UNSUPERVISED LEARNING 2011

LECTURE : INTRODUCTION

Rita Osadchy

Some of the slides are due to O. Veksler and Z. Ghahramani

About the course

- Course Homepage:
  http://www.cs.haifa.ac.il/~rita/uml_course/course.html
- Office hours: request meeting by email
- Contact:
  - You contact me by email: rita@cs.haifa.ac.il
  - I contact you by email: All announcement and guidelines will be distributed by email.

    You must send me an email by November 14 from your active address with the subject “UML course contact”.

  - Those who do not send their contact address on time will not be added to the contact list!!!

About the course

- **Mandatory Prerequisites**

  The course assumes some basic knowledge of probability theory and linear algebra; for example, you should be familiar with

  - Joint and marginal probability distributions
  - Normal (Gaussian) distribution
  - Expectation and variance
  - Statistical correlation and statistical independence
  - Eigen value decomposition

Links to tutorial in the course homepage.

Suggested Prerequisites: Introduction to Machine Learning

About the course

- **Textbook:**

  Pattern Recognition and Machine Learning, by Christopher Bishop. Springer, August 2006.

- Course material: lecture notes and reading material in: http://www.cs.haifa.ac.il/~rita/UML_course/course.htm

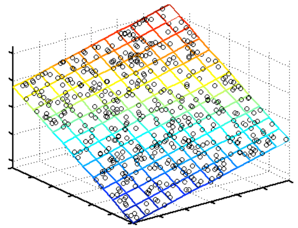
About the course

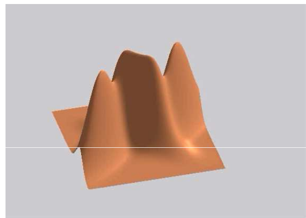
- Grading: Exam

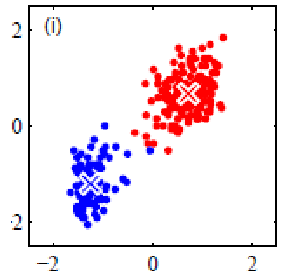
Syllabus at glance

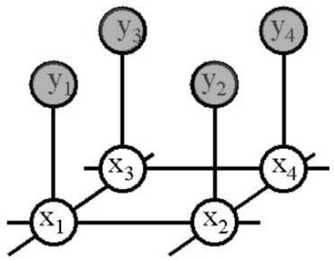
- Dimensionality Reduction
  - Linear
  - Manifold Mapping
  - KPCA
- Density Estimation
  - Mixture of Gaussians
  - EM
  - Factor Analysis
- Clustering
  - K-means
  - Spectral Clustering
- Introduction to Graphical Models
  - Belief Propagation
  - MRF

More topics (see the extended syllabus) if time permits.









Goal

- Machine learning is a study of algorithms that improve their performance at some task with experience.
- Machine learning is an interdisciplinary field focusing on both the mathematical foundations and practical applications of systems that learn, reason and act.
- The goal of this course: to introduce basic concepts, models and algorithms in machine learning with particular emphasis on unsupervised learning.

Types of Learning Problems

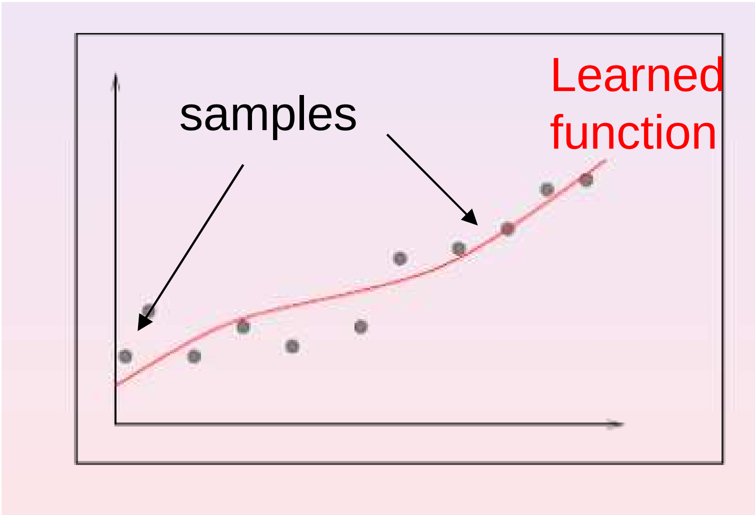
Assume that we are given a set of training inputs (e.g.,a series of sensory inputs) $x_1, x_2,.., x_n$

- Supervised learning: We are also given the desired outputs $y_1, y_2,.., y_n$ and the goal is to learn to produce the correct output given a new input.

