Structure prediction of globular and membrane proteins

February 18 - March 18

Teachers

- Erik Lindahl, erik.lindahl@gmail.com
- Arne Elofsson, arne@bioinfo.se
- Karolis Uziela karolis.uziela@scilifelab.se (Assistant)
- Oxana Sachenkova, oxana.sachenkova@scilifelab.se (Assistant)
- Stefan Fleischmann, sfle@kth.se (Tech. assistant)

Course Overview

To pass the course you will have to:

- Develop a functional predictor for the "feature" assigned to you
- Write a report where you compare your predictor as well as review the field for your predictors
- Complete software carpentry exercises

Weekly schedule

- Bioinformatics club at scilifelab (Fridays 10am) https://docs.google.com/spreadsheets/d/1JoZ0lNCfco5L0WA4SoUoptusi1CyaZYqHzTJoZ-BLf4/edit#gid=0
- Elofsson Lab group meeting (Fridays 13:30)
- Lindahl group meeting (Mondays 13:00)
- Assistants available daily (13:00, Black hole)

The Project

Develop your own predictor using Support Vector Machines!

Globular:

- DSSP Helix predictor (H vs rest)
- DSSP Sheet predictor (S vs rest)
- DSSP Coil predictor (C vs rest)
- STRIDE Helix predictor (H vs rest)
- STRIDE Sheet predictor (S vs rest)
- Burried residue predictor (b vs e)

Membrane-alpha

- Membrane region predictor (M vs rest)

Membrane-beta

- Membrane region predictor (P and L vs rest)

Datasets can be found on Mondo (Resources->Final_datasets)

Support is given at specific times and by email through scilifelab-project-2016@googlegroups.com mailing list.

Project points

1. Extract the feature from your database
2. Create cross-validated sets (make sure that there are no homologs in the same set by running CD-hit)
3. Train a SVM using single sequence information
4. **Add evolutionary information by running psi-blast and extracting the information**
5. **Train a SVM using multiple sequence information**
6. **Optimize the performance of the SVM**
7. Analyze the results and compare it to previous work
8. **Set up a web-server**
9. Review the state of art for your predictor
10. Write a report

Deadlines

Week 1

- Thu: Course starts
- Fri: Computer set up and programs installed

Week 2

- Mon Project plan submitted (make deadlines for yourself)
- Fri: Finished the Software carpentry tasks

Week 3

- Mon: Demonstration of program that can create input for svm-light
- Fri: A list of "state-of-the-art" papers for your field

Week 4

- Mon: Outline of report

Week 5

- Tue The predictor working
- Fri: Report submitted.

Additional tasks

Week 1 (Day 1-2)

- Bash etc http://swcarpentry.github.io/shell-novice/
- Python http://swcarpentry.github.io/python-novice-inflammation/

Week 2

- Git http://swcarpentry.github.io/git-novice
- More Python: https://github.com/bast/python-tdd-exercises
- Data structures in python http://www.datacarpentry.org/python-ecology/

Literature

- Jones DT. (1999) Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 292: 195-202.
- Noble WS (2009) A Quick Guide to Organizing Computational Biology Projects. PLoS Comput Biol 5(7): e1000424.
- Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

VPN

- In order to connect to your workstations from home, you have to set up VPN. Follow instructions on http://intranet.scilifelab.se/it/scilifelab-vpn/ (works only when you are in scilifelab network)
- The computer names are: elofsson, illergard, elof01...05
- So you have to type ssh <username>@<pc-name>.scilifelab.se

Introduction to Machine Learning

Slides by courtesy of Lukas Käll

Machine learning - Wikipedia, the free encyclopedia

http://en.wikipedia.org/wiki/Machine_learning

machine learning

Apple Google Maps YouTube Wikipedia

Machine learning - Wikipedia, the...

Try Beta **Log in / create account**

article discussion **edit this page** history

WIKIPEDIA
The Free Encyclopedia

navigation

- Main page
- Contents
- Featured content
- Current events
- Random article

search

Go Search

interaction

- About Wikipedia
- Community portal
- Recent changes
- Contact Wikipedia
- Donate to Wikipedia
- Help

toolbox

- What links here
- Related changes
- Upload file
- Special pages
- Printable version

Machine learning

From Wikipedia, the free encyclopedia

This article **does not cite any references or sources**. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. *(August 2008)*

For the journal, see Machine Learning (journal).

Artificial intelligence portal

Machine learning is a scientific discipline that is concerned with the design and development of algorithms that allow computers to learn based on data, such as from sensor data or databases. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data. Hence, machine learning is closely related to fields such as statistics, probability theory, data mining, pattern recognition, artificial intelligence, adaptive control, and theoretical computer science.

Contents [hide]

Applications [edit]

Applications of ML within Bioinformatics

"Classical" Bioinformatics

- Gene prediction
- Protein family classification
- Protein structure prediction
- Secondary structure prediction
- Transmembrane topology prediction
- Protein compartment prediction
- Sequence alignment

Interpretation of high throughput experiments

- Chromatin Structure prediction by DNA hypersensitive site assays
- Copy Number Variation
- Transcription factor analysis by ChIP-Seq
- Peptide/protein inference from shotgun proteomics
- Clustering analysis of transcriptomics/ proteomics data

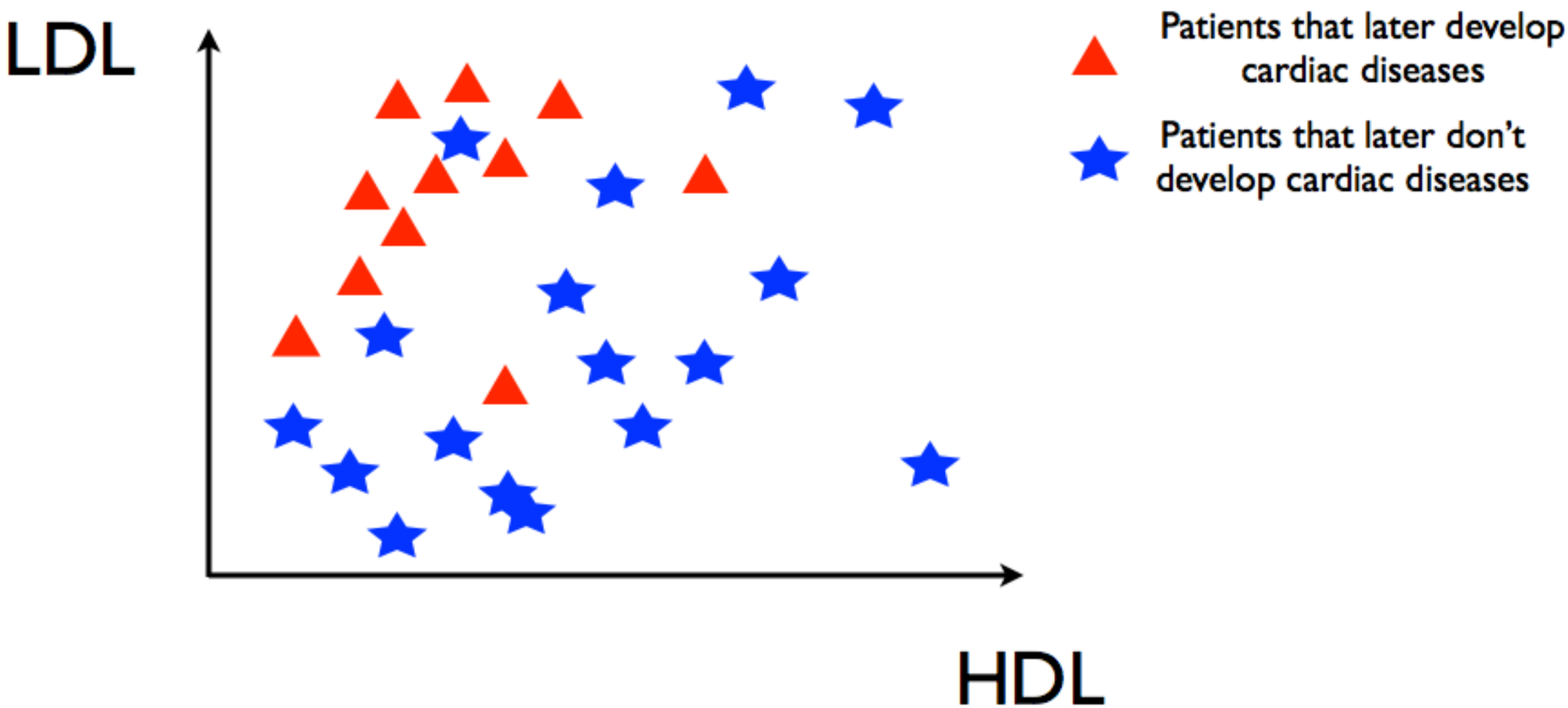
Terms

- Regression: Given a set of features of an example predict one (or more) variables (dependent variables)
- Classification: Given a set of features of an example predict which class the example belongs to

