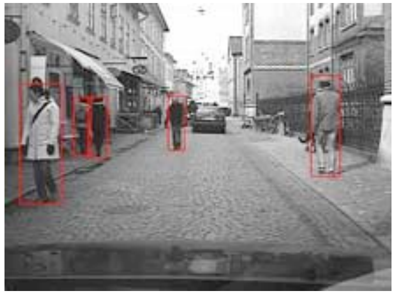
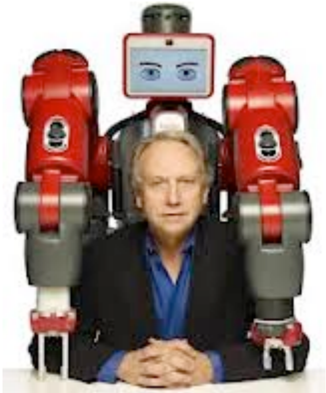
Machine Learning: a crush course

machine learning applications

Francesca Odone & Lorenzo Rosasco

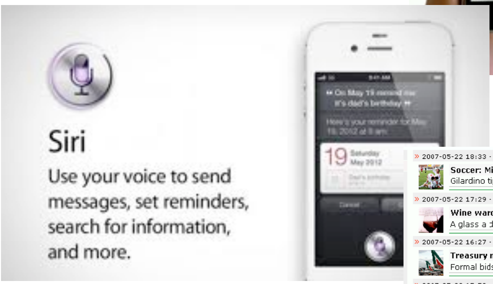
ML applications

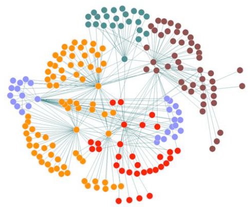


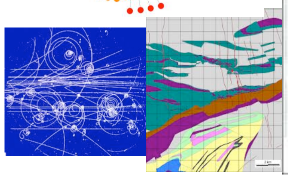


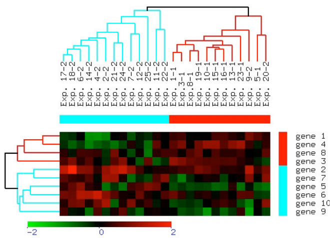
Machine Learning

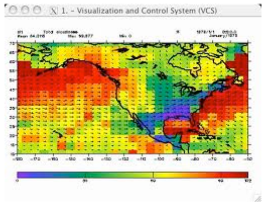
systems are trained on *examples* rather than being programmed





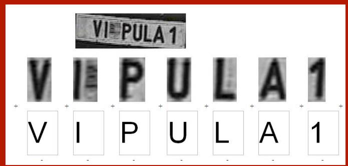






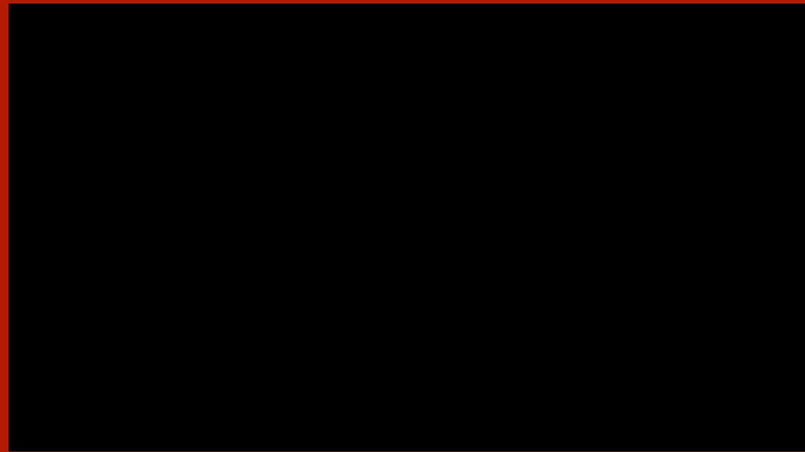
quite a few belong to our everyday experience

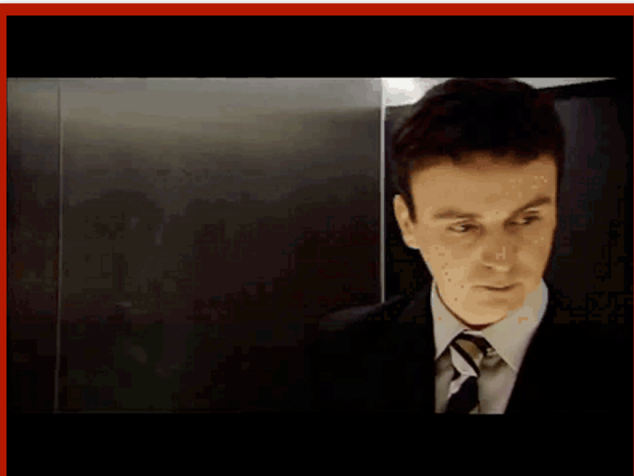
OCR



face detection

Pedestrian detection





speech recognition

but how do they relate with the course contents?

