Pattern Recognition Lecture "Clustering: Schemes Based on Function Optimisation"

Prof. Dr. Marcin Grzegorzek

Research Group for Pattern Recognition
www.pr.informatik.uni-siegen.de

Institute for Vision and Graphics
University of Siegen, Germany

Overview

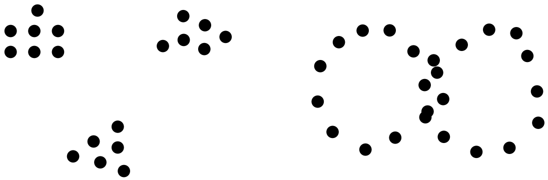
Overview

Introduction (1)

- One of the commonly used families of clustering schemes relies on the optimisation of a cost function J using differential calculus techniques.
- The cost function is a function of the vectors from X and it is parameterised in terms of an unknown parameter vector $\boldsymbol{\theta}$.
- For most of the schemes of the family, the number of clusters, m, is assumed to be known.

Introduction (2)

- Our goal is the estimation of $\boldsymbol{\theta}$ that characterises best the clusters underlying X.
- The parameter vector $\boldsymbol{\theta}$ is strongly dependent on the shape of the clusters.



(a) (b)

- For compact clusters (a) it is reasonable to adopt as parameters a set of m points, $\mathbf{m}_i$, corresponding to the clusters.
- If ring-shaped clusters are expected (b), it is reasonable to use m hyperspheres $C(\mathbf{c}_i, r_i)$ as representatives.

Introduction (3)

- Cluster representatives in this lecture are computed using all the vectors of X, and not only vectors assigned to a particular cluster.
- In this lecture, a basic idea of the mixture decomposition models will be presented and we will take a deeper look into the fuzzy clustering algorithms.
- For the mixture decomposition models, the cost function is constructed on the basis of random vectors, and assignment to clusters follows probabilistic arguments in the spirit of the Bayesian classification.
- In the fuzzy approach a proximity function between a vector and a cluster is defined, and the "grade of membership" of a vector in a cluster is provided by the set of membership functions.

Overview

Basic Idea

- The number of clusters, m, expected in the data X is given.
- Each vector, $\mathbf{x}_i; i = 1, \ldots, N$, belongs to a cluster C_j with a probability $P(C_j|\mathbf{x}_i)$.
- A vector $\mathbf{x}_i$ is appointed to the cluster C_j if

$$P(C_j|\mathbf{x}_i) > P(C_k|\mathbf{x}_i); \qquad k = 1, \ldots, m;\ k \neq j$$

Differences to the Supervised Bayes Classifier

- The differences to the supervised Bayesian classification approach are that (a) no training data with known cluster labelling are available and (b) the a priori probabilities $P(C_j) \equiv P_j$ are not known either.

- It can be considered as a “supervised” task with an incomplete training data set.

- We are missing the corresponding cluster labelling information for each data point $\mathbf{x}_i$.

Overview

Basic Idea

- In fuzzy schemes a vector belongs simultaneously to more than one cluster with different "grades of membership".

- A fuzzy m-clustering of X is defined by a set of functions $u_j : X \to A; j = 1, \ldots, m$, where $A = [0, 1]$. This can be also written as

$$u_j(\mathbf{x}_i) = u_{ij} \in A$$

  u_{ij} denotes the "grade of membership" of vector $\mathbf{x}_i$ to the cluster C_j.

- In the case where $A = \{0, 1\}$, a hard m-clustering of X is defined. In this case, each vector belongs exclusively to a single cluster.

Notation

- $\boldsymbol{\theta}_j$ is the parametrised representative of the j-th cluster. A particular clustering result is represented by a vector containing parametrised representatives of all clusters

$$\boldsymbol{\theta} \equiv [\boldsymbol{\theta}^{\mathsf{T}}{}_1, \ldots, \boldsymbol{\theta}^{\mathsf{T}}{}_m]^{\mathsf{T}}$$

- $\mathbf{U}$ is an $N \times m$ matrix whose (i, j) element equals $u_{ij} = u_j(\mathbf{x}_i)$.

- $d(\mathbf{x}_i, \boldsymbol{\theta}_j)$ is the dissimilarity between $\mathbf{x}_i$ and $\boldsymbol{\theta}_j$.

- q is a parameter called a fuzzifier.

