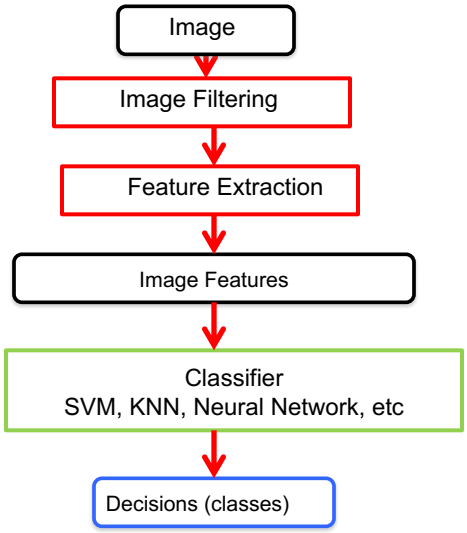
# ECE 5470 Lecture 16a
## Machine Learning
## (Statistical Pattern Recognition)



## Image Analysis

1. Image processing (filtering)
2. Segmentation
   (object, region of interest)
3. Feature Extraction
   or Measurement
4. Classification



# Topics: Pattern Recognition

- Pattern Recognition Examples
- Discriminant Functions
- Parametric Classifiers
- Non-Parametric Classifiers
- Neural Networks



# Statistical Pattern Recognition

- Concept:
  - Represent an entity by a feature vector $\underline{x}$
  - Determine a strategy to find the class of the entity from its feature vector such that the classification error is minimized
- Definition

  $w_i$ is a "state of nature" or class
- Problem

  find the most probable $w_i$ for a given $\underline{x}$

# Features

- Shape: length, width
- Surface:  brightness(density) surface texture, color
- Temporal: motion of image (growth)



# Datasets and Dataset Issues

- Datasets, for supervised learning, consist of examples (x) with associated labels or classes $w_i$ (or y)
- A classifier is trained with a **train dataset** and is then evaluated with a **test dataset.**
- Examples form the test dataset must not be used, even indirectly, for training the classifier, else the evaluation will not be valid.
- Dataset balancing: should the number of instances of each class be representative of it's natural occurrence?

Iris: three classes, 150 examples, 50 for each class, test/train partition not specified
MNIST: 10 classes, 70,000 examples, 7000 for each class, 60,000 train, 10,000 test

WIKIPEDIA
The Free Encyclopedia

# Iris Dataset

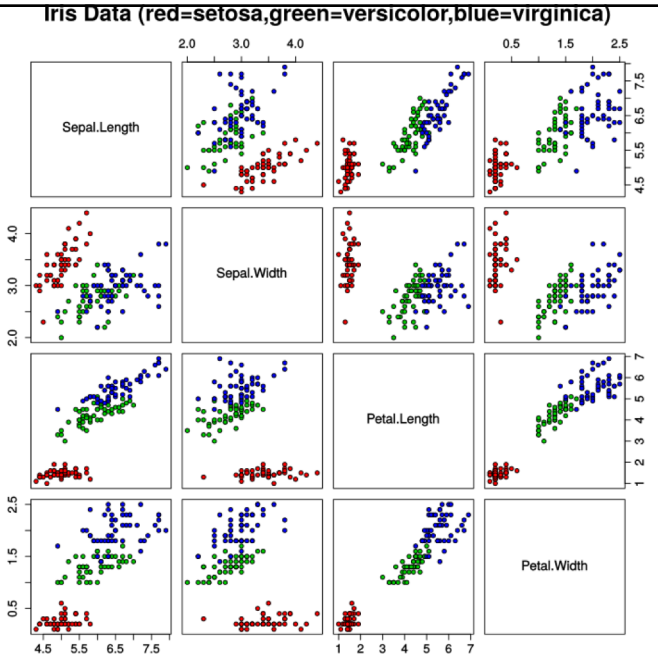
The ***Iris* flower data set** or **Fisher's *Iris* data set** is a multivariate data set introduced by the British statistician and biologist Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems* as an example of linear discriminant analysis.

The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*).
Four (4) features were measured from each sample: the length and the width of the sepals and petals, in centimeters.


*Iris setosa*

*Iris versicolor*

*Iris virginica*

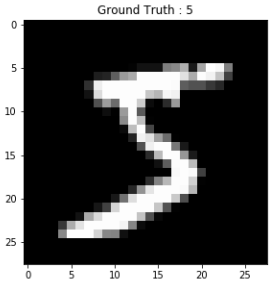



4-dimensional hyperspace

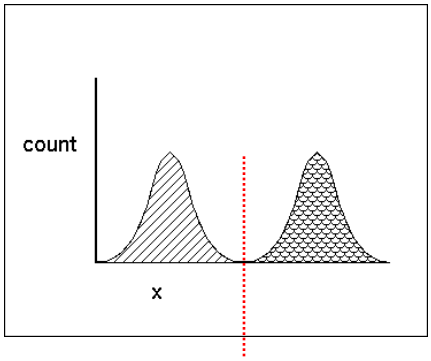
## Modified National Institute of Science and Technology (MNIST) Dataset

Original Images
1. Binary Images of handwritten digits (0-9)
2. Training set 60,000
3. Test set 10,000
4. 10 classes evenly distributed through both training and test sets

Image Preprocessing (for Modified dataset):
1. Size normalization to fit into a 20 x 20 pixel Bounding Box. Some resulting pixels have grey levels (not 0 or 255) due to interpolation.
2. Centered in a 28 x 28 image by computing the center of mass (COM) of the pixels and translating the image so the COM is at the image center.



