Introduction to Machine Learning

Feature Generation

Andres Mendez-Vazquez

February 3, 2023

Outline

Outline

What do we want?

What

- Given a set of measurements, the goal is to discover compact and informative representations of the obtained data.

What do we want?

What

- Given a set of measurements, the goal is to discover compact and informative representations of the obtained data.

Our Approach

- We want to “squeeze” in a relatively small number of features, leading to a reduction of the necessary feature space dimension.

What do we want?

What

- Given a set of measurements, the goal is to discover compact and informative representations of the obtained data.

Our Approach

- We want to "squeeze" in a relatively small number of features, leading to a reduction of the necessary feature space dimension.

Properties

- Thus removing information redundancies - Usually produced and the measurement.

What Methods we will see?

Fisher Linear Discriminant

1. Squeezing to the maximum.
2. From Many to One Dimension

What Methods we will see?

Fisher Linear Discriminant

1. Squeezing to the maximum.
2. From Many to One Dimension

Principal Component Analysis

1. Not so much squeezing
2. You are willing to lose some information

Outline

Rotation

Projecting

Projecting well-separated samples onto an arbitrary line usually produces a confused mixture of samples from all of the classes and thus produces poor recognition performance.

Rotation

Projecting

Projecting well-separated samples onto an arbitrary line usually produces a confused mixture of samples from all of the classes and thus produces poor recognition performance.

Something Notable

However, moving and rotating the line around might result in an orientation for which the projected samples are well separated.

Rotation

Projecting

Projecting well-separated samples onto an arbitrary line usually produces a confused mixture of samples from all of the classes and thus produces poor recognition performance.

Something Notable

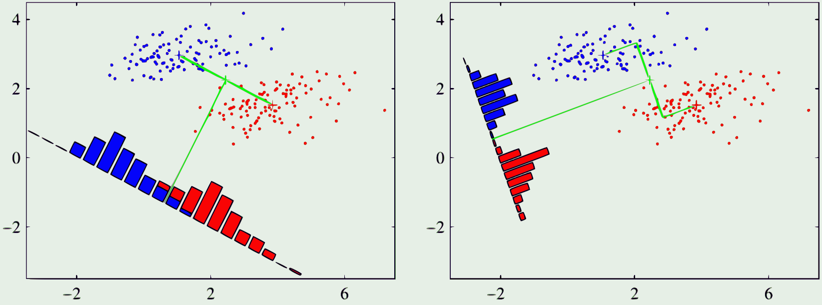
However, moving and rotating the line around might result in an orientation for which the projected samples are well separated.

Fisher linear discriminant (FLD)

It is a discriminant analysis seeking directions that are efficient for discriminating binary classification problem.

Example

Example - From Left to Right the Improvement



This is actually comming from...

Classifier as

A machine for dimensionality reduction.

This is actually comming from...

Classifier as

A machine for dimensionality reduction.

Initial Setup

We have:

- N d-dimensional samples $x_1, x_2, ..., x_N$
- N_i is the number of samples in class C_i for i=1,2.

This is actually comming from...

Classifier as

A machine for dimensionality reduction.

Initial Setup

We have:

- N d-dimensional samples $x_1, x_2, ..., x_N$
- N_i is the number of samples in class C_i for i=1,2.

Then, we ask for the projection of each x_i into the line by means of

$$y_i = \boldsymbol{w}^T \boldsymbol{x}_i \tag{1}$$

Outline

Use the mean of each Class

Then

Select $\boldsymbol{w}$ such that class separation is maximized

Use the mean of each Class

Then

Select $\boldsymbol{w}$ such that class separation is maximized

We then define the mean sample for ecah class

1. $C_1 \Rightarrow \boldsymbol{m}_1 = \frac{1}{N_1}\sum_{i=1}^{N_1} \boldsymbol{x}_i$
2. $C_2 \Rightarrow \boldsymbol{m}_2 = \frac{1}{N_2}\sum_{i=1}^{N_2} \boldsymbol{x}_i$

Use the mean of each Class

Then

Select $\boldsymbol{w}$ such that class separation is maximized

We then define the mean sample for ecah class

1. $C_1 \Rightarrow \boldsymbol{m}_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} \boldsymbol{x}_i$
2. $C_2 \Rightarrow \boldsymbol{m}_2 = \frac{1}{N_2} \sum_{i=1}^{N_2} \boldsymbol{x}_i$

Ok!!! This is giving us a measure of distance

Thus, we want to maximize the distance the projected means:

$$m_1 - m_2 = \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2) \quad (2)$$

where $m_k = \boldsymbol{w}^T \boldsymbol{m}_k$ for $k = 1, 2$.

However

We could simply seek

$$\max \boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)$$
$$s.t. \sum_{i=1}^{d} w_i = 1$$

However

We could simply seek

$$\max \boldsymbol{w}^T \left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)$$

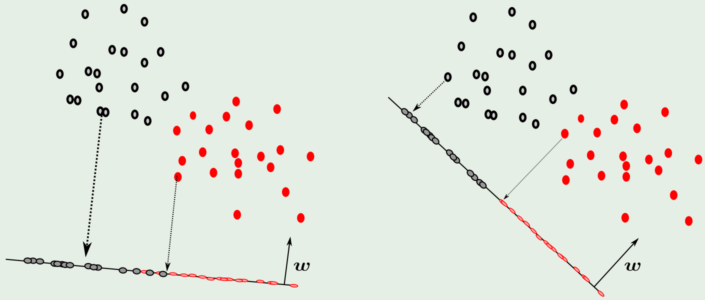
$$s.t. \sum_{i=1}^{d} w_i = 1$$

After all

We do not care about the magnitude of $\boldsymbol{w}$.

Example

Here, we have the problem



Outline

Fixing the Problem

To obtain good separation of the projected data

The difference between the means should be large relative to some measure of the standard deviations for each class.

Fixing the Problem

To obtain good separation of the projected data

The difference between the means should be large relative to some measure of the standard deviations for each class.