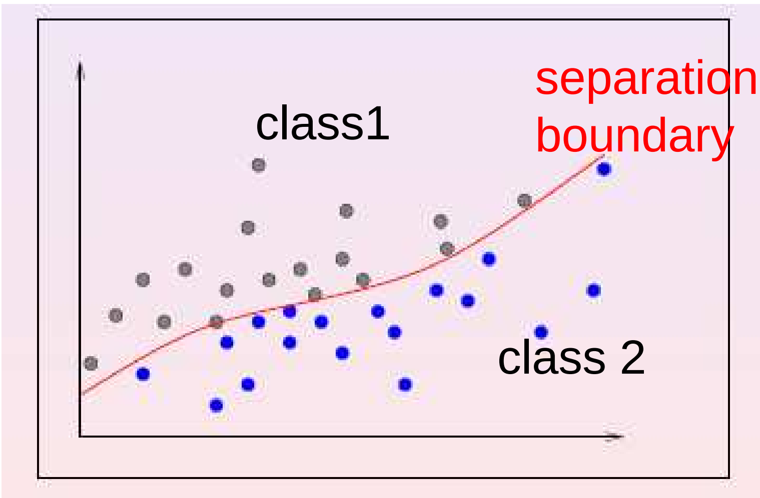
- Unsupervised learning: The goal is to build a model of x that can be used for reasoning, decision making, predicting things, communicating etc.

- Reinforcement learning: where we only get feedback in the form of how well we are doing (e.g., the outcome of the game). The goal is to learn to act in a way that maximises rewards in the long term.

Two kinds of Supervised Learning



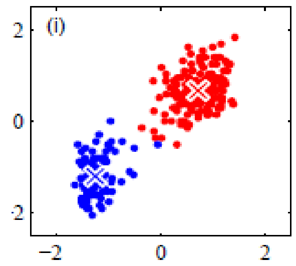
- Regression: Learn a continuous input-output mapping from a limited number of examples in order to predict the output accurately for new inputs.



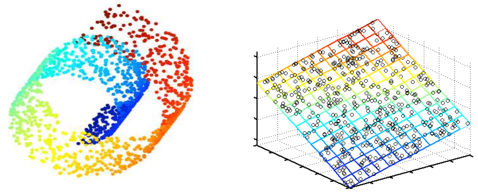
- Classification: outputs are discrete variables (category labels). Learn a decision boundary that separates one class from the other to classify new inputs correctly

Goals of Unsupervised Learning

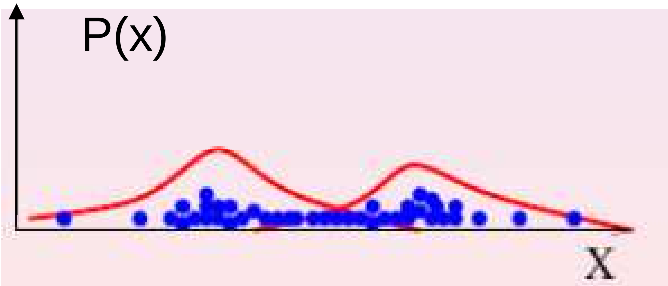
- Clustering: discover "clumps" of points



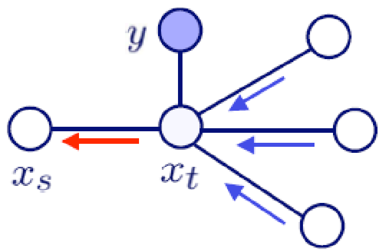
- Embedding: discover low-dimensional manifold or surface near which the data lives.



- Density Estimation. Find a function f such f(X) approximates the probability density of X, p(X), as well as possible.



