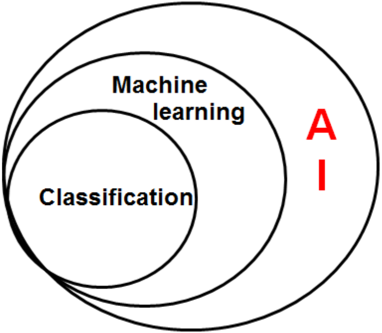
Data Analysis via Classification Algorithms

Lydia Y. Chen

Research Staff Member
IBM Research Zurich Lab, Switzerland



Classification Example: Email Spam

- Observe the set of emails (strings)
- Check if new emails are spam or non-spam

`From: medshop@spam.com`
`Subject: Viagra`
`cheap meds...`
	Spam

`From: my.instructor@Zurich.ibm.com`
`Subject: important information`
`here's how to ace the test...`
	No-spam

`From: mike@example.org`
`Subject: you need to see this`
`how to win $1,000,000...`
	???

Classification Example: Credit Card Fraud

- Observe a set of old transactions
- Predict if new transactions are fraud or normal

Cash advance \$1,000 Cash advance \$10,000 Flight out of the country \$1,200	Fraud
Candy bar \$1.20 Groceries \$67.10 Restaurant \$35.82 …… ...	Normal
Flight out of the country \$380 Hotel booking \$210 Groceries \$69.20	???

Classification Example: Diabetes Diagnoses

- Observe a set of patients and their physical conditions
- Predict if new patients will have diabetes or not

Patient Alice Age 60 Blood pressure 150 mmHG Weight 78 kg	Diabetes
Patient Bob Age 30 Blood pressure 120 mmHG Weight 70 kg	No Diabetes
Patient Maira Age 55 Blood pressure 150 mmHG Weight 60 kg	???

Classification

Classification Problem

- Input X (observable features)
  - One to multiple
  - E.g., age, blood pressure, and weight
- Output Y (class outcomes)
  - Binary (y=1 or -1) or multiple classes (G=1,2,..k)
  - E.g., the sign of diabetes or not
- Objective
  - Optimize the conditional probability of data set

$$\max Pr(G = k \mid X = x)$$

Building Classifier

- Making assumptions about the data structure
- Choosing a classifier, e.g., linear v.s. nonlinear, generative v.s. discriminant
- Parameterizing/fitting the model (training phase), involving optimization procedures
- Classifying the new data (inference phase)

Classification Algorithms

Discriminative v.s. Generative

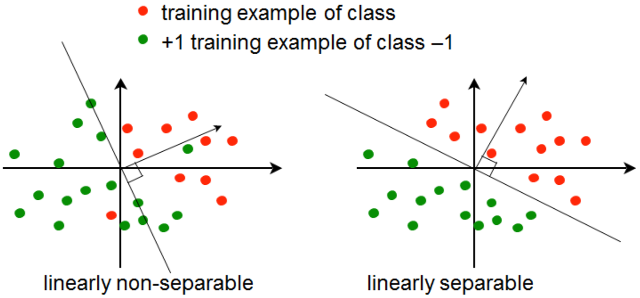
- Discriminative models capture $P(Y \mid X)$ directly and generative models capture $P(X)$, $P(y)$
- Generative models are computational more expensive

Classification Algorithms

- Logistic regression (Discriminative)
- Decision Tree (Discriminative)
- Linear Discriminant analysis (Generative)
- Quadratic Discriminant analysis (Generative)
- Support Vector Machine (Discriminative)

Linear Classification

- What is meant by linear classification?
  - The decision boundaries in the feature (input) space is linear
- May seem ‘too simple’, but in high-dimensions it is surprisingly powerful



Linear Classification

- Training data (x_i, y_i) for $i = 1 \ldots N$, and we wish to predict y' corresponding to new data vectors x'
- $x_i \in R^n$, and $y_i \in \{0, 1\}$
- Predicted value $\hat{y}$ is given by taking the sign of a linear function of the input:

$$\hat{y} = sign(\theta^T x), sign(z) = \begin{cases} 0 & if\ z \geq 0 \\ 1 & if\ z < 0 \end{cases}$$

Logistic Regression

Logistic Regression

- Classification based on probability
- Instead of just predicting the class, give the probability of the instance being that class
  - Directly learn conditional probability $Pr(G = k|X = x) = p_\theta(x)$
- Focus on binary classification

Simple Example

Cancer diagnosis from tumor size.