We define a SCATTER measure (Based in the Sample Variance)

$$s_k^2 = \sum_{\boldsymbol{x}_i \in C_k} \left(\boldsymbol{w}^T \boldsymbol{x}_i - m_k \right)^2 = \sum_{y_i = \boldsymbol{w}^T \boldsymbol{x}_i \in C_k} \left(y_i - m_k \right)^2 \quad (3)$$

Fixing the Problem

To obtain good separation of the projected data

The difference between the means should be large relative to some measure of the standard deviations for each class.

We define a SCATTER measure (Based in the Sample Variance)

$$s_k^2 = \sum_{\boldsymbol{x}_i \in C_k} \left(\boldsymbol{w}^T \boldsymbol{x}_i - m_k \right)^2 = \sum_{y_i = \boldsymbol{w}^T \boldsymbol{x}_i \in C_k} \left(y_i - m_k \right)^2 \tag{3}$$

We define then within-class variance for the whole data

$$s_1^2 + s_2^2 \tag{4}$$

Outline

Finally, a Cost Function

The between-class variance

$$(m_1 - m_2)^2 \tag{5}$$

Finally, a Cost Function

The between-class variance

$$(m_1 - m_2)^2 \tag{5}$$

The Fisher criterion

$$\frac{\text{between-class variance}}{\text{within-class variance}} \tag{6}$$

Finally, a Cost Function

The between-class variance

$$(m_1 - m_2)^2 \tag{5}$$

The Fisher criterion

$$\frac{\text{between-class variance}}{\text{within-class variance}} \tag{6}$$

Finally

$$J(\boldsymbol{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \tag{7}$$

We use a transformation to simplify our life

First

$$J(\boldsymbol{w}) = \frac{\left(\boldsymbol{w}^T\boldsymbol{m}_1 - \boldsymbol{w}^T\boldsymbol{m}_2\right)^2}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}(y_i - m_k)^2 + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}(y_i - m_k)^2}$$

We use a transformation to simplify our life

First

$$J(\boldsymbol{w}) = \frac{\left(\boldsymbol{w}^T\boldsymbol{m}_1 - \boldsymbol{w}^T\boldsymbol{m}_2\right)^2}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}(y_i - m_k)^2 + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}(y_i - m_k)^2}$$

Second

$$= \frac{\left(\boldsymbol{w}^T\boldsymbol{m}_1 - \boldsymbol{w}^T\boldsymbol{m}_2\right)\left(\boldsymbol{w}^T\boldsymbol{m}_1 - \boldsymbol{w}^T\boldsymbol{m}_2\right)^T}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)^T + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)^T}$$

We use a transformation to simplify our life

First

$$J(\boldsymbol{w}) = \frac{\left(\boldsymbol{w}^T\boldsymbol{m}_1 - \boldsymbol{w}^T\boldsymbol{m}_2\right)^2}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}(y_i - m_k)^2 + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}(y_i - m_k)^2}$$

Second

$$= \frac{\left(\boldsymbol{w}^T\boldsymbol{m}_1 - \boldsymbol{w}^T\boldsymbol{m}_2\right)\left(\boldsymbol{w}^T\boldsymbol{m}_1 - \boldsymbol{w}^T\boldsymbol{m}_2\right)^T}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)^T + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)\left(\boldsymbol{w}^T\boldsymbol{x}_i - m_k\right)^T}$$

Third

$$= \frac{\boldsymbol{w}^T(\boldsymbol{m}_1 - \boldsymbol{m}_2)\left(\boldsymbol{w}^T(\boldsymbol{m}_1 - \boldsymbol{m}_2)\right)^T}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}\boldsymbol{w}^T(\boldsymbol{x}_i - \boldsymbol{m}_1)\left(\boldsymbol{w}^T(\boldsymbol{x}_i - \boldsymbol{m}_1)\right)^T + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}\boldsymbol{w}^T(\boldsymbol{x}_i - \boldsymbol{m}_2)\left(\boldsymbol{w}^T(\boldsymbol{x}_i - \boldsymbol{m}_2)\right)^T}$$

Transformation

Fourth

$$= \frac{\boldsymbol{w}^T (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w}}{\sum_{y_i = \boldsymbol{w}^T \boldsymbol{x}_i \in C_1} \boldsymbol{w}^T (\boldsymbol{x}_i - \boldsymbol{m}_1)(\boldsymbol{x}_i - \boldsymbol{m}_1)^T \boldsymbol{w} + \sum_{y_i = \boldsymbol{w}^T \boldsymbol{x}_i \in C_2} \boldsymbol{w}^T (\boldsymbol{x}_i - \boldsymbol{m}_2)(\boldsymbol{x}_i - \boldsymbol{m}_2)^T \boldsymbol{w}}$$

Transformation

Fourth

$$= \frac{\boldsymbol{w}^T\left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)\left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)^T\boldsymbol{w}}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}\boldsymbol{w}^T\left(\boldsymbol{x}_i - \boldsymbol{m}_1\right)\left(\boldsymbol{x}_i - \boldsymbol{m}_1\right)^T\boldsymbol{w} + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}\boldsymbol{w}^T\left(\boldsymbol{x}_i - \boldsymbol{m}_2\right)\left(\boldsymbol{x}_i - \boldsymbol{m}_2\right)^T\boldsymbol{w}}$$

Fifth

$$= \frac{\boldsymbol{w}^T\left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)\left(\boldsymbol{m}_1 - \boldsymbol{m}_2\right)^T\boldsymbol{w}}{\boldsymbol{w}^T\left[\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}\left(\boldsymbol{x}_i - \boldsymbol{m}_1\right)\left(\boldsymbol{x}_i - \boldsymbol{m}_1\right)^T + \sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}\left(\boldsymbol{x}_i - \boldsymbol{m}_2\right)\left(\boldsymbol{x}_i - \boldsymbol{m}_2\right)^T\right]\boldsymbol{w}}$$

