Principal Component Analysis (PCA) + K-Means

Matt Gormley
Lecture 25
Apr. 18, 2022
Reminders

• Homework 8: Reinforcement Learning
  – Out: Tue, Apr. 12
  – Due: Thu, Apr. 21 at 11:59pm
DIMENSIONALITY REDUCTION
High Dimension Data

Examples of high dimensional data:
– High resolution images (millions of pixels)
High Dimension Data

Examples of high dimensional data:

– Multilingual News Stories
  (vocabulary of hundreds of thousands of words)
High Dimension Data

Examples of high dimensional data:
– Brain Imaging Data (100s of MBs per scan)

Image from (Wehbe et al., 2014)
High Dimension Data

Examples of high dimensional data:
– Customer Purchase Data
Learning Representations

**Dimensionality Reduction Algorithms:**
Powerful (often unsupervised) learning techniques for extracting hidden (potentially lower dimensional) structure from high dimensional datasets.

**Examples:**
PCA, Kernel PCA, ICA, CCA, t-SNE, Autoencoders, Matrix Factorization

**Useful for:**
- Visualization
- More efficient use of resources (e.g., time, memory, communication)
- Statistical: fewer dimensions $\rightarrow$ better generalization
- Noise removal (improving data quality)

Slide adapted from Nina Balcan
Shortcut Example

https://www.youtube.com/watch?v=MlJN9pEfPfE

Photo from https://www.springcarnival.org/booth.shtml
This section in one slide...

1. Dimensionality reduction:

2. Random Projection:

3. Definition of PCA:

Choose the matrix $V$ that either...
1. minimizes reconstruction error
2. consists of the $K$ eigenvectors with largest eigenvalue

The above are equivalent definitions.

4. Algorithm for PCA:

The option we'll focus on:

Run Singular Value Decomposition (SVD) to obtain all the eigenvectors. Keep just the top-$K$ to form $V$. Play some tricks to keep things efficient.

5. An Example
DIMENSIONALITY REDUCTION BY RANDOM PROJECTION
Random Projection

Whiteboard

– Random linear projection
Johnson-Lindenstrauss Lemma

Q: But how could we ever hope to preserve any useful information by randomly projecting into a low-dimensional space?

A: Even random projection enjoys some surprisingly impressive properties. In fact, a standard of the J-L lemma starts by assuming we have a random linear projection obtained by sampling each matrix entry from a Gaussian(0,1).

An Elementary Proof of a Theorem of Johnson and Lindenstrauss

Sanjoy Dasgupta, Anupam Gupta

**Abstract:** A result of Johnson and Lindenstrauss [13] shows that a set of $n$ points in high dimensional Euclidean space can be mapped into an $O(\log n/\epsilon^2)$-dimensional Euclidean space such that the distance between any two points changes by only a factor of $(1 \pm \epsilon)$. In this note, we prove this theorem using elementary probabilistic techniques. © 2003 Wiley Periodicals, Inc. Random Struct. Alg., 22: 60–65, 2002

DEFINITION OF PRINCIPAL COMPONENT ANALYSIS (PCA)
Principal Component Analysis (PCA)

- **Assumption**: the data lies on a low K-dimensional linear subspace
- **Goal**: identify the axes of that subspace, and project each point onto hyperplane
- **Algorithm**: find the K eigenvectors with largest eigenvalue using classic matrix decomposition tools

![PCA Example: 2D Gaussian Data](https://commons.wikimedia.org/wiki/File:Scatter_diagram_for_quality_characteristic_XXX.svg)
Data for PCA

$$\mathcal{D} = \{ \mathbf{x}^{(i)} \}_{i=1}^{N}$$

$$\mathbf{x}^{(i)} \in \mathbb{R}^M$$

$$\mathbf{X} = \begin{bmatrix} (\mathbf{x}^{(1)})^T \\ (\mathbf{x}^{(2)})^T \\ \vdots \\ (\mathbf{x}^{(N)})^T \end{bmatrix}$$

We assume the data is **centered**

$$\mu = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}^{(i)} = 0$$

**Q:** What if your data is **not** centered?

**A:** Subtract off the sample mean

$$\tilde{\mathbf{x}}^{(i)} = \mathbf{x}^{(i)} - \mu, \quad \forall i$$
The sample covariance matrix $\Sigma \in \mathbb{R}^{M \times M}$ is given by:

$$
\Sigma_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_j^{(i)} - \mu_j)(x_k^{(i)} - \mu_k)
$$

Since the data matrix is centered, we rewrite as:

$$
\Sigma = \frac{1}{N} X^T X
$$

$$
X = \begin{bmatrix}
(x^{(1)})^T \\
(x^{(2)})^T \\
\vdots \\
(x^{(N)})^T
\end{bmatrix}
$$
Principal Component Analysis (PCA)

Linear Projection:
Given KxM matrix $V$, and Mx1 vector $x^{(i)}$ we obtain the Kx1 projection $u^{(i)}$ by:
$$u^{(i)} = V^T x^{(i)}$$

Definition of PCA:
PCA repeatedly chooses a next vector $v_j$ that minimizes the reconstruction error s.t. $v_j$ is orthogonal to $v_1, v_2, ..., v_{j-1}$.