Class #	Day	Date	Month	Daily Schedule	Subject
1	Tue	18	February	9:00 - 11:00	Introduction to Machine Learning
2	Tue	18	February	11:00 - 1:00	Local Methods and Model Selection
3	Tue	18	February	14:00 - 16:00	***Lab** on LM: K-NN, PW for classification*
4	Wed	19	February	9:00 - 11:00	Regularization Networks I: Linear Models
5	Wed	19	February	11:00 - 1:00	Regularization Networks II: Kernels
6	Wed	19	February	14:00 - 16:00	***Lab** on Regularization Networks*
7	Thu	20	February	9:00 - 11:00	Dimensionality Reduction and PCA
8	Thu	20	February	11:00 - 1:00	Variable Selection and Sparsity
9	Thu	20	February	14:00 - 16:00	***Lab** PCA and Sparsity*
10	Fri	21	February	9:00 - 11:00	Applications of Machine Learning

plan (longer than needed)

medical image analysis: image segmentation

bioinformatics: gene selection

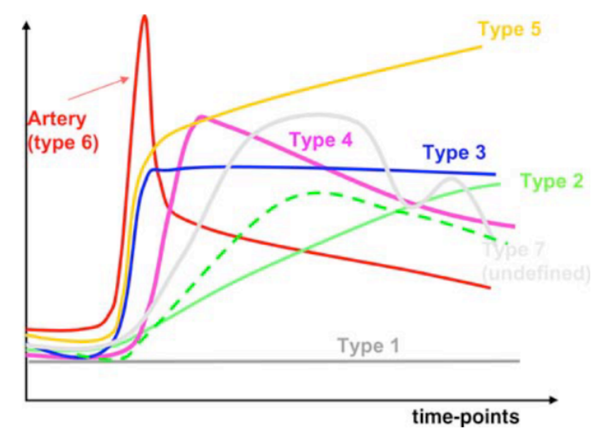
computer vision: object detection, object recognition, ...

human-machine interaction : action recognition, emotion recognition

video-surveillance: behavior analysis, pose detection

Dynamic Contrast Enhanced MRI analysis

Goal: study and implement methods to automatically discriminate different tissues based on different enhancement curve types



Approach:

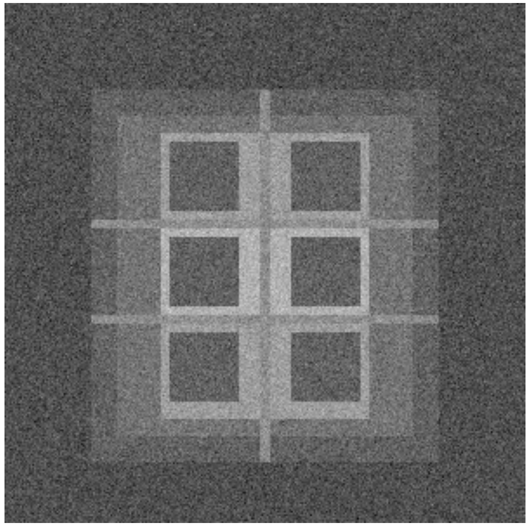
- learn from data basis signals and express the enhancement curves as linear combinations of those signals

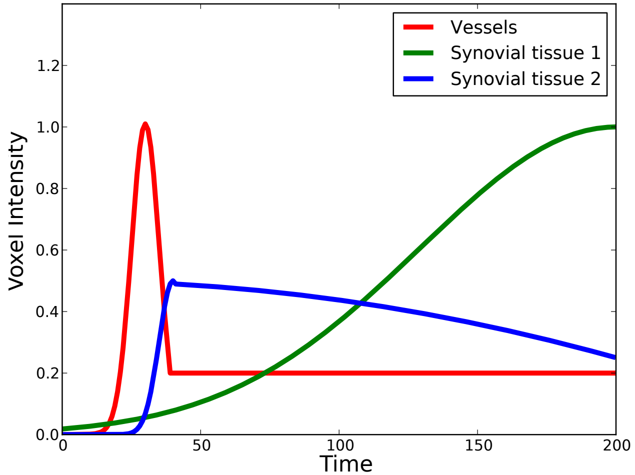
$$\mathcal{D} = \{\phi_j, j = 1, \ldots, p \mid \phi_j : X \to R \;\; \forall j\} \;\; \text{with} \;\; p \leq \infty$$

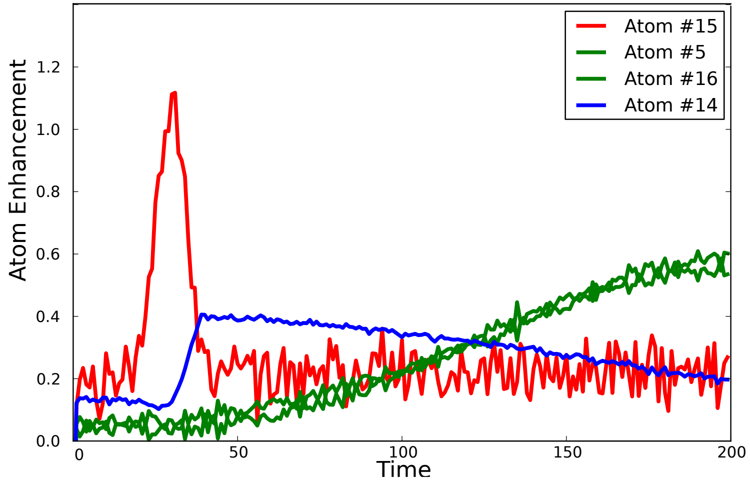
$$K(x, x') = \sum_{j=1}^{p} \phi_j(x)\phi_j(x')$$

Dynamic Contrast Enhanced MRI analysis

the dictionary is learnt from data: $\min_{D,U} ||X - DU||^2 + \tau ||U||_1 \quad s.t. ||d_i||_2 \leq 1$





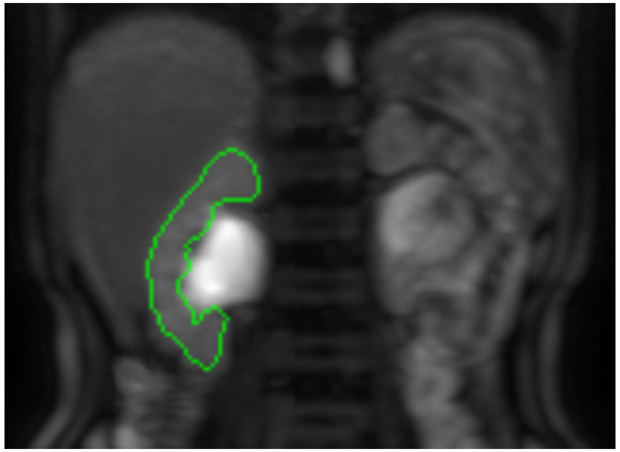


Left: the three different types of generated ECs corresponding to different tissues in the simulated phantom. Right: the four most used atoms, corresponding to the EC patterns associated with each phantom regions.