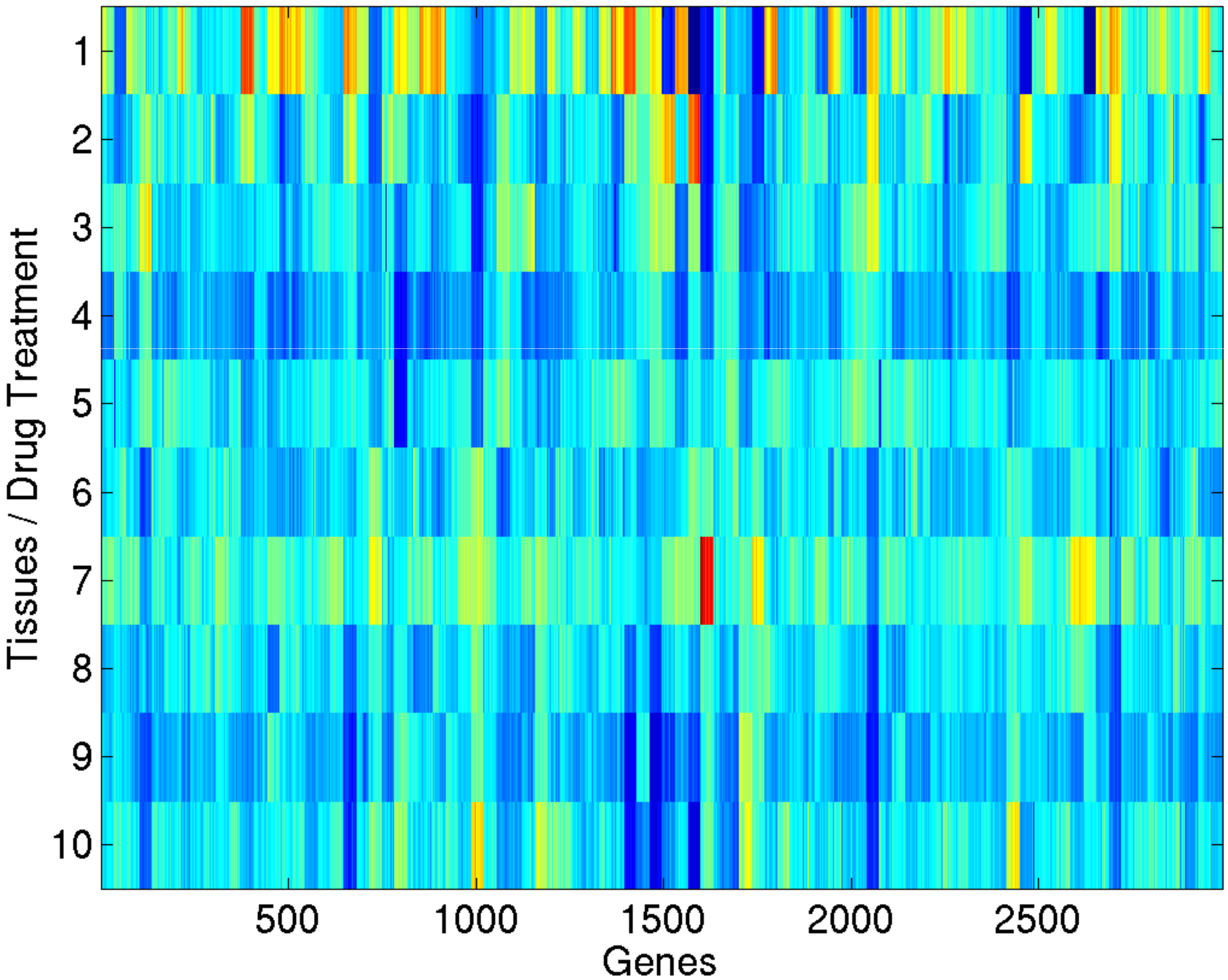
- Finding good explanations (hidden causes)of the data;