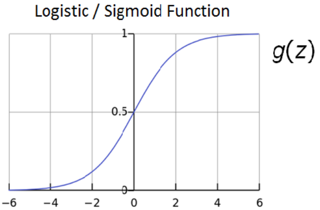
$x = \begin{bmatrix} x_0 \\ x \end{bmatrix} = \begin{bmatrix} 1 \\ Tummor\ Size \end{bmatrix}$

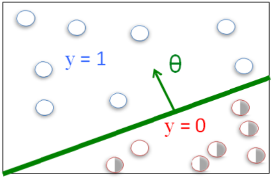
$\Longrightarrow p(x) = 0.7$

$\Longrightarrow$ Patient that 70% chance of tumor being malignant

Probability to Classification

- Logistic regression model assumes
  - $p_\theta = g(\theta^T x)$
  - $g(Z) = \frac{1}{1+\exp(-Z)}$
- Assume a threshold and
  - Predict $y = 1$, if $p_\theta \geq 0.5$
  - Predict $y = 0$, if $p_\theta < 0.5$





Assumptions to Model

- Logisitic regression model

$$p_\theta(x) = \frac{1}{1 + \exp\left(-\theta^T x\right)}$$

$$\log \frac{Pr(y = 1 \mid x; \theta)}{Pr(y = 0 \mid x; \theta)} = \theta_0 + \theta_1 x_1 + \ldots \theta_d x_d$$

- In other words, logistic regression assumes that the log odds is a linear function of

Comparison with Linear Regression on Indicators

- Similarities:
  - Both attempt to estimate$Pr(G = k \mid X = x)$
  - Both have linear classification boundaries
- Differences:
  - Linear regression on indicator matrix: approximate $Pr(G = k \mid X = x)$ by a linear function of x
    $Pr(G = k \mid X = x)$ is not guaranteed to fall between 0 and 1 and to sum up to 1
  - Logistic regression: $Pr(G = k \mid X = x)$ is a nonlinear function of x . It is guaranteed to range from 0 to 1 and to sum up to 1.

Fitting Logistic Regression

- Criteria: find parameters that maximize the conditional likelihood of G given X using the training data.
  - Denote $p_k(x_i;\theta) = Pr(G = k|X = x_i;\theta)$.
  - Given the first input x1, the posterior probability of its class being g_1 is $Pr(G = g1 \mid X = x1)$.
- Since samples in the training data set are independent, the posterior probability for the N samples each having class g_i , $i = 1, 2, \ldots, N$, given their inputs $x_1, x_2, \ldots, x_N$ is:

$$\prod_{i=1}^{N} Pr(G = g_i \mid X = x_i)$$

Conditional Log Likelihood

$$\max \prod_{i=1}^{N} Pr(G = g_i \mid X = x_i) = \max \log \prod_{i=1}^{N} Pr(G = g_i \mid X = x_i)$$

- The conditional log-likelihood of the class labels in the training data set is

$$\begin{aligned} L(\theta) &= \sum_{i=1}^{N} \log Pr(G = g_i \mid X = x_i) \\ &= \sum_{i=1}^{N} \log p_{g_i}(x_i; \theta)) \end{aligned}$$

i denotes the index of data

Derivation of $L(\theta)$ for Binary Classifier

- For binary classification, if $g_i = 1$, denote $y_i = 1$; if $g_i = 2$, denote $y_i = 0$.
- Let $p_1(x;\theta) = p(x;\theta)$, then $p_2(x;\theta) = 1 - p_1(x;\theta) = 1 - p(x;\theta)$
- Since $K = 2$, the parameters $\theta = \{\beta_{10}, \beta_1\}$. We denote $\beta = (\beta_{10}, \beta_1)^T$

