Clustering

Instructor: Jessica Wu -- Harvey Mudd College

The instructor gratefully acknowledges Andrew Ng (Stanford), Eric Eaton (UPenn), David Kauchak (Pomona), David Sontag (NYU), Piyush Rai (Uath), and the many others who made their course materials freely available online.

Robot Image Credit: Viktoriya Sukhanova © 123RF.com

Clustering
Learning Goals

• Describe goal of clustering
• Describe common applications of clustering
Clustering

Unsupervised learning technique that detects patterns

- Informally: find natural groups among objects
- Formally: organize data into clusters such that there is
  - High intra-cluster similarity
  - Low inter-cluster similarity

Most frequently, when people think of unsupervised learning, they think of clustering

- Useful if you do not know what you are looking for
- But can produce gibberish

Gene Expression Data

Goal: Identify groups of genes with similar expression profiles

Cluster analysis and display of genome-wide expression patterns
MB. Eaton, PT. Spellman, PD. Brown. - Proceedings of the ... 1999 - National Acad Sciences
Abstract A system of cluster analysis for genome-wide expression data from DNA microarray
hybridization is described that uses standard statistical algorithms to arrange genes
according to similarity in pattern of gene expression. The output is displayed graphically,
conveying the clustering and the underlying expression data simultaneously in a form
intuitive for biologists. We have found in the budding yeast Saccharomyces cerevisiae that ...
Cited by 10411 Related articles All 128 versions Web of Science: 10393 Cite Save
Face Clustering

Goal: Identify similar faces

Image Segmentation

Goal: Break up image into meaningful or perceptually similar regions
Social Graphs

Goal: Identify groups within Facebook friends.

Clustering

Group together similar points

Issues

- How do we represent an example?
- How do we compute similarity between two examples?
- How to cluster?
- How many clusters?

Domain knowledge – we will assume these are given / known

Based on slides by David Kauchak and David Sontag
Clustering Algorithms

**Flat / Partitional**
- Construct various partitions then evaluate by some criterion
- Examples
  - k-means
  - mixture of Gaussians
  - spectral

**Hierarchical**
- Create hierarchical decomposition
- Examples
  - agglomerative (bottom-up)
  - divisive (top-down)

Based on slides by David Kauchak and David Sontag

(This slide intentionally left blank.)
K-Means Clustering
Learning Goals

• Describe \( k \)-means algorithm
• Describe \( k \)-means objective
• Describe \( k \)-means limitations and extensions

**k-Means Algorithm**

Given training set \( \left\{ x^{(i)} \right\}_{i=1}^{n} \), \( x^{(i)} \in \mathbb{R}^{d} \), and number of clusters \( k \)

Goal: Group data into cohesive “clusters”

1. Initialize cluster centers
2. Repeat until convergence {
   Assign each example to closest center
   Update cluster centers
}

What details do we have to specify?
**k-Means Algorithm**

1. Initialize $k$ cluster centroids randomly
   \[ \mu_1, \ldots, \mu_k \in \mathbb{R}^d \]

2. Repeat until convergence {
   \[
   \text{for } i = 1, \ldots, n \\
   \quad \text{set } c^{(i)} = \arg \min_{j=1,\ldots,k} \| x^{(i)} - \mu_j \|^2 \\
   \text{for } j = 1, \ldots, k \\
   \quad \text{set } \mu_j = \frac{\sum_{i=1}^n I\left[ c^{(i)} = j \right] x^{(i)}}{\sum_{i=1}^n I\left[ c^{(i)} = j \right]} \\
   \]

Interactive Demo

https://www.naftaliharris.com/blog/visualizing-k-means-clustering/

Based on notes by Andrew Ng
Exercise

You are to cluster eight points:

\[ x^{(1)} = [2,10]^T \quad x^{(2)} = [2,5]^T \quad x^{(3)} = [8,4]^T \quad x^{(4)} = [5,8]^T \]
\[ x^{(5)} = [7, 5]^T \quad x^{(6)} = [6,4]^T \quad x^{(7)} = [1,2]^T \quad x^{(8)} = [4,9]^T \]

You assign \( x^{(1)} \), \( x^{(3)} \), and \( x^{(7)} \) as initial centers (\( k = 3 \)). Run \( k \)-means using Manhattan distance (\( \ell_1 \)-norm). Compute cluster centers and assignments for each round until convergence.

(hint: you may not need all the plots)

cluster 1 ⇒
cluster 2 ⇒
cluster 3 ⇒
Optimization Objective

Is $k$-means guaranteed to converge? yes

Define **distortion function**

$$J(c, \mu) = \sum_{i=1}^{n} \|x^{(i)} - \mu_{c(i)}\|^2$$

where $c = [c^{(1)}, \ldots, c^{(n)}]^T$ and $\mu = [\mu_1, \ldots, \mu_k]^T$.

$J$ measures sum of squared distances between each training example $x^{(i)}$ and the cluster centroid to which it has been assigned. In other words, it measures intra-class variance.

