Machine Learning & Data Mining
CS/CNS/EE 155

Lecture 9:
Clustering & Dimensionality Reduction
Kaggle Competition

• Released soon

• Teams of 2-3

• Competition closes Tuesday Feb 12\(^{th}\), 2pm
  – Winners announced in class
  – Report due Feb 14\(^{th}\), 9pm
Lecture 9: Clustering & Dimensionality Reduction
Today
(Unsupervised Learning)

• Clustering

• Dimensionality Reduction
  – Matrix Factorization
What is Clustering?

- Clustering is the process of grouping data points into “clusters”.
- High intra-cluster similarity
- Low inter-cluster similarity
Example
Example
Unsupervised Learning

• **Given:** unlabeled data:
  – Only input features
  – No labels

• **Goal:** find hidden structure/patterns
  – E.g., hidden structure is a clustering of data
  – A generative model of data $P(x)$
    • Discussed further in future lectures
  – I.e., a low dimensional summary of the data

$$S = \{ x_i \}_{i=1}^{N}$$
Why is Clustering Useful?

• Clustering is a “summary” of data
  – Can just inspect cluster centers
  – Or inspect a few data points per cluster
Images Related to “Pluto”

Each Row is a Cluster

Why is Clustering Useful?

• Clustering is a “summary” of data
  – Can just inspect cluster centers
  – Or inspect a few data points per cluster

• Compact pre-processing of data before supervised training
Centroid Based Clustering
(K-Means)
Centroid Based Clustering
(K-Means)
Centroid Based Clustering

(K-Means)
Centroid Based Clustering
(K-Means)
Centroid Based Clustering

(K-Means)
Centroid Based Clustering

(K-Means)
Centroid Based Clustering
(K-Means)
Centroid Based Clustering (K-Means)
Centroid Based Clustering

(K-Means)
Centroid Based Clustering

(K-Means)
Centroid Based Clustering
(K-Means)
Centroid Based Clustering

(K-Means)
Centroid Based Clustering

(K-Means)
Centroid Based Clustering
(K-Means)
K-Means Objective

\[ S = \left\{ x_i \right\}_{i=1}^{N} \]

\[ \text{argmin}_{S=C_1 \cup \ldots \cup C_K, \ \{ c_1, \ldots, c_K \}} \sum_{k} \sum_{x \in C_k} \| x - c_k \|^2 \]

Equivalent!

\[ \text{argmin}_{S=C_1 \cup \ldots \cup C_K} \sum_{k} |C_k| \text{var}(C_k) \]
EM Algorithm for K-Means
(Expectation/Maximization)

\[ S = \{ x_i \}_{i=1}^{N} \]

- **E-Step**
  - Estimate \( C_k \)
  - Estimate cluster membership

- **M-Step**
  - Estimate \( c_k \)
  - Estimate model parameters

\[
\arg\min_{S=C_1 \cup \ldots \cup C_K, \{c_1, \ldots, c_K\}} \sum_{k} \sum_{x \in C_k} \|x - c_k\|^2
\]
E-Step

\[
\text{argmin}_{S=C_1 \cup \ldots \cup C_K, \{c_1, \ldots, c_K\}} \sum_k \sum_{x \in C_k} \| x - c_k \|^2
\]

\[S = \{x_i\}_{i=1}^N\]

• For each x:
  – Assign to cluster \( C_k \) with smallest distance to \( c_k \)
M-Step

\[
\arg\min_{S = \bigcup_{k=1}^{K} C_k, \{c_1, \ldots, c_K\}} \sum_{k} \sum_{x \in C_k} \|x - c_k\|^2
\]

• For each \(c_k\):
  – Compute \(c_k = \text{mean}(C_k)\)
Interpretation

• Summarize data by cluster membership
• Learn clustering to minimize intra-cluster variance
  – “Best reconstruction of the data”

\[
\text{argmin}_{S=C_1 \cup \ldots \cup C_K, \{c_1, \ldots, c_K\}} \sum_k \sum_{x \in C_k} \|x - c_k\|^2
\]

\[
\text{argmin}_{S=C_1 \cup \ldots \cup C_K} \sum_k |C_k| \text{var}(C_k)
\]
Recap: K-Means

• Centroid-based Clustering
  – Defines clusters using a notional of centrality
  – E.g., all items in the cluster must be close to each other

• Solve using EM algorithm
  – Also probabilistic variant (Gaussian Mixture Models)