Uses of Unsupervised Learning

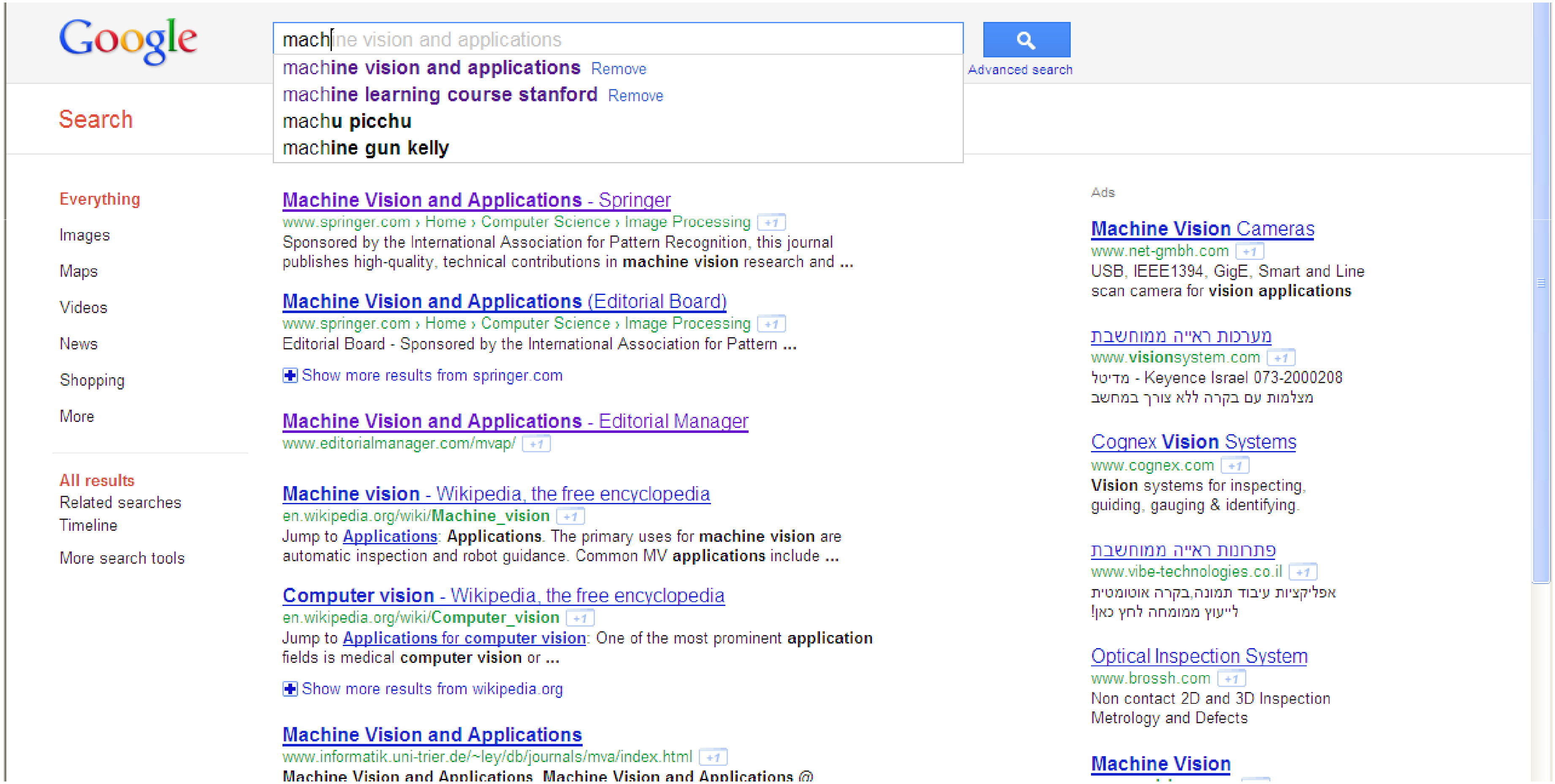
- data compression
- outlier detection
- help classification
- make other learning tasks easier
- use as a theory of human learning and perception

