Clustering, K-Means, and K-Nearest Neighbors

CMSC 678
UMBC
March 7\textsuperscript{th}, 2018

Most slides courtesy Hamed Pirsiavash
Recap from last time...
Objective: to **rigidly rotate** the axes of the D-dimensional space to new positions (**principal axes**):

ordered such that **principal axis 1 has the highest variance**, axis 2 has the next highest variance, ..., and axis D has the lowest variance.

covariance among each pair of the principal axes is zero (**the principal axes are uncorrelated**)
L-Dimensional PCA

1. Compute mean $\mu$, priors, and common covariance $\Sigma$

$$\Sigma = \frac{1}{N} \sum_{i:y_i=k} (x_i - \mu)(x_i - \mu)^T$$
$$\mu = \frac{1}{N} \sum_i x_i$$

2. Sphere the data (zero-mean, unit covariance)

3. Compute the (top L) eigenvectors, from sphere-d data, via $V$

$$X^* = V D_B V^T$$

4. Project the data
Outline

**Clustering basics**

K-means: basic algorithm & extensions

Cluster evaluation

Non-parametric mode finding: density estimation

Graph & spectral clustering

Hierarchical clustering

K-Nearest Neighbor
Clustering

Basic idea: group together similar instances

Example: 2D points
Basic idea: group together similar instances
Example: 2D points

One option: small Euclidean distance (squared)
\[
\text{dist}(x, y) = \|x - y\|_2^2
\]

Clustering results are crucially dependent on the measure of similarity (or distance) between points to be clustered.
Clustering algorithms

Simple clustering: organize elements into k groups
  K-means
  Mean shift
  Spectral clustering

Hierarchical clustering: organize elements into a hierarchy
  Bottom up - agglomerative
  Top down - divisive
Clustering examples: Image Segmentation

image credit: Berkeley segmentation benchmark
Clustering examples: News Feed

Nuclear deal within reach, vows Iran and Russia
The Australian - 2 hours ago
Russia and Iran claimed a breakthrough in talks on a framework deal cutting back Tehran's nuclear program, but the US denied everything had been agreed as discussions were due to resume overnight.

Religious Freedom Act: Are businesses becoming more socially activist? (+video)
Christian Science Monitor - 10 minutes ago
The companies castigating Indiana's RFRA law are not promoting liberal idealism over profits: Their response is a recognition that - at least when it comes to the issue of gay marriage - social activism is also good business.

ISIS’ legacy in Tikrit: booby traps, IEDs and fear
CNN - 1 hour ago
Tikrit, Iraq (CNN) ISIS is gone, but the fear remains. As Iraqi forces, aided by Shiite militiamen, took control Wednesday of the northern city of Tikrit, they found vehicles laden with explosives and buildings that might be booby-trapped.

Germanwings Crash: Video May Show Plane’s Final Moments
ABC News - 1 hour ago
Two magazines have reported details of a disturbing video taken from inside the doomed Germanwings plane moments before it crashed into the French Alps, but investigators have denied its existence.
Clustering examples: Image Search
Outline

Clustering basics

K-means: basic algorithm & extensions

Cluster evaluation

Non-parametric mode finding: density estimation

Graph & spectral clustering

Hierarchical clustering

K-Nearest Neighbor
Clustering using k-means

Data: D-dimensional observations \((x_1, x_2, ..., x_n)\)

Goal: partition the \(n\) observations into \(k\ (\leq n)\) sets \(S = \{S_1, S_2, ..., S_k\}\) so as to minimize the within-cluster sum of squared distances

\[
\arg\min_S \sum_{i=1}^{k} \sum_{x \in S_i} \|x - \mu_i\|^2
\]

cluster center
Lloyd’s algorithm for k-means

Initialize k centers by picking k points randomly among all the points

Repeat till convergence (or max iterations)

Assign each point to the nearest center (assignment step)

Estimate the mean of each group (update step)

https://www.csee.umbc.edu/courses/graduate/678/spring18/kmeans/
Properties of the Lloyd’s algorithm

Guaranteed to converge in a finite number of iterations
  objective decreases monotonically
  local minima if the partitions don’t change.
  finitely many partitions \( \rightarrow \) k-means algorithm must converge

Running time per iteration
  Assignment step: \( O(NKD) \)
  Computing cluster mean: \( O(ND) \)

Issues with the algorithm:
  Worst case running time is super-polynomial in input size
  No guarantees about global optimality
  Optimal clustering even for 2 clusters is NP-hard [Aloise et al., 09]
k-means++ algorithm

A way to pick the good initial centers
    Intuition: spread out the k initial cluster centers

The algorithm proceeds normally once the centers are initialized

[Arthur and Vassilvitskii’07] The approximation quality is $O(\log k)$ in expectation

k-means++ algorithm for initialization:

1. Chose one center uniformly at random among all the points
2. For each point $x$, compute $D(x)$, the distance between $x$ and the nearest center that has already been chosen
3. Chose one new data point at random as a new center, using a weighted probability distribution where a point $x$ is chosen with a probability proportional to $D(x)^2$
4. Repeat Steps 2 and 3 until k centers have been chosen
k-means for image segmentation

Grouping pixels based on intensity similarity

feature space: intensity value (1D)
Outline

Clustering basics

K-means: basic algorithm & extensions

Cluster evaluation

Non-parametric mode finding: density estimation

Graph & spectral clustering

Hierarchical clustering

K-Nearest Neighbor
Clustering Evaluation

(Classification: accuracy, recall, precision, F-score)

Greedy mapping: one-to-one

Optimistic mapping: many-to-one

Rigorous/information theoretic: V-measure
Clustering Evaluation: One-to-One

Each modeled cluster can at most only map to one gold tag type, and vice versa.

Greedily select the mapping to maximize accuracy.
Clustering Evaluation: Many (classes)-to-One (cluster)

Each modeled cluster can map to at most one gold tag types, but multiple clusters can map to the same gold tag.

For each cluster: select the majority tag.
Clustering Evaluation: V-Measure

Rosenberg and Hirschberg (2008): harmonic mean of homogeneity and completeness

\[ H(X) = - \sum_{i} p(x_i) \log p(x_i) \]
Clustering Evaluation: V-Measure

Rosenberg and Hirschberg (2008): harmonic mean of homogeneity and completeness

\[ H(X) = - \sum_{i} p(x_i) \log p(x_i) \]

entropy(point mass) = 0  entropy(uniform) = log K
Clustering Evaluation: V-Measure

Rosenberg and Hirschberg (2008): harmonic mean of *homogeneity* and *completeness*

Homogeneity: how well does each gold class map to a single cluster?

“In order to satisfy our homogeneity criteria, a clustering must assign only those datapoints that are members of a single class to a single cluster. That is, the class distribution within each cluster should be skewed to a single class, that is, zero entropy.”

\[
\text{homogeneity} = \begin{cases} 
1, & H(K, C) = 0 \\
1 - \frac{H(C|K)}{H(C)}, & \text{o/w}
\end{cases}
\]

relative entropy is maximized when a cluster provides no new info. on class grouping \(\rightarrow\) not very homogeneous
Clustering Evaluation: V-Measure

Rosenberg and Hirschberg (2008): harmonic mean of *homogeneity* and *completeness*

Completeness: how well does each learned cluster cover a *single* gold class?

\[
\text{completeness} = \begin{cases} 
1, & H(K, C) = 0 \\
1 - \frac{H(K|C)}{H(K)}, & \text{o/w}
\end{cases}
\]

“In order to satisfy the completeness criteria, a clustering must assign all of those datapoints that are members of a single class to a single cluster.”

Relative entropy is maximized when each class is represented uniformly (relatively) → not very complete
Clustering Evaluation: V-Measure

Rosenberg and Hirschberg (2008): harmonic mean of homogeneity and completeness

Homogeneity: how well does each gold class map to a single cluster?

Homogeneity = \[
\begin{cases}
1, & H(K, C) = 0 \\
1 - \frac{H(C|K)}{H(C)}, & \text{o/w}
\end{cases}
\]

Completeness: how well does each learned cluster cover a single gold class?

Completeness = \[
\begin{cases}
1, & H(K, C) = 0 \\
1 - \frac{H(K|C)}{H(K)}, & \text{o/w}
\end{cases}
\]
Clustering Evaluation: V-Measure

Rosenberg and Hirschberg (2008): harmonic mean of homogeneity and completeness

Homogeneity: how well does each gold class map to a single cluster?

Completeness: how well does each learned cluster cover a single gold class?

\[ a_{ck} = \# \text{ elements of class } c \text{ in cluster } k \]

\[ H(C|K) = -\sum_{k} \sum_{c} \frac{a_{ck}}{K} \log \frac{a_{ck}}{\sum_{c'} a_{ck'}} \]

\[ H(K|C) = -\sum_{c} \sum_{k} \frac{a_{ck}}{C} \log \frac{a_{ck}}{\sum_{k'} a_{ck'}} \]

\[ \text{homogeneity} = \begin{cases} 1, & H(K, C) = 0 \\ 1 - \frac{H(C|K)}{H(C)}, & \text{o/w} \end{cases} \]

\[ \text{completeness} = \begin{cases} 1, & H(K, C) = 0 \\ 1 - \frac{H(K|C)}{H(K)}, & \text{o/w} \end{cases} \]
Clustering Evaluation: V-Measure

Rosenberg and Hirschberg (2008): harmonic mean of homogeneity and completeness

Homogeneity: how well does each gold class map to a single cluster?

