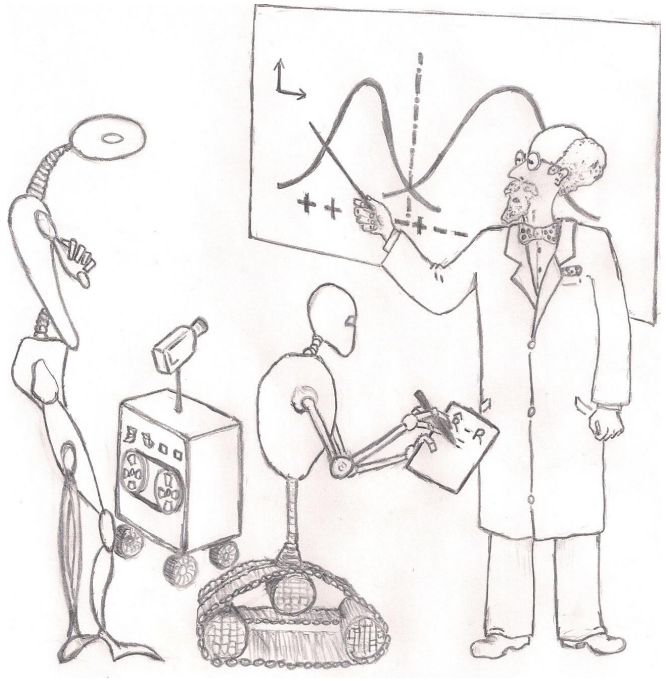
Machine Learning

A Brief Introduction

-Jyotikrishna Dass

Contents

- What do we mean by Machine Learning (ML) ?
- Applications of ML
- ML and the Jargons
- Defining a ML Problem Statement
- Learning Styles in ML
- Various ML Models



What do we mean by Machine Learning ?

- Refers to computers learning to predict from the data
- Predictions of events that are unknown to the computer i.e. something you haven't inputted or programmed into it

"Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed"

-Arthur Samuel (1959)

"A (machine learning) computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E"

- Tom Mitchell

Checker Learning Problem

A computer program that learns to play checkers might improve its performance as ***measured by its ability to win*** at the class of tasks involving ***playing checkers games***, through experience ***obtained*** by ***playing games against itself***

- Task **T** : playing checkers
- Performance measure **P**: % of game won against opponents
- Training experience **E** : playing practice game against itself

A Handwritten Recognition Problem

- Task T : recognizing and classifying handwritten words within images
- Performance measure P: % of words correctly classified
- Training experience E : a database of handwritten words with given classifications

A Robot Driving Learning Problem

- Task T : driving on public four-lane highways using vision sensors

Performance measure P: average distance traveled before an error (as judged by human overseer)

- Training experience E : a sequence of images and steering commands recorded while observing a human driver

Applications of ML

Applications: autonomous driving

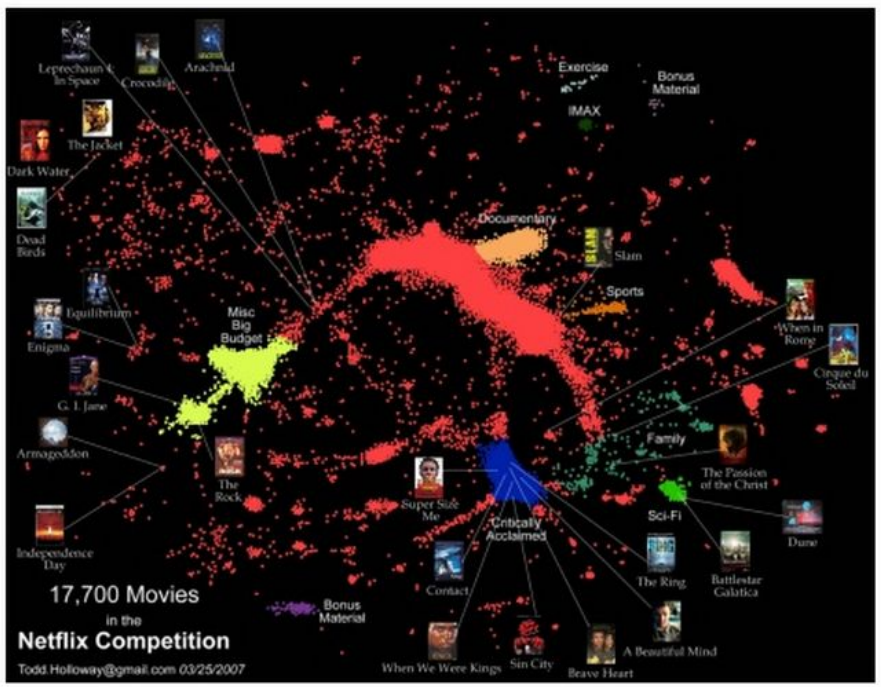
- DARPA Grand challenge 2005: build a robot capable of navigating 175 miles through desert terrain in less than 10 hours, with no human intervention
- The actual wining time of Stanley [Thrun et al., 05] was 6 hours 54 minutes.

http://www.darpa.mil/grandchallenge/

Applications: recommendation system

- Netflix prize: predict how much someone is going to love a movie based on their movies preferences
- Data: over 100 million ratings that over 480,000 users gave to nearly 18,000 movies
- Reward: $1,000,000 dollars if 10% improvement with respect to Netflix's current system

http://www.netflixprize.com



Applications: credit risk analysis

- Data:

Customer103: (time=t0)	*Customer103:* (time=t1)	...	*Customer103:* (time=tn)
Years of credit: 9	Years of credit: 9		Years of credit: 9
Loan balance: $2,400	Loan balance: $3,250		Loan balance: $4,500
Income: $52k	Income: ?		Income: ?
Own House: Yes	Own House: Yes		Own House: Yes
Other delinquent accts: 2	Other delinquent accts: 2		Other delinquent accts: 3
Max billing cycles late: 3	Max billing cycles late: 4		Max billing cycles late: 6
Profitable customer?: ?	Profitable customer?: ?		**Profitable customer?: No**
...	...		...

- Logical rules automatically learned from data:

```
If   Other-Delinquent-Accounts > 2, and
     Number-Delinquent-Billing-Cycles > 1
Then Profitable-Customer? = No
     [Deny Credit Card application]
If   Other-Delinquent-Accounts = 0, and
     (Income > $30k)  OR  (Years-of-Credit > 3)
Then Profitable-Customer? = Yes
     [Accept Credit Card application]
```

Applications: Machine Translation

Applications: Speech Recognition

Apple's Siri learns from data to predict the meanings of human voice and the desired answers or actions to be performed

ML and the Jargons

- Predictive Analytics
- Data Mining
- Data Science
- Statistical Analysis
- Artificial Intelligence
- Business Intelligence
- Big Data

Data Mining

- Computational process of **discovering patterns** in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems
- Overall goal of the data mining process is to **extract information** from a data set and transform it into an understandable structure for further use by **using specific algorithms**
- One of the many steps in KDD

Knowledge Discovery in Databases (KDD)

- Refers to the **overall process** of discovering useful knowledge from data
- Involves data preparation, data selection, data cleaning, incorporation of appropriate prior knowledge, and proper interpretation of the results of mining

Predictive Analytics

- Technology that learns from experience [i.e. data] to predict the future behavior of individuals in order to drive better decisions
- Core of predictive analytics relies on capturing relationships between explanatory variables (features) and the predicted variables from past occurrences, and exploiting them to predict the unknown outcome

Statistical Analysis

- Teaching humans what has happened or what is happening by looking at data, in order to make better decisions
- "In statistics, going from particular observations to general descriptions is called **inference** and learning is called **estimation**."

Artificial Intelligence

- Constructing a computer system, called intelligent agent, to behave and perform tasks like a human
- Intelligent System is a system that perceives its environment and takes actions that maximize its chances of success
- Machine Learning is a specific class of problems in Artificial Intelligence

Business Intelligence

- Referring to commercial organizations using data to learn about the business, market or customers and to make factually-supported decisions
- May also involve making prediction for the future or the unknown for the benefit of the business
- May be described as the KDD for Business

Data Science

- Umbrella term for everything mentioned above that makes use of data, with an emphasis of the use of sophisticated algorithms or scientific methods
- International Federation of Classification Societies describes it as “research in problems of classification, data analysis, and systems for ordering knowledge”
- Data warehousing is usually about tasks that produce one-off reporting in an offline mode, while Data Analytics can be in a real-time online environment

Big Data

- “Big data is high-volume, -velocity and -variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight and decision making.”
- It stretches the limits of scalability, computational power, processing speed and flexibility of existing machine learning technology

Defining a ML Problem Statement

A Tree Recognition Example (1/2)

- Suppose that you have never seen trees before, and I give you some “EXAMPLES”

Trees examples

‘Not’ Trees examples

A Tree Recognition Example (2/2)

- I will ask you if these unseen photos are trees or not.

Query Images

Is it a tree?

YES

or

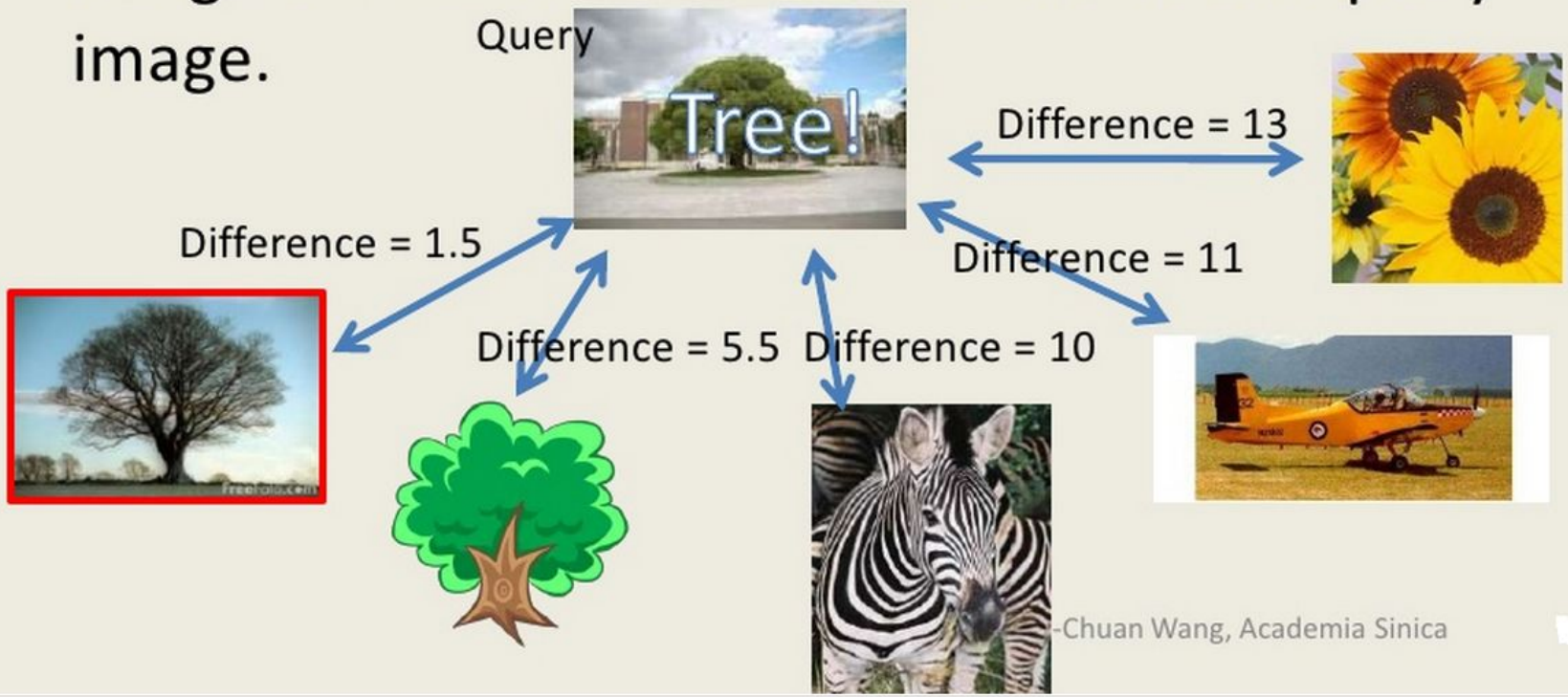
NO

(AND) How much confidence?

Shao-Chuan Wang, Academia Sinica

A Simple Algorithm: Nearest Neighbor

- For a given query image
  - Find the nearest image to the query image in the database
  - Assign the label of the nearest one to the query image.



What Were We Modeling?

Query

TREE

Human Concept
(exist but unknown)

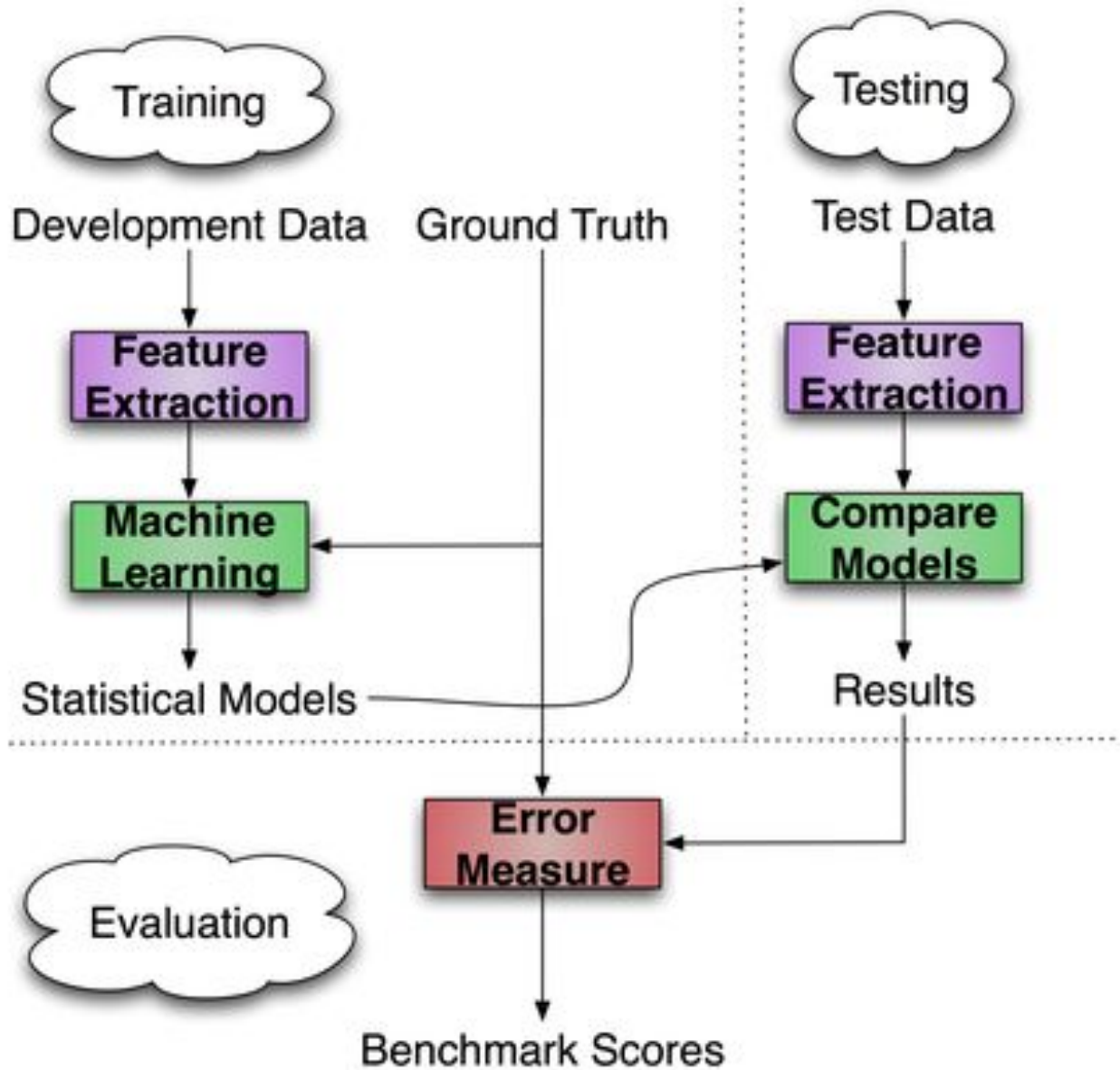
YES
YES
YES
NO
NO
NO
...

