Image Analysis - Lecture 7

Energy and Graph based Segmentation

Kalle Åström

September 15, 2016

Lecture 7

Contents

Clustering

Goal: Partition a set on n feature vectors with d components

$$x_1, \ldots x_n$$

i.e. $x_i \in \mathbb{R}^d$, in groups (clusters) such that all examples in the same group are similar.

Classification

Goal: Given a number of examples, that already are partitioned into groups

$$(x_1, f_1), \ldots (x_n, f_n)$$

construct a function

$$R : x \longrightarrow f$$

such that $R(x_i) = f_i$ as good as possible
Here $x_i \in \mathbb{R}^n$ are examples and $f_i \in \mathbb{Z}$ is a number representing, which class/group it belongs to.

The Clustering Problem

Input: n examples $x_1, \ldots x_n$.
Output: A mapping

$$c : \{1, \ldots, n\} \mapsto \{1, \ldots k\}.$$

Example: **K-means algorithm**
Choose k cluster centres $m_1, \ldots, m_k$ and a clustering function c that minimises

$$f(c, m) = \sum_{i=1}^{n} |x_i - m_{c(i)}|^2.$$

K-means

The problem

$$\min_{c,m} f(c,m) = \min_{c,m} \sum_{i=1}^{n} |x_i - m_{c(i)}|^2$$

is a non-linear optimization problem that can be solved with many methods. However, most only give a local optima.

K-means implementation

One popular implementation is the following

1. Randomly choose k cluster centra (e.g. k examples)
2. **Optimize** c**:** For every example x_i assign $c(i)$ such that $|x_i - m_{c(i)}|$ is minimized, i.e.

$$c = \text{argmin}_c f(c, m).$$

3. **Optimize** m**:** For every group j change m_j as the centre of mass of corresponding examples, i.e.

$$m = \text{argmin}_m f(c, m).$$

Hierarchical methods

A popular clustering method is hierarchical clustering.
Start with one cluster or regard each point as a separate cluster.
Reduce one after one or merge one after one.
Stop when you are finished.

1. Start with n clusters
2. Choose error criteria, e.g. $f(c, m)$ as above
3. Merge the two clusters that minimizes the error criteria (or split the cluster with the biggest error criteria)
4. If the number of clusters is more than k go to 3.

The result is a clustering for every number of clusters, k, between 1 and n. This can often be represented in a so called dendrogram, where you map error criteria and clustering.

There are **many** other methods

- EM-method (Similar to k-means, but better (and slower))
- k-mediods
- adaptive resonance theory
- fuzzy clustering
- auto-class
- mode-separator
- Self Organizing Maps
- Agglomerative hierarchical clustering
- Divisive hierarchical clustering
- Iso-map

Graph Theory

A graph $G = (V, E)$ consists of vertices(nodes) V and edges E.
Every edge connects two vertices.
In a directed graph, every edge has an orientation.
In a weighted graph, every edge has a weight (a number).
A graph is connected if one can 'walk' between all pairs of vertices through one or several edges.
Every graph can be split into a disjoint set of connected components.

Matrix representation

Weighted graphs can be represented as a matrix. A weighted edge between vertex i and vertex j with v is represented by matrix element (i, j).
For un-directed graphs, half the weight is put at position (i, j) and half in (j, i).
Connected components - blocks in block diagonal matrices.

Affinity measures

When solving clustering problems with graph theoretical methods one need a closeness measure $v_{i,j}$, for every pair of nodes (i, j). A large number means that they are close. A small number means that they are different.
The affinity measure depends on which problem one has. Usual ingredients are

- Distance - e.g. $aff(x, y) = e^{-(x-y)^T(x-y)/(2\sigma_d^2)}$
- Intensity - e.g. $aff(x, y) = e^{-(I(x)-I(y))^T(I(x)-I(y))/(2\sigma_I^2)}$
- Color - e.g. $aff(x, y) = e^{-dist(c(x),c(y))^2/(2\sigma_c^2)}$
- Texture - e.g. $aff(x, y) = e^{-(f(x)-f(y))^T(f(x)-f(y))/(2\sigma_f^2)}$

Eigenvectors and segmentation

Assume that w_n is a vector of ones for those elements that belong to a particular cluster and zeros otherwise. Then the sum of all weights for edges within a cluster is

$$w_n^T A w_n.$$

By maximizing $w_n^T A w_n$ with the constraint $w_n^T w_n = 1$ one might argue that we maximize clustering.
Maxima with this problem corresponds to stationary points of the Lagrange function.

Eigenvectors and segmentation

Maximize $w_n^T A w_n$ with constraint $w_n^T w_n = 1$.
Study the Lagrange function

$$L(w_n, \lambda) = w_n^T A w_n + \lambda(w_n^T w_n - 1).$$

Differentiate and divide with two gives

$$A w_n = -\lambda w_n.$$

This is an eigenvalue problem.

Cut-methods for segmentation

- Popular method: Normalized cuts. Similar to previos graph method, but objective is modified to avoid small clusters. See Section 14.5.5. Works poorly sometimes.
- Even more popular method: Graph cuts. Works very well. See "Graph cuts homepage" on the internet for recent applications and tutorials. Next time.
- Very efficient methods exist to find global minima for graph cuts

Masters thesis suggestion of the day: Cheese production

When producing cheese with large holes (e.g. 'Grevé) sometimes there are large cracks in the cheese. These are more frequent at certain periods of the year, but no-one knows why. The masters thesis project is aimed at developing a system for automatic cheese analysis from images that can be used at the factories in Kristianstad and Umeå, where cheeses are stored and cut into pieces.

Review - Lecture 7

- Clustering
  - K-means algorithm
  - Other clustering methods
  - Segmentation using clustering
- Energy based segmentation
  - The Mumford-Shah functional
  - Two-phase Mumford-Shah
  - Statistical interpretation
- Graph cuts
  - Optimization tool with many applications