Completeness: how well does each learned cluster cover a single gold class?

\[
H(C|K) = - \sum_{k} \sum_{c} \frac{a_{ck}}{N} \log \left( \frac{a_{ck}}{\sum_{c'} a_{c'k}} \right)
\]

\[
H(K|C) = - \sum_{c} \sum_{k} \frac{a_{ck}}{N} \log \left( \frac{a_{ck}}{\sum_{k'} a_{ck'}} \right)
\]

<table>
<thead>
<tr>
<th>$a_{ck}$</th>
<th>K=1</th>
<th>K=2</th>
<th>K=3</th>
</tr>
</thead>
<tbody>
<tr>
<td>□</td>
<td>3</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>○</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>★</td>
<td>1</td>
<td>3</td>
<td>1</td>
</tr>
</tbody>
</table>

Homogeneity = Completeness = V-Measure = 0.14
Outline

Clustering basics

K-means: basic algorithm & extensions

Cluster evaluation

Non-parametric mode finding: density estimation

Graph & spectral clustering

Hierarchical clustering

K-Nearest Neighbor
One issue with k-means is that it is sometimes hard to pick k.

The mean shift algorithm seeks modes or local maxima of density in the feature space.

Mean shift automatically determines the number of clusters.

\[
K(x) = \frac{1}{Z} \sum_i \exp \left( -\frac{||x - x_i||^2}{h} \right)
\]

Small \( h \) implies more modes (bumpy distribution).
Mean shift algorithm

For each point $x_i$:

set $m_i = x_i$

while not converged:

compute $m_i = \frac{\sum_i x_i K(m_i, x_i)}{\sum_i K(m_i, x_i)}$

return \{m_i\}

**Pros:**
- Does not assume shape on clusters
- Generic technique
- Finds multiple modes
- Parallelizable

**Cons:**
- Slow: $O(DN^2)$ per iteration
- Does not work well for high-dimensional features

self-clustering to based on kernel (similarity to other points)
Mean shift clustering results

http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html
Outline

Clustering basics

K-means: basic algorithm & extensions

Cluster evaluation

Non-parametric mode finding: density estimation

Graph & spectral clustering

Hierarchical clustering

K-Nearest Neighbor
Spectral clustering

[Shi & Malik ‘00; Ng, Jordan, Weiss NIPS ‘01]
Group points based on the links in a graph

How do we create the graph?
Weights on the edges based on similarity between the points
A common choice is the Gaussian kernel

$$W(i, j) = \exp \left( - \frac{||x_i - x_j||^2}{2\sigma^2} \right)$$

One could create
- A fully connected graph
- k-nearest graph (each node is connected only to its k-nearest neighbors)
Consider a partition of the graph into two parts A and B

Cut(A, B) is the weight of all edges that connect the two groups

$$\text{Cut}(A, B) = \sum_{i \in A, j \in B} W(i, j) = 0.3$$

An intuitive goal is to find a partition that minimizes the cut

min-cuts in graphs can be computed in polynomial time
Problem with min-cut

The weight of a cut is proportional to number of edges in the cut; tends to produce small, isolated components.

We would like a balanced cut

Fig. 1. A case where minimum cut gives a bad partition.

[Shi & Malik, 2000 PAMI]
Let \( W(i, j) \) denote the matrix of the edge weights.

The degree of node in the graph is:

\[
d(i) = \sum_j W(i, j)
\]

The volume of a set \( A \) is defined as:

\[
\text{Vol}(A) = \sum_{i \in A} d(i)
\]
the connectivity between the groups relative to the volume of each group:

\[ \text{NCut}(A, B) = \frac{\text{Cut}(A, B)}{\text{Vol}(A)} + \frac{\text{Cut}(A, B)}{\text{Vol}(B)} \]

\[ \text{NCut}(A, B) = \text{Cut}(A, B) \left( \frac{\text{Vol}(A) + \text{Vol}(B)}{\text{Vol}(A)\text{Vol}(B)} \right) \]

minimized when \( \text{Vol}(A) = \text{Vol}(B) \)

\[ \Rightarrow \text{a balanced cut} \]

Minimizing normalized cut is NP-Hard even for planar graphs [Shi & Malik, 00]
Solving normalized cuts

$W$: the similarity matrix

$D$: a diagonal matrix with $D(i,i) = d(i)$ — the degree of node $i$

$y$: a vector $\{1, -b\}^N$, $y(i) = 1 \iff i \in A$

The matrix $(D-W)$ is called the Laplacian of the graph

$$\min_{x} \text{NCut}(x) = \min_{y} \frac{y^T (D - W)y}{y^T D y}$$

subject to: $y^T D 1 = 0$

$y(i) \in \{1, -b\}$
Solving normalized cuts

Normalized cuts objective:
\[ \min_{\mathbf{x}} \text{NCut}(\mathbf{x}) = \min_{\mathbf{y}} \left( \frac{\mathbf{y}^T(D - W)\mathbf{y}}{\mathbf{y}^T D \mathbf{y}} \right) \]

subject to: \( \mathbf{y}^T D \mathbf{1} = 0 \)
\( \mathbf{y}(i) \in \{1, -b\} \)