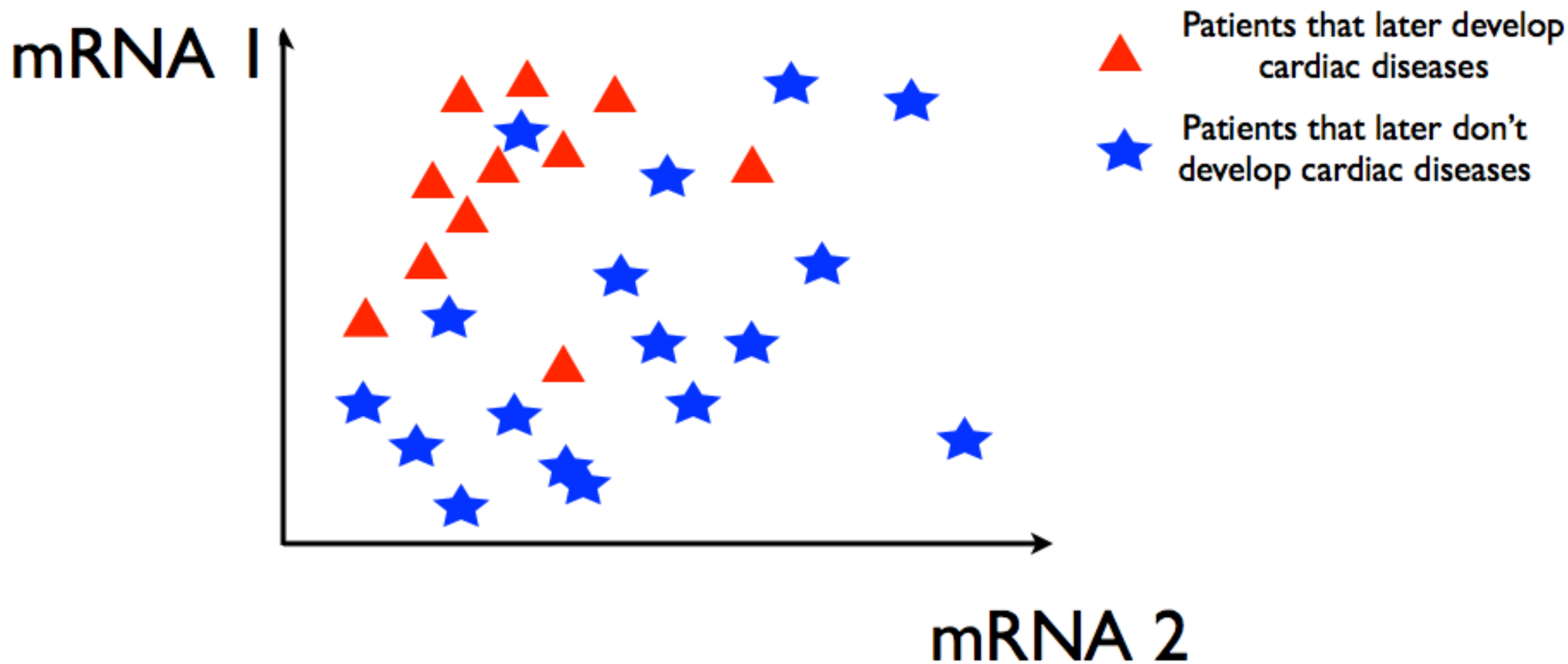
More Terms

- Supervised learning: All training examples are labeled
- Unsupervised learning: No examples are labeled (Clustering)
- Semi-supervised learning a few examples are labeled, but the bulk of our set is unlabeled