Dynamic Contrast Enhanced MRI analysis

- Automatic segmentation is obtained by means of an unsupervised method: each voxel is represented by its code (the coefficients u providing the lower reconstruction error w.r.t. the learnt basis D)
- Codes are clustered in 7 main groups (following the expert prior)

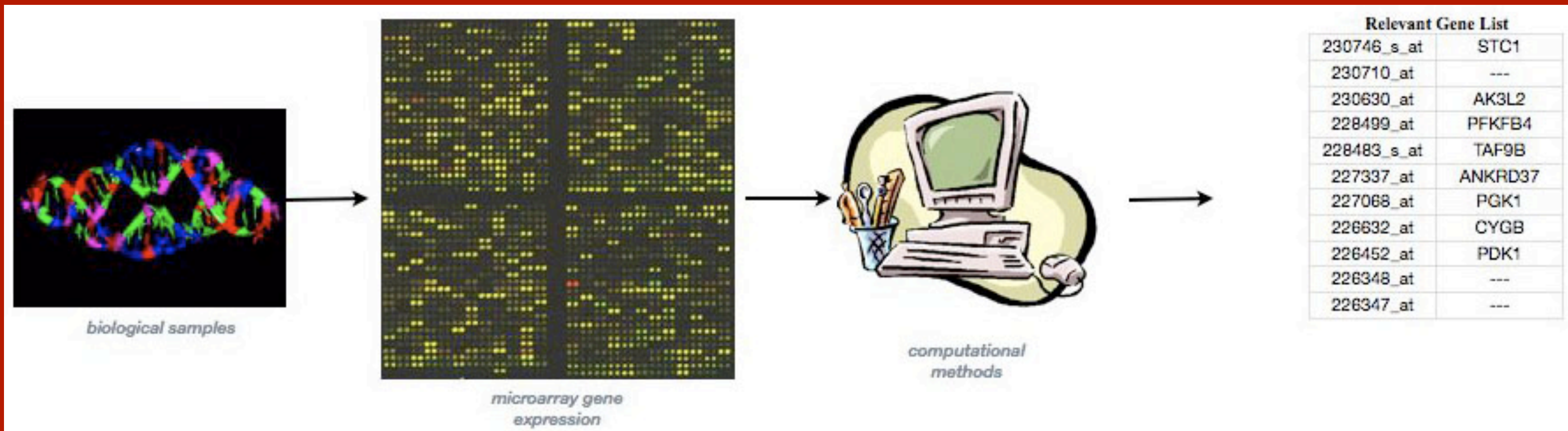
manual annotation provided by the expert



automatic segmentation

Goals:

- Design methods able to identify a gene signature, i.e., a panel of genes potentially interesting for further screening
- Learn the gene signatures, i.e., select the most discriminant subset of genes on the available data



A typical "-omics" scenario:

High dimensional data - Few samples per class

- tenths of data - tenths of thousands genes

  → Variable selection

High risk of selection bias

- data distortion arising from the way the data are collected due to the small amount of data available

  → Model assessment needed

Elastic net and gene selection

$$\min_{\beta \in R^p} ||Y - \beta X||^2 + \tau(||\beta||_1 + \epsilon||\beta||_2^2)$$

Consistency guaranteed - the more samples available the better the estimator

Multivariate - it takes into account many genes at once

Output: One-parameter family of nested lists with equivalent prediction ability and increasing correlation among genes

- $\epsilon \to 0$ minimal list of prototype genes
- $\epsilon_1 < \epsilon_2 < \epsilon_3 < \dots$ longer lists including correlated genes

Double optimization approach

Variable selection step (elastic net)

$$\min_{\beta \in R^p} ||Y - \beta X||^2 + \tau(||\beta||_1 + \epsilon||\beta||_2^2)$$