Vector $v_j$ is called the $j$th principal component.

Notice: Two vectors $a$ and $b$ are orthogonal if $a^T b = 0$.

$\rightarrow$ the K-dimensions in PCA are uncorrelated
Vector Projection

Recall: Projection

\[ \vec{x} \]
\[ \vec{v} \]
\[ a \]

**length of projection of \( \vec{x} \) onto \( \vec{v} \)**

\[ a = \frac{\vec{v} \cdot \vec{x}}{||\vec{v}||_2} \]

if \( ||\vec{v}||_2 = 1 \)

otherwise

**projection of \( \vec{x} \) onto \( \vec{v} \)**

\[ \overrightarrow{\vec{u}} = a \vec{v} = \left( \frac{\vec{v} \cdot \vec{x}}{||\vec{v}||_2^2} \right) \vec{v} \]

if \( ||\vec{v}||_2 = 1 \)

otherwise
Principal Component Analysis (PCA)

*Whiteboard*

– Objective functions for PCA
Maximizing the Variance

**Quiz:** Consider the two projections below

1. Which maximizes the variance?
2. Which minimizes the reconstruction error?

![Option A Diagram](image1)
![Option B Diagram](image2)
For a square matrix $A$ (n x n matrix), the vector $v$ (n x 1 matrix) is an **eigenvector** iff there exists **eigenvalue** $\lambda$ (scalar) such that:

$$Av = \lambda v$$

The linear transformation $A$ is only stretching vector $v$.

That is, $\lambda v$ is a *scalar multiple* of $v$. 
Principal Component Analysis (PCA)

Whiteboard

– PCA, Eigenvectors, and Eigenvalues
**Equivalence of Maximizing Variance and Minimizing Reconstruction Error**

**Claim:** Minimizing the reconstruction error is equivalent to maximizing the variance.

**Proof:** First, note that:

\[
||\mathbf{x}^{(i)} - (\mathbf{v}^T \mathbf{x}^{(i)})\mathbf{v}||^2 = ||\mathbf{x}^{(i)}||^2 - (\mathbf{v}^T \mathbf{x}^{(i)})^2
\]

since \(\mathbf{v}^T \mathbf{v} = ||\mathbf{v}||^2 = 1\).

Substituting into the minimization problem, and removing the extraneous terms, we obtain the maximization problem.

\[
\mathbf{v}^* = \arg\min_{\mathbf{v}:||\mathbf{v}||^2 = 1} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}^{(i)} - (\mathbf{v}^T \mathbf{x}^{(i)})\mathbf{v}||^2
\]

\[
= \arg\min_{\mathbf{v}:||\mathbf{v}||^2 = 1} \frac{1}{N} \sum_{i=1}^{N} ||\mathbf{x}^{(i)}||^2 - (\mathbf{v}^T \mathbf{x}^{(i)})^2
\]

\[
= \arg\max_{\mathbf{v}:||\mathbf{v}||^2 = 1} \frac{1}{N} \sum_{i=1}^{N} (\mathbf{v}^T \mathbf{x}^{(i)})^2
\]
The First Principal Component

**Claim:** The vector that maximizes the variances is the eigenvector of $\Sigma$ with largest eigenvalue.

**Proof Sketch:** To find the first principal component, we wish to solve the following constrained optimization problem (variance minimization).

$$v_1 = \arg\max_{v: ||v||^2 = 1} v^T \Sigma v$$  \hspace{1cm} (1)

So we turn to the method of Lagrange multipliers. The Lagrangian is:

$$\mathcal{L}(v, \lambda) = v^T \Sigma v - \lambda(v^T v - 1)$$  \hspace{1cm} (2)

Taking the derivative of the Lagrangian and setting to zero gives:

$$\frac{d}{dv} (v^T \Sigma v - \lambda(v^T v - 1)) = 0$$  \hspace{1cm} (3)

$$\Sigma v - \lambda v = 0$$  \hspace{1cm} (4)

$$\Sigma v = \lambda v$$  \hspace{1cm} (5)

Recall: For a square matrix $A$, the vector $v$ is an eigenvector iff there exists eigenvalue $\lambda$ such that:

$$Av = \lambda v$$  \hspace{1cm} (6)

Rewriting the objective of the maximization shows that not only will the optimal vector $v_1$ be an eigenvector, it will be one with maximal eigenvalue:

$$v^T \Sigma v = v^T \lambda v$$  \hspace{1cm} (7)

$$= \lambda v^T v$$  \hspace{1cm} (8)

$$= \lambda ||v||^2$$  \hspace{1cm} (9)

$$= \lambda$$  \hspace{1cm} (10)
PCA: the First Principal Component

To find the first principal component, we wish to solve the following constrained optimization problem (variance minimization).

\[ \mathbf{v}_1 = \arg\max_{\mathbf{v}: ||\mathbf{v}||^2 = 1} \mathbf{v}^T \Sigma \mathbf{v} \]  

(1)

