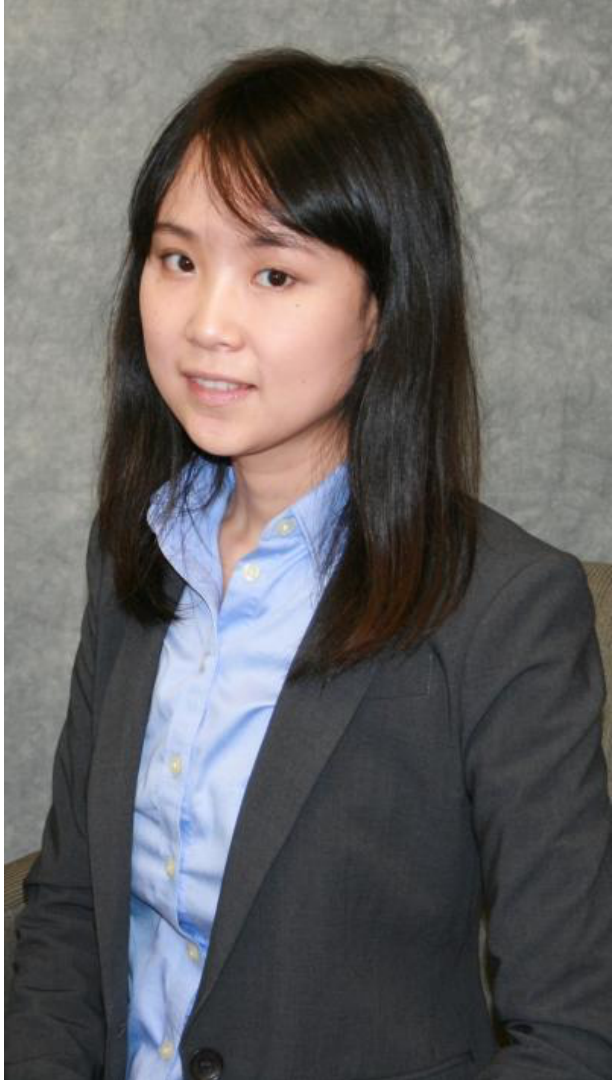
# Computational Data Analysis

## Machine Learning



**Yao Xie, Ph.D.**

*Associate Professor*

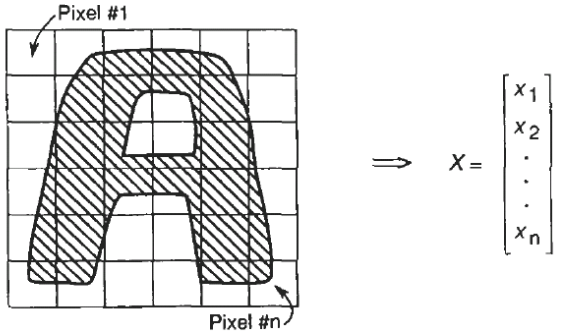
Harold R. and Mary Anne Nash Early Career Professor

