Computing Methods for Experimental Physics and Data Analysis

Data Analysis in Medical Physics

Lecture 7: Analysis of image features with predictive models based on Machine-Learning: classification, regression and unsupervised models

Alessandra Retico
alessandra.retico@pi.infn.it
INFN - Pisa

Machine Learning

- **Supervised learning:**
  - Predicting values, **known** data labels
  - The machine uses the label information to guess the right answer on new data

Regression	**Classification**
Estimate continuous values (real-valued output)	Identify a unique class (Boolean, discrete values, categories)

- **Unsupervised learning:**
  - Search for structure in data; **unknown** data labels
  - The machine find useful information hidden in data

Cluster Analysis	**Dimensionality reduction**
Group data points into sets	Select relevant variables

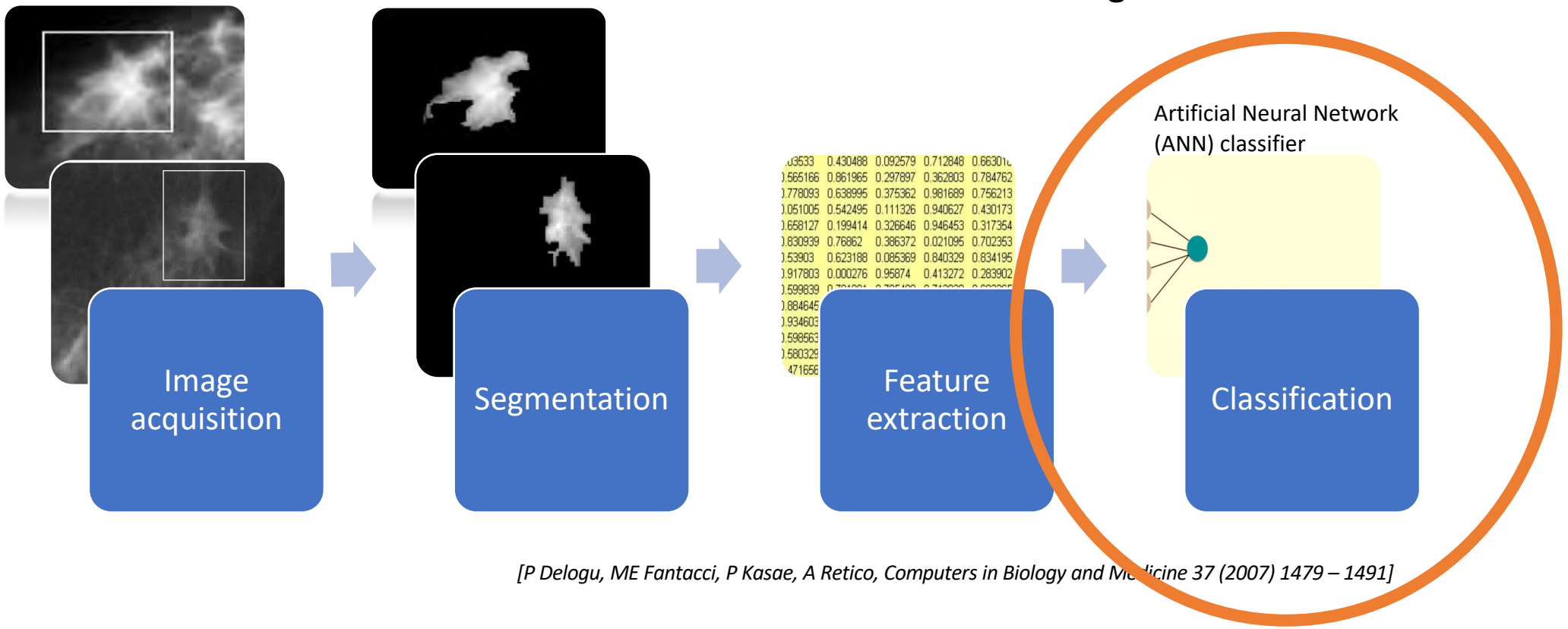
Outline

- Explore the use of predictive models in medical data analysis
  - Supervised classification (categorization)
  - Regression models
  - Unsupervised learning (clustering data)

- Quantification of model performance
  - Metrics/Figures of merit: Sensitivity, Specificity, AUC , Precision, Recall, F1
- Estimate of model robustness
  - Cross validation methods: k-fold cross validation, leave-one out cross validation

Hand-crafted feature + Machine Learning classification

Automated classification (benign/malignant) of mass lesions in mammograms



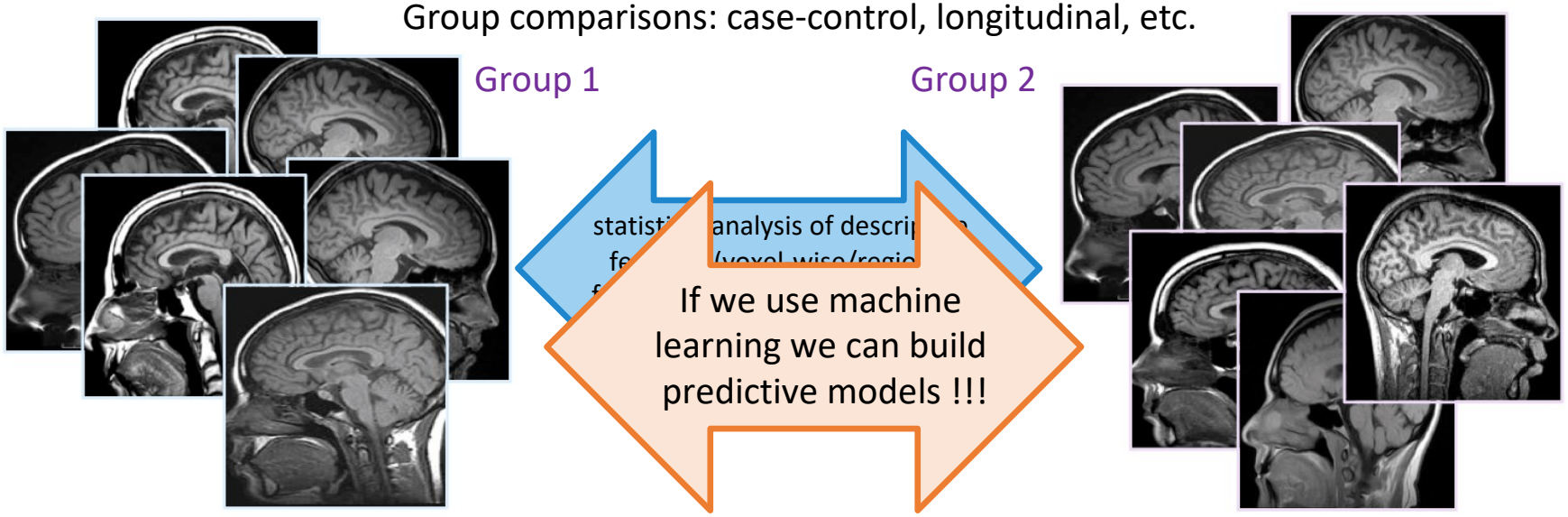
[P Delogu, ME Fantacci, P Kasae, A Retico, Computers in Biology and Medicine 37 (2007) 1479 – 1491]

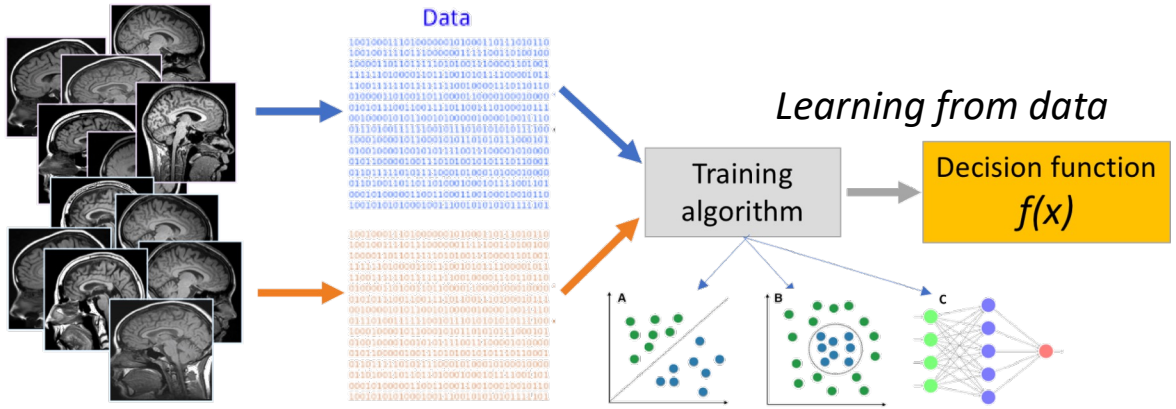
Machine learning in Medical Imaging

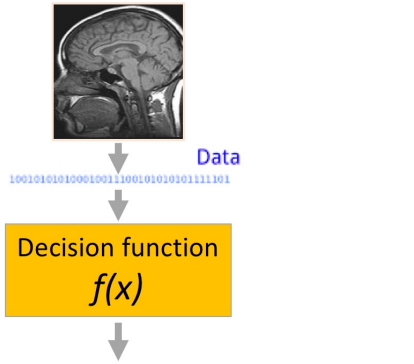
- Machine learning approaches are gaining popularity in the Medical Imaging community
- In addition to group characterization, they allow categorization of individual's previously unseen data: predictive diagnosis.
- Interesting studies have already been carried out in tumor segmentation and classification, in the study of neurological and psychiatric disorders (Alzheimer's disease, Parkinson's disease, autism spectrum disorders, schizophrenia, bipolar disorder,…), etc.
- International research consortia are collecting more and more data sample, including, MRI, PET, RX, CT, US data, to enable the training of machine-learning algorithms

Between-group comparisons

Group comparisons: case-control, longitudinal, etc.