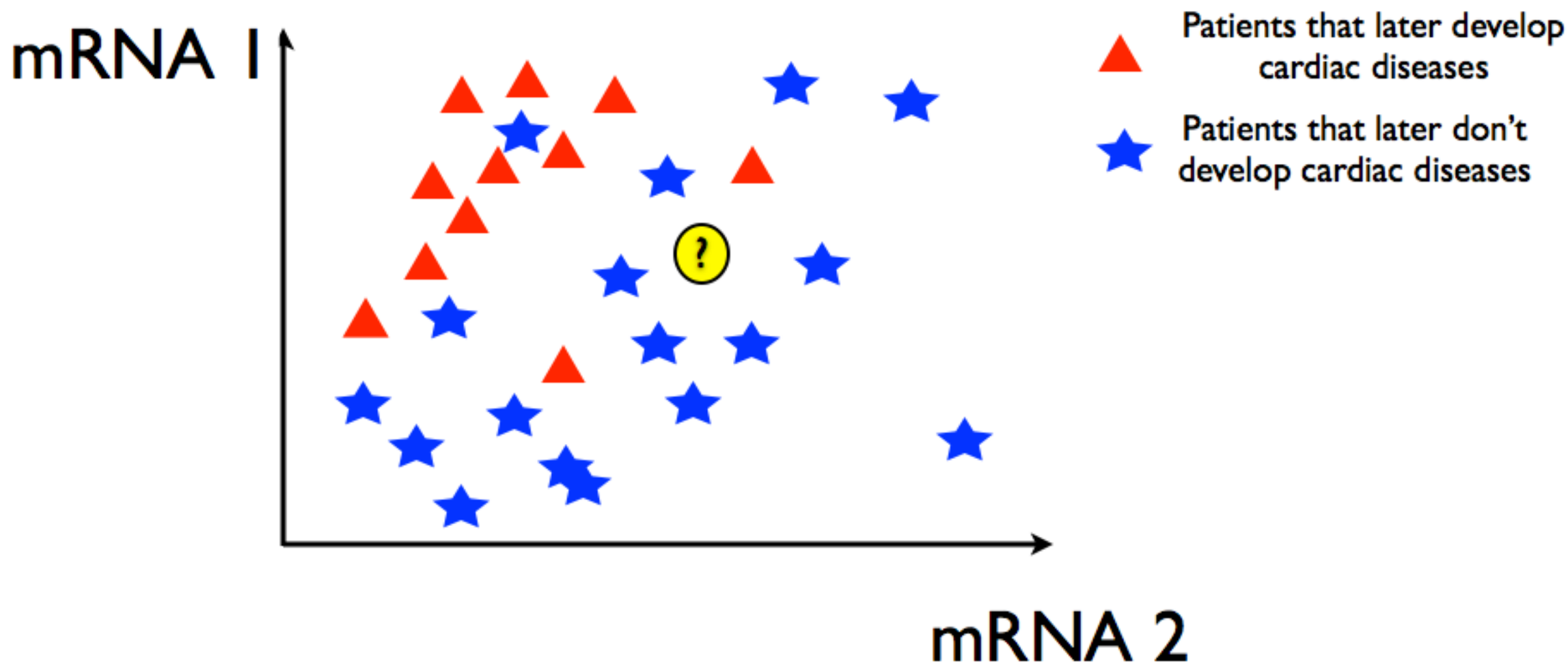
Classification - supervised learning



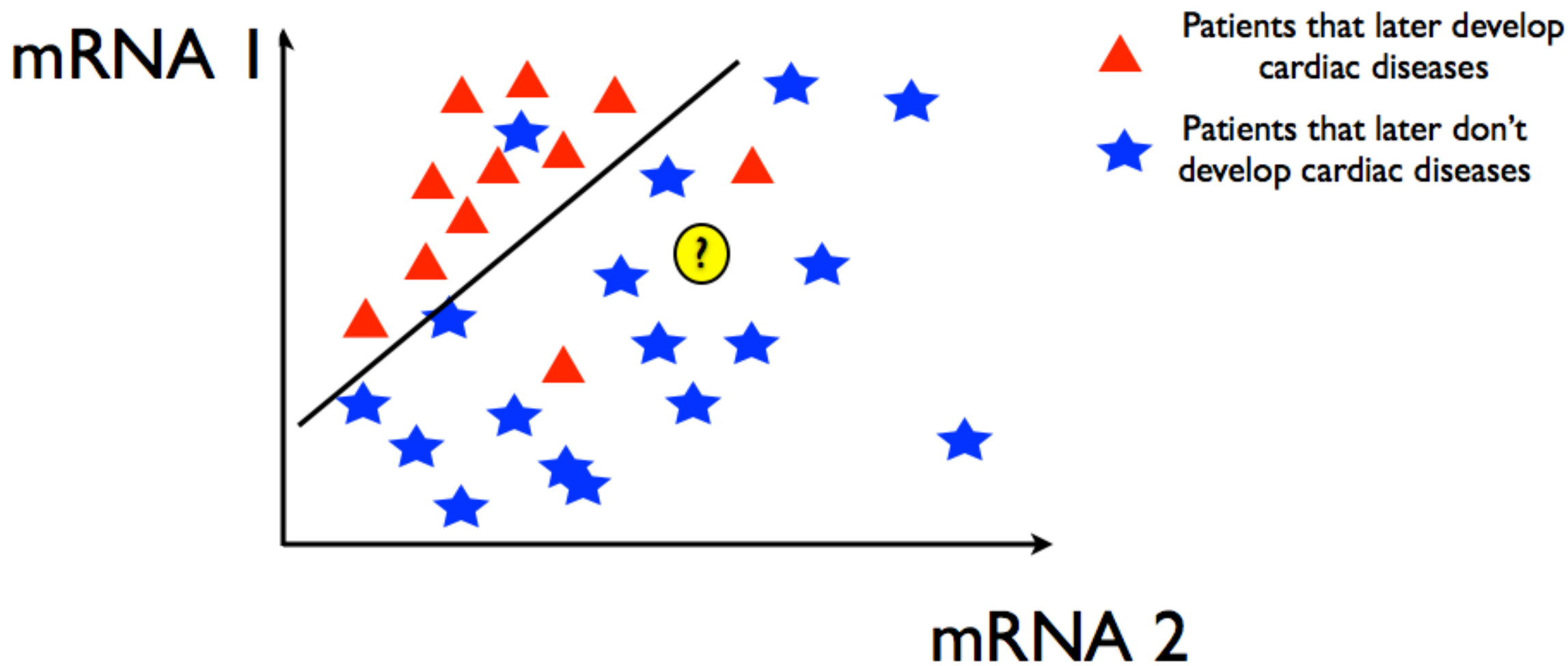
Classification - supervised learning



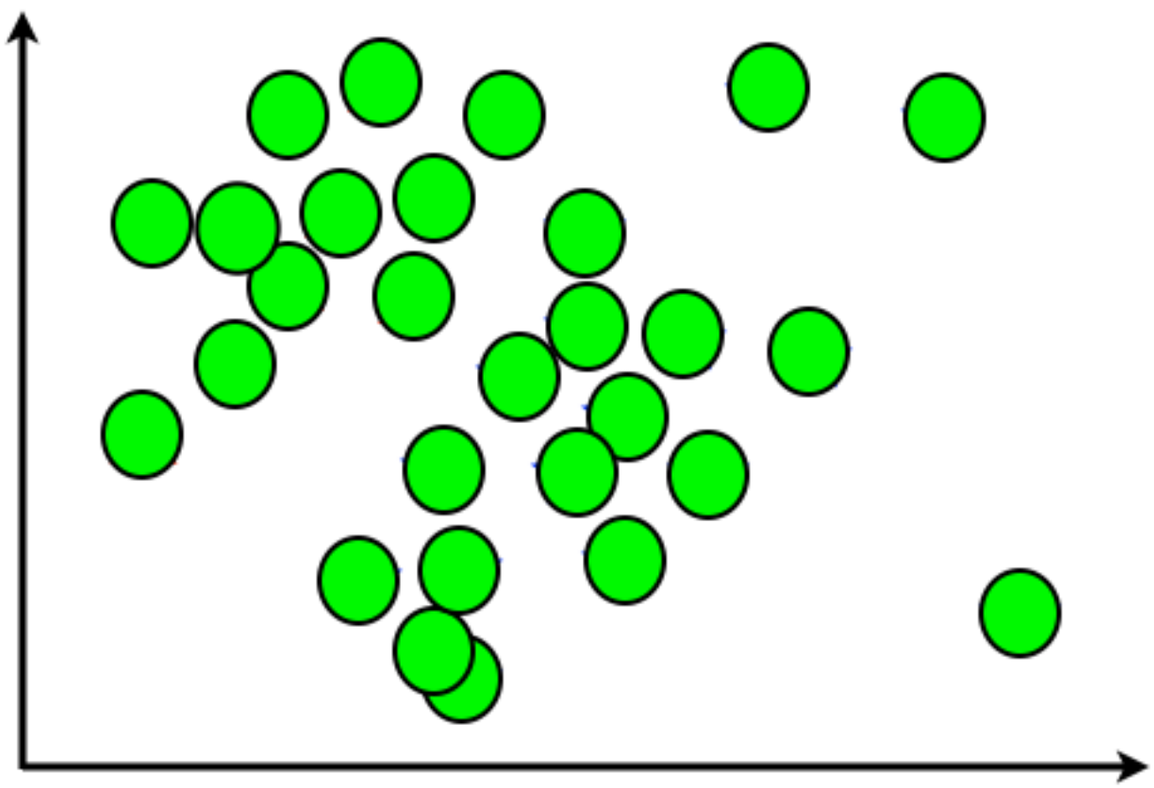
Classification - supervised learning



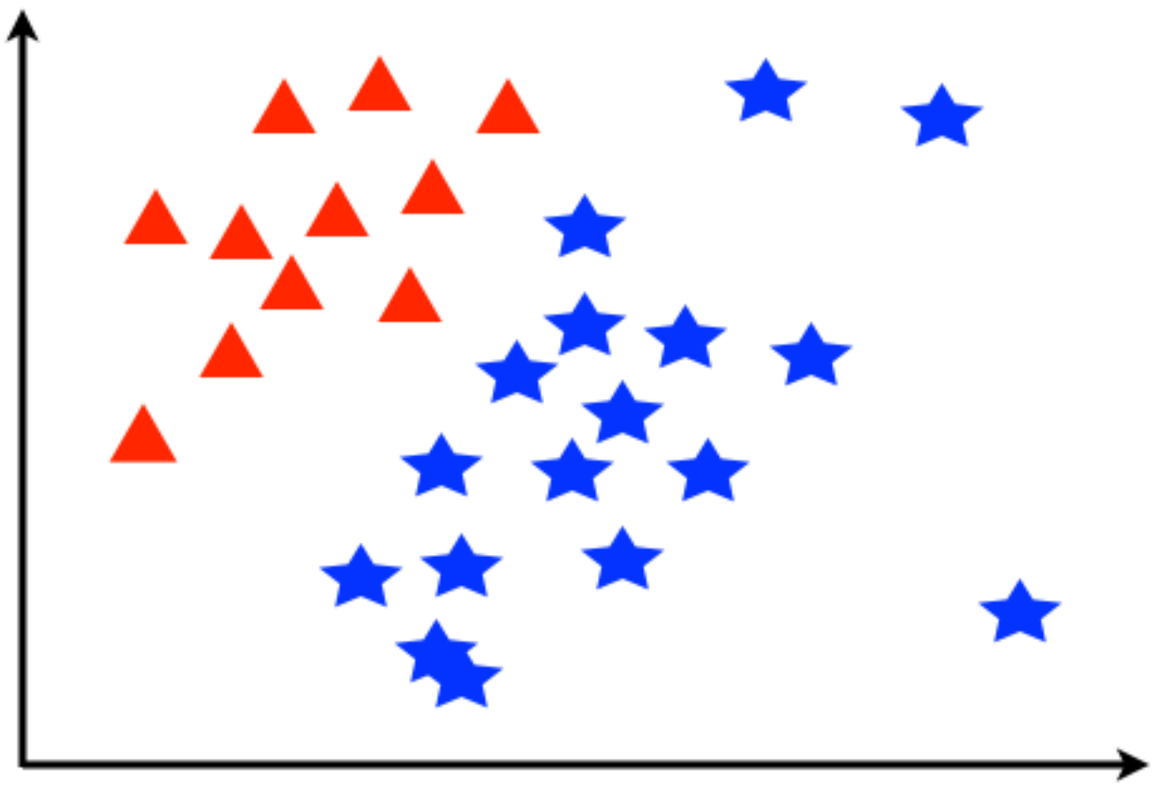
Classification - supervised learning



Classification - unsupervised learning



Classification - unsupervised learning



Supervised learning

Terms again

- Generative models - design a model of each class of data and calculate a probability that each example was generated by each model
- Discriminative models - build a model that separates the classes of data

Choices

To build a good classifier we need to select:

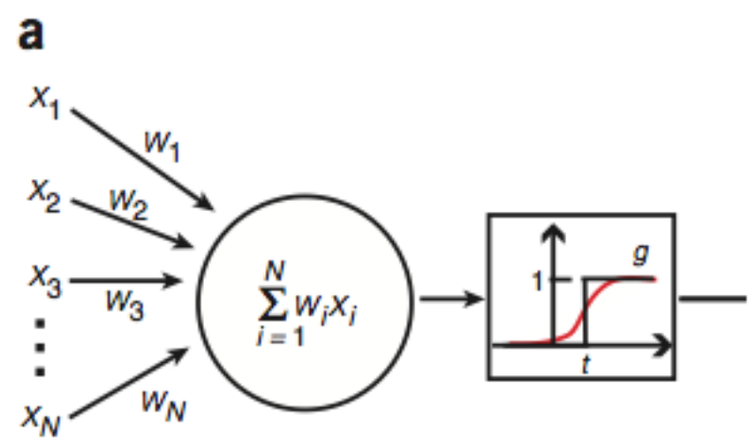
- Relevant features
- Learner
- Validation strategy

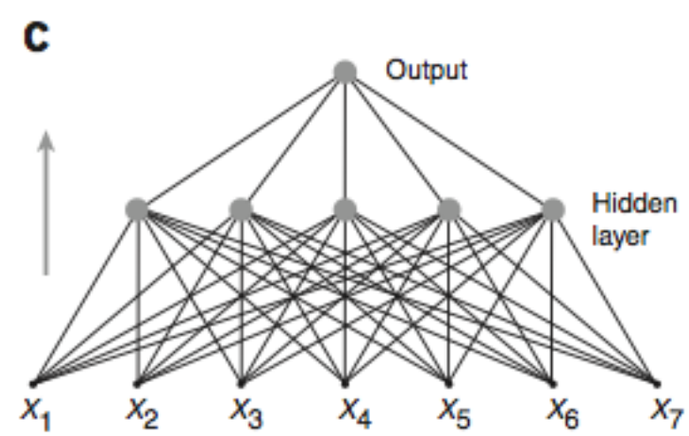
Features

- Normalize features so that they have a uniform spread
- Sparse representation of amino acids - each position in a string

