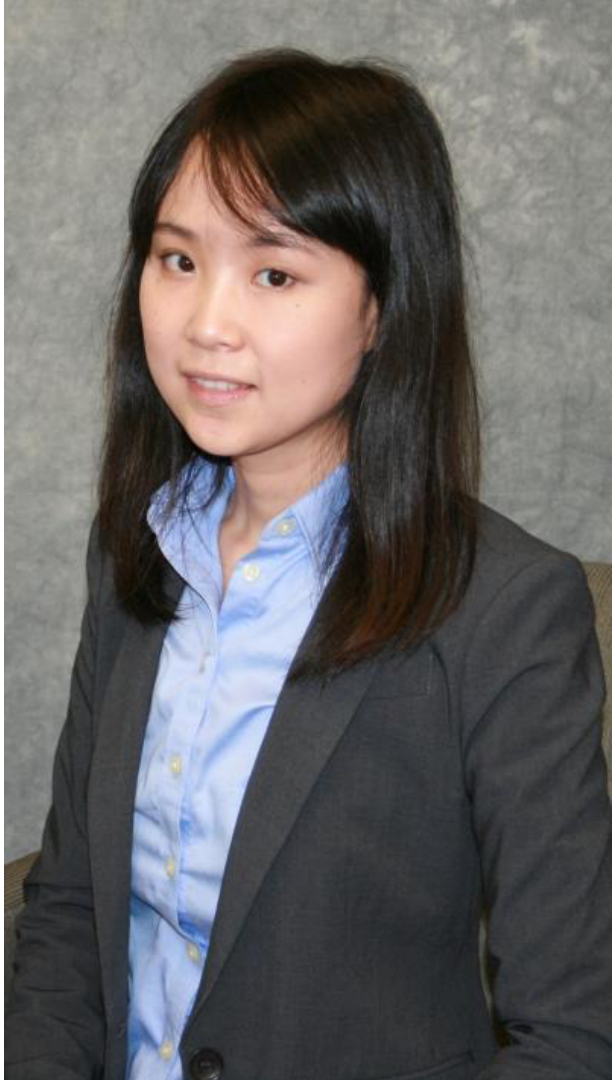
Computational Data Analysis

Machine Learning



Yao Xie, Ph.D.

Associate Professor

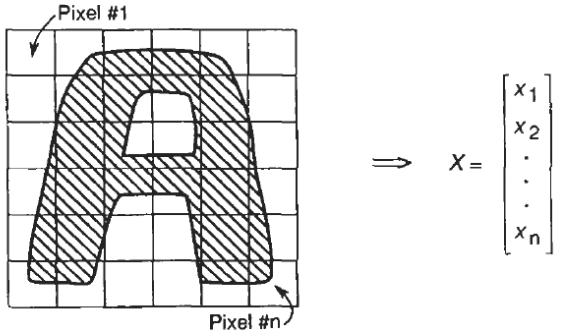
Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering

Classification (I)

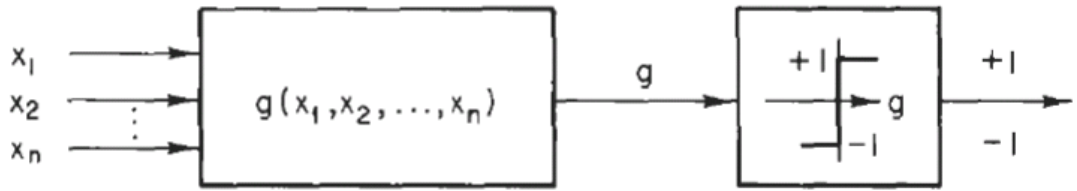
Classification

- Represent the data as a vector



- A label is provided for each data point, eg., $y \in \{-1, +1\}$

- Classifier

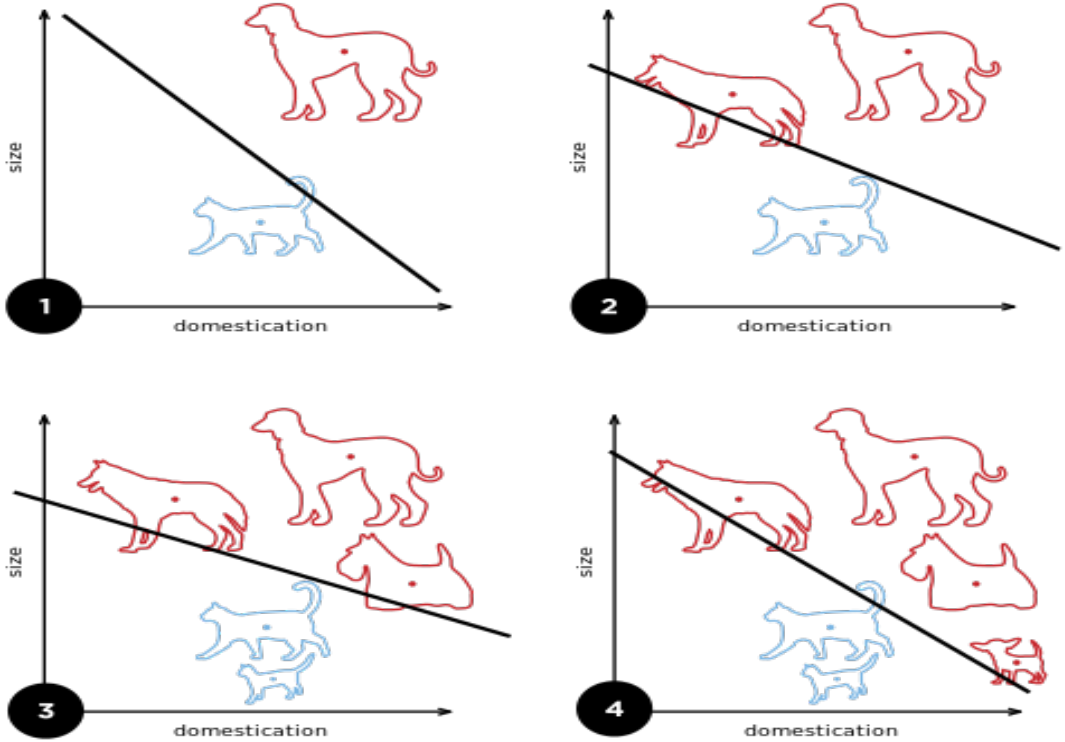


Classification algorithms

- Bayes classifier
- K-nearest neighbors
- Logistic regression

(more to come)