Transformation

Fourth

$$= \frac{\boldsymbol{w}^T\left(\boldsymbol{m}_1-\boldsymbol{m}_2\right)\left(\boldsymbol{m}_1-\boldsymbol{m}_2\right)^T\boldsymbol{w}}{\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}\boldsymbol{w}^T\left(\boldsymbol{x}_i-\boldsymbol{m}_1\right)\left(\boldsymbol{x}_i-\boldsymbol{m}_1\right)^T\boldsymbol{w}+\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}\boldsymbol{w}^T\left(\boldsymbol{x}_i-\boldsymbol{m}_2\right)\left(\boldsymbol{x}_i-\boldsymbol{m}_2\right)^T\boldsymbol{w}}$$

Fifth

$$= \frac{\boldsymbol{w}^T\left(\boldsymbol{m}_1-\boldsymbol{m}_2\right)\left(\boldsymbol{m}_1-\boldsymbol{m}_2\right)^T\boldsymbol{w}}{\boldsymbol{w}^T\left[\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_1}\left(\boldsymbol{x}_i-\boldsymbol{m}_1\right)\left(\boldsymbol{x}_i-\boldsymbol{m}_1\right)^T+\sum_{y_i=\boldsymbol{w}^T\boldsymbol{x}_i\in C_2}\left(\boldsymbol{x}_i-\boldsymbol{m}_2\right)\left(\boldsymbol{x}_i-\boldsymbol{m}_2\right)^T\right]\boldsymbol{w}}$$

Now Rename

$$J\left(\boldsymbol{w}\right)=\frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}}{\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}} \tag{8}$$

Derive with respect to $\boldsymbol{w}$

Thus

$$\frac{dJ\left(\boldsymbol{w}\right)}{d\boldsymbol{w}} = \frac{d\left(\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}\right)\left(\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}\right)^{-1}}{d\boldsymbol{w}} = 0 \tag{9}$$

Derive with respect to $\boldsymbol{w}$

Thus

$$\frac{dJ\left(\boldsymbol{w}\right)}{d\boldsymbol{w}}=\frac{d\left(\boldsymbol{w}^{T}\boldsymbol{S}_{B}\boldsymbol{w}\right)\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)^{-1}}{d\boldsymbol{w}}=0 \tag{9}$$

Then

$$\frac{dJ\left(\boldsymbol{w}\right)}{d\boldsymbol{w}}=\left(\boldsymbol{S}_{B}\boldsymbol{w}+\boldsymbol{S}_{B}^{T}w\right)\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)^{-1}-\left(\boldsymbol{w}^{T}\boldsymbol{S}_{B}\boldsymbol{w}\right)\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)^{-2}\left(\boldsymbol{S}_{w}\boldsymbol{w}+\boldsymbol{S}_{w}^{T}w\right)=0 \tag{10}$$

Derive with respect to $\boldsymbol{w}$

Thus

$$\frac{dJ\left(\boldsymbol{w}\right)}{d\boldsymbol{w}} = \frac{d\left(\boldsymbol{w}^{T}\boldsymbol{S}_{B}\boldsymbol{w}\right)\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)^{-1}}{d\boldsymbol{w}} = 0 \tag{9}$$

Then

$$\frac{dJ\left(\boldsymbol{w}\right)}{d\boldsymbol{w}} = \left(\boldsymbol{S}_{B}\boldsymbol{w} + \boldsymbol{S}_{B}^{T}w\right)\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)^{-1} - \left(\boldsymbol{w}^{T}\boldsymbol{S}_{B}\boldsymbol{w}\right)\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)^{-2}\left(\boldsymbol{S}_{w}\boldsymbol{w} + \boldsymbol{S}_{w}^{T}w\right) = 0 \tag{10}$$

Now because the symmetry in $\boldsymbol{S}_B$ and $\boldsymbol{S}_w$

$$\frac{dJ\left(\boldsymbol{w}\right)}{d\boldsymbol{w}} = \frac{\boldsymbol{S}_{B}\boldsymbol{w}}{\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)} - \frac{\boldsymbol{w}^{T}\boldsymbol{S}_{B}\boldsymbol{w}\boldsymbol{S}_{w}\boldsymbol{w}}{\left(\boldsymbol{w}^{T}\boldsymbol{S}_{w}\boldsymbol{w}\right)^{2}} = 0 \tag{11}$$

Derive with respect to $\boldsymbol{w}$

Thus

$$\frac{dJ\left(\boldsymbol{w}\right)}{d\boldsymbol{w}} = \frac{\boldsymbol{S}_B\boldsymbol{w}}{\left(\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}\right)} - \frac{\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}\boldsymbol{S}_w\boldsymbol{w}}{\left(\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}\right)^2} = 0 \tag{12}$$

Derive with respect to $\boldsymbol{w}$

Thus

$$\frac{dJ(\boldsymbol{w})}{d\boldsymbol{w}} = \frac{\boldsymbol{S}_B \boldsymbol{w}}{(\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w})} - \frac{\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w} \boldsymbol{S}_w \boldsymbol{w}}{(\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w})^2} = 0 \tag{12}$$

Then

$$\left(\boldsymbol{w}^T \boldsymbol{S}_w \boldsymbol{w}\right) \boldsymbol{S}_B \boldsymbol{w} = \left(\boldsymbol{w}^T \boldsymbol{S}_B \boldsymbol{w}\right) \boldsymbol{S}_w \boldsymbol{w} \tag{13}$$

Now, Several Tricks!!!

First

$$\boldsymbol{S}_B \boldsymbol{w} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w} = \alpha (\boldsymbol{m}_1 - \boldsymbol{m}_2) \quad (14)$$

Now, Several Tricks!!!

First

$$\boldsymbol{S}_B\boldsymbol{w} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w} = \alpha(\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{14}$$

Where $\alpha = (\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w}$ is a simple constant

It means that $\boldsymbol{S}_B\boldsymbol{w}$ is always in the direction $\boldsymbol{m}_1 - \boldsymbol{m}_2$!!!