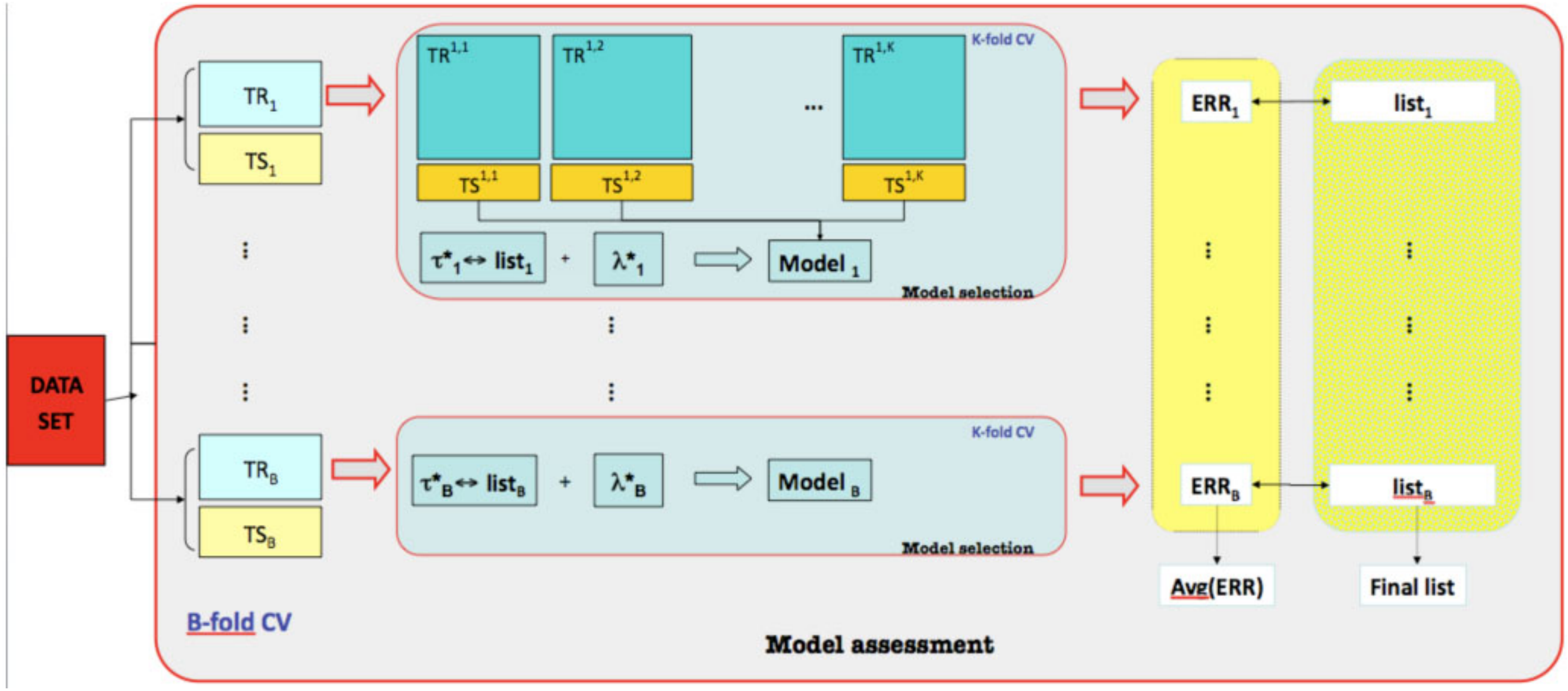
Classification step (RLS)

$$||Y - \beta X||_2^2 + \lambda||\beta||_2^2$$

for each ϵ we have to choose λ and τ

the combination prevents the elastic net shrinking effect

Dealing with selection bias



$$\lambda \quad \to \quad (\lambda_1, \ldots, \lambda_A)$$
$$\tau \quad \to \quad (\tau_1, \ldots, \tau_B)$$

the optimal pair (λ^*, τ^*) is one of the possible $A \cdot B$ pairs (λ, τ)

Computational issues

- Computational time for LOO (for one task)
  $time_{1-\text{optim}} = (2.5s \;\; to \;\; 25s)$
  depending on the correlation parameter

$$\begin{aligned} \text{total time} \quad &= \quad A \cdot B \cdot N_{\text{samples}} \cdot time_{1-\text{optim}} \\ &\sim \quad 20 \cdot 20 \cdot 30 \cdot time_{1-\text{optim}} \\ &\sim \quad 2 \cdot 10^4 s \;\; to \;\; 2 \cdot 10^5 s \end{aligned}$$

- 6 tasks $\rightarrow$ 1 week!!

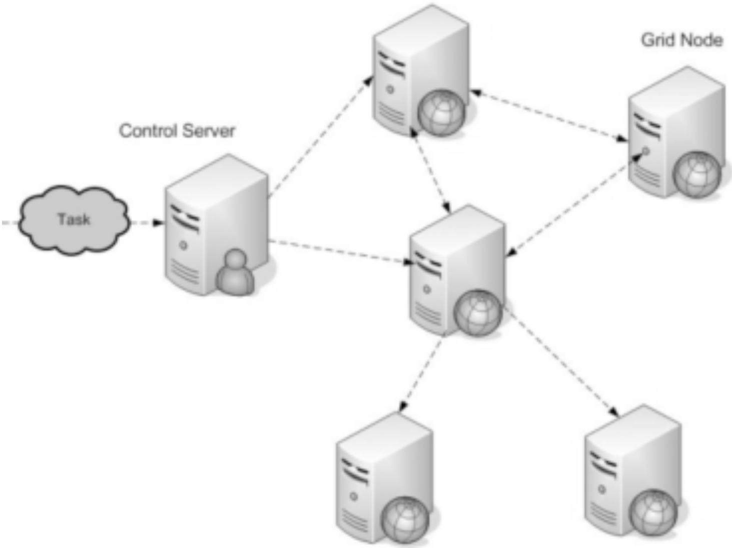


Image understanding

Image understanding is still largely unsolved

- today we are (almost) able to answer more specific questions such as object detection, image categorization, ...