Derivation of $L(\theta)$ for Binary Classifier

- If $y_i = 1$, i.e., $g_i = 1$,

$$\begin{aligned} \log p_{g_i}(x;\beta) &= \log p_1(x;\beta) \\ &= 1 \cdot \log p_1(x;\beta) \\ &= y_i \cdot \log p_1(x;\beta) \end{aligned}$$

- If $y_i = 0$, i.e., $g_i = 2$,

$$\begin{aligned} \log p_{g_i}(x;\beta) &= \log p_2(x;\beta) \\ &= 1 \cdot \log(1 - p_1(x;\beta)) \\ &= (1 - y_i) \cdot \log(1 - p(x;\beta)) \end{aligned}$$

- Since either $y_i = 0$ or $1 - y_i = 0$, we have

$$\log p_{g_i}(x;\beta) = y_i \cdot \log p(x;\beta) + (1 - y_i) \cdot \log(1 - p(x;\beta))$$

Binary Classifier of Logistic Regression

- Assumptions of Logistic Regression

$$\begin{aligned} p(x;\beta) &= Pr(G=1 \mid X=x) = \frac{\exp\left(\beta^T x\right)}{1+\exp\left(\beta^T x\right)} \\ 1-p(x;\beta) &= Pr(G=2 \mid X=x) = \frac{1}{1+\exp\left(\beta^T x\right)} \end{aligned}$$

- The conditional likelihood

$$L(\beta) = \sum_{i=1}^{N} \log p_{g_i}(x_i;\beta)$$

- Put together

$$L(\beta) = \sum_i y_i \beta^T x_i - \log\left(1+\exp \beta^T x_i\right)$$

Finding β

How to solve
$\max_\beta L(\beta) = \max_\beta \sum_i^N y_i\beta^T x_i - \log\left(1 + \exp \beta^T x_i\right)$

- Non-linear optimization
- Stochastic gradient decent

Example

- Diabetes data set. Input X is two dimensional. X_1 and X_2 are the two principal components of the original 8 variables.
- Class 1: without diabetes; Class 2: with diabetes.
- Applying logistic regression, we obtain

$$\beta = (0.7679, -0.6816, -0.3664)^T$$

Example

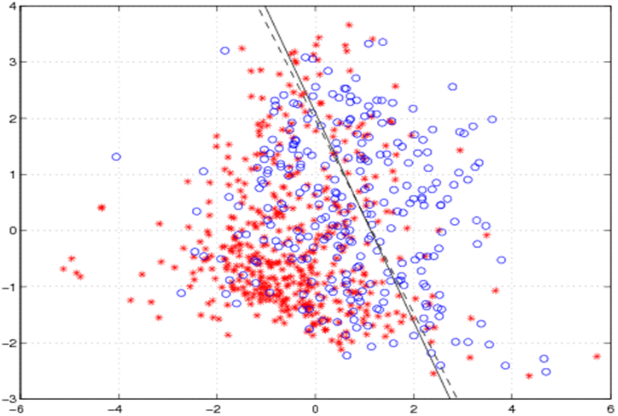
- The posterior probabilities

$$
\begin{aligned}
Pr(G = 1 \mid X = x) &= \frac{\exp\left(0.77 - 0.68x_1 - 0.37x_2\right)}{1 + \exp\left(0.77 - 0.68x_1 - 0.37x_2\right)} \\
Pr(G = 2 \mid X = x) &= \frac{1}{1 + \exp\left(0.77 - 0.68x_1 - 0.37x_2\right)}
\end{aligned}
$$

- The classification rule

$$
G\hat{(x)} = \begin{cases} 0 & \textit{if } 10.77 - 0.68x_1 - 0.37x_2 \geq 0 \\ 1 & \textit{if } 10.77 - 0.68x_1 - 0.37x_2 < 0 \end{cases}
$$

Example



Solid line: decision boundary obtained by logistic regression.
Dash line: decision boundary obtained by LDA (next topic).