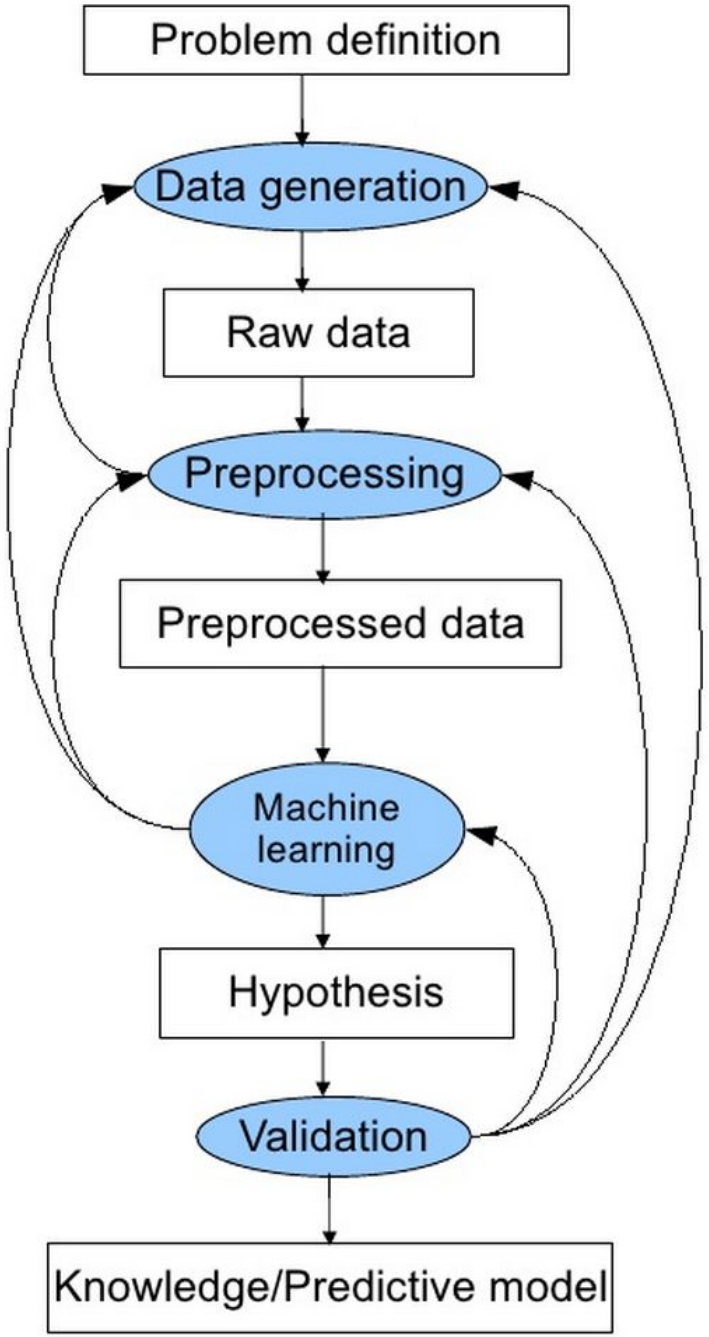
Prediction: NO

Infer

A Machine
(A learning algorithm)

Training...


Training
Testing
Development Data
Ground Truth
Test Data
Feature Extraction
Feature Extraction
Machine Learning
Compare Models
Statistical Models
Results
Error Measure
Evaluation
Benchmark Scores



One part of the data mining process

- Each step generates many questions:
  - Data generation: data types, sample size, online/offline...
  - Preprocessing: normalization, missing values, feature selection/extraction...
  - Machine learning: hypothesis, choice of learning paradigm/algorithm...
  - Hypothesis validation: cross-validation, model deployment...

Glossary

- Data=a table (dataset, database, sample)

Variables (attributes, features) =
measurements made on objects

	VAR 1	VAR 2	VAR 3	VAR 4	VAR 5	VAR 6	VAR 7	VAR 8	VAR 9	VAR 10	VAR 11	...
Object 1	0	1	2	0	1	1	2	1	0	2	0	...
Object 2	2	1	2	0	1	1	0	2	1	0	2	...
Object 3	0	0	1	0	1	1	2	0	2	1	2	...
Object 4	1	1	2	2	0	0	0	1	2	1	1	...
Object 5	0	1	0	2	1	0	2	1	1	0	1	...
Object 6	0	1	2	1	1	1	1	1	1	1	1	...
Object 7	2	1	0	1	1	2	2	2	1	1	1	...
Object 8	2	2	1	0	0	0	1	1	1	1	2	...
Object 9	1	1	0	1	0	0	0	0	1	2	1	...
Object 10	1	2	2	0	1	0	1	2	1	0	1	...
...	...	...	...	...	...	...	...	...	...	...	...	...

Objects (samples, observations,
individuals, examples, patterns)

Dimension=number of variables
Size=number of objects

- Objects: samples, patients, documents, images...
- Variables: genes, proteins, words, pixels...

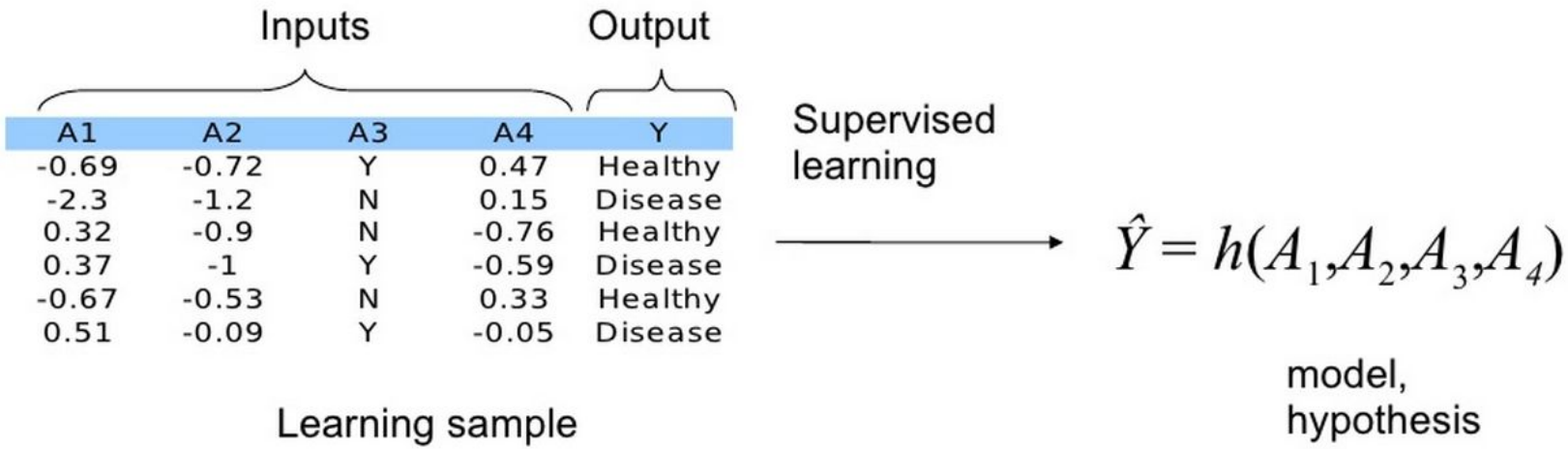
Learning Styles in ML

Learning Styles

Way of organizing machine learning algorithms which is useful because it forces you to think about the roles of the input data and the model preparation process and select one that is the most appropriate for your problem in order to get the best result

1. Supervised Learning
2. Unsupervised Learning
3. Semi-supervised Learning
4. Reinforcement Learning

Supervised learning



- Goal: from the database (learning sample), find a function *h* of the inputs that approximates ***at best*** the output
- Symbolic output ⇒ *classification* problem,
- Numerical output ⇒ *regression* problem

Two main goals

- Predictive:

  Make predictions for a ***new*** sample described by its attributes

A1	A2	A3	A4	Y
0.83	-0.54	T	0.68	Healthy
-2.3	-1.2	F	-0.83	Disease
0.08	0.63	F	0.76	Healthy
0.06	-0.29	T	-0.57	Disease
-0.98	-0.18	F	-0.38	Healthy
-0.68	0.82	T	-0.95	Disease
0.92	-0.33	F	-0.48	?

- Informative:

  Help to understand the relationship between the inputs and the output

  Y=disease if $A3$=F and $A2$<0.3

  Find the most relevant inputs

Unsupervised Learning

- So far, in all the learning techniques we considered, a training example consisted of a set of attributes (or features) and either a class (in the case of classification) or a real number (in the case of regression) attached to it

- Unsupervised Learning takes as training examples the set of attributes/features alone

- The purpose of unsupervised learning is to attempt to find natural partitions in the training set

- Two general strategies for Unsupervised learning include: ***Clustering*** and ***Hierarchical Clustering***

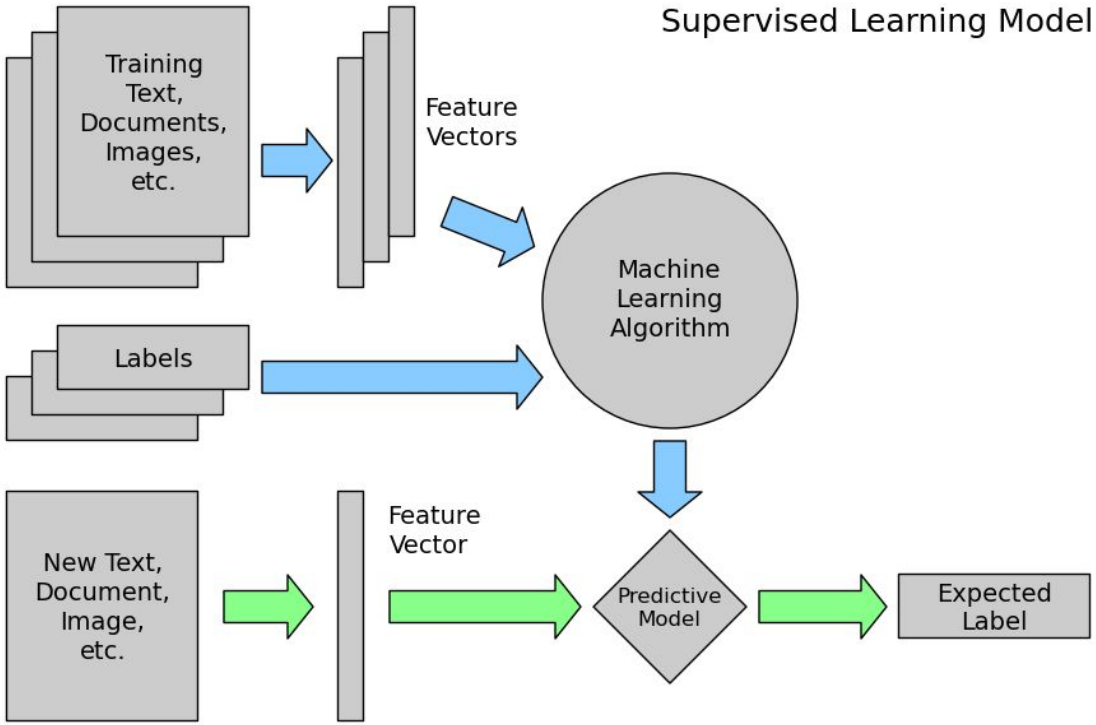
Why Unsupervised ?

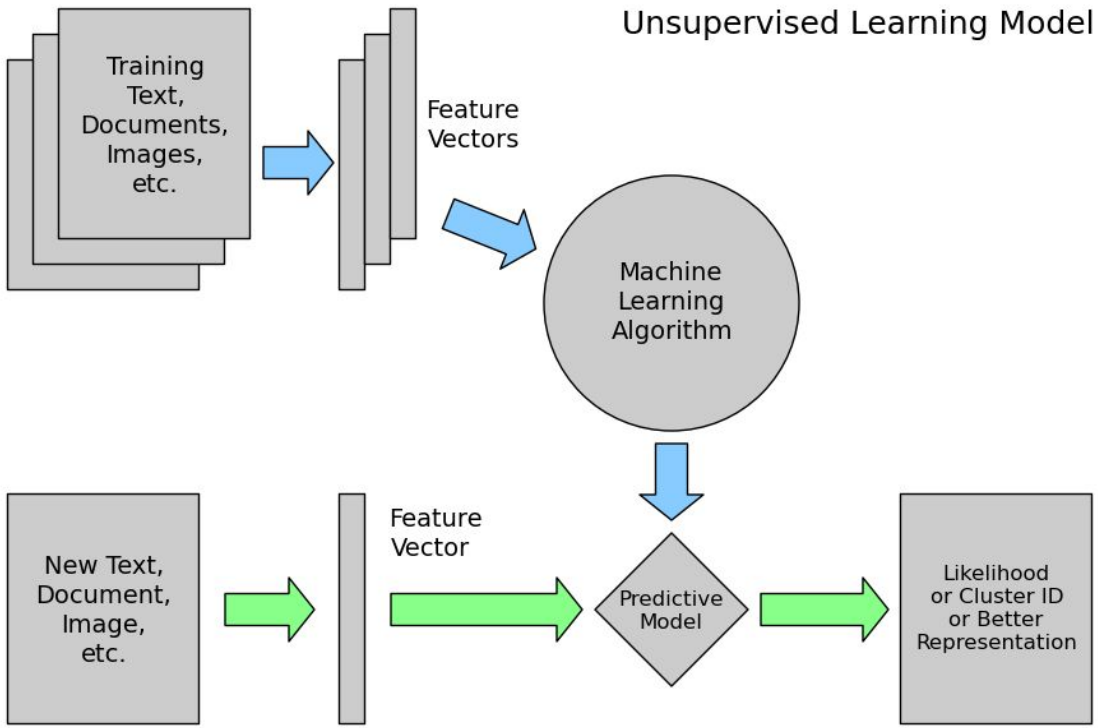
- Collecting and labeling a large set of sample patterns can be very expensive. By designing a basic classifier with a small set of labeled samples, and then tuning the classifier up by allowing it to run without supervision on a large, unlabeled set, much time and trouble can be saved

- Training with large amounts of often less expensive, unlabeled data, and then using supervision to label the groupings found. This may be used for large "data mining" applications where the contents of a large database are not known beforehand

- Unsupervised methods can also be used to find features which can be useful for categorization. There are unsupervised methods that represent a form of data-dependent "smart pre-processing" or "smart feature extraction.“

- Lastly, it can be of interest to gain insight into the nature or structure of the data. The discovery of similarities among patterns or of major departures from expected characteristics may suggest a significantly different approach to designing the classifier.


Supervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
Labels
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Expected Label

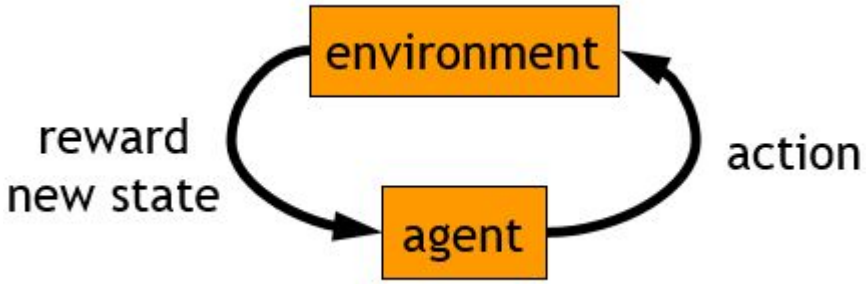

Unsupervised Learning Model
Training Text, Documents, Images, etc.
Feature Vectors
Machine Learning Algorithm
New Text, Document, Image, etc.
Feature Vector
Predictive Model
Likelihood or Cluster ID or Better Representation

Semi Supervised Learning

- Need for an intermediate approach
- Input data is a mixture of labelled and unlabeled examples
- There is a desired prediction problem but the model must learn the structures to organize the data as well as make predictions

Reinforcement Learning

- Input data is provided as stimulus to a model from an environment to which the model must respond and react
- Feedback is provided not from of a teaching process as in supervised learning, but as punishments and rewards in the environment
- Example problems are systems and robot control



Various ML Models

- We can choose many models, each based on a set of different assumptions regarding the underlying distribution of data

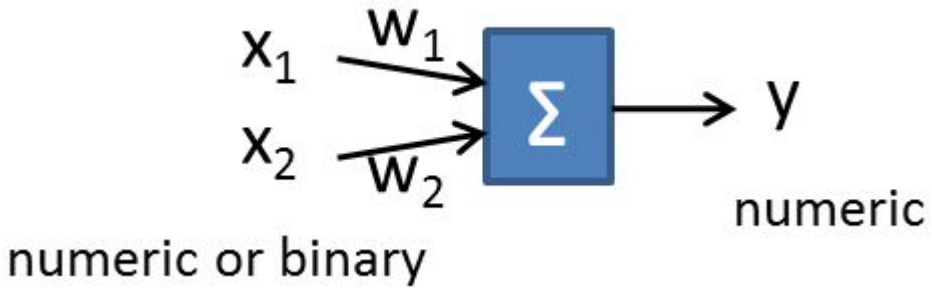
- Therefore, we are interested in two general types of problems in this discussion:

1. **Classification**—about predicting a category (a value that is discrete, finite with no ordering implied), and

2. **Regression**—about predicting a numeric quantity (a value that's continuous and infinite with ordering)

Linear Regression

- Most popular ML model
- Based on the assumption that a linear relationship exists between the input and output variables

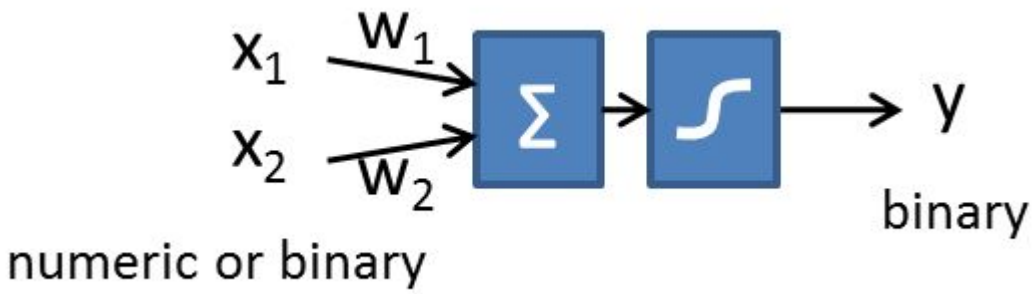


- The learning algorithm will learn the set of parameters such that the sum of square error $(y_{actual} - y_{estimate})^2$ is minimized
- The goal of minimizing the square error makes linear regression very sensitive to outliers that greatly deviate in the output. It is a common practice to identify those outliers, remove them, and then rerun the training

Logistic Regression

- In a classification problem, the output is binary rather than numeric

- We can imagine doing a linear regression and then compressing the numeric output into a 0..1
  range using the logit function $1/(1+e^{-t})$, shown here:



$$y = 1/(1 + e^{-(w0 + w1x1 + w2x2 + \ldots)})$$

…where y is the 0 .. 1 value, and xi is the input numeric value

- The learning algorithm will learn the set of parameters such that the cost

$(y_{actual} * \log y_{estimate} + (1 - y_{actual}) * \log(1 - y_{estimate}))$ is minimized.