Machine learning has been the key to solve this kind of problems:

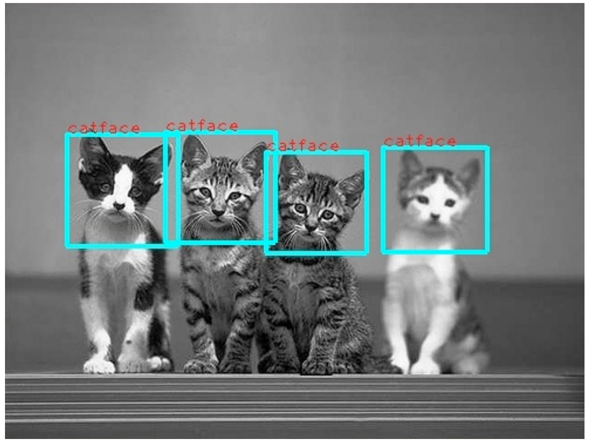
- it deals with noise and intra-class variability by collecting appropriate data and finding suitable descriptions
- Notice that images are relatively easy to gather (but not to label!)
- many large benchmark datasets (with a lot of bias)
- labeling tools

iCubWorld

Object detection in images

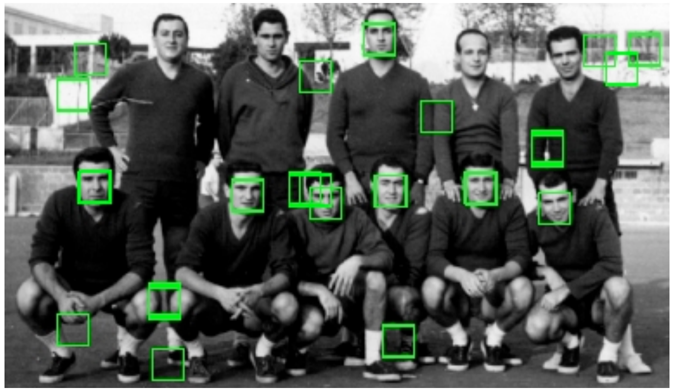
object detection is in essence a binary classification problem

- image regions of variable size are classified: *is it an instance of the object or not?*



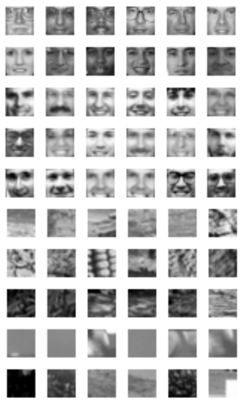
unbalanced classes

- in this 380x220 px image we perform ~6.5×10^5 tests and we should find only 11 positives



the training set contains

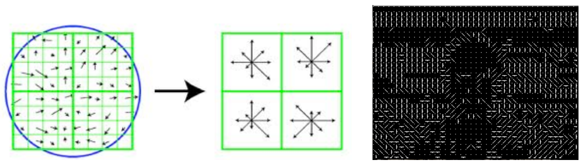
- images of positive examples (the object)
- negative examples (background)

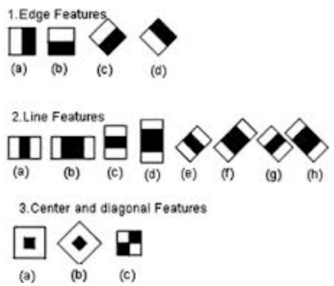


Representing the image content

There is a lot of prior knowledge coming from the computer vision literature (filters, features, ...)

- often it is easier and more effective to find explicit mappings towards high dimensional feature spaces
- feature selection has been used to get rid of redundancy and **speed up computation**





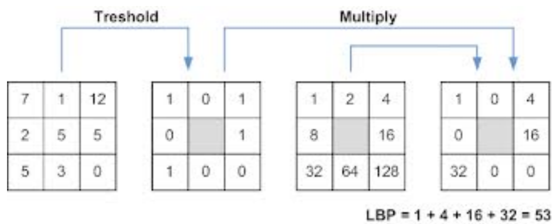
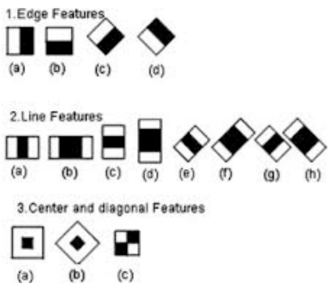
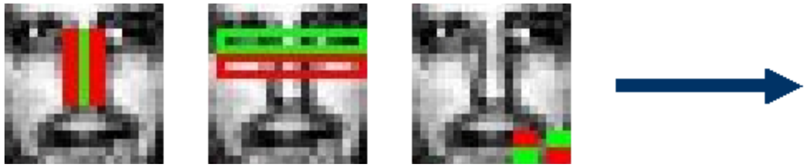


Image feature selection

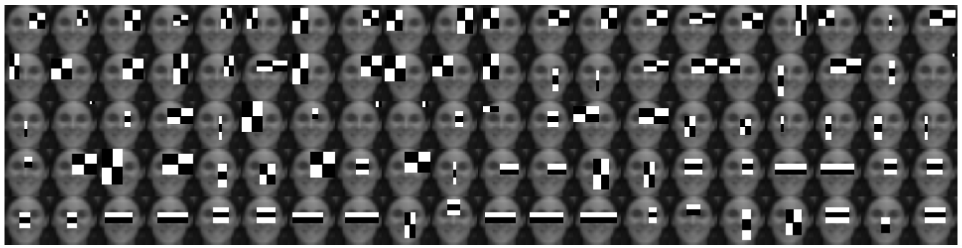




0,413 -0,886 0.547

rectangle features or **Haar-like features** (Viola & Jones) are one of the most effective representations of images for face detection