Now, Several Tricks!!!

First

$$\boldsymbol{S}_B\boldsymbol{w} = (\boldsymbol{m}_1 - \boldsymbol{m}_2)(\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w} = \alpha(\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{14}$$

Where $\alpha = (\boldsymbol{m}_1 - \boldsymbol{m}_2)^T \boldsymbol{w}$ is a simple constant

It means that $\boldsymbol{S}_B\boldsymbol{w}$ is always in the direction $\boldsymbol{m}_1 - \boldsymbol{m}_2$!!!

In addition

$\boldsymbol{w}^T\boldsymbol{S}_w\boldsymbol{w}$ and $\boldsymbol{w}^T\boldsymbol{S}_B\boldsymbol{w}$ are constants

Now, Several Tricks!!!

Finally

$$\boldsymbol{S}_w \boldsymbol{w} \propto (\boldsymbol{m}_1 - \boldsymbol{m}_2) \Rightarrow \boldsymbol{w} \propto \boldsymbol{S}_w^{-1} (\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{15}$$

Now, Several Tricks!!!

Finally

$$\boldsymbol{S}_w \boldsymbol{w} \propto (\boldsymbol{m}_1 - \boldsymbol{m}_2) \Rightarrow \boldsymbol{w} \propto \boldsymbol{S}_w^{-1} (\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{15}$$

Once the data is transformed into y_i

- Use a threshold $y_0 \Rightarrow x \in C_1$ iff $y(x) \geq y_0$ or $x \in C_2$ iff $y(x) < y_0$

Now, Several Tricks!!!

Finally

$$\boldsymbol{S}_w \boldsymbol{w} \propto (\boldsymbol{m}_1 - \boldsymbol{m}_2) \Rightarrow \boldsymbol{w} \propto \boldsymbol{S}_w^{-1} (\boldsymbol{m}_1 - \boldsymbol{m}_2) \tag{15}$$

Once the data is transformed into y_i

- Use a threshold $y_0 \Rightarrow x \in C_1$ iff $y(x) \geq y_0$ or $x \in C_2$ iff $y(x) < y_0$
- Or ML with a Gussian can be used to classify the new transformed data using a Naive Bayes (Central Limit Theorem and $y = \boldsymbol{w}^T \boldsymbol{x}$ sum of random variables).

Please

Your Reading Material, it is about the Multiclass

4.1.6 Fisher's discriminant for multiple classes AT "Pattern Recognition" by Bishop

Outline

Did you noticed?

That Rotations really do not exist

- Actually, they are mappings or projections in linear algebra

Did you noticed?

That Rotations really do not exist

- Actually, they are mappings or projections in linear algebra

Thus, Can we get more powerful mappings?

- To obtain better features

Did you noticed?

That Rotations really do not exist

- Actually, they are mappings or projections in linear algebra

Thus, Can we get more powerful mappings?

- To obtain better features

Clearly... Yes

- For example, Principal Components or Singular Value Decomposition's

Outline

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Also Known as Karhunen-Loeve Transform

Setup

- Consider a data set of observations $\{\boldsymbol{x}_n\}$ with $n = 1, 2, ..., N$ and $\boldsymbol{x_n} \in R^d$.

Goal

Project data onto space with dimensionality $m < d$ (We assume m is given)

Dimensional Variance

Remember the Variance Sample in $\mathbb{R}$

$$VAR(X) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N-1} \tag{16}$$

Dimensional Variance

Remember the Variance Sample in $\mathbb{R}$

$$VAR(X) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N-1} \quad (16)$$

You can do the same in the case of two variables X and Y

$$COV(x, y) = \frac{\sum_{i=1}^{N} (x_i - \overline{x})(y_i - \overline{y})}{N-1} \quad (17)$$

Basically

Principal Component Analysis

- Attempts to maximize the variance in certain vectors

Basically

Principal Component Analysis

- Attempts to maximize the variance in certain vectors

Basically Linear Algebra

- Basically discover the basis that describe best the data dispersion in specific directions

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{18}$$

where $\boldsymbol{x}_i$ is a column vector

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{18}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x}_i \tag{19}$$

Now, Define

Given the data

$$\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N \tag{18}$$

where $\boldsymbol{x}_i$ is a column vector

Construct the sample mean

$$\overline{\boldsymbol{x}} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i \tag{19}$$

Center data

$$\boldsymbol{x}_1 - \overline{\boldsymbol{x}}, \boldsymbol{x}_2 - \overline{\boldsymbol{x}}, ..., \boldsymbol{x}_N - \overline{\boldsymbol{x}} \tag{20}$$

Build the Sample Mean

The Covariance Matrix

$$S=\frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)^T \tag{21}$$

Build the Sample Mean

The Covariance Matrix

$$S = \frac{1}{N-1} \sum_{i=1}^{N} \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right) \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{21}$$

Properties

1. The ijth value of S is equivalent to σ_{ij}^2.
2. The iith value of S is equivalent to σ_{ii}^2.

Outline

Clearly

We need to build a projection

- Remember a square matrix is basically a projection

$$A\boldsymbol{x} = \boldsymbol{x}' \left\{ \text{Projections into the Column Space} \right.$$

Clearly

We need to build a projection

- Remember a square matrix is basically a projection

$$A\boldsymbol{x} = \boldsymbol{x}' \left\{ \text{Projections into the Column Space} \right.$$

Thus, we want to have the larger dispesrions

- Why not start with a column space of a single dimension == single vector

Using S to Project Data

For this we use a $\boldsymbol{u}_1$ (The single vector!!!)

- with $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$, an orthonormal vector

Using S to Project Data

For this we use a $\boldsymbol{u}_1$ (The single vector!!!)

- with $\boldsymbol{u}_1^T \boldsymbol{u}_1 = 1$, an orthonormal vector

Question

- What is the Sample Variance of the Projected Data?