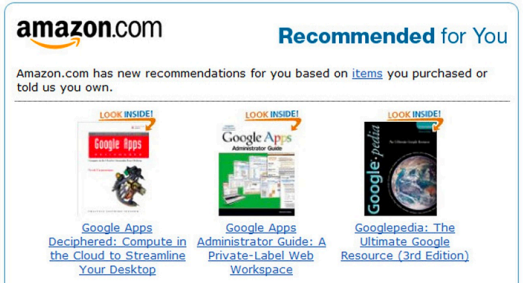
Classify cats from dogs



Classification

- Classification is a predictive task in which the response takes the values across several categories (in the fundamental case, two categories)
- Supervised classification: know the labels for the training data
- Examples:
  - Predicting whether a new patient will develop breast cancer or remain healthy, given genetic information
  - Predicting whether or not a user will like a new product, based on user covariates and a history of his/her previous ratings
  - Predicting the voting preference based on voter's social, political, and economical status

'Doctor in clinic': https://www.leafly.com/news/health/how-to-find-a-doctor-or-clinic-that-specializes-in-medical-mariju

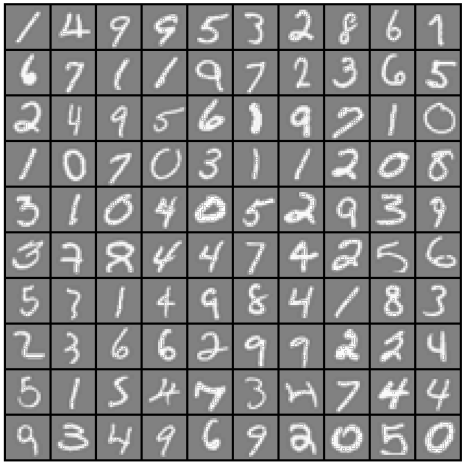


'amazon recommendation': https://www.mageplaza.com/blog/product-recomendation-how-amazon-succeeds-with-it.html

'vote': http://www.livenewsmalta.com/index.php/2017/05/04/early-voting-in-2017-general-election/

Georgia Tech

Classification versus clustering



Classification (training with label)

?

5

?

7

Supervised learning

Clustering (no label)

Unsupervised learning

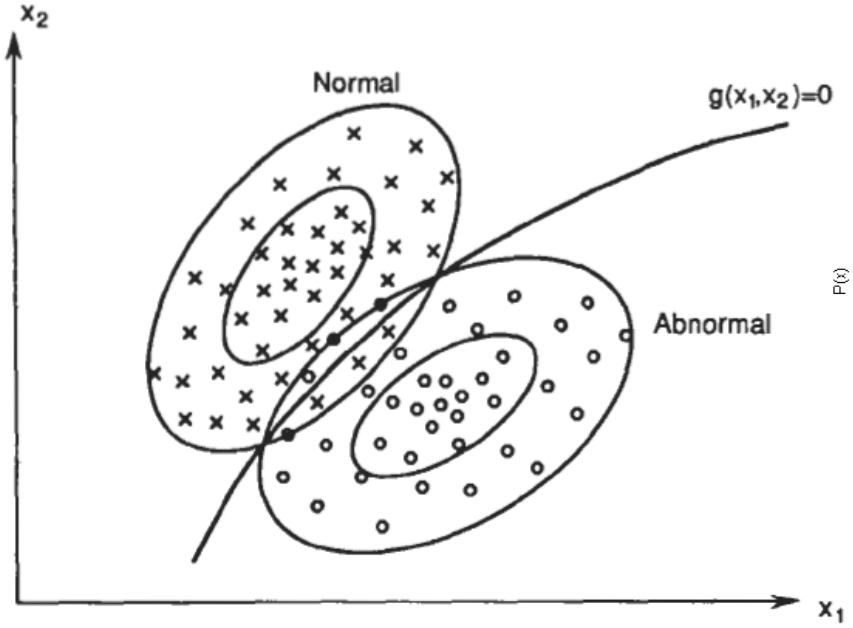
Classification algorithms

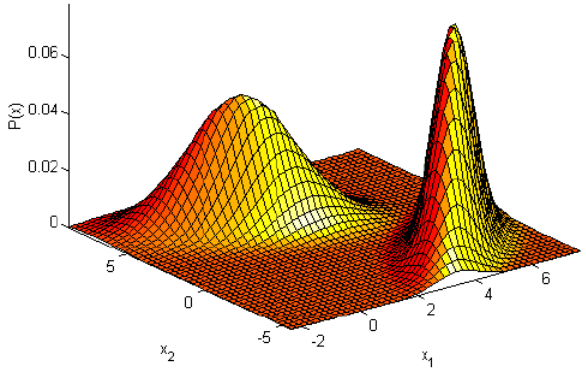
- Bayes classifier
- K-nearest neighbors
- Logistic regression

(more to come)

Decision-making: divide high-dimensional space

- Distributions of sample from normal (positive class) and abnormal (negative class) tissues