So we turn to the method of Lagrange multipliers. The Lagrangian is:

\[ L(\mathbf{v}, \lambda) = \mathbf{v}^T \Sigma \mathbf{v} - \lambda (\mathbf{v}^T \mathbf{v} - 1) \]  

(2)

Taking the derivative of the Lagrangian and setting to zero gives:

\[ \frac{d}{d\mathbf{v}} \left( \mathbf{v}^T \Sigma \mathbf{v} - \lambda (\mathbf{v}^T \mathbf{v} - 1) \right) = 0 \]  

(3)

\[ \Sigma \mathbf{v} - \lambda \mathbf{v} = 0 \]  

(4)

\[ \Sigma \mathbf{v} = \lambda \mathbf{v} \]  

(5)

Recall: For a square matrix \( \mathbf{A} \), the vector \( \mathbf{v} \) is an \textit{eigenvector} iff there exists an \textit{eigenvalue} \( \lambda \) such that:

\[ \mathbf{A} \mathbf{v} = \lambda \mathbf{v} \]  

(6)
Principal Component Analysis (PCA)

\[(X X^T) v = \lambda v\], so \(v\) (the first PC) is the eigenvector of sample correlation/covariance matrix \(X X^T\)

Sample variance of projection \(v^T X X^T v = \lambda v^T v = \lambda\)

Thus, the eigenvalue \(\lambda\) denotes the amount of variability captured along that dimension (aka amount of energy along that dimension).

Eigenvalues \(\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots\)

- The 1\(^{st}\) PC \(v_1\) is the the eigenvector of the sample covariance matrix \(X X^T\) associated with the largest eigenvalue
- The 2nd PC \(v_2\) is the the eigenvector of the sample covariance matrix \(X X^T\) associated with the second largest eigenvalue
- And so on ...

Slide from Nina Balcan
ALGORITHMS FOR PCA
Algorithms for PCA

How do we find principal components (i.e. eigenvectors)?

• Power iteration (aka. Von Mises iteration)
  – finds each principal component one at a time in order

• Singular Value Decomposition (SVD)
  – finds all the principal components at once
  – two options:
    • Option A: run SVD on $X^T X$
    • Option B: run SVD on $X$
      (not obvious why Option B should work…)

• Stochastic Methods (approximate)
  – very efficient for high dimensional datasets with lots of points
SVD

\[ X = USV^T \]

Data \( X \), one row per data point

\( US \) gives coordinates of rows of \( X \) in the space of principle components

\( S \) is diagonal, \( S_k > S_{k+1} \), \( S_k^2 \) is kth largest eigenvalue

Rows of \( V^T \) are unit length eigenvectors of \( X^TX \)

If cols of \( X \) have zero mean, then \( X^TX = c \Sigma \) and eigenvects are the Principle Components

[from Wall et al., 2003]
Singular Value Decomposition

To generate principle components:

- Subtract mean $\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x^n$ from each data point, to create zero-centered data
- Create matrix $X$ with one row vector per (zero centered) data point
- Solve SVD: $X = USV^T$
- Output Principle components: columns of $V$ (= rows of $V^T$)
  - Eigenvectors in $V$ are sorted from largest to smallest eigenvalues
  - $S$ is diagonal, with $s_k^2$ giving eigenvalue for $k$th eigenvector
Singular Value Decomposition

To project a point (column vector $x$) into PC coordinates:

$$VTx$$

If $x_i$ is $i^{th}$ row of data matrix $X$, then

- $(i^{th}$ row of $US) = VT x_i^T$
- $(US)^T = VT X^T$

To project a column vector $x$ to M dim Principle Components subspace, take just the first M coordinates of $VTx$
How Many PCs?

• For $M$ original dimensions, sample covariance matrix is $M \times M$, and has up to $M$ eigenvectors. So $M$ 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
  – $M$ dimensions in original data
  – calculate $M$ eigenvectors and eigenvalues
  – choose only the first $D$ eigenvectors, based on their eigenvalues
  – final data set has only $D$ dimensions

Variance (%) = ratio of variance along given principal component to total variance of all principal components
PCA EXAMPLES
Projecting MNIST digits

Task Setting:
1. Take 25x25 images of digits and project them down to K components
2. Report percent of variance explained for K components
3. Then project back up to 25x25 image to visualize how much information was preserved
Projecting MNIST digits

**Task Setting:**
1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points
3. Here we look at all ten digits 0 - 9
Projecting MNIST digits

**Task Setting:**
1. Take 25x25 images of digits and project them down to 2 components
2. Plot the 2 dimensional points
3. Here we look at just four digits 0, 1, 2, 3
Learning Objectives

Dimensionality Reduction / PCA

You should be able to...

1. Define the sample mean, sample variance, and sample covariance of a vector-valued dataset
2. Identify examples of high dimensional data and common use cases for dimensionality reduction
3. Draw the principal components of a given toy dataset
4. Establish the equivalence of minimization of reconstruction error with maximization of variance
5. Given a set of principal components, project from high to low dimensional space and do the reverse to produce a reconstruction
6. Explain the connection between PCA, eigenvectors, eigenvalues, and covariance matrix
7. Use common methods in linear algebra to obtain the principal components