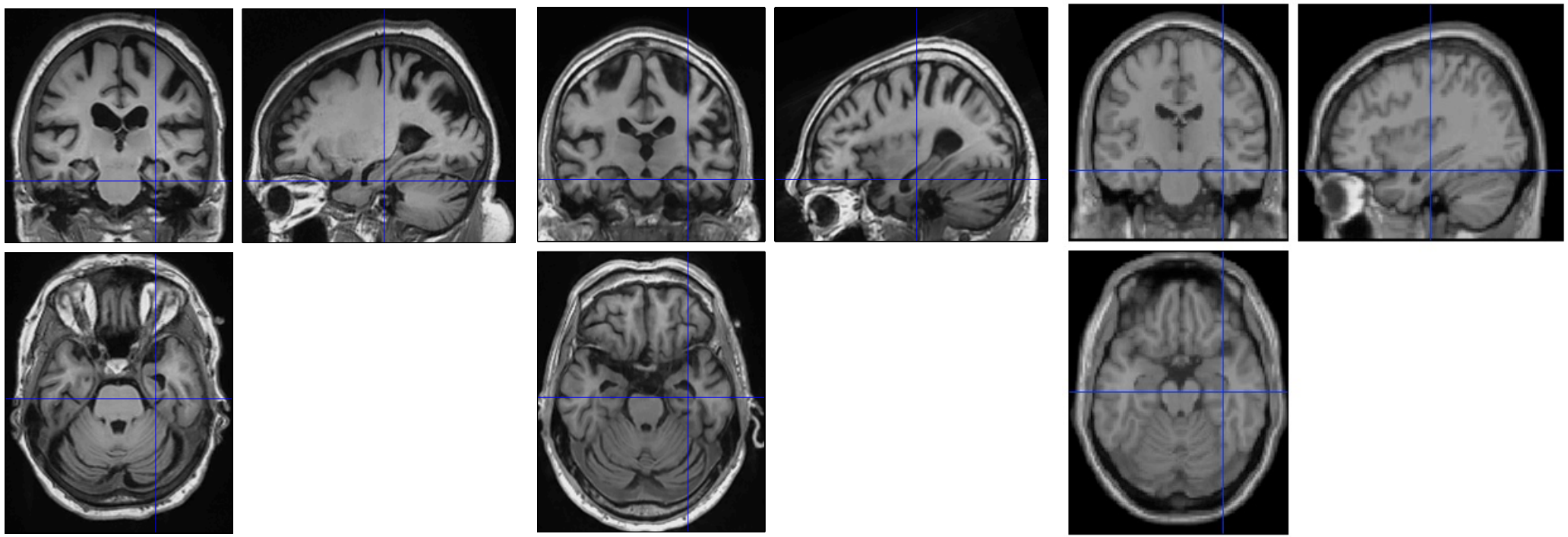






Class prediction on previously unseen example

Studying brain atrophy

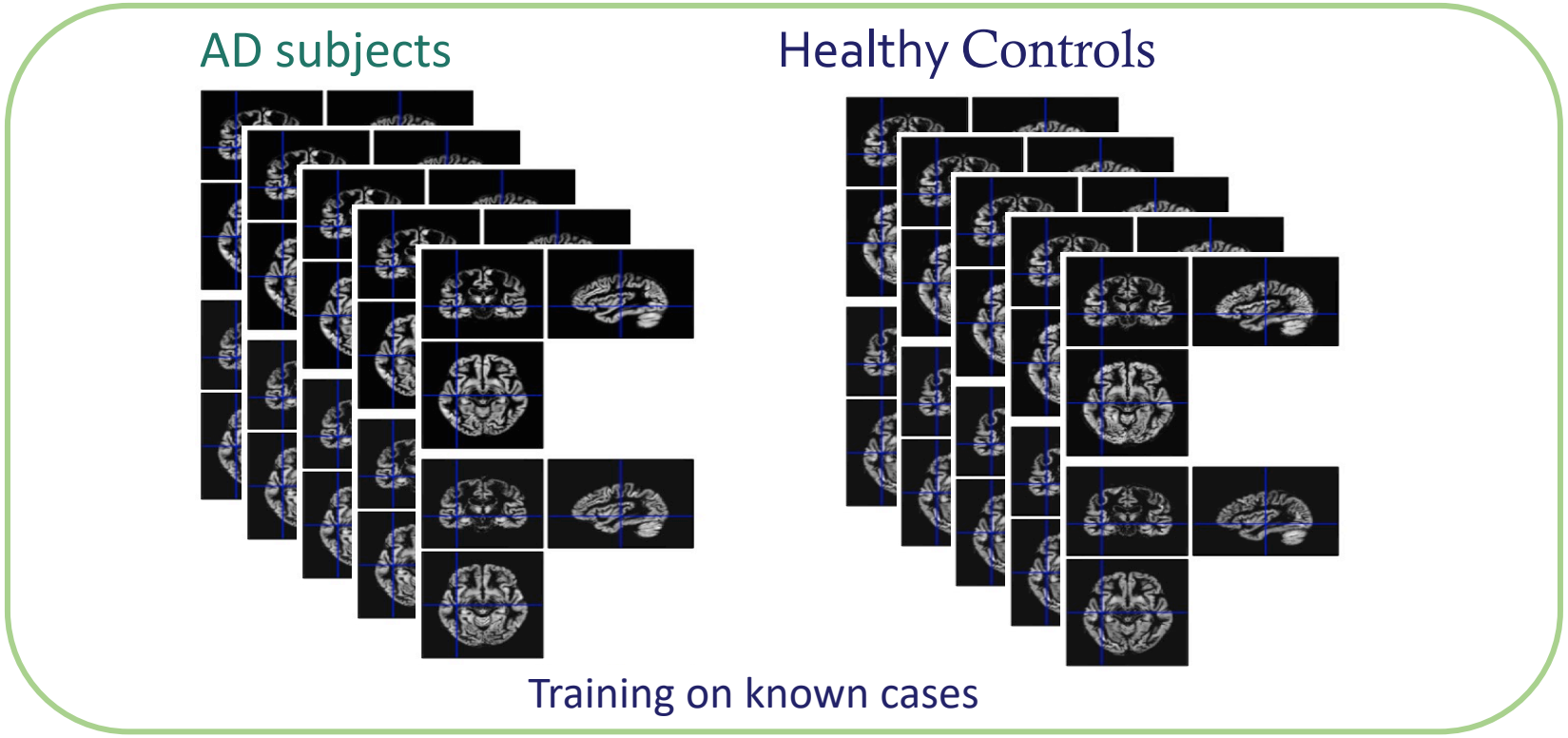


AD

CTRL

Healthy younger subject

Example of use of machine learning

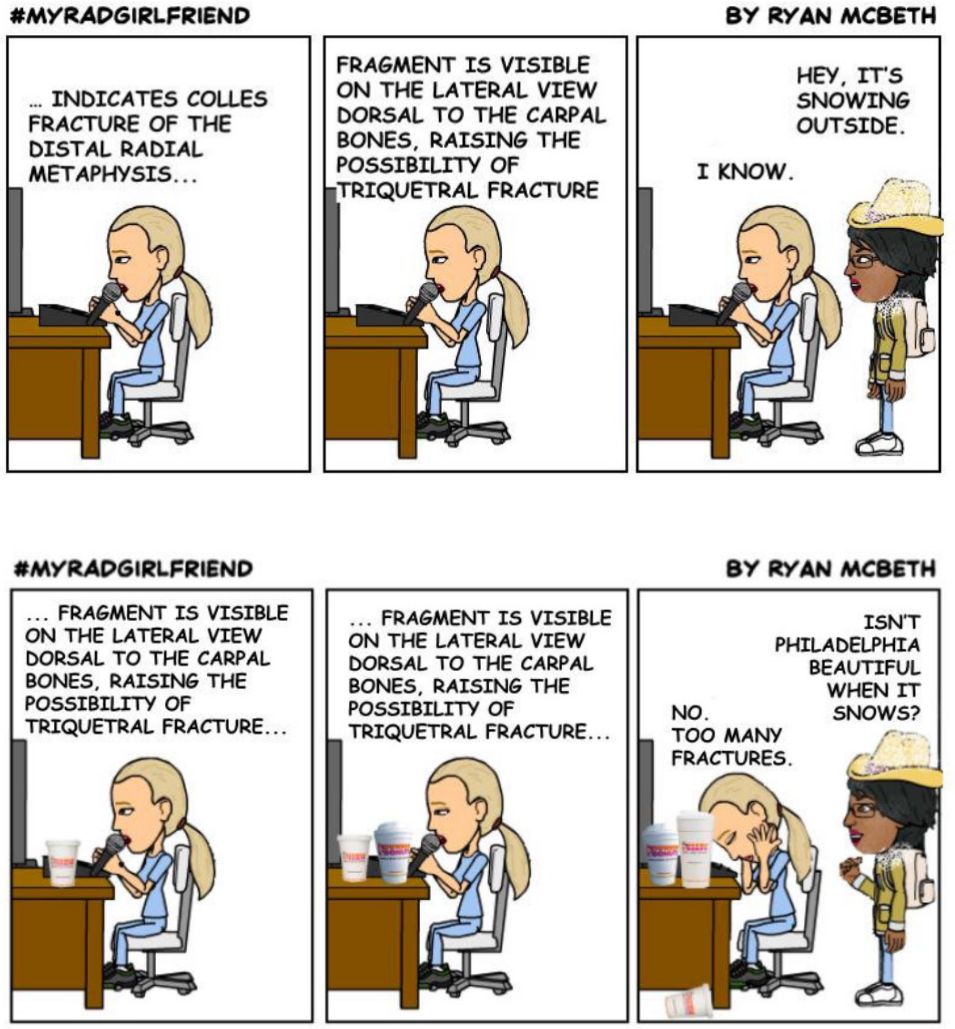


AD: subjects with Alzheimer's Disease

MCI: subjects affected by Mild Cognitive Impairment

Why using machine-learning techniques?

- Some tasks cannot be defined well, except by examples.
- Relationships and correlations can be hidden within large amounts of data.
  - Machine Learning/Data Mining may be able to find these relationships.
- The amount of knowledge available about certain tasks might be too large for explicit encoding by humans (e.g., medical diagnostic).
  - ***Learning from examples***



Ingredients for classification problems

Training a classifier

- Train, test and validation sets
- Choose the model

Classifier performance evaluation

- Metrics/Figures of merit
  - Sensitivity, Specificity and Receiver Operative Characteristic curve (ROC)
  - Area under the ROC curve (AUC)
- Cross validation procedure

Dimensionality problems

- Feature reduction

Train, Test, Validation sets

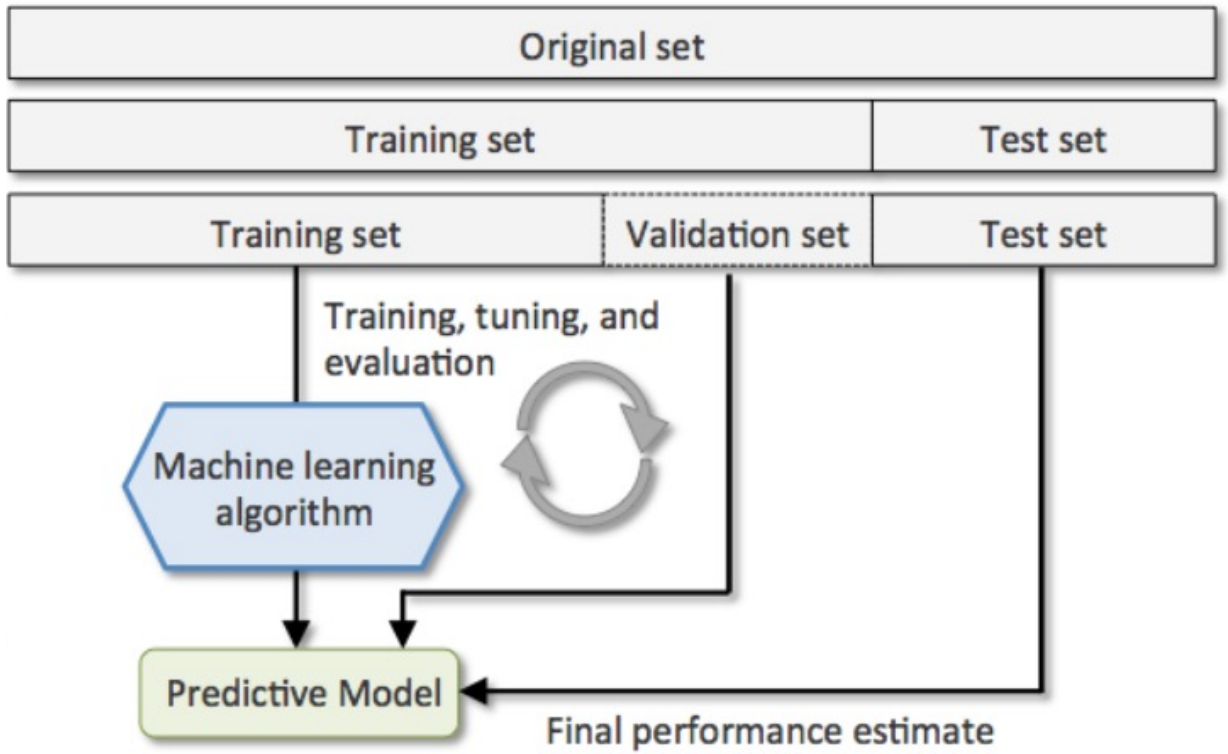
The data sample should be partitioned into subsets for training, validation and testing the classification algorithms.

