CSCE 970 Lecture 10: Clustering Algorithms Based on Cost Function Optimization

Stephen D. Scott

April 9, 2001

Introduction

- A very popular family of clustering algorithms

- General procedure: Define a cost function J measuring the goodness of a clustering and search for a parameter vector $\boldsymbol{\theta}$ to minimize/maximize J

- Search is subject to certain constraints, e.g. that the result fits the definition of a clustering (slide 7.8)

- E.g. $\boldsymbol{\theta} = \left[\mathbf{m}_1^T, \mathbf{m}_2^T, \ldots, \mathbf{m}_m^T\right]^T$ = the point representatives of the m clusters

- $\boldsymbol{\theta}$ can also be a vector of parameters for m hyperplanar or hyperspherical representatives

- Typically algorithms assume m is known, so may have to try several values in practice

- Skipping Bayesian-style approaches (Sec. 14.2) and focusing on hard c-means (Isodata) and fuzzy c-means algorithms

Hard Clustering Algorithms
Definitions

- Let U be an $N \times m$ matrix whose (i,j)th entry u_{ij} is 1 if f.v. $\mathbf{x}_i$ is in cluster C_j and 0 otherwise

- To meet definition of hard clustering, $\forall i \in \{1, \ldots, N\}$ need $\sum_{j=1}^{m} u_{ij} = 1$ and $u_{ij} \in \{0, 1\}$

- Let $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m]$ be an m-vector of representatives of the m clusters

- Use cost function

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

- Globally minimizing J given a set of f.v.'s means finding both a set of representatives $\boldsymbol{\theta}$ and assignment to clusters U that minimizes DM between each f.v. and its representative

Hard Clustering Algorithms
Minimizing J

- For a fixed $\boldsymbol{\theta}$, J is minimized and constraints met iff

$$u_{ij} = \begin{cases} 1 & \text{if } d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right) = \min_{1 \le k \le m} \left\{ d\left(\mathbf{x}_i, \boldsymbol{\theta}_k\right) \right\} \\ 0 & \text{otherwise} \end{cases}$$

- For a fixed U, minimize J by minimizing w.r.t. each $\boldsymbol{\theta}_j$ independently, so for $j \in \{1, \ldots, m\}$, take gradient and set to $\mathbf{0}$ vector:

$$\sum_{i=1}^{N} u_{ij} \frac{\partial d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}{\partial \boldsymbol{\theta}_j} = \mathbf{0}$$

- Can alternate between the above steps until a termination condition is met

Hard Clustering Algorithms

Generalized Hard Algorithmic Scheme (GHAS)

- Initialize $t = 0$, $\boldsymbol{\theta}_j(t)$ for $j = 1, \ldots, m$

- Repeat until termination condition met

  - For $i = 1$ to N (assign f.v.'s to clusters)

    * For $j = 1$ to m

$$u_{ij}(t) = \begin{cases} 1 & \text{if } d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right) = \min_{1 \le k \le m} \left\{ d\left(\mathbf{x}_i, \boldsymbol{\theta}_k\right) \right\} \\ 0 & \text{otherwise} \end{cases}$$

  - $t = t + 1$

  - For $j = 1$ to m, solve

$$\sum_{i=1}^{N} u_{ij}(t-1) \frac{\partial d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}{\partial \boldsymbol{\theta}_j} = \mathbf{0} \qquad (1)$$

    for $\boldsymbol{\theta}_j$ and set $\boldsymbol{\theta}_j(t)$ equal to it

- Example termination condition: Stop when $\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)\| < \epsilon$, where $\|\cdot\|$ is any vector norm and ϵ is user-specified

Isodata (a.k.a. Hard k-Means a.k.a. Hard c-Means) Algorithm

- Special case of GHAS: Each $\boldsymbol{\theta}_j$ is point rep. of C_j and DM is squared Euclidean distance