Applications



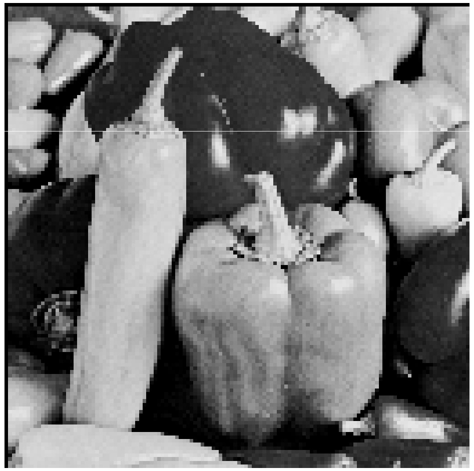
Applications

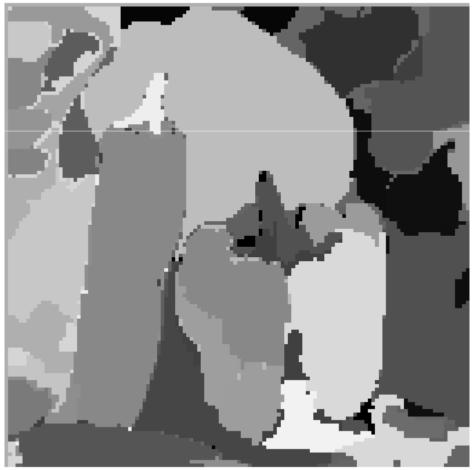
- Profiling Web Users



Applications

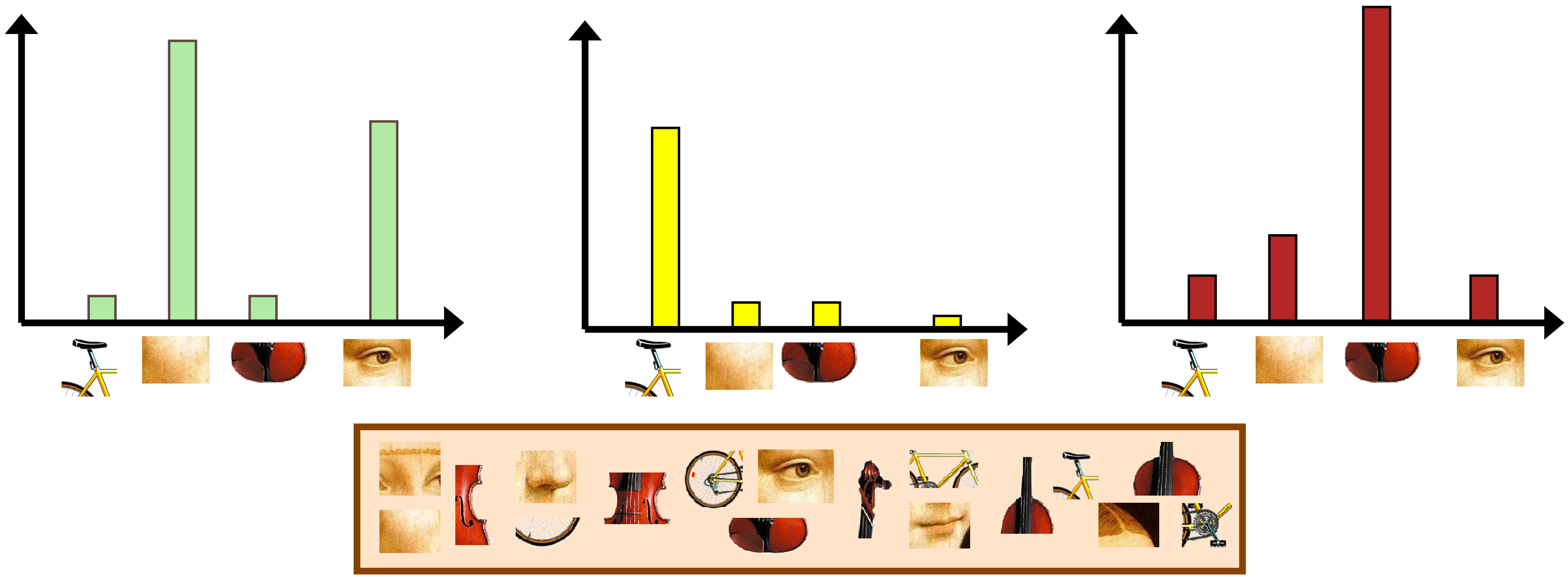
- Image Processing





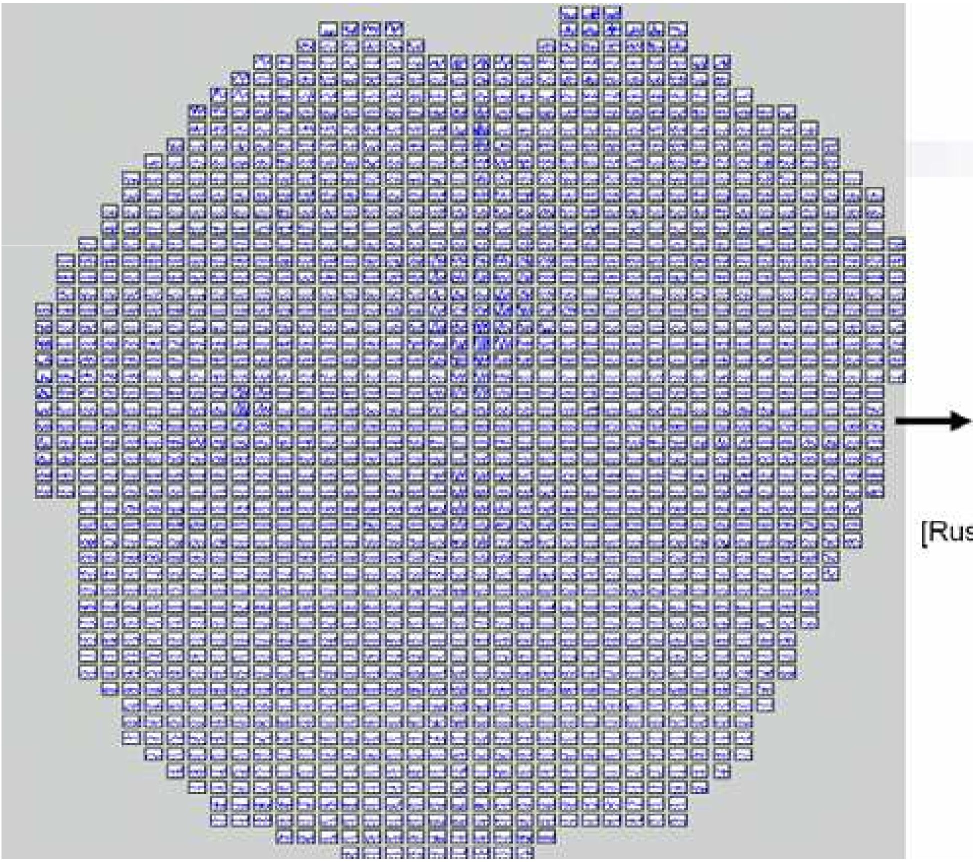