Minimising a Cost Function (1)

- Most of the well-known fuzzy clustering algorithms are those derived by minimising a cost function of the form

$$J_q(\boldsymbol{\theta}, \mathbf{U}) = \sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}^q d(\mathbf{x}_i, \boldsymbol{\theta}_j)$$

with respect to $\boldsymbol{\theta}$ and $\mathbf{U}$, subject to following constraints

$$\sum_{j=1}^{m} u_{ij} = 1, \quad i = 1, \ldots, N$$

$$u_{ij} \in [0, 1], \quad i = 1, \ldots, N, \quad j = 1, \ldots, m$$

$$0 < \sum_{i=1}^{N} u_{ij} < N, \quad j = 1, 2, \ldots, m$$

Minimising a Cost Function (2)

- In other words, the grade of membership of $\mathbf{x}_i$ in the j-th cluster is related to the grade of membership of $\mathbf{x}_i$ to the remaining $m-1$ clusters.
- Different values of q bias $J_q(\boldsymbol{\theta}, \mathbf{U})$ toward either the fuzzy or the hard clusterings.
- More specifically, for fixed $\boldsymbol{\theta}$, if $q = 1$, no fuzzy clustering is better than the best hard clustering in terms of $J_q(\boldsymbol{\theta}, \mathbf{U})$. However, if $q > 1$, there are cases in which fuzzy clusterings lead to lower values of $J_q(\boldsymbol{\theta}, \mathbf{U})$ than the best hard clustering.

Cost Function Minimisation with Respect to $\mathbf{U}$

$$\frac{\partial J_q(\boldsymbol{\theta}, \mathbf{U})}{\partial \mathbf{U}} = 0$$

$$\Downarrow$$

Lots of mathematics brings us to the formula for computing the elements of the matrix $\mathbf{U}$ ($u_{rs}, r = 1, \ldots, N, s = 1, \ldots m$)

$$\Downarrow$$

$$u_{rs} = \frac{1}{\sum_{j=1}^{m} \left(\frac{d(\mathbf{x}_r, \boldsymbol{\theta}_s)}{d(\mathbf{x}_r, \boldsymbol{\theta}_j)} \right)^{\frac{1}{q-1}}} \tag{1}$$

Cost Function Minimisation with Respect to $\boldsymbol{\theta}$

- Taking a gradient of $J_q(\boldsymbol{\theta}, \mathbf{U})$ with respect to $\boldsymbol{\theta}_j$ and setting it equal to zero, we obtain

$$\frac{\partial J_q(\boldsymbol{\theta}, \mathbf{U})}{\partial \boldsymbol{\theta}_j} = \sum_{i=1}^{N} u_{ij}^q \frac{\partial d(\mathbf{x}_i, \boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = 0, \quad j = 1, \ldots, m \qquad (2)$$

- Equations (1) and (2) are coupled and, in general, cannot give closed-form solutions. One way to proceed is to employ the following iterative algorithmic scheme (next slide), in order to obtain estimates for $\mathbf{U}$ and $\boldsymbol{\theta}$.

Generalised Fuzzy Algorithmic Scheme (GFAS)

- Choose $\boldsymbol{\theta}_j(0)$ as initial estimates for $\boldsymbol{\theta}_j$; $\quad j = 1, \ldots, m$.
- $t = 0$
- Repeat
  - For $i = 1$ to N
    - For $j = 1$ to m $\quad -u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\mathbf{x}_i, \boldsymbol{\theta}_j(t))}{d(\mathbf{x}_i, \boldsymbol{\theta}_k(t))} \right)^{\frac{1}{q-1}}}$
    - End {For-j}
  - End {For-i}
  - $t = t + 1$
  - For $j = 1$ to m
    - Solve: $\quad \sum_{i=1}^{N} u_{ij}^{q}(t-1) \frac{\partial d(\mathbf{x}_i, \boldsymbol{\theta}_j)}{\partial \boldsymbol{\theta}_j} = 0$ with respect to $\boldsymbol{\theta}_j$ and set $\boldsymbol{\theta}_j(t)$ equal to this solution.
  - End {For-j}
- Until a termination criterion is met.

GFAS - Final Remarks

- The algorithmic scheme may also be initialised from $\mathbf{U}(0)$ instead of $\boldsymbol{\theta}_j(0), j = 1, \ldots, m$, and start iterations with computing $\boldsymbol{\theta}_j$ first.

- The above iterative algorithmic scheme is also known as the alternating optimisation (AO) scheme, since at each iteration step $\mathbf{U}$ is updated for fixed $\boldsymbol{\theta}$, and then $\boldsymbol{\theta}$ is updated for fixed $\mathbf{U}$.