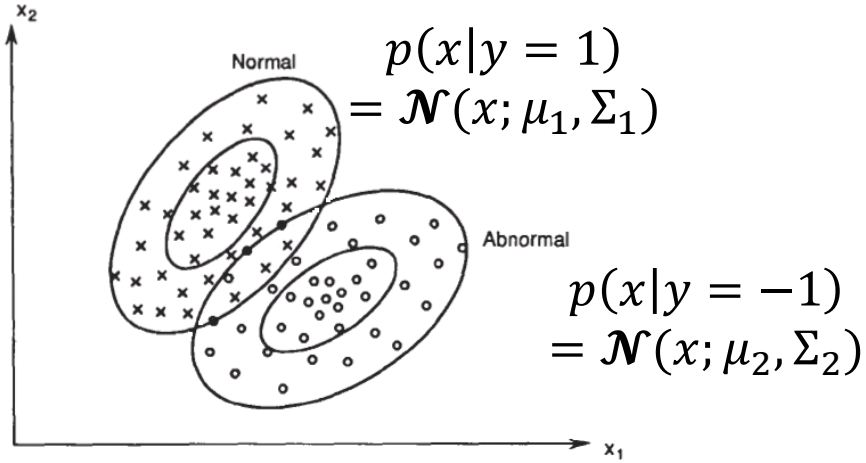


Georgia Tech

Class conditional distribution

- In classification case, we are given the label for each data point

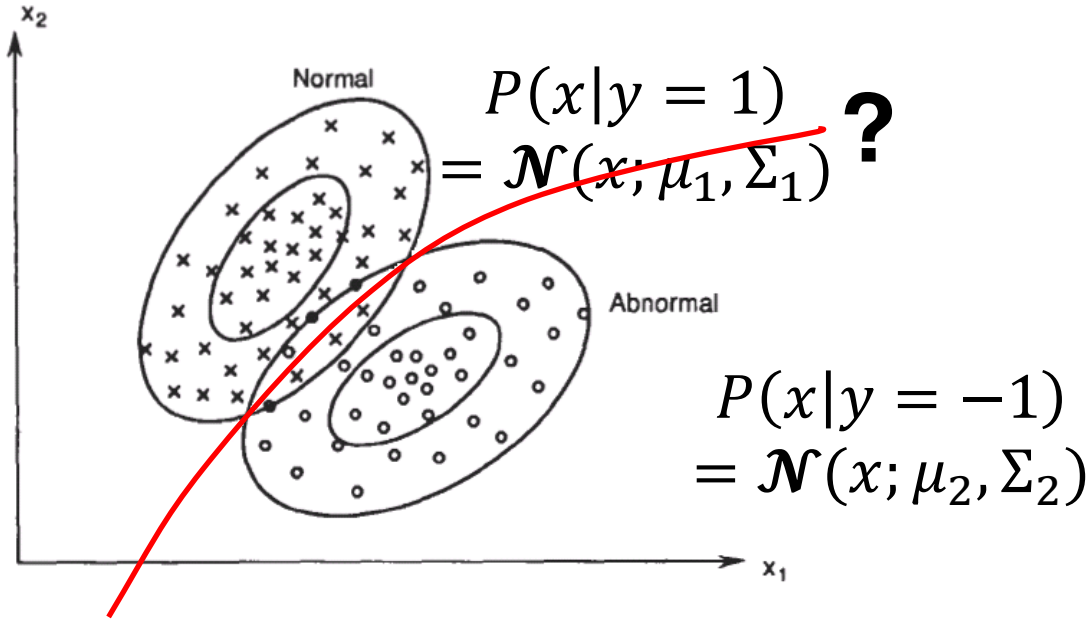
- Class conditional distribution: $P(x|y=1), P(x|y=-1)$ (how to compute?)



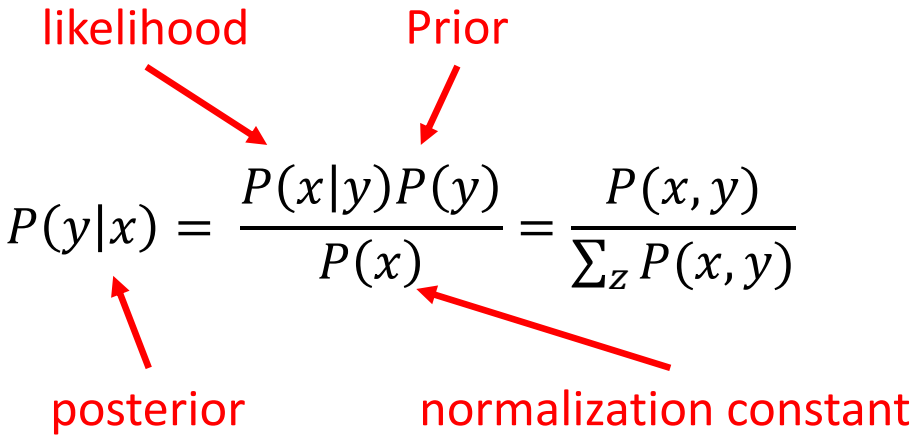
- Class prior: $P(y=1), P(y=-1)$

How to come up with decision boundary

- Given class conditional distribution: $P(x|y=1), P(x|y=-1)$, and class prior: $P(y=1), P(y=-1)$



Use Bayes rule



Prior: $P(y)$

Likelihood (class conditional distribution : $p(x|y) = \mathcal{N}(x|\mu_y, \Sigma_y)$

Posterior: $P(y|x) = \frac{P(y)\mathcal{N}(x|\mu_y,\Sigma_y)}{\sum_y P(y)\mathcal{N}(x|\mu_y,\Sigma_y)}$

Bayes Decision Rule

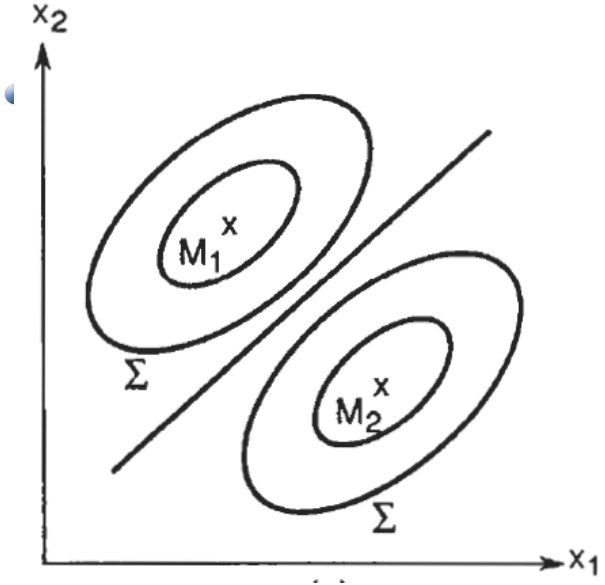
- Learning: prior: $p(y)$,class conditional distribution : $p(x|y)$
- The poster probability of a test point

