COSC 348: Computing for Bioinformatics

Lecture 20:
Microarray Data Analysis: classification

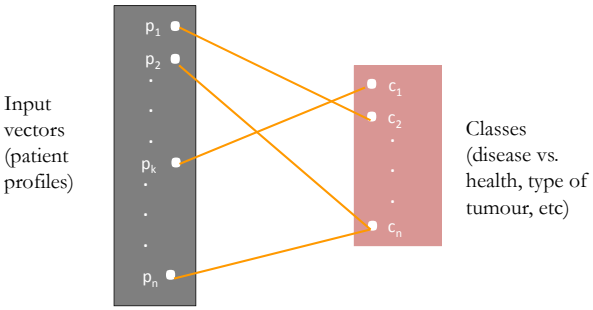
Lubica Benuskova

http://www.cs.otago.ac.nz/cosc348/

Steps of microarray analysis

- Gene Selection (after filtering and normalisation)
  - find genes, which would be the best predictors
- Clustering
  - K-means clustering
  - Hierarchical clustering
- Classification
  - K nearest neighbour
  - Support vector machine (SVM)
  - Artificial neural networks (ANN)
- Gene ontology:
  - record/compare/evaluate functions of genes

Classification: objects are assigned to classes



- Popular methods: K nearest neighbours, SVM and ANN.

Basic concepts for classification models

- *Training set* = a set of input vectors with known class labels
- *Test set* = a set of input vectors with *unknown* class labels
- *Stochastic training* = vectors are chosen randomly from the training set
- *Batch training* = all training vectors are presented to the model before the model parameters are updated
- *On-line protocol* = each pattern is presented once and model parameters are adjusted
- *Epoch* = a single presentation of all patterns in the training set.

Generalisation

- Generalisation:
  - to correctly classify data outside the training set, *e.g.* data in the test set.
  - *i.e.* the method must successfully classify (interpolate) to vectors it has not seen before.
- How can we test generalisation?
  - **Cross Validation (Leave one out):** the class of each patient in the available data set is predicted while the rest of the data is regarded as the training set.

Classification results

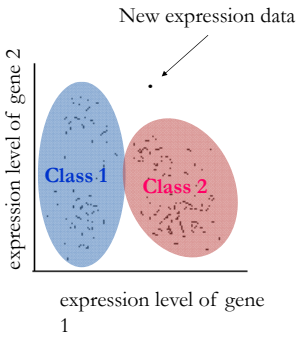
- **Percentage of correctly classified objects,**
- **Confusion matrix** is a table with two rows and two columns that reports the number of True Negatives, False Positives (type I error), False Negatives (type II error), and True Positives.

	Predicted Negative	Predicted Positive
Negative Cases	True Negatives	False Positives
Positive Cases	False Negatives	True Positives

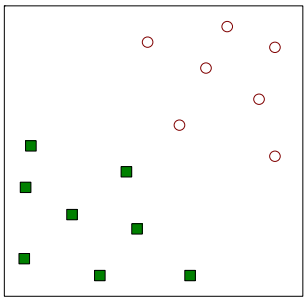
	Predicted Disease	Predicted Healthy
Disease Cases	True Disease	False Healthy
Healthy Cases	False Disease	True Healthy

Classification based on *K* nearest neighbours

- Does the new expression data belong to class 1 or 2?
- *K*-NN algorithm computes the distance of a test sample to each of the training set samples, each of which has an associated class label, and then predicts the class of the test sample to be that of the majority of the *K*-closest samples.
- Optimal *K* is chosen based on cross-validation.

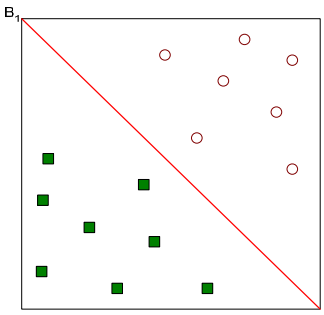


Linear classification



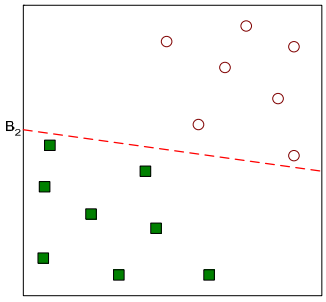
- Find a linear hyperplane (decision boundary) that will separate the data