Applications

- Visual Object Categorization



Applications

- Understanding Brain Activities

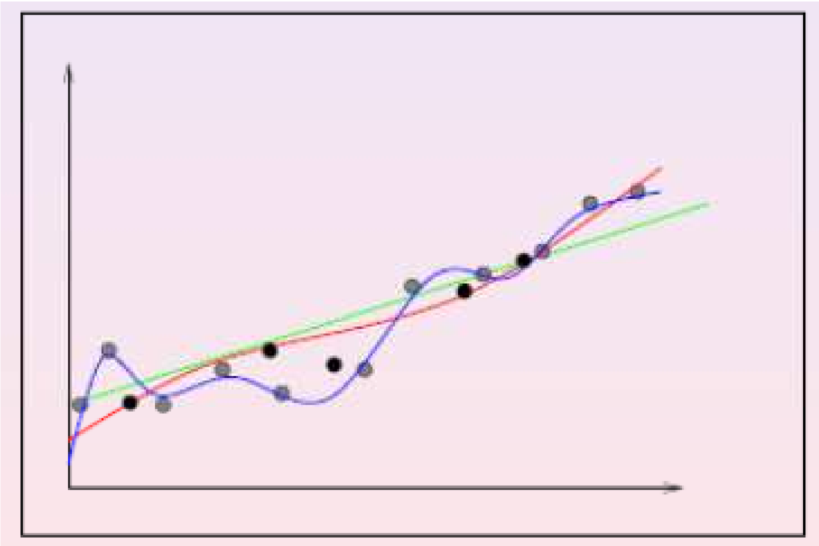


Reading a noun (vs verb)

[Rustandi et al., 2005]

Why Learning is Difficult?