Regression with Regularization

- To avoid an over-fitting problem (the trained model fits too well with the training data and is not generalized enough), the regularization technique is used to shrink the magnitude of w_i.
- This is done by adding a penalty (a function of the sum of w_i) into the cost function.
- In L2 regularization (also known as Ridge regression), w_i^2 will be added to the cost function
- In L1 regularization (also known as Lasso regression), $||w_i||$ will be added to the cost function.
- Both L1, L2 will shrink the magnitude of w_i
- For variables that are inter-dependent, L2 tends to spread the shrinkage such that all interdependent variables are equally influential.
- On the other hand, L1 tends to keep one variable and shrink all the other dependent variables to values very close to zero. In other words, L1 shrinks the variables in an uneven manner so that it can also be used to select input variables.

- Combining L1 and L2, the general form of the cost function becomes the following:

$$\text{Cost} = \text{Non-regularization-cost} + \lambda\,(\alpha.\Sigma\,||w_i|| + (1-\alpha).\Sigma\,w_i^2)$$

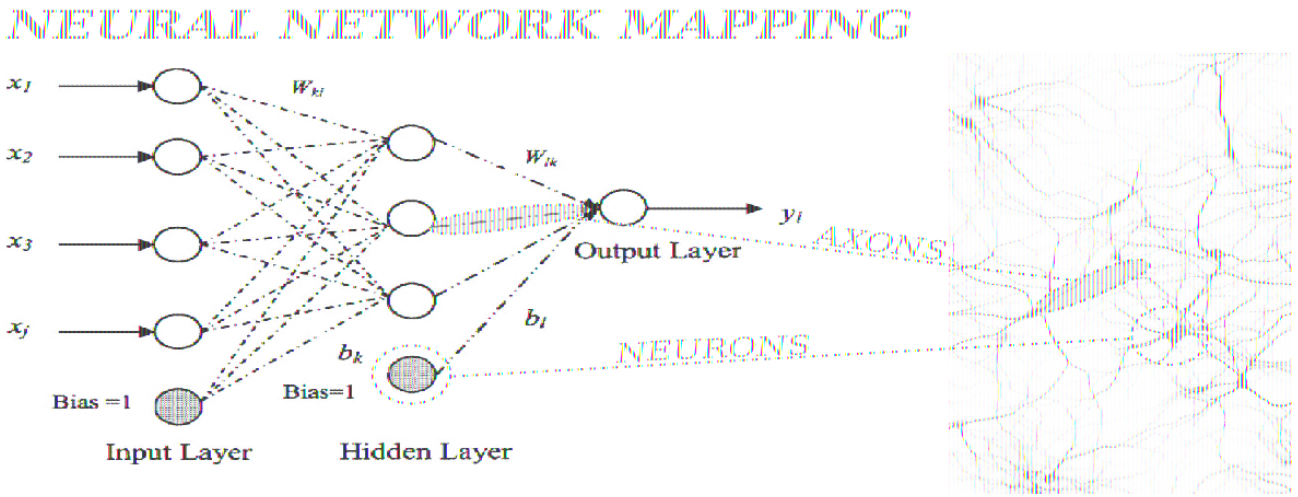
Where,

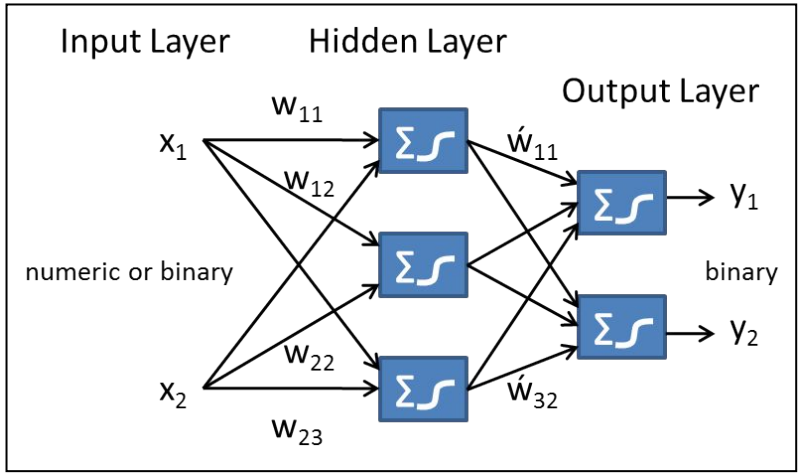
- Lambda (λ) controls the degree of regularization (0 means no regularization and infinity means ignoring all input variables because all coefficients of them will be zero).
- Alpha (α) controls the degree of mix between L1 and L2 (0 means pure L2 and 1 means pure L1). The alpha parameter needs to be supplied based on the application's need, i.e., its need for selecting a reduced set of variables

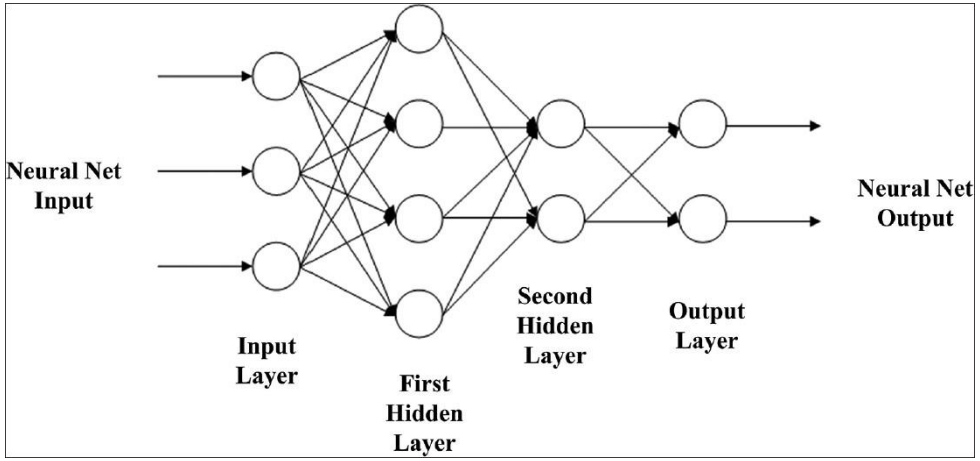
Neural Network

- Emulates the structure of a human brain as a network of neurons that are interconnected to each other

- Each neuron is technically equivalent to a logistic regression unit







- Neurons are organized in multiple layers where every neuron at layer **i** connects to every neuron at layer **i+1** and nothing else

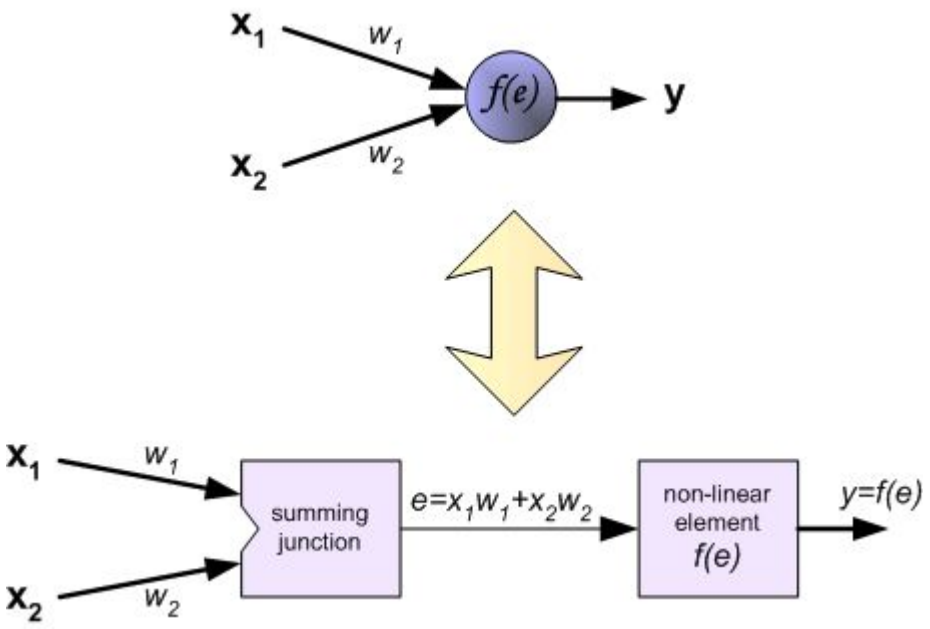
- The tuning parameters in a neural network include the number of hidden layers (commonly set to 1), the number of neurons in each layer (which should be same for all hidden layers and usually at 1 to 3 times the input variables), and the learning rate

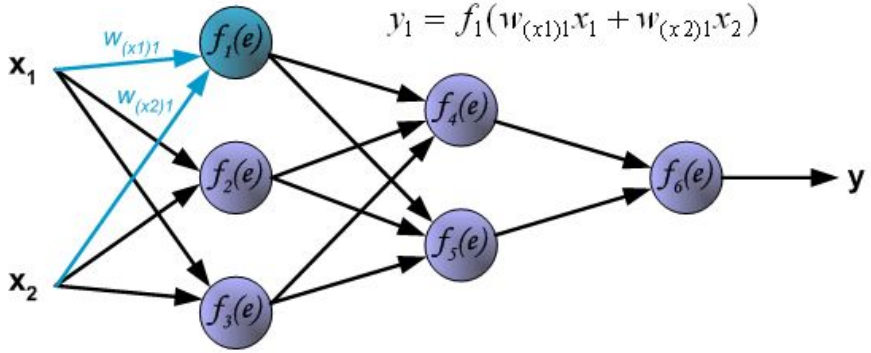
- On the other hand, the number of neurons at the output layer depends on how many binary outputs need to be learned
- In a classification problem, this is typically the number of possible values at the output category

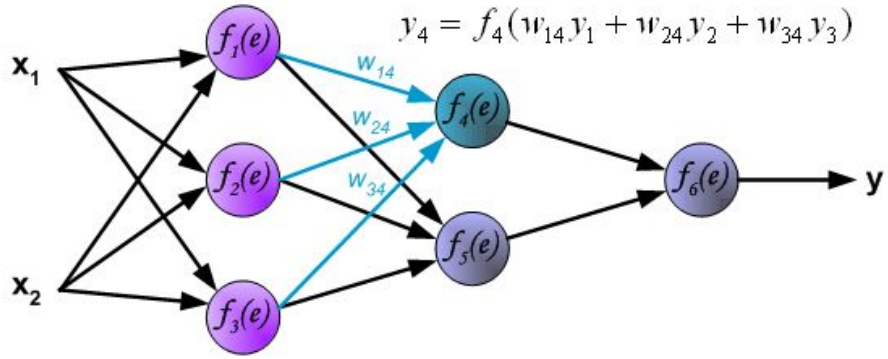
- The learning happens via an iterative feedback mechanism where the error of training data output is used to adjust the corresponding weights of input

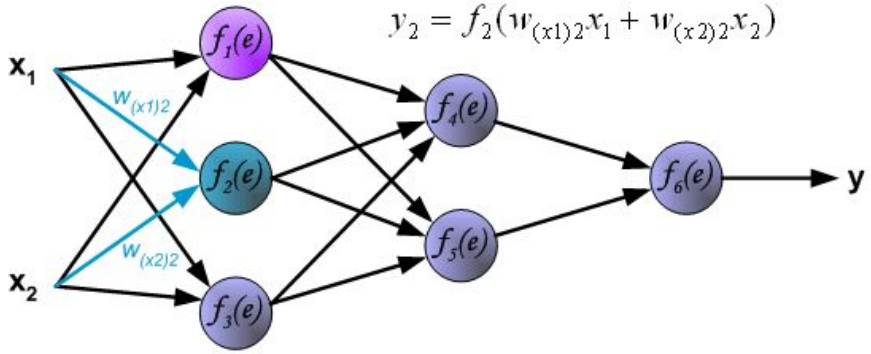
- This adjustment propagates to previous layers and the learning algorithm is known as "backpropagation."

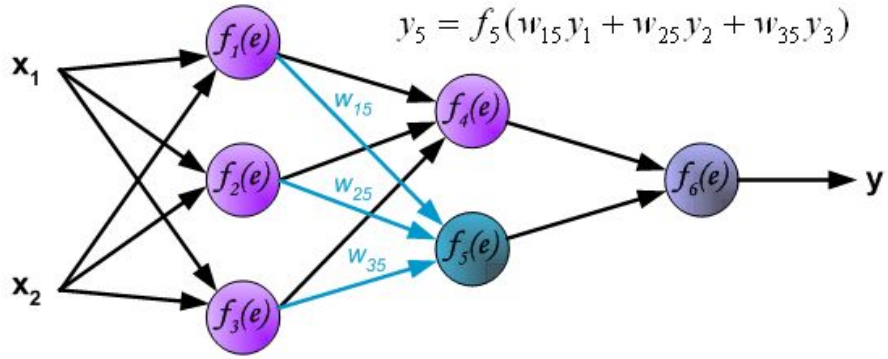
Illustration

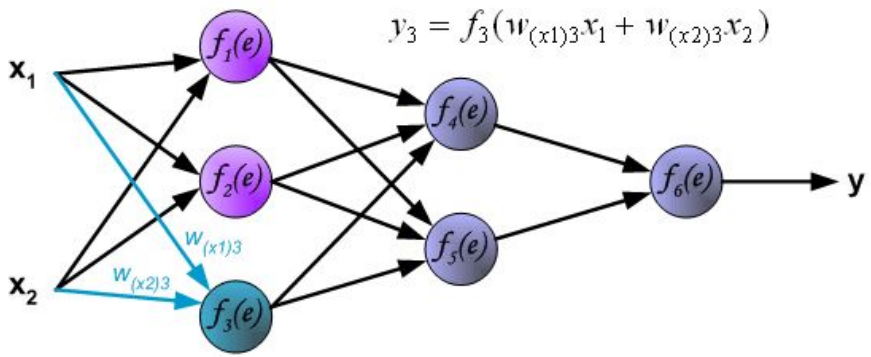


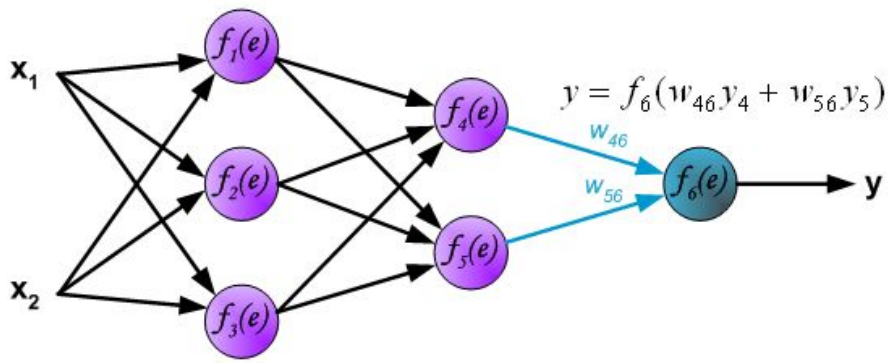

$y_1 = f_1(w_{(x1)1}x_1 + w_{(x2)1}x_2)$
x1
x2
w(x1)1
w(x2)1
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y


$y_4 = f_4(w_{14}y_1 + w_{24}y_2 + w_{34}y_3)$
x1
x2
w14
w24
w34
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y


$y_2 = f_2(w_{(x1)2}x_1 + w_{(x2)2}x_2)$
x1
x2
w(x1)2
w(x2)2
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y