Discriminant Analysis

Gaussian Discriminant Analysis

- Assume distribution of underlaying data
- Robust for multi-classes analysis
- Classifiers can be obtained by closed form solutions
- And, we look into $X = (X_1, X_2 \dots X_D)$

Notion

- The prior probability of class k is π_k
  - π_k is usually estimated simply by empirical frequencies of the training set

$$\pi_k = \frac{\text{no. of samples in class k}}{\text{Total no. of samples}}$$

- The class-conditional density of X in class $G = k$ is $f_k(x)$
- Computing the posterior probability

$$Pr(G = k \mid X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_k(x)\pi_l}$$

- Obtain classifier By MAP (the Bayes rule for 0-1 loss)

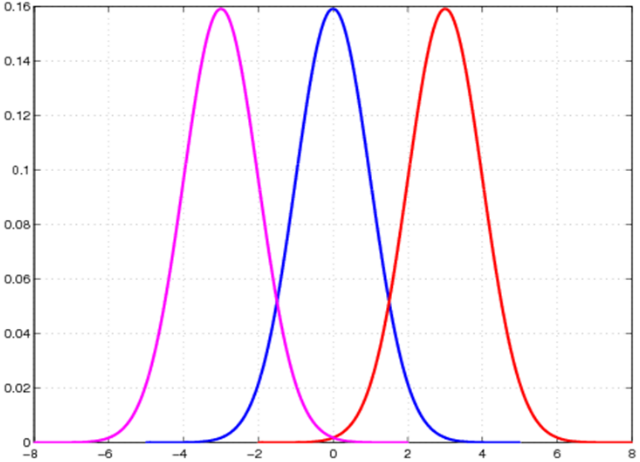
$$\begin{aligned} G\hat{(}x) &= \arg\max_k Pr(G = k \mid X = x) \\ &= \arg\max_k f_k(x)\pi_k \end{aligned}$$

Linear Discriminant Analysis

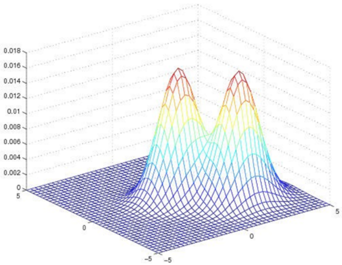
- Assumption 1: multivariate gaussian distribution
  - $$f_k(x) = \frac{1}{(2\pi)^{p/2}} \exp\left(-\frac{1}{2}(x-\mu_k)^T \sum_k^{-1}(x-\mu_k)\right)$$

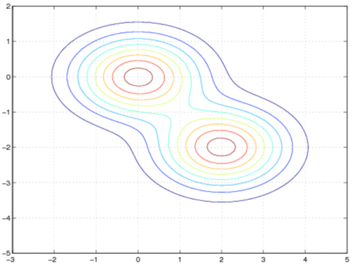
    E.g., Bivariate normal distribution
    $\mu_k = (0, 0)$, and $\Sigma_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$
- Assumption 2: $\Sigma_k = \Sigma$, $\forall k$
  - The Gaussian distributions are shifted versions of each other.

Gaussian Distributions: Univariate



Gaussian Distributions: Bivariate





Optimal Classifier $G\hat{(}x)$

$$
\begin{aligned}
G\hat{(}x) &= \arg\max_k Pr(G = k \mid X = x) \\
&= \arg\max_k f_k(x)\pi_k = \arg\max_k \log\left(f_k(x)\pi_k\right) \\
&= \arg\max_k \Big[-\log((2\pi)^{p/2}|\Sigma|^{1/2}) \\
&\quad -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log(\pi_k)\Big] \\
&= \arg\max_k -\frac{1}{2}(x-\mu_k)^T\Sigma^{-1}(x-\mu_k) + \log(\pi_k)
\end{aligned}
$$

Optimal Classifier $\hat{G(x)}$

- Define the linear discriminant function

$$\hat{G(x)} = \arg\max_k [x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log(\pi_k)]$$

