Unsupervised Learning:

Clustering

Slides adapted from Vibhav Gogate, Carlos Guestrin, Dan Klein & Luke Zettlemoyer

Clustering

Clustering systems:

- Unsupervised learning
- Requires data, but no labels
- Detect patterns eg in
  - Group emails or search results
  - Customer shopping patterns
  - Program executions (intrusion detection)
- Useful when don’t know what you’re looking for
- But: often get gibberish

Clustering

• Basic idea: group together similar instances
• Example: 2D point patterns

What could “similar” mean?
- One option: small (squared) Euclidean distance

\[ \text{dist}(x, y) = (x - y)^T (x - y) = \sum_i (x_i - y_i)^2 \]

Outline

• K-means & Agglomerative Clustering
• Expectation Maximization (EM)
• Principle Component Analysis (PCA)

K-Means

• An iterative clustering algorithm
  - Pick K random points as cluster centers (means)
  - Alternate:
    - Assign data instances to closest cluster center
    - Change the cluster center to the average of its assigned points
  - Stop when no points’ assignments change

K-Means

• An iterative clustering algorithm
  - Pick K random points as cluster centers (means)
  - Alternate:
    - Assign data instances to closest cluster center
    - Change the cluster center to the average of its assigned points
  - Stop when no points’ assignments change
K-means clustering: Example

(a) Pick K random points as cluster centers (means)

Iterative Step 1
• Assign data instances to closest cluster center

Iterative Step 2
• Change the cluster center to the average of the assigned points

• Repeat until convergence
K-means clustering: Example

- Consider the total distance to the means:
  \[ \phi(\{x_i\}, \{a_i\}, \{c_k\}) = \sum_i \text{dist}(x_i, c_{a_i}) \]

- Two stages each iteration:
  1. Update assignments: fix means c, change assignments a
  2. Update means: fix assignments a, change means c

- Co-ordinate Gradient Descent
- Will it converge?
  - Yes!, if you can argue that each update can’t increase \( \Phi \)

Phase I: Update Assignments

- For each point, re-assign to closest mean:
  \[ a_i = \arg\min_k \text{dist}(x_i, c_k) \]

- Can only decrease total distance \( \Phi \):
  \[ \phi(\{x_i\}, \{a_i\}, \{c_k\}) = \sum_i \text{dist}(x_i, c_{a_i}) \]

Phase II: Update Means

- Move each mean to the average of its assigned points:
  \[ c_k = \frac{1}{|\{i : a_i = k\}|} \sum_{i : a_i = k} x_i \]

- Also can only decrease total distance… (Why?)

- Fun fact: the point y with minimum squared Euclidean distance to a set of points \( \{x\} \) is their mean

Initialization

- K-means is non-deterministic
  - Requires initial means
  - It does matter what you pick!
  - What can go wrong?
  - Various schemes for preventing this kind of thing: variance-based split / merge, initialization heuristics

K-Means Getting Stuck

A local optimum:
**K-Means Questions**

- Will K-means converge?
  - To a global optimum?

- Will it always find the true patterns in the data?
  - If the patterns are very very clear?

- Runtime?

- Do people ever use it?

- How many clusters to pick?

**Agglomerative Clustering**

- How should we define “closest” for clusters with multiple elements?

- How should we define “closest” for clusters with multiple elements?

- Many options:
  - Closest pair (single-link clustering)
  - Farthest pair (complete-link clustering)
  - Average of all pairs
  - Ward’s method (min variance, like k-means)

- Different choices create different clustering behaviors

**Agglomerative Clustering**

- Agglomerative clustering:
  - First merge very similar instances
  - Incrementally build larger clusters out of smaller clusters

- Algorithm:
  - Maintain a set of clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there’s only one cluster left

- Produces not one clustering, but a family of clusterings represented by a dendrogram
Clustering Behavior

- **Average**
- **Farthest**
- **Nearest**

Mouse tumor data from [Hastie]

Agglomerative Clustering Questions

- Will agglomerative clustering converge?
  - To a global optimum?
- Will it always find the true patterns in the data?
- Do people ever use it?
- How many clusters to pick?