- So (1) becomes

$$\sum_{i=1}^{N} u_{ij}(t-1) \frac{\partial \left(\sum_{k=1}^{\ell} \left(x_{ik} - \theta_{jk} \right)^2 \right)}{\partial \boldsymbol{\theta}_j} = \mathbf{0}$$

$$\Downarrow$$

$$\sum_{i=1}^{N} u_{ij}(t-1) \begin{bmatrix} 2 \left(\theta_{j1} - x_{i1} \right) \\ \vdots \\ 2 \left(\theta_{j\ell} - x_{i\ell} \right) \end{bmatrix} = \mathbf{0}$$

$$\Downarrow$$

$$2\boldsymbol{\theta}_j \sum_{i=1}^{N} u_{ij}(t-1) = 2 \sum_{i=1}^{N} u_{ij}(t-1)\mathbf{x}_i$$

$$\Downarrow$$

$$\boxed{\boldsymbol{\theta}_j = \frac{1}{|C_j(t-1)|} \sum_{\mathbf{x}_i \in C_j(t-1)} \mathbf{x}_i = C_j(t-1)\text{'s mean vector}}$$

Isodata Algorithm

Pseudocode

- Initialize $t = 0$, $\boldsymbol{\theta}_j(t)$ for $j = 1, \ldots, m$

- Repeat until $\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)\| = 0$

  - For $i = 1$ to N

    * Find closest rep. for $\mathbf{x}_i$, say $\boldsymbol{\theta}_j$, and set $b(i) = j$

  - For $j = 1$ to m

    * Set $\boldsymbol{\theta}_j =$ mean of $\{\mathbf{x}_i \in X : b(i) = j\}$

- Guaranteed to converge to global minimum of J if squared Euclidean distance used

- If e.g. Euclidean distance used, cannot guarantee this

Fuzzy Clustering Algorithms

Definitions

- Let U be an $N \times m$ matrix whose (i, j)th entry u_{ij} quantifies the fuzzy membership of $\mathbf{x}_i$ in cluster C_j

- To meet definition of fuzzy clustering, $\forall i \in \{1, \ldots, N\}$ need $\sum_{j=1}^{m} u_{ij} = 1$, $\forall j \in \{1, \ldots, m\}$ need $0 < \sum_{i=1}^{N} u_{ij} < N$ and $\forall i, j$ need $u_{ij} \in [0, 1]$

- Let $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_m]$ be an m-vector of representatives of the m clusters

- Use cost function

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

  where q is a fuzzifier that, when > 1, allows for fuzzy clusterings to have lower cost than hard clusterings (Example 14.4, pp. 453–454)

Fuzzy Clustering Algorithms

Minimizing J_q

- When fixing $\boldsymbol{\theta}$ and minimizing w.r.t. U while satisfying constraints, cannot use simple method from slide 4

- Instead, use Lagrange multipliers (pp. 610–611) to enforce constraint $\sum_{j=1}^{m} u_{ij} = 1 \, \forall i$

$$\mathcal{J}(\boldsymbol{\theta}, U) = \overbrace{\sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}^{q} \, d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}^{J_q} - \overbrace{\sum_{i=1}^{N} \lambda_i \left(\sum_{j=1}^{m} u_{ij} - 1 \right)}^{=0 \text{ when } \sum_{j=1}^{m} u_{ij} = 1 \, \forall i}$$

- Now minimize $\mathcal{J}$ w.r.t. U, yielding update equation in terms of λ_i's, solve for λ_i's using constraint, and end up with final update equation for U

Fuzzy Clustering Algorithms

Minimizing J_q (cont'd)

$$\frac{\partial \mathcal{J}(\boldsymbol{\theta}, U)}{\partial u_{rs}} = q\, u_{rs}^{q-1}\, d(\mathbf{x}_r, \boldsymbol{\theta}_s) - \lambda_r = 0$$

$$\Downarrow$$

$$u_{rs} = \left(\frac{\lambda_r}{q\, d(\mathbf{x}_r, \boldsymbol{\theta}_s)} \right)^{1/(q-1)}, \; s = 1, \ldots, m$$

$$\Downarrow \; \left(\text{use constraint } \sum_{j=1}^{m} u_{rj} = 1 \right)$$

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

$$\Downarrow$$

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

$$\Downarrow$$

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

Fuzzy Clustering Algorithms

Minimizing J_q (cont'd)