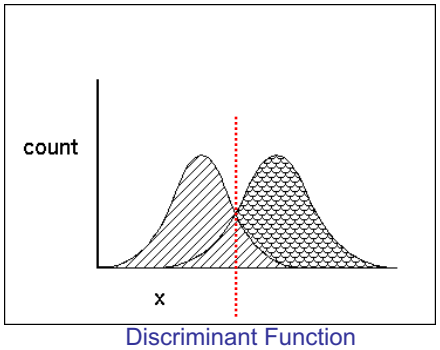

## Example 1: Image Intensity




Discriminant Function

Task: Select the best threshold (to separate foreground from background)

# Example 1a: Image Intensity

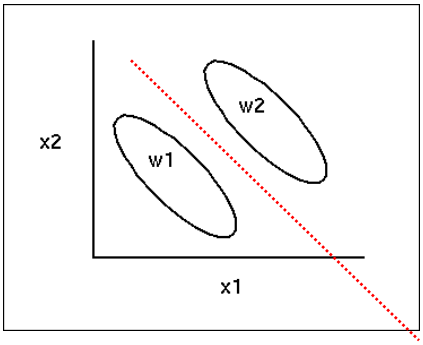



Discriminant Function

What to do when distributions overlap?



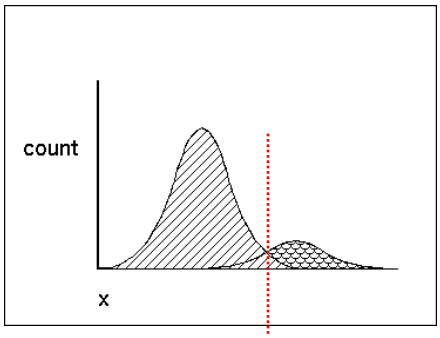
# Example 2: Colors




Discriminant Function

The distributions may be *separable* when more than one parameter is considered.

## Example 1b: Image Intensity




Discriminant Function

What to do when one distribution is more probable than the other?



## Bayes' Theorem

Given the class conditional probability density functions $p(\underline{x} \mid w_i)$

Given the a priori probability of $w_i$, $P(w_i)$

The a posteriori probability of a state is given by:

$$P(w_i \mid \underline{x}) = \frac{p(\underline{x} \mid w_i)P(w_i)}{p(\underline{x})}$$

Where

$$p(\underline{x}) = \sum_i p(\underline{x} \mid w_i)P(w_i)$$

# Discriminant Function

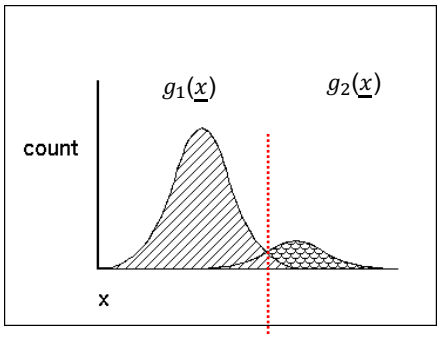
Since p(x) is the same for all $w_i$ it behaves as a scale factor which may be neglected when designing a decision rule

Define a discriminant function

$$g_i(\underline{x}) = P(w_i \mid \underline{x})$$

Decision Rule:

choose the class for which $g_i(\underline{x})$ is the largest




Discriminant Function



# Discriminant Function

Consider as a simplification that all classes are equally likely

$$P(w_i) = P(w_j) \forall i, j$$

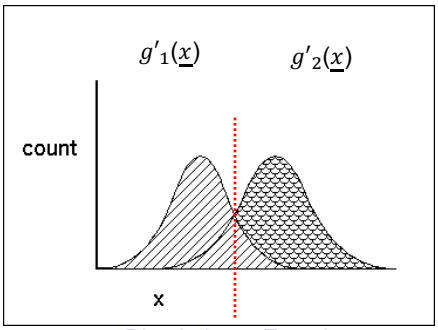
Define a new discriminant function

$$g_i'(\underline{x}) = p(\underline{x} \mid w_i)$$

Decision Rule:

choose the class for which $g_i(\underline{x})$ is the largest

Problem: How to determine $p(\underline{x} \mid w_i)$ ?



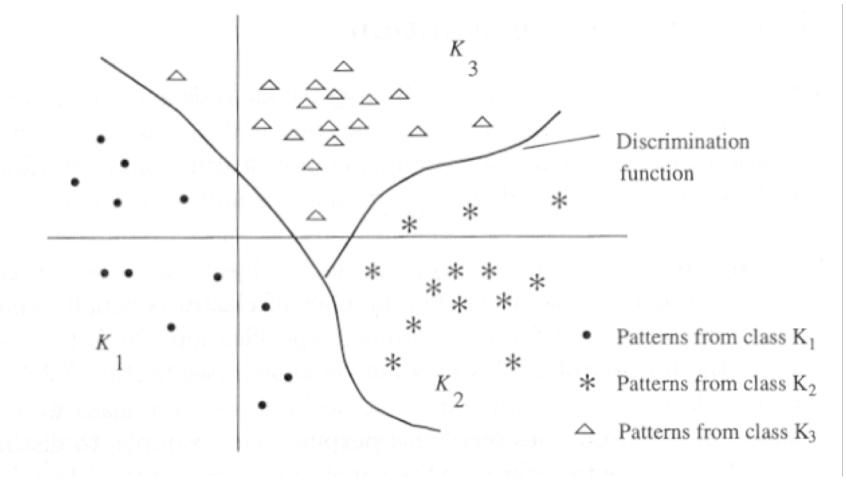

Discriminant Function

# Topics: Pattern Recognition

- Pattern Recognition Examples
- Discriminant Functions
- Parametric Classifiers
- Logisitic Regression
- Non-Parametric Classifiers
- Neural Networks



