10-701 Introduction to Machine Learning

Principal Component Analysis and Dimensionality Reduction

Readings:
Bishop Ch. 12
Murphy Ch. 12

Matt Gormley
Lecture 14
October 24, 2016
Reminders

• Homework 3:
  – due 10/24/16 (tonight)
DIMENSIONALITY REDUCTION
Big & High-Dimensional Data

• High-Dimensions = Lot of Features

Document classification
Features per document =
thousands of words/unigrams
millions of bigrams, contextual information

Surveys - Netflix
480189 users x 17770 movies

<table>
<thead>
<tr>
<th></th>
<th>movie 1</th>
<th>movie 2</th>
<th>movie 3</th>
<th>movie 4</th>
<th>movie 5</th>
<th>movie 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tom</td>
<td>5</td>
<td>?</td>
<td>?</td>
<td>1</td>
<td>3</td>
<td>?</td>
</tr>
<tr>
<td>George</td>
<td>?</td>
<td>?</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Susan</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>?</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>Beth</td>
<td>4</td>
<td>3</td>
<td>?</td>
<td>2</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>
Big & High-Dimensional Data

- High-Dimensions = Lot of Features

MEG Brain Imaging
120 locations x 500 time points
x 20 objects

Or any high-dimensional image data
• **Big & High-Dimensional Data.**

• Useful to learn lower dimensional representations of the data.
Learning Representations

PCA, Kernel PCA, ICA: Powerful unsupervised learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.

Useful for:

• Visualization

• More efficient use of resources (e.g., time, memory, communication)

• Statistical: fewer dimensions $\rightarrow$ better generalization

• Noise removal (improving data quality)

• Further processing by machine learning algorithms
Principal Component Analysis (PCA)

**What is PCA:** Unsupervised technique for extracting variance structure from high dimensional datasets.

- PCA is an orthogonal projection or transformation of the data into a (possibly lower dimensional) subspace so that the variance of the projected data is maximized.
Principal Component Analysis (PCA)

Intrinsically lower dimensional than the dimension of the ambient space.

If we rotate data, again only one coordinate is more important.

Only one relevant feature

Both features are relevant

Question: Can we transform the features so that we only need to preserve one latent feature?
Principal Component Analysis (PCA)

In case where data lies on or near a low $d$-dimensional linear subspace, axes of this subspace are an effective representation of the data.

Identifying the axes is known as Principal Components Analysis, and can be obtained by using classic matrix computation tools (Eigen or Singular Value Decomposition).

Slide from Nina Balcan
2D Gaussian dataset

Slide from Barnabas Poczos
1st PCA axis
2\textsuperscript{nd} PCA axis

Slide from Barnabas Poczos
PCA ALGORITHMS
PCA algorithm I (sequential)

Given the centered data \( \{x_1, \ldots, x_m\} \), compute the principal vectors:

\[
  w_1 = \arg \max_{\|w\|=1} \frac{1}{m} \sum_{i=1}^{m} \{ (w^T x_i)^2 \} \quad 1^{\text{st}} \text{ PCA vector}
\]

We maximize the variance of projection of \( x \)

\[
  w_k = \arg \max_{\|w\|=1} \frac{1}{m} \sum_{i=1}^{m} \{ [w^T (x_i - \sum_{j=1}^{k-1} w_j w_j^T x_i)]^2 \} \quad k^{\text{th}} \text{ PCA vector}
\]

We maximize the variance of the projection in the residual subspace

\[
  x' = w_1(w_1^T x) + w_2(w_2^T x)
\]
Maximizing the Variance

• Consider the two projections below
• Which maximizes the variance?