Training: allowing the algorithm to set weights and "learn" how to classify data, using the training set

- Training set: a subset of items, with known classification
- This produces classifier, i.e. a mapping from features to class.
- Validation set: a subset of items, with known classification, that can be used to optimize the system performance

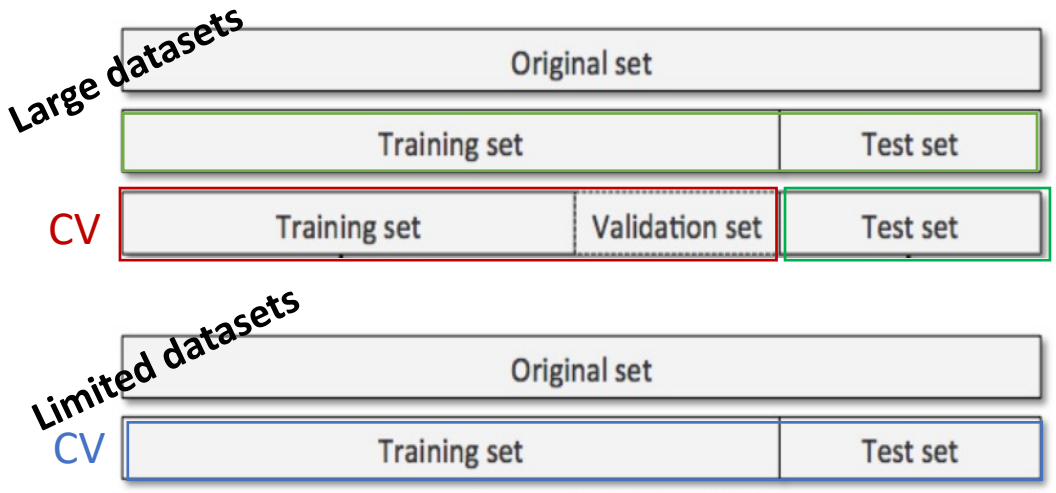
Classification: using these training-set weights to classify data of previously unknown category

- Independent validation set (test set): a subset of items, with known classification, where to evaluate the classifier performance



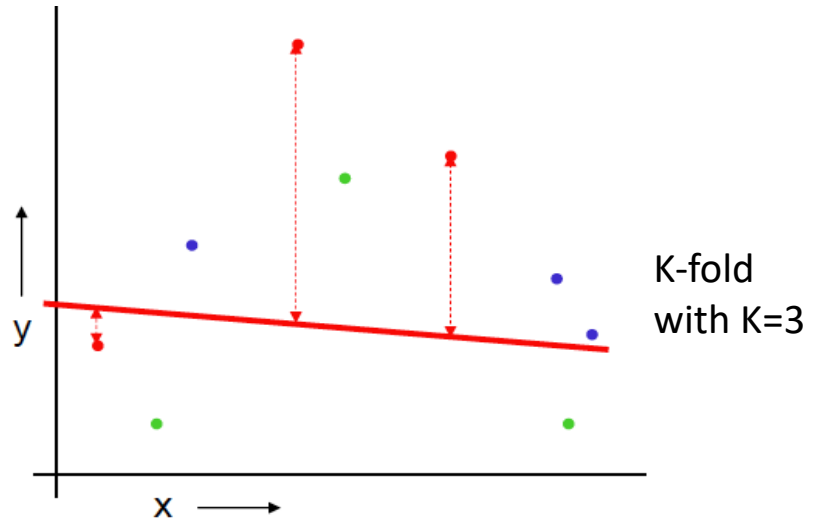
Cross validation procedures

Cross-validation: data resampling procedures are used to evaluate machine-learning models on a limited data sample.



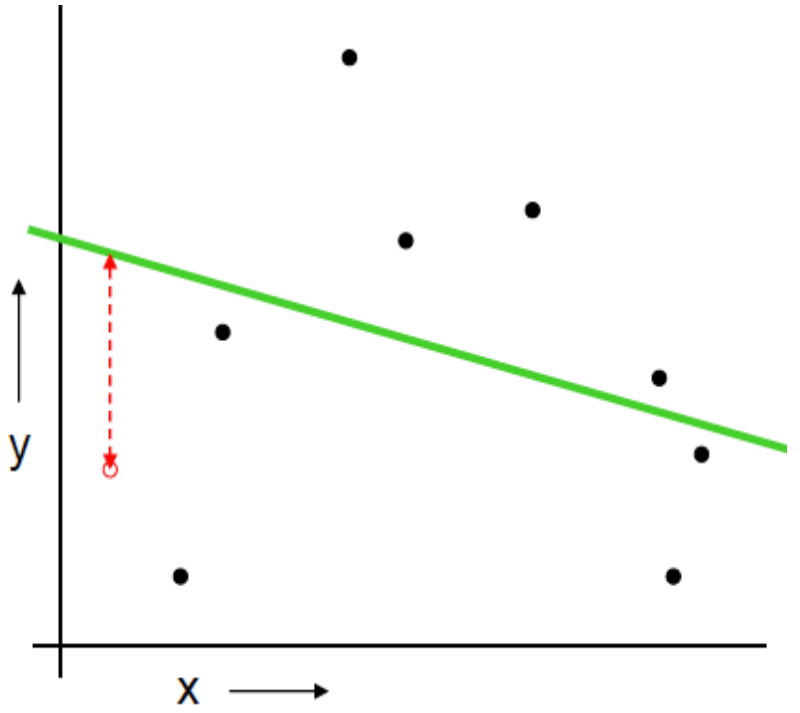
K-fold cross-validation:

- Original sample partitioned into K subsamples. Of K subsamples, one is retained as validation data while remaining (K − 1) subsamples are used as training data.
- Cross-validation process is repeated K times (the folds), with each of the K subsamples used exactly once as the validation data.
- The K results from the folds then can be averaged (or otherwise combined) to produce a single estimation.



Cross validation procedures

- Leave-one-out cross-validation (LOOCV):
  - It is a k fold CV with k=N, where N is the number of subjects.
  - Uses a single observation from original sample as validation data, and the remaining observations as training data.
  - This is repeated such that each observation in the sample is used once as the validation data.
  - The results obtained on the single observations should be combined to produce a the model estimation.
  - We cannot assign an error to the performance we obtained.



Figures of merit to quantify classification performance

Sensitivity = True Positive Rate (TPR)

Specificity =1 - False Positive Rate (FPR)

$$TPR = \frac{TP}{TP + FN} \quad \text{(true positive rate)}$$

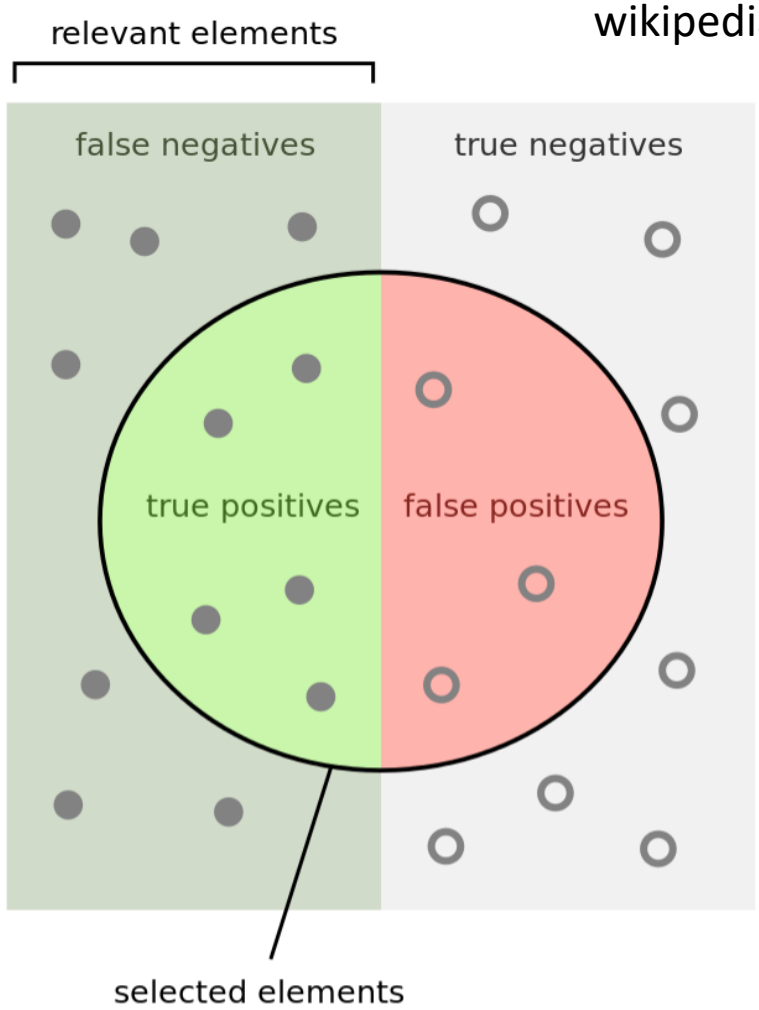
$$FPR = \frac{FP}{FP + TN} \quad \text{(false positive rate)}$$

Accuracy

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

CONFUSION MATRIX		Predicted class	
		YES	NO
Actual class	YES	True Positive (TP)	False Negative (FN)
	NO	False Positive (FP)	True Negative (TN)

wikipedia.org



(elements classified as relevant by the model)

Figures of merit to quantify classification performance

Precision and recall

How many selected items are relevant?

Precision = 

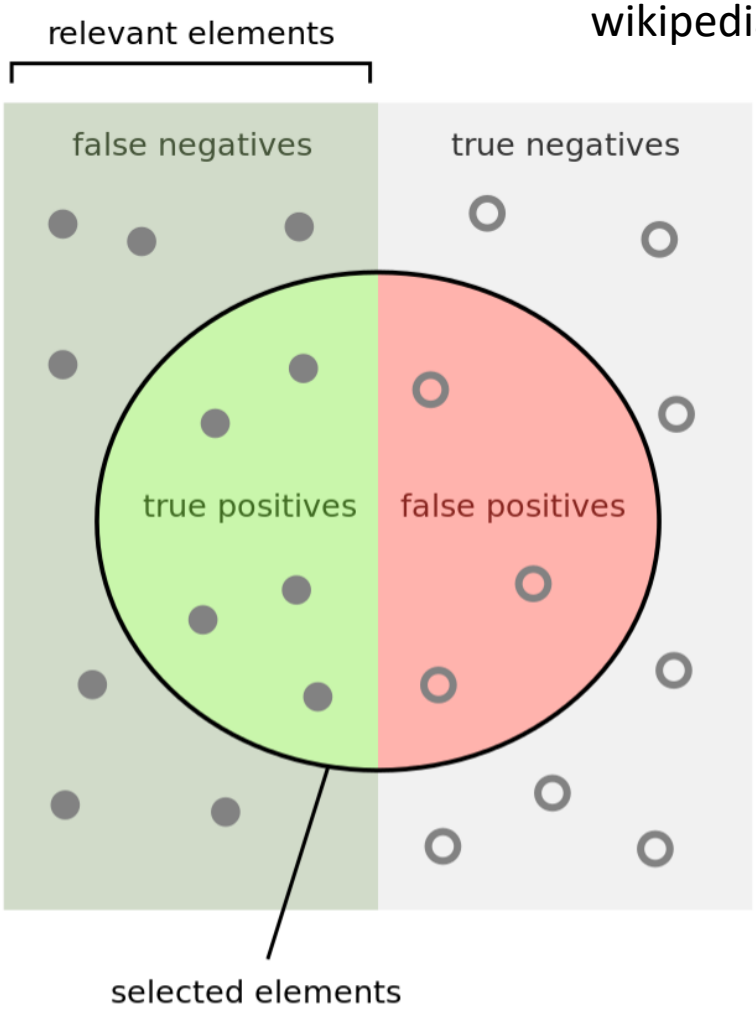
or Positive Predictive Value (PPV)

How many relevant items are selected?