# General Multi-class Discrimination Functions



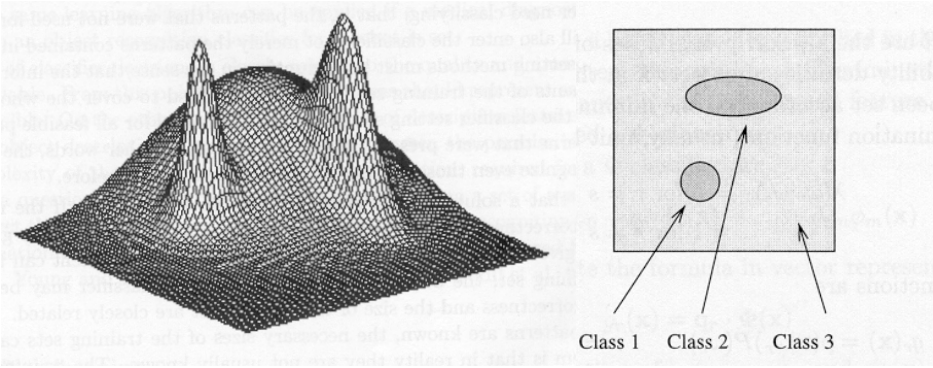
## Linear Discriminant Functions

Straight line partitions of hyperspace may be achieved with linear discriminant functions which have the form

$$g_i(\underline{x}) = q_{r0} + q_{r1}x_1 + \cdots + q_{rn}x_n$$



## Minimum error classifier: Gaussian distributions



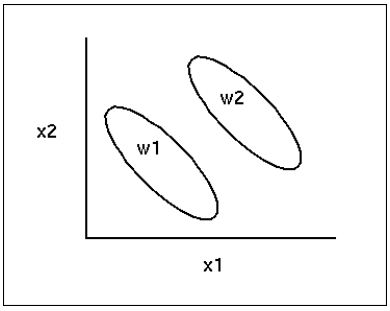
## Grand Assumption

Problem: How to determine $p(\underline{x} \mid w_i)$ ?

Grand Assumption:

$p(\underline{x})$ is a multivariate Gaussian distribution

We can specify $p(\underline{x})$ by the mean $\underline{\mu}$ and the covariance matrix $\Sigma$






## Multivariate Gaussian Distribution

Given $p(\underline{x})$ is a multivariate Gaussian distribution

$$p(\underline{x}) = (2\pi)^{-n/2}|\Sigma|^{-1/2}\exp\left(-\tfrac{1}{2}(\underline{x}-\underline{\mu})^T\Sigma^{-1}(\underline{x}-\underline{\mu})\right)$$

Specification of the distribution requires:

n + n(n+1)/2 values

$\underline{\mu}$ $\Sigma$

Methods: directly from training set compute $\underline{\mu}$, use SVD for $\Sigma^{-1}$

## Multivariate Gaussian Distribution

Define $g_i''(\underline{x}) = \log p(\underline{x} \mid w_i)$

Any monotonic function of $p(\underline{x} \mid w_i)$ is OK

then

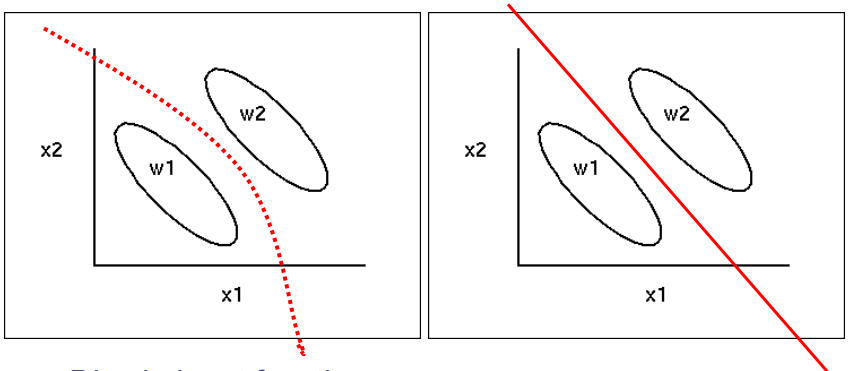
$$g_i''(\underline{x}) = \tfrac{1}{2}(\underline{x} - \underline{\mu_i})^T \Sigma_i^{-1}(\underline{x} - \underline{\mu_i}) - \tfrac{n}{2}\log(2\pi) - \log|\Sigma_i|$$

constant

Constant if the same for all $w_i$



## Multivariate Gaussian Distribution




Discriminant function is a hyperquadratic

$\Sigma = I$ implies a linear discriminant function

# Multivariate Gaussian Distribution

Main problem is the estimation of $\underline{\mu}$ and $\Sigma$.