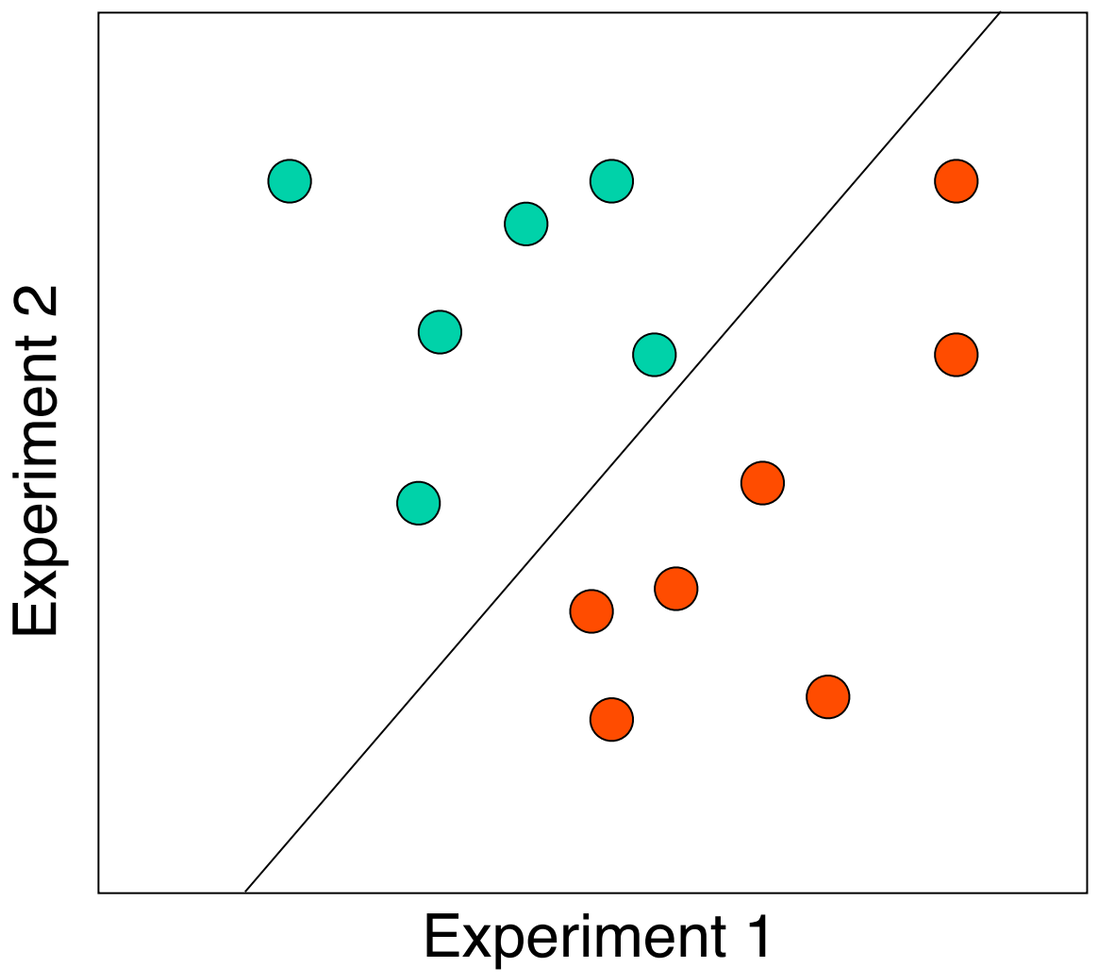
A	C	D	E	F	...	Y
0	1	0	0	0	...	0

Neural Nework





Support Vector Machines



- Learning in SVMs involves finding a hyperplane (decision surface) that separates the examples of one class from another.

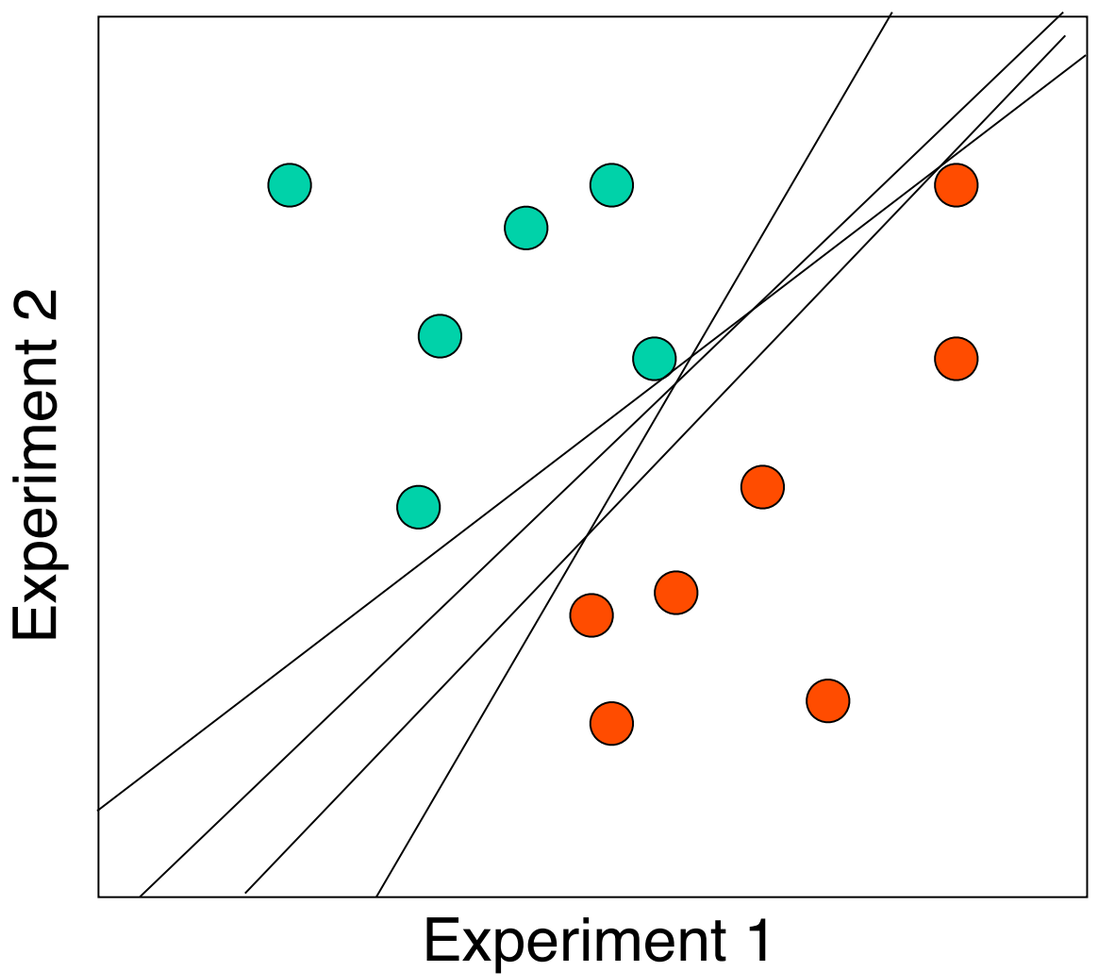
Support Vector Machines

- For the ith example, let x_i be the vector of expression measurements, and y_i be +1, if the example is in the class of interest; and –1, otherwise
- The hyperplane is given by:

$$w \cdot x + b = 0$$

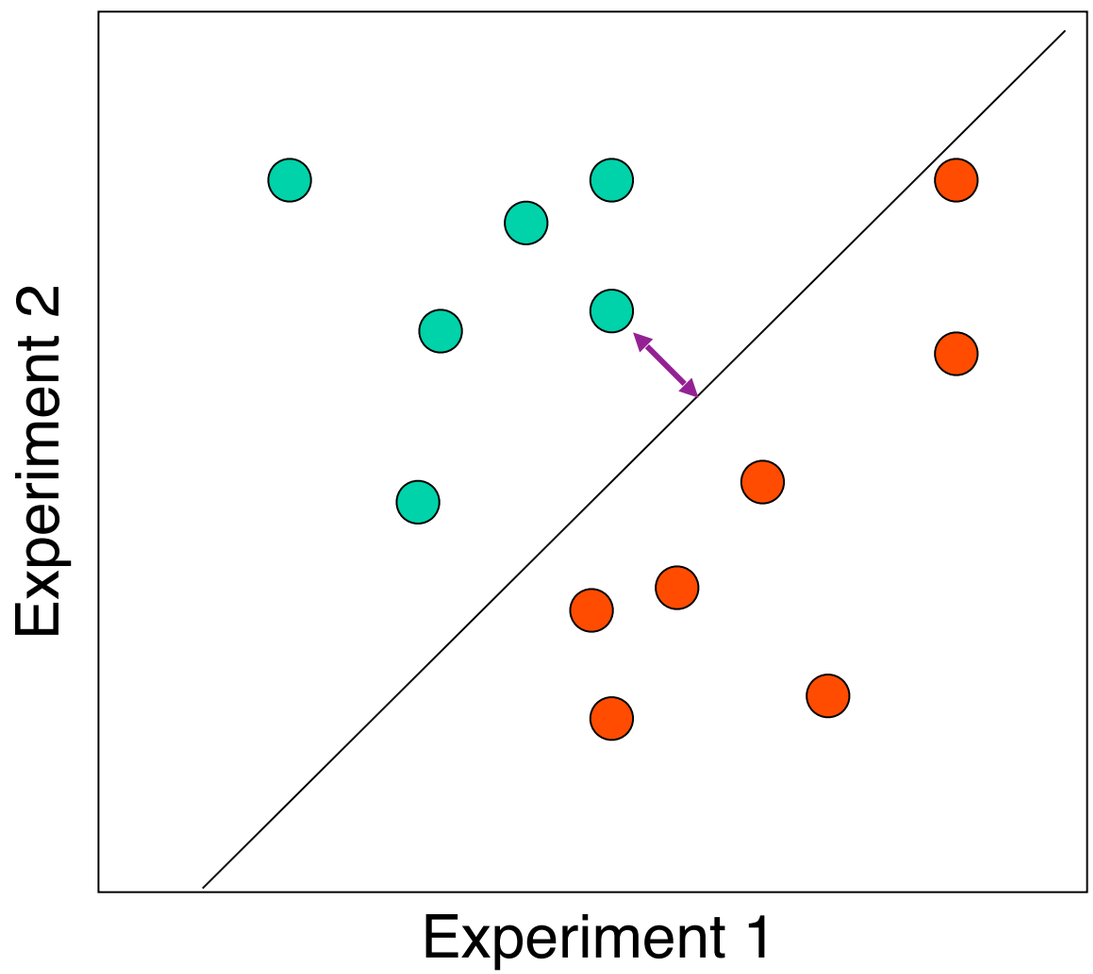
where b = constant and w= vector of weights

Support Vector Machines



- There may be many such hyperplanes..
- Which one should we choose?