Recall = 

or True Positive Rate (TPR), or Sensitivity

wikipedia.org



(elements classified as relevant by the model)

F_1 metric

The F_1 score is the harmonic mean of the precision and recall

$$F_1 = \left(\frac{2}{\mathrm{recall}^{-1} + \mathrm{precision}^{-1}} \right) = 2 \cdot \frac{\mathrm{precision} \cdot \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}}.$$

Figures of merit to quantify classification performance

Area under the Receiver Operating Characteristic (ROC) curve

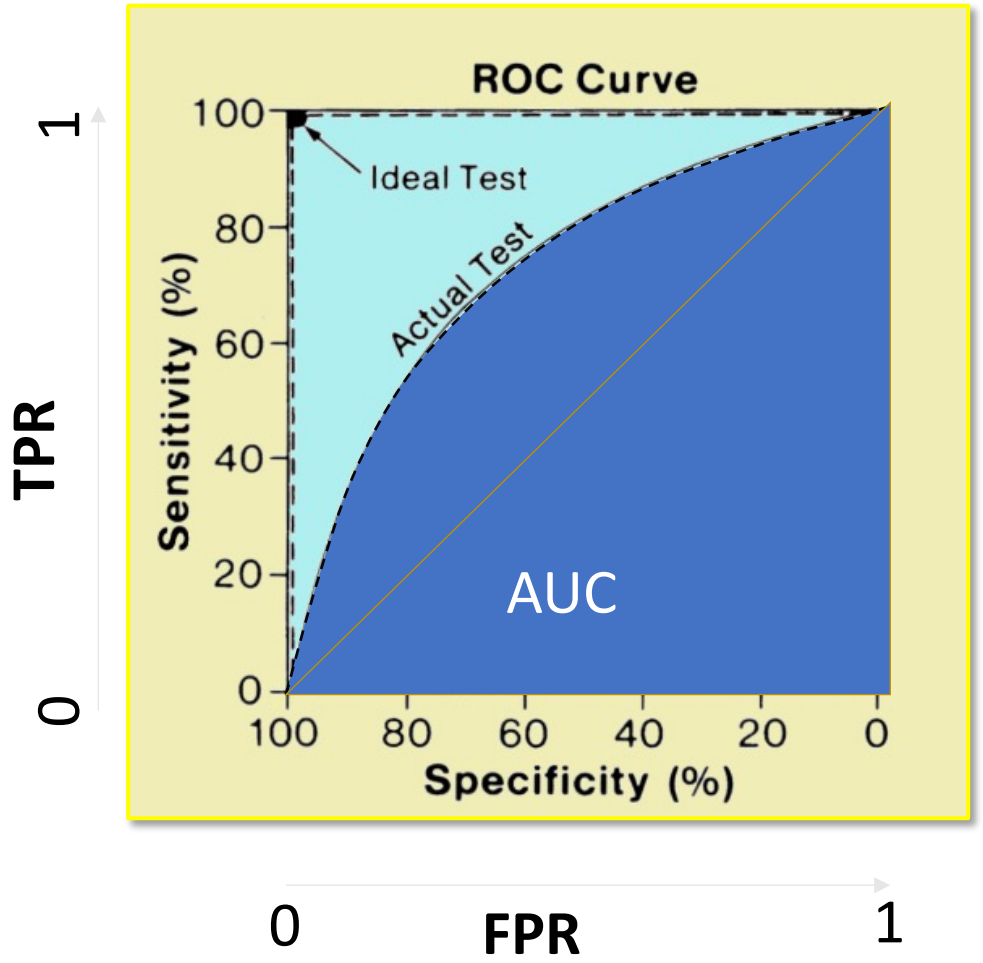
For each value of the decisional threshold (operative point) on the classifier output (generally in the [0,1] range), sensitivity and specificity values can be computed to obtain:

- the Receiver Operating Characteristic (ROC) curve

The classifier performance can thus expressed in terms of a single number:

Area Under the ROC Curve (AUC)

It is very useful to compare classifiers working at different operative points.



Area under the ROC curve (AUC)

- To compare classifiers we may want to reduce the ROC performance to a single scalar value representing expected performance

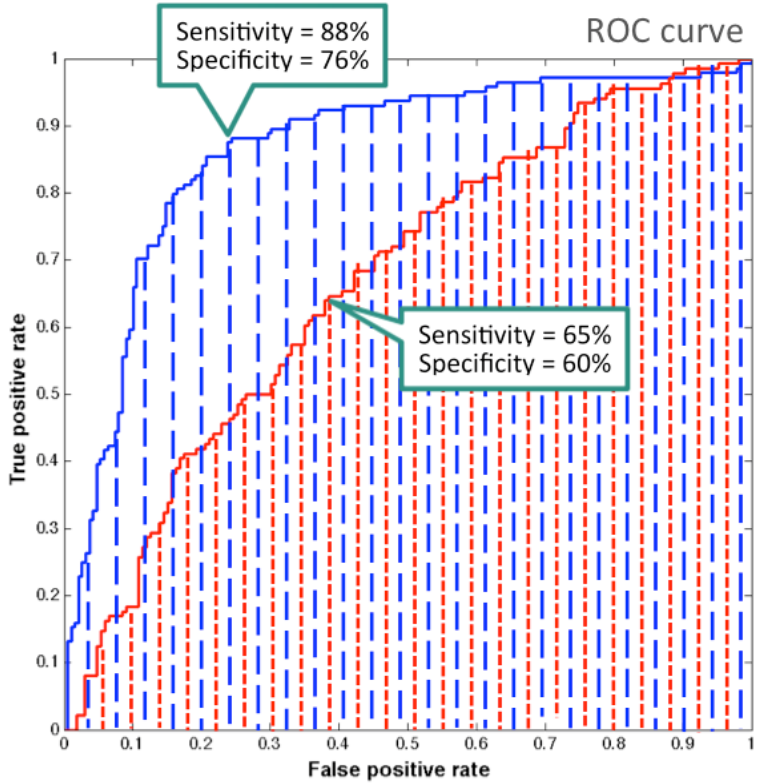
  → Calculate the AUC

- Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1

- However, because random guessing produces the diagonal line between (0, 0) and (1, 1), which has an area of 0.5, *no realistic classifier should have an AUC less than 0.5*

- An ideal classifier has an area of 1

- **Important statistical property:** AUC is equivalent to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance



Comparing two ROC curves:

- The graph represents the areas under two ROC curves, A and B. Classifier B has greater area and therefore better average performance

Result reporting

Mind out:

- the test set should representative of the whole sample (e.g. with respect to demographic characteristics)

For large cohorts, you should compute:

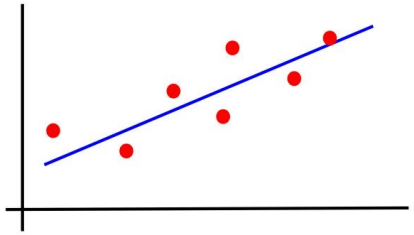
- Sensitivity & specificity / accuracy / AUC on the test set (independent validation)

For small cohorts, you should compute:

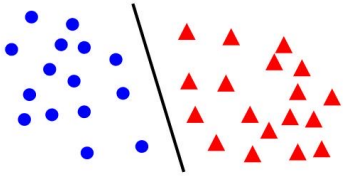
- Average sensitivity & specificity / accuracy / AUC and standard deviation over N repetitions of the k-fold cross validation on the test set

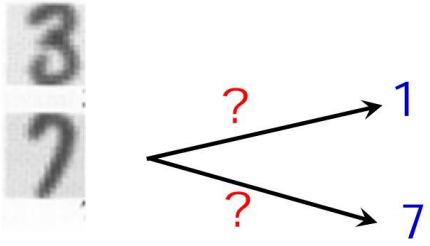
Three canonical learning problems

1. Regression - supervised
   - estimate parameters, e.g. of weight vs height

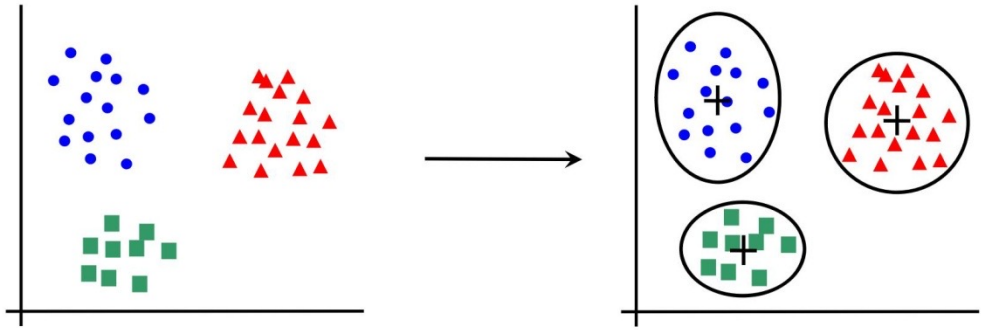


2. Classification - supervised
   - estimate class, e.g. handwritten digit classification

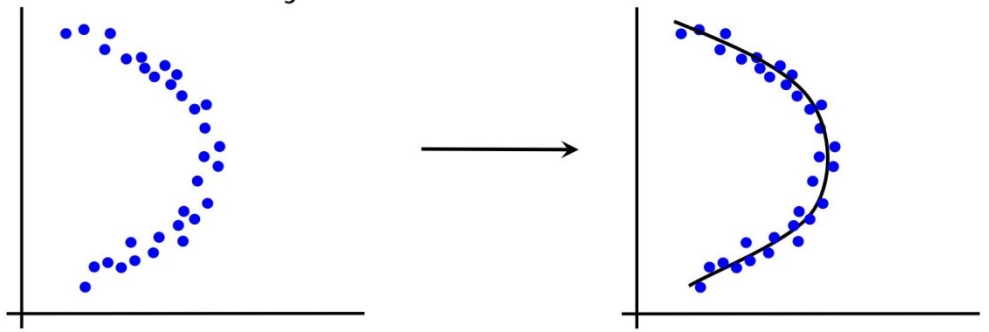




3. Unsupervised learning – model the data
   - clustering



   - dimensionality reduction



Labeled data for supervised classification/regression

In supervised learning an algorithm is employed to learn the mapping function

$$y_i = f(x_i)$$

- In the machine learning framework, each image $\mathbf{x}_i \in \mathbf{R}^n$ is considered as a point in a n-dimensional space (n is the number of voxels/features in the image).