Consider:

n=6 $\Rightarrow$ 27 parameters

n=8 $\Rightarrow$ 44 parameters

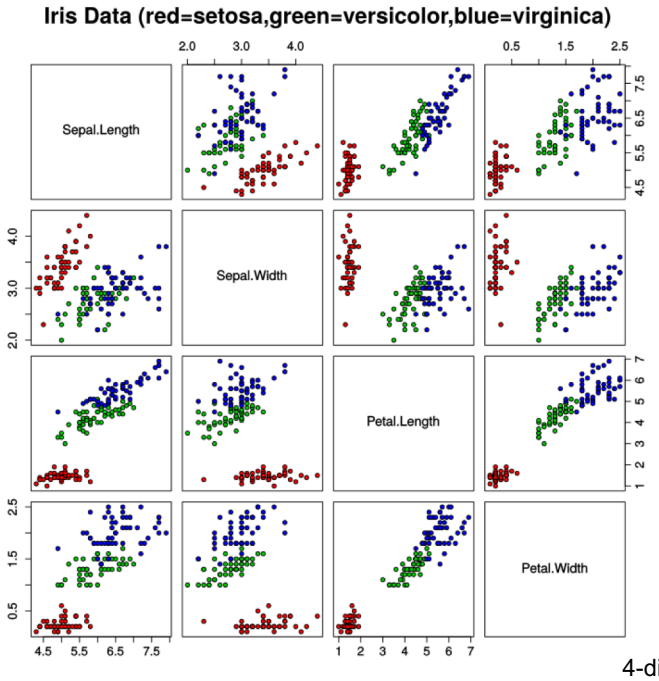
For a linear discriminant function

$$g_i''(\underline{x}) = \underline{m}_i^T \underline{x} + m_{i0}$$

where $m_{i0} = -\frac{1}{2} \underline{\mu}_i^T \underline{\mu}_i \qquad \underline{m}_i = \underline{\mu}_i$

n=6 $\Rightarrow$ 7 parameters






4 classes

n + n(n+1)/2
$\Rightarrow$ 14 parameters

4-dimensional hyperspace

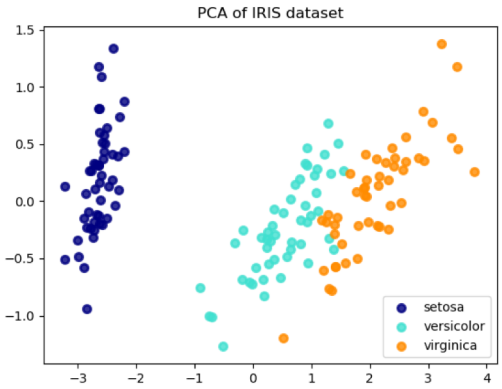
# Method: Feature Dimension Reduction

PCA Example

Iris dataset n=4

PCA is used to decompose a multivariate dataset in a set of successive orthogonal components that explain a maximum amount of the variance.

Dimension reduction **may** improve Performance of some classifiers




http://scikit-learn.org/stable/modules/decomposition.html#decompositions

# Methods: Feature Scaling

Feature scaling through standardization (or Z-score normalization) can be an important preprocessing step for many machine learning algorithms.

**Standardization** involves rescaling the features such that they have the properties of a standard normal distribution with a mean of zero and a standard deviation of one.

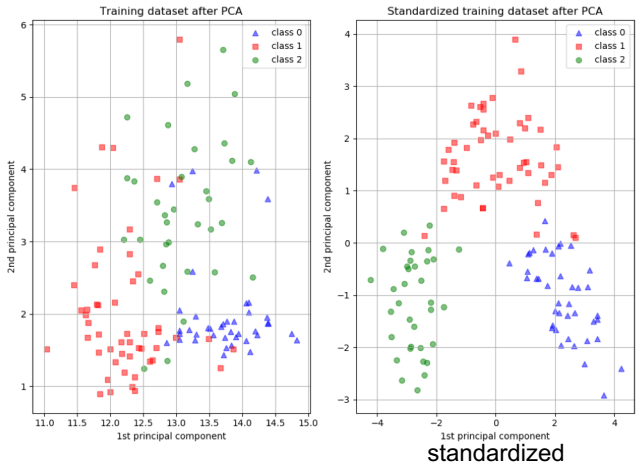
For each input vector $x_i$ $\quad x_i' = (x_i - \mu)/\sigma$

- Many algorithms (such as SVM, K-nearest neighbors, and logistic regression) may benefit from scaling (depending on the task and the feature set).
- Scaling is especially Important when inputs have different scales or units.
- For standardization, the values of $\mu$ and $\sigma$ must be computed from the full training set, when possible, and saved for when the classifier is used or tested.
- Principle Component Analysis (PCA) as being a prime example of when normalization is important.
- Other scaling methods include: min-max normalization, and mean normalization.

http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html#sphx-glr-auto-examples-preprocessing-plot-scaling-importance-py

## Impact of standardization on PCA for Wine dataset

The dataset used is the Wine Dataset available at UCI. This dataset has continuous features that are heterogeneous in scale due to differing properties that they measure (i.e., alcohol content, and malic acid). 13 features, 3 cultivars




standardized

May improve performance on some datasets and classifiers

1) Alcohol
2) Malic acid
3) Ash
4) Alcalinity of ash
5) Magnesium
6) Total phenols
7) Flavanoids
8) Nonflavanoid phenols
9) Proanthocyanins
10)Color intensity
11)Hue
12)OD280/OD315 of diluted wines
13)Proline