- For a fixed U, minimize J_q by minimizing w.r.t. each $\boldsymbol{\theta}_j$ independently, so for $j \in \{1, \ldots, m\}$, take gradient and set to $\mathbf{0}$ vector:

$$\frac{\partial J(\boldsymbol{\theta}, U)}{\partial \boldsymbol{\theta}_j} = \sum_{i=1}^{N} u_{ij}^{q} \frac{\partial d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}{\partial \boldsymbol{\theta}_j} = \mathbf{0}$$

- As with hard clustering scheme, alternate between U and $\boldsymbol{\theta}$ until a termination condition is met

Fuzzy Clustering Algorithms

Generalized Fuzzy Algorithmic Scheme (GFAS)

- Initialize $t = 0$, $\boldsymbol{\theta}_j(t)$ for $j = 1, \ldots, m$

- Repeat until termination condition met

  - For $i = 1$ to N (assign memb. values to f.v.'s)

    * For $j = 1$ to m

$$u_{ij}(t) = \frac{1}{\sum_{k=1}^{m} \left(\frac{d(\mathbf{x}_i, \boldsymbol{\theta}_j)}{d(\mathbf{x}_i, \boldsymbol{\theta}_k)} \right)^{1/(q-1)}}$$

  - $t = t + 1$

  - For $j = 1$ to m, solve

$$\sum_{i=1}^{N} u_{ij}^{q}(t-1) \frac{\partial d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}{\partial \boldsymbol{\theta}_j} = \boxed{\mathbf{0}} \qquad (2)$$

    for $\boldsymbol{\theta}_j$ and set $\boldsymbol{\theta}_j(t)$ equal to it

- Example termination condition: Stop when $\|\boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)\| < \epsilon$, where $\| \cdot \|$ is any vector norm and ϵ is user-specified

Fuzzy c-Means (a.k.a. Fuzzy k-Means) Algorithm

- If we use GFAS with $\boldsymbol{\theta}_j$ = point rep. of C_j and DM = squared Euclidean dist., (2) becomes

$$2 \sum_{i=1}^{N} u_{ij}^{q}(t-1) \left(\boldsymbol{\theta}_j - \mathbf{x}_i\right) = \mathbf{0}$$

$$\Downarrow$$

$$\boldsymbol{\theta}_j(t) = \frac{\sum_{i=1}^{N} u_{ij}^{q}(t-1)\, \mathbf{x}_i}{\sum_{i=1}^{N} u_{ij}^{q}(t-1)} \qquad (3)$$

- Convergence guarantees?

- Instead of squared Euclidean distance, can use

$$d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right) = \left(\mathbf{x}_i - \boldsymbol{\theta}_j\right)^T A \left(\mathbf{x}_i - \boldsymbol{\theta}_j\right), \qquad (4)$$

  where A is symmetric and positive definite

- Use of (4) doesn't change (3)

Possibilistic Clustering Algorithms

- Similar to fuzzy clustering algorithms, but don't require $\sum_{j=1}^{m} u_{ij} = 1 \; \forall i \in \{1, \ldots, N\}$

- Still need $u_{ij} \in [0,1] \; \forall i, j$ and $0 < \sum_{i=1}^{N} u_{ij} \boxed{\leq} N$ $\forall j \in \{1, \ldots, m\}$

- In addition, require some $u_{ij} > 0 \; \forall i \in \{1, \ldots, N\}$

- Instead of measuring degree of membership of $\mathbf{x}_i$ in C_j, now u_{ij} measures degree of compatability, i.e. the possibility that $\mathbf{x}_i$ belongs in C_j