Binary classification

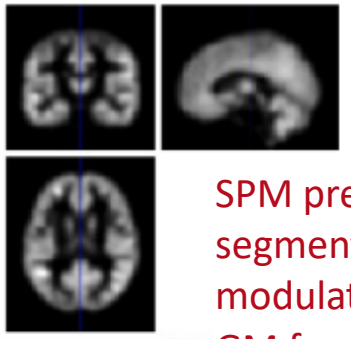
- In a two-class classification (e.g. patients vs. controls) the i-th image can be labelled with y_i:

$$y_i \in \{-1, 1\} \qquad \text{where } i = 1, \ldots, I.$$

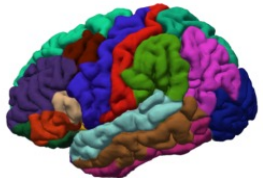
Regression

- In a regression to estimate continuous parameters (e.g. age of subjects) the i-th image can be labeled with $y_i \in \{age_{min}, age_{max}\}$

FEATURES



SPM preprocessed data: segmented, normalized, modulated and smoothed GM for each subject



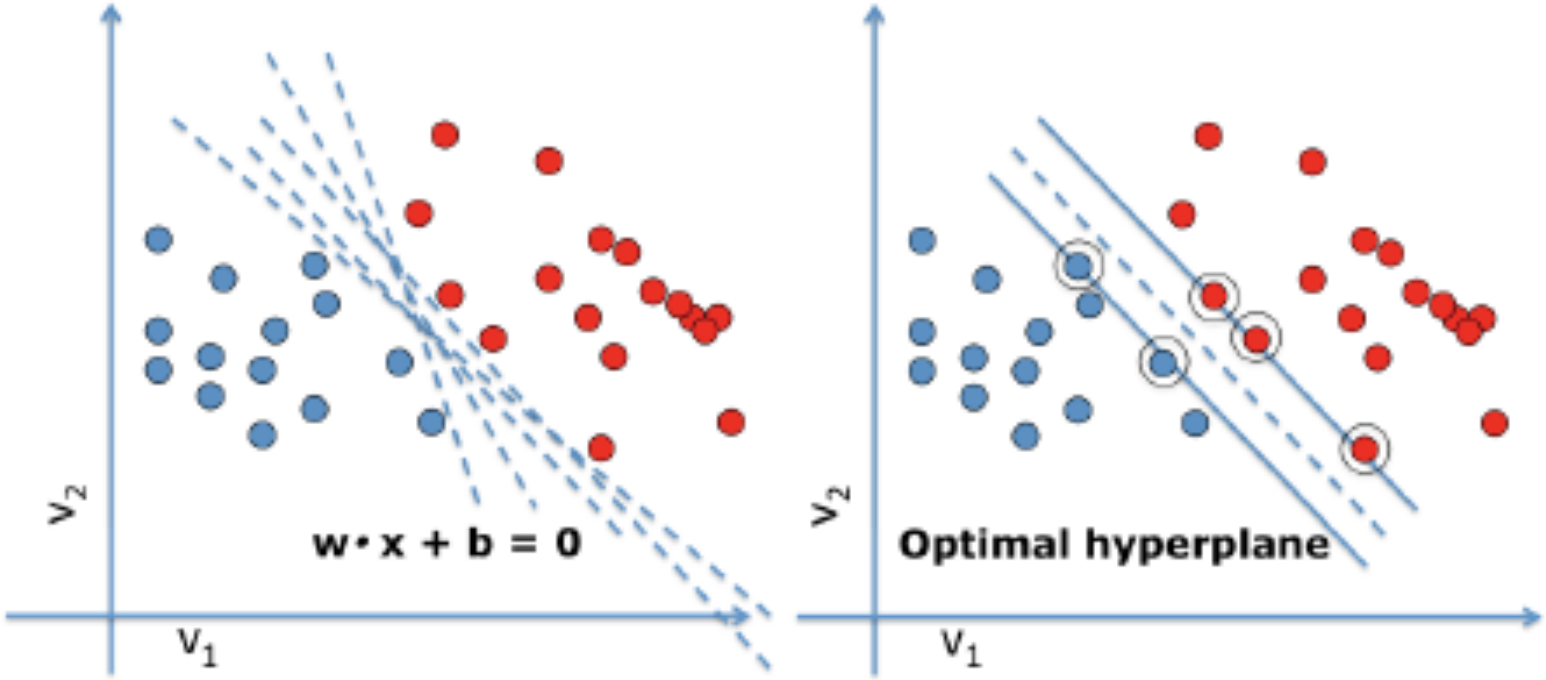
Freesurfer features: volume and thickness of brain parcels for each subject

http://freesurfer.net

Support Vector Machines (SVM)

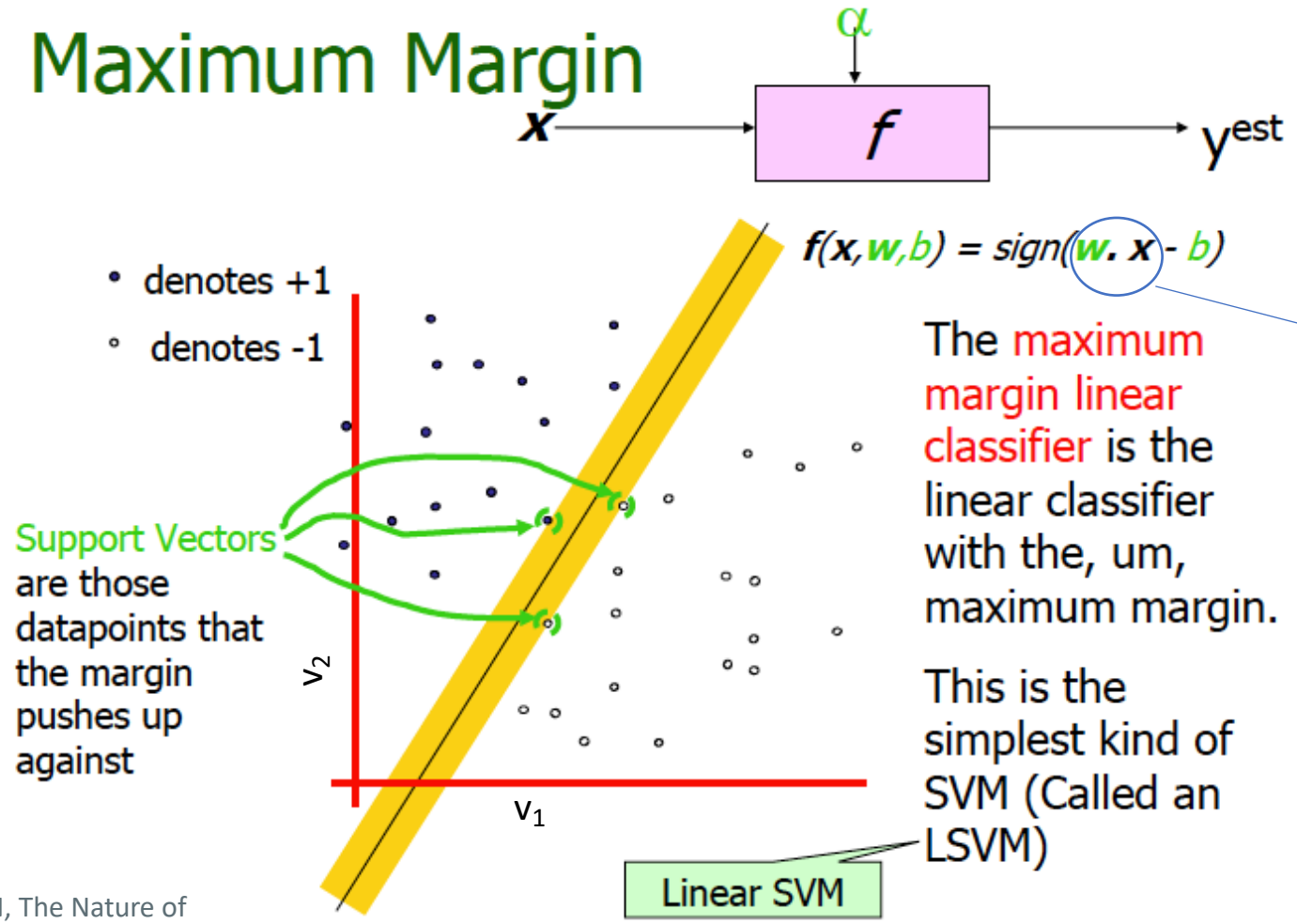
[Vapnik VN, The Nature of Statistical Learning Theory. New York: Springer (1995)]

- The SVM linear classification method aims to estimate the separating hyperplane between positive and negative examples characterized by the largest margin.



See the Andrew Moore's SVM tutorial https://www.cs.cmu.edu/~./awm/tutorials/svm.html

Support Vector Machines (SVM)



Linear kernel

$$K(x,y) = x^T y, \quad x,y \in \mathbb{R}^d$$

Gaussian kernel (RBF Kernel):

$$K(x,y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}, \quad x,y \in \mathbb{R}^d, \sigma > 0$$

[Vapnik VN, The Nature of Statistical Learning Theory. New York: Springer (1995)]

Slides from Andrew Moore's SVM tutorial https://www.cs.cmu.edu/~./awm/tutorials/svm.html

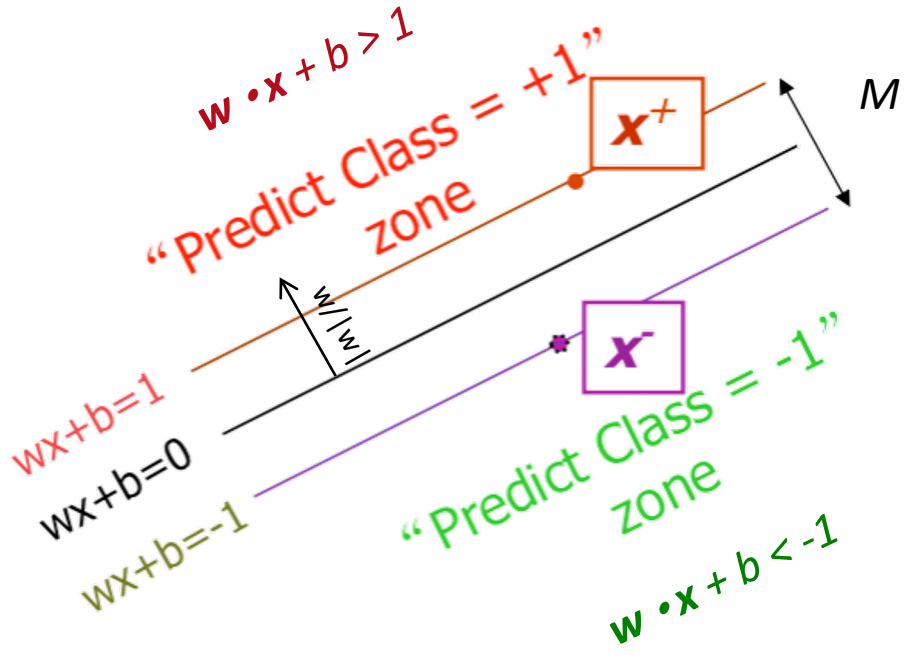
Support Vector Machines (SVM)

Computing the margin width (*M*)

What we know:

- $\boldsymbol{w} \bullet \boldsymbol{x}^{+} + b = +1$
- $\boldsymbol{w} \bullet \boldsymbol{x}^{-} + b = -1$
- $\boldsymbol{w} \bullet (\boldsymbol{x}^{+} - \boldsymbol{x}^{-}) = 2$

- $M = \boldsymbol{w}/|\boldsymbol{w}| \bullet (\boldsymbol{x}^{+} - \boldsymbol{x}^{-}) = 2/|w|$



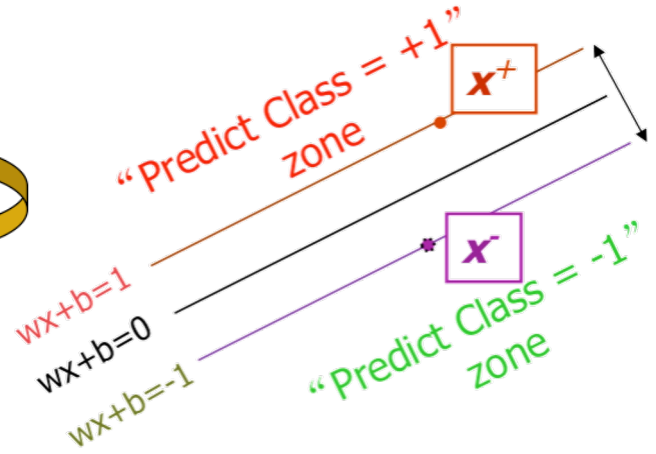
Support Vector Machines (SVM)

- Goal: **1) Correctly classify all training data**

$$wx_i + b \geq 1 \quad \text{if } y_i = +1$$
$$wx_i + b \leq -1 \quad \text{if } y_i = -1$$
$$y_i(wx_i + b) \geq 1 \quad \text{for all i}$$

2) Maximize the Margin $M = \frac{2}{|w|}$

same as minimize $\frac{1}{2} w^t w$



- **We can formulate a Quadratic Optimization Problem and solve for w and b**

Find $\boldsymbol{w}$ and b such that
$\Phi(\boldsymbol{w}) = \frac{1}{2}\, \boldsymbol{w}^t\boldsymbol{w}$ is minimized;
and for all $\{(\boldsymbol{x}_i, y_i)\}$: $y_i\,(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$

[Vapnik VN, The Nature of Statistical Learning Theory. New York: Springer (1995)]

- 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.