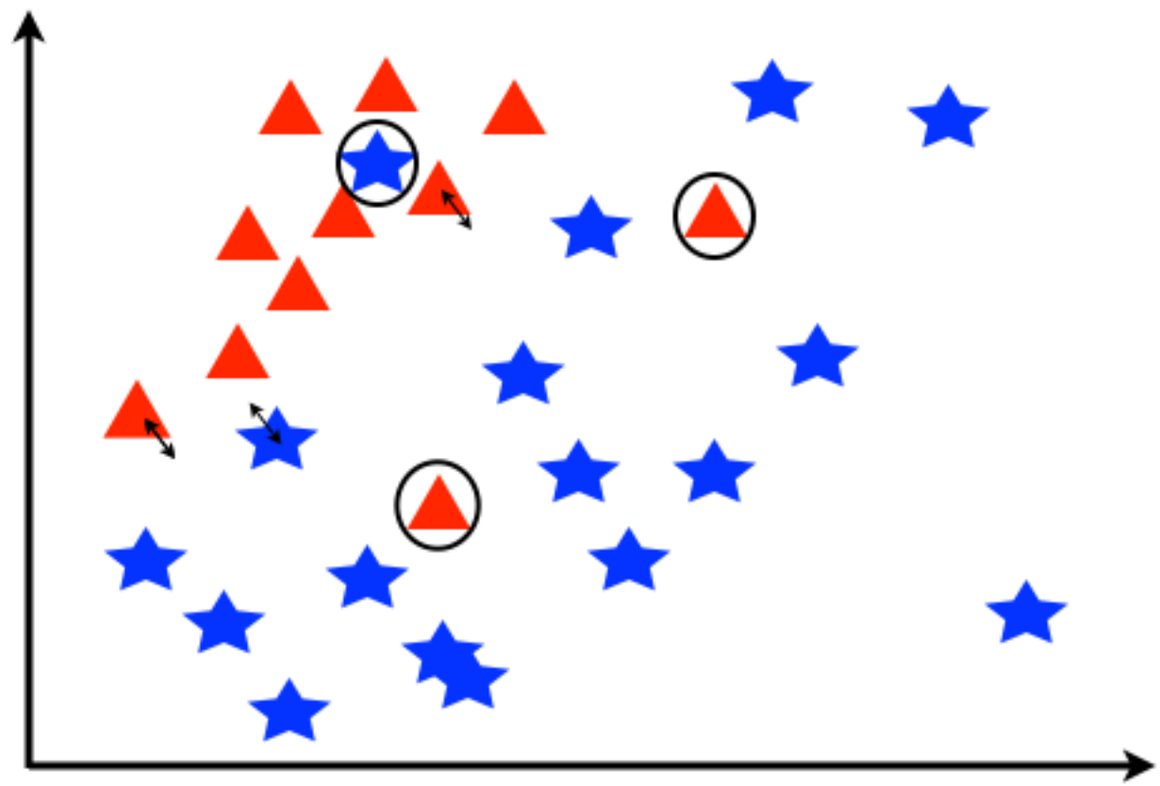
Maximizing the Margin



- Key SVM idea
  - Pick the hyperplane that maximizes the margin—the distance to the hyperplane from the closest point
  - Motivation: Obtain tightest possible bounds on the error rate of the classifier.

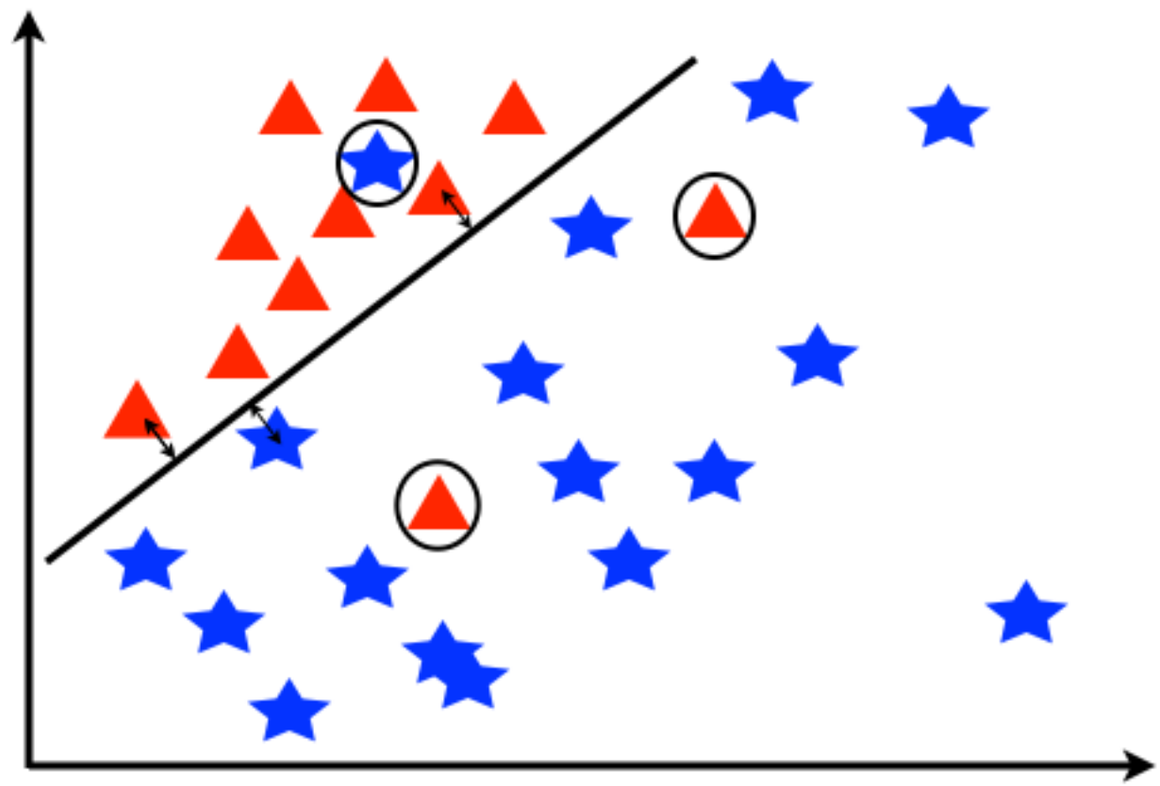
Support Vector Machine (SVM)

- Select a Maximum-margin separating hyper plane
- Soft margin, i.e. allow some data points to push their way through the margin of the separating hyperplane without affecting the end result



Support Vector Machine (SVM)

- Select a Maximum-margin separating hyper plane
- Soft margin, i.e. allow some data points to push their way through the margin of the separating hyperplane without affecting the end result