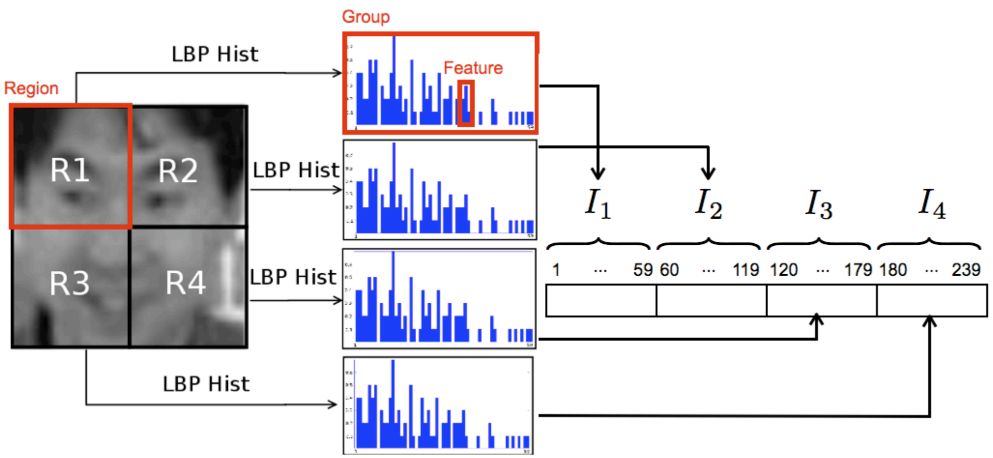
- size of the initial dictionary: a 19 x 19 px image is mapped into a 64.000-dim feature vector!
- feature selection may help us reducing the size and keeping only informative elements



Selecting feature groups

Many image features have a characteristic internal structure

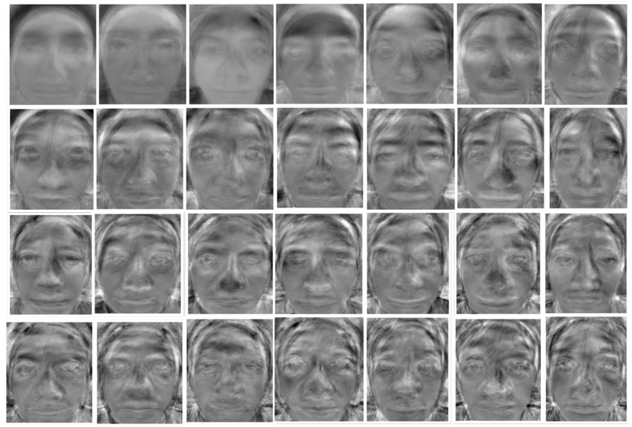
An image patch is divided in *regions* or *cells* and represented according to the specific description, then all representations are concatenated



Feature selection can be designed so to extract an entire group instead than a single feature

an interesting study case: Eigenfaces

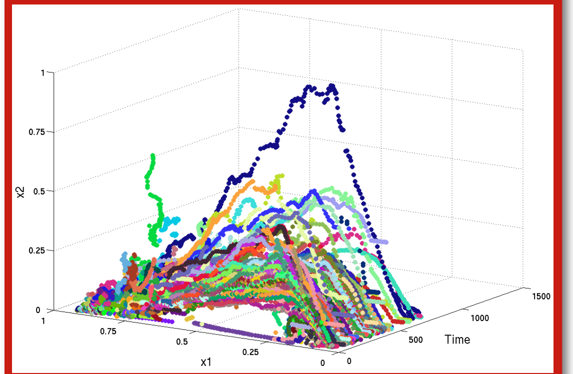
- Goal: represent face images for recognition purposes *(who's that face?)*



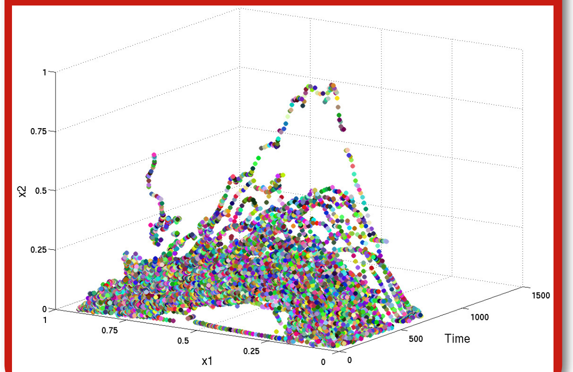
- build X - data matrix where each row is a face image (unfolded)
- PCA(X^TX): each eigenvector can be seen as an image, the *eigenface*;
  - they are the directions in which the images differ from the mean image.
- eigenvectors with the largest eigenvalues are kept
- at run time an image is represented by projecting it onto the chosen directions
- many variants...
  - this simple idea is more appropriate for image matching
  - not robust to illumination and view-point changes

Learning common patterns in temporal sequences

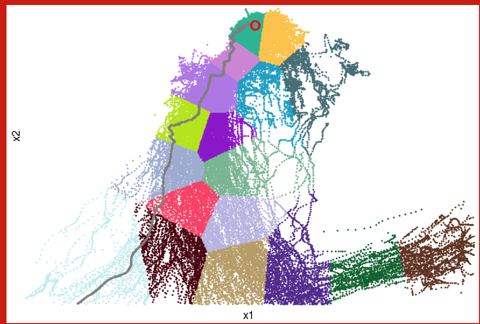
temporal sequences

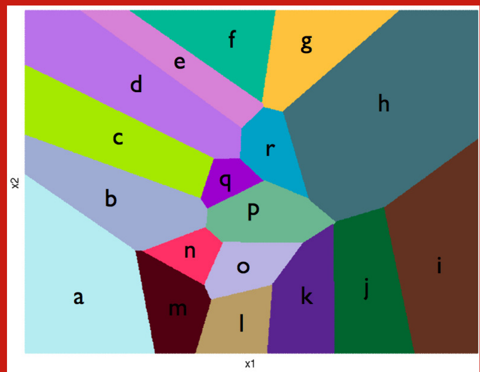


$$\{\mathbf{x}_i\}_{i=1}^N \quad \text{with} \quad \mathbf{x}_i = (x_i^1, x_i^2, \ldots, x_i^{k_i})^\top$$