• Useful when centrality assumption is good
  – But bad when centrality assumption is bad...
Thought Experiment

What is good clustering?
Linkage Based Clustering
(Hierarchical Clustering)

• K-Means used centroid clustering structure
  – Clustered data points are “close” to cluster center

• Sometimes a linkage structure is better...
  – Employ hierarchical clustering
  – E.g., agglomerative clustering
Agglomerative Clustering
Agglomerative Clustering

• Equivalent to finding minimum spanning tree
  – Kruskal’s Algorithm

• Order that edges are added defines the cluster hierarchy

• Equivalent to finding a binary tree partitioning with progressively smaller partition distances
Recap: Clustering

• Unsupervised learning
  – Finds the clustering structure of input features

• Centroid based
  – Clusters should be clumped together
  – K-Means

• Linkage Based
  – Clusters can be organized hierarchically
  – Agglomerative Clustering

• Works great when clustering assumption is good!
Limitations of Clustering
Principal Component Analysis
Summarizing Data

- Summarize data using smaller #attributes $S = \{x_i\}_{i=1}^N$

- Clustering: summarize data via clusters
  - K-Means: summarize via cluster membership
  - Gaussian Mixture Model: Summarize via distribution over K clusters

- PCA: summarize via orthogonal projections
  - Define new feature representation
  - Rotation + Projection
Principal Component Analysis
Principal Component Analysis

New Feature Representation!
Orthogonal Matrix

• A matrix $U$ is orthogonal if $UU^T = U^TU = I$
  - For any column $u$: $u^Tu = 1$
  - For any two columns $u, u'$: $u^Tu' = 0$
  - $U$ is a rotation matrix, and $U^T$ is the inverse rotation
  - If $x' = U^Tx$, then $x = Ux'$

PCA finds a specific orthogonal $U$
Properties of Orthogonal Matrices

• \( x' = U^T x, \ x = Ux' \)

• Norm preserving:
  \[
  x'^T x' = (U^T x)^T (U^T x) = x^T U U^T x = x^T x
  \]

• Preserves Total Variance:
  \[
  \sum_{d=1}^{D} \sum_{i=1}^{N} (x_i^{(d)})^2 = \sum_{d=1}^{D} \sum_{i=1}^{N} (x_i'^{(d)})^2
  \]
  Assuming zero mean
Principal Component Analysis

Summarize Using 1 Feature?
Principal Component Analysis

Summarize Using 1 Feature?
Principal Component Analysis

Summarize Using 1 Feature?

Works with arbitrary subsets of columns of orthogonal $U$

E.g., $U' = [u_1, u_5, u_{20}]$

"projection" $u_1^T x$

"reconstruction" $u_1 u_1^T x$
PCA Formal Definition

• Define $M$=matrix of all data:

\[ X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N} \]

• Mean center:

\[ \bar{X} = X - [\bar{x}, \ldots, \bar{x}] \]

• PCA:

\[ \bar{X}\bar{X}^T = U \Lambda U^T \]

- Symmetric
- Orthogonal
- Diagonal
Properties of PCA

\[ XX^T = U \Lambda U^T \]

Assuming zero mean

- Each column of \( U \) is an Eigenvector
- Each \( \lambda \) is an Eigenvalue
  \[- \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_D \]

\[ (XX^T)u_d = \lambda_d u_d \]
Interpretation

Feature Covariance Matrix:

\[ \Sigma = XX^T = U \Lambda U^T \]

- \( \Sigma_{dd'} \) is the covariance of features \( d \) & \( d' \) in training data.
- The first column \( u_1 \) is the single direction of greatest variation
  - \( \lambda_1 \) is the total variation along \( u_1 \):

\[
\lambda_1 = \sum_{i=1}^{N} (u_1^T x_i)^2 = \sum_{i=1}^{N} (x_i^{(1)})^2
\]
• The first column $u_1$ is the single direction that minimizes the squared loss of reconstructing the original $x$’s
  – I.e., minimizes the amount of residual variation

• One can prove that:

$$u_1 = \text{argmin}_{u: u^Tu=1} \sum_{i=1}^{N} \left\| x_i - uu^T x_i \right\|^2$$

“Residual”