- Then Classification rule

$$\hat{G(x)} = \arg\max_k \delta_k(x)$$

- The decision boundary between class k and l is:

$$\{x : \delta_k(x) = \delta_l x\}$$

- Or equivalently the following holds

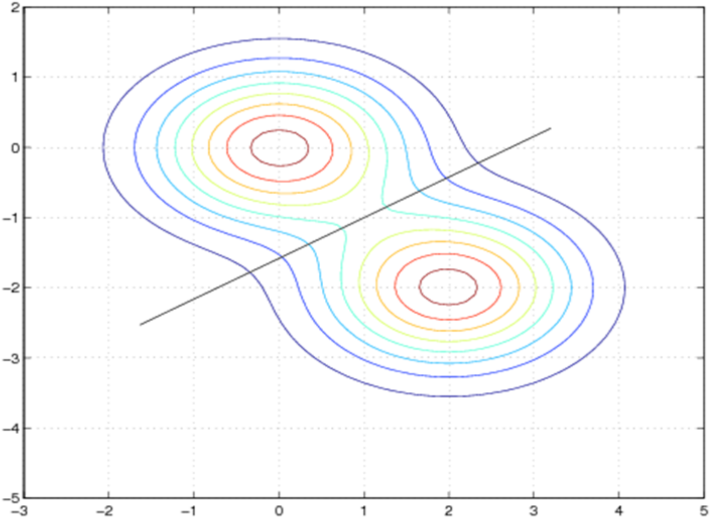
$$\log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1} (\mu_k - \mu_l) + x^T \Sigma^{-1} (\mu_k - \mu_l) = 0$$

A Numerical Example

- Binary classification ($k = 1$, $l = 2$)
  - Define $a_0 = \log \frac{\pi_1}{\pi_2} - \frac{1}{2}(\mu_1 + \mu_2)^T \Sigma^{-1}(\mu_1 - \mu_2)$
  - Define $(a_1, a_2, \dots a_D)^T = \Sigma^{-1}(\mu_1 - \mu_2)$
  - Classify to class 1 if $a + \sum_{j=1}^{D} a_j x_k > 0$; to class 2 otherwise
- Example
  - $\pi_1 = \pi_2 = 0.5$, $\mu_1 = (0, 0)^T$, and $\mu_2 = (0, 0)^T$
  - $\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 0.56 \end{bmatrix}$
  - Decision boundary

$$5.56 - 2.00x_1 + 3.56x_2 = 0$$

Example



Estimating Gaussian Distributions

- We need to estimate the Gaussian distribution

$$\hat{\pi}_k = N_k/N,$$

$$\hat{\mu}_k = \sum_{g_i=k} x^{(i)}/N_k$$

$$\hat{\Sigma} = \sum_{k=1}^{K} \sum_{g_i=k} (x^{(i)} - \hat{\mu}_k)(x^{(i)} - \hat{\mu}_k)^T/(N-K)$$

N_k is the number of class-k samples
$x^{(i)}$ denotes the $i^{(th)}$ sample vector

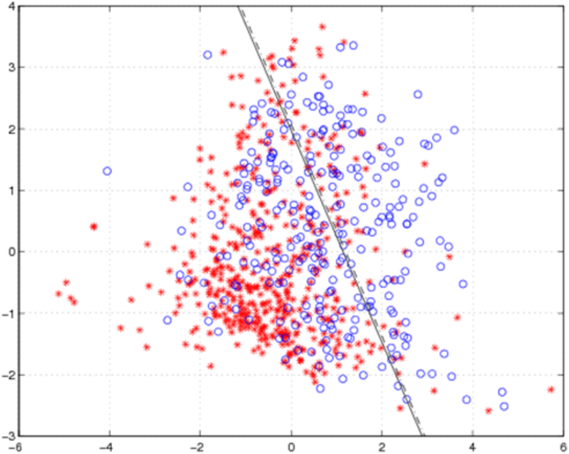
Diabetes Data Set

- Two input variables computed from the principal components of the original 8 variables.
- Prior probabilities: $\hat{\pi}_1 = 0.651$, $\hat{\pi}_2 = 0.349$
  $\hat{\mu}_1 = (-0.43, -0.19)^T$, $\hat{\mu}_2 = (0.75, 0.36)^T$
  $\Sigma = \begin{bmatrix} 1.79 & -0.15 \\ -0.15 & 1.66 \end{bmatrix}$
- Classification rules:

$$\hat{G(x)} = \begin{cases} 1 & 0.77 - 0.67x_1 - 0.39x_2 \geq 0 \\ 2 & \text{otherwise} \end{cases}$$