Outline

Thus we have

Variance of the projected data

$$\frac{1}{N-1}\sum_{i=1}^{N}\left[\boldsymbol{u}_1\boldsymbol{x}_i - \boldsymbol{u}_1\overline{\boldsymbol{x}}\right] = \boldsymbol{u}_1^T S \boldsymbol{u}_1 \tag{22}$$

Thus we have

Variance of the projected data

$$\frac{1}{N-1}\sum_{i=1}^{N}\left[\boldsymbol{u}_1\boldsymbol{x}_i - \boldsymbol{u}_1\overline{\boldsymbol{x}}\right] = \boldsymbol{u}_1^T S \boldsymbol{u}_1 \quad (22)$$

Use Lagrange Multipliers to Maximize

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 + \lambda_1\left(1 - \boldsymbol{u}_1^T \boldsymbol{u}_1\right) \quad (23)$$

Derive by $\boldsymbol{u}_1$

We get that

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{24}$$

Derive by $\boldsymbol{u}_1$

We get that

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \quad (24)$$

Then

- $\boldsymbol{u}_1$ is an eigenvector of S.

Derive by $\boldsymbol{u}_1$

We get that

$$S\boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1 \tag{24}$$

Then

- $\boldsymbol{u}_1$ is an eigenvector of S.

If we left-multiply by $\boldsymbol{u}_1$

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{25}$$

What about the Second Vector $\boldsymbol{u}_2$?

We have the following optimization problem

$$\begin{aligned} \max \ & \boldsymbol{u}_2^T S \boldsymbol{u}_2 \\ \text{s.t. } & \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1 \\ & \boldsymbol{u}_2^T \boldsymbol{u}_1 = 0 \end{aligned}$$

What about the Second Vector $\boldsymbol{u}_2$?

We have the following optimization problem

$$\begin{aligned} \max \ & \boldsymbol{u}_2^T S \boldsymbol{u}_2 \\ \text{s.t. } & \boldsymbol{u}_2^T \boldsymbol{u}_2 = 1 \\ & \boldsymbol{u}_2^T \boldsymbol{u}_1 = 0 \end{aligned}$$

We can build the Lagrangian function

$$L\left(\boldsymbol{u}_2, \lambda_1', \lambda_2'\right) = \boldsymbol{u}_2^T S \boldsymbol{u}_2 - \lambda_1' \left(\boldsymbol{u}_2^T \boldsymbol{u}_2 - 1\right) - \lambda_2' \left(\boldsymbol{u}_2^T \boldsymbol{u}_1 - 0\right)$$

Explanation

First the constrained maximize

- We want to to maximize $\boldsymbol{u}_2^T S \boldsymbol{u}_2$ (For the second vector)

Explanation

First the constrained maximize

- We want to to maximize $\boldsymbol{u}_2^T S \boldsymbol{u}_2$ (For the second vector)

Given that the second eigenvector is orthonormal

- We have then $\boldsymbol{u}_2^T \boldsymbol{u}_2 = 1$

Explanation

First the constrained maximize

- We want to to maximize $\boldsymbol{u}_2^T S \boldsymbol{u}_2$ (For the second vector)

Given that the second eigenvector is orthonormal

- We have then $\boldsymbol{u}_2^T \boldsymbol{u}_2 = 1$

Under orthonormal vectors

- The covariance goes to zero
  $cov\left(\boldsymbol{u}_1, \boldsymbol{u}_2\right) = \boldsymbol{u}_2^T S \boldsymbol{u}_1 = \boldsymbol{u}_2 \lambda_1 \boldsymbol{u}_1 = \lambda_1 \boldsymbol{u}_1^T \boldsymbol{u}_2 = 0$

Meaning

The PCA's are perpendicular

$$L\left(\boldsymbol{u}_2, \lambda_1', \lambda_2'\right) = \boldsymbol{u}_2^T S \boldsymbol{u}_2 - \lambda_2' \left(\boldsymbol{u}_2^T \boldsymbol{u}_2 - 1\right) - \lambda_1' \left(\boldsymbol{u}_2^T \boldsymbol{u}_1 - 0\right)$$

Meaning

The PCA's are perpendicular

$$L\left(\boldsymbol{u}_2, \lambda_1', \lambda_2'\right) = \boldsymbol{u}_2^T S \boldsymbol{u}_2 - \lambda_2'\left(\boldsymbol{u}_2^T \boldsymbol{u}_2 - 1\right) - \lambda_1'\left(\boldsymbol{u}_2^T \boldsymbol{u}_1 - 0\right)$$

The the derivative with respect to $\boldsymbol{u}_2$

$$\frac{\partial L\left(\boldsymbol{u}_2, \lambda_1', \lambda_2'\right)}{\partial \boldsymbol{u}_2} = 2S\boldsymbol{u}_2 - \lambda_2'\boldsymbol{u}_2 - \lambda_1'\boldsymbol{u}_1 = 0$$

Meaning

The PCA's are perpendicular

$$L\left(\boldsymbol{u}_2, \lambda_1', \lambda_2'\right) = \boldsymbol{u}_2^T S \boldsymbol{u}_2 - \lambda_2' \left(\boldsymbol{u}_2^T \boldsymbol{u}_2 - 1\right) - \lambda_1' \left(\boldsymbol{u}_2^T \boldsymbol{u}_1 - 0\right)$$

The the derivative with respect to $\boldsymbol{u}_2$

$$\frac{\partial L\left(\boldsymbol{u}_2, \lambda_1', \lambda_2'\right)}{\partial \boldsymbol{u}_2} = 2S\boldsymbol{u}_2 - \lambda_2' \boldsymbol{u}_2 - \lambda_1' \boldsymbol{u}_1 = 0$$

Then, we note the following

$$\boldsymbol{u}_1^T \left[S - \lambda_1' I\right] \boldsymbol{u}_2 - \lambda_1' \boldsymbol{u}_1^T \boldsymbol{u}_1 = 0$$