Based on notes by Andrew Ng

---

Optimization Objective

Claim: $k$-means is **coordinate descent** on $J$.

Inner loop of $k$-means algorithm repeatedly...

- holds $\mu$ fixed and minimizes $J(c, \mu)$ w.r.t. $c$, and then
- holds $c$ fixed and minimizes $J(c, \mu)$ w.r.t. $\mu$.

Thus

- $J$ must monotonically decrease, and
- the value of $J$ must converge.

Usually, this implies that $c$ and $\mu$ will converge, too. In theory, it is possible for $k$-means to oscillate between a few different clusterings (a few different values for $c$ and/or $\mu$ that have exactly the same value of $J$), but this almost never happens in practice.

Exact optimization of $J$ is NP-hard. $K$-means is heuristic that converges to local optimum.

Based on notes by Andrew Ng
Choosing the Number of Clusters

Elbow method
- Try different values of $k$
- Plot objective vs $k$
- Look for “elbow”

Application-Specific
- If running $k$-means to use clusters for some later purpose
- Evaluate $k$-means based on metric for how well it performs for that later purpose
- E.g. t-shirt size, clusters using height and weight as features
  - 3 clusters ⇒ S, M, L ⇒ fewer sizes, can make shirts more cheaply
  - 5 clusters ⇒ XS, S, M, L, XL ⇒ more sizes, can make better-fitting shirts

Based on slides by Andrew Ng

---

$k$-Means Time Complexity

(1) Initialize $k$ cluster centroids randomly
\[ \mu_1, \ldots, \mu_k \in \mathbb{R}^d \]

(2) Repeat until convergence {
\[
\text{for } i = 1, \ldots, n \\
\quad \text{set } c^{(i)} = \arg \min_{j=1,\ldots,k} \| x^{(i)} - \mu_j \|^2 \\
\text{for } j = 1, \ldots, k \\
\quad \text{set } \mu_j = \frac{\sum_{i=1}^n I \left[ \left[ c^{(i)} = j \right] \right] x^{(i)}}{\sum_{i=1}^n I \left[ \left[ c^{(i)} = j \right] \right]} \\
\}

What is time complexity?
(k clusters, $n$ training examples, $d$ features, $i$ iterations)
Initialization Issues

Problem
• Often randomly pick $k < n$ examples as starting centers
• But $k$-means is extremely sensitive to cluster center initialization

Reasoning
• Distortion function $J$ is non-convex function
• $k$-means can be susceptible to local optima

Bad initialization can lead to
• Poor convergence speed
• Bad overall clusterings

Based on notes by Andrew Ng

Initialization Issues

In practice
• Very often $k$-means works fine and comes up with very good clusterings despite local minima

Safeguarding approaches
• Run $k$-means many times (using different random initializations), then choose clustering that gives lowest distortion
• $k$-means++: spread out cluster centers
  – choose first center uniformly at random from examples
  – choose remaining centers from remaining examples with probability proportional to squared distance from example’s closest cluster center

Based on notes by Andrew Ng
Limitations

$k$-means works well only for round-shaped, roughly equal size / density clusters

non-convex / non-round shaped clusters

different density clusters

Extensions

Problem: clusters have non-convex shapes

• spectral clustering (value connectivity over compactness)

• kernelized $k$-means
Extensions

Problem: sensitive to outlier examples
- $k$-medians
  - median more robust than mean in presence of outliers

Problem: cluster centers not an actual example
- $k$-medoids
  - medoid = element of cluster whose average dissimilarity to all elements in cluster is minimal ("most centrally located point in cluster")

Based on slides by Piyush Rai

Extensions

Problem: makes hard assignments of points to clusters
- a point completely belongs to cluster or not at all
- no notion of soft assignment

- fuzzy $k$-means
  - let weight $w_{ij} \in [0,1]$ measure "degree of belonging"
    (degree to which element $x^{(i)}$ belongs to cluster $c_j$)
  \[
  \mu_j = \frac{\sum_{i=1}^{n} w_{ij} x^{(i)}}{\sum_{i=1}^{n} w_{ij}}
  \]
  \[
  J(c, \mu) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{ij} \|x^{(i)} - \mu_{c(i)}\|^2
  \]
  - $w_{ij}$ typically inversely related to distance from cluster center
  - compare standard (hard assignment) $k$-means: $w_{ij} = \mathbb{I}\left[c^{(i)} = j\right]$

- Gaussian Mixture Models (GMMs)
  - more statistically formalized method
  - probabilistic assignments to clusters and multivariate Gaussian distributions
  - next time...
Hierarchical Clustering

Learning Goals

• Describe agglomerative clustering algorithm
• Define single link, complete link, and average link and describe how they affect clustering

Agglomerative (aka bottom-up hierarchical) Clustering

Idea

• Start with each item in its own cluster
• Find best pair to merge into new cluster
• Repeat until all clusters are merged