H. Milton Stewart School of Industrial and Systems Engineering

## Classification (I)

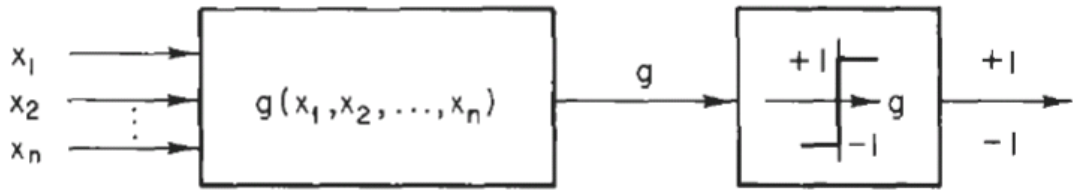
# Classification

- Represent the data as a vector




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

- Classifier

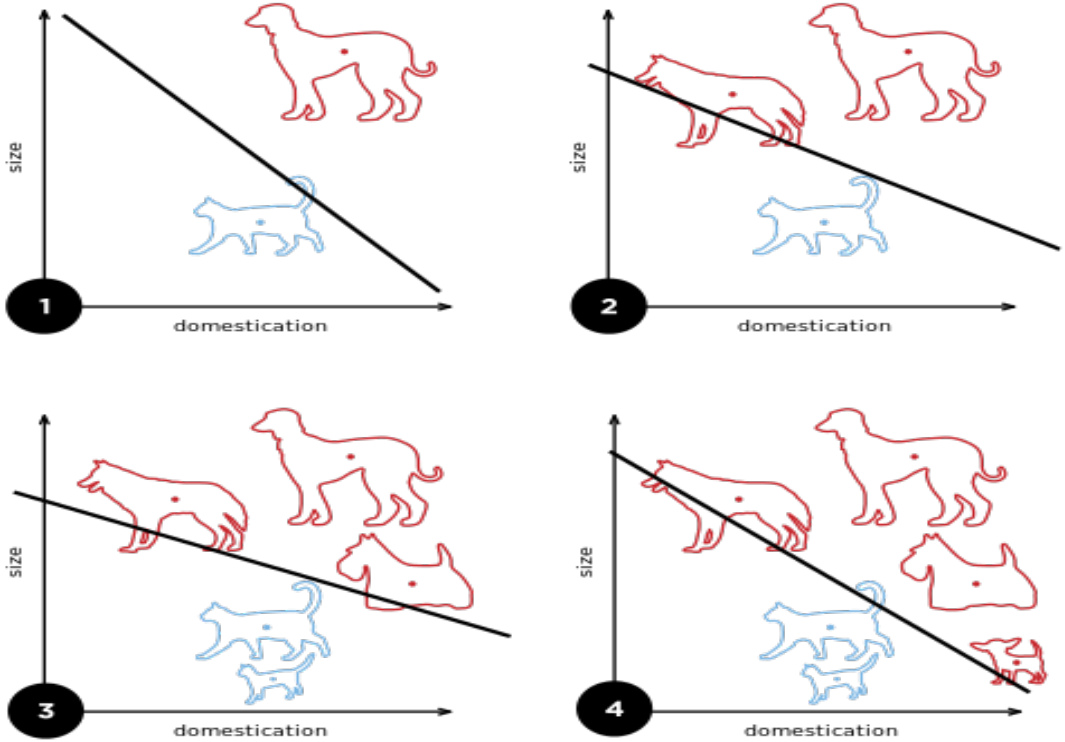


# Classification algorithms

- Bayes classifier
- K-nearest neighbors
- Logistic regression

(more to come)

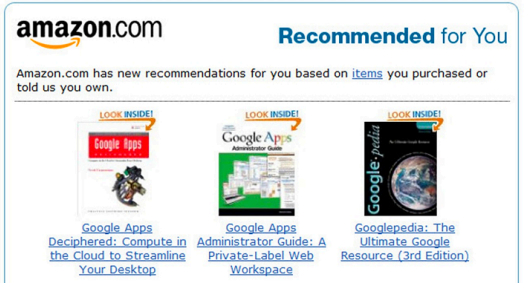
# Classify cats from dogs



# 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

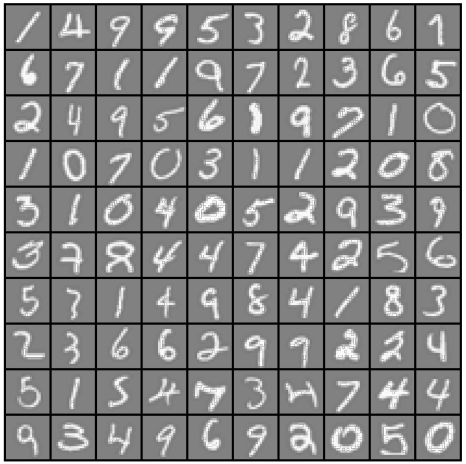



'amazon recommendation': https://www.mageplaza.com/blog/product-recomendation-how-amazon-succeeds-with-it.html


'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

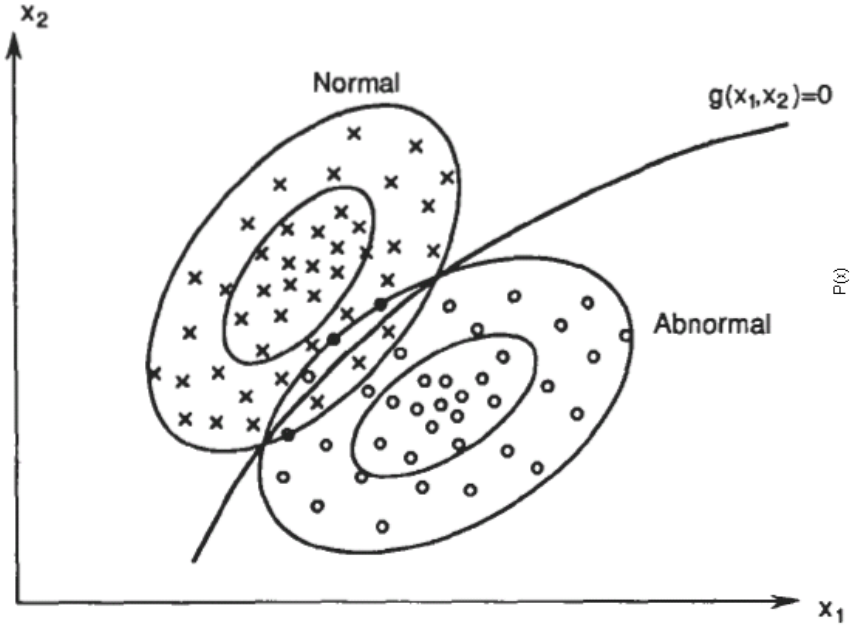
# Classification algorithms

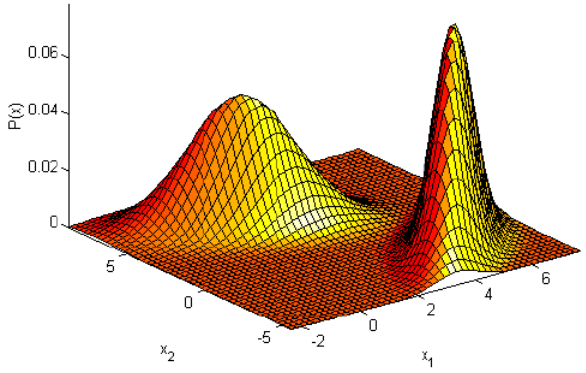
- Bayes classifier
- K-nearest neighbors
- Logistic regression