- Cannot directly use fuzzy cost function of slide 8 since it's trivially minimized with $U = 0$, violating our new constraint, so use:

$$J(\boldsymbol{\theta}, U) = \overbrace{\sum_{i=1}^{N} \sum_{j=1}^{m} u_{ij}^{q} \, d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}^{\text{original cost func.}} + \overbrace{\sum_{j=1}^{m} \eta_j \sum_{i=1}^{N} \left(1 - u_{ij}\right)^{q}}^{\text{decreases as } u_{ij}\text{'s increase}}$$

where $\eta_j > 0 \; \forall j$

Possibilistic Clustering Algorithms

Minimizing J

$$\frac{\partial J(\boldsymbol{\theta}, U)}{\partial u_{ij}} = q u_{ij}^{q-1}\, d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right) - q\eta_j \left(1 - u_{ij}\right)^{q-1} = 0$$

$$\Downarrow$$

$$u_{ij} = \frac{1}{1 + \left(\frac{d(\mathbf{x}_i, \boldsymbol{\theta}_j)}{\eta_j}\right)^{1/(q-1)}} \tag{5}$$

- Updating for $\boldsymbol{\theta}$ is same as for GFAS:

$$\sum_{i=1}^{N} u_{ij}^{q}(t-1)\, \frac{\partial d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}{\partial \boldsymbol{\theta}_j} = \mathbf{0}$$

- Setting η_j's can be done by running GFAS then taking a weighted average of DM between $\mathbf{x}_i$'s and $\boldsymbol{\theta}_j$:

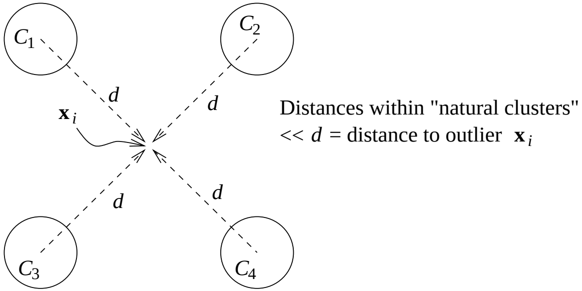
$$\eta_j = \frac{\sum_{i=1}^{N} u_{ij}^{q}\, d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)}{\sum_{i=1}^{N} u_{ij}^{q}}$$

[then run GPAS after setting η_j's]

Possibilistic Clustering Algorithms

Benefit: Less Sensitivity to Outliers

- Since u_{ij} is inversely proportional to $d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)$, updates to $\boldsymbol{\theta}$ are less sensitive to outliers



- For GHAS (slide 5), $u_{ij} = 1$ for one cluster, 0 for others

- For GFAS (slide 12), $u_{ij} = 1/m = 1/4$ for each cluster

- For GPAS, u_{ij} gets arbitrarily small as d grows, and is independent of distances to other $\boldsymbol{\theta}_k$'s

Possibilistic Clustering Algorithms

Benefit: Mode-Seeking Property

- Can write cost function as $J(\boldsymbol{\theta}, U) = \sum_{j=1}^{m} J_j$ for

$$J_j = \sum_{i=1}^{N} u_{ij}^q \, d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right) + \eta_j \sum_{i=1}^{N} \left(1 - u_{ij}\right)^q \quad (6)$$

  and get same u_{ij} updates by minimizing J_j's individually

- Rewriting (5) as $d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right) = \eta_j \left(\frac{1 - u_{ij}}{u_{ij}}\right)^{q-1}$ and substituting into (6) gives

$$J_j = \sum_{i=1}^{N} u_{ij}^q \eta_j \left(\frac{1 - u_{ij}}{u_{ij}}\right)^{q-1} + \eta_j \sum_{i=1}^{N} \left(1 - u_{ij}\right)^q$$

$$= \eta_j \sum_{i=1}^{N} \left(1 - u_{ij}\right)^{q-1} \left(u_{ij} + 1 - u_{ij}\right) = \eta_j \sum_{i=1}^{N} \left(1 - u_{ij}\right)^{q-1}$$

- So minimizing J $\Rightarrow$ minimizing J_j's $\Rightarrow$ maximizing u_{ij}'s $\Rightarrow$ minimizing $d\left(\mathbf{x}_i, \boldsymbol{\theta}_j\right)$ for each j

Possibilistic Clustering Algorithms

Benefit: Mode-Seeking Property (cont'd)

- Thus GPAS seeks out regions dense in f.v.'s, i.e. if $m > k =$ number of natural clusters, then properly initialized GPAS will have coincident clusters

- So m need not be known a priori, only upper bound and proper initialization