Lecture 13: Unsupervised Learning
Cognitive Systems - Machine Learning

Part IV: Further Topics

k-means clustering, hierarchical clustering, competitive learning, SOMs

last change January 27, 2014
Motivation

- Unsupervised learning: trying to find hidden structure in unlabeled data
- Knowledge Discovery
- Data can be categorized based on some similarity measure
- Typical domains of application
  - Data mining (categorize unlabelled data)
  - Data compression (represent data sets by their prototype)
  - Categorize data when labeling is costly, e.g. in speech recognition
Similarity Measures

- On metrical data (feature vectors): typically distance metrics
- On categorial features: contrast measures
- On structured data: extract features OR use structural similarity measures such as edit distance

Euklid:

\[ d(\vec{x}_i, \vec{x}_j) = \left( \sum_{l=1}^{n} (x_{il} - x_{jl})^2 \right)^{\frac{1}{2}} \]

City Block:

\[ d(\vec{x}_i, \vec{x}_j) = \sum_{l=1}^{n} (x_{il} - x_{jl}) \]
Distance Metric

Distance Function/Measure
- Non-negativity: $d(x, y) \geq 0$
- Identity: $d(x, y) = 0$ iff $x = y$
- Symmetry: $d(x, y) = d(y, x)$

Metric
- Triangle equality (subadditivity): $d(x, y) \leq d(x, z) + d(z, y)$

Ultra-Metric: $d(x, y) \leq \max(d(x, z), d(y, z))$

Similarity Function/Measure
- $s(x, y) \leq 1$
- Identity: $s(x, y) = 1$ iff $x = y$
- Symmetry: $s(x, y) = s(y, x)$

Metric
- $s(x, y) \geq s(x, z) \times s(z, y)$

Ultra-Metric: $s(x, y) \geq \min(s(x, z), s(y, z))$
Minkowski-Metric

\[ d(\vec{x}_i, \vec{x}_j) = \left( \sum_{l=1}^{n} (x_{il} - x_{jl})^m \right)^{\frac{1}{m}} \]

- \( m = 1 \): City Block/Manhattan
- \( m = 2 \): Euklid
- \( m = \infty \): Supremum

City Block: 2 + 5
Supremum: 5
Mahalanobis Distance

\[ d(\vec{x}, \vec{y}) = \sqrt{ (\vec{x} - \vec{y})^T S^{-1} (\vec{x} - \vec{y}) } \]

- \( S \) is the covariance matrix (symmetric, with variance in the diagonal)
- considers differences in scale (e.g. one feature has min=0 and max=1, another has min=0 and max=100)
- considers correlation of features
- If covariances are 1, Mahalanobis is equivalent to Euclidean distance
Measures for Categorial Variables

Jaccard similarity coefficient:

\[ J(A, B) = \frac{|A \cap B|}{|A \cup B|} \]

Jaccard distance:

\[ d_J(A, B) = 1 - J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \]

Tanimoto similarity for binary variables:

\[ f(A, B) = \frac{A \cdot B}{|A|^2 + |B|^2 - A \cdot B} \]
**k-means Clustering**

Iterative Algorithm:

1. Define the number of clusters $k$
2. Initialize clusters by
   - an arbitrary assignment of examples to clusters or
   - an arbitrary set of cluster centers (examples assigned to nearest centers)
3. Compute the sample mean of each cluster
4. Reassign each example to the cluster with the nearest mean
5. If the classification of all samples has not changed, stop, else go to step 3

Sample mean ($S$ is the set of objects in one cluster)

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Distance between objects in same cluster should be minimal and distance between clusters maximal.
k-means Clustering

Criterion function (sum of squared errors):

$$\arg \min_s \sum_{i=1}^{k} \sum_{x_j \in S_i} \| x_j - \mu_i \|^2$$

- Data points with largest distances greatest influence!
  k-means clustering is vulnerable to outliers!
- Effort: proportional to number of objects times number of clusters
- Danger of local optima: run several times with different starting points
Illustration

Initial Centroids

Initial Partition

Iteration Number 2

Iteration Number 20
Determine Number of Clusters

- Previous knowledge from the application domain
e.g., classify plants in three groups reflecting their vitality
- Determine optimal $K^*$ such that within-cluster distance is “optimal”

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(i')=k} d(\vec{x}_i, \vec{x}_{i'})$$

- Select $K$ with minimal descend from $W_k - W_{k+1}$
Hierarchical Clustering

- Cluster are not on one level but constitute a taxonomy
- Lowest level: Objects, Top-Level: Single Cluster
- Each level contains cluster which subsume two clusters of the level below
- There are top-down and bottom-up (agglomerative) approaches

![Hierarchical Clustering Diagram](image-url)
Clustering of Shakespeare Plays

www.winedarksea.org, Michael Witmore, Published: November 29, 2009
Agglomerative Clustering

- User decides which level classifies data best
- Dis-similarity of clusters guides agglomeration: the two clusters with the least dis-similarity are joined
- Dis-similarity $d$ between clusters $G$ and $H$: distance between objects (e.g. feature vectors) in $G$ and $H$
- Three important measures:
  - single linkage
  - complete linkage
  - average linkage
Single Linkage

\[ D_{SL}(A, B) := \min_{a \in A, b \in B} \{d(a, b)\} \]

- Dis-similarity determined by dis-similarity of nearest points
- Problem: Clusters with large dis-similarities, “chains”
Complete Linkage

\[ D_{CL}(A, B) := \max_{a \in A, b \in B} \{ d(a, b) \} \]

- Dis-similarity determined by dis-similarity of farest points
- Compact clusters, but objects in a cluster might be more similar to objects in another cluster
Average Linkage

\[ D_{AL}(A, B) := \frac{1}{|A||B|} \sum_{a \in A, b \in B} d(a, b) \]

- Dis-similarity determined by dis-similarity of farthest points
- Compact clusters, but objects in a cluster might be more similar to objects in another cluster
Self-organizing Maps (SOMs)

- Proposed by Kohonen, 1995
- An artificial neural net approach for unsupervised learning
- Can be applied for cluster analysis

Structure
- Input layer for $n$ input values
- Completely connected to competitive layer
- In the competitive layer all neurons are inhibitorily connected

Learning
- Cause different parts of the network to respond similarly to certain input patterns
- Competitive learning
- Iterative optimization:
  - Best matching unit (BMU): neuron whose weight vector is most similar to the input
  - Weights of BMU and neighbours are adjusted towards input vector
Example: Voting Behavior

- A study of voting behavior of different countries during the yearly EuroVision Song Contests

---

*E.V. Sanonova et al. / Neural Networks 19 (2006) 935–949*