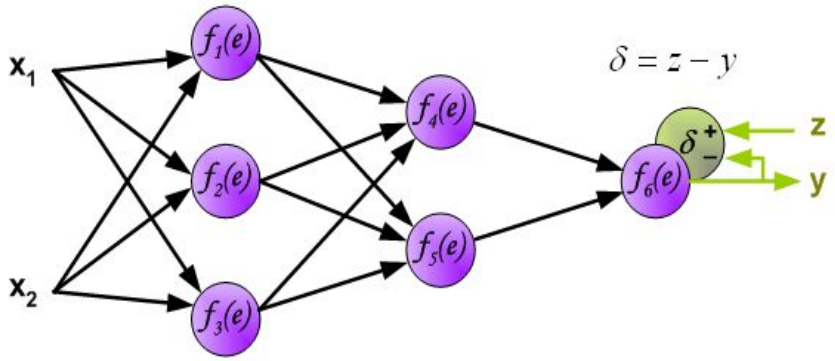

$y_5 = f_5(w_{15}y_1 + w_{25}y_2 + w_{35}y_3)$
x1
x2
w15
w25
w35
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y


$y_3 = f_3(w_{(x1)3}x_1 + w_{(x2)3}x_2)$
x1
x2
w(x1)3
w(x2)3
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y

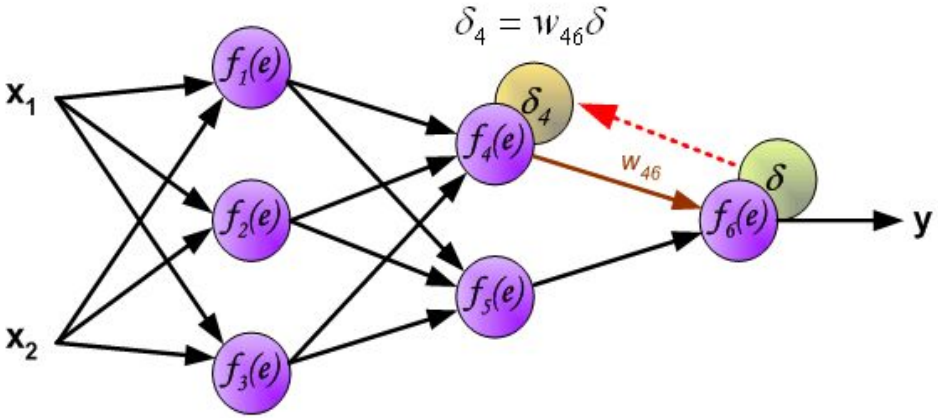

$y = f_6(w_{46}y_4 + w_{56}y_5)$
x1
x2
w46
w56
f1(e)
f2(e)
f3(e)
f4(e)
f5(e)
f6(e)
y

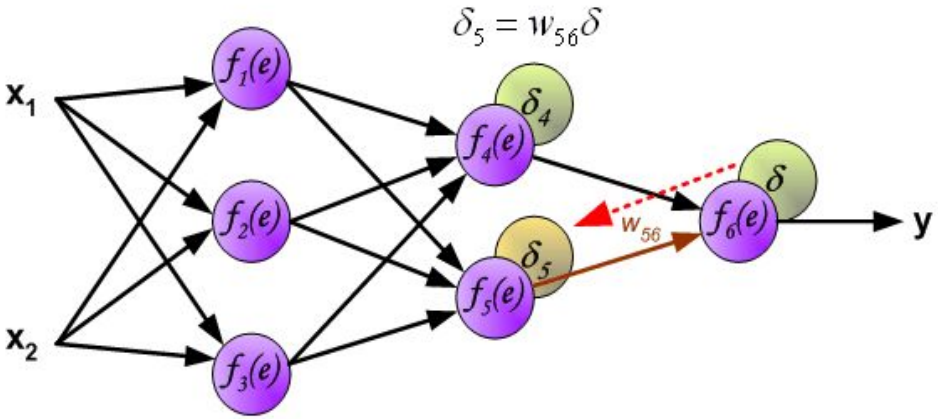
Computing Error Signal

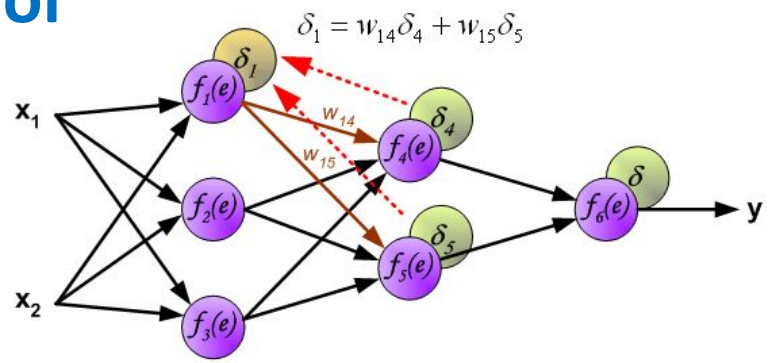
In the next algorithm step the output signal of the network y is compared with the desired output value (the target), which is found in training data set. The difference is called error signal δ of output layer neuron.

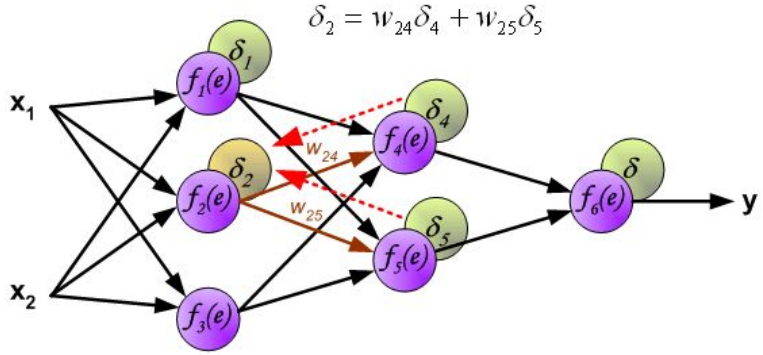


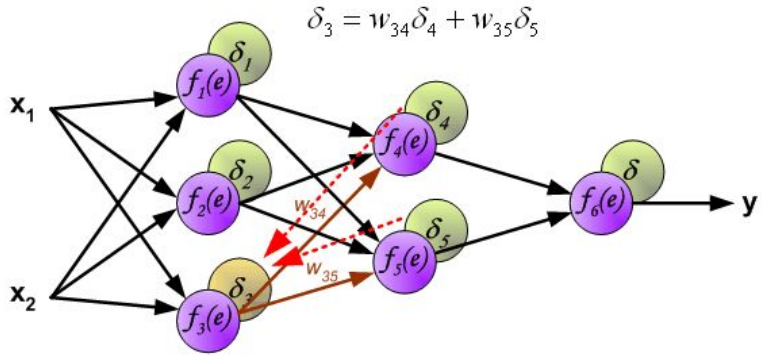
Backpropagation of Error







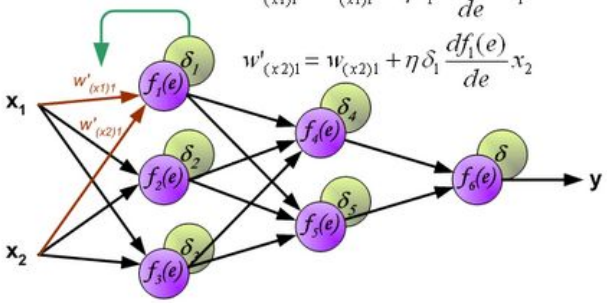




Weights Update

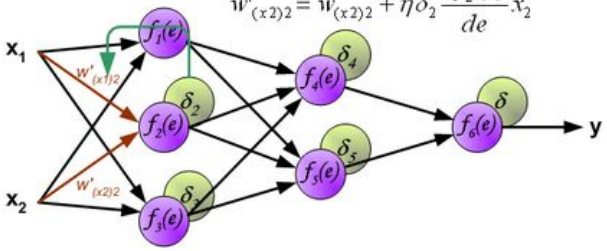
$$w'_{(x1)1} = w_{(x1)1} + \eta\delta_1 \frac{df_1(e)}{de} x_1$$

$$w'_{(x2)1} = w_{(x2)1} + \eta\delta_1 \frac{df_1(e)}{de} x_2$$



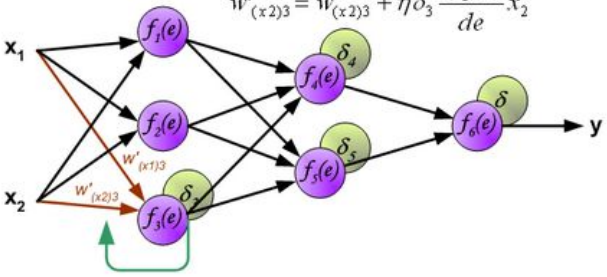
$$w'_{(x1)2} = w_{(x1)2} + \eta\delta_2 \frac{df_2(e)}{de} x_1$$

$$w'_{(x2)2} = w_{(x2)2} + \eta\delta_2 \frac{df_2(e)}{de} x_2$$



$$w'_{(x1)3} = w_{(x1)3} + \eta\delta_3 \frac{df_3(e)}{de} x_1$$

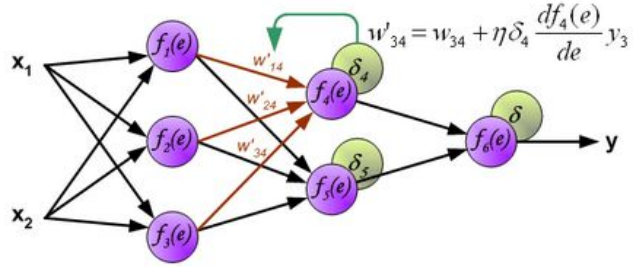
$$w'_{(x2)3} = w_{(x2)3} + \eta\delta_3 \frac{df_3(e)}{de} x_2$$



$$w'_{14} = w_{14} + \eta\delta_4 \frac{df_4(e)}{de} y_1$$

$$w'_{24} = w_{24} + \eta\delta_4 \frac{df_4(e)}{de} y_2$$

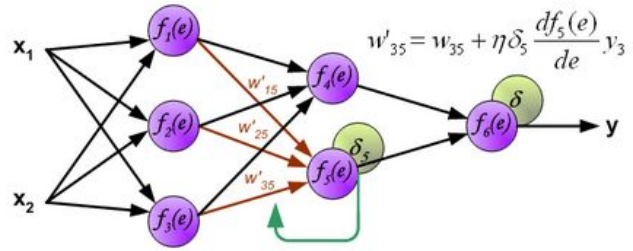
$$w'_{34} = w_{34} + \eta\delta_4 \frac{df_4(e)}{de} y_3$$



$$w'_{15} = w_{15} + \eta\delta_5 \frac{df_5(e)}{de} y_1$$

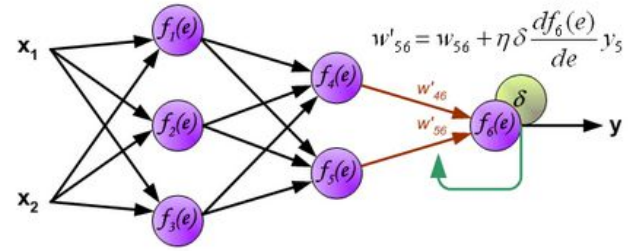
$$w'_{25} = w_{25} + \eta\delta_5 \frac{df_5(e)}{de} y_2$$

$$w'_{35} = w_{35} + \eta\delta_5 \frac{df_5(e)}{de} y_3$$



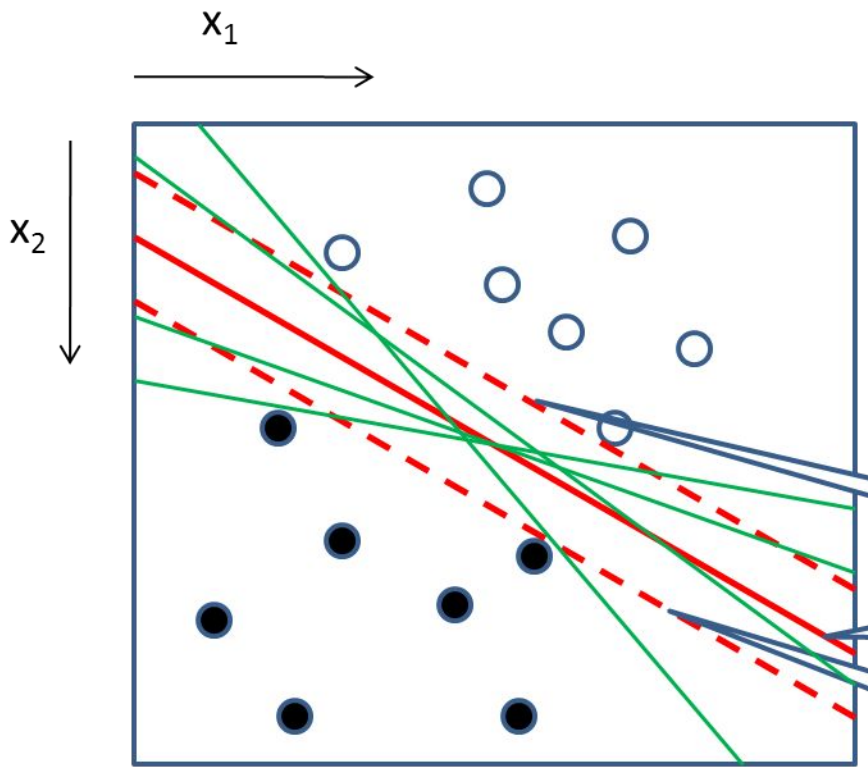
$$w'_{46} = w_{46} + \eta\delta \frac{df_6(e)}{de} y_4$$

$$w'_{56} = w_{56} + \eta\delta \frac{df_6(e)}{de} y_5$$



Support Vector Machines (SVM)

- Provides a binary classification mechanism based on finding a hyperplane between a set of samples with +ve and -ve outputs
- Assumes the data is linearly separable



All green lines can separate the points
Which one is the best ?
The red line has biggest margin

All +ve samples lies above this line
$y_i = 1$ if $\vec{w}_i . \vec{x} + b > 1$

$\vec{w} . \vec{x} + b = 1$

$\vec{w} . \vec{x} + b = 0$

$\vec{w} . \vec{x} + b = -1$

All -ve samples lies below this line
$y_i = -1$ if $\vec{w}_i . \vec{x} + b < -1$

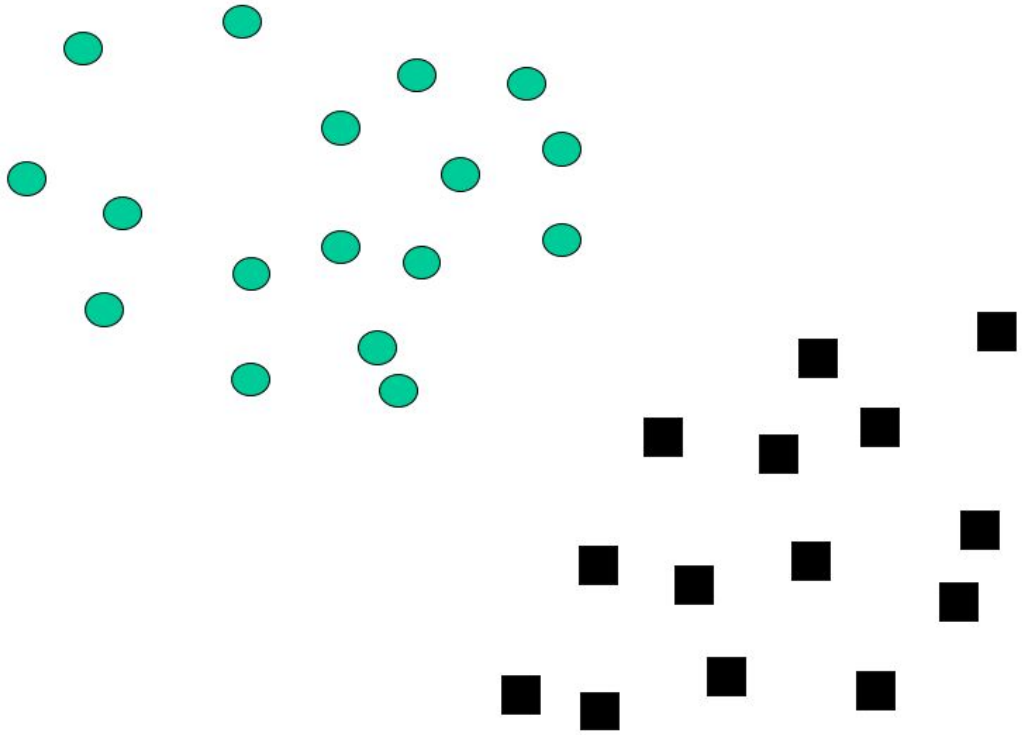
Margin = $(point_{upperline} - point_{lowerline}) . \vec{w} / |w|$
since $(point_{upperline}) . \vec{w} + b = 1$
And $(point_{lowerline}) . \vec{w} + b = -1$
So margin = $(1 - b + 1 + b) / |w| = 2 / |w|$
Maximize margin is max $2 / |w|^2$ is min $|w|^2 / 2$

Problem:

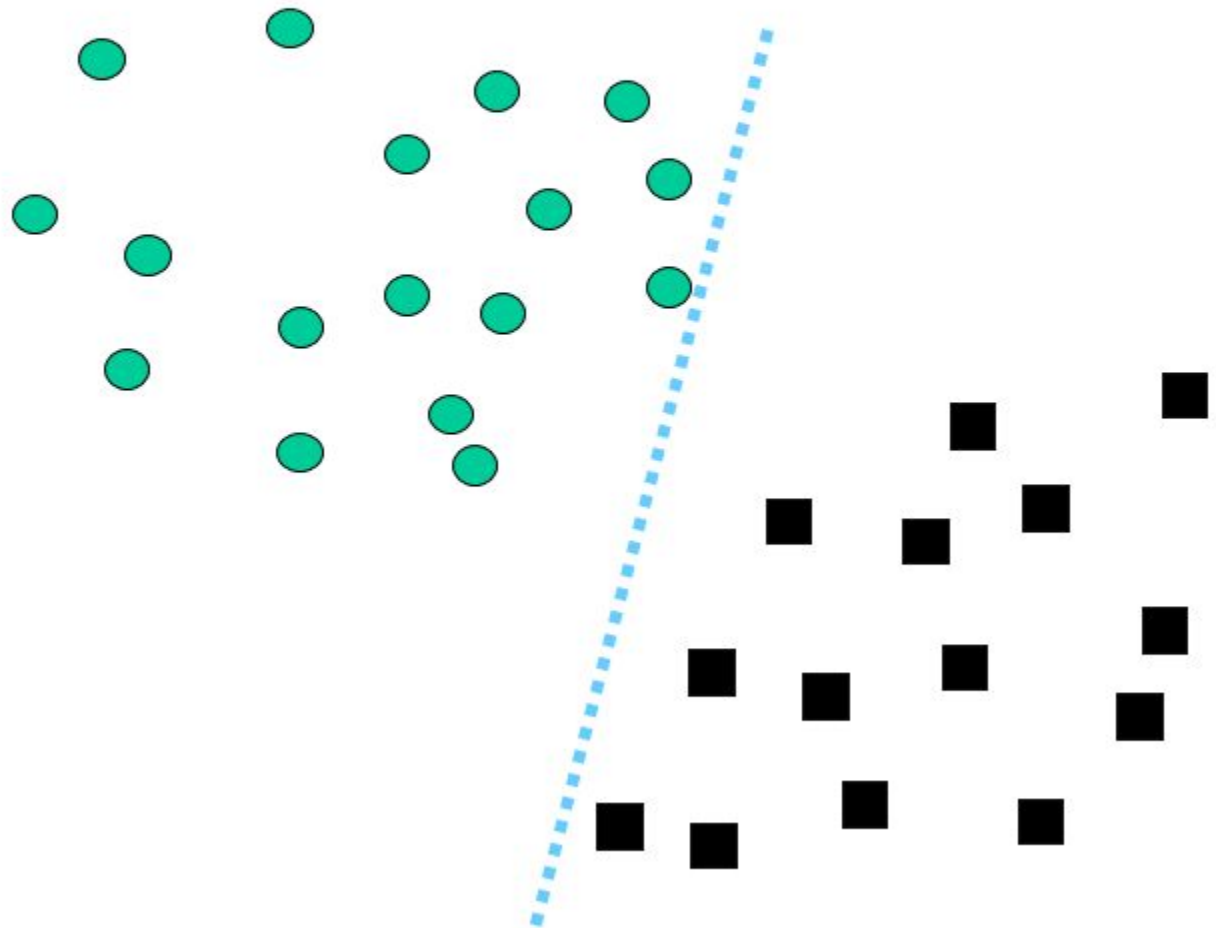
Minimize $(½) \vec{w} . \vec{w}$

With constraint: $y_i(\vec{w}_i . \vec{x} + b) > 1$

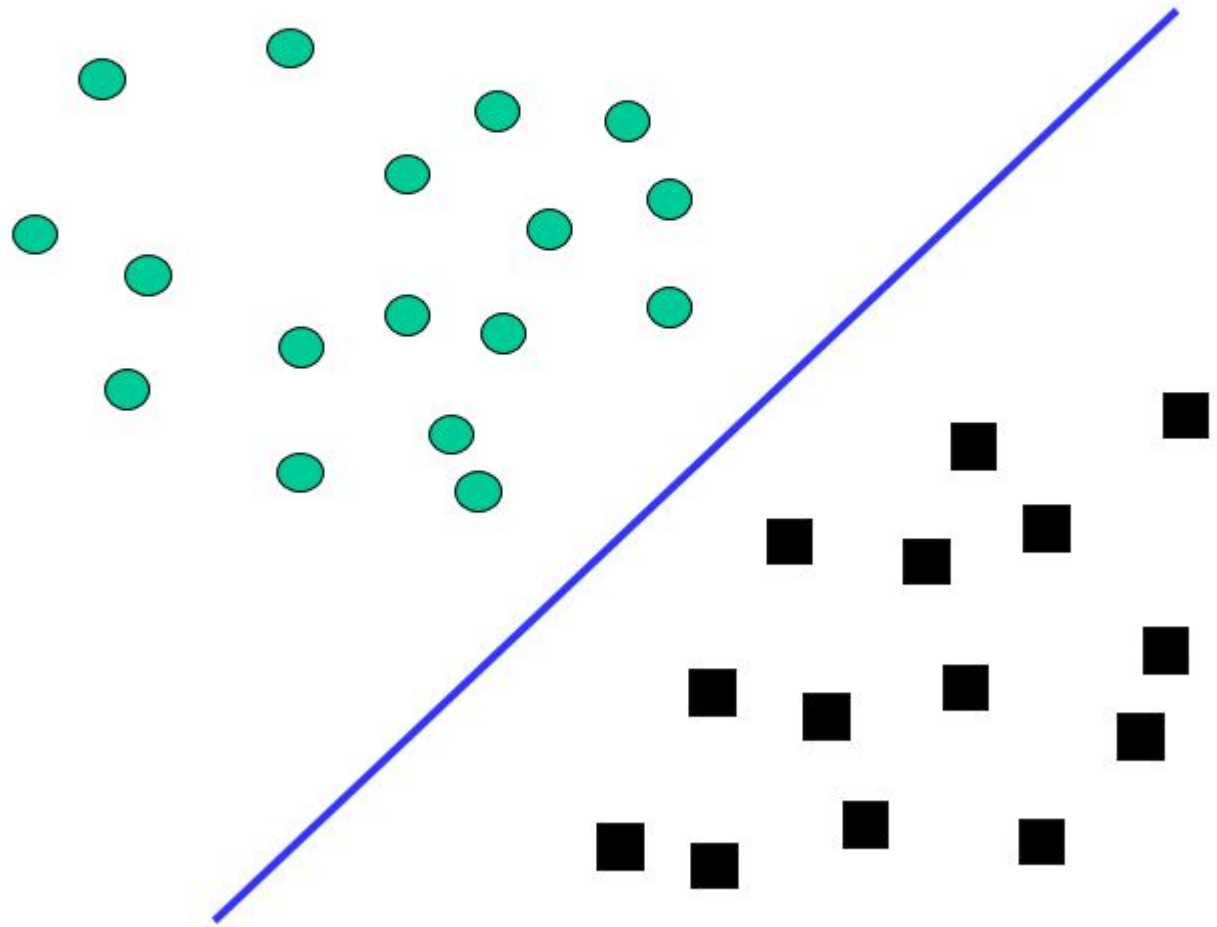
Which of the linear separators is optimal?



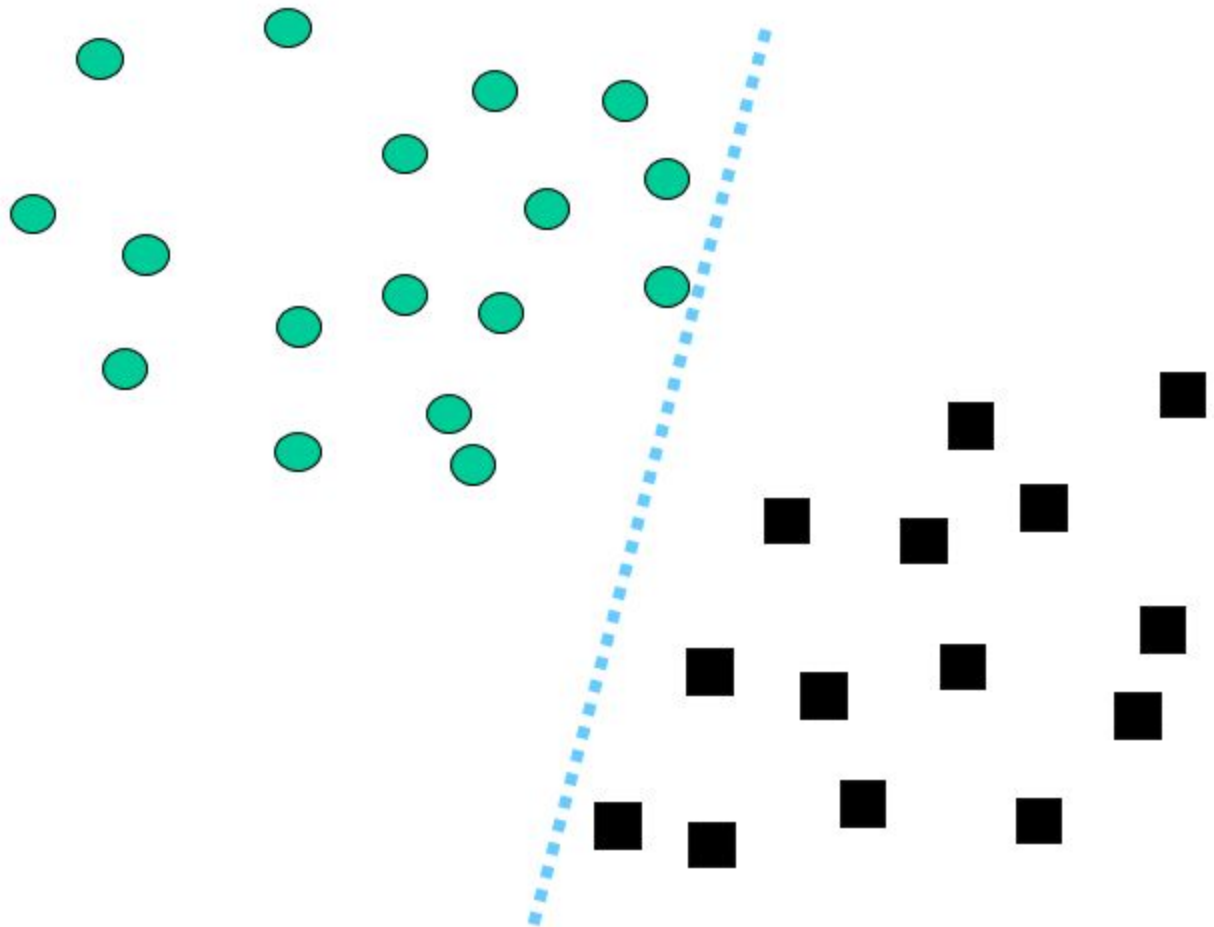
Best Linear Separator?



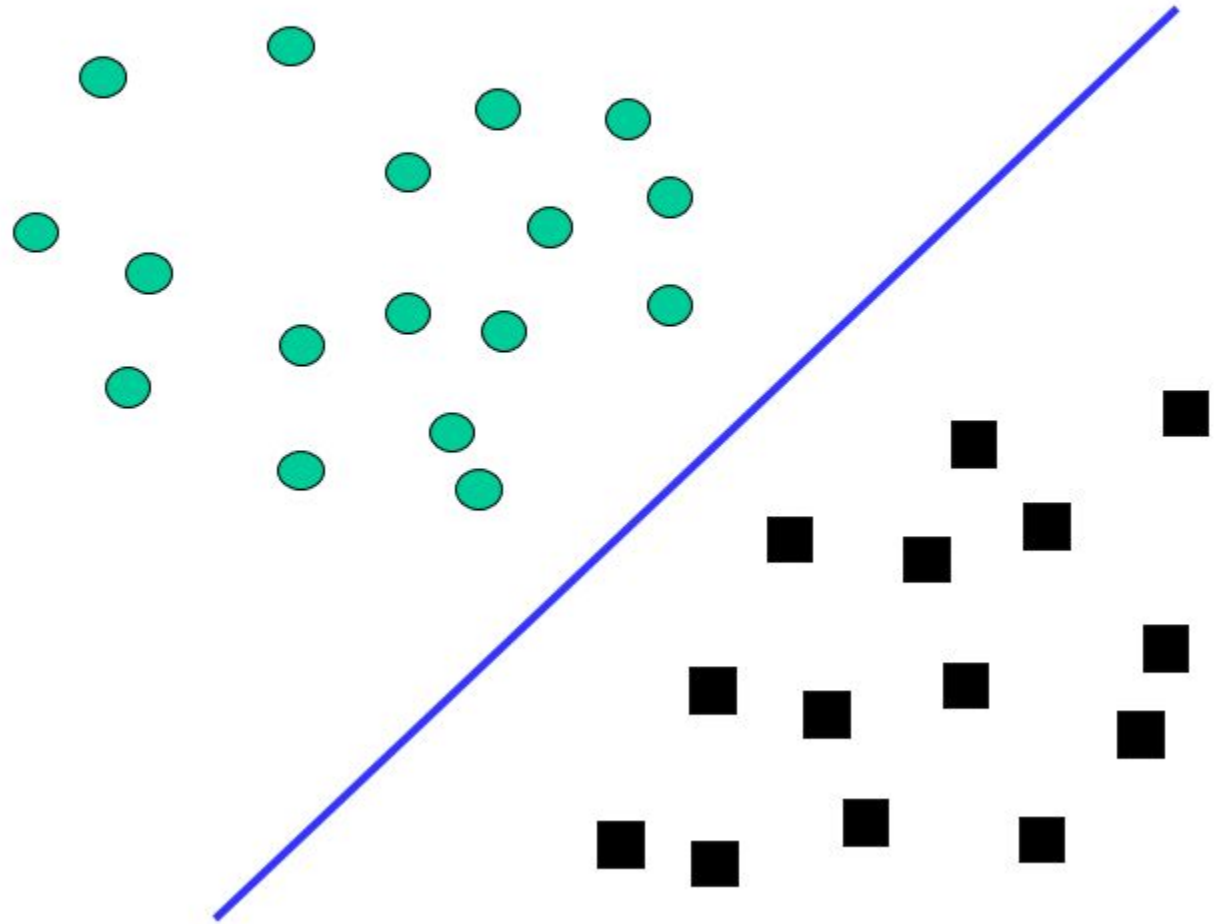
Best Linear Separator?



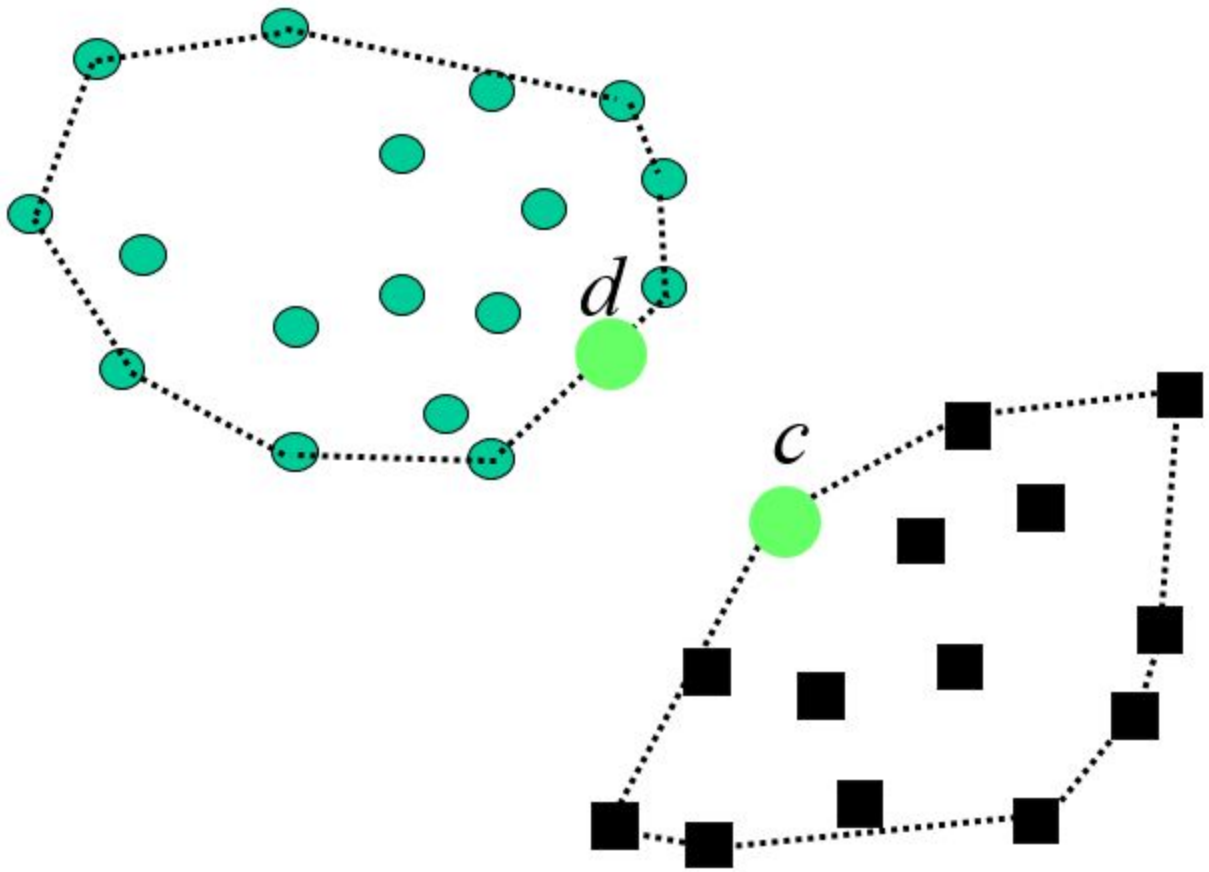
Best Linear Separator?



