CS 4774 Machine Learning

Dimensionality Reduction

Yangfeng Ji

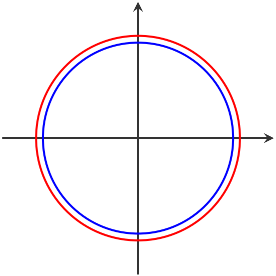
Information and Language Processing Lab
Department of Computer Science
University of Virginia

Overview

Reducing Dimensions

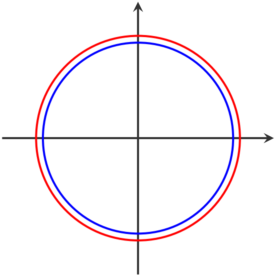
Curse of Dimensionality

What is the volume difference between two d-dimensional balls with radii $r_1 = 1$ and $r_2 = 0.99$



Curse of Dimensionality

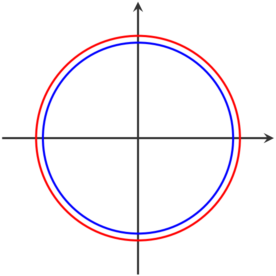
What is the volume difference between two d-dimensional balls with radii $r_1 = 1$ and $r_2 = 0.99$



- $d = 2$: $\frac{1}{2}\pi(r_1^2 - r_2^2) \approx 0.03$
- $d = 3$: $\frac{4}{3}\pi(r_1^3 - r_2^3) \approx 0.12$

Curse of Dimensionality

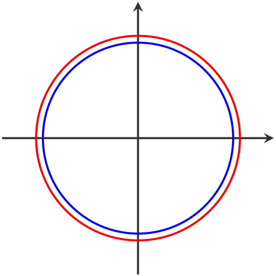
What is the volume difference between two d-dimensional balls with radii $r_1 = 1$ and $r_2 = 0.99$



- $d = 2$: $\frac{1}{2}\pi(r_1^2 - r_2^2) \approx 0.03$
- $d = 3$: $\frac{4}{3}\pi(r_1^3 - r_2^3) \approx 0.12$
- General form: $\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}(r_1^d - r_2^d)$ with $r_2^d \longrightarrow 0$ when $d \longrightarrow \infty$
  - E.g., $r_2^{500} = 0.00657$

Curse of Dimensionality

What is the volume difference between two d-dimensional balls with radii $r_1 = 1$ and $r_2 = 0.99$



- $d = 2$: $\frac{1}{2}\pi(r_1^2 - r_2^2) \approx 0.03$
- $d = 3$: $\frac{4}{3}\pi(r_1^3 - r_2^3) \approx 0.12$
- General form: $\frac{\pi^{d/2}}{\Gamma(\frac{d}{2}+1)}(r_1^d - r_2^d)$ with $r_2^d \longrightarrow 0$ when $d \longrightarrow \infty$
  - E.g., $r_2^{500} = 0.00657$

Question: what will happen if we uniformly sample from a d-dimensional ball?

Curse of Dimensionality (II)

If we randomly sample 1K unit vectors from a d-dimensional space and calculate the the Euclidean distance between any two vectors, then the distance distribution looks like

Curse of Dimensionality (II)

If we randomly sample 1K unit vectors from a d-dimensional space and calculate the the Euclidean distance between any two vectors, then the distance distribution looks like

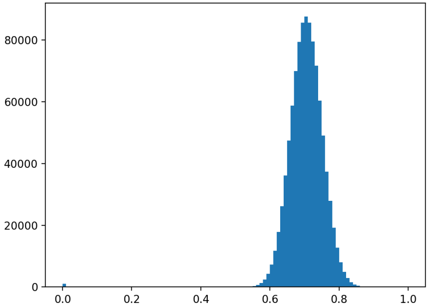


Figure: $d = 100$

Curse of Dimensionality (II)

If we randomly sample 1K unit vectors from a d-dimensional space and calculate the the Euclidean distance between any two vectors, then the distance distribution looks like