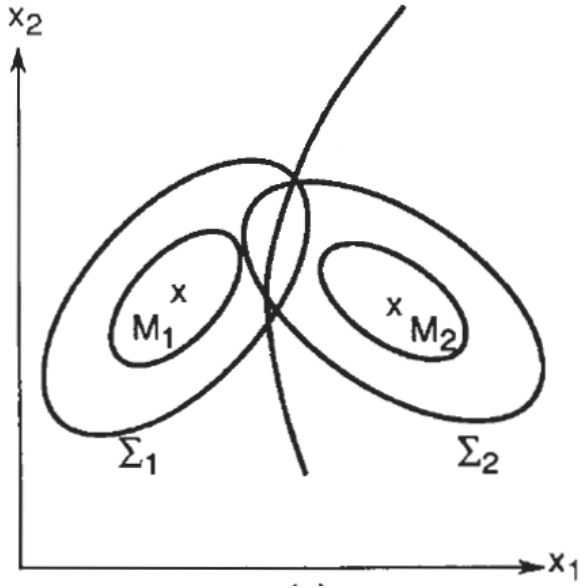
$$q_i(x) := P(y = i|x) = \frac{P(x|y)P(y)}{P(x)}$$

- Bayes decision rule:
  - If $q_i(x) > q_j(x)$, then $y = i$, otherwise $y = j$
- Alternatively:
  - If ratio $l(x) = \frac{P(x|y=i)}{P(x|y=j)} > \frac{P(y=j)}{P(y=i)}$, then $y = i$, otherwise $y = j$
  - Or look at the log-likelihood ratio $\mathrm{h(x)} = -\ln \frac{q_i(x)}{q_j(x)}$

Example: Gaussian class conditional distribution

- Depending on the Gaussian distributions, the decision boundary can be very different





- Decision boundary: $\mathrm{h(x)} = -\ln \frac{q_i(x)}{q_j(x)} = 0$

Naïve Bayes Classifier

- Use Bayes decision rule for classification

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

- But assume $p(x|y=1)$ is fully factorized

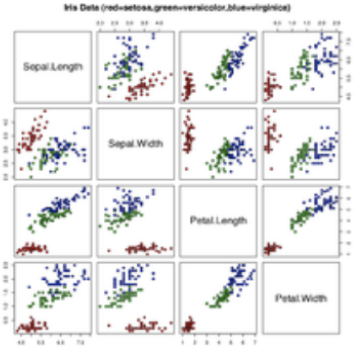
$$p(x|y=1) = \prod_{i=1}^{d} p(x_i|y=1)$$

- Or the variables corresponding to each dimension of the data are independent given the label

So this so-called Naive Bayes is a simplification of this approach.
Basically saying, okay, so maybe if this covariance matrix is hard to
estimate then let's just assume this chorus matrix is identity matrix.
So that means I'm going to zoom different x different feature
vectors are independent of each other, okay?
So this way, the.
Joint the distribution of p given x given y can be very simple.
It's basically going to be the product of pxi given y.
So that means I just need to capture.
The distribution for each of the features and then multiply them
together without considering the correlation between different features.
So this naive bayes is very simple and sometimes can lead to better resolve
because you don't have to estimate a lot of parameters,
especially when you don't have a lot of training data, this can be helpful.

Fisher's Iris Example

- The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Sir Ronald Fisher (1936) as an example of discriminant analysis.
- The data set consists of 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor). Four features were measured from each sample: the length and the width of the sepals and petals, in centimetres.
- Based on the combinaBon of these four features, Fisher developed a linear discriminant model to disBnguish the species from each other
- Demo:



Flower:
https://wnellie.tumblr.com/post/143155937907/analysis-of-the-famous-iris-flower-dataset-part
Fisher:
https://www.umass.edu/wsp/resources/tales/fisher.html
Data set image :
https://en.wikipedia.org/wiki/Iris_flower_data_set

Classification algorithms

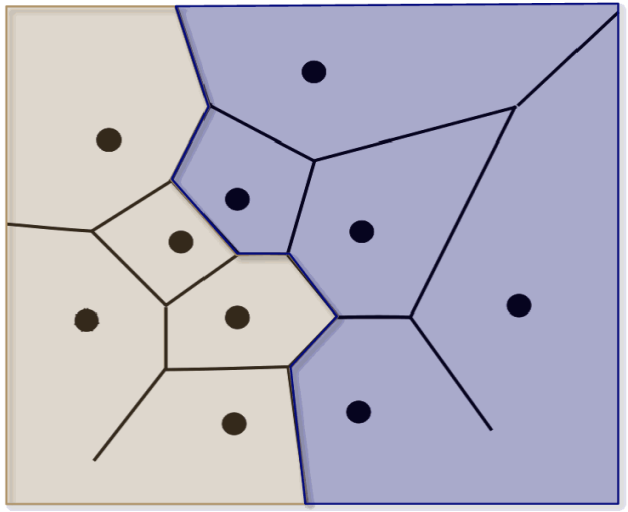
- Bayes classifier
- K-nearest neighbors
- Logistic regression

(more to come)

Nearest neighbor classifier

- The **nearest neighbor classifier**: assign x the same label as the closest training point x_i

- The nearest neighbor rule defines a **Vornoi partition** of the input space



- Nonparametric
- Nonlinear
- Easy to kernelize

K-nearest neighbors

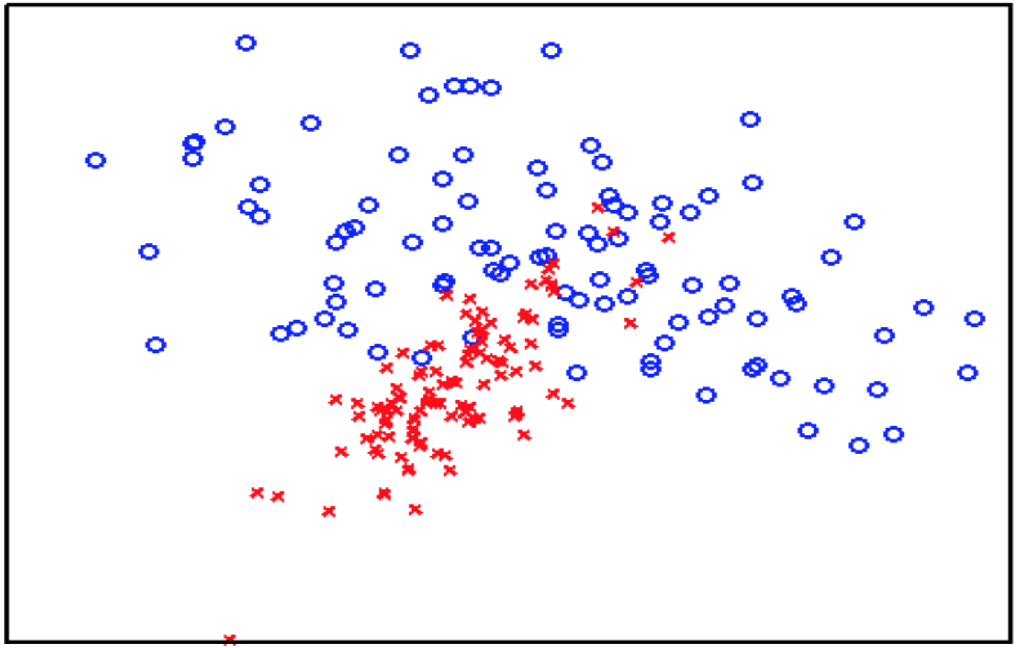
- ***k*-nearest neighbor classifier:** assign x a label by taking a majority vote over the k training points x_i closest to x