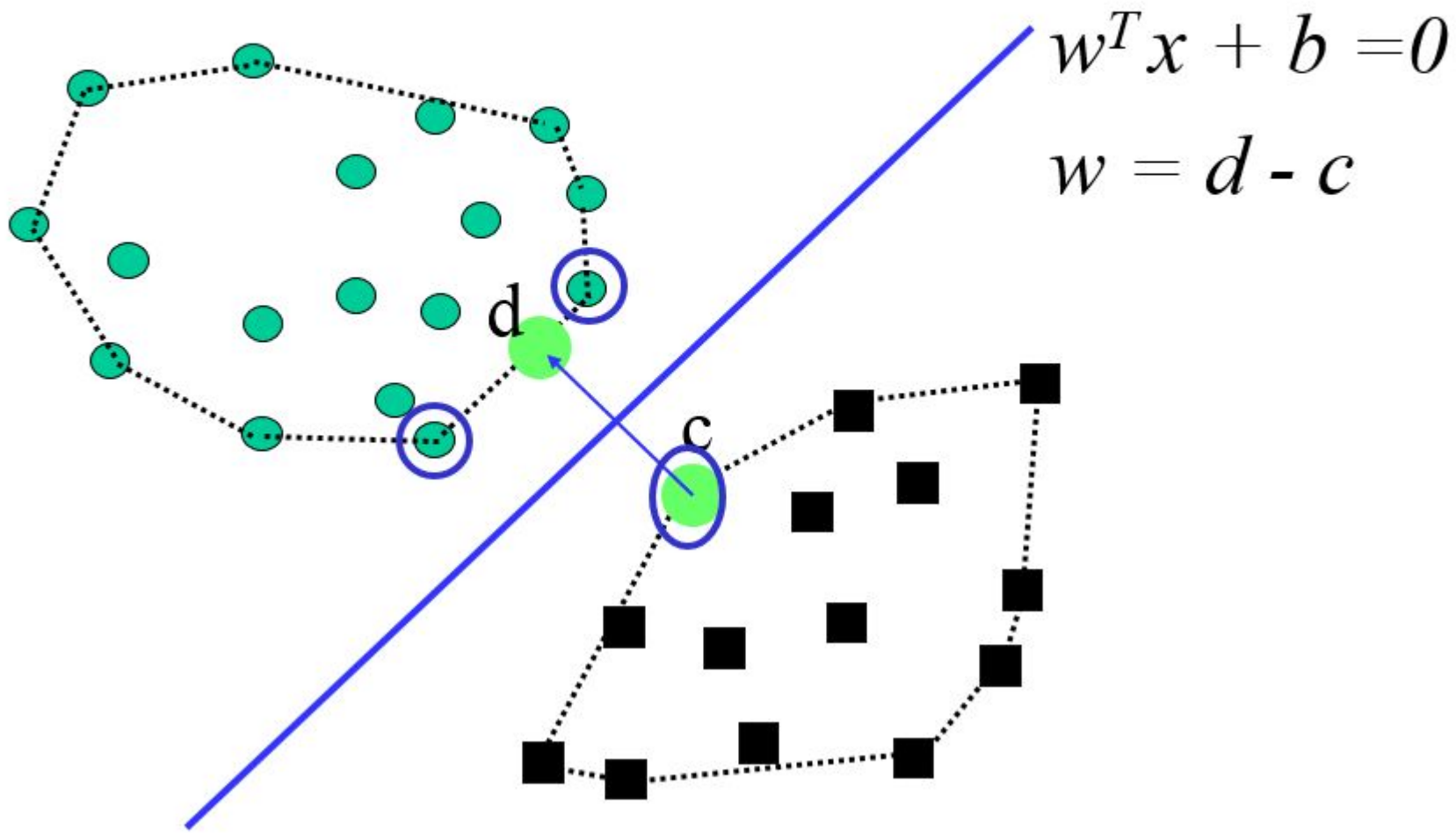
Best Linear Separator?



Find Closest Points in Convex Hulls

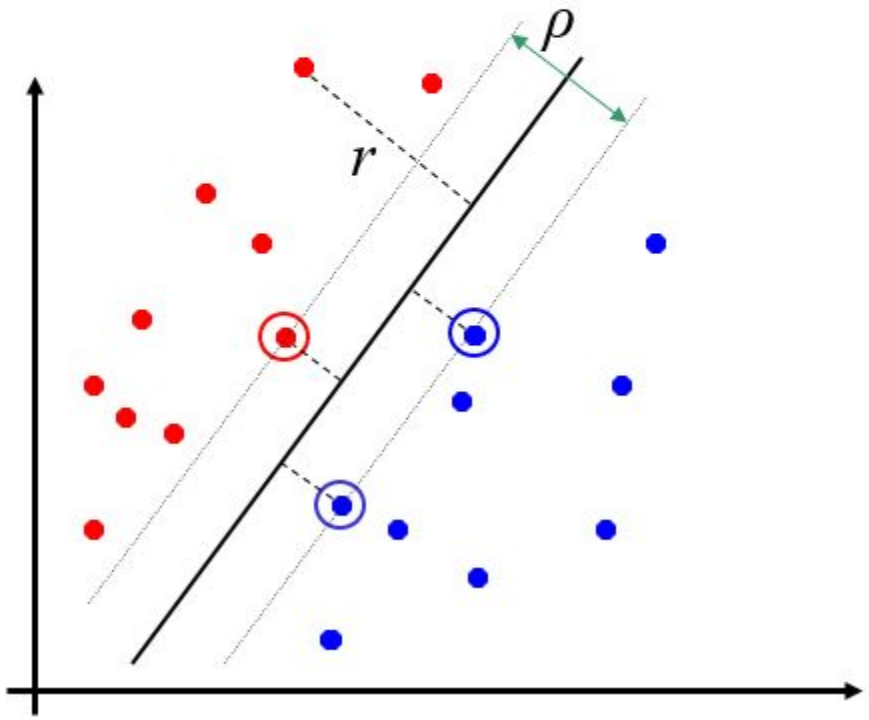


Plane Bisect Closest Points



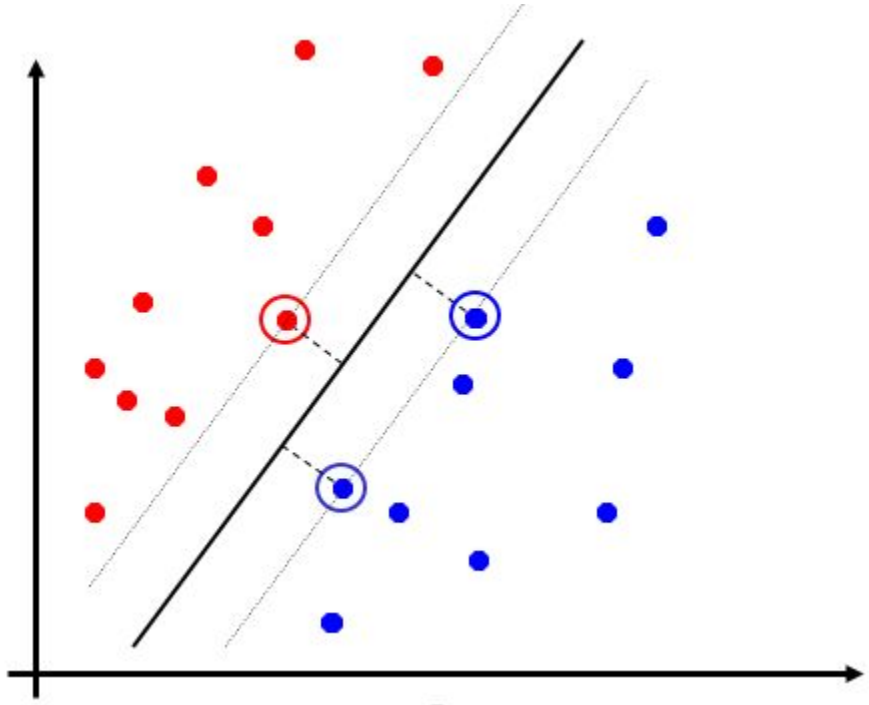
Classification Margin

- Distance from example data to the separator is $r = \dfrac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$
- Data closest to the hyperplane are ***support vectors***.
- ***Margin*** ρ of the separator is the width of separation between classes.



Maximum Margin Classification

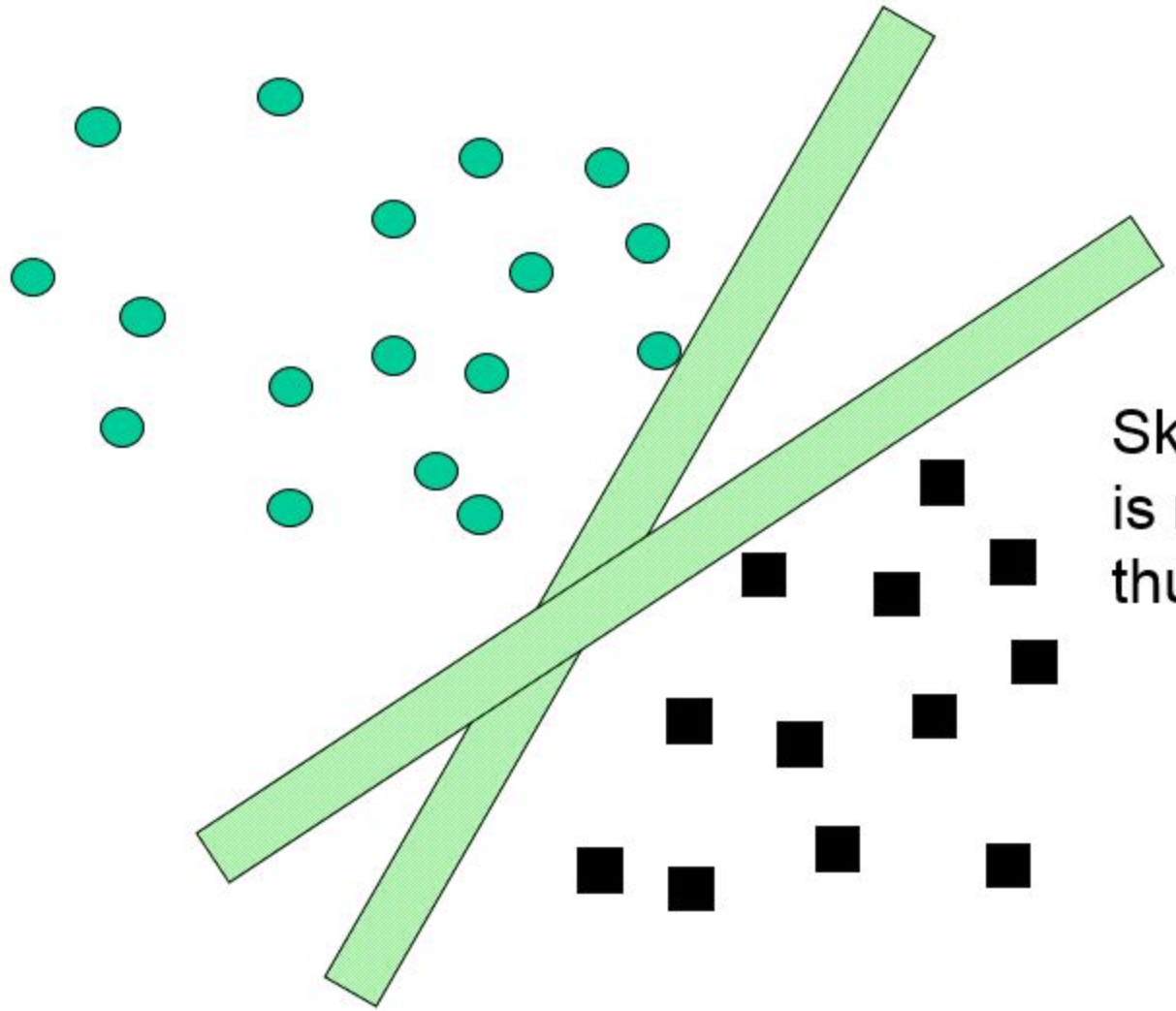
- Maximizing the margin is good according to intuition and theory.
- Implies that only support vectors are important; other training examples are ignorable.



Statistical Learning Theory

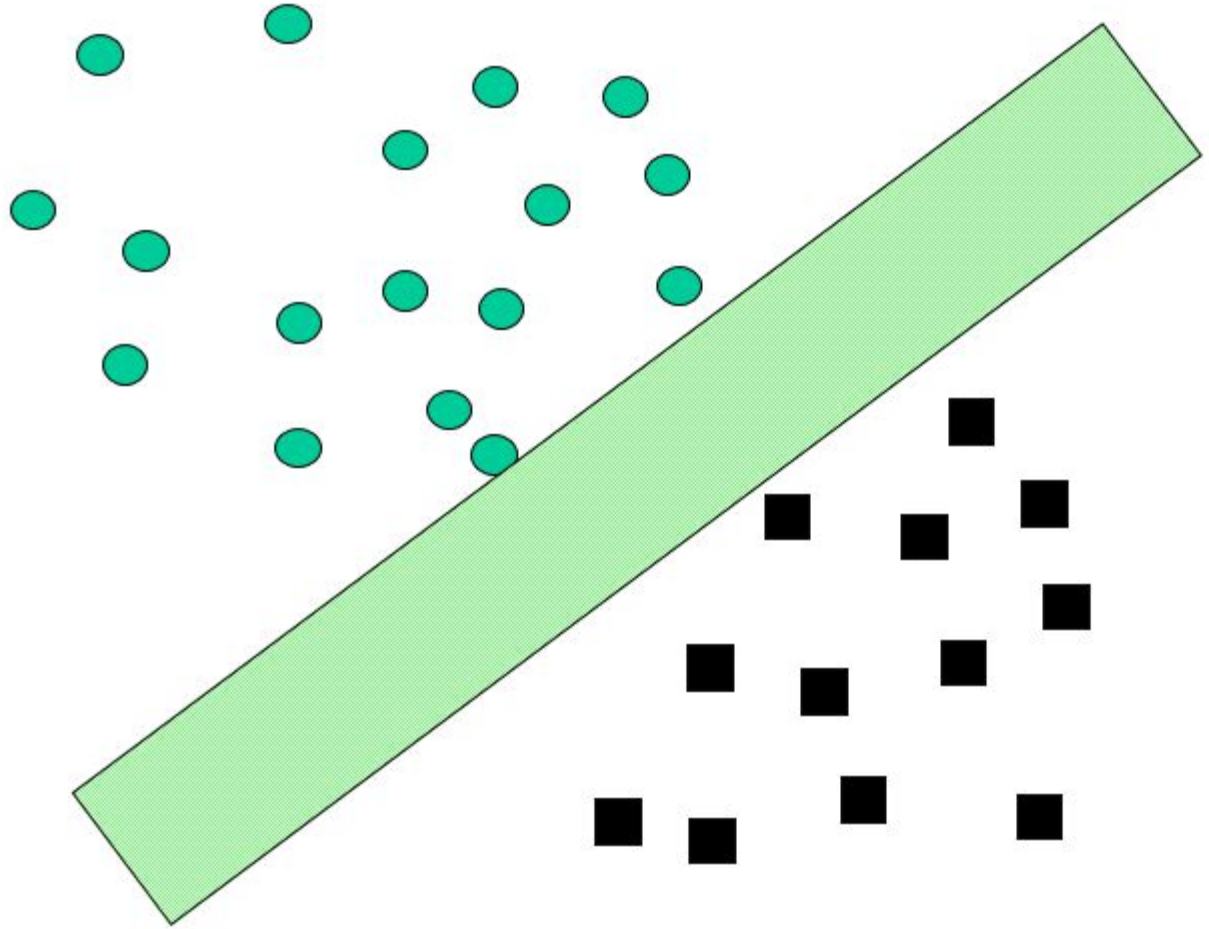
- Misclassification error and the function complexity bound generalization error.
- Maximizing margins minimizes complexity.
- “Eliminates” overfitting.
- Solution depends only on *Support Vectors* not number of attributes.

Margins and Complexity



Skinny margin
is more flexible
thus more complex.

Margins and Complexity



Fat margin
is less complex.

