Principal Component Analysis (PCA)
Singular Value Decomposition (SVD)

Based on slides from Iyad Batal, Eric Strobl & Milos Hauskrecht

Outline

• Principal Component Analysis (PCA)
• Singular Value Decomposition (SVD)
• Multi-Dimensional Scaling (MDS)
• Non-linear PCA extension:
  • Kernel PCA
Outline

• Principal Component Analysis (PCA)

• Singular Value Decomposition (SVD)

• Multi-Dimensional Scaling (MDS)

• Non-linear extensions:
  • Kernel PCA

Real-World Data

Real world data and information therein may be:

• **Redundant**
  – One variable may carry the same information as the other variable
  – Information covered by a set of variables may overlap

• **Noisy**
  – Some dimensions may not carry any useful information and the variation in that dimension is purely due to noise in the observations

**Important questions:**

• how to reduce the dimensionality of the data
• what is the intrinsic dimensionality of the data?
Example

Three cameras tracking the movement of a ball on a string in 3D space.
- The ball moves in 2D space (one dimension is redundant)
- Information collected by 3 cameras overlap.

PCA

PCA finds a linear projection of data into orthogonal basis system that has the minimum redundancy and preserves the variance in data.

Applications:
- Identify the intrinsic dimensionality of the data
- Lower dimensional representation of data with the smallest reconstruction error.
PCA/SVD applications

- Dimensionality reduction
- Kleinberg/Hits algorithm
- Google/PageRank algorithm (random walk with restart).
- Image-compression (eigen faces)
- Data visualization (by projecting the data on 2D).

Background: eigenvectors

- If $A$ is a square matrix, a non-zero vector $v$ is an eigenvector of $A$ if there is a scalar $\lambda$ (eigenvalue) such that
  \[ Av = \lambda v \]

- Example: \[
\begin{pmatrix}
2 & 3 \\
2 & 1
\end{pmatrix}
\begin{pmatrix}
3 \\
2
\end{pmatrix}
= \begin{pmatrix}
12 \\
8
\end{pmatrix}
= 4 \begin{pmatrix}
3 \\
2
\end{pmatrix}
\]

- If we think of the squared matrix as a transformation matrix, then multiply it with the eigenvector do not change its direction.

*What are the eigenvectors of the identity matrix?*
The Covariance Matrix of X

\[ C_X = \frac{1}{n-1}X^TX \]

**Diagonal terms:** variance
Large values = signal

**Off-diagonal:** covariance
Large values = high redundancy

**Covariance matrix is always symmetric**
\[ C_X^T = \frac{1}{n-1}(X^TX)^T = \frac{1}{n-1}(X^TX) = C_X \]

---

**Matrix decomposition**

**Theorem 1:** if square \( d \times d \) matrix \( S \) is a real and symmetric matrix \( (S = S^T) \) then
\[ S = V\Lambda V^T \]

where \( V = [v_1 \ldots v_d] \) are the eigenvectors of \( S \) and \( \Lambda = diag(\lambda_1, \ldots, \lambda_d) \) are the corresponding eigenvalues.

**Proof:**
\[ SV = V\Lambda \]
\[ [Sv_1, Sv_2, \ldots Sv_d] = [\lambda_1v_1, \lambda_2v_2, \ldots, \lambda_dv_d] \]
\[ SVV^{-1} = V\Lambda V^{-1} \]
\[ S = V\Lambda V^T \]
Covariance matrix decomposition

\[ X^T X = V \Lambda V^T \]

where:
- \( V \) is a matrix of eigenvectors of \( X^T X \) (arranged in columns);
- \( \Lambda \) is a diagonal matrix of corresponding eigenvalues

**Proof:**

\[
\begin{align*}
(X^T X)V &= V \Lambda D \\
(X^T X)VV^T &= V \Lambda V^T \\
X^T X &= V \Lambda V^T \quad \text{since eigenvectors are orthonormal}
\end{align*}
\]

---

Change of Basis

**Assume:**
- \( X \) is an \( n \times d \) data matrix
- **Linear (affine) transformation:** \( A \)

\[ Y = XA \]

where
- \( A \) is a matrix that transforms \( X \) into \( Y \)
- *Columns* of \( A \) are formed by basis vectors that re-express the rows of \( X \) in the new coordinate system
Change of Basis

- But, what is the best “basis” vector?
  - **PCA assumption**: the direction with the largest variance

![Graph showing the basis is just the best fit line]

Goal and Assumptions of PCA

\[ Y = XA \]

**Goal**: Find the best transformation \( A \), so that \( Y \) has the minimal noise and redundancy

**Assumptions**

1) \( A \) contains orthonormal basis vectors (makes computations easier)
2) Covariance matrix captures all the information about \( X \) (only true for exponential family distributions)
PCA Derivation