- For $K \geq 1$, the *k*-nearest neighbor rule generalizes the nearest neighbor rule

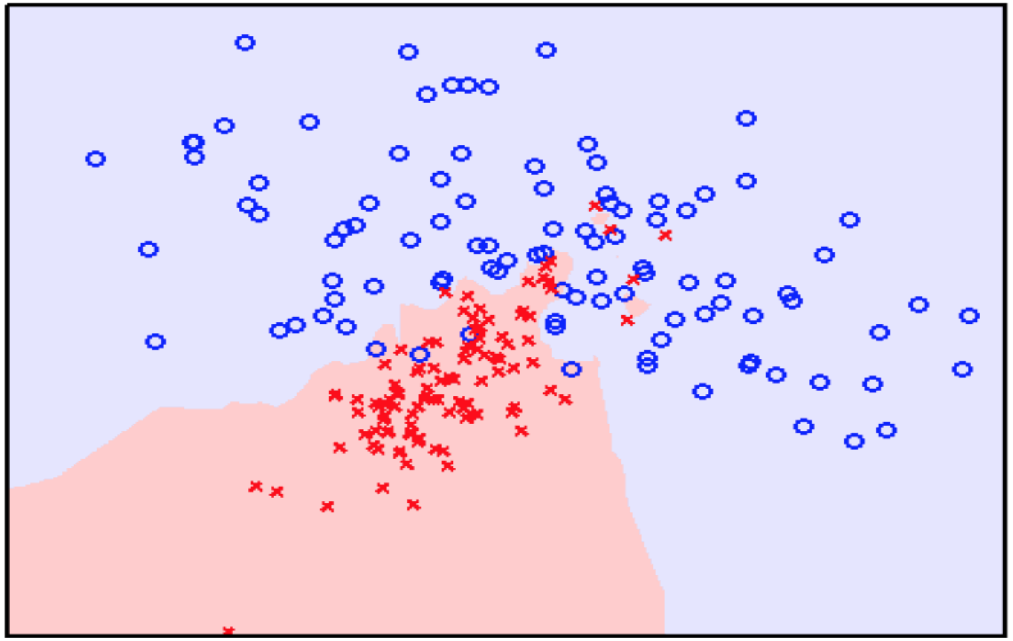
- To define this more mathematically:
  - $\mathrm{I}_k(x) \coloneqq$ indices of the k training points closest to x.
  - If $y_i = \pm 1$, then we can write the k-nearest neighbor classifier as:

$$f_k(x) \coloneqq sign\left(\sum_{i \in I_k(x)} y_i\right)$$

Example

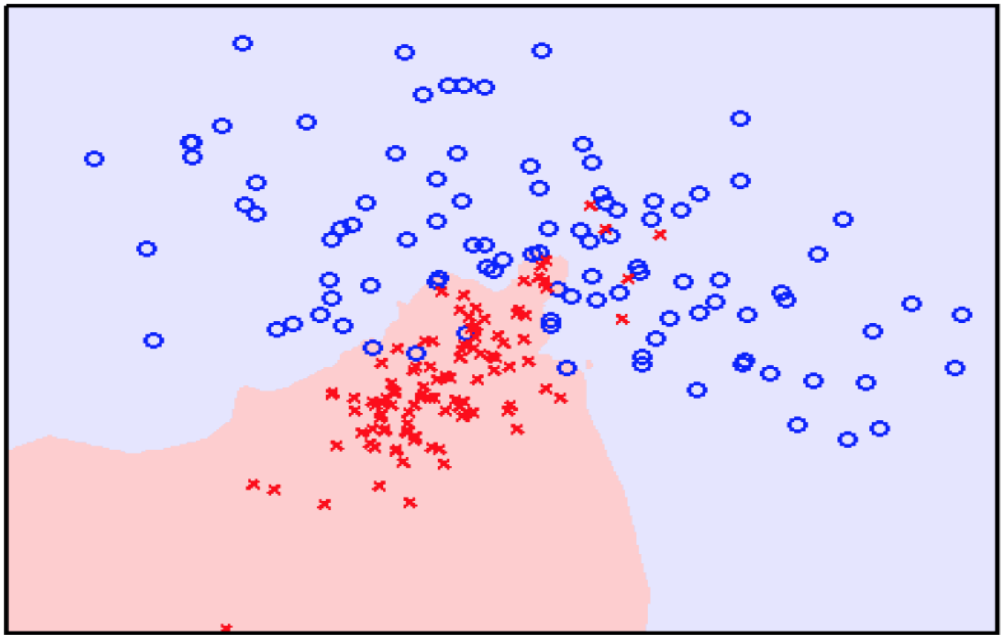


Example



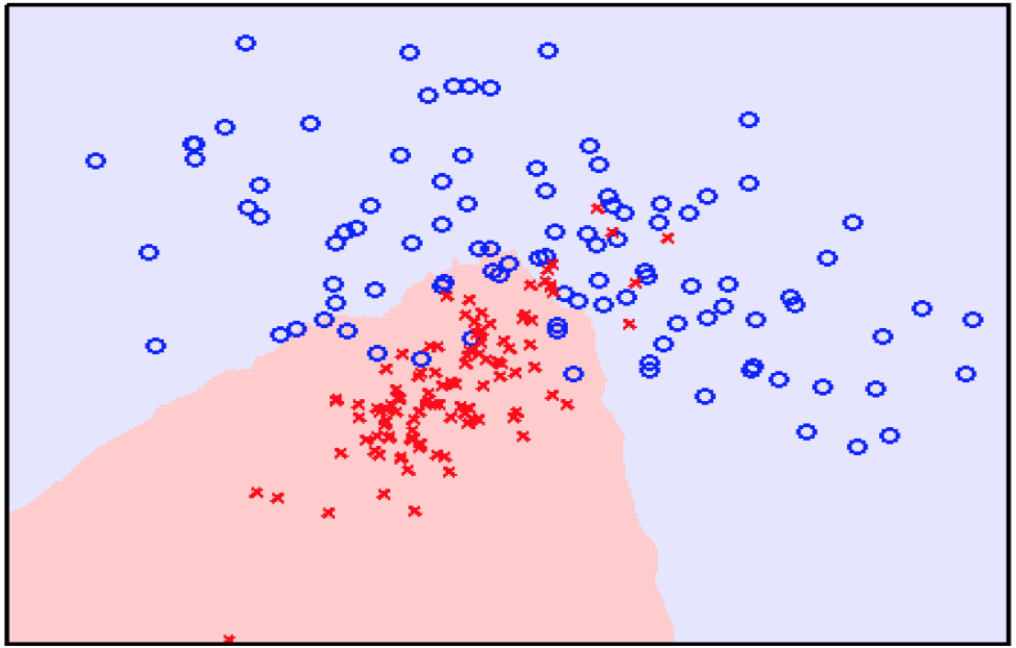
K = 3

Example



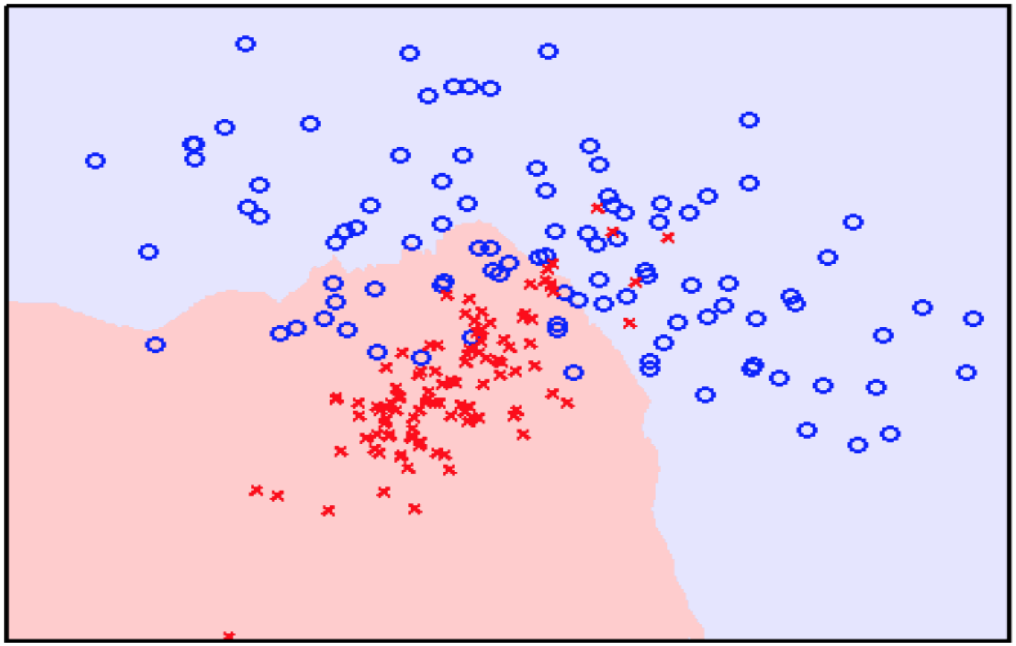
K = 5

Example



K = 25

Example



K = 51

Classification algorithms

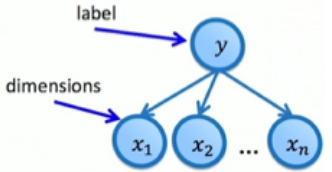
- Bayes classifier
- K-nearest neighbors
- Logistic regression

(more to come)

Naïve Bayes **classifier is a generative model**

- Once you have the model, you can generate sample from it:
  - For each data point x:
    - Sample a label, $y = \{1,2\}$, with according to the class prior $p(y)$
    - Sample the value of x from class conditional $p(x|y^i)$
      - **Naïve Bayes**: conditioned on y, generate first dimension x_1, second dimension x_2, ..., independently

- Difference from mixture of Gaussian models
  - Purpose is different (density estimation vs. classification)
  - Data different (with/without labels)
  - Learning different (em/or not)



Georgia Tech

Classification from a generative model

- Assume data are from a generative model determined by
  - For each data point
  - Sample a label y^i according to a prior $p(y), y = 0, 1$
  - Sample the value of x^i from the conditional probability $p(x|y^i, \theta)$
- Logistic regression: specify $p(x|y^i, \theta)$ using logistic function