Option A

Option B

Figures from Andrew Ng (CS229 Lecture Notes)
PCA algorithm I (sequential)

Given the centered data \( \{x_1, \ldots, x_m\} \), compute the principal vectors:

\[
  w_1 = \arg \max_{\|w\|=1} \frac{1}{m} \sum_{i=1}^{m} \left\{ (w^T x_i)^2 \right\} \quad 1\text{st PCA vector}
\]

We maximize the variance of projection of \( x \)

\[
  w_k = \arg \max_{\|w\|=1} \frac{1}{m} \sum_{i=1}^{m} \left\{ [w^T (x_i - \sum_{j=1}^{k-1} w_j w_j^T x_i)]^2 \right\} \quad k\text{th PCA vector}
\]

We maximize the variance of the projection in the residual subspace

\[
x' = w_1 (w_1^T x) + w_2 (w_2^T x)
\]
PCA algorithm II
(sample covariance matrix)

• Given data \{x_1, \ldots, x_m\}, compute covariance matrix \( \Sigma \)

\[
\Sigma = \frac{1}{m} \sum_{i=1}^{m} (x_i - \bar{x})(x - \bar{x})^T
\]

where \[
\bar{x} = \frac{1}{m} \sum_{i=1}^{m} x_i
\]

• **PCA** basis vectors = the eigenvectors of \( \Sigma \)

We get the eigenvectors using an eigendecomposition. Power iteration (Von Mises iteration is a standard algorithm for this)

• Larger eigenvalue \( \Rightarrow \) more important eigenvectors
Why the Eigenvectors?

Maximise \( u^T X X^T u \)

s.t \( u^T u = 1 \)

Construct Langrangian \( u^T X X^T u - \lambda u^T u \)

Vector of partial derivatives set to zero
\[ x x^T u - \lambda u = (x x^T - \lambda I) u = 0 \]

As \( u \neq 0 \) then \( u \) must be an eigenvector of \( X X^T \) with eigenvalue \( \lambda \)
Eigenvalues & Eigenvectors

- For symmetric matrices, eigenvectors for distinct eigenvalues are orthogonal:
  \[ S v_{\{1,2\}} = \lambda_{\{1,2\}} v_{\{1,2\}}, \text{ and } \lambda_1 \neq \lambda_2 \implies v_1 \cdot v_2 = 0 \]

- All eigenvalues of a real symmetric matrix are real.
  \[ \text{if } |S - \lambda I| = 0 \text{ and } S = S^T \implies \lambda \in \mathbb{R} \]

- All eigenvalues of a positive semidefinite matrix are non-negative
  \[ \forall w \in \mathbb{R}^n, w^T Sw \geq 0, \text{ then if } Sv = \lambda v \implies \lambda \geq 0 \]
Eigen/diagonal Decomposition

- Let $S \in \mathbb{R}^{m \times m}$ be a square matrix with $m$ linearly independent eigenvectors (a “non-defective” matrix).

- **Theorem**: Exists an eigen decomposition $S = U\Lambda U^{-1}$

  - Columns of $U$ are eigenvectors of $S$
  - Diagonal elements of $\Lambda$ are eigenvalues of $S$

  $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m)$, $\lambda_i \geq \lambda_{i+1}$

(Cf. matrix diagonalization theorem)
How Many PCs?

- For n original dimensions, sample covariance matrix is nxn, and has up to n eigenvectors. So n PCs.
- Where does dimensionality reduction come from?
  Can *ignore* the components of lesser significance.

You do lose some information, but if the eigenvalues are small, you don’t lose much
- n dimensions in original data
- calculate n eigenvectors and eigenvalues
- choose only the first p eigenvectors, based on their eigenvalues
- final data set has only p dimensions
Eigen/diagonal Decomposition

- Let $S \in \mathbb{R}^{m \times m}$ be a square matrix with $m$ linearly independent eigenvectors (a “non-defective” matrix)

- **Theorem**: Exists an eigen decomposition

  $S = U \Lambda U^{-1}$

  (cf. matrix diagonalization theorem)

- Columns of $U$ are eigenvectors of $S$

- Diagonal elements of $\Lambda$ are eigenvalues of $S$

\[ \Lambda = \text{diag}(\lambda_1, \ldots, \lambda_m), \quad \lambda_i \geq \lambda_{i+1} \]

Unique for distinct eigenvalues
Singular Value Decomposition

For an \( m \times n \) matrix \( A \) of rank \( r \) there exists a factorization (Singular Value Decomposition = SVD) as follows:

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

- The columns of \( U \) are orthogonal eigenvectors of \( AA^T \).
- The columns of \( V \) are orthogonal eigenvectors of \( A^T A \).
- Eigenvalues \( \lambda_1 \ldots \lambda_r \) of \( AA^T \) are the eigenvalues of \( A^T A \).

\[
\sigma_i = \sqrt{\lambda_i}
\]

\[\Sigma = \text{diag}(\sigma_1 \ldots \sigma_r)\]

Singular values.
PCA: Two Interpretations

E.g., for the first component.