- $C_Y$ : Covariance of $Y$ expressed in terms of $A$

$$C_Y = \frac{1}{n-1} Y^T Y$$

$$= \frac{1}{n-1} (XA)^T (XA)$$

$$= \frac{1}{n-1} A^T X^T X A$$

$$= \frac{1}{n-1} A^T (X^T X) A$$

---

PCA

- Find the direction for which the variance is maximized:

$$v_1 = \arg \max_{v_1} \text{var}(Xv_1)$$

Subject to: $v_1^T v_1 = 1$

- Rewrite in terms of the covariance matrix:

$$\text{var}(Xv_1) = \frac{1}{N-1} (Xv_1)^T (Xv_1) = v_1^T \frac{1}{N-1} X^T X v_1 = v_1^T C v_1$$

- Solve via constrained optimization:

$$L(v_1, \lambda_1) = v_1^T C v_1 + \lambda_1 (1 - v_1^T v_1)$$
PCA

- Constrained optimization:
  \[ L(v_1, \lambda_1) = v_1^T C v_1 + \lambda_1 (1 - v_1^T v_1) \]

- Gradient with respect to \( v_1 \):
  \[ \frac{dL(v_1, \lambda_1)}{dv_1} = 2Cv_1 - 2\lambda_1 v_1 \Rightarrow Cv_1 = \lambda_1 v_1 \]
  This is the eigenvector problem!

- Multiply by \( v_1^T \):
  \[ \lambda_1 = v_1^T C v_1 \]
  The projection variance is the eigenvalue

---

PCA Derivation

- Assuming \( A = V \), i.e. each column is an eigenvector of \( X^T X \)
  \[
  C_Y = \frac{1}{n-1} V^T (X^T X) V \\
  = \frac{1}{n-1} V^T (VDV^T) V \\
  = \frac{1}{n-1} V^T V D V^T V \\
  = \frac{1}{n-1} V^{-1} V D V^{-1} V \\
  = \frac{1}{n-1} D
  \]

After the transformation of \( X \) with \( V \), the covariance matrix becomes diagonal
PCA as dimensionality reduction

(1) If the data lives in a lower dimensional space \( d' \), then some of the eigenvalues in \( D \) matrix are set to 0

(2) If we want to reduce the dimensionality of the data from \( d \) to some fixed \( k \), we choose the eigenvectors with the \( k \) highest eigenvalues – the dimensions that preserve most of the variance in the data

(3) This selection also minimizes the data reconstruction error (so the best \( k \) dimensions lead to best error).

<table>
<thead>
<tr>
<th>Original</th>
<th>( M = 1 )</th>
<th>( M = 10 )</th>
<th>( M = 50 )</th>
<th>( M = 250 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

PCA for dimensionality reduction

**PCA steps:** transform an \( N \times d \) matrix \( X \) into an \( N \times m \) matrix \( Y \):

- Centralized the data (subtract the mean).
- Calculate the \( d \times d \) covariance matrix: \( C = \frac{1}{N-1}X^TX \) ([different notation from tutorial!!])
  - \( C_{i,j} = \frac{1}{N-1} \sum_{q=1}^{N} X_{q,i} X_{q,j} \)
  - \( C_{i,i} \) (diagonal) is the variance of variable \( i \).
  - \( C_{i,j} \) (off-diagonal) is the covariance between variables \( i \) and \( j \).
- Calculate the eigenvectors of the covariance matrix (**orthonormal**).
- Select \( m \) eigenvectors that correspond to the largest \( m \) eigenvalues to be the new basis.
PCA: example

$X$ : the data matrix with $N=11$ objects and $d=2$ dimensions.

Step 1: subtract the mean and calculate the covariance matrix $C$.

$$C = \begin{pmatrix} 0.716 & 0.615 \\ 0.615 & 0.616 \end{pmatrix}$$
PCA: example

Step 2: Calculate the eigenvectors and eigenvalues of the covariance matrix:

\[ \lambda_1 \approx 1.28, \ v_1 \approx [-0.677 \ -0.735]^T, \ \lambda_2 \approx 0.49, \ v_2 \approx [-0.735 \ 0.677]^T \]

Notice that \( v_1 \) and \( v_2 \) are orthonormal:

\[ |v_1| = 1 \]
\[ |v_2| = 1 \]
\[ v_1 \cdot v_2 = 0 \]

PCA: example

Step 3: project the data

Let \( V = [v_1, \ldots, v_m] \) is \( d \times m \) matrix where the columns \( v_i \) are the eigenvectors corresponding to the largest \( m \) eigenvalues.

The projected data: \( Y = XV \) is \( N \times m \) matrix.

If \( m = d \) (more precisely rank(\( X \))), then there is no loss of information!
Step 3: project the data

\[ \lambda_1 \approx 1.28, \mathbf{v}_1 \approx [ -0.677 \ -0.735 ]^T, \lambda_2 \approx 0.49, \mathbf{v}_2 \approx [ -0.735 \ 0.677 ]^T \]

The eigenvector with the highest eigenvalue is the principle component of the data.

If we are allowed to pick only one dimension, the principle component is the best direction (retain the maximum variance).