Then, we have that

We have because of Orthogonality and Othonormalidad

$$
\begin{aligned}
\boldsymbol{u}_1^T \left[S - \lambda_1' I \right] \boldsymbol{u}_2 - \lambda_1' \boldsymbol{u}_1^T \boldsymbol{u}_1 &= \boldsymbol{u}_1^T S \boldsymbol{u}_2 - \boldsymbol{\lambda_1' u}_1^T \boldsymbol{u}_2 - \lambda_1' \\
&= \boldsymbol{u}_1^T S \boldsymbol{u}_2 - \lambda_1'
\end{aligned}
$$

Then, we have that

We have because of Orthogonality and Othonormalidad

$$
\begin{aligned}
\boldsymbol{u}_1^T\left[S-\lambda_1' I\right]\boldsymbol{u}_2-\lambda_1'\boldsymbol{u}_1^T\boldsymbol{u}_1 &= \boldsymbol{u}_1^T S\boldsymbol{u}_2-\boldsymbol{\lambda_1'}\boldsymbol{u}_1^T\boldsymbol{u}_2-\lambda_1' \\
&= \boldsymbol{u}_1^T S\boldsymbol{u}_2-\lambda_1'
\end{aligned}
$$

We can prove that

$$
\begin{aligned}
\boldsymbol{u}_1^T S\boldsymbol{u}_2 &= \boldsymbol{u}_2^T S\boldsymbol{u}_1 \\
&= \lambda_1\boldsymbol{u}_2^T u_1 \\
&= 0
\end{aligned}
$$

Thus, we have that

Making this to zero, we have the following implication

$$\boldsymbol{u}_1^T S \boldsymbol{u}_2 - \lambda_1' = 0 \longrightarrow \lambda_1' = 0$$

Therefore, we have

Then, for this setup $\lambda'_1 = 0$

$$S\boldsymbol{u}_2 = \lambda'_2 \boldsymbol{u}_2$$

Therefore, we have

Then, for this setup $\lambda'_1 = 0$

$$S\boldsymbol{u}_2 = \lambda'_2 \boldsymbol{u}_2$$

Proving u_2 is the eigenvector of S

- Corresponding to the second largest eigenvalue λ'_2

Thus, we have

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{26}$$

is set to the largest eigenvalue. Also know as the First Principal Component

Thus, we have

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{26}$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Thus, we have

Variance will be the maximum when

$$\boldsymbol{u}_1^T S \boldsymbol{u}_1 = \lambda_1 \tag{26}$$

is set to the largest eigenvalue. Also know as the First Principal Component

By Induction

It is possible for M-dimensional space to define M eigenvectors $\boldsymbol{u}_1, \boldsymbol{u}_2, ..., \boldsymbol{u}_M$ of the data covariance S corresponding to $\lambda_1, \lambda_2, ..., \lambda_M$ that maximize the variance of the projected data.

Computational Cost of PCA

1. Full eigenvector decomposition $O\left(d^3\right)$
2. Power Method $O\left(Md^2\right)$ "Golub and Van Loan, 1996)"
3. Use the Expectation Maximization Algorithm

Outline

We have the following steps

Determine covariance matrix

$$S = \frac{1}{N-1} \sum_{i=1}^{N} \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right) \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \quad (27)$$

We have the following steps

Determine covariance matrix

$$S=\frac{1}{N-1}\sum_{i=1}^{N}\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)\left(\boldsymbol{x}_i-\overline{\boldsymbol{x}}\right)^T \tag{27}$$

Generate the decomposition

$$S=U\Sigma U^T$$

We have the following steps

Determine covariance matrix

$$S = \frac{1}{N-1} \sum_{i=1}^{N} \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right) \left(\boldsymbol{x}_i - \overline{\boldsymbol{x}}\right)^T \tag{27}$$

Generate the decomposition

$$S = U \Sigma U^T$$

With

- Eigenvalues in Σ and eigenvectors in the columns of U.

Then

Project samples $\boldsymbol{x}_i$ into subspaces dim$=k$

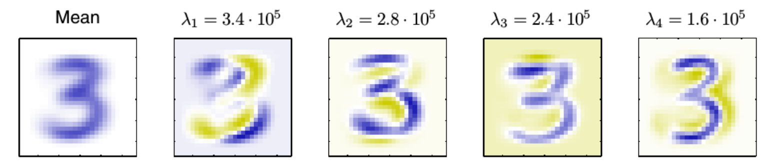
$$z_i = U_K^T \boldsymbol{x}_i$$

- With U_k is a matrix with k columns

Outline

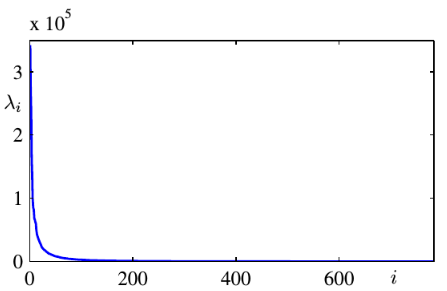
Example

From Bishop



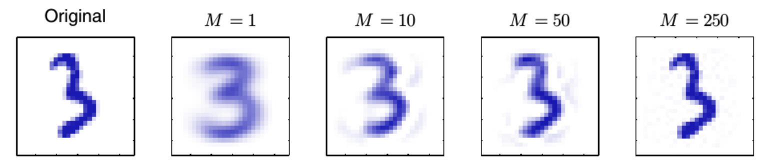
Example

From Bishop



Example

From Bishop



Outline

Outline

What happened with no-square matrices

We can still diagonalize it

Thus, we can obtain certain properties.

What happened with no-square matrices

We can still diagonalize it

Thus, we can obtain certain properties.

We want to avoid the problems with

- The decomposition $A = Q\Lambda Q^{-1}$ (Eigendecomposition) because...

What happened with no-square matrices

We can still diagonalize it

Thus, we can obtain certain properties.

We want to avoid the problems with

- The decomposition $A = Q\Lambda Q^{-1}$ (Eigendecomposition) because...

