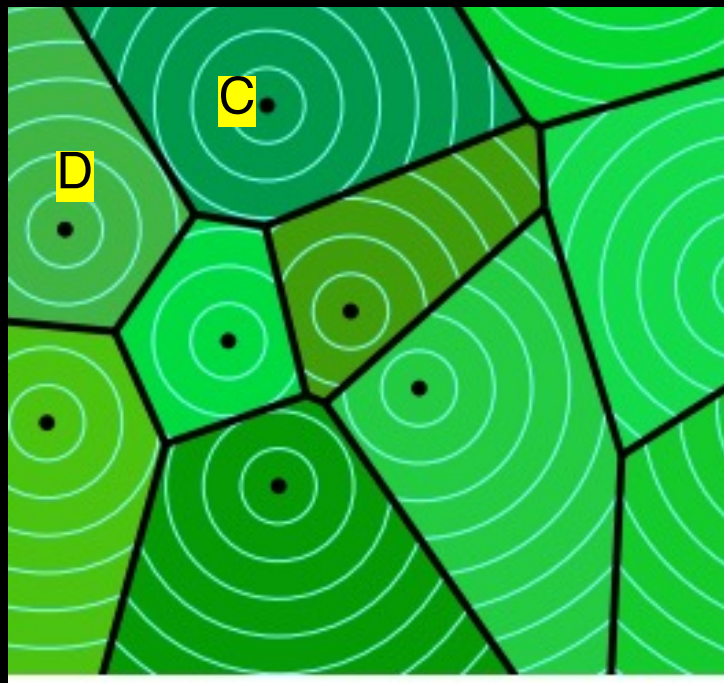




Suppose there are only 2 classes C, D; the optimal decision boundary is formed where the contours of the class-conditional densities intersect – because this is where the classes' discriminant functions are equal that determine the shape of these contours.

If we assumed that variances were the same, i.e. $\sigma_C^2 = \sigma_D^2$ then decision boundary is linear in x

$$\frac{2m_C x_j^{\mu} - m_C^2}{2\sigma^2} + \frac{-2m_D x_j^{\mu} + m_D^2}{2\sigma^2} > 0$$



The likelihood function changes for arbitrary number of features with different variances along different directions

$$\mathrm{L}\left(x_{jk}\middle|\theta=\vec{m},\bar{\bar{\Sigma}}\right)=\prod_{\mu}\frac{1}{\left(\sqrt{2\pi}\right)^{d}\sqrt{\|\bar{\bar{\Sigma}}\|}}exp\left(-\left(\overrightarrow{x^{\mu}}-\vec{m}\right)^{T}\bar{\bar{\Sigma}}^{-1}\left(\overrightarrow{x^{\mu}}-\vec{m}\right)\right)$$

$\bar{\bar{\Sigma}}$ is the d × d covariance matrix; $\bar{\bar{\Sigma}}=\begin{bmatrix} var & \cdots & covar \\ \vdots & \ddots & \vdots \\ covar & \cdots & var \end{bmatrix}$

If independent features then covar=0 and its just independent Gaussians. Decision boundary is high dimensional quadratic. If var(i)=var(j)=var(k) etc, once again quadratic terms cancel so the decision function is linear and the decision boundary is a hyperplane.

If full non-zero covariance – we now acknowledge interdependency of the data and the decision boundaries are many curvatures!

# Anisotropic Gaussians