(more to come)

# Decision-making: divide high-dimensional space

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

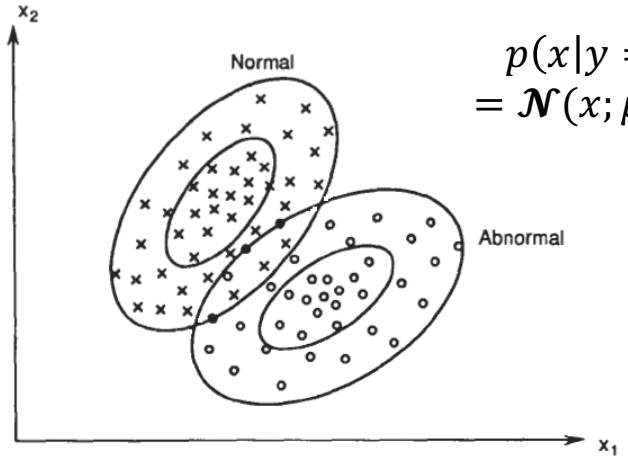



Georgia Tech

# 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?)



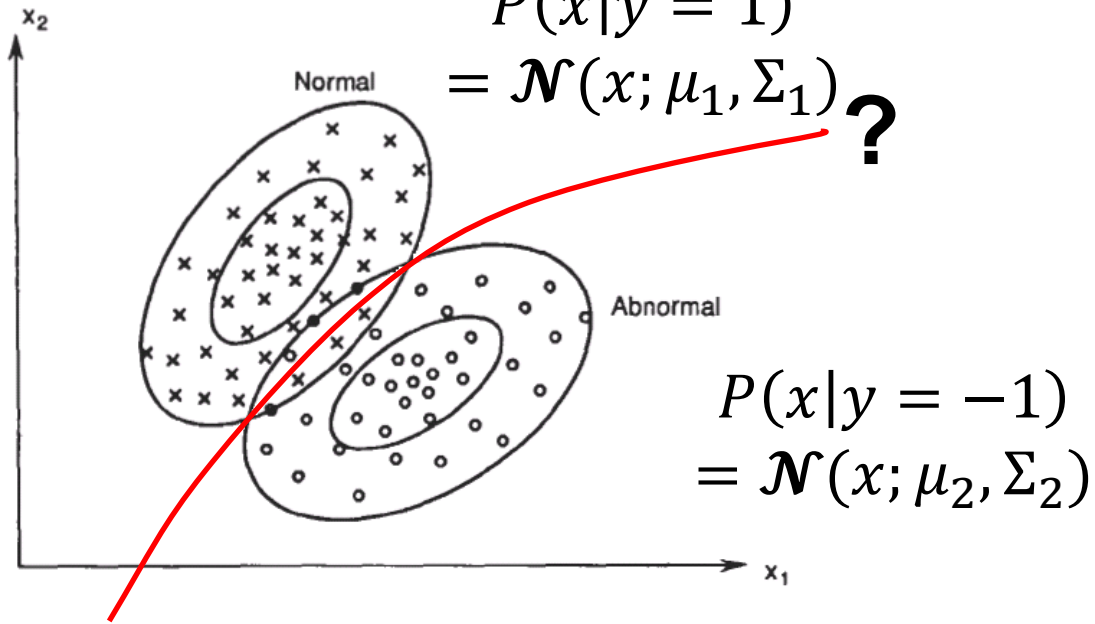

$$p(x|y=1) = \mathcal{N}(x; \mu_1, \Sigma_1)$$

$$p(x|y=-1) = \mathcal{N}(x; \mu_2, \Sigma_2)$$

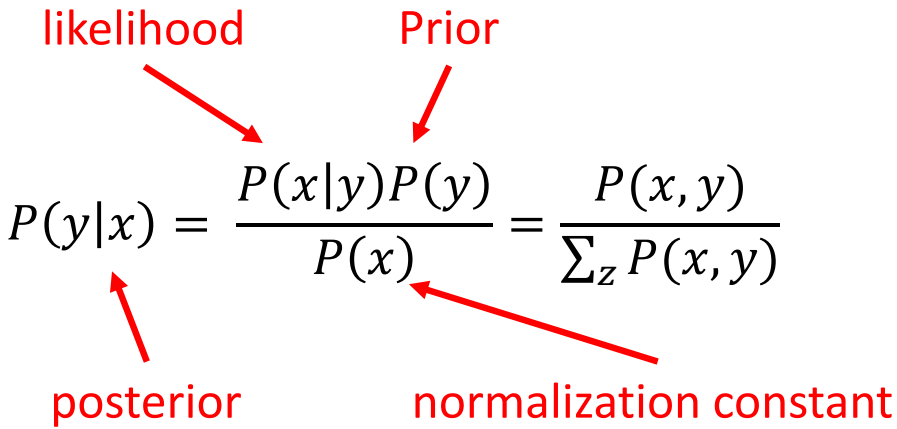
- Class prior: $P(y=1), P(y=-1)$

# How to come up with decision boundary

Given class conditional distribution (likelihood): $P(x|y=1), P(x|y=-1)$, 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/specify class **prior**: $p(y)$, **likelihood**: $p(x|y)$