- Given a finite amount of training data, you have to derive a relation for an infinite domain
- In fact, there is an infinite number of such relations

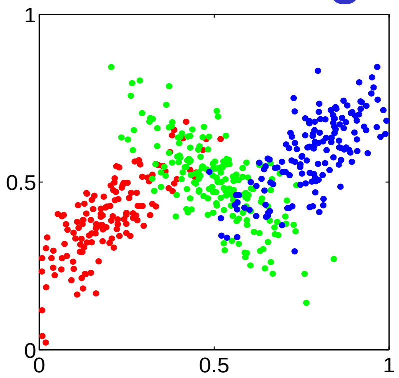


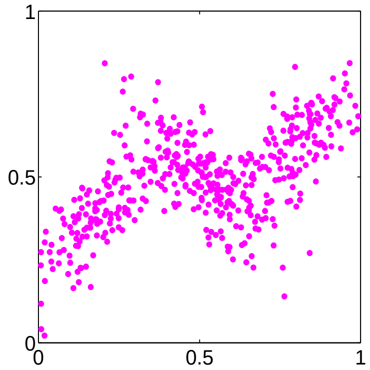
- ... the hidden test points...

What about Unsupervised Learning

- ☺ Obtaining unlabelled data is easier (at least for some applications)
  - More training data – better models
- ☹ We know very little about the data.
  - Less than in supervised learning:

Build classifier for 3 classes, given the training data





A Recap from Introductory Course

- The next several slides summarize few topics from the Introductory course in Machine Learning.
- These definitions and formulations are the basis required to understand the material in this course.
- For more details see: http://www.cs.haifa.ac.il/~rita/ml_course/course.html

Basic Rules of Probability

Probabilities are non-negative $P(x) \geq 0 \; \forall x$.

Probabilities normalise: $\sum_{x \in \mathcal{X}} P(x) = 1$ for distributions if x is a discrete variable and $\int_{-\infty}^{+\infty} p(x)dx = 1$ for probability densities over continuous variables

The joint probability of x and y is: $P(x, y)$.

The marginal probability of x is: $P(x) = \sum_y P(x, y)$, assuming y is discrete.

The conditional probability of x given y is: $P(x|y) = P(x, y)/P(y)$

Bayes Rule:

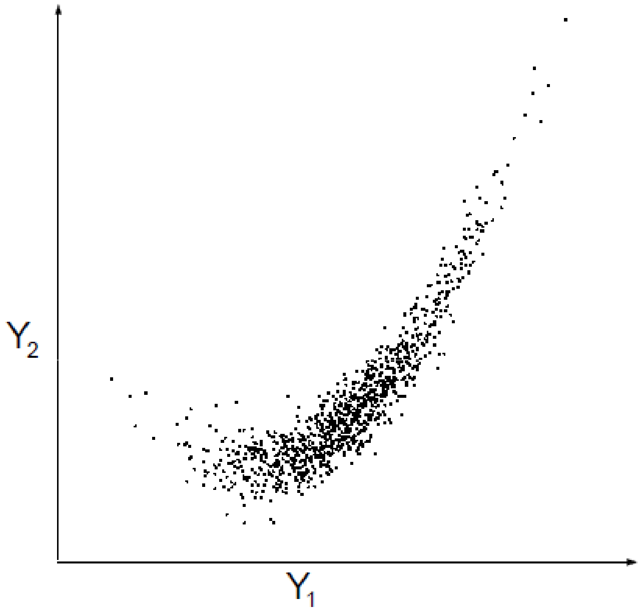
$$P(x, y) = P(x)P(y|x) = P(y)P(x|y) \quad \Rightarrow \quad \boxed{P(y|x) = \frac{P(x|y)P(y)}{P(x)}}$$

Warning: I will not be obsessively careful in my use of p and P for probability density and probability distribution. Should be obvious from context.

Probability Density estimation

- Parametric methods – assume we know the shape of the distribution, but not the parameters θ.
  - Maximum Likelihood Estimation
  - Bayesian Estimation
- Non parametric methods – the form of the density is entirely determined by the data without any model.

Simple Statistical Modeling : modeling correlations



Assume:

- we have a data set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$
- each data point is a vector of D features: $\mathbf{y}_i = [y_{i1} \ldots y_{iD}]$
- the data points are i.i.d. (independent and identically distributed).