Soft Clustering

- Clustering typically assumes that each instance is given a “hard” assignment to exactly one cluster.
- Does not allow uncertainty in class membership or for an instance to belong to more than one cluster.
- **Soft clustering** gives probabilities that an instance belongs to each of a set of clusters.
- Each instance is assigned a probability distribution across a set of discovered categories (probabilities of all categories must sum to 1).

Expectation Maximization (EM)

- Probabilistic method for soft clustering.
- Direct method that assumes $k$ clusters: $\{c_1, c_2, \ldots, c_k\}$
- Soft version of $k$-means.
- Assumes a probabilistic model of categories that allows computing $P(c_i \mid E)$ for each category, $c_i$, for a given example, $E$.
- For text, typically assume a naïve-Bayes category model.
  - Parameters $\theta = \{P(c_i), P(w_j \mid c_i) : i \in \{1, \ldots, k\}, j \in \{1, \ldots, |V|\}\}$

EM Algorithm

- Iterative method for learning probabilistic categorization model from unsupervised data.
- Initially assume random assignment of examples to categories.
- Learn an initial probabilistic model by estimating model parameters $\theta$ from this randomly labeled data.
- Iterate following two steps until convergence:
  - **Expectation (E-step):** Compute $P(c_i \mid E)$ for each example given the current model, and probabilistically re-label the examples based on these posterior probability estimates.
  - **Maximization (M-step):** Re-estimate the model parameters, $\theta$, from the probabilistically re-labeled data.

Reconsidering “hard assignments”?

- Clusters may overlap
- Some clusters may be “wider” than others
- Distances can be deceiving!
Acknowledgements

• K-means & Gaussian mixture models presentation contains material from excellent tutorial by Andrew Moore:
  – http://www.autonlab.org/tutorials/
• K-means Applet:
  – http://www.elet.polimi.it/upload/matteucc/Clustering/tutorial_html/AppletKM.html
• Gaussian mixture models Applet:
  – http://www.neurosci.aist.go.jp/%7Eakaho/MixtureEM.html

Dimensionality Reduction

• Given data points in $d$ dimensions
• Convert them to data points in $r < d$ dimensions
• With minimal loss of information

Principal Component Analysis

Goal: Find $r$-dim projection that best preserves variance
1. Compute mean vector $\mu$ and covariance matrix $\Sigma$
of original points
2. Compute eigenvectors and eigenvalues of $\Sigma$
3. Select top $r$ eigenvectors
4. Project points onto subspace spanned by them:
   \[
   y = A(x - \mu)
   \]
   where $y$ is the new point, $x$ is the old one, and
   the rows of $A$ are the eigenvectors

Multidimensional Scaling

Goal: Find projection that best preserves inter-point distances
$x_i$ Point in $d$ dimensions
$y_i$ Corresponding point in $r < d$ dimensions
$d_{ij}$ Distance between $x_i$ and $x_j$
$d_{ij}$ Distance between $y_i$ and $y_j$

• Define (e.g.) $E(y) = \sum_{i \neq j} \left( \frac{d_{ij} - d_{ij}}{\delta_{ij}} \right)^2$
• Find $y_i$’s that minimize $E$ by gradient descent
• Invariant to translations, rotations and scalings

Isomap

Goal: Find projection onto nonlinear manifold
1. Construct neighborhood graph $G$:
   For all $x_i, x_j$
   If distance($x_i, x_j$) $< \epsilon$
   Then add edge ($x_i, x_j$) to $G$
2. Compute shortest distances along graph $\delta_G(x_i, x_j)$
   (e.g., by Floyd’s algorithm)
3. Apply multidimensional scaling to $\delta_G(x_i, x_j)$