Calculate **posterior** probability of a test sample $x$

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

Bayes decision rule:

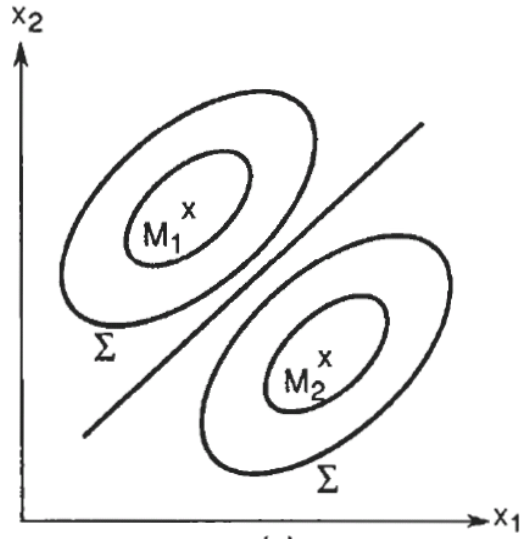
- If $q_i(x) > q_j(x)$, then $y = i$, otherwise $y = j$

This is equivalent to:

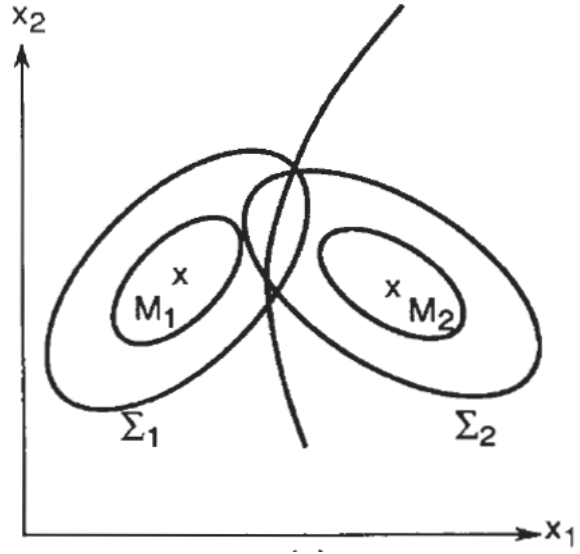
- If ratio $l(x) = \frac{P(x|y=i)}{P(x|y=j)} > \frac{P(y=j)}{P(y=i)}$, then $y = i$, otherwise $y = j$
- We can also use the log-likelihood ratio $h(\mathrm{x}) = -\ln \frac{q_i(x)}{q_j(x)}$ to compare it with 0

# Example: Gaussian class conditional distribution

Depending on assumptions made for the likelihood, the decision boundary can be very different




LDA




QDA

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

# Naïve Bayes Classifier

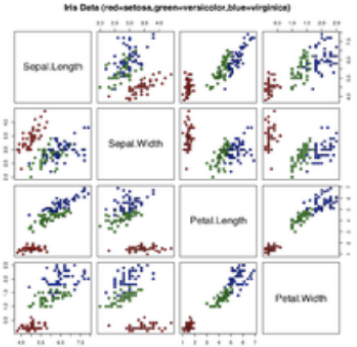
- Use Bayes decision rule for classification
- Simplify by assuming all features (dimensions of the vector $x$) are independent given the label
- Thus likelihood $p(x|y=1)$ is fully factorized

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

- Similarly for $p(x|y=0)$

# 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

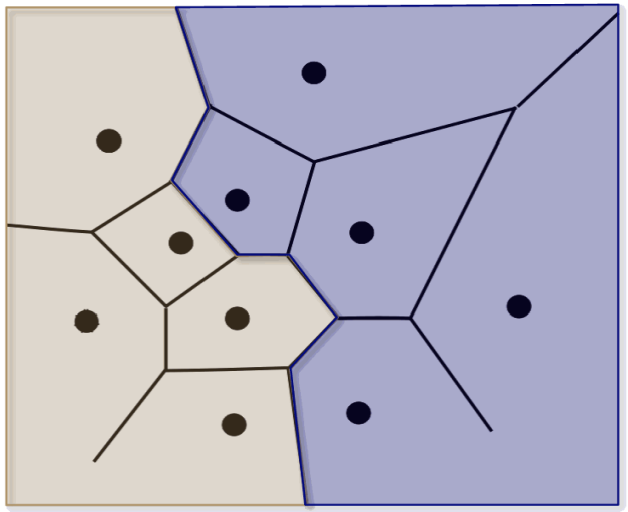
(more to come)