Linear SVM Mathematically

- Assuming all data is at distance larger than 1 from the hyperplane, the following two constraints follow for a training set $\{(\mathbf{x_i}, y_i)\}$

$$\mathbf{w^T x_i} + b \geq 1 \quad \text{if } y_i = 1$$

$$\mathbf{w^T x_i} + b \leq -1 \quad \text{if } y_i = -1$$

- For support vectors, the inequality becomes an equality; then, since each example's distance from the

- hyperplane is $r = \dfrac{\mathbf{w}^T\mathbf{x} + b}{\|\mathbf{w}\|}$ the margin is: $\rho = \dfrac{2}{\|\mathbf{w}\|}$

Linear SVMs Mathematically (cont.)

- Then we can formulate the *quadratic optimization problem:*

Find $\mathbf{w}$ and b such that

$$\rho = \frac{2}{\|\mathbf{w}\|} \quad \text{is maximized and for all } \{(\mathbf{x_i}, y_i)\}$$

$$\mathbf{w^T x_i} + b \geq 1 \text{ if } y_i = 1; \quad \mathbf{w^T x_i} + b \leq -1 \quad \text{if } y_i = -1$$

A better formulation:

Find $\mathbf{w}$ and b such that

$$\Phi(\mathbf{w}) = \tfrac{1}{2}\, \mathbf{w}^T\mathbf{w} \quad \text{is minimized and for all } \{(\mathbf{x_i}, y_i)\}$$

$$y_i\,(\mathbf{w^T x_i} + b) \geq 1$$

Solving the Optimization Problem

Find $\mathbf{w}$ and b such that

$\Phi(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ is minimized and for all $\{(\mathbf{x_i}, y_i)\}$

$y_i\,(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- Need to optimize a *quadratic* function subject to *linear* constraints.
- Quadratic optimization problems are a well-known class of mathematical programming problems, and many (rather intricate) algorithms exist for solving them.
- The solution involves constructing a *dual problem* where a *Lagrange multiplier* α_i is associated with every constraint in the primary problem:

Find $\alpha_1 \ldots \alpha_N$ such that

$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ is maximized and

(1) $\Sigma\alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all α_i

The Optimization Problem Solution

- The solution has the form:

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w}^T \mathbf{x_k} \text{ for any } \mathbf{x_k} \text{ such that } \alpha_k \neq 0$$

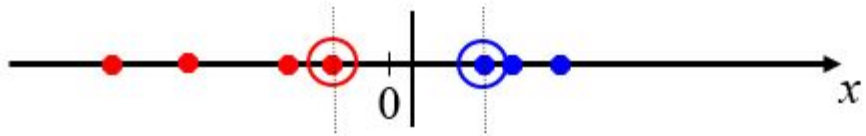
- Each non-zero α_i indicates that corresponding $\mathbf{x_i}$ is a support vector.
- Then the classifying function will have the form:

$$f(\mathbf{x}) = \Sigma \alpha_i y_i \mathbf{x_i}^T \mathbf{x} + b$$

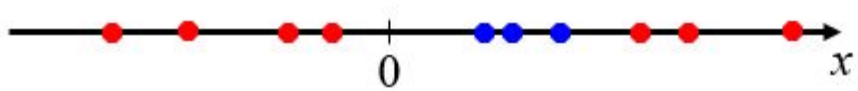
- Notice that it relies on an *inner product* between the test point $\mathbf{x}$ and the support vectors $\mathbf{x_i}$ – we will return to this later!
- Also keep in mind that solving the optimization problem involved computing the inner products $\mathbf{x_i}^T \mathbf{x_j}$ between all training points!

Non-linear SVMs

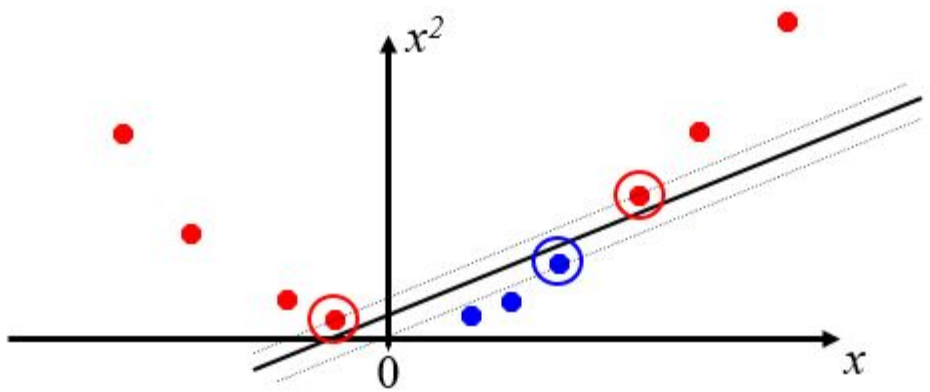
- Datasets that are linearly separable with some noise work out great:



- But what are we going to do if the dataset is just too hard?

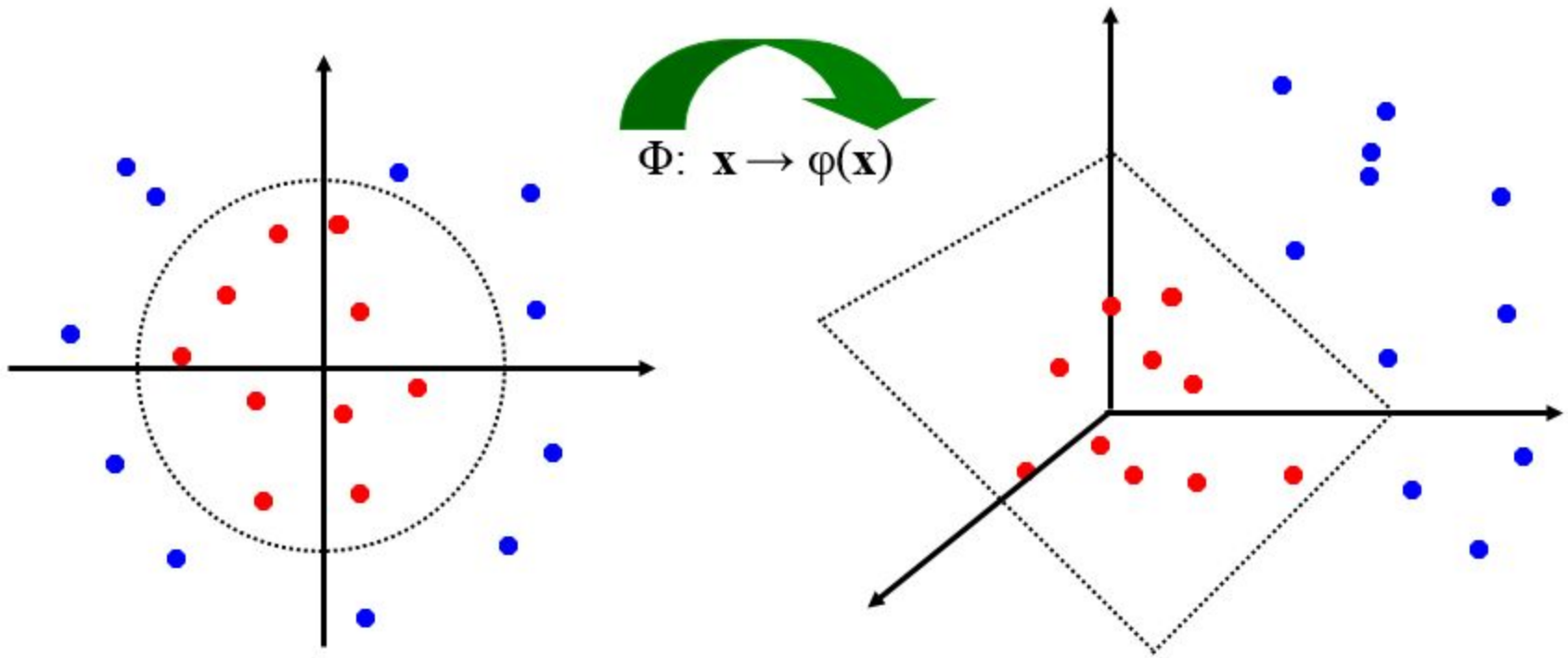


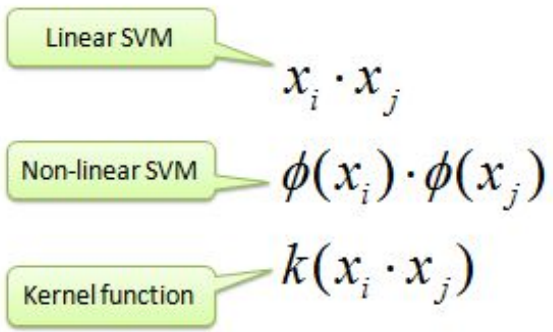
- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature spaces

- General idea: the original feature space can always be mapped to some higher-dimensional feature space where the training set is separable:





If the data distribution is fundamentally non-linear,

- The trick is to transform the data to a higher dimension so the data will be linearly separable.
- The optimization term turns out to be a dot product of the transformed points in the high-dimension space, which is found to be equivalent to performing a kernel function in the original (before transformation) space.
- The kernel function provides a cheap way to equivalently transform the original point to a high dimension (since we don't actually transform it) and perform the quadratic optimization in that high-dimension space.

The "Kernel Trick"

- The linear classifier relies on inner product between vectors $K(\mathbf{x_i},\mathbf{x_j})=\mathbf{x_i}^T\mathbf{x_j}$
- If every datapoint is mapped into high-dimensional space via some transformation Φ: $\mathbf{x} \rightarrow \varphi(\mathbf{x})$, the inner product becomes:

$$K(\mathbf{x_i},\mathbf{x_j})= \varphi(\mathbf{x_i})^T\varphi(\mathbf{x_j})$$

- A *kernel function* is some function that corresponds to an inner product into some feature space.
- Example:

  2-dimensional vectors $\mathbf{x}=[x_1 \;\; x_2]$; let $K(\mathbf{x_i},\mathbf{x_j})=(1 + \mathbf{x_i}^T\mathbf{x_j})^2$,

  Need to show that $K(\mathbf{x_i},\mathbf{x_j})= \varphi(\mathbf{x_i})^T\varphi(\mathbf{x_j})$:

$$K(\mathbf{x_i},\mathbf{x_j})=(1 + \mathbf{x_i}^T\mathbf{x_j})^2 = 1+ x_{i1}^2x_{j1}^2 + 2\, x_{i1}x_{j1}\, x_{i2}x_{j2}+ x_{i2}^2x_{j2}^2 + 2x_{i1}x_{j1} + 2x_{i2}x_{j2}=$$
$$= [1 \;\; x_{i1}^2 \;\; \sqrt{2}\, x_{i1}x_{i2} \;\; x_{i2}^2 \;\; \sqrt{2}x_{i1} \;\; \sqrt{2}x_{i2}]^T\, [1 \;\; x_{j1}^2 \;\; \sqrt{2}\, x_{j1}x_{j2} \;\; x_{j2}^2 \;\; \sqrt{2}x_{j1} \;\; \sqrt{2}x_{j2}] =$$
$$= \varphi(\mathbf{x_i})^T\varphi(\mathbf{x_j}), \quad \text{where } \varphi(\mathbf{x}) = [1 \;\; x_1^2 \;\; \sqrt{2}\, x_1x_2 \;\; x_2^2 \;\; \sqrt{2}x_1 \;\; \sqrt{2}x_2]$$

Examples of Kernel Functions

- Linear: $K(\mathbf{x_i},\mathbf{x_j}) = \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j}$
- Polynomial of power p: $K(\mathbf{x_i},\mathbf{x_j}) = (1+ \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j})^p$
- Gaussian (radial-basis function network): $K(\mathbf{x_i},\mathbf{x_j}) = e^{-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{2\sigma^2}}$
- Two-layer perceptron: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0 \mathbf{x_i}^{\mathbf{T}}\mathbf{x_j} + \beta_1)$

Non-linear SVMs Mathematically

- Dual problem formulation:

Find $\alpha_1 \ldots \alpha_N$ such that
$\mathbf{Q}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j})$ is maximized and
(1) $\Sigma\alpha_i y_i = 0$
(2) $\alpha_i \geq 0$ for all α_i

- The solution is:

$f(\mathbf{x}) = \Sigma\alpha_i y_i K(\mathbf{x_i}, \mathbf{x_j}) + b$

- Optimization techniques for finding α_i's remain the same!

Bayesian Network and Naïve Bayes

- From a probabilistic viewpoint, the predictive problem can be viewed as a conditional probability estimation; trying to find Y where **P(Y | X)** is maximized.

- From the Bayesian rule, **P(Y | X) == P(X | Y) * P(Y) / P(X)**

  Posterior = Likelihood * Priori / Evidence

- This is equivalent to finding Y where **P(X | Y) * P(Y)** is maximized.

Let's say the input X contains 3 categorical features— X1, X2, X3.

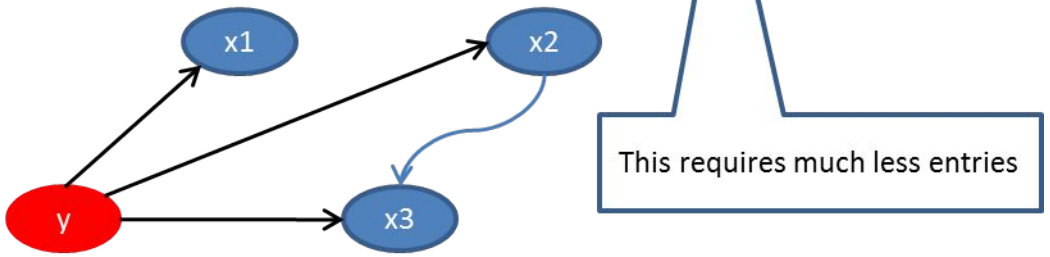
In the general case, we assume each variable can potentially influence any other variable.

Therefore the joint distribution becomes:

$$P(X \mid Y) = P(X1 \mid Y) * P(X2 \mid X1, Y) * P(X3 \mid X1, X2, Y)$$

Bayesian network (some independence assumption)

$$P(x1 \wedge x2 \wedge x3 \mid y) = P(x1 \mid y) * P(x2 \mid y \wedge x1) * P(x3 \mid y \wedge x1 \wedge x2)$$
$$= P(x1 \mid y) * P(x2 \mid y) * P(x3 \mid y \wedge x2)$$



- Assuming X1,X2 and X3 to be conditionally independent,

$$P(X \mid Y) == P(X1 \mid Y) * P(X2 \mid Y) * P(X3 \mid Y),$$

we need to find the Y that maximizes $P(X1 \mid Y) * P(X2 \mid Y) * P(X3 \mid Y) * P(Y)$

- Each term on the right hand side can be learned by counting the training data. Therefore we can estimate $P(Y \mid X)$ and pick Y to maximize its value.
- But it is possible that some patterns never show up in training data, e.g., $P(X1=a \mid Y=y)$ is 0. To deal with this situation, we pretend to have seen the data of each possible value one more time than we actually have.

$$P(X1=a \mid Y=y) == (count(a, y) + 1) / (count(y) + m)$$

…where m is the number of possible values in X1.

- When the input features are numeric, say a = 2.75, we can assume X1 is the normal distribution. Find out the mean and standard deviation of X1 and then estimate P(X1=a) using the normal distribution function.

Clustering Algorithms

- Goal: grouping a collection of objects into subsets or “clusters”, such that those within each cluster are more closely related to one another than objects assigned to different clusters

- Two essential components of cluster analysis:
  - **Distance measure:** A notion of distance or similarity of two objects: When are two objects close to each other?
  - **Cluster algorithm:** A procedure to minimize distances of objects within groups and/or maximize distances between groups

Distance Measures

Names	Formula
Euclidean distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
maximum distance	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the Covariance matrix
Cosine similarity	$\frac{a \cdot b}{\|a\|\|b\|}$

Clustering algorithms

- Popular algorithms for clustering
  - hierarchical clustering
  - K-means
  - SOMs (Self-Organizing Maps)
  - autoclass, mixture models...
- Hierarchical clustering allows the choice of the dissimilarity matrix.
- k-Means and SOMs take original data directly as input. Attributes are assumed to live in Euclidean space.