Our PC is \( \mathbf{v}_1 \approx [ -0.677 \ -0.735 ]^T \)

---

We lost variance along the other component (lossy compression!)
Outline

- Principal Component Analysis (PCA)
- Singular Value Decomposition (SVD)
- Multi-Dimensional Scaling (MDS)
- Non-linear extensions:
  - Kernel PCA

SVD

Any $N \times d$ matrix $X$ can be uniquely expressed as:

$$X = U \times \Sigma \times V^T$$

- $r$ is the rank of the matrix $X$ (# of linearly independent columns/rows).
- $U$ is a column-orthonormal $N \times r$ matrix.
- $\Sigma$ is a diagonal $r \times r$ matrix where the singular values $\sigma_i$ are sorted in descending order.
- $V$ is a column-orthonormal $d \times r$ matrix.
SVD example

The rank of this matrix $r=2$ because we have 2 types of documents (CS and Medical documents), i.e. 2 concepts.

U: document-to-concept similarity matrix
V: term-to-concept similarity matrix.

Example: $U_{1,1}$ is the weight of CS concept in document $d_1$, $\sigma_1$ is the strength of the CS concept, $V_{1,1}$ is the weight of ‘data’ in the CS concept. $V_{1,2} = 0$ means ‘data’ has zero similarity with the 2nd concept (Medical). **What does $U_{d,1}$ means?**
PCA and SVD relation

**Theorem:** Let $X = U \Sigma V^T$ be the SVD of an $N \times d$ matrix $X$ and $C = \frac{1}{N-1} X^T X$ be the $d \times d$ covariance matrix. The eigenvectors of $C$ are the same as the right singular vectors of $X$.

**Proof:**

$$X^T X = V \Sigma U^T U \Sigma V^T = V \Sigma \Sigma V^T = V \Sigma^2 V^T$$

$$C = V \frac{\Sigma^2}{N-1} V^T$$

But $C$ is symmetric, hence $C = V \Lambda V^T$ (according to theorem1).

Therefore, the eigenvectors of the covariance matrix are the same as matrix $V$ (right singular vectors) and the eigenvalues of $C$ can be computed from the singular values $\lambda_i = \frac{\sigma_i^2}{N-1}$.

---

Summary for PCA and SVD

**Objective:** project an $N \times d$ data matrix $X$ using the largest $m$ principal components $V = [v_1 \ldots v_m]$.

1. zero mean the columns of $X$.
2. Apply PCA or SVD to find the principle components of $X$.

**PCA:**

I. Calculate the covariance matrix $C = \frac{1}{N-1} X^T X$.

II. $V$ corresponds to the eigenvectors of $C$.

**SVD:**

I. Calculate the SVD of $X = U \Sigma V^T$.

II. $V$ corresponds to the right singular vectors.

3. Project the data in an $m$ dimensional space: $Y = X V$
Outline

- Principal Component Analysis (PCA)
- Singular Value Decomposition (SVD)
- Multi-Dimensional Scaling (MDS)
- Non-linear extensions:
  - Kernel PCA

MDS

- Multi-Dimensional Scaling [Cox and Cox, 1994].
- MDS give points in a low dimensional space such that the Euclidean distances between them best approximate the original distance matrix.

Given distance matrix

$$\Delta := \begin{pmatrix} \delta_{1,1} & \delta_{1,2} & \cdots & \delta_{1,L} \\ \delta_{2,1} & \delta_{2,2} & \cdots & \delta_{2,L} \\ \vdots & \vdots & \ddots & \vdots \\ \delta_{L,1} & \delta_{L,2} & \cdots & \delta_{L,L} \end{pmatrix}$$

Map input points $x_i$ to $z_i$ such as $||z_i - z_j|| \approx \delta_{i,j}$

- Classical MDS: the norm $|| \cdot ||$ is the Euclidean distance.
- Distances $\rightarrow$ inner products (Gram matrix) $\rightarrow$ embedding
  There is a formula to obtain Gram matrix $G$ from distance matrix $\Delta$. 
MDS example

Given pairwise distances between different cities (Δ matrix), plot the cities on a 2D plane (recover location)!!

PCA and MDS relation

- Preserve Euclidean distances = retaining the maximum variance.
- Classical MDS is equivalent to PCA when the distances in the input space are the Euclidean distance.
- PCA uses the \( d \times d \) covariance matrix: \( C = \frac{1}{N-1} X^T X \)
- MDS uses the \( N \times N \) Gram (inner product) matrix: \( G = X X^T \)
- If we have only a distance matrix (we don’t know the points in the original space), we cannot perform PCA!
- Both PCA and MDS are invariant to space rotation!
Kernel PCA

- Given input \((x_1, \ldots, x_N)\), kernel PCA computes the principal components in the feature space \((\varphi(x_1), \ldots, \varphi(x_N))\).
- Avoid explicitly constructing the covariance matrix in feature space.
- The kernel trick: formulate the problem in terms of the kernel function \(k(x, x') = \varphi(x) \cdot \varphi(x')\) without explicitly doing the mapping.
- Kernel PCA is non-linear version of MDS use Gram matrix in the feature space (a.k.a Kernel matrix) instead of Gram matrix in the input space.