# Nearest neighbor classifier

- The **nearest neighbor classifier**:

  assign $x$ the same label as the closest training samples $x^i$

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



- Nonparametric
- Nonlinear
- Easy to kernelize

# *K*-nearest neighbors

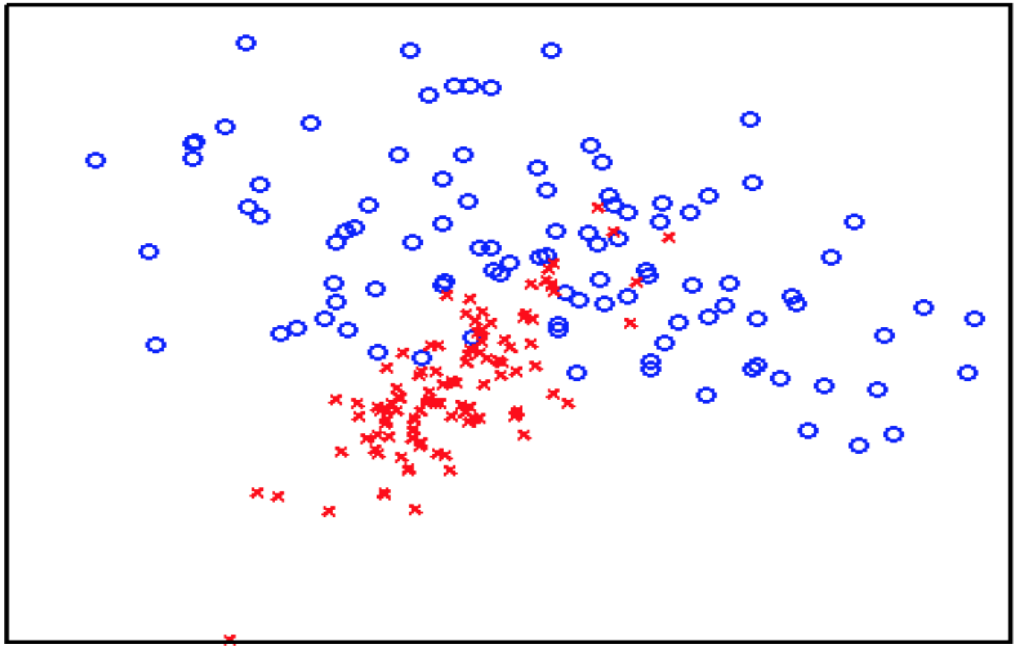
***K*-nearest neighbor classifier:** assign $x$ a label by taking a **majority vote** over the $K$ training points $x^i$ closest to $x$

To define this more mathematically:

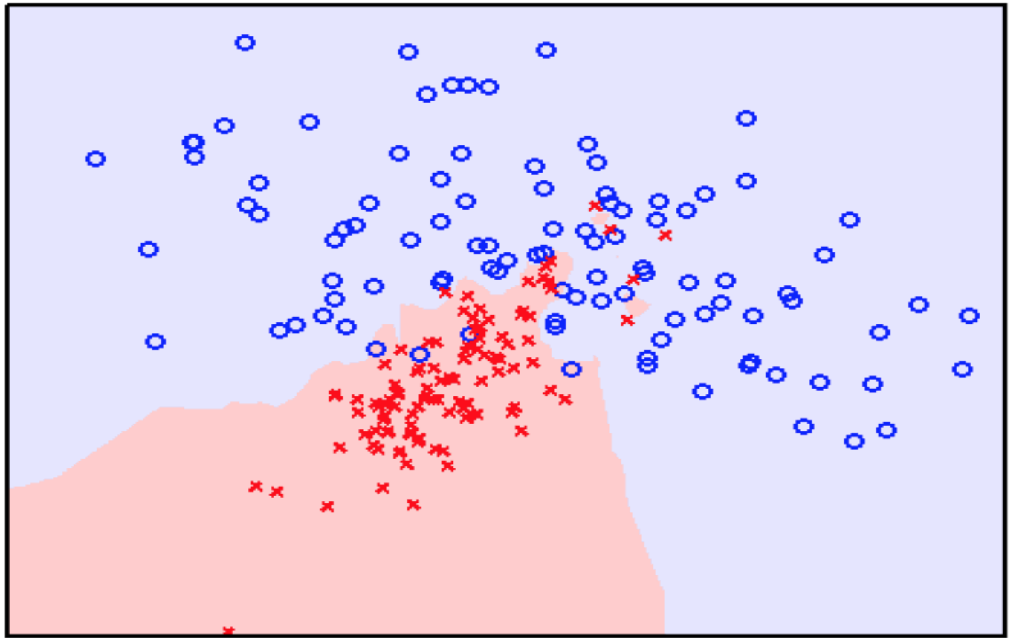
- $I_k(x) \coloneqq$ indices of the $k$ training points closest to $x$.
- When labels $y_i = \pm 1$, then we can write the $k$-nearest neighbor classifier as:

$$\underbrace{f_k(x)}_{\text{predicted label for test point}} \coloneqq sign\left(\sum_{i \in I_k(x)} y^i\right)$$