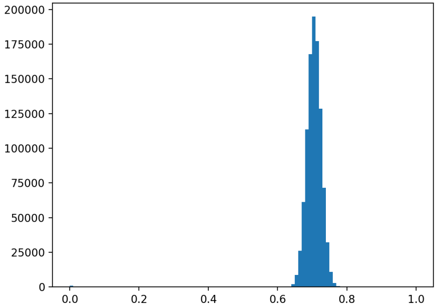


Figure: $d = 500$

Curse of Dimensionality (II)

If we randomly sample 1K unit vectors from a d-dimensional space and calculate the the Euclidean distance between any two vectors, then the distance distribution looks like

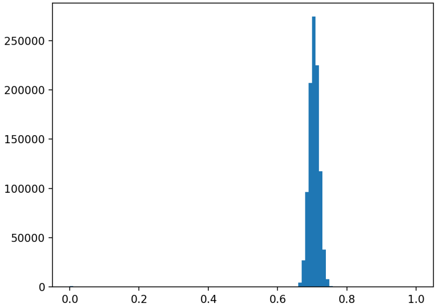


Figure: $d = 1000$

Dimensionality Reduction

Dimensionality Reduction is the process of taking data in a high dimensional space and mapping it into a new space whose dimensionality is much smaller.

Dimensionality Reduction

Dimensionality Reduction is the process of taking data in a high dimensional space and mapping it into a new space whose dimensionality is much smaller.

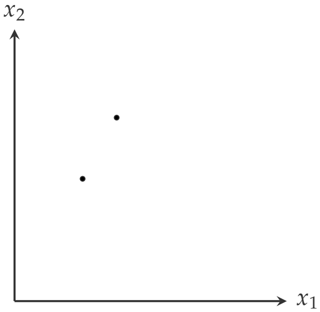
Mathematically, it means

$$f : \boldsymbol{x} \rightarrow \tilde{\boldsymbol{x}} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^d$, $\tilde{\boldsymbol{x}} \in \mathbb{R}^n$ with $n < d$

Reducing Dimensions: A toy example

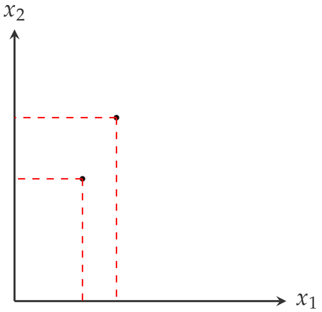
For the purpose of reducing dimensions, we can project $\boldsymbol{x} = (x_1, x_2)$ into the direction along x_1 or x_2



Question: Given these two data examples, which direction we should pick? x_1 or x_2?

Reducing Dimensions: A toy example

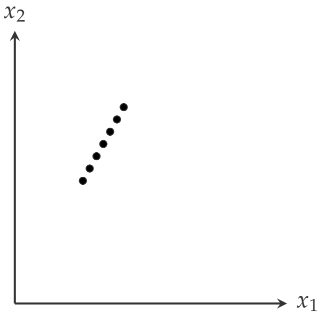
For the purpose of reducing dimensions, we can project $\boldsymbol{x} = (x_1, x_2)$ into the direction along x_1 or x_2



Question: Given these two data examples, which direction we should pick? x_1 or x_2?

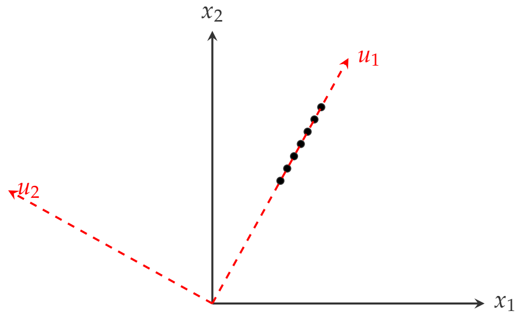
Reducing Dimensions: A toy example (II)

There is a better solution if we are allowed to rotate the coordinate



Reducing Dimensions: A toy example (II)

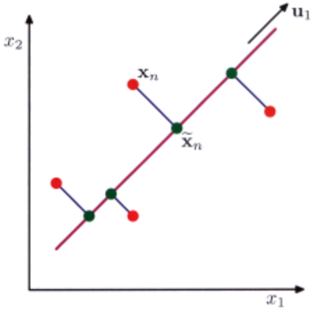
There is a better solution if we are allowed to rotate the coordinate