2) N=178

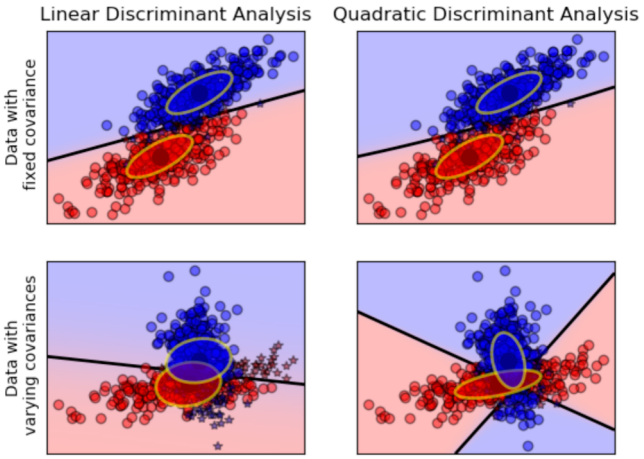
http://scikit-learn.org/stable/auto_examples/preprocessing/plot_scaling_importance.html#sphx-glr-auto-examples-preprocessing-plot-scaling-importance-py

scikit learn

## Examples

LDA: Linear Discriminant Analysis

QDA: Quadratic Discriminant Analysis




This example plots the covariance ellipsoids of each class and decision boundary learned by LDA and QDA. The ellipsoids display the double standard deviation for each class. With LDA, the standard deviation is the same for all the classes, while each class has its own standard deviation with QDA.

http://scikit-learn.org/stable/auto_examples/classification/plot_lda_qda.html#sphx-glr-auto-examples-classification-plot-lda-qda-py

# Topics: Pattern Recognition

- Pattern Recognition Examples
- Discriminant Functions
- Parametric Classifiers
- Logisitic Regression
- Non-Parametric Classifiers
- Neural Networks



WIKIPEDIA
The Free Encyclopedia

## Regression Analysis

In **statistical modeling**, **regression analysis** is a set of statistical processes for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between **a dependent variable** and **one or more independent variables** (or 'predictors').

More specifically, regression analysis helps one understand how the typical value of the dependent variable (or 'criterion variable') changes when any one of the independent variables is varied, while the other independent variables are held fixed.

A function of the independent variables called the **regression function** is to be estimated.

**Historical Note:**
The term "regression" was coined by Francis Galton in the nineteenth century to describe a biological phenomenon. The phenomenon was that the heights of descendants of tall ancestors tend to regress down towards a normal average (a phenomenon also known as regression toward the mean)

## Regression Models

Regression models involve the following parameters and variables:
•The **unknown parameters**, denoted as $\beta$, which may represent a scalar or a vector.
•The **independent variables**, $X$.
•The **dependent variable**, $Y$.

A regression model relates $Y$ to a function of $X$ and $\beta$.

$$Y \approx f(X, \beta)$$

The approximation is usually formalized as:

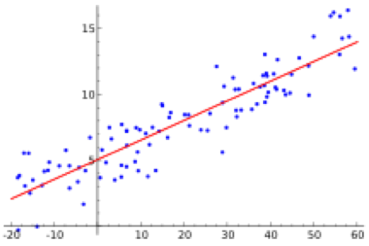
$$\mathrm{E}(Y|X) = f(X, \beta)$$

For linear models estimation methods such as ordinary least squares are used to find $\beta$
**When the model function is not linear in the parameters, the sum of squares must be minimized by an iterative procedure.**

## Linear Regression

**Linear Regression**
Find optimal weights for a linear model for a continuous valued output

Not a categorical classification, however, one could add a threshold on the output to achieve a classification.




Simple Linear Regression e.g., two independent variables

straight line: $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i, \quad i = 1, \ldots, n.$

Multiple Linear Regression

parabola: $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \varepsilon_i, \; i = 1, \ldots, n.$

Still linear in $\beta$

## Logistic Regression

Two class problems

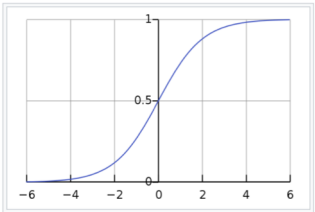


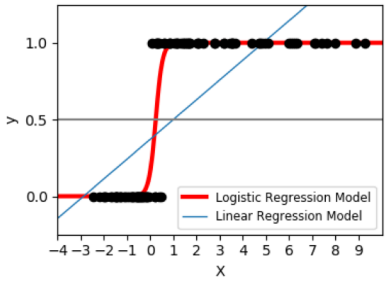

Figure 1. The standard logistic function $\sigma(t)$; note that $\sigma(t) \in (0,1)$ for all $t$.

$$\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$$

Given $t = \beta_0 + \beta_1 x$

Probability of 1 class:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$




The logarithm of the odds is the logit of the probability,

$$\operatorname{logit} p = \ln \frac{p}{1-p} \quad \text{for } 0 < p < 1.$$

Model $\operatorname{logit} \mathrm{E}(Y) = \alpha + \beta x$

Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.

## Multinomial Logistic Regression

K-class problems

For K classes create K linear models one for each class
Correct class is identified by the output with the highest probability

As in other forms of linear regression, multinomial logistic regression uses a linear predictor function to predict the probability that observation *i* has outcome *k*, of the following form:

$$f(k,i) = \beta_{0,k} + \beta_{1,k} x_{1,i} + \beta_{2,k} x_{2,i} + \cdots + \beta_{M,k} x_{M,i},$$

$$f(k,i) = \boldsymbol{\beta}_k \cdot \mathbf{x}_i,$$

Class probability is given by $\Pr(Y_i = c) = \dfrac{e^{\boldsymbol{\beta}_c \cdot \mathbf{X}_i}}{\sum_{k=1}^{K} e^{\boldsymbol{\beta}_k \cdot \mathbf{X}_i}}$

Define $\operatorname{softmax}(k, x_1, \ldots, x_n) = \dfrac{e^{x_k}}{\sum_{i=1}^{n} e^{x_i}}$