# Example

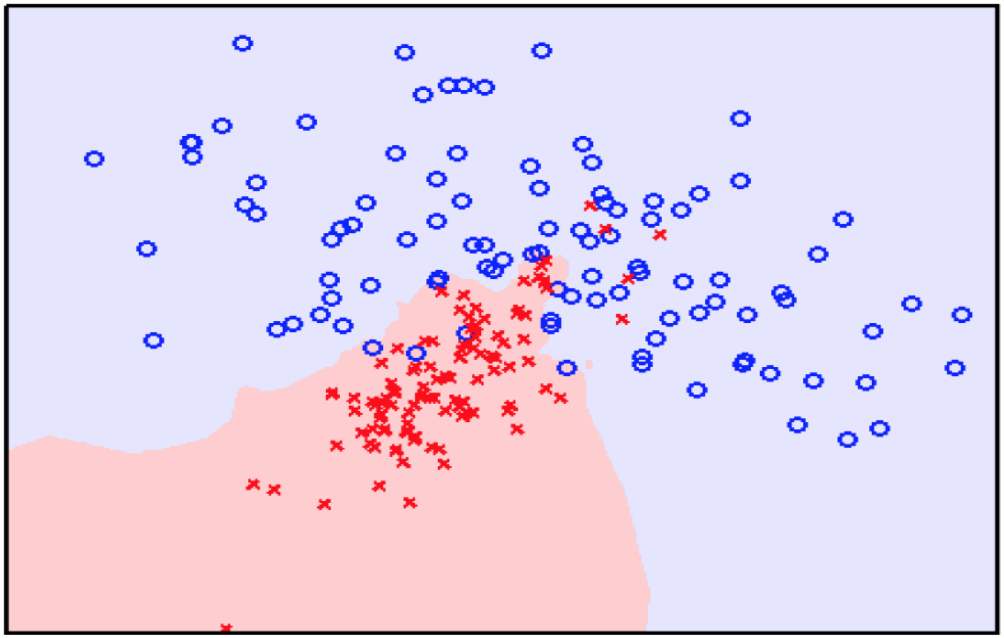


# Example



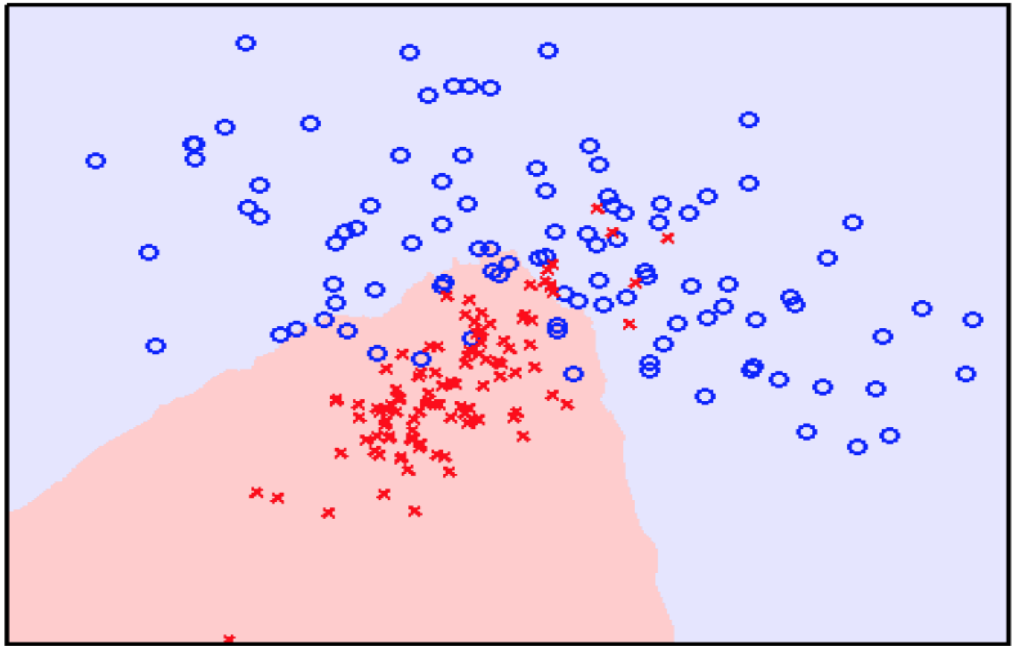
*K = 3*

# Example



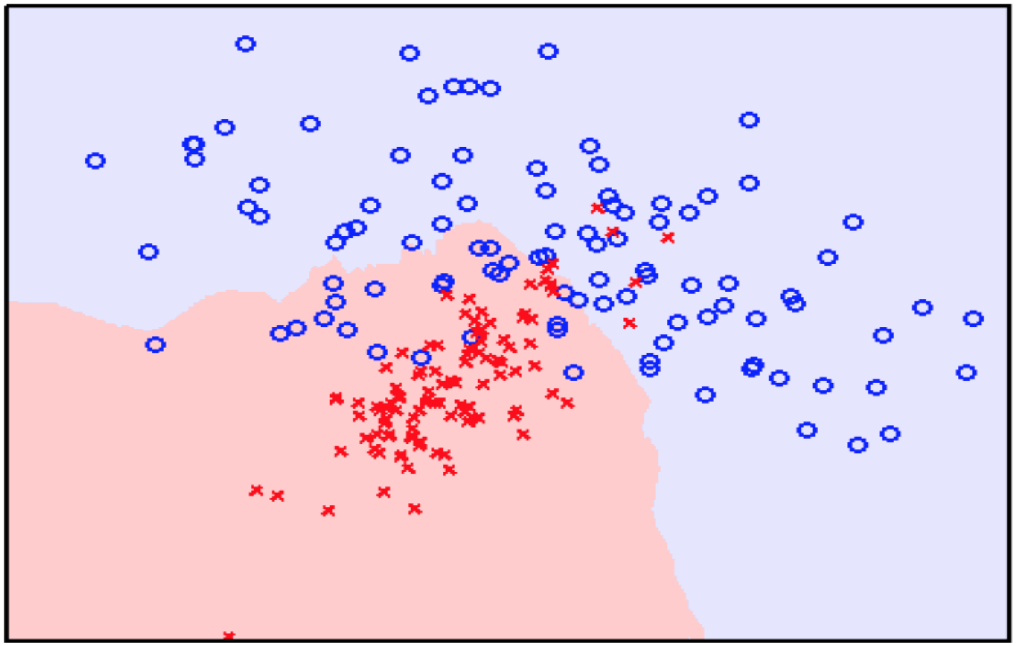
*K = 5*

# Example



*K = 25*

# Example



*K = 51*

# Classification algorithms

- Bayes classifier
- K-nearest neighbors
- Logistic regression

(more to come)

# 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 $p\left(x\middle|y^i, \theta\right)$
- Logistic regression: specify $p\left(x\middle|y^i, \theta\right)$ using logistic function