The eigenvectors in A have three big problems

1. They are usually not orthogonal.
2. There are not always enough eigenvectors.
3. $A\boldsymbol{x} = \lambda\boldsymbol{x}$ requires A to be square.

Therefore, we can look at the following problem

We have a series of vectors

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$$

Therefore, we can look at the following problem

We have a series of vectors

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$$

Then imagine a set of projection vectors and differences

$$\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_n\}$$

Therefore, we can look at the following problem

We have a series of vectors

$$\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$$

Then imagine a set of projection vectors and differences

$$\{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_n\}$$

We want to know a little bit of the relations between them

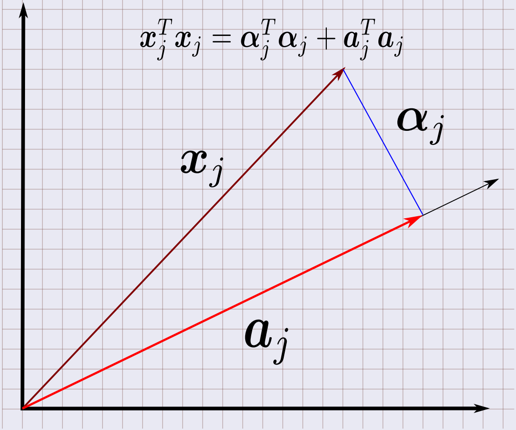
- After all, we are looking at the possibility of using them for our problem

Using the Hypotenuse to build a Relation

A little bit of Geometry, we get

$\boldsymbol{x}_j$	The Data
$\boldsymbol{\alpha}_j$	The Perp defining the projection
$\boldsymbol{a}_j$	The Projection

$$\boldsymbol{x}_j^T\boldsymbol{x}_j = \boldsymbol{\alpha}_j^T\boldsymbol{\alpha}_j + \boldsymbol{a}_j^T\boldsymbol{a}_j$$



Therefore

We have two possible quantities for each j relating them

$$\boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j = \boldsymbol{x}_j^T \boldsymbol{x}_j - \boldsymbol{a}_j^T \boldsymbol{a}_j$$
$$\boldsymbol{a}_j^T \boldsymbol{a}_j = \boldsymbol{x}_j^T \boldsymbol{x}_j - \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j$$

Therefore

We have two possible quantities for each j relating them

$$\boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j = \boldsymbol{x}_j^T \boldsymbol{x}_j - \boldsymbol{a}_j^T \boldsymbol{a}_j$$
$$\boldsymbol{a}_j^T \boldsymbol{a}_j = \boldsymbol{x}_j^T \boldsymbol{x}_j - \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j$$

Then, we can minimize and maximize them given that $\boldsymbol{x}_j^T \boldsymbol{x}_j$ is a constant

- Actually, when looking at the previous figure maximize $\boldsymbol{a}_j$ will minimize α_j
  - Which is similar to minimize α_j will maximize $\boldsymbol{a}_j$

Basically

Something Notable when summing over all the sought vectors

$$\min \sum_{j=1}^{n} \boldsymbol{\alpha}_j^T \boldsymbol{\alpha}_j \Leftrightarrow \max \sum_{j=1}^{n} \boldsymbol{a}_j^T \boldsymbol{a}_j$$

Actually this is know as the dual problem (There are many)

An example of this is the following minimization

$$
\begin{aligned}
\min \ \boldsymbol{w}^T\boldsymbol{y}& \\
s.t\ \mathsf{A}\boldsymbol{y} \geq& \boldsymbol{c} \\
\boldsymbol{y} \geq& 0
\end{aligned}
$$

Actually this is know as the dual problem (There are many)

An example of this is the following minimization

$$\begin{aligned} \min \ & \boldsymbol{w}^T \boldsymbol{y} \\ s.t \ \mathsf{A}\boldsymbol{y} & \geq \boldsymbol{c} \\ \boldsymbol{y} & \geq 0 \end{aligned}$$

Then, we have the following maximization

$$\begin{aligned} \max \ & \boldsymbol{c}^T \boldsymbol{x} \\ s.t \ \mathsf{A}\boldsymbol{x} & \leq \boldsymbol{c} \\ \boldsymbol{x} & \geq 0 \end{aligned}$$

Outline

We have then

Stack such vectors that in the d-dimensional space the

- In a matrix A of $n \times d$, here each vecrtor has dimension d

$$A = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{bmatrix}$$

We have then

Stack such vectors that in the d-dimensional space the

- In a matrix A of $n \times d$, here each vecrtor has dimension d

$$A = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{bmatrix}$$

The matrix works as a Projection Matrix

- We are looking for a unit vector $\boldsymbol{v}$ such that length of the projection is maximized.

Why? Do you remember the Projection to a single vector $\boldsymbol{p}$?

Definition of the projection under unitary vector

$$\boldsymbol{p} = \frac{\boldsymbol{v}^T \boldsymbol{a}_i}{\boldsymbol{v}^T \boldsymbol{v}} \boldsymbol{v} = \left[\boldsymbol{v}^T \boldsymbol{a}_i \right] \boldsymbol{v}$$

Why? Do you remember the Projection to a single vector $\boldsymbol{p}$?