**Maximum Variance Direction:** $1^{st}$ PC a vector $v$ such that projection on to this vector capture maximum variance in the data (out of all possible one dimensional projections)

$$
\frac{1}{n} \sum_{i=1}^{n} (v^T x_i)^2 = v^T X X^T v
$$

**Minimum Reconstruction Error:** $1^{st}$ PC a vector $v$ such that projection on to this vector yields minimum MSE reconstruction

$$
\frac{1}{n} \sum_{i=1}^{n} \|x_i - (v^T x_i) v\|^2
$$
PCA: Two Interpretations

E.g., for the first component.

**Maximum Variance Direction:** 1st PC a vector \( v \) such that projection on to this vector capture maximum variance in the data (out of all possible one dimensional projections)

\[
\frac{1}{n} \sum_{i=1}^{n} (v^T x_i)^2 = v^T X X^T v
\]

**Minimum Reconstruction Error:** 1st PC a vector \( v \) such that projection on to this vector yields minimum MSE reconstruction

\[
\frac{1}{n} \sum_{i=1}^{n} \|x_i - (v^T x_i) v\|^2
\]

\( \text{blue}^2 + \text{green}^2 = \text{black}^2 \)

\( \text{black}^2 \) is fixed (it’s just the data)

So, maximizing \( \text{blue}^2 \) is equivalent to minimizing \( \text{green}^2 \)
PCA algorithm III
(SVD of the data matrix)

Singular Value Decomposition of the centered data matrix $X$.

$$X = [x_1, \ldots, x_m] \in \mathbb{R}^{N \times m},$$

$m$: number of instances,
$N$: dimension

$$X_{\text{features} \times \text{samples}} = USV^T$$

$X = \begin{bmatrix}
\text{samples} \\
\text{significant} \\
\text{noise}
\end{bmatrix}$

$U = \begin{bmatrix}
\text{significant} \\
\text{noise}
\end{bmatrix}$

$S = \begin{bmatrix}
\text{sig.} & \text{noise} \\
\text{noise} & \text{noise}
\end{bmatrix}$

$V^T = \begin{bmatrix}
\text{significant}
\end{bmatrix}$
PCA algorithm III

- **Columns of U**
  - the principal vectors, \( \{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \} \)
  - orthogonal and has unit norm – so \( \mathbf{U}^T \mathbf{U} = \mathbf{I} \)
  - Can reconstruct the data using linear combinations of \( \{ \mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(k)} \} \)

- **Matrix S**
  - Diagonal
  - Shows importance of each eigenvector

- **Columns of \( \mathbf{V}^T \)**
  - The coefficients for reconstructing the samples
SVD and PCA

- The first root is called the principal eigenvalue which has an associated orthonormal ($u^T u = 1$) eigenvector $u$
- Subsequent roots are ordered such that $\lambda_1 > \lambda_2 > \ldots > \lambda_M$ with $\text{rank}(D)$ non-zero values.
- Eigenvectors form an orthonormal basis i.e. $u_i^T u_j = \delta_{ij}$
- The eigenvalue decomposition of $XX^T = U \Sigma U^T$
- where $U = [u_1, u_2, \ldots, u_M]$ and $\Sigma = \text{diag}[\lambda_1, \lambda_2, \ldots, \lambda_M]$
- Similarly the eigenvalue decomposition of $X^TX = V \Sigma V^T$
- The SVD is closely related to the above $X = U \Sigma^{1/2} V^T$
- The left eigenvectors $U$, right eigenvectors $V$,
- singular values = square root of eigenvalues.
Low-rank Approximation