Pick u_1, then we preserve all the variance of the examples

Reducing Dimensions: A toy example (III)

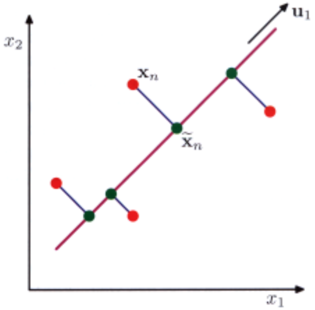
Consider a general case, where the examples do not lie on a perfect line



[Bishop, 2006, Section 12.1]

Reducing Dimensions: A toy example (III)

Consider a general case, where the examples do not lie on a perfect line



We can follow the same idea by finding a direction that can preserve most of the variance of the examples

[Bishop, 2006, Section 12.1]

Principal Component Analysis

Formulation

Given a set of example $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$

- Centering the data by removing the mean $\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \bar{\boldsymbol{x}} \quad \forall i \in [m] \tag{2}$$

Formulation

Given a set of example $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$

- Centering the data by removing the mean $\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \bar{\boldsymbol{x}} \quad \forall i \in [m] \tag{2}$$

- Assume the direction that we would like to project the data is $\boldsymbol{u}$, then the objective function is the data variance

$$J(\boldsymbol{u}) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{u}^{\top}\boldsymbol{x}_i)^2 \tag{3}$$

Formulation

Given a set of example $S = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}$

- Centering the data by removing the mean $\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}_i$

$$\boldsymbol{x}_i \leftarrow \boldsymbol{x}_i - \bar{\boldsymbol{x}} \quad \forall i \in [m] \tag{2}$$

- Assume the direction that we would like to project the data is $\boldsymbol{u}$, then the objective function is the data variance

$$J(\boldsymbol{u}) = \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{u}^\top \boldsymbol{x}_i)^2 \tag{3}$$

- Maximize $J(\boldsymbol{u})$ is trivial, if there is no constriant on $\boldsymbol{u}$. Therefore, we set $\|\boldsymbol{u}\|_2^2 = \boldsymbol{u}^\top \boldsymbol{u} = 1$

Covariance Matrix

The definition of $J(\boldsymbol{u})$ can be written as

$$\begin{aligned}
J(\boldsymbol{u}) &= \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{u}^\top \boldsymbol{x}_i)^2 && (4)\\
&= \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{u}^\top \boldsymbol{x}_i \boldsymbol{u}^\top \boldsymbol{x}_i && (5)\\
&= \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{u}^\top \boldsymbol{x}_i \boldsymbol{x}_i^\top \boldsymbol{u} && (6)\\
&= \boldsymbol{u}^\top\left(\frac{1}{m}\sum_{i=1}^{m}\boldsymbol{x}_i \boldsymbol{x}_i^\top\right)\boldsymbol{u} && (7)\\
&= \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} && (8)
\end{aligned}$$

where $\boldsymbol{\Sigma}$ is the data covariance matrix

Optimization

- The optimization of finding a single direction projection is

$$\max_{\boldsymbol{u}} J(\boldsymbol{u}) = \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \tag{9}$$

$$\text{s.t.} \quad \boldsymbol{u}^\top \boldsymbol{u} = 1 \tag{10}$$

Optimization

- The optimization of finding a single direction projection is

$$\max_{\boldsymbol{u}} J(\boldsymbol{u}) = \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \tag{9}$$

$$\text{s.t.} \quad \boldsymbol{u}^\top \boldsymbol{u} = 1 \tag{10}$$

- It can be converted to an unconstrained optimization problem with a Lagrange multiplier

$$\max_{\boldsymbol{u}} \left\{ \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} + \lambda (1 - \boldsymbol{u}^\top \boldsymbol{u}) \right\} \tag{11}$$

Optimization

- The optimization of finding a single direction projection is

$$\max_{\boldsymbol{u}} J(\boldsymbol{u}) = \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} \tag{9}$$

$$\text{s.t.} \quad \boldsymbol{u}^\top \boldsymbol{u} = 1 \tag{10}$$

- It can be converted to an unconstrained optimization problem with a Lagrange multiplier

$$\max_{\boldsymbol{u}} \left\{ \boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} + \lambda (1 - \boldsymbol{u}^\top \boldsymbol{u}) \right\} \tag{11}$$