  - logistic regression model

$$p(y = 1|x, \theta) = \frac{1}{1 + \exp(-\theta^\top x)}$$

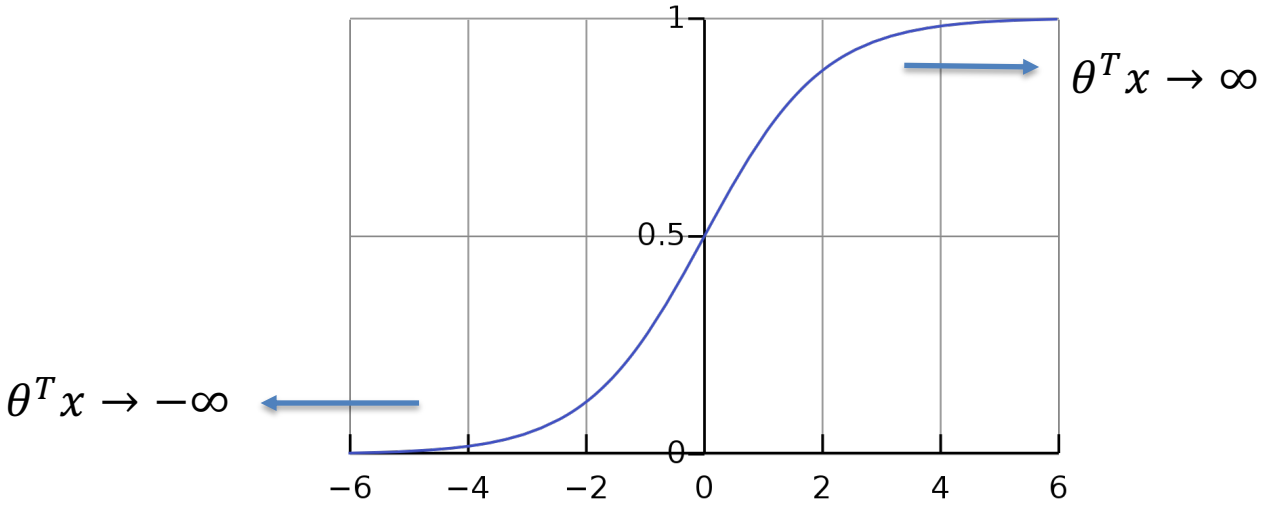
  - Note that

$$p(y = 0|x, \theta) = 1 - \frac{1}{1 + \exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1 + \exp(-\theta^\top x)}$$

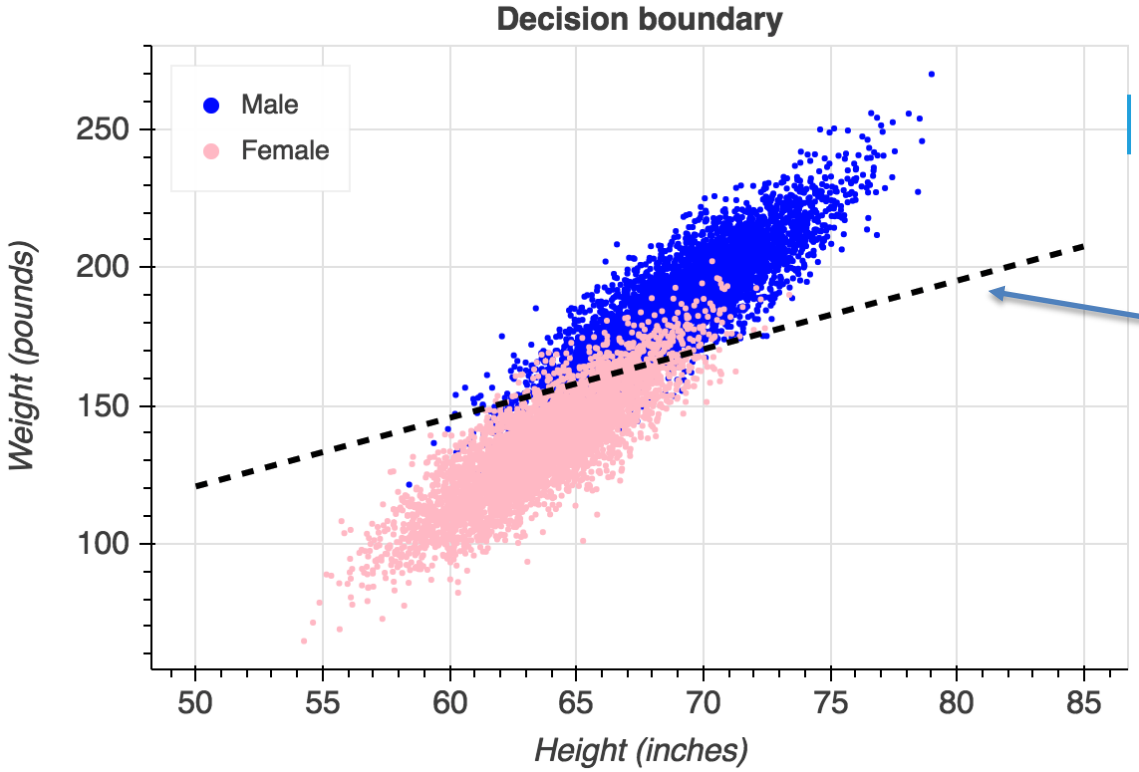
A closer look at logistic regression model

- logistic regression model

$$p(y = 1|x, \theta) = \frac{1}{1 + \exp(-\theta^\top x)}$$



Decision boundary of logistic regression



Change θ will change this decision boundary

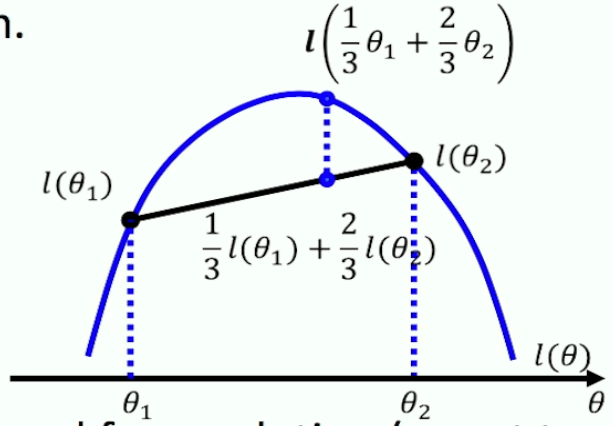
Learning in logistic regression is to find θ to optimally separate the classes in training data

Learning parameters in logistic regression

- Find θ, such that the conditional likelihood of the labels is maximized

$$\max_{\theta} l(\theta) := \log \prod_{i=1}^{m} P\left(y^i \middle| x^i, \theta\right)$$

- Good news: $l(\theta)$ is concave function of θ, and there is a single global optimum.