Discrimination maps

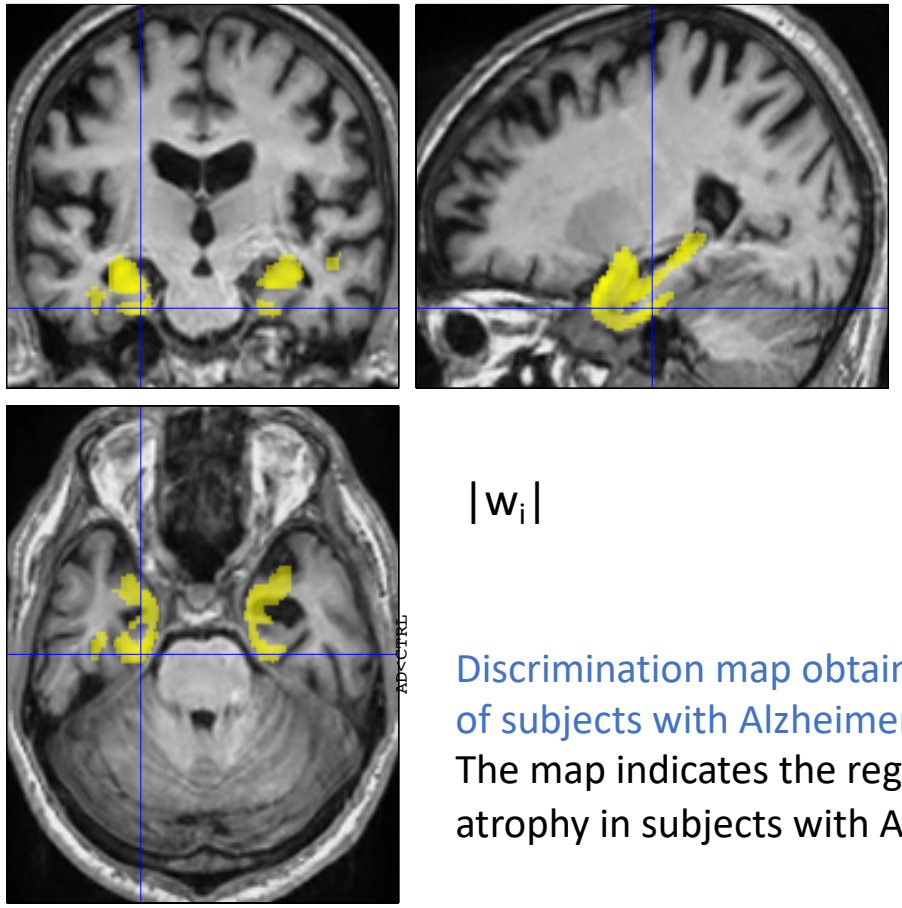
Linear-kernel SVMs allow direct extraction of the weight vector as an image.

- During the SVM training the separating hyperplane is identified so that

  $\underline{w} \bullet \underline{x} + b = 0$

  where $\underline{x}$ is a data pattern, $\underline{w}$ is the weight vector and b is an offset

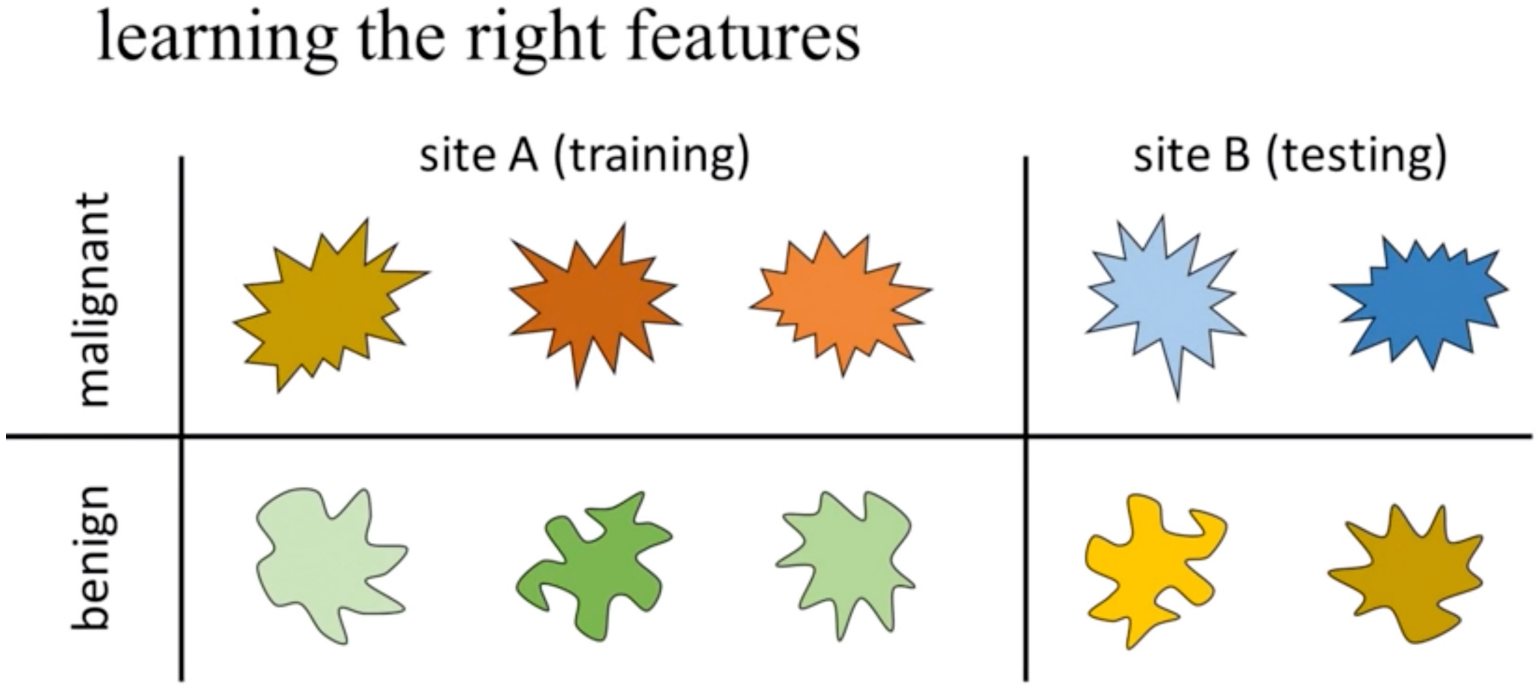
- $\underline{w}$ can be used to generate a map of the most discriminating voxels/regions



$|w_i|$

Discrimination map obtained in the classification of subjects with Alzheimer's Disease vs. Control
The map indicates the regions of local gray matter atrophy in subjects with Alzheimer's Disease

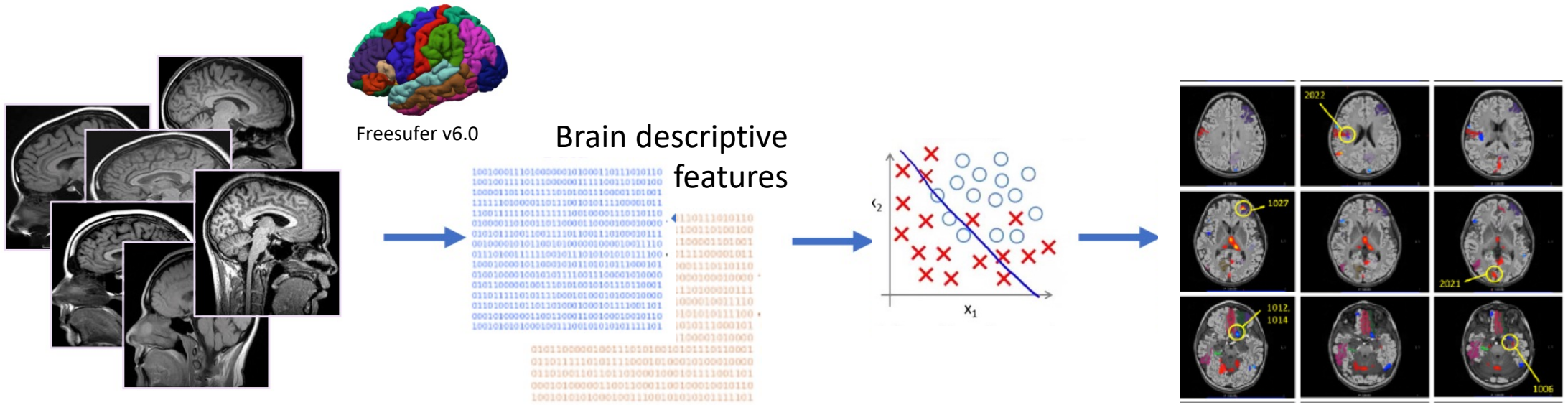
Learning the right information: confounders



- If we obtain good discrimination performance between malignant and benign masses are we sure the classifier is exploiting the right mass properties?
- A classifier trained on data from site A which learnt to distinguish masses according to color tones, will not work on data from site B.

All possible confounder variables should be accounted for in the analysis.
Classification results should be cross checked.