- The optimal solution is given by

$$\boldsymbol{\Sigma} \boldsymbol{u} - \lambda \boldsymbol{u} = 0 \tag{12}$$

$$\boldsymbol{\Sigma} \boldsymbol{u} = \lambda \boldsymbol{u} \tag{13}$$

Two Observations

There are two observations from

$$\boldsymbol{\Sigma u} = \lambda \boldsymbol{u} \tag{14}$$

- First, λ is an eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{u}$ is the corresponding eigenvector

Two Observations

There are two observations from

$$\boldsymbol{\Sigma u} = \lambda \boldsymbol{u} \tag{14}$$

- First, λ is an eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{u}$ is the corresponding eigenvector
- Second, multiplying $\boldsymbol{u}^{\top}$ on both sides, we have

$$\boldsymbol{u}^{\top}\boldsymbol{\Sigma u} = \lambda \tag{15}$$

In order to maximize $J(\boldsymbol{u})$, λ has to the largest eigenvalue $\boldsymbol{u}$ is the corresponding eigen vector.

- As $\boldsymbol{u}$ indicates the first major direction that can preserve the data variance, it is called the first principal component

Principal Component Analysis

- As $\boldsymbol{u}$ indicates the first major direction that can preserve the data variance, it is called the first principal component
- In general, with eigen decomposition, we have

$$\boldsymbol{U}^{\top}\boldsymbol{\Sigma}\boldsymbol{U} = \boldsymbol{\Lambda} \tag{16}$$

  - Eigenvalues $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_d)$
  - Eigenvectors $\boldsymbol{U} = [\boldsymbol{u}_1, \dots, \boldsymbol{u}_d]$

Principal Component Analysis (II)

Assume in $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \tag{17}$$

Principal Component Analysis (II)

Assume in $\mathbf{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \tag{17}$$

To reduce the dimensionality of $\boldsymbol{x}$ from d to n, with $n < d$

- Take the first n eigenvectors in $\boldsymbol{U}$ and form

$$\tilde{\boldsymbol{U}} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n] \in \mathbb{R}^{d \times n} \tag{18}$$

Principal Component Analysis (II)

Assume in $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \ldots, \lambda_d)$,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \tag{17}$$

To reduce the dimensionality of $\boldsymbol{x}$ from d to n, with $n < d$

- Take the first n eigenvectors in $\boldsymbol{U}$ and form

$$\tilde{\boldsymbol{U}} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n] \in \mathbb{R}^{d \times n} \tag{18}$$

- Reduce the dimensionality of $\boldsymbol{x}$ as

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{U}}^{\top} \boldsymbol{x} \in \mathbb{R}^n \tag{19}$$

Principal Component Analysis (II)

Assume in $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$,

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d \tag{17}$$

To reduce the dimensionality of $\boldsymbol{x}$ from d to n, with $n < d$

- Take the first n eigenvectors in $\boldsymbol{U}$ and form

$$\tilde{\boldsymbol{U}} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_n] \in \mathbb{R}^{d \times n} \tag{18}$$

- Reduce the dimensionality of $\boldsymbol{x}$ as

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{U}}^\top \boldsymbol{x} \in \mathbb{R}^n \tag{19}$$

- The value of n can be determined by the following

$$\frac{\sum_{i=1}^{n} \lambda_i}{\sum_{i=1}^{d} \lambda_i} \approx 0.95 \tag{20}$$

Reconstructing $\boldsymbol{x}$?

What if we want to reconstruct $\boldsymbol{x}$ ($\mathbb{R}^d$) from $\tilde{\boldsymbol{x}}$ ($\mathbb{R}^n$)?

- The answer is

$$\boldsymbol{x}_{\text{pca}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{x}} \in \mathbb{R}^d$$

Reconstructing $\boldsymbol{x}$?

What if we want to reconstruct $\boldsymbol{x}$ ($\mathbb{R}^d$) from $\tilde{\boldsymbol{x}}$ ($\mathbb{R}^n$)?