Hierarchical clustering

Agglomerative clustering:

1. Each object is assigned to its own cluster
2. Iteratively:
   - the two most similar clusters are joined and replaced by a new one
   - the distance matrix is updated with this new cluster replacing the two joined clusters

(divisive clustering would start from a big cluster)

Distance between two clusters

- Single linkage uses the smallest distance

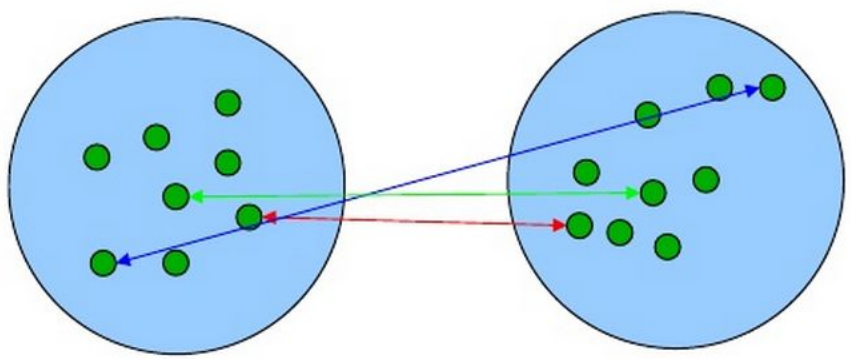
$$d_S(G,H) = \min_{i \in G,\, j \in H} d_{ij}$$

- Complete linkage uses the largest distance

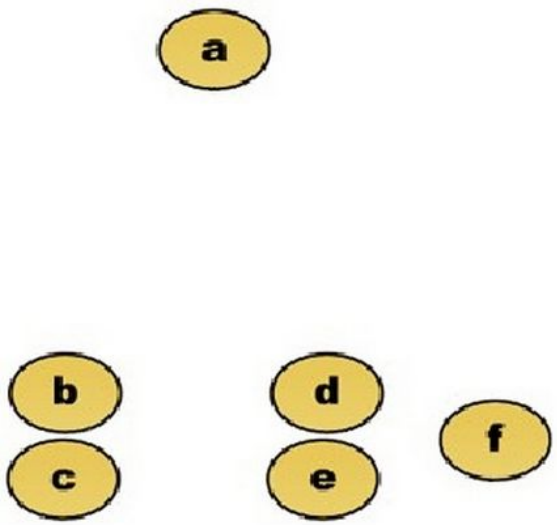
$$d_C(G,H) = \max_{i \in G,\, j \in H} d_{ij}$$

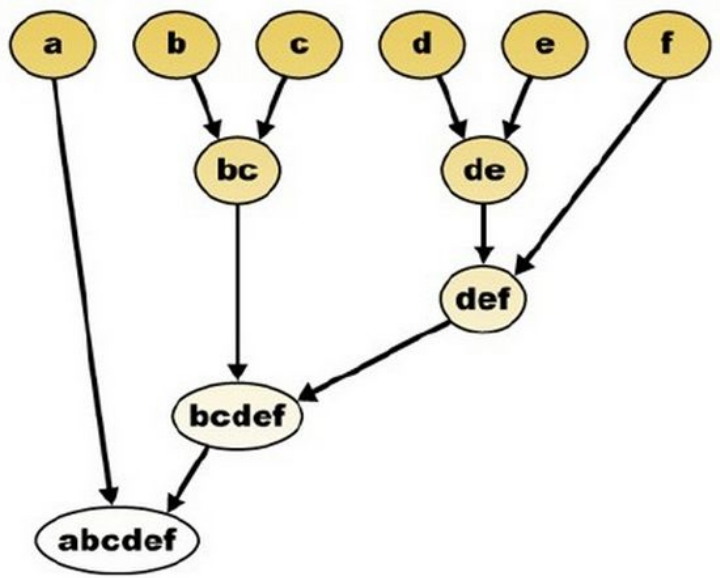
- Average linkage uses the average distance

$$d_A(G,H) = \frac{1}{N_G N_H} \sum_{i \in G} \sum_{j \in H} d_{ij}$$



Hierarchical clustering

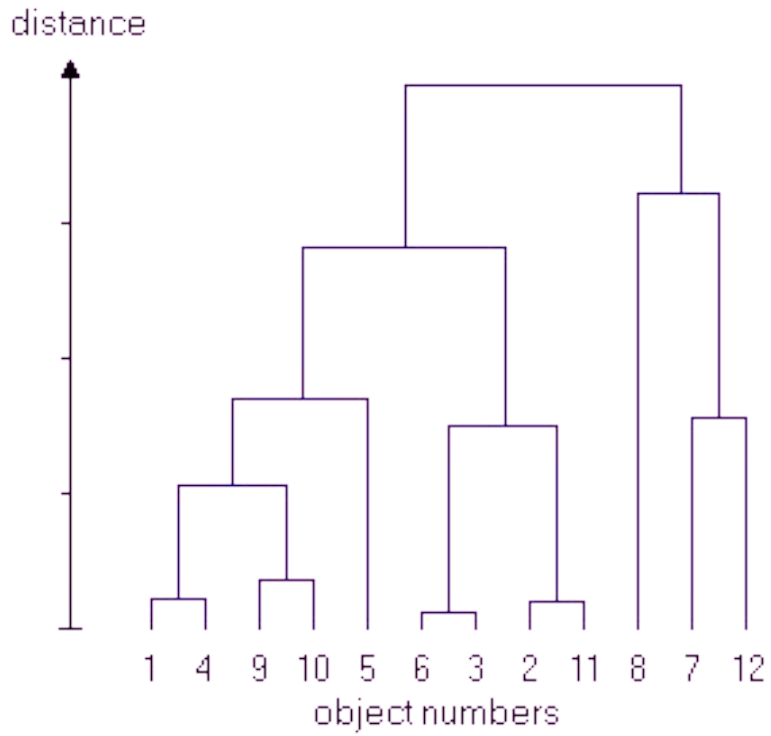




(wikipedia)

Dendrogram

- Hierarchical clustering are visualized through dendrograms
  - Clusters that are joined are combined by a line
  - Height of line is distance between clusters
  - Can be used to determine visually the number of clusters



Hierarchical clustering

- Strengths
  - No need to assume any particular number of clusters
  - Can use any distance matrix
  - Find sometimes a meaningful taxonomy
- Limitations
  - Find a taxonomy even if it does not exist
  - Once a decision is made to combine two clusters it cannot be undone
  - Not well theoretically motivated

k-Means clustering

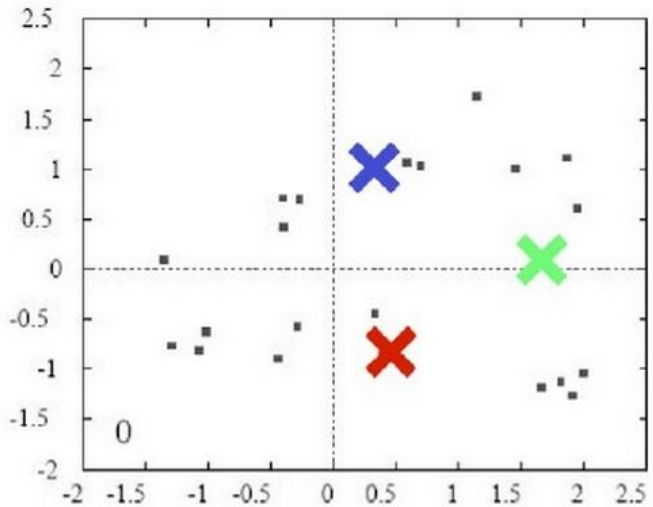
- Partitioning algorithm with a prefixed number k of clusters
- Use Euclidean distance between objects
- Try to minimize the sum of intra-cluster variances

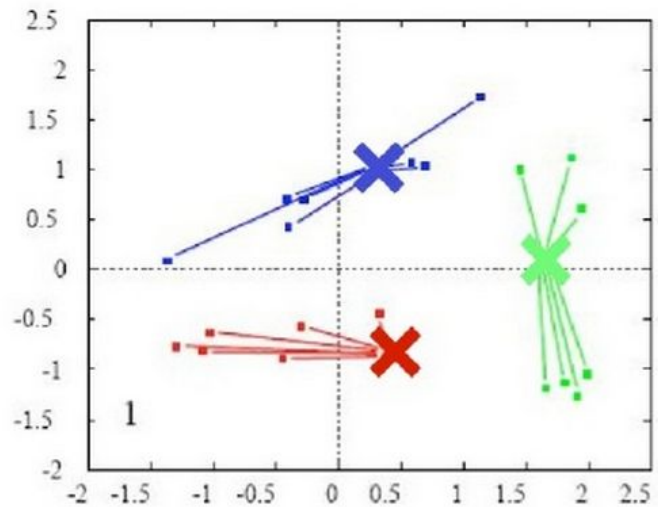
$$\sum_{j=1}^{k} \sum_{o \in Cluster\ j} d^2(o, c_j)$$

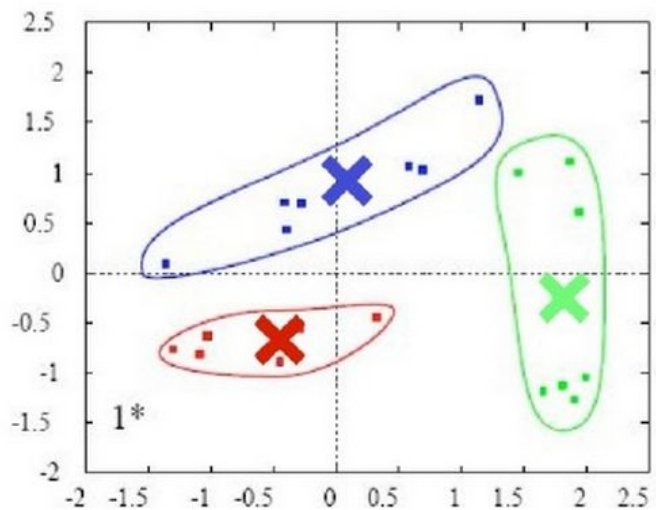
where c_j is the center of cluster j and d^2 is the Euclidean distance

k-Means clustering

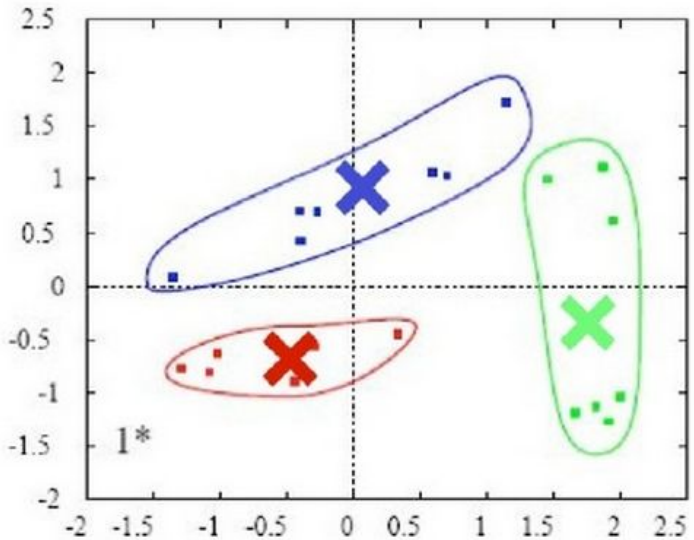
start

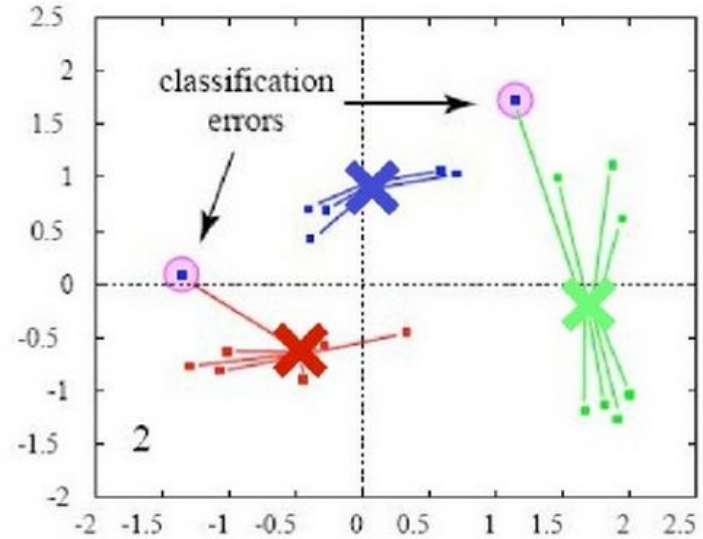


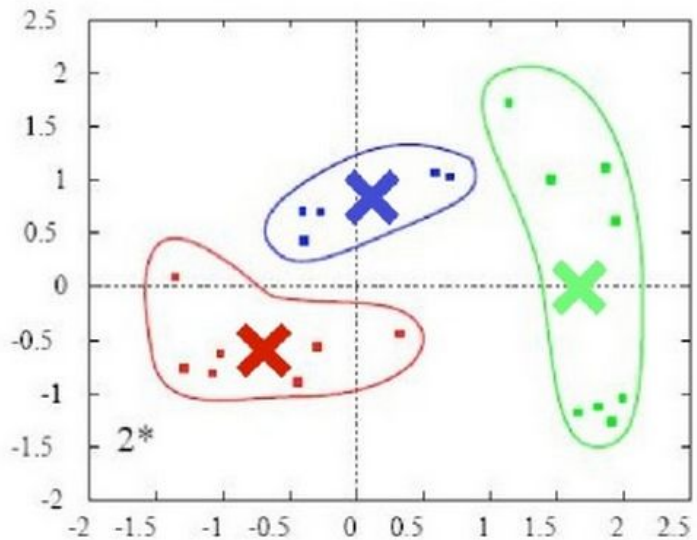




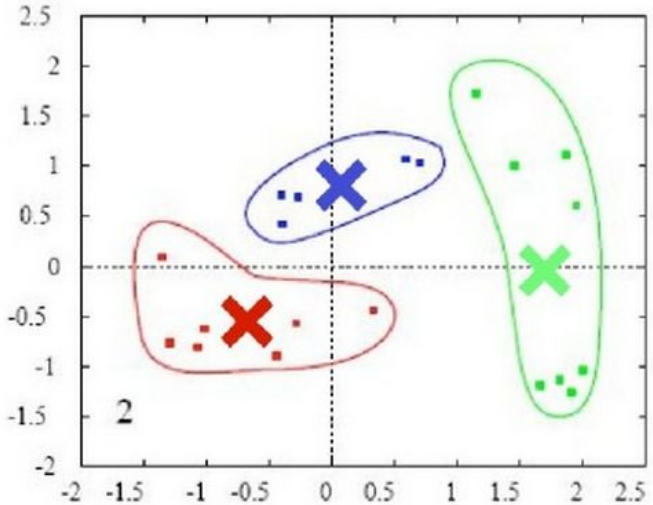
k-Means clustering

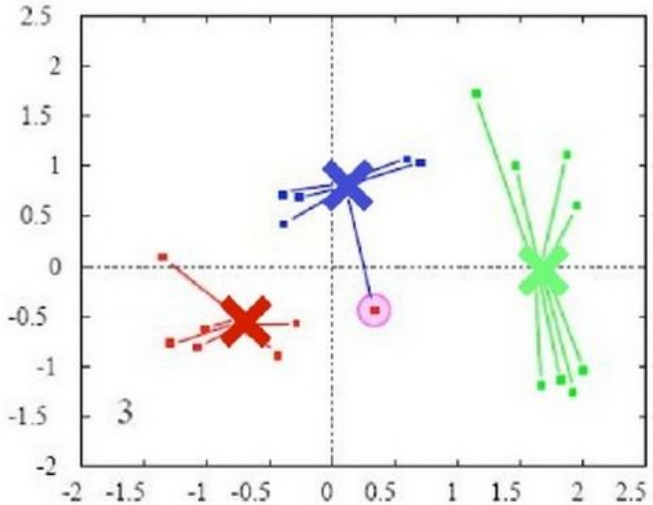


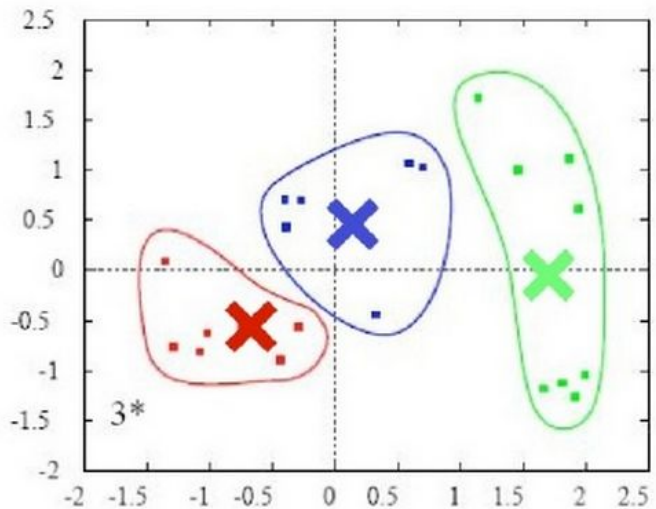




k-Means clustering







end

k-Means clustering

- Strengths
  - Simple, understandable
  - Can cluster any new point (unlike hierarchical clustering)
  - Well motivated theoretically
- Limitations
  - Must fix the number of clusters beforehand
  - Sensitive to the initial choice of cluster centers
  - Sensitive to outliers

Thank You !

References

http://refcardz.dzone.com/refcardz/machine-learning-predictive
http://en.wikipedia.org/wiki/Machine_learning
http://en.wikipedia.org/wiki/List_of_machine_learning_algorithms
http://en.wikipedia.org/wiki/Data_mining
http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/
http://machinelearningmastery.com/machine-learning-foundations/
http://www.cs.cmu.edu/~tom/pubs/MachineLearning.pdf
http://www.holehouse.org/mlclass/10_Advice_for_applying_machine_learning.html
http://www.holehouse.org/mlclass/17_Large_Scale_Machine_Learning.html
http://courses.cs.tamu.edu/choe/14spring/633/#WeeklySchedule
http://work.caltech.edu/library/
http://blog.prediction.io/machine-learning
http://www.slideshare.net/butest/an-introduction-to-machine-learning?next_slideshow=1