Notation change:
$Y_i \equiv w_i$

$$\Pr(Y_i = c) = \operatorname{softmax}(c, \boldsymbol{\beta}_1 \cdot \mathbf{X}_i, \ldots, \boldsymbol{\beta}_K \cdot \mathbf{X}_i)$$

Classification rule: choose c for which $\Pr(Y_i = c)$ is maximum

## Logistic Regression

**Notes for scikit-learn: LogisticRegression**

*class* sklearn.linear_model.LogisticRegression ()

1. Multiple iterative solvers supported
2. Fit Criterion
   a) No Regularization (set C=1e15)

$$\min_{w,c} \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

   b) L2 Regularization

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

   c) L1 Regularization

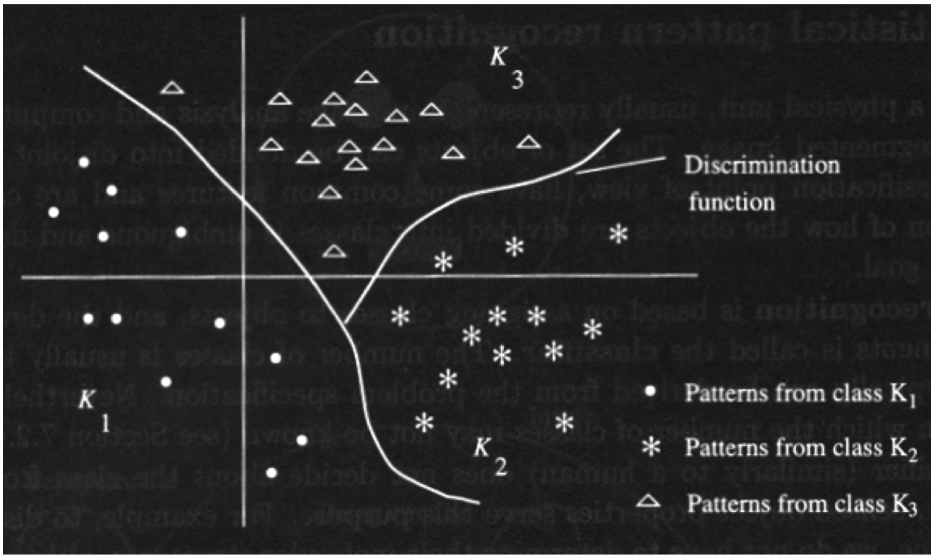
$$\min_{w,c} \|w\|_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

Notation change: Where *C* is the inverse of regularization strength,
*w* are the internal weights (previously $\beta$),
*c* is the bias
$y_i \in \{-1, +1\}$

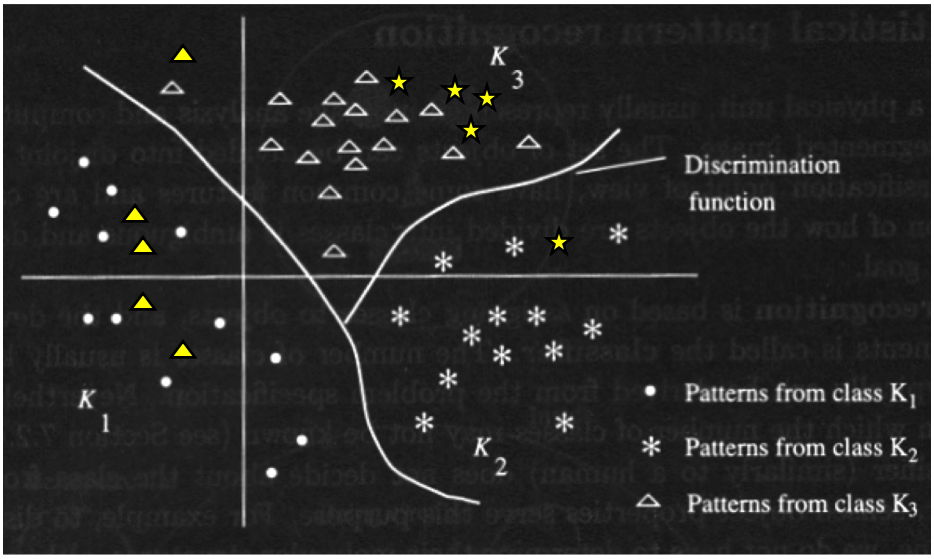
## Topics: Pattern Recognition

- Pattern Recognition Examples
- Discriminant Functions
- Parametric Classifiers
- Logisitic Regression
- Non-Parametric Classifiers
- Neural Networks

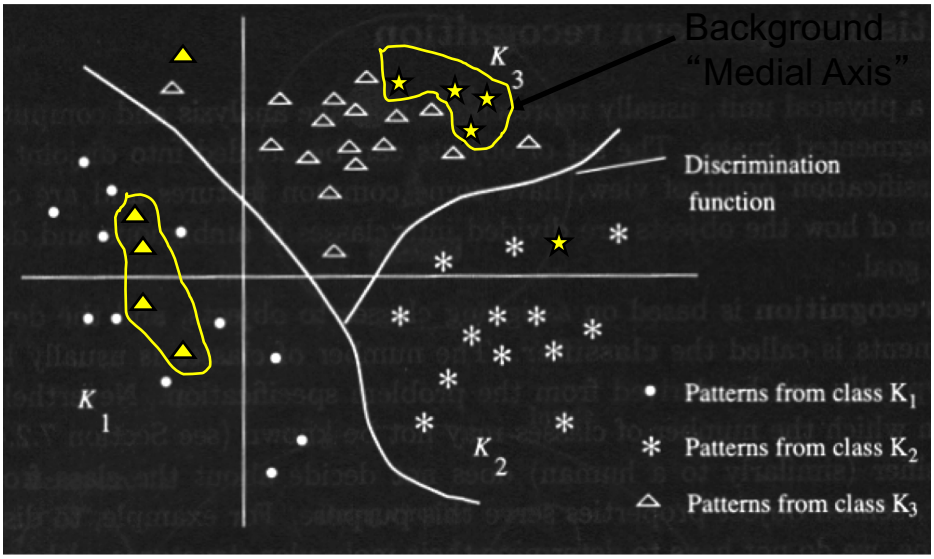
## Separable Problem




## Not Separable with Gaussian Model



## Non-parametric model




## Non-Parametric Classifier

The Near Neighbor Rule:

Given a training set H

For a vector $\underline{x}$ determine the vector in H that is closest to $\underline{x}$ denoted $\underline{h}$. The class of the vector $\underline{h}$ is assigned to $\underline{x}$.

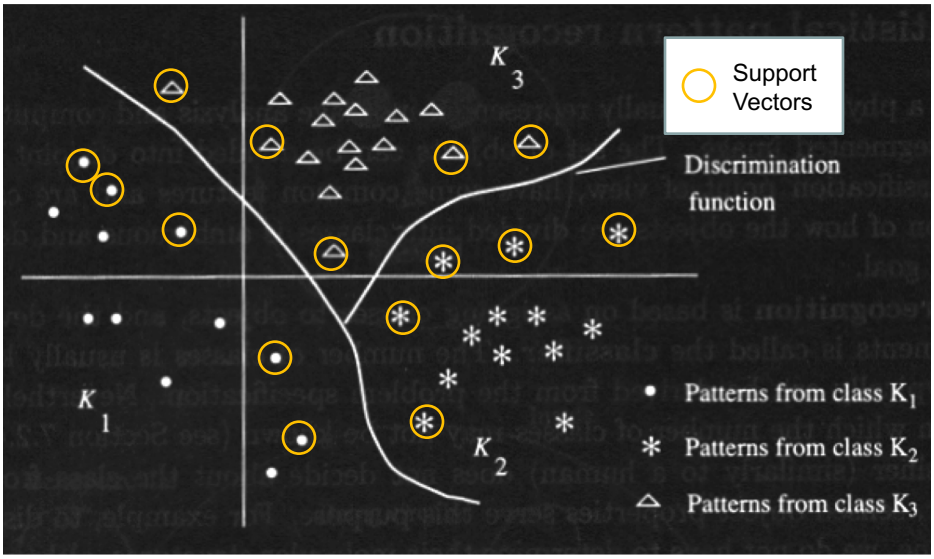
Advantage: decision surface is unconstrained

Disadvantage: decision time increases with the size of the training set.

Problem: Overlapping classes

→ use K-nearest neighbors

## Support Vector Machine (SVM)




## Support Vector Machine (SVM)

- Concept: find an "optimal" hyperplane that that has the largest distance to the nearest training data points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.
- The original linear (parametric) SVM results in a linear discriminant function
- Other SVM models use a higher dimension hyperspace (non-linear classification) e.g.,

- Gaussian Radial basis functions

$$k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$$

- Polynomial functions

$$k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j})^d$$

## Support Vector Machine (SVM)

We are given some training data $\mathcal{D}$, a set of *n* points of the form

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}\}_{i=1}^{n}$$

where the $y_i$ is either 1 or −1, indicating the class to which the point belongs. Each is a *p*-dimensional real vector. We want to find the maximum-margin hyperplane that divides the points having $y_i$ = 1 from those having $y_i$ = − 1. Any hyperplane can be written as the set of points satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0,$$

Minimize (in $\mathbf{w}, b$ )

$$\|\mathbf{w}\|$$

subject to (for any $i = 1, \ldots, n$)

$$y_i(\mathbf{w} \cdot \mathbf{x_i} - b) \geq 1.$$

*Wikipedia 2011*



## Linear Support Vector Machine

- Linear (parametric) SVM

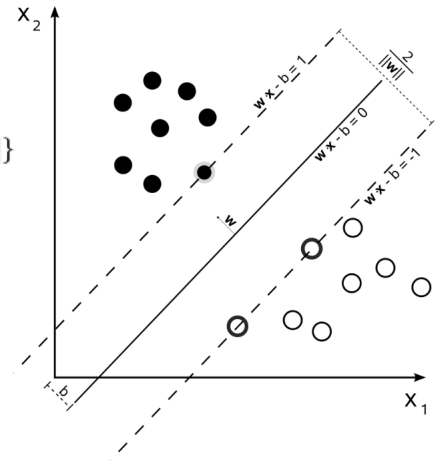
Training

$$\min_{\mathbf{w},b} \max_{\boldsymbol{\alpha}} \{\frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w} \cdot \mathbf{x_i} - b) - 1]\}$$

Solution may be expressed as

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x_i}$$

Where most $\alpha_i$ are zero




*Soft margin methods developed to handle "mislabeled" data*

## SVM: Gaussian Radial Basis Functions

Linear SVM: Any hyperplane can be written as the set of points satisfying

$$\mathbf{w} \cdot \mathbf{x} - b = 0,$$

Nonlinear SVM: the dot product is replaced by a kernel function

Gaussian radial basis functions $k(\mathbf{x_i}, \mathbf{x_j}) = \exp(-\gamma \|\mathbf{x_i} - \mathbf{x_j}\|^2)$