- Solution via SVD

\[ A_k = U \text{ diag}(\sigma_1, \ldots, \sigma_k, 0, \ldots, 0)V^T \]

set smallest \( r-k \) singular values to zero

\[ A_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T \quad \text{column notation: sum of rank 1 matrices} \]
Approximation error

- How good (bad) is this approximation?
- It’s the best possible, measured by the Frobenius norm of the error:

\[
\min_{X: \text{rank}(X) = k} \left\| A - X \right\|_F = \left\| A - A_k \right\|_F = \sigma_{k+1}
\]

where the \( \sigma_i \) are ordered such that \( \sigma_i \geq \sigma_{i+1} \).
Suggests why Frobenius error drops as \( k \) increased.
Slides from Barnabas Poczos

Original sources include:
  • Karl Booksh Research group
  • Tom Mitchell
  • Ron Parr

PCA EXAMPLES
Face recognition
Challenge: Facial Recognition

- Want to identify specific person, based on facial image
- Robust to glasses, lighting,…
  → Can’t just use the given 256 x 256 pixels
Applying PCA: Eigenfaces

**Method A:** Build a PCA subspace for each person and check which subspace can reconstruct the test image the best

**Method B:** Build one PCA database for the whole dataset and then classify based on the weights.

- Example data set: Images of faces
  - Famous Eigenface approach
    [Turk & Pentland], [Sirovich & Kirby]

- Each face $\mathbf{x}$ is ...

- $256 \times 256$ values (luminance at location)
  - $\mathbf{x}$ in $\mathbb{R}^{256 \times 256}$ (view as 64K dim vector)

- Form $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]$ centered data mtx

- Compute $\Sigma = \mathbf{X}\mathbf{X}^T$

- Problem: $\Sigma$ is $64K \times 64K$ ... HUGE!!!
Computational Complexity

• Suppose $m$ instances, each of size $N$
  – Eigenfaces: $m=500$ faces, each of size $N=64K$
• Given $N \times N$ covariance matrix $\Sigma$, can compute
  – all $N$ eigenvectors/eigenvalues in $O(N^3)$
  – first $k$ eigenvectors/eigenvalues in $O(k \, N^2)$

• But if $N=64K$, EXPENSIVE!
A Clever Workaround

• Note that $m << 64K$
• Use $L = X^T X$ instead of $\Sigma = X X^T$
• If $v$ is eigenvector of $L$
  then $X v$ is eigenvector of $\Sigma$

Proof:

$L \ v = \gamma \ v$

$X^T X \ v = \gamma \ v$

$X (X^T X \ v) = X(\gamma \ v) = \gamma Xv$

$(XX^T)X \ v = \gamma (Xv)$

$\Sigma (Xv) = \gamma (Xv)$
Principle Components (Method B)
Reconstructing... (Method B)

- ... faster if train with...
  - only people w/out glasses
  - same lighting conditions
Shortcomings

- Requires carefully controlled data:
  - All faces centered in frame
  - Same size
  - Some sensitivity to angle
- Alternative:
  - “Learn” one set of PCA vectors for each angle
  - Use the one with lowest error
- Method is completely knowledge free
  - (sometimes this is good!)
  - Doesn’t know that faces are wrapped around 3D objects (heads)
  - Makes no effort to preserve class distinctions
Facial expression recognition
Happiness subspace (method A)
Disgust subspace (method A)
Facial Expression Recognition
Movies (method A)
Facial Expression Recognition
Movies (method A)
Facial Expression Recognition
Movies (method A)
Image Compression
• Divide the original 372x492 image into patches:
  • Each patch is an instance that contains 12x12 pixels on a grid
• View each as a 144-D vector
$L_2$ error and PCA dim
PCA compression: 144D ) 60D
PCA compression: 144D ) 16D
16 most important eigenvectors
PCA compression: 144D \to 6D
6 most important eigenvectors
PCA compression: 144D \ 3D
3 most important eigenvectors
PCA compression: 144D \ 1D
60 most important eigenvectors

Looks like the discrete cosine bases of JPG!...
2D Discrete Cosine Basis

Noise Filtering
Noise Filtering, Auto-Encoder...
Denoised image using 15 PCA components