SVM: Finding the Hyperplane

- Can be formulated as an optimization task
  - Minimize

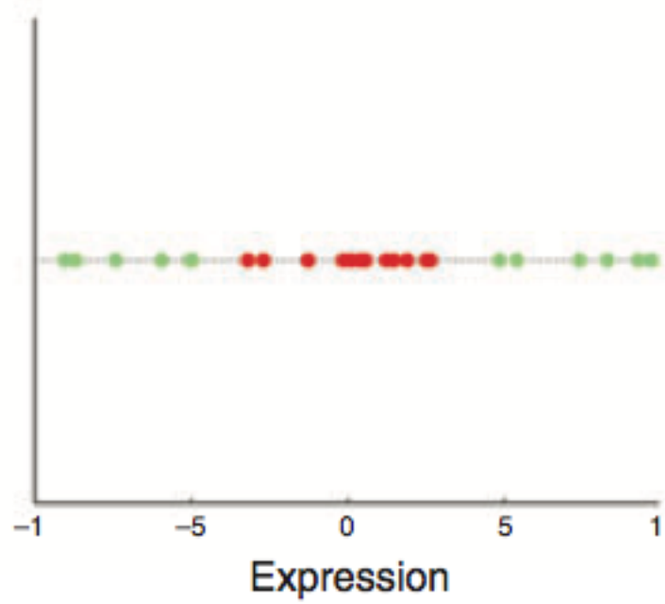
$$\sum_{i=1}^{n} w_i^2$$

  - Subject to

$$8\ i:\ y_i[w\ ¢\ x + b]\ ¸\ 1$$

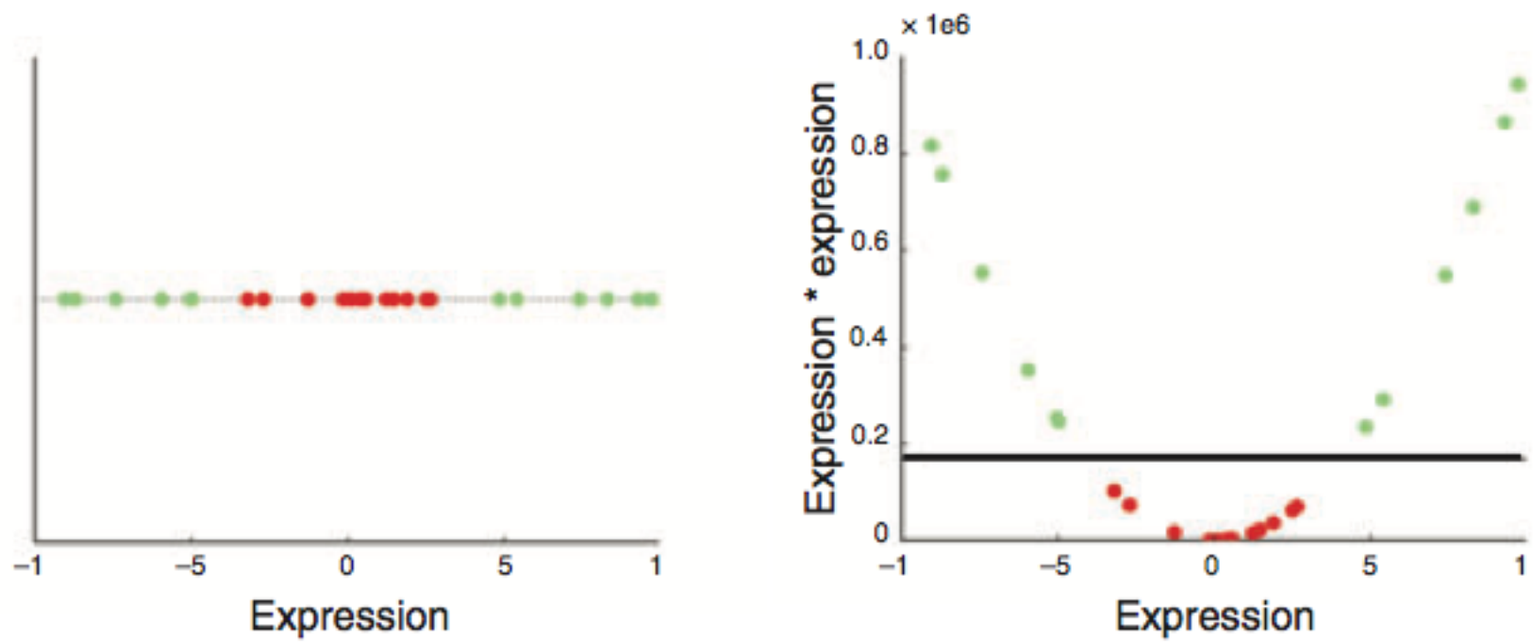
Support Vector Machine (2)

- Kernels: Any non-linear problem may be transformed into a linear problem if we select the right kernel

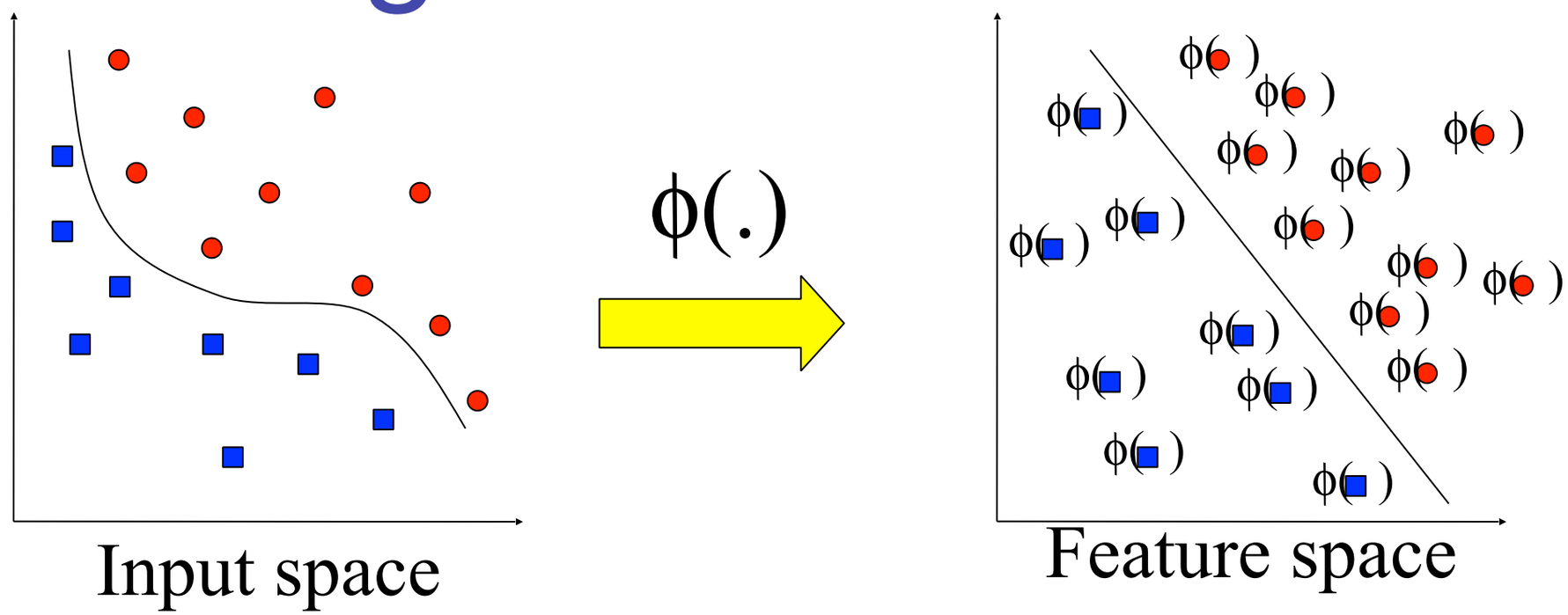


Support Vector Machine (2)

- Kernels: Any non-linear problem may be transformed into a linear problem if we select the right kernel



Transforming the Data



Note: feature space is of higher dimension than the input space in practice

- Computation in the feature space can be costly because it is high dimensional
  - The feature space is typically infinite-dimensional!
- The kernel trick comes to rescue

Kernel Functions

- In practical use of SVM, the user specifies the kernel function; the transformation $\phi(.)$ is not explicitly stated
- Given a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, the transformation $\phi(.)$ is given by its eigenfunctions (a concept in functional analysis)
  - Eigenfunctions can be difficult to construct explicitly
  - This is why people only specify the kernel function without worrying about the exact transformation
- Another view: kernel function, being an inner product, is really a similarity measure between the objects

Examples of Kernel Functions

- Polynomial kernel with degree *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Radial basis function kernel with width σ

$$K(\mathbf{x}, \mathbf{y}) = \exp(-||\mathbf{x} - \mathbf{y}||^2/(2\sigma^2))$$

  - Closely related to radial basis function neural networks
  - The feature space is infinite-dimensional
- Sigmoid with parameter κ and θ

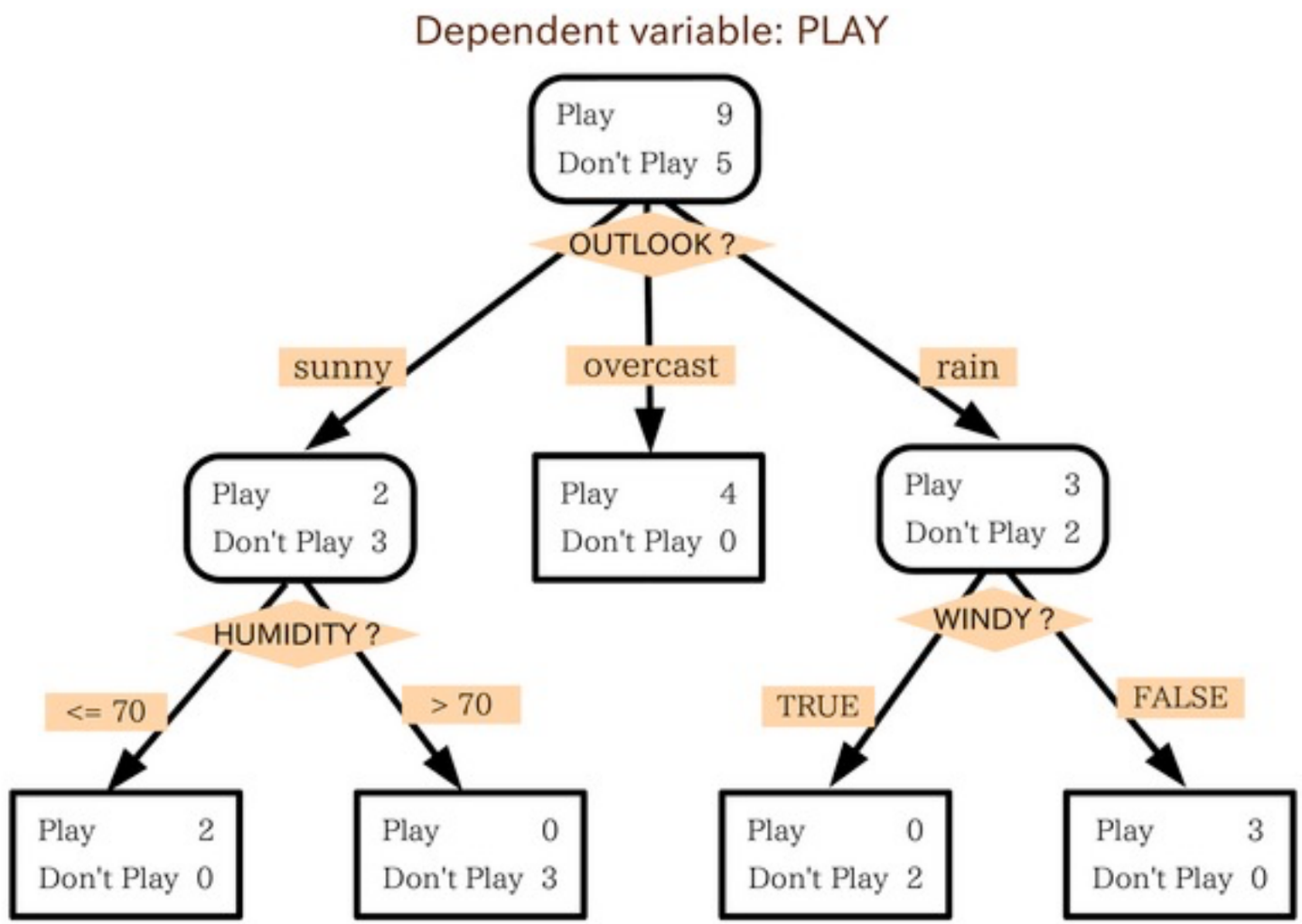
$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

  - It does not satisfy the Mercer condition on all κ and θ

Strengths and Weaknesses of SVM

- Strengths
  - Training is relatively easy
    - No local optimal, unlike in neural networks
  - It scales relatively well to high dimensional data
  - Tradeoff between classifier complexity and error can be controlled explicitly
  - Non-traditional data like strings and trees can be used as input to SVM, instead of feature vectors
- Weaknesses
  - Need to choose a “good” kernel function.