Linear classification



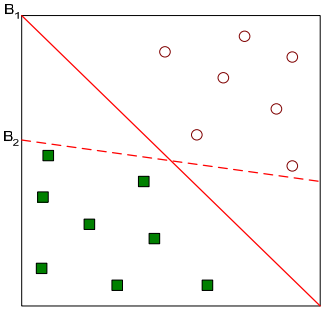
- One Possible Solution

Linear classification



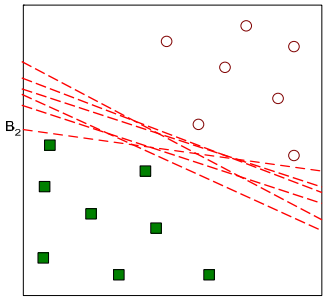
- Another possible solution

Linear classification



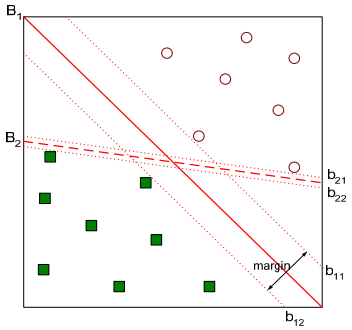
- Which one is better? B1 or B2?
- How do you define better?

Linear classification



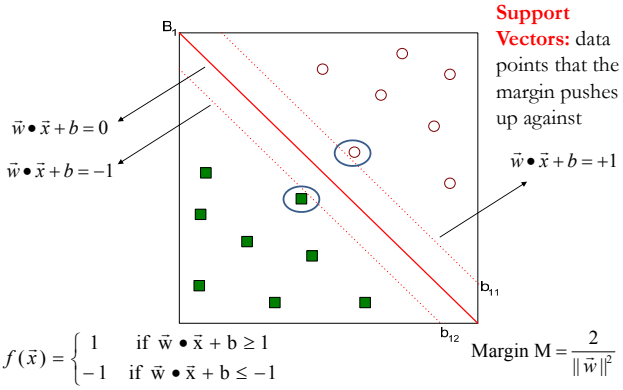
- There are many possible solutions – which one is the best?

Linear Support Vector Machine (SVM)



- The hyperplane that maximizes the margin B1 is better than B2

SVM: find support vectors to maximize margin



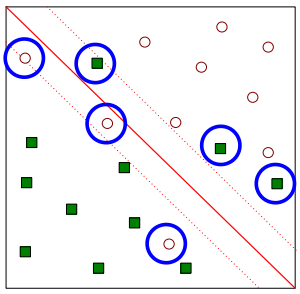
Support Vectors: data points that the margin pushes up against

$$f(\vec{x}) = \begin{cases} 1 & \text{if } \vec{w} \bullet \vec{x} + b \geq 1 \\ -1 & \text{if } \vec{w} \bullet \vec{x} + b \leq -1 \end{cases}$$

Margin $M = \frac{2}{\|\vec{w}\|^2}$

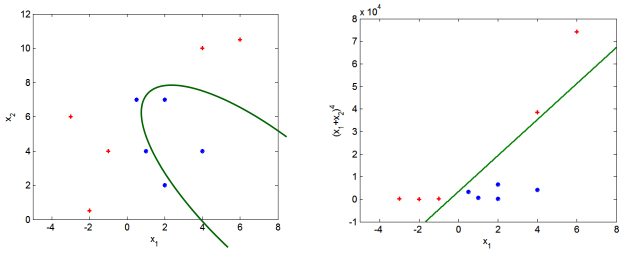
Nonlinear problem

- What if the problem is not linearly separable?



"Nonlinear" SVM

- Transform data into higher dimensional space
- Reduce the problem to a linear one

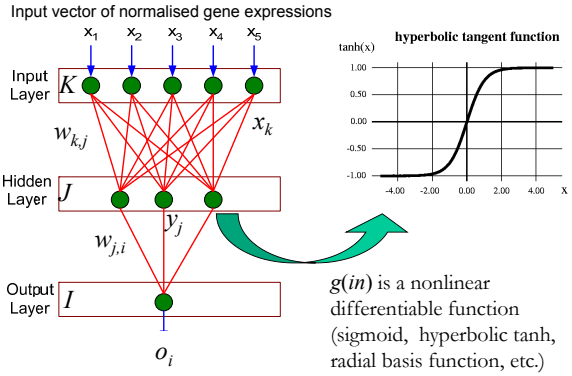


Original data | Transformed data

Properties of SVM

- The optimised margin function has a single minimum, this avoids the problem of local minima.
- The maximum margin requirements are appropriate conditions when the underlying statistical properties of the data are not known (which is often so).
- SVM works well when few observations are available (when the data space is very sparsely populated), which is often the case of medical problems.
- If there is a "high" nonlinearity in data, ANN may be more appropriate, as they are inherently nonlinear model.