• (From definition in previous slide)
Definition: $u_1$ is the direction that captures the most variation

$$u_1 = \arg \max_{u: u^T u = 1} \sum_{i=1}^{N} \| u^T x_i \|^2$$

Step 1: for any $x$, its residual direction is orthogonal to $u_1$

Residual: $x - u_1 u_1^T x$

$$\left( x - u_1 u_1^T x \right)^T u_1 = x^T u_1 - x^T u_1 u_1^T u_1 = x^T u_1 - x^T u_1 = 0$$

Step 2: establish relationship and complete proof

$$\sum_{i=1}^{N} \| x_i - uu^T x_i \|^2 = \sum_{i=1}^{N} (x_i - uu^T x_i)^T (x_i - uu^T x_i) = \sum_{i=1}^{N} (x_i^T x_i - 2x_i^T uu^T x_i + x_i^T uu^T uu^T x_i)$$

$$= \sum_{i=1}^{N} (x_i^T x_i - x_i^T uu^T x_i) = \sum_{i=1}^{N} (x_i^T x_i) - \sum_{i=1}^{N} (x_i^T uu^T x_i)$$
Interpretation Continued

Find the $u_1$ that minimizes the residual squared norm:

$$u_1 u_1^T x$$
Solving PCA
(Iterative Algorithm)

• Given: \( X = [x_1, \ldots, x_N] \in \mathbb{R}^{D \times N} \) Assuming zero mean

• Init: \( X_1 = X \)

• For \( d=1, \ldots, D \)
  – Solve: \( u_d = \arg\min_{u: u^T u=1} \left\| X_d - uu^T X_d \right\|_{\text{Fro}}^2 \)
  – Update: \( X_{d+1} = X_d - u_d u_d^T X_d \)
Property of PCA

\[ XX^T = U \Lambda U^T \]

• The first K columns of U are guaranteed to be the K-dimensional subspace that captures the most variability of X

• We just proved K=1 a few slides ago
Dimensionality Reduction

- Solve PCA:
  \[ XX^T = U \Lambda U^T \]

- Use first \( K \) columns of \( U \) to create \( K \)-dim representation:
  \[ x' = U_{1:K}^T x \]

- This creates a compact summary of original dataset
  - E.g., \( K = 50, \ D = 1,000,000 \)
Example: Eigenfaces

PCA on a corpus of faces.
Every pixel is a "feature"
Visualizing the top Eigenvectors of U

http://www.cs.princeton.edu/~cdecoro/eigenfaces/
Example: Eigenfaces

Visualizing Projection using top K Eigenvectors: $U_{1:K} U_{1:K}^T x$

http://www.cs.princeton.edu/~cdecoro/eigenfaces/
CS 155 Eigenfaces

Avg Face
Singular Value Decomposition

\[ X = U \Sigma V^T \]

- SVD operates on \( X \), as opposed to \( XX^T \)
- Equivalence between SVD & PCA
  \[
  XX^T = (U \Sigma V^T)(U \Sigma V^T)^T = U \Sigma V^T V \Sigma U^T = U \Sigma^2 U^T
  \]
- \( V \) corresponds to new representation \( x' \)
Eigenfaces Step 1

- Flatten each image into vector

Each Column is Image

HxWx3

(3*H*W)xN

225000-dimensional!
Eigenfaces Step 2

- Mean center

\[ X' = X - \text{Mean} \]

Per-column subtraction
Eigenfaces Step 3

• Singular Value Decomposition: \( X' = U \Sigma V^T \)
Eigenfaces Step 4

- Merging $\Sigma$ into $U$ and $V$: $X' = U\Sigma V^T = U'V'^T$
Interpreting U & V

• Each col of $U'$ is an “Eigenface”
• Each col of $V'^T$ = coefficients of a student
Lecture 9: Clustering & Dimensionality Reduction
Limitations of Eigenfaces

• Each dimension is a pixel (\& color channel)
  – Not semantically meaningful
  – Squared reconstruction error in pixel space

• Suppose each dimension had more meaning
  – E.g., \( \text{dim 1} = \text{location of left eye} \)
  – Then U components would have cleaner visualization
Summary

• Clustering & PCA (and SVD) reduce the dimensionality of data representation.

• For each data point
  – Store K numbers
  – Cluster membership probabilities
  – Coefficients in K-dimensional projection

• Nice visualization & interpretation?
  – Depends on semantics of raw dimensions...
Next 2 Lectures

• Latent Factor Models

• Matrix Factorization with Missing Values
  – E.g., the “Netflix Problem”

• Embeddings