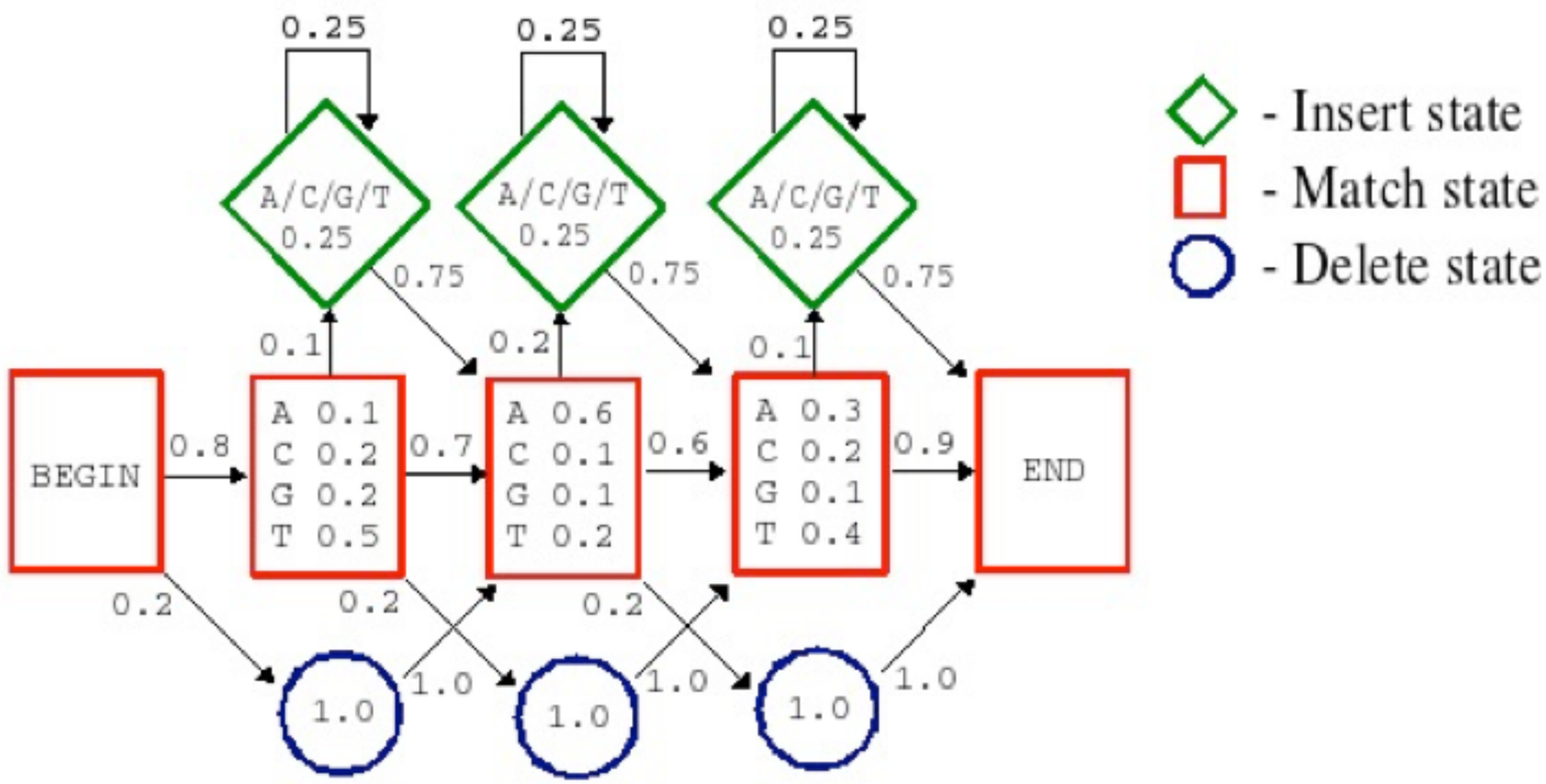
Random Forests



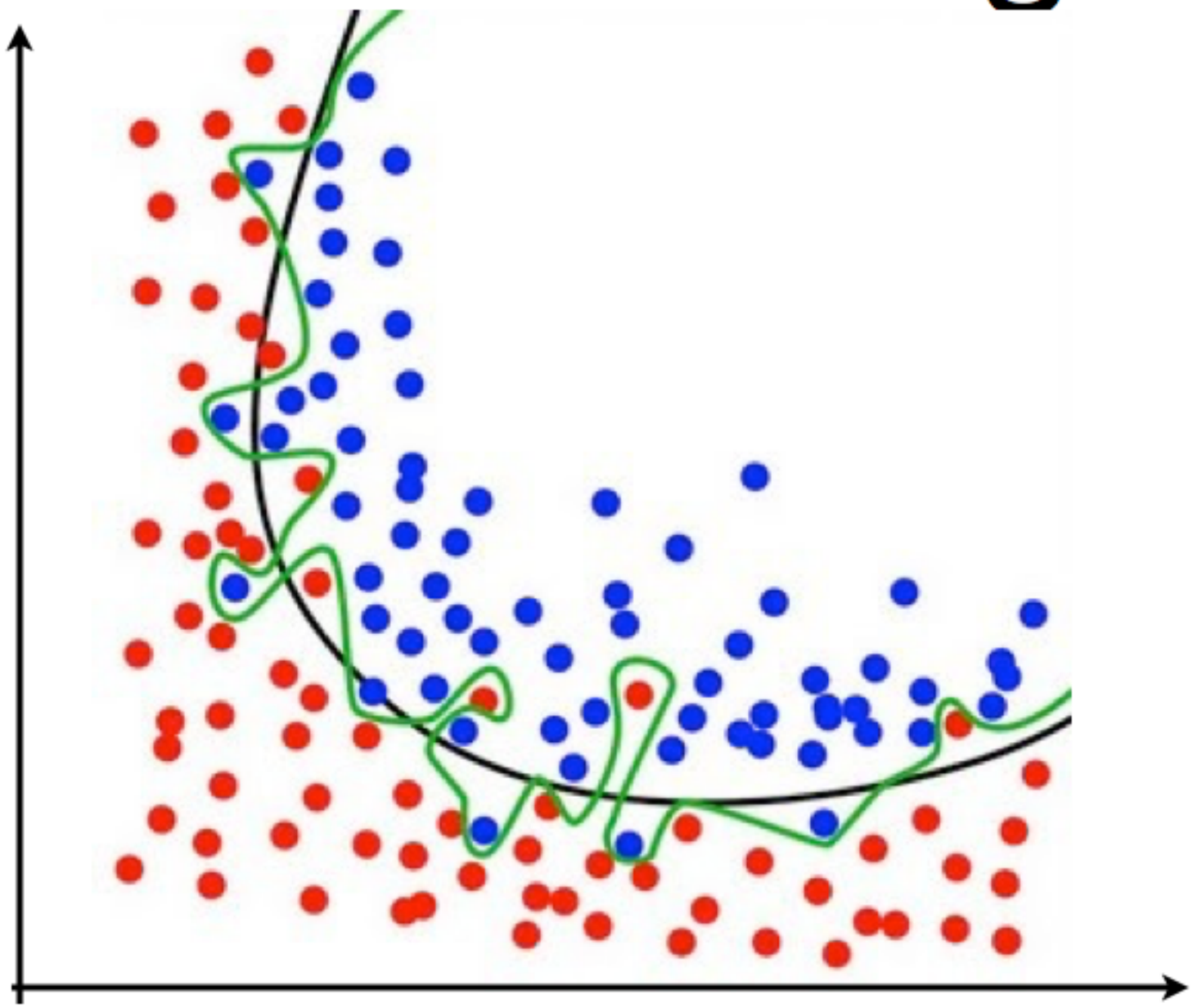
Genetic Algorithm

- Initialization
  - Randomly distribute weights/parameters
- Selection
  - Based on fitness as measured by a fitness function
- Reproduction
  - Recombination and/or mutation
- Termination
  - Minimal criterium / Fixed number of generation

Hidden Markov Models and Dynamic Bayesian Networks



Over-fitting



Validation strategies

- If we want to be able to detect over-fitting we need to train our method examples in a training set that is separate from the examples that we test our method with the test set. HOMOLOGY REDUCTION (sequence data)
- If we need to select hyper-parameters we need to yet another separate test set to find an optimal value.