- The answer is

$$\boldsymbol{x}_{\text{pca}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{x}} \in \mathbb{R}^d$$

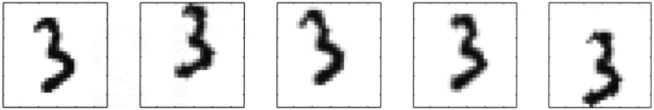
- Therefore, we have

$$\boldsymbol{x}_{\text{pca}} = \tilde{\boldsymbol{U}}\tilde{\boldsymbol{U}}^\top \boldsymbol{x}$$

  as a reasonable approximation of $\boldsymbol{x}$

Applications: Image Processing

Reduce the dimensionality of an image dataset from $28 \times 28 = 784$ to M

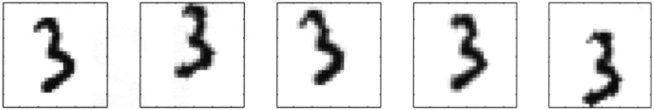


(a) Original data

[Bishop, 2006, Section 12.1]

Applications: Image Processing

Reduce the dimensionality of an image dataset from $28 \times 28 = 784$ to M



(a) Original data

(b) With the first M principal components

[Bishop, 2006, Section 12.1]

A Different Viewpoint of PCA

Data Reconstruction

Another way to formulate the objective function of PCA

$$\min_{\boldsymbol{W},\boldsymbol{U}} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{W}\boldsymbol{x}_i\|_2^2 \tag{21}$$

where

- $\boldsymbol{W} \in \mathbb{R}^{n \times d}$: mapping $\boldsymbol{x}_i$ from the original space to a lower-dimensional space $\mathbb{R}^n$
- $\boldsymbol{U} \in \mathbb{R}^{d \times n}$: mapping back the original space $\mathbb{R}^d$

[Shalev-Shwartz and Ben-David, 2014, Chap 23]

Data Reconstruction

Another way to formulate the objective function of PCA

$$\min_{W,U} \sum_{i=1}^{m} \|x_i - UWx_i\|_2^2 \tag{21}$$

where

- $W \in \mathbb{R}^{n\times d}$: mapping x_i from the original space to a lower-dimensional space $\mathbb{R}^n$
- $U \in \mathbb{R}^{d\times n}$: mapping back the original space $\mathbb{R}^d$
- Dimensionality reduction is performed as $\tilde{x} = Ux$, while W make sure the reduction does not loss much information

[Shalev-Shwartz and Ben-David, 2014, Chap 23]

Optimization

Consider the optimization problem

$$\min_{W,V} \sum_{i=1}^{m} \|x_i - UWx_i\|_2^2 \tag{22}$$

- Let W, U be a solution of equation 24 [Shalev-Shwartz and Ben-David, 2014, Lemma 23.1]
  - the columns of U are orthonormal
  - $W = U^\top$

Optimization

Consider the optimization problem

$$\min_{\boldsymbol{W},\boldsymbol{V}} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{W}\boldsymbol{x}_i\|_2^2 \tag{22}$$

- Let $\boldsymbol{W}, \boldsymbol{U}$ be a solution of equation 24 [Shalev-Shwartz and Ben-David, 2014, Lemma 23.1]
  - the columns of $\boldsymbol{U}$ are orthonormal
  - $\boldsymbol{W} = \boldsymbol{U}^\top$
- The optimization problem can be simplified as

$$\min_{\boldsymbol{U}^\top\boldsymbol{U}=\boldsymbol{I}} \sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{U}^\top\boldsymbol{x}_i\|_2^2 \tag{23}$$

The solution will be the same.

Nonlinear Extension

If we extend the both mappings to be nonlinear, then the model becomes a simple encoder-decoder neural network model

$$\min_{W,V} \sum_{i=1}^{m} \|x_i - \tanh(U \cdot \tanh(W x_i))\|_2^2 \tag{24}$$

where

- $\tilde{x} = \tanh(W x_i)$ is a simple encoder
- $x = \tanh(U \tilde{x})$ is a simple decoder
- No closed-form solutions of W, U, although the backpropagation algorithm still applies here

Reference

Bishop, C. M. (2006).
Pattern recognition and machine learning.
Springer.

Shalev-Shwartz, S. and Ben-David, S. (2014).
Understanding machine learning: From theory to algorithms.
Cambridge university press.