  - logistic regression model

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

  - Note that

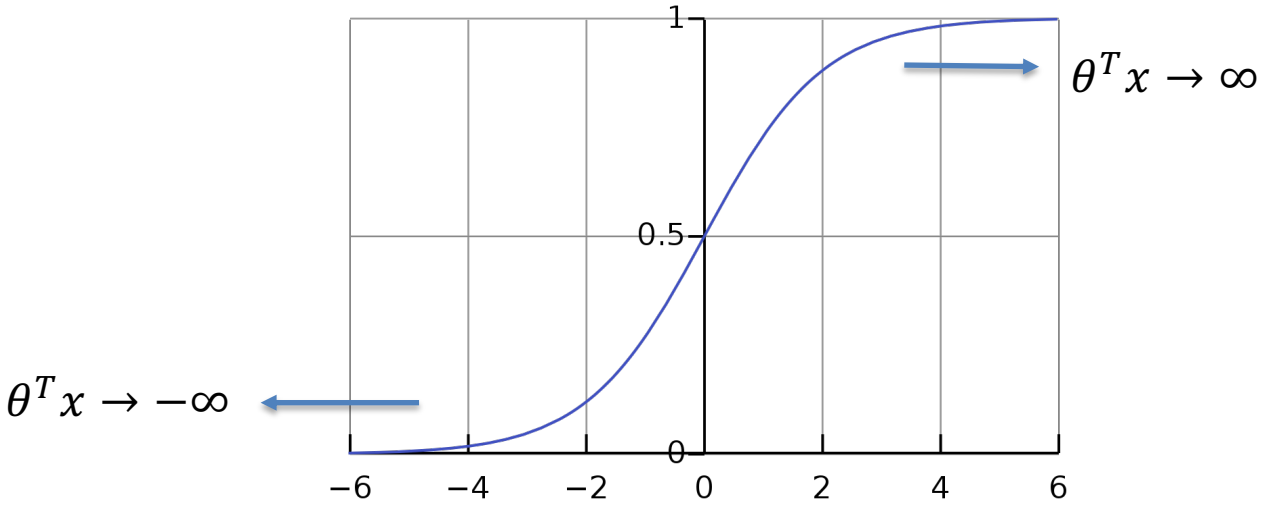
$$p(y = 0|x, \theta) = 1 - \frac{1}{1 + \exp(-\theta^\top x)} = \frac{\exp(-\theta^\top x)}{1 + \exp(-\theta^\top x)}$$

# A closer look at logistic regression model

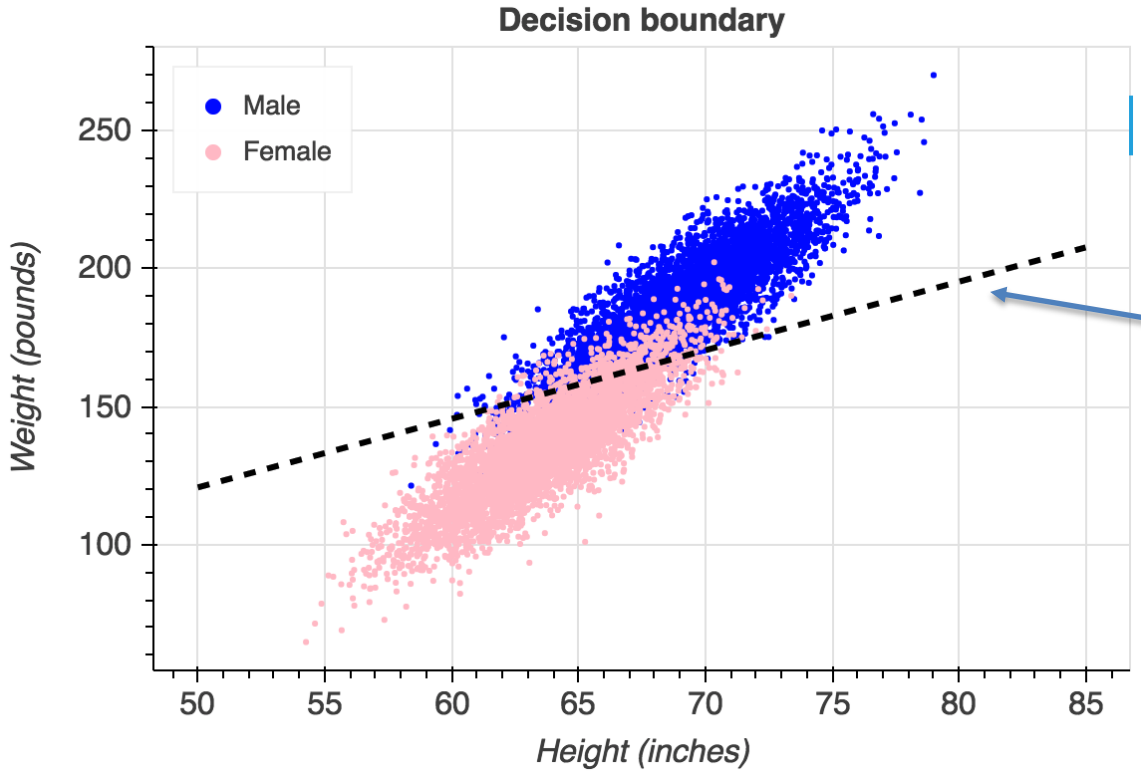
- Assume that the posterior distribution $p(y = 1|x)$ take a particular form