Cross Validation

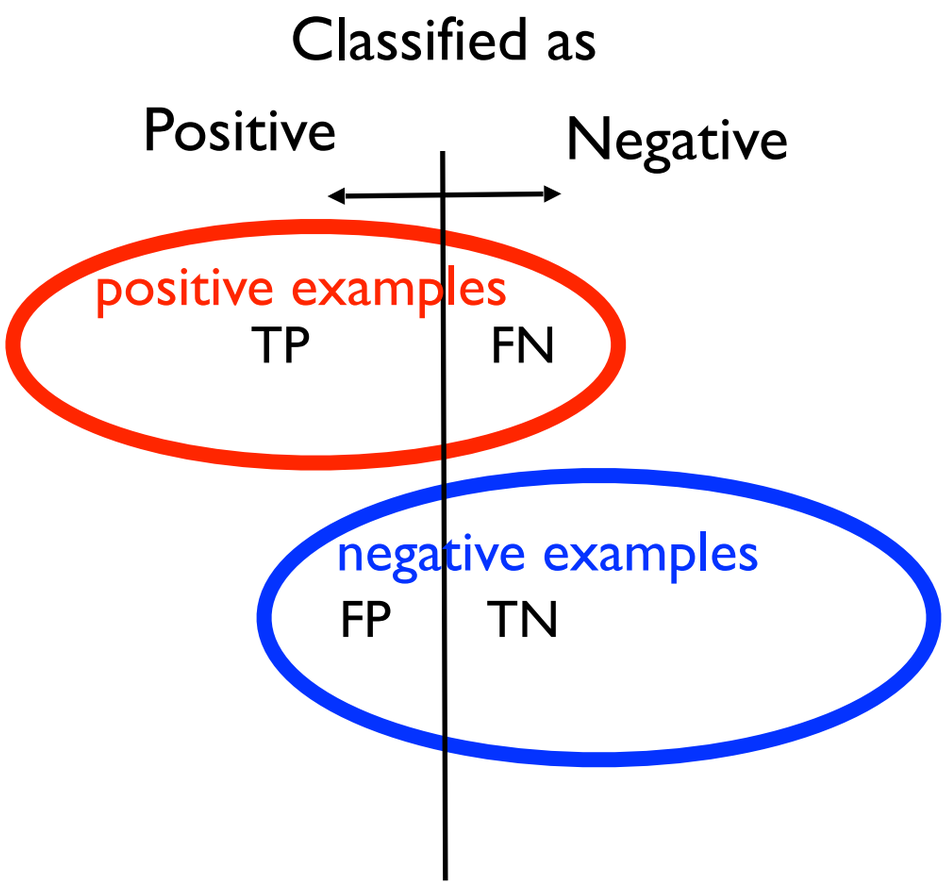
3-fold cross validation

lists of scores

score	type
7.5	+ Label
7.2	+ Label
6.9	+ Label
6.8	+ Label
6.7	- Label
6.5	+ Label
6.4	+ Label
6.4	+ Label
6.3	- Label
6.1	+ Label
6.0	- Label
5.9	+ Label
5.7	- Label
5.6	- Label
5.4	+ Label
5.3	- Label
5.2	- Label

threshold

Metrics

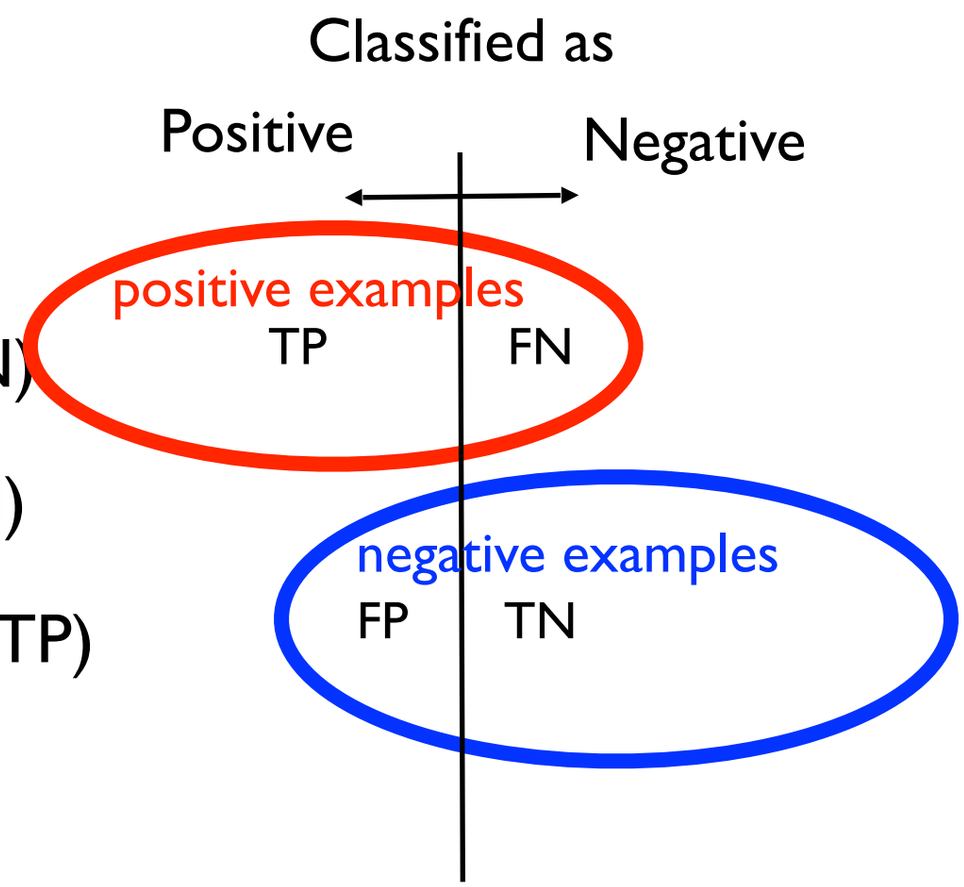


- TP = True positive = Correctly classified as positive example
- FP = False positive = Incorrectly classified as positive example
- FN = False negative = Incorrectly classified as negative example
- TN = True negative = Correctly classified as negative example

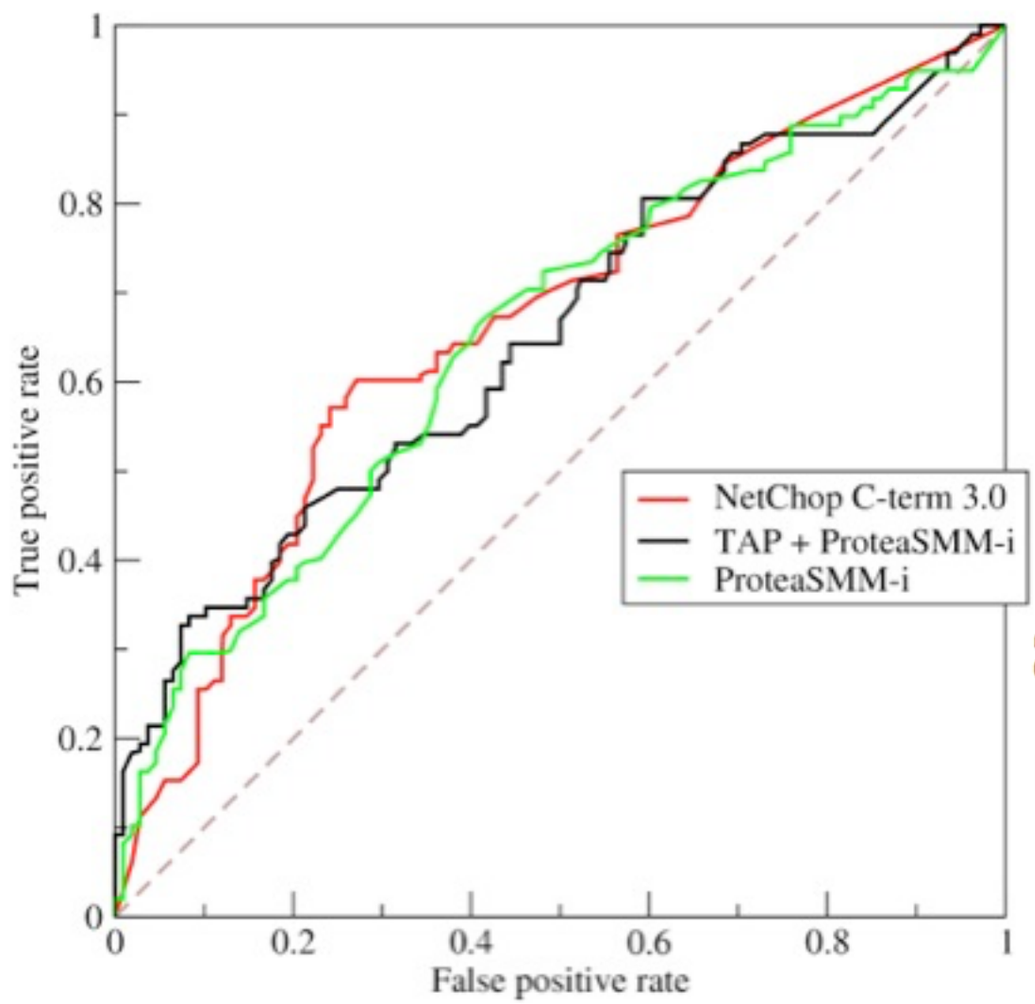
	Classified as positive	Classified as negative
Positive example	TP	FN
Negative example	FP	TN

Metrics

- precision=accuracy=TP/(TP+FP)
- recall=sensitivity=TP/(TP+FN)
- True Positive Rate=TPR=TP/(TP+FN)
- False Positive Rate=FPR=FP/(FP+TN)
- False Discovery Rate=FDR=FP/(FP+TP)
  - accuracy=1-FDR
- Matthews correlation coefficient=
  MCC=(TP*TN-FP*TN)/sqrt(TP+FP)(TP+FN)(TN+FP)(TN+FN)
  MCC is in the range +1 to -1.
  MCC of +1 represents a perfect prediction, 0 an average random prediction and −1 an inverse prediction.



Receiver operating characteristic (ROC) plot



ROC score =
area under the ROC curve

ROC_{50} score =
area under the ROC curve
up to 50 negative examples

Implementations

- General Toolboxes
  - Weka
  - R
  - Matlab
- SVM
  - SVMlight
  - linSVN
  - PyML
- Random Forests
  - C4.5
- DBNs
  - Graphical Models Toolkit
- HMM
  - HMMer
  - SAM