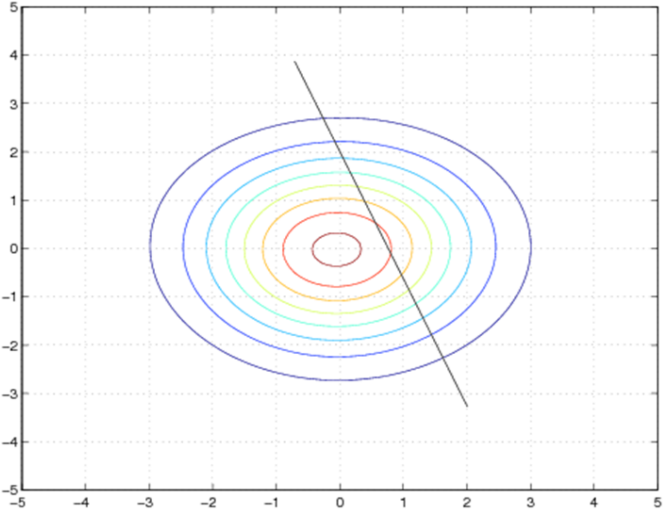
Example

The scatter plot follows. Without diabetes: stars (class 1), with diabetes: circles (class 2). Solid line: classification boundary obtained by LDA. Dash dot line: boundary obtained by linear regression of indicator matrix.



Contour Plot

Contour plot for the density (mixture of two Gaussians) of the diabetes data

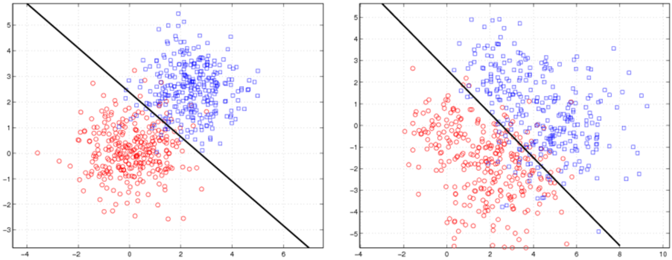


Poor Results of LDA

- LDA is not necessarily bad when the assumptions about the density functions are violated.
- In certain cases, LDA may yield poor results.

Poor Results of LDA

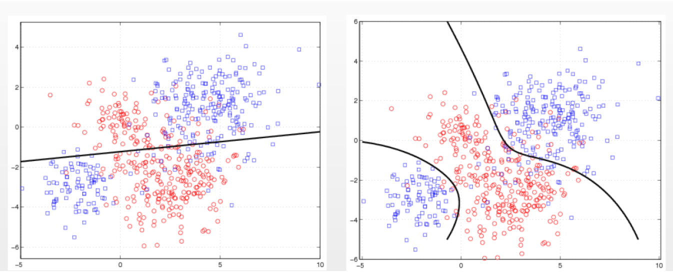
.



Left: The true within class densities are Gaussian with identical covariance matrices across classes. Right: The true within class densities are mixtures of two Gaussians

Poor Results of LDA

.



Left: Decision boundary by LDA. Right: Decision boundaries obtained by modeling each class by a mixture of two Gaussian

Quadratic Discriminant analysis

- Estimate the covariance matrix Σ_k separately for each class k, $k = 1, 2, \ldots, K$.
- Quadratic discriminant function:

$$\delta_k(x) = \frac{1}{2}\log|\Sigma_k| - \frac{1}{2}(x-\mu_k)^T {\Sigma_k}^{-1}(x-\mu_k) + \log\pi_k$$

- Classification rule:

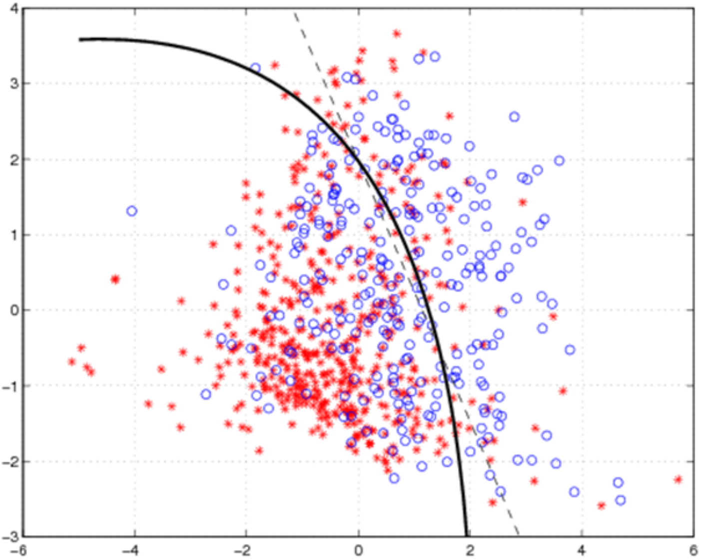
$$G\hat{(}x) = \arg\max_k \delta_k(x)$$

- Decision boundaries are quadratic equations in x .
- QDA fits the data better than LDA, but has more parameters to estimate.

Diabetes Data Set

- Prior probabilities: $\pi_1 = 0.651$, $\pi_2 = 0.349$
- Prior probabilities: $\hat{\pi}_1 = 0.651$, $\hat{\pi}_2 = 0.349$
- $\hat{\mu_1} = (-0.4035, -0.1935)^T$, $\hat{\mu_2} = (0.7528, 0.3611)^T$
- $\hat{\Sigma_1} = \begin{bmatrix} 1.6769 & -0.1461 \\ -0.1461 & 1.5964 \end{bmatrix}$
- $\hat{\Sigma_2} = \begin{bmatrix} 2.0087 & -0.3330 \\ -0.3330 & 1.7887 \end{bmatrix}$

Example: QDA



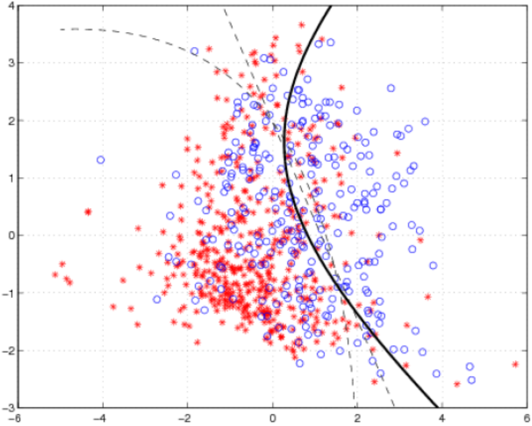
LDA on Expanded Bases

- Expand input space to include X_1, X_2, X_2, and X_2.
- Input is five dimensional: $X = (X_1, X_2, X_1 X_2, X_1^2, X_2^2)$.
- Classification boundary:

$$0.65 - 0.73x_1 - 0.55x_2 - 0.006x_1x_2 - 0.07x_1^2 + 0.17x_2^2$$

- If the linear function on the right hand side is non-negative, classify as 1; otherwise 2

Example: ELDA



Classification boundaries obtained by LDA using the expanded input space $X_1, X_2, X_1X_2, X_1^2, X_1^2$. Boundaries obtained by LDA and QDA

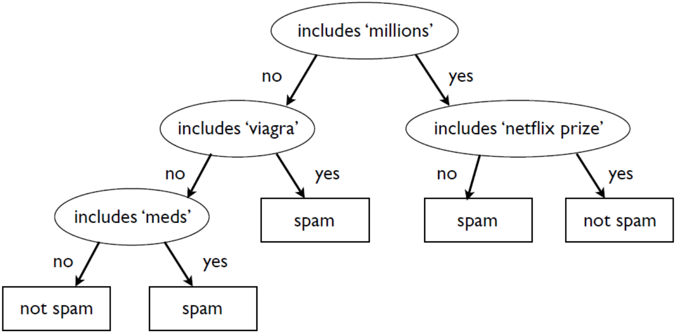
Other Methods

Other Learning Approaches

- Rule based methods
  - Decision Tree
  - Random Forest
- Perceptron methods
  - Support Vector Machines
  - Neural Networks

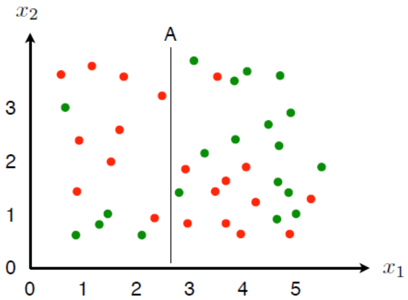
Decision Tree

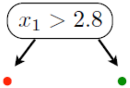
- Idea: Ask a sequence of questions (as in the '20 questions' game) to infer the class