- Bad new: no closed form solution (resort to numerical method)

Learning parameters in logistic regression

- logistic regression model

$$p(y = 1|x, \theta) = \frac{1}{1 + \exp(-\theta^\top x)}$$

- Note that

$$p(y = 0|x, \theta) = 1 - \frac{1}{1 + \exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1 + \exp(-\theta^\top x)}$$

- Plug in

$$l(\theta) := \log \prod_{i=1}^{m} P\left(y^i | x^i, \theta\right)$$

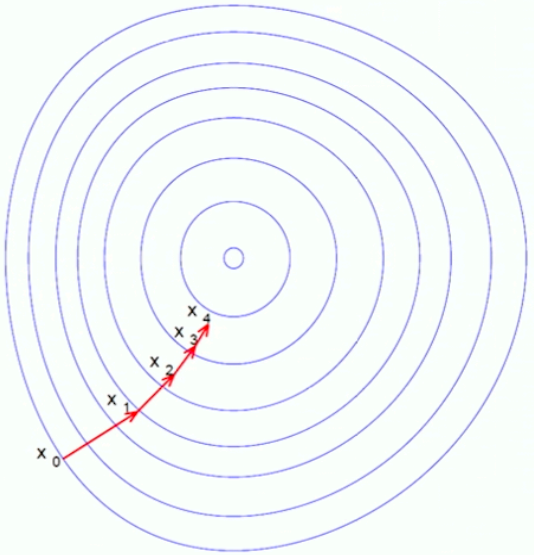
$$= \sum_i (y^i - 1)\, \theta^\top x^i - \log(1 + \exp(-\theta^\top x^i))$$

Gradient descent to minimize $f(x)$

- One way to solve an *unconstrained* optimization problem is gradient descent
- Given an initial guess, we *iteratively* refine the guess by taking the direction of the negative gradient
- Think about going down a hill by taking the steepest direction at each step
- Update rule

$$x_{k+1} = x_k - \gamma_k \nabla f(x_k)$$

γ_k is called the step size or learning rate



Gradient Ascent/Descent Algorithm

$$\max_{\theta} l(\theta) := \log \prod_{i=1}^{m} P(y^i | x^i, \theta)$$

$$= \sum_i (y^i - 1)\, \theta^\top x^i - \log(1 + \exp(-\theta^\top x^i))$$

- Initialize parameter θ^0
- Do

Negative gradient $\nabla l(\theta)$ since we are solving maximization

$$\theta^{t+1} \leftarrow \theta^t + \eta \sum_i (y^i - 1)\, x^i + \frac{\exp(-\theta^\top x^i) x^i}{1 + \exp(-\theta^\top x)}$$

- While the $||\theta^{t+1} - \theta^t|| > \epsilon$

Batch gradient vs stochastic gradient

Note that gradient decouples: consists of sum of evaluations over individual samples

$$\nabla l(\theta) = \sum_i (y^i - 1)\,\theta^\top x^i - \log(1 + \exp(-\theta^\top x^i))$$

Batch gradient descent

- To compute the gradient at *each* iteration, we need to sum over *all* data points in the dataset
- What if we have a huge dataset? For example, 1 Million data points?

Stochastic gradient descent

- At each iteration, we randomly sample a small subset S_k data point $(x_i, y_i), i \in S_k$(in the extreme case, just one sample in S_k), and update using gradient estimated using a small subset of data

$$\nabla \hat{l}(\theta) = \sum_{i \in S_k} (y^i - 1)\, \theta^\top x^i - \log(1 + \exp(-\theta^\top x^i))$$

- Using a different subset each time, so eventually loop through entire training data

Stochastic gradient descent (SGD) vs. Batch Gradient Descent (GD)

- Time complexity: $O(1)$ (SGD) vs $O(n)$ (GD)
- The key thing is that the gradient is unbiased, thus the expectation equals the true gradient

Naïve Bayes vs. logistic regression

- Consider $y \in \{1, -1\}, x \in R^n$
- Number of parameters
  - Naïve Bayes : $2n + 1$
    - When all random variables are binary
    - 4n+1 for Gaussians: $2n$ mean, $2n$ variance, and 1 for prior
  - logistic regression: $n + 1$
    - $\theta_0, \theta_1, \theta_2, \ldots, \theta_n$

Naïve Bayes vs. logistic regression II

- Asymptotic comparison (# training examples → infinity)
- When model assumptions correct
  - Naïve Bayes, logistic regression produce identical classifiers
- When model assumptions incorrect
  - logistic regression is less biased – does not assume conditional independence
  - logistic regression has fewer parameters
  - therefore expected to outperform Naïve Bayes

Step sizes and stopping criterion

- For SGD, due to noise introduced by random sampling data, need to use decreasing step size, such as $O\left(\frac{1}{k}\right)$
  - Step size too small: not making too much progress
  - Step size too large: overshoot

- The optimality condition is $\nabla l(\theta) = 0$

Stop when the gradient becomes small $\|\nabla l(\theta)\|$, e.g., *l*-2 norm

Comparing Bayes classifier and logistic regression

- Consider $y \in \{1, -1\}, x \in R^n$
- Number of parameters
  - Naïve Bayes : $2n + 1$
    - When all random variables are binary
    - 4n+1 for Gaussians: $2n$ mean, $2n$ variance, and 1 for prior
  - logistic regression: $n + 1$
    - $\theta_0, \theta_1, \theta_2, \ldots, \theta_n$

- Asymptotic comparison (# training examples → infinity)

- When model assumptions correct
  - Naïve Bayes, logistic regression produce identical classifiers

- When model assumptions incorrect
  - logistic regression is less biased – does not assume conditional independence
  - logistic regression has fewer parameters
  - therefore expected to outperform Naïve Bayes

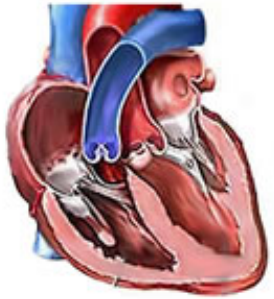
- Estimation method:
  - Naïve Bayes parameter estimates are decoupled (super easy)
  - Logistic regression parameter estimates are coupled (less easy)
- How to estimate the parameters in logistic regression?
  - Maximum likelihood estimation
  - More specifically, maximize the conditional likelihood the label

Implementation

Look at logistic regression:
https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
Nearest neighbors:
https://scikit-learn.org/stable/modules/neighbors.html
Naïve Bayes:
https://scikit-learn.org/stable/modules/naive_bayes.html

Demo: Statlog (Heart) Data Set

- 13 attributes (see heart.docx for details)
  - 2 demographic (age, gender)
  - 11 clinical measures of cardiovascular status and performance
- 2 classes: absence (1) or presence (2) of heart disease
- 270 samples
- Dataset taken from UC Irvine Machine Learning Repository
- Demo code




SUMMARY