MLP (Multilayer Perceptron)



$g(in)$ is a nonlinear differentiable function (sigmoid, hyperbolic tanh, radial basis function, etc.)

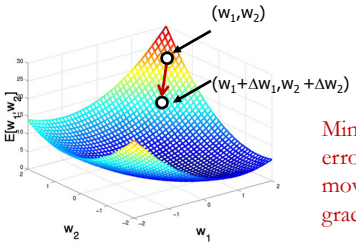
MLP representational power

- In principle any function can be approximated with arbitrarily small error by a two-layer feed-forward MLP.
- Sigmoid functions in the hidden layer act as a set of basis functions for composing more complex functions (like sine waves in Fourier analysis).
- The fit is adjusted thanks to a set of parameters of the MLP, which is the set of all weights.
- There are training algorithms which are able to find an optimal set of weights for each problem.

Training MLP based on input vectors

- Initialize the weights $\mathbf{w} = (w_0, w_1, \ldots, w_k)$ as small random numbers
- For each input vector calculate the MLP output and adjust the weights in such a way that the output of MLP is consistent with class labels of training examples c_i
  - Let the error function be: $E = \sum_{i=1}^{M} (c_i - o_i)^2$
  - The task is to find the weights w_i's that minimize the above error function:
    - e.g., the error backpropagation algorithm:

Weights optimization by gradient descent



Minimization of the error function E – moving against the gradient of E

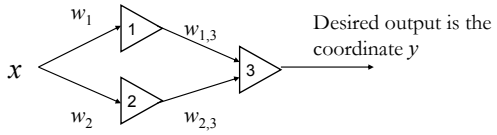
$$w_i \leftarrow w_i + \alpha \, \Delta w_i \qquad 0 < \alpha < 1$$

$$\Delta w_i = -\frac{\partial E}{\partial w_i}$$

α is the learning speed

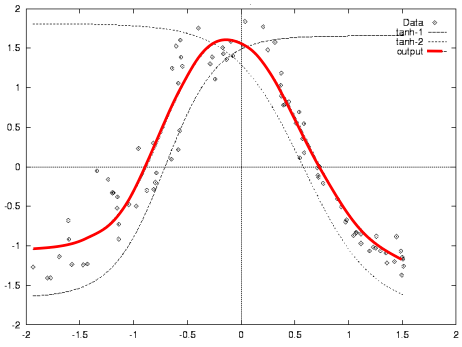
Example of MLP

- Inputs are x coordinates of points in 2D space.
- 2 hidden units (1 and 2) have hyperbolic tangent activation function.
- One output unit (No. 3) sums linearly the outputs of 2 hidden units minus an adjustable bias and predicts the y coordinate.



Approximation of the nonlinear function y = f(x)

- Weights are adjusted based on input x data. MLP is then able to predict y values also for unseen x values.