adaptive space quantization





P-spectrum kernel for sequences

$$\phi_u^P(s) = |\{(v_1, v_2) : s = v_1 u v_2\}|$$

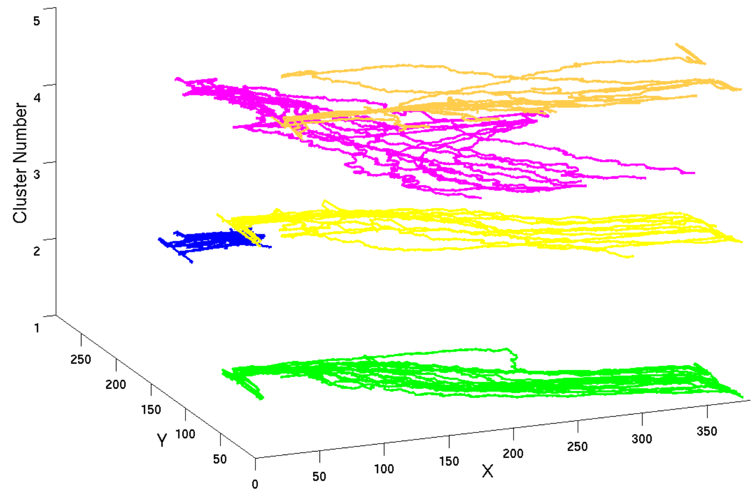
where $u \in \mathcal{A}^P$, while v_1, v_2 are substrings such that $v_1 \in \mathcal{A}^{P_1}$, $v_2 \in \mathcal{A}^{P_2}$, and $P_1 + P_2 + P = |\, s \,|$.

The associated kernel between two strings s_1 and s_2 is defined as:

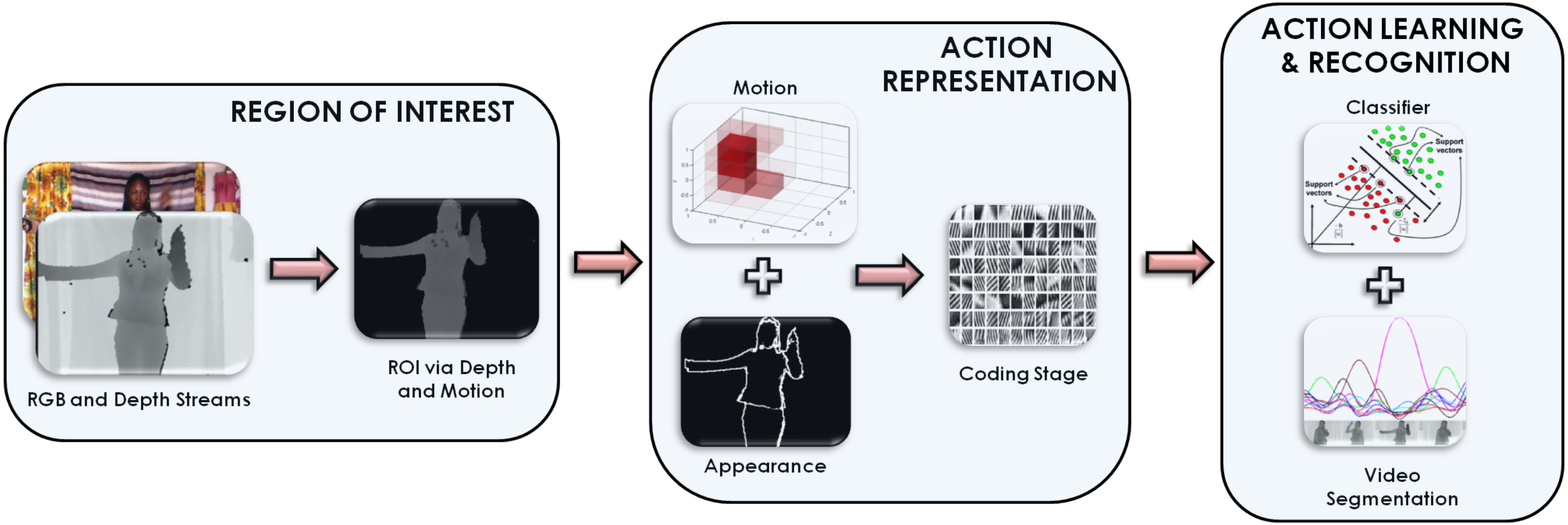
$$K_P(s_1, s_2) = \langle \phi^P(s_1), \phi^P(s_2) \rangle = \sum_{u \in \mathcal{A}^P} \phi_u^P(s_1)\phi_u^P(s_2).$$

String length independence is achieved with an appropriate normalization

$$\hat{K}_P(s_1, s_2) = \frac{K_P(s_1, s_2)}{\sqrt{K_P(s_1, s_1)}\sqrt{K_P(s_2, s_2)}}.$$



