Lecture 12:
Clustering & Dimensionality Reduction
Miniproject 2

• Train HMM on dataset of sonnets
  – No need to implement your own

• Generate new sonnets
  – I.e., sample from trained HMM

• Work in teams
Topic Overview

Supervised Learning
- Linear Models
- Overfitting
- Loss Functions
- Non-Linear Models
- Learning Algorithms & Optimization
- Probabilistic Modeling

Unsupervised Learning
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

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

• **Goal:** find hidden structure/patterns
  – E.g., hidden structure is a clustering of data
  – Previously: generative model of \( P(x) \)
  – I.e., a low dimensional summary of the data
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)
Centroid Based Clustering
(K-Means)
K-Means Objective

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

\[
\text{argmin} \sum_{k} \sum_{x \in C_k} \| x - c_k \|^2
\]

\[
\text{argmin} \sum_{k} \left| C_k \right| \text{var}(C_k)
\]

Equivalent!
EM Algorithm for K-Means

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

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

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

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

\[
\text{argmin}_{S = C_1 \cup \ldots \cup C_K, \{c_1, \ldots, c_K\}} \sum_k \sum_{x \in C_k} \|x - c_k\|^2 \quad S = \{x_i\}_{i=1}^N
\]

- For each \(c_k\):
  - Compute \(c_k = \text{mean}(C_k)\)
Aside: Gaussian Mixture Models

- Each data point is associated with a membership to a Gaussian distribution
  - Denoted by z variable

- 1D Example with 3 Gaussians

"Nonbayesian-gaussian-mixture" by Benwing –
Created using LaTeX, TikZ. Licensed under CC BY 3.0 via Commons
Aside: Gaussian Mixture Models

\[ P(x \in C_k | c_1, \ldots, c_K) = \frac{\exp\left\{ -\|x - c_k\|^2 / 2\sigma^2 \right\}}{\sum_{k'} \exp\left\{ -\|x - c_{k'}\|^2 / 2\sigma^2 \right\} } \]

\[ \propto \exp\left\{ -\|x - c_k\|^2 / 2\sigma^2 \right\} \]

- Prob of cluster membership proportional to \( \exp(-\text{dist}^2/2\sigma^2) \)

- “Sharpness” of distribution increases as \( \sigma \) decreases

- Converges to K-Means as \( \sigma \) goes to 0
Aside: Gaussian Mixture Models

\[ \text{argmax} \prod_{x \in S} P(x) = \prod_{x \in S} \sum_{k} P(x \in C_k) P(k) \quad S = \{x_i\}_{i=1}^N \]

- E-Step: Estimate probabilities
- M-Step: Maximize model parameters \(c_1, \ldots, c_k\)

Prior on each Gaussian mixture (can assume = 1/K for simplicity)

\[ P(x \in C_k | c_1, \ldots, c_K) = \frac{\exp \left\{ -\|x - c_k\|^2 / 2\sigma^2 \right\}}{\sum_{k'} \exp \left\{ -\|x - c_{k'}\|^2 / 2\sigma^2 \right\} } \]

Lecture 12: Clustering & Dimensionality Reduction
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
    – http://en.wikipedia.org/wiki/Kruskal%27s_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

Lecture 12: Clustering & Dimensionality Reduction
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!

Lecture 12: Clustering & Dimensionality Reduction
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

• \( x' = U^T x, \quad x = U x' \)

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

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

Summarize Using 1 Feature?
Principal Component Analysis

Summarize Using 1 Feature?

\[ u_1^T x \]

Works with arbitrary subsets of columns of orthogonal \( U \)

E.g., \( U' = [u_1, u_5, u_{20}] \)
PCA Formal Definition

• Define $M$=matrix of all data:

$$X = \begin{bmatrix} x_1, \ldots, x_N \end{bmatrix} \in \mathbb{R}^{D \times N}$$

• Mean center:

$$\bar{X} = X - \begin{bmatrix} \bar{x}, \ldots, \bar{x} \end{bmatrix}$$

• 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
  
  \[ \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_D \]

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

Feature Covariance Matrix:

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

Assuming zero mean

• \( \Sigma_{dd'} \) 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:

\[
\mathbf{u}_1 = \arg\min_{\mathbf{u}: \mathbf{u}^T \mathbf{u} = 1} \sum_{i=1}^{N} \left\| \mathbf{x}_i - \mathbf{u} \mathbf{u}^T \mathbf{x}_i \right\|^2
\]

"Residual"

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

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

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

Residual: $x - u_1 u_1^T x$

$$ (x - u_1 u_1^T x)^T u_1 = x^T u_1 - x^T u_1 u_1^T u_1 = x^T u_1 - x^T u_1 = 0 $$

Step 2: establish relationship and complete proof

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

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

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

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

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

• Init: \( X_1 = X \)

• For \( d=1,\ldots,D \)
  
  – Solve: \( u_d = \arg \min_{u: \, u^T u = 1} \| X_d - uu^T X_d \|_{Fro}^2 \)
  
  – 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:

\[ \mathbf{XX}^T = \mathbf{U} \Lambda \mathbf{U}^T \]

• Use first K columns of U to create K-dim representation:

\[ \mathbf{x}' = \mathbf{U}^T_{1:K} \mathbf{x} \]

• 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
Lecture 12: Clustering & Dimensionality Reduction
Singular Value Decomposition

\[ X = UΣV^T \]

- SVD operates on \( X \), as opposed to \( XX^T \)
- Equivalence between SVD & PCA

\[ XX^T = (UΣV^T)(UΣV^T)^T = UΣV^TVΣU^T = UΣ^2U^T \]
- \( V \) corresponds to new representation \( x' \)
Eigenfaces Step 1

- Flatten each image into a vector

\[ \text{HxWx3} \rightarrow (3\times H \times W) \times N \]

Each Column is Image

225000-dimensional!
Eigenfaces Step 2

• Mean center

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

Per-column subtraction
• 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 12: 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., dim 1 = 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 Lecture

• Latent Factor Models

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

• No Recitation Today