One of the simplest forms of unsupervised learning: model the **mean** of the data and the **correlations** between the D features in the data
We can use a multivariate Gaussian model:

$$p(\mathbf{y}|\mu, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\mathbf{y}-\mu)^\top \Sigma^{-1}(\mathbf{y}-\mu)\right\}$$

Maximum Likelihood Parameter Estimation

- Parameters θ are unknown but fixed (i.e. not random variables).
- Given the training data, choose the parameter value θ that makes the data most probable (i.e., maximizes the probability of obtaining the sample that has actually been observed)

Maximum Likelihood Estimation

- Consider the following function, which is called likelihood of θ with respect to the set of samples *D*

$$p(D\,|\,\theta) = \prod_{k=1}^{k=n} p(x_k\,|\,\theta) = F(\theta)$$

- Maximum likelihood estimate (abbreviated MLE) of θ is the value of θ that maximizes the likelihood function $p(D|\theta)$

$$\boxed{\hat{\theta} = \underset{\theta}{\arg\max}\big(p(D\,|\,\theta)\big)}$$

ML Estimation of a Gaussian

Data set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\}$, likelihood: $p(Y|\mu, \Sigma) = \prod_{n=1}^{N} p(\mathbf{y}_n|\mu, \Sigma)$

Maximize likelihood $\Leftrightarrow$ maximize log likelihood
Goal: find μ and Σ that maximise log likelihood:

$$\mathcal{L} = \log \prod_{n=1}^{N} p(\mathbf{y}_n|\mu, \Sigma) = \sum_n \log p(\mathbf{y}_n|\mu, \Sigma)$$

$$= -\frac{N}{2} \log |2\pi\Sigma| - \frac{1}{2} \sum_n (\mathbf{y}_n - \mu)^\top \Sigma^{-1} (\mathbf{y}_n - \mu)$$

Note: equivalently, minimise $-\mathcal{L}$, which is *quadratic* in μ
Procedure: take derivatives and set to zero:

$$\frac{\partial \mathcal{L}}{\partial \mu} = 0 \qquad \Rightarrow \qquad \hat{\mu} = \frac{1}{N} \sum_n \mathbf{y}_n \qquad \text{(sample mean)}$$

$$\frac{\partial \mathcal{L}}{\partial \Sigma} = 0 \qquad \Rightarrow \qquad \hat{\Sigma} = \frac{1}{N} \sum_n (\mathbf{y}_n - \hat{\mu})(\mathbf{y}_n - \hat{\mu})^\top \qquad \text{(sample covariance)}$$

Bayesian Parameter Estimation

- θ is a random variable with prior $\boldsymbol{p}(\theta)$
  - Unlike MLE case, p(x|θ) is a conditional density
- The training data D allow us to convert p(θ) to a posterior probability density p(θ|D) .
  - After we observe the data D, using Bayes rule we can compute the posterior p(θ|D)
- But θ is not our final goal, our final goal is the unknown ***p***(***x***)
- Therefore a better thing to do is to maximize ***p***(***x*|*D***), this is as close as we can come to the unknown ***p***(***x***) !

Bayesian Estimation: Formula for p(x|D)

- From the definition of joint distribution:

$$p(x \mid D) = \int p(x, \theta \mid D) d\theta$$

- Using the definition of conditional probability:

$$p(x \mid D) = \int p(x \mid \theta, D) p(\theta \mid D) d\theta$$

- But $p(x|\theta,D)=p(x|\theta)$ since $p(x|\theta)$ is completely specified by θ

$$p(x \mid D) = \int \underbrace{p(x \mid \theta)}_{\text{known}} \underbrace{p(\theta \mid D)}_{\text{unknown}} d\theta$$

- Using Bayes formula,

$$p(\theta \mid D) = \frac{p(D \mid \theta) p(\theta)}{\int p(D \mid \theta) p(\theta) d\theta}$$

$$p(D \mid \theta) = \prod_{k=1}^{n} p(x_k \mid \theta)$$

Bayesian Estimation vs. MLE

$$p(x \mid D) = \int \underbrace{p(x \mid \theta)}_{\text{proposed model with certain } \theta} \overbrace{p(\theta \mid D)}^{\text{support } \theta \text{ receives from the data}} d\theta$$

- The above equation implies that if we are less certain about the exact value of **θ,** we should consider a weighted average of *p(***x**|**θ***)* over the possible values of **θ.**

- Contrast this with the MLE solution which always gives us a single model:

$$p(x \mid \hat{\theta})$$

Recommended Reading

- More basic topics from the introductory course that we will not talk about in this course, but are related to unsupervised learning
  - Parzen Window
  - Hierarchical Clustering
  - Naïve Bayese Model