- This is much more complex to evaluate when trained and many more support vectors may be in involved
- Similar in concept to nearest neighbor

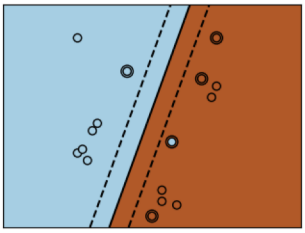
OR Polynomial Kernel Function $k(x_i, x_j) = (x_i^T x_j + c)^d$

Where *d* is the degree of the polynomial (typically *d* =2)

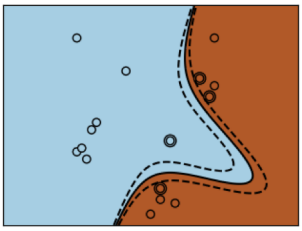
[ c.f. linear $k(\mathbf{x_i}, \mathbf{x_j}) = (\mathbf{x_i} \cdot \mathbf{x_j})$ ]



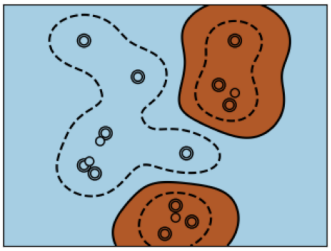
## SVM: Kernels



Linear



Polynomial



Radial

Different kernels extend the range of SVM applications
At a significant increase in computational cost

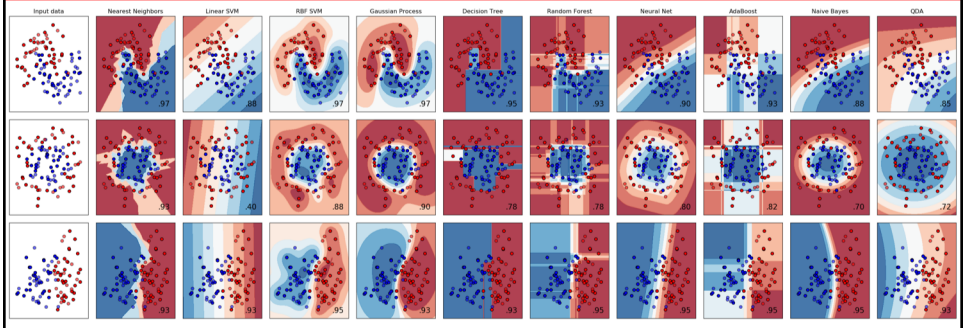
## Other Classifier Ideas and Designs:

1. **Boosting**
   Use a number of "small" weak independent classifiers and combine results (Weak → low performance)
   **Adaboost:** train a set of classifiers in sequence: for later classifiers weigh the error for examples that are incorrect in previous classifiers more heavily

2. **Random Forrest of Trees** (useful for many classes)
   Similar concept to boosting. Each tree is trained on a random training subsets

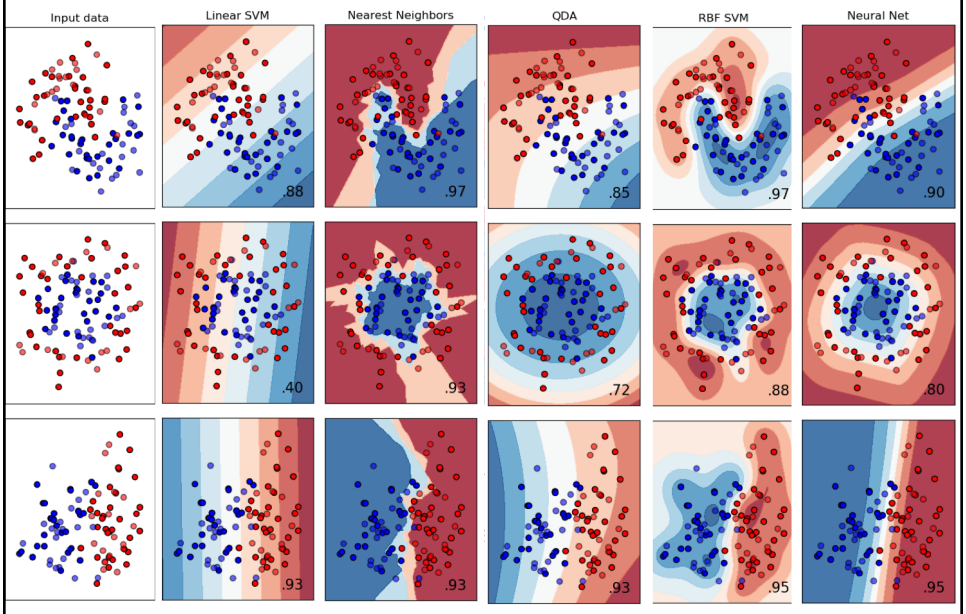
## Classifier comparison

A comparison of a several classifiers in scikit-learn on synthetic datasets.




The point of this example is to illustrate the nature of decision boundaries of different classifiers. This should be taken with a grain of salt, as the intuition conveyed by these examples does not necessarily carry over to real datasets.

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html




http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

## Statistical Pattern Recognition Summary

- Bays' Theorem gives minimum error classification
- Linear discrimination functions: simple hyperplanes
- Multivariate Gaussian assumption: hyperquadratic discriminant functions, many parameters to estimate
- Non-parametric methods, no function constraints, cost depends upon size of training set
- Logistic regression, designed for categorical outputs, involves an optimization iterative solver.
- SVM methods, recently very popular. Use optimization methods for training. (basic linear, kernels: polynomial and RBF)
- Novel methods such as Boosting and and random forest of trees are currently very popular.