Analysis of MRI data: confounding parameters

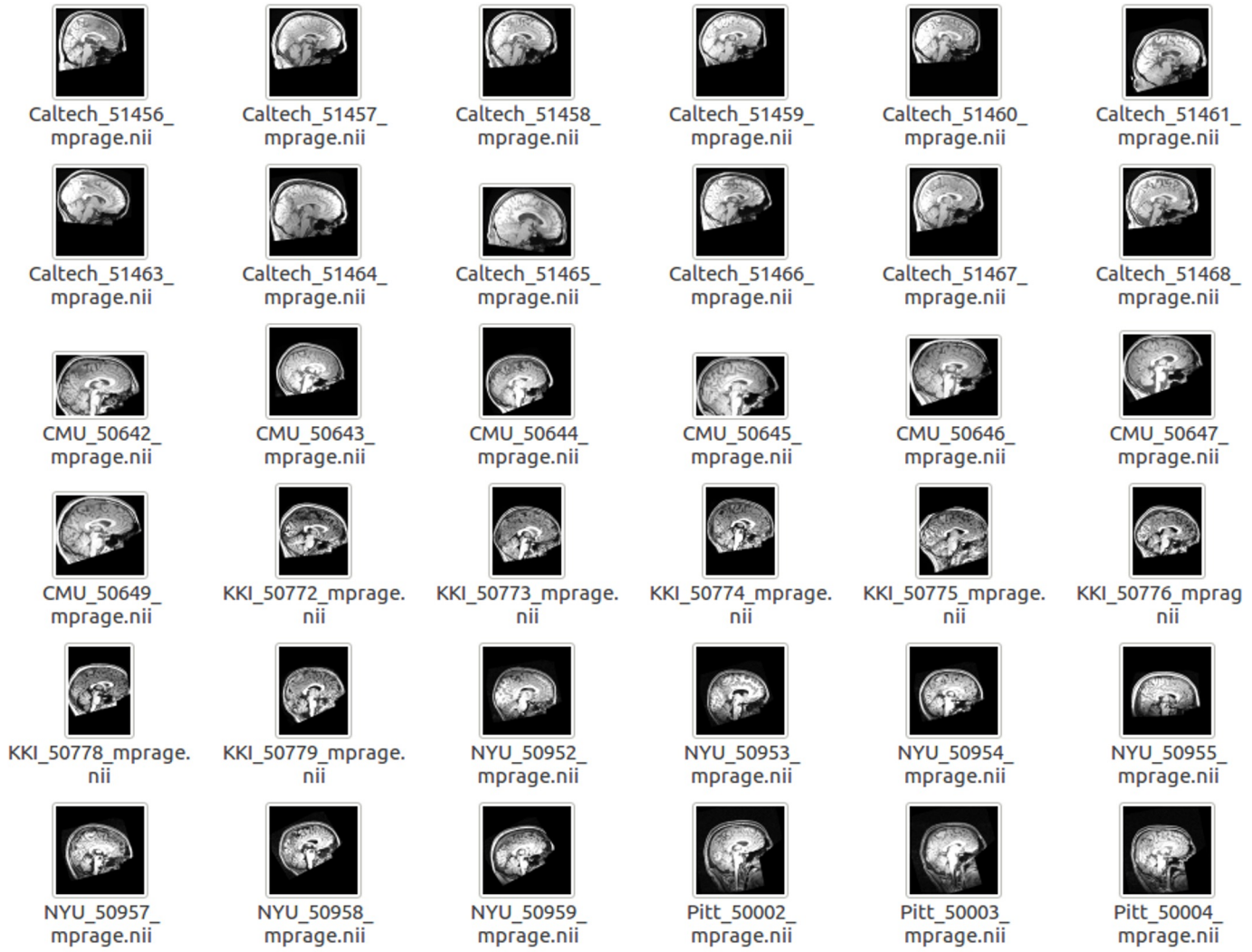


- **Confounding variables** introduce bias in the classifier training phase, suggesting correlations that in fact are not there.
- Biases introduced by the MRI **acquisition site** strongly affect the classification results

Multicenter MRI datasets: the ABIDE sample

Data gathered by different sites and/or acquisition systems carries local “fingerprint”, which can hide subtle information of interest.

This problem is similar to the management of **systematic errors**



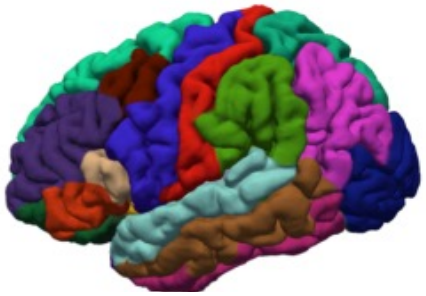
Autism Brain Imaging Data Exchange

ABIDE

NITRC Neuroimaging Tools & Resources Collaboratory

http://fcon_1000.projects.nitrc.org/indi/abide

Site dependence of sMRI data



http://freesurfer.net

volume and thickness of 62 brain parcels for each subject

- ABIDE healthy subjects
- $Site_i$ vs. $Site_j$ binary classification

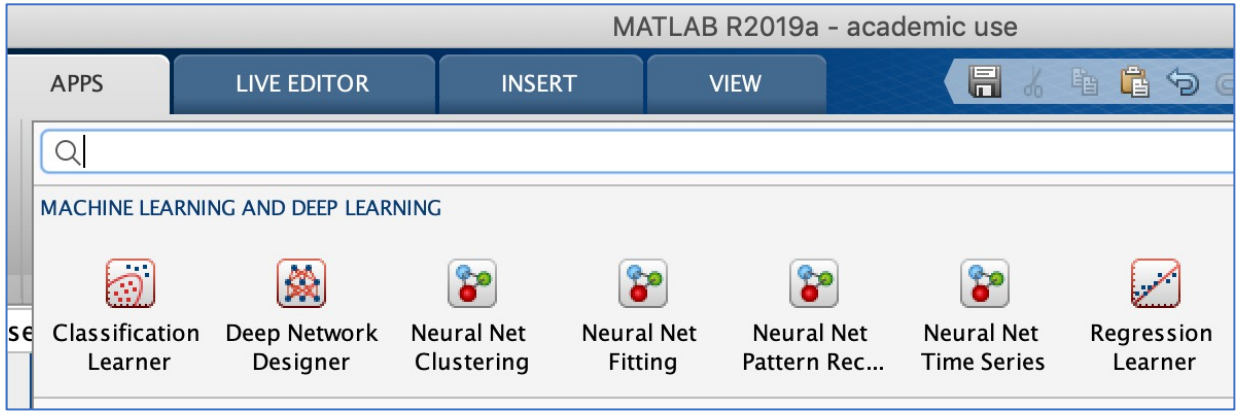
*E. Ferrari et al., "Dealing with confounders and outliers in classification medical studies: the Autism Spectrum Disorders case study", **AIIM** 108:101926, 2020*

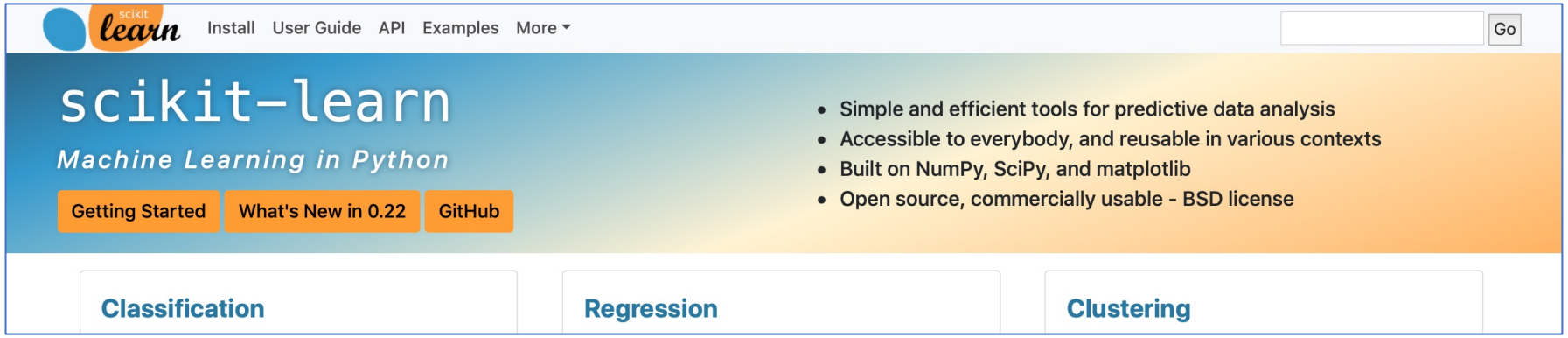
Areas Under the ROC Curve (AUC) obtained in two-class classification

	NYU ABIDE1	NYU-1 ABIDE2	NYU-2 ABIDE2	OHSU ABIDE1	OHSU ABIDE2	USM ABIDE1	USM ABIDE2	UM-1 ABIDE1	UM-2 ABIDE1
NYU ABIDE1	-	0.78	0.89	0.99	1.00	0.99	1.00	0.99	0.98
NYU-1 ABIDE2		-	0.70	0.99	1.00	1.00	1.00	0.99	0.98
NYU-2 ABIDE2			-	1.00	0.98	0.99	0.99	1.00	1.00
OHSU ABIDE1				-	0.63	0.97	0.96	1.00	1.00
OHSU ABIDE2					-	0.99	0.96	0.98	0.98
USM ABIDE1						-	0.75	0.99	0.99
USM ABIDE2							-	0.97	0.97
UM-1 ABIDE1								-	0.96
UM-2 ABIDE1									-

How can we eliminate/mitigate the bias due to data acquisition information?

Tools to solve classification problems





https://github.com/retico/cmepda_medphys/tree/master/L7_code

- see Lecture7_demo_classification.mlx (use the function SVMtrainCV.m) and Lecture7_ML_prediction.ipynb
- The data samples used (brain features from ABIDE and features of malignant/benign mammographic masses) are in DATASETS/FEATURES/

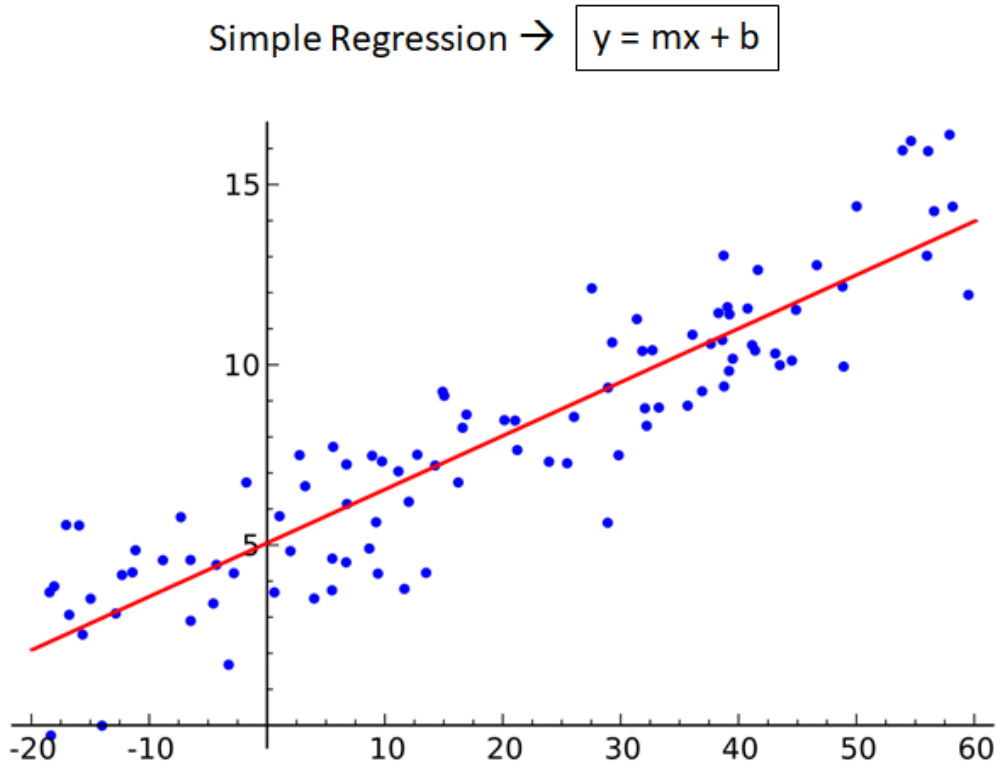
Regression predictive models

- Supervised learning: it aims to model the relationship between a certain number of features (multivariate regression problem) and a continuous target variable.

$$y = f(x)$$

- the most common figure of merit used to estimate the performance of a regression predictive model is the root mean squared error (RMSE)

$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$



See demo code on
https://github.com/retico/cmepda_medphys
L7_code/Lecture7_demo_regression.m
L7_code/Lecture7_regression_models.ipynb

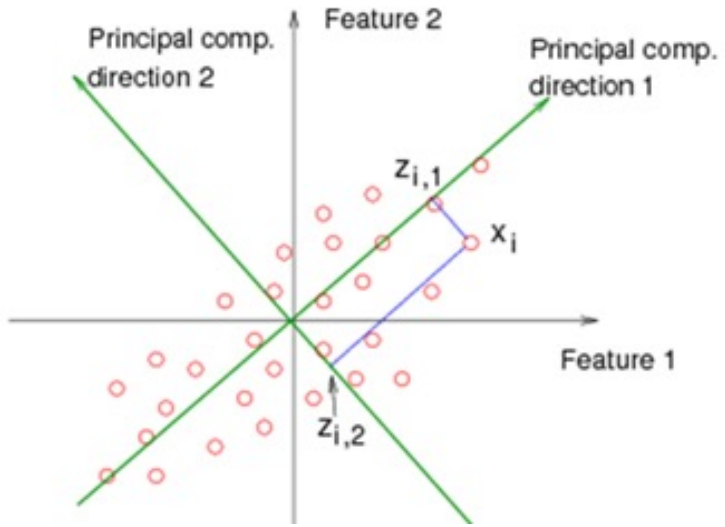
Dimensionality reduction

- The image feature extraction procedures may generate a high amount of features.
- As the number of available subjects to analyse in medical imaging studies is very often limited to ~100 or ~1000 at most, it is important to identify the most informative input variables (feature selection).
- What is the best number of features to consider?
  - A rule of the thumb say that almost 10 examples should be available for each model weight to be trained.
- Dimensionality reduction procedures aim to reduce the number of variables to a set of principal variables.
  - **Feature selection** (a set of reduced number of features is retained)
  - **Feature transformation** (a set of new features is generated by the combination of the existing features)