Properties

• Produces not one clustering but a family of clusterings represented by a dendrogram
• # of dendograms with $n$ leaves = $\frac{(2n-3)!}{(2^{n-2})(n-2)!}$
• Compare to agglomerative algorithm $n$ loops, on each loop compute $n^2$ distances
time complexity: $O(n^3)$

Based on slides by Eric Eaton
Distance Matrix

Contains distances between every pair of objects in training set

\[ d(\text{object}_1, \text{object}_2) = 8 \]
\[ d(\text{object}_3, \text{object}_4) = 1 \]

Based on slides by Ziv Bar-Joseph

Bottom-up (agglomerative)
- Starting with each item in its own cluster
- Find best pair to merge into new cluster
- Repeat until all clusters are merged

Consider all possible merges...
Choose the best

Consider all possible merges...
Choose the best

Consider all possible merges...
Choose the best

Based on slides by Ziv Bar-Joseph
Distance between Clusters

Match the linkage type to the behavior and picture (separately)

- **single linkage**
  - use **closest** pair
  - tight (small, round) clusters

- **complete linkage**
  - use **farthest** pair
  - robust against noise

- **average linkage**
  - use **average** of all pairs
  - potentially long and skinny clusters

Exercise

Given distance matrix, run single-link (closest pair) clustering.

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0</td>
<td>2</td>
<td>6</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>0</td>
<td>3</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>C</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>D</td>
<td>10</td>
<td>9</td>
<td>7</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>9</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>0</td>
</tr>
</tbody>
</table>
# Summary Comparison

<table>
<thead>
<tr>
<th>Flat</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Partitions independent of one another</td>
<td>• Produces different partitionings depending on level of granularity (refine or coarsen clusters by picking different $k$)</td>
</tr>
<tr>
<td>• Produces single partitioning</td>
<td>• Partitions nested within one another</td>
</tr>
<tr>
<td>• Requires $k$ as input</td>
<td>• Can pick number of clusters after clustering</td>
</tr>
<tr>
<td>• More efficient runtime-wise</td>
<td>• Can be slow (has to make several merge/split decisions)</td>
</tr>
</tbody>
</table>

No clear consensus on which produces better clustering

---

# Evaluation

## Learning Goals

- Describe how to validate clusters produced by algorithms
Validation

How “good” are our clusters?

• **external** validation
  - match to known categories (cluster data without labels, see how well we reproduce labels)
  - more common

• **internal** validation
  - no external labels

---

**External Validation**

**purity** = proportion of dominant class in cluster

Cluster I
- red dots = dominant class
- blue dots = other classes
- purity = \( \frac{\max(3, 1, 0)}{4} = \frac{3}{4} \)

Cluster II
- red dots = dominant class
- blue dots = other classes
- purity = \( \frac{\max(1, 4, 1)}{6} = \frac{4}{6} \)

Cluster III
- red dots = dominant class
- blue dots = other classes
- purity = \( \frac{\max(2, 0, 3)}{5} = \frac{3}{5} \)

**overall purity**

\[
\frac{\frac{3}{4} + \frac{4}{6} + \frac{3}{5}}{3} = 0.672
\]

Cluster average

\[
\frac{\frac{3}{4} + \frac{4}{6} + \frac{3}{5}}{3} = 0.672
\]

Weighted average

\[
\frac{4\left(\frac{3}{4}\right) + 6\left(\frac{4}{6}\right) + 5\left(\frac{3}{5}\right)}{15} = \frac{3 + 4 + 3}{15} = \frac{2}{3}
\]

Based on slides by Ziv Bar-Joseph and David Kauchak.
Internal Validation

**stability:** if clusters capture real structure, they should be stable to minor perturbation (e.g. subsampling) of data

- Need measure of similarity between two $k$-clusterings
  - For any set of clusters $C$, define $L(C)$ as matrix of 0/1 labels
    - $L(C)_{ij} = 1$ if examples $x^{(i)}$ and $x^{(j)}$ belong to same cluster
    - $L(C)_{ij} = 0$ otherwise
  - Let $S = sim(L(C), L(C'))$ be similarity between two matrices
    - e.g. fraction of identical elements in matrices

- Let...
  - $C$ denote clusters from all samples
  - $C'_i$ denote clusters from $i$th randomly chosen subset of samples
  - $S_n$ denote average of $n$ scores between $L(C)$ and $L(C'_i)$

- Have high confidence in $C$ if $S_n \to 1$ with high probability
  (where comparison is done over samples common to both)

Based on slides by Ziv Bar-Joseph

---

Take-Aways

- Clustering basics
  - what it is
  - why it is useful

- Clustering algorithms
  - $k$-means
    - algorithm
    - objective (distortion function)
    - issues (initialization, choosing $k$, convex clusters)
    - extensions ($k$-medians, $k$-medoids)
  - agglomerative
    - algorithm
    - single-linkage, complete-linkage, average-linkage

- Clustering metrics
  - external
  - internal