Lecture 3

Cezar Ionescu

Unsupervised learning

Unsupervised learning is a form of "learning from experience". It is related to data mining: identifying patterns in data, and to classification.

Example: Assume we have a sequence of images that fall into `k` classes. They could be images of cats and dogs, or of a possibly malfunctioning engines (in both cases we would have `k = 2`), or handwritten digits (in which case `k = 10`). We want the learning program to determine the two classes by itself.

- Task: `Task = Image -> {0, 1}`
- Experience: `Experience = List(Image)`
- Performance: `perf : (Task, List(Image)) → ℝ`
  - `perf(t, [img₁, ..., imgₙ]) = sum [correct(t, img₁), ..., correct(t, img₂)] / n`
  - the function `correct : (Task, List(Image)) → {0, 1}` is a "hypothetical" function, since we might not know the correct classification (as in data mining).

A typical situation in which this type of learning can be achieved is when the data can be seen as having been generated from `k` "ideal" exemplars, distorted in some way. Unsupervised learning can then be seen as an attempt to reconstruct these exemplars from the data.

The data, and hence the ideal exemplars, usually comes in the form of tuples of fixed length (the **features**):

$$\xi = (\xi_1, \ldots, \xi_m) \in (\Omega_1, \ldots, \Omega_m), \Omega_i \subseteq \mathbb{R}$$

We can model the "distortion" of the data by assuming that the data is obtained by perturbing the various features with a "noise":

$$x = (x_1, \ldots, x_m) = (\xi_1 + \varepsilon_1, \ldots, \xi_m + \varepsilon_m)$$

where ε_j is a random variable with mean 0.

Thus, if we only had one exemplar, we could reconstruct it by estimating the mean value of the data.

Statistics

Definitions:

- **random variable**
- **mean**
- **standard deviation**

The reason the standard deviation is an interesting measure, is that, no matter what the probability distribution has generated the data, samples are "unlikely" to be "many" standard deviations away from the mean. The following remarkable result formalises this intuition:

Theorem (Chebyshev)

Example:

Exercise:

The normal distribution

It is well-known that the normal distribution is ubiquitous. If the data is normally distributed, then we can "tighten" the Chebyshev bounds considerably.

Example:

Exercise:

The central limit theorem

The central limit theorem shows that means of samples are normally distributed around the real mean. Hence, we can use the tighter bounds for estimates.

Theorem:

Example:

Exercise:

Self-organising maps

EM algorithm

The self-organising maps algorithm assigns a class to a data point based on the distance to the centres of the clusters. However, this is not always the optimal assignment. Consider the following example:

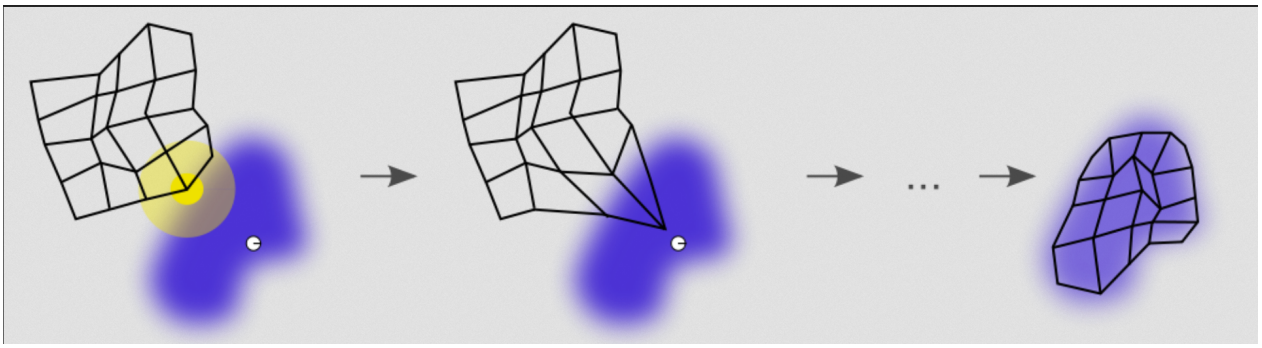

Figure 1: Self-organising map

Example: one dimensional data drawn from two normal distributions, one with very small dispersion, the other with big dispersion.

This example shows that if we have a model of the probability distributions that distort the exemplars, we can do better.

The EM algorithm is designed to obtain the maximum likelihood values for the probability distributions that generate the data. In order to explain the algorithm, we first consider its application in a simple case.

Example (EM algorithm):

Derivation of the EM algorithm

EM algorithm