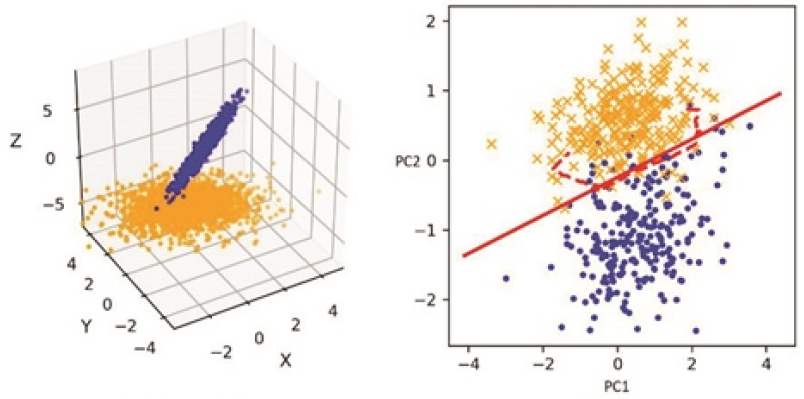
Principal Component Analysis (PCA)

- PCA is an unsupervised technique which converts a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.
- It uses an orthogonal transformation and finds a sequence of linear combinations of the variables that have maximal variance and are mutually uncorrelated.
- The principal components of a set of features $X_1, X_2, \ldots, X_p$ are the normalized linear combinations of the features. The first principal component is:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \ldots + \phi_{p1} X_p \qquad \sum_{j=1}^{p} \phi_{j1}^2 = 1$$



- PCA is very effective in dimensionality reduction when variables are highly correlated
- PCA also serves as a tool for data visualization, as it finds a lower-dimensional representation of a dataset.
- The information about the relevance of the original imaging feature can be retrieved



Autoencoders

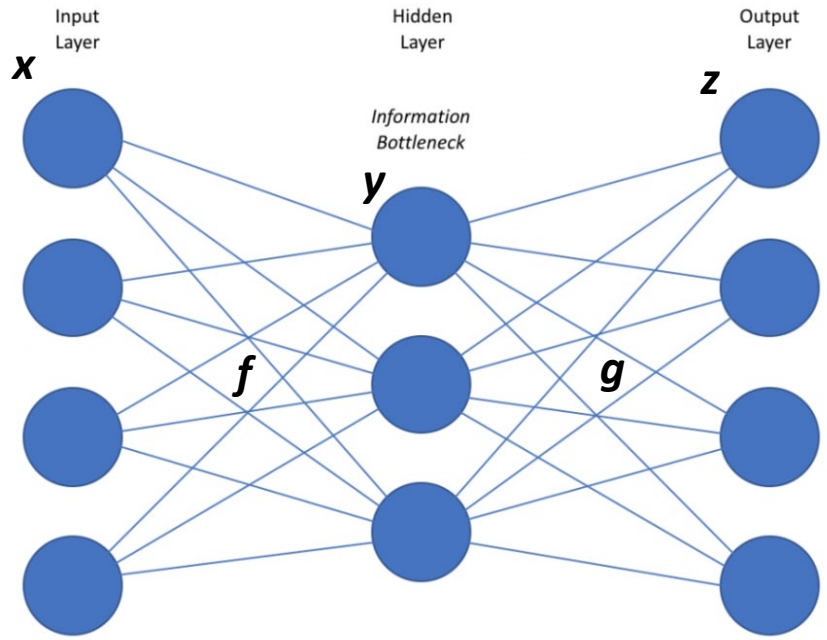
Unsupervised learning.

Artificial Neural Network able to compress (encode) information automatically.

An *encoder* is a deterministic mapping ***f*** that transforms an input vector ***x*** into hidden representation ***y***

A *decoder* maps back via ***g*** the hidden representation ***y*** to the reconstructed input ***z***.

An **autoencoder** compares the reconstructed input ***z*** to the original input ***x*** and try to minimize the reconstruction *error*.



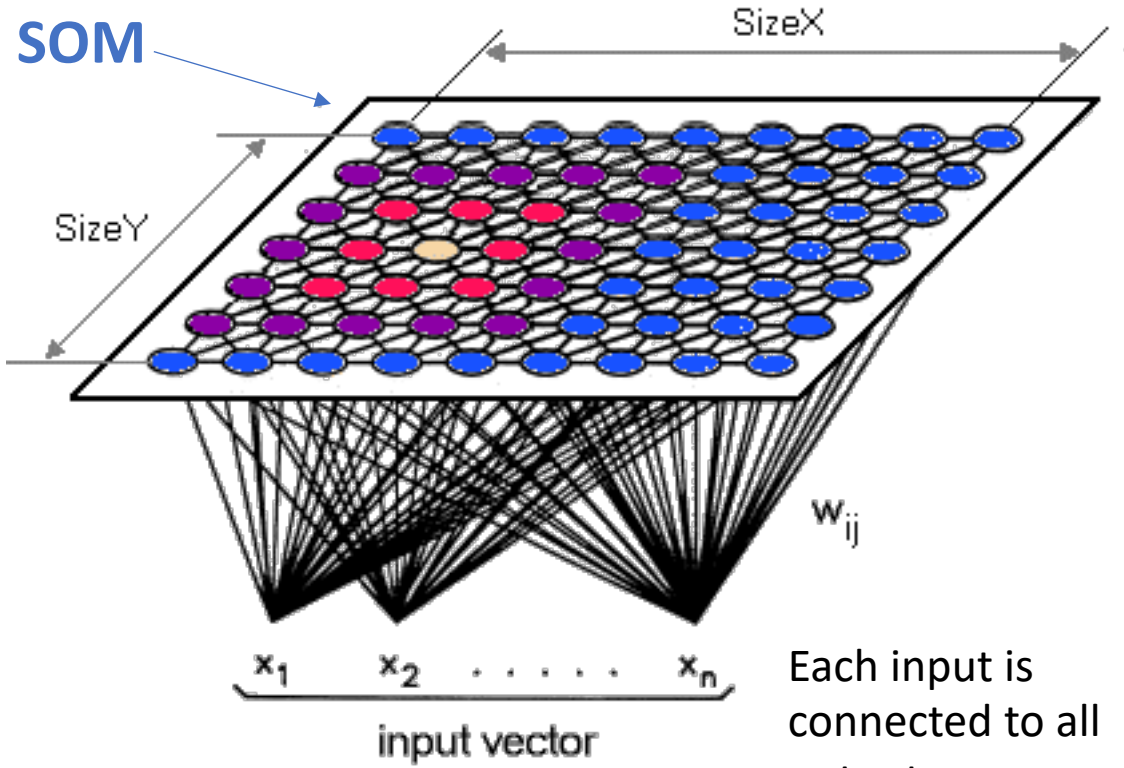
Autoencoders can be used:

- to compress the information (using the higher-level representations ***y***).
- to denoise images, as ***y*** are relatively stable and robust to input corruption.

Bourlard, H.; Kamp, Y. (1988). "Auto-association by multilayer perceptrons and singular value decomposition". *Biological Cybernetics*. **59** (4–5): 291–294

Self Organizing Maps (SOM)

- Unsupervised learning.
- Unsupervised Artificial Neural Networks that maps multidimensional data onto a 2 dimensional grid
- SOM are known also as Kohonen feature maps, as they were introduced by T. Kohonen in 1982
- SOM combine a **competitive learning** principle with a topological structuring of neurons such that adjacent neurons tend to have similar weight vectors.

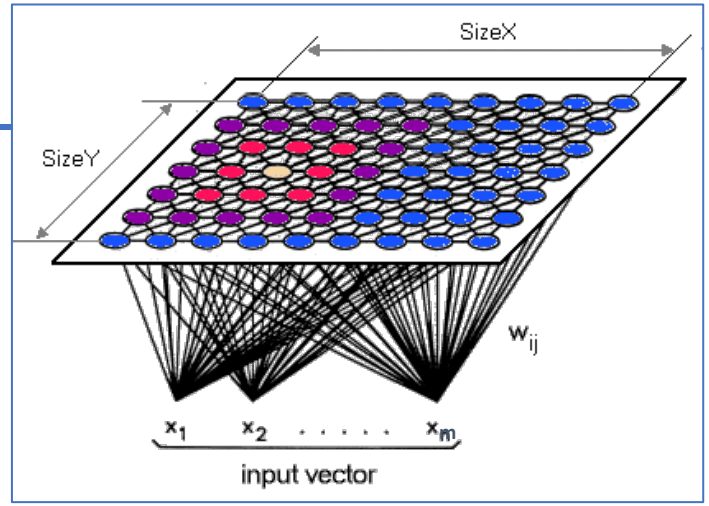


Each input is connected to all output neurons

- SOM can be used for detecting similarity and degrees of similarity
- Underlying neurobiological hypothesis:
  - Structure self-organises based on learning rules and system interaction.
  - Axons physically maintain neighborhood relationships as they grow.

[Teuvo Kohonen. Self-Organization of very large document collections: State of the art (1998)]

Self Organizing Maps (SOM)



Define the SOM size: choose the size of the SOM (e.g. NxN neurons)

Initialization: choose random small values for weights such that

$w_j(0)$ is different for all neurons j.

Sampling: get a sample example x from the input space.

Similarity matching: find the best matching (Euclidean dist.) winning neuron $i(x)$ at step n:

$$i(x) = arg\ min_j\ ||\ x - w_j(n)\ ||\qquad j \in [1,2,\ldots,NxN] \quad \rightarrow\ i \text{ is the winner neuron}$$

Updating: adjust synaptic weight vectors of winning neuron i and its neighbors

$$w_j(n+1) = w_j(n) + \eta(n)\, h(j,i,n)\, (x - w_j(n))$$

where η(t) is a leaning coefficient (decreasing with increasing n) and $h(j,i,n)$ defines the neuron neighboring condition, which has its max for $j=i$ (it can be Gaussian shaped) and its extent can decrease with increasing n.

Continuation: go to Sampling step until no noticeable changes in the feature map are observed.

SOM interpretation

- There are two ways to interpret a SOM.
  - Because in the training phase weights of the whole neighborhood are moved in the same direction, similar items tend to excite adjacent neurons. Therefore, SOM forms a semantic map where similar samples are mapped close together and dissimilar ones apart.
  - The other way is to think of neuronal weights as pointers to the input space. They form a discrete approximation of the distribution of training samples. More neurons point to regions with high training sample concentration and fewer where the samples are scarce.

Data-mining applications: discovering similarities in data.

Large SOMs display emergent properties. In maps consisting of thousands of nodes, it is possible to perform cluster operations on the map itself.

See demo code on https://github.com/retico/cmepda_medphys
L7_code/Lecture7_demo_SOM.m

See also
https://pypi.org/project/sklearn-som/

References and sources

- https://www.datasciencecentral.com/profiles/blogs/roc-curve-explained-in-one-picture
- https://www.cs.cmu.edu/~./awm/tutorials/svm.html
- https://it.mathworks.com/help/stats/classificationlearner-app.html
- https://scikit-learn.org/stable/
- https://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html
- https://pypi.org/project/sklearn-som/