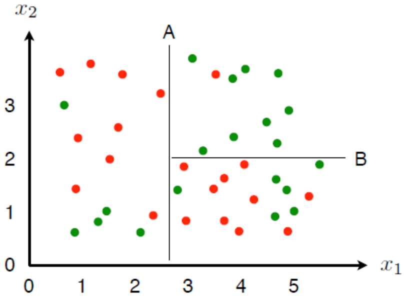
Decision Tree: General Ideas I

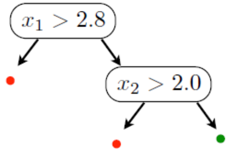
- Simple idea: recursively divide up the space into pieces which are as pure as possible



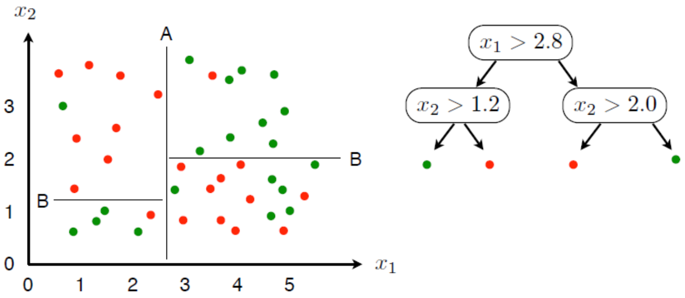


Decision Tree: General Ideas II

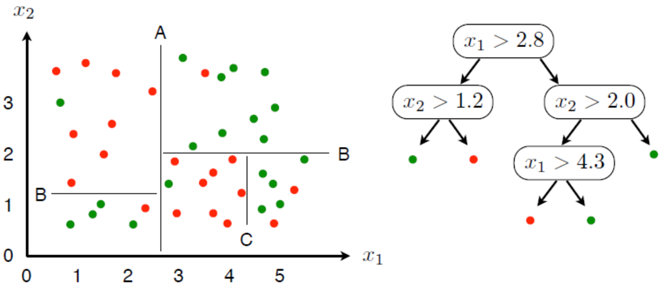




Decision Tree: General Ideas III



Decision Tree: General Ideas IV



Decision Tree: General Ideas

- Key steps to build the tree
  1. A goodness of split criterion that can be evaluated for any split s of any node.
  2. A stop-splitting rule.
  3. A rule for assigning every terminal node to a class.
- How to measure if a subset of records is 'pure' or 'impure'
  - Entropy
  - Gini index
  - Classification error

Characteristics of Decision Tree

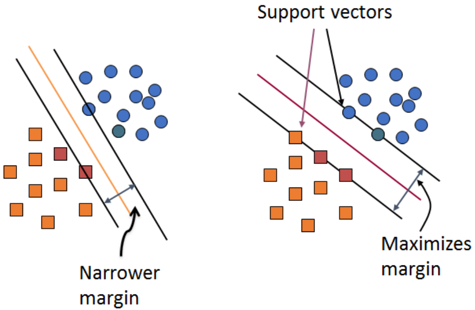
- Nonparametric approach
  - Can approximate any decision boundary (hyper plane) to arbitrary precision
  - A practical starting point
- Local, greedy learning to find a reasonable solution in reasonable time
- Relatively easy to interpret (by experts or regular users of the system)
- Data fragmentation problem: the nodes far down the tree are based on a very small fraction of the data, even only on a few data points) typically not very reliable information

Support Vector Machine

- Construct linear decision boundaries that explicitly try to separate the data into different classes as well as possible.
- Good separation is defined in a certain form mathematically.
- Even when the training data can be perfectly separated by hyperplanes, LDA or other linear methods developed under a statistical framework may not achieve perfect separation.

Optimal Separating Hyperplane

- The optimal separating hyperplane separates the two classes and maximizes the distance to the closest point from either class.



Key Ideas of SVM

- Seek large margin separator to improve generalization
- Use optimization to find optimal hyperplane with few errors via slack variables