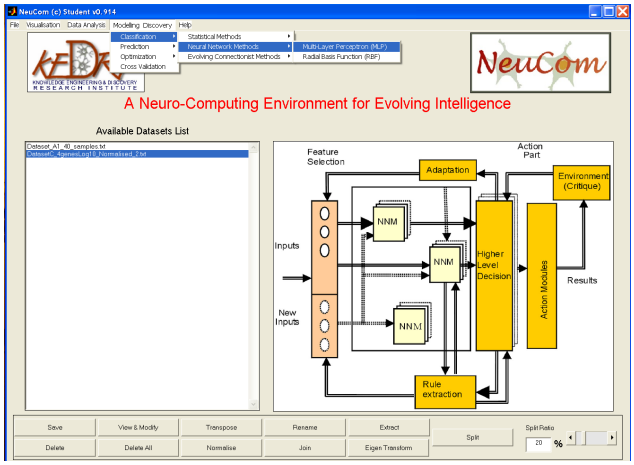
MLP: advantages versus disadvantages

Advantages

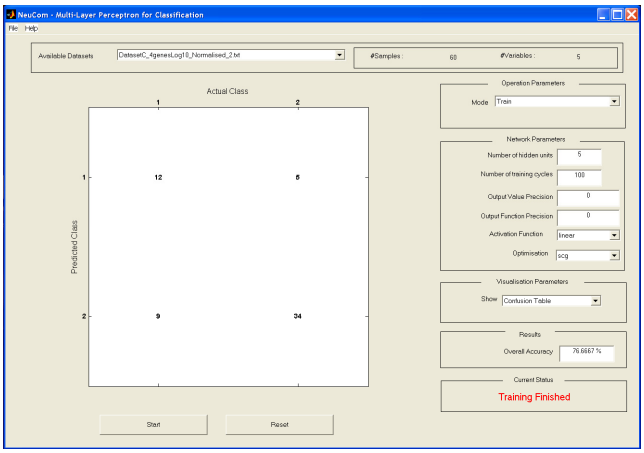
- Guaranteed to converge to local minimum of error;
- Even when the training data contains high noise;
- Even when training data are nonlinearly separable – in fact MLP can approximate arbitrary nonlinear function.

Disadvantage

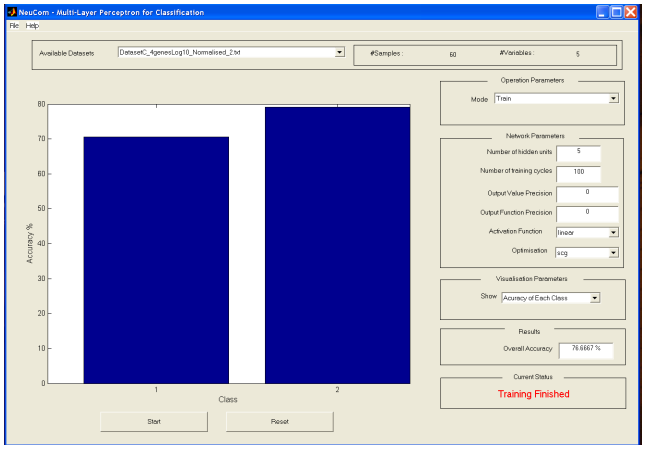
- Learning often gets stuck in a bad local minimum;
- No guarantee to converge to global minimum;
- Curse of dimensionality
  - For N=2 dimensional inputs, we need 100 = 10^2 training data
  - For N=3 dimensional inputs, 10^3 = 1000 training data are needed
  - In general, 10^N training data vectors are needed


NeuCom (c) Student v0.914
File Visualisation Data Analysis Modelling Discovery Help
Classification
Prediction
Optimization
Cross Validation
Statistical Methods
Neural Network Methods
Evolving Connectionist Methods
Multi-Layer Perceptron (MLP)
Radial Basis Function (RBF)
NeuCom
A Neuro-Computing Environment for Evolving Intelligence
Available Datasets List
Feature Selection
Adaptation
Action Part
Environment (Critique)
Inputs
NNM
NNM
NNM
Higher Level Decision
Action Modules
Results
New Inputs
Rule extraction
Save
View & Modify
Transpose
Rename
Extract
Delete
Delete All
Normalise
Join
Eigen Transform
Split
Split Ratio
20
%

25


NeuCom - Multi-Layer Perceptron for Classification
File Help
Available Datasets
DatasetC_4genesLog10_Normalised_2.txt
#Samples : 60
#Variables : 5
Actual Class
Predicted Class
1
2
12
6
9
34
Operation Parameters
Mode Train
Network Parameters
Number of hidden units 5
Number of training cycles 100
Output Value Precision 0
Output Function Precision 0
Activation Function linear
Optimisation scg
Visualisation Parameters
Show Confusion Table
Results
Overall Accuracy 76.6667 %
Current Status
Training Finished
Start
Reset

26


NeuCom - Multi-Layer Perceptron for Classification
File Help
Available Datasets
DatasetC_4genesLog10_Normalised_2.txt
#Samples : 60
#Variables : 5
Accuracy %
Class
Operation Parameters
Mode Train
Network Parameters
Number of hidden units 5
Number of training cycles 100
Output Value Precision 0
Output Function Precision 0
Activation Function linear
Optimisation scg
Visualisation Parameters
Show Accuracy of Each Class
Results
Overall Accuracy 76.6667 %
Current Status
Training Finished
Start
Reset

27