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

- Logistic function $f(u) = \frac{1}{1+\exp(-u)}$



# Decision boundary of logistic regression




Change $\theta$ will change this decision boundary

**Learning** in logistic regression is to find $\theta$ to optimally separate the classes in training data

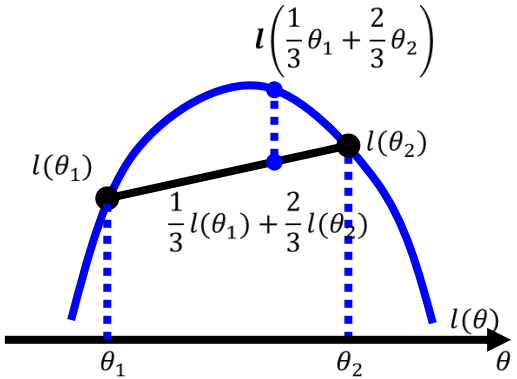
# Learning parameters in logistic regression

Find $\theta$, 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 $\theta$, 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

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

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

# Gradient of $l(\theta)$

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

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

Gradient

$$\frac{\partial l(\theta)}{\partial \theta} = \sum_i (y^i - 1)\, x^i + \frac{\exp(-\theta^\top x^i) x^i}{1 + \exp(-\theta^\top x)}$$

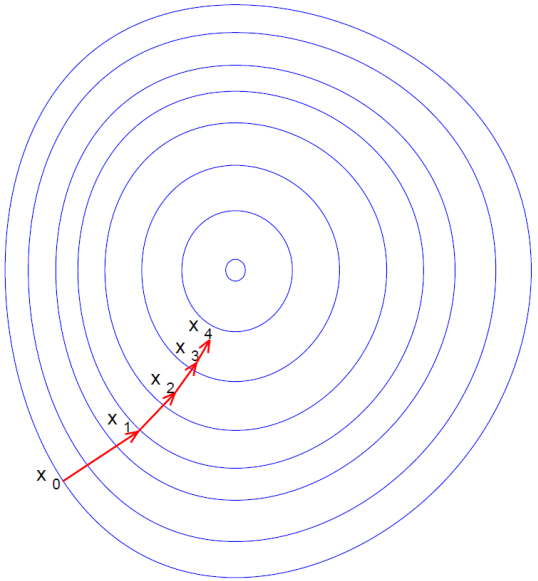
Setting it to 0 does not lead to closed form solution

# 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

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

  $\gamma_k$ is called the step size or learning rate



# Gradient Ascent/Descent Algorithm

Initialize parameter $\theta^0$

Do

gradient of $\nabla l(\theta)$ 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

- The algorithm on the previous page is called the 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? 1million data point?
- Note that gradient de-couples: it consists of sum of evaluations over individual samples

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

# 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, there is just one sample in $S_k$)
- Use gradient estimated using a small subset of data (instead of full data)

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

- Each iteration use a different subset $S_k, k = 1, 2, \ldots.$

- Eventually loop through the entire training data, and may loop through the data multiple times

# 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

# 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 Naïve Bayes and logistic regression

- Consider $y \in \{1, -1\}, x \in R^n$

- Number of parameters

  - Naïve Bayes : $2n + 1$
    - $n$ mean, $n$ variance, and 1 for prior

  - logistic regression: $n + 1$
    - $\theta_0, \theta_1, \theta_2, \ldots, \theta_n$

- Asymptotic comparison (# training examples $\rightarrow$ 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

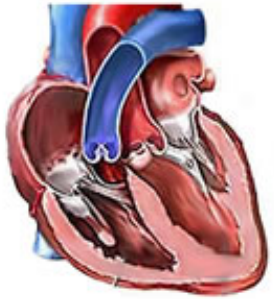
- Estimation method:
  - Naïve Bayes parameter estimates are decoupled (super easy)
  - Logistic regression parameter estimates are coupled (less easy)
- How to estimate the parameters in logistic regression?
  - Maximum likelihood estimation
  - More specifically, maximize the conditional likelihood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