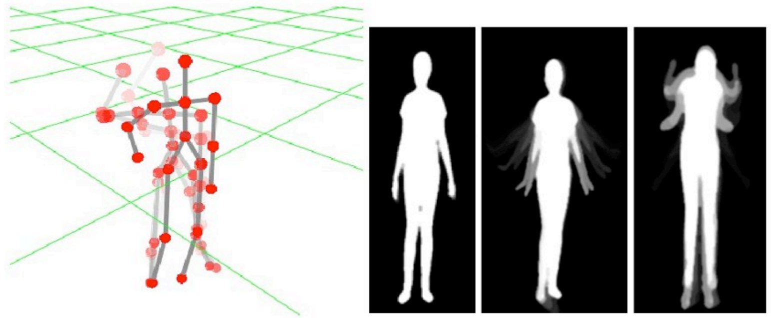
HMI: iCub recognizing actions



HMI: iCub recognizing actions

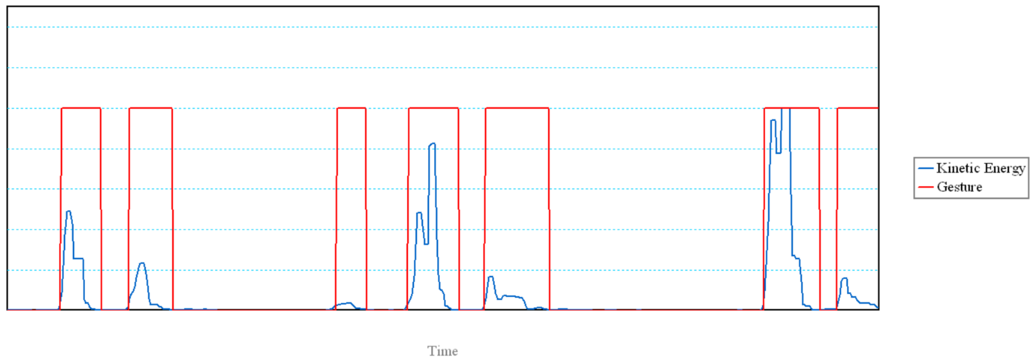
HMI: emotion recognition from body movements

- input data: streams of 3D measurements



- intermediate representations: dimensions suggested by psychologists, related to space occupation or the quality of motion

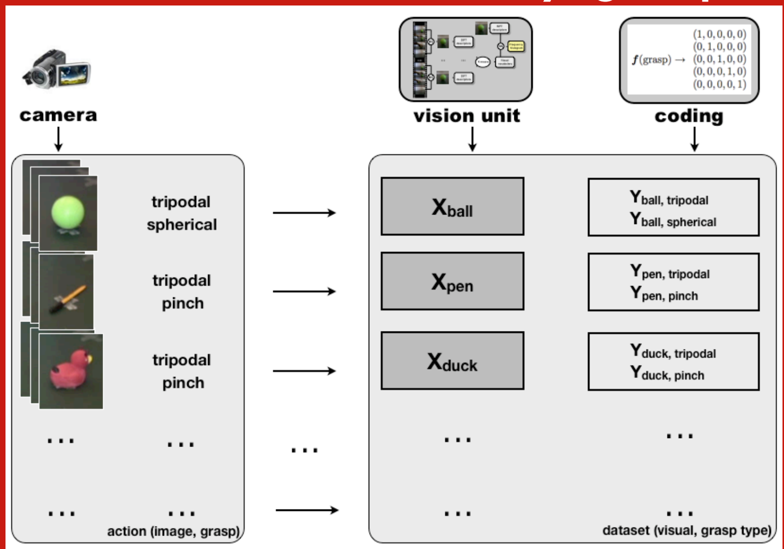
- gesture segmentation



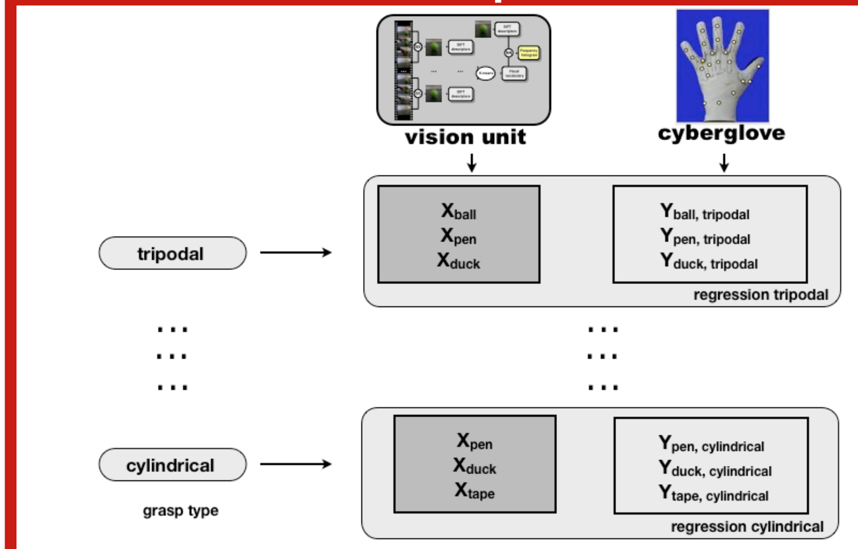
- multi-class classification of 6 emotions based on a combination of binary SVM classifiers

Learning the appropriate type of grasp

estimate the most likely grasps



estimate the hand posture vector

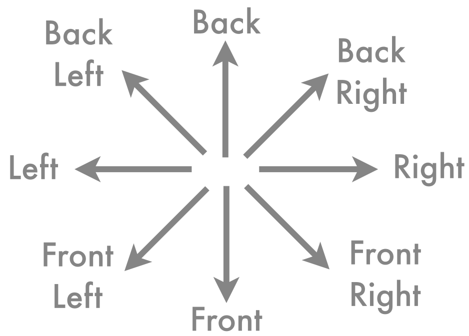


Semi-supervised pose classification

The capability of classifying people with respect to their orientation in space is important for a number of tasks

- An example is the analysis of **collective activities**, where the reciprocal orientation of people within a group is an important feature

- The typical approach relies on quantizing the possible orientations in 8 main angles



- Appearance changes very smoothly and labeling may