Relax the integer constraint on \( \mathbf{y} \):
\[ \min_{\mathbf{y}} \mathbf{y}^T(D - W)\mathbf{y}; \text{ subject to: } \mathbf{y}^T D \mathbf{y} = 1, \mathbf{y}^T D \mathbf{1} = 0 \]

Same as: \((D - W)\mathbf{1} = 0\) (Generalized eigenvalue problem)
\((D - W)\mathbf{y} = \lambda D \mathbf{y} \Rightarrow \) the first eigenvector is \( \mathbf{y}_1 = \mathbf{1} \), with the corresponding eigenvalue of 0

The eigenvector corresponding to the second smallest eigenvalue is the solution to the relaxed problem
Outline

Clustering basics

K-means: basic algorithm & extensions

Cluster evaluation

Non-parametric mode finding: density estimation

Graph & spectral clustering

Hierarchical clustering

K-Nearest Neighbor
Hierarchical clustering

**Agglomerative:** a “bottom up” approach where elements start as individual clusters and clusters are merged as one moves up the hierarchy.

**Divisive:** a “top down” approach where elements start as a single cluster and clusters are split as one moves down the hierarchy.
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
How should we define “closest” for clusters with multiple elements?

Closest pair: single-link clustering
Farthest pair: complete-link clustering
Average of all pairs
Agglomerative clustering

Closest pair
(single-link clustering)

Farthest pair
(complete-link clustering)

[Pictures from Thorsten Joachims]
Outline

Clustering basics

K-means: basic algorithm & extensions

Cluster evaluation

Non-parametric mode finding: density estimation

Graph & spectral clustering

Hierarchical clustering

K-Nearest Neighbor
Nearest neighbor classifier

Will Alice like the movie?
  Alice and James are similar
  James likes the movie →
  Alice must/might also like the movie

Represent data as vectors of feature values

Find closest (Euclidean norm) points

Euclidean distance (d) = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}
Nearest neighbor classifier

Training data is in the form of $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$

Fruit data:
- label: \{apples, oranges, lemons\}
- attributes: \{width, height\}
Nearest neighbor classifier

Fruit data

(a, b) ?
lemon

(c, d) ?
apple

test data
k-Nearest neighbor classifier

Take majority vote among the $k$ nearest neighbors

Fruit data

- **apples**
- **oranges**
- **lemons**

Outlier
k-Nearest neighbor classifier

Take majority vote among the k nearest neighbors

Fruit data

- apples
- orange
- lemons

What is the effect of k?
Decision boundaries: 1NN
Inductive bias of the kNN classifier

Choice of features

We are assuming that all features are equally important
What happens if we scale one of the features by a factor of 100?

Choice of distance function

Euclidean, cosine similarity (angle), Gaussian, etc ...
Should the coordinates be independent?

Choice of k
An example: Synthesizing one pixel

What is $P(x | \text{neighborhood of pixels around } x)$?

Find all the windows in the image that match the neighborhood.

To synthesize $x$:
- Pick one matching window at random.
- Assign $x$ to be the center pixel of that window.

An exact match might not be present, so find the best matches using Euclidean distance and randomly choose between them, preferring better matches with higher probability.
kNN: Scene Completion

“Scene completion using millions of photographs”, Hayes and Efros, TOG 2007
kNN: Scene Completion

Nearest neighbors

“Scene completion using millions of photographs”, Hayes and Efros, TOG 2007
kNN: Scene Completion

“Scene completion using millions of photographs”, Hayes and Efros, TOG 2007
kNN: Scene Completion

“Scene completion using millions of photographs”, Hayes and Efros, TOG 2007
Practical issue when using kNN: speed

Time taken by kNN for N points of D dimensions
  - time to compute distances: $O(ND)$
  - time to find the k nearest neighbor
    - $O(k N)$: repeated minima
    - $O(N \log N)$: sorting
    - $O(N + k \log N)$: min heap
    - $O(N + k \log k)$: fast median
  - Total time is dominated by distance computation

We can be faster if we are willing to sacrifice exactness
Practical issue when using kNN: Curse of dimensionality

- \( \text{#bins} = 10 \times 10 \)
- \( d = 2 \)
- \( \text{#bins} = 10^d \)
- \( d = 1000 \)

Atoms in the universe: \( \sim 10^{80} \)

How many neighborhoods are there?