Definition of the projection under unitary vector

$$\boldsymbol{p} = \frac{\boldsymbol{v}^T \boldsymbol{a}_i}{\boldsymbol{v}^T \boldsymbol{v}} \boldsymbol{v} = \left[\boldsymbol{v}^T \boldsymbol{a}_i \right] \boldsymbol{v}$$

Therefore the length of the projected vector is

$$\left\| \left[\boldsymbol{v}^T \boldsymbol{a}_i \right] \boldsymbol{v} \right\| = \left| \boldsymbol{v}^T \boldsymbol{a}_i \right|$$

Then

Thus with a little bit of notation

$$Av = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} \boldsymbol{a}_1^T \boldsymbol{v} \\ \boldsymbol{a}_2^T \boldsymbol{v} \\ \vdots \\ \boldsymbol{a}_n^T \boldsymbol{v} \end{bmatrix}$$

Then

Thus with a little bit of notation

$$A\boldsymbol{v} = \begin{bmatrix} \boldsymbol{a}_1^T \\ \boldsymbol{a}_2^T \\ \vdots \\ \boldsymbol{a}_n^T \end{bmatrix} \boldsymbol{v} = \begin{bmatrix} \boldsymbol{a}_1^T\boldsymbol{v} \\ \boldsymbol{a}_2^T\boldsymbol{v} \\ \vdots \\ \boldsymbol{a}_n^T\boldsymbol{v} \end{bmatrix}$$

Therefore

$$\|A\boldsymbol{v}\| = \sqrt{\sum_{i=1}^{d} \left(\boldsymbol{a}_i^T\boldsymbol{v}\right)^2}$$

Then

It is possible to ask to maximize the longitude of such vector (Singular Vector)

$$\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$$

Then

It is possible to ask to maximize the longitude of such vector (Singular Vector)

$$\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$$

Then, we can define the following singular value

$$\sigma_1(A) = \|A\boldsymbol{v}_1\|$$

This is known as

Definition

- The **best-fit line problem** describes the problem of finding the best line for a set of data points, where the quality of the line is measured by the sum of squared (perpendicular) distances of the points to the line.
  - Remember, we are looking at the dual problem....
    - $\min$ sum of squared (perpendicular) distances $\Leftrightarrow$ $\max$ the projections

This is known as

Definition

- The **best-fit line problem** describes the problem of finding the best line for a set of data points, where the quality of the line is measured by the sum of squared (perpendicular) distances of the points to the line.
    - Remember, we are looking at the dual problem....
        - ★ $\min$ sum of squared (perpendicular) distances $\Leftrightarrow$ $\max$ the projections

Generalization

- This can be transferred to higher dimensions: One can find the best-fit d-dimensional subspace, so the subspace which minimizes the sum of the squared distances of the points to the subspace

Then, in a Greedy Fashion

The second singular vector $\boldsymbol{v}_2$

$$\boldsymbol{v}_2 = \arg \max_{\boldsymbol{v} \perp \boldsymbol{v}_1, \|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$$

Then, in a Greedy Fashion

The second singular vector $\boldsymbol{v}_2$

$$\boldsymbol{v}_2 = \arg \max_{\boldsymbol{v} \perp \boldsymbol{v}_1, \|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$$

Them you go through this process

- Stop when we have found all the following vectors:

$$\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_r$$

Then, in a Greedy Fashion

The second singular vector $\boldsymbol{v}_2$

$$\boldsymbol{v}_2 = \arg \max_{\boldsymbol{v} \perp \boldsymbol{v}_1, \|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$$

Them you go through this process

- Stop when we have found all the following vectors:

$$\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_r$$

As singular vectors and

$$\arg \max_{\substack{\boldsymbol{v} \perp \boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_r \\ \|\boldsymbol{v}\| = 1}} \|A\boldsymbol{v}\|$$

Proving that the strategy is good

Theorem

- Let A be an $n \times d$ matrix where $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_r$ are the singular vectors defined above. For $1 \leq k \leq r$, let V_k be the subspace spanned by $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_k$. Then for each k, V_k is the best-fit k-dimensional subspace for A.

Proof

The statement is obviously true for $k = 1$

- What about $k = 2$? Let W be a best-fit 2- dimensional subspace for A.

Proof

The statement is obviously true for $k = 1$

- What about $k = 2$? Let W be a best-fit 2- dimensional subspace for A.

For any basis $\boldsymbol{w}_1, \boldsymbol{w}_2$ of W

- $\|A\boldsymbol{w}_1\|^2 + \|A\boldsymbol{w}_2\|^2$ is the sum of the squared lengths of the projections of the rows of A to W.

Proof

The statement is obviously true for $k = 1$

- What about $k = 2$? Let W be a best-fit 2- dimensional subspace for A.

For any basis $\boldsymbol{w}_1, \boldsymbol{w}_2$ of W

- $\|A\boldsymbol{w}_1\|^2 + \|A\boldsymbol{w}_2\|^2$ is the sum of the squared lengths of the projections of the rows of A to W.

Now, choose a basis $\boldsymbol{w}_1, \boldsymbol{w}_2$ so that $\boldsymbol{w}_2$ is perpendicular to $\boldsymbol{v}_1$

- This can be a unit vector perpendicular to $\boldsymbol{v}_1$ projection in W.

Do you remember $\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$?

Therefore

$$\|A\boldsymbol{w}_1\|^2 \leq \|A\boldsymbol{v}_1\|^2 \text{ and } \|A\boldsymbol{w}_2\|^2 \leq \|A\boldsymbol{v}_2\|^2$$

Do you remember $\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$?

Therefore

$$\|A\boldsymbol{w}_1\|^2 \leq \|A\boldsymbol{v}_1\|^2 \text{ and } \|A\boldsymbol{w}_2\|^2 \leq \|A\boldsymbol{v}_2\|^2$$

Then

$$\|A\boldsymbol{w}_1\|^2 + \|A\boldsymbol{w}_2\|^2 \leq \|A\boldsymbol{v}_1\|^2 + \|A\boldsymbol{v}_2\|^2$$

Do you remember $\boldsymbol{v}_1 = \arg\max_{\|\boldsymbol{v}\|=1} \|A\boldsymbol{v}\|$?

Therefore

$$\|A\boldsymbol{w}_1\|^2 \leq \|A\boldsymbol{v}_1\|^2 \text{ and } \|A\boldsymbol{w}_2\|^2 \leq \|A\boldsymbol{v}_2\|^2$$

Then

$$\|A\boldsymbol{w}_1\|^2 + \|A\boldsymbol{w}_2\|^2 \leq \|A\boldsymbol{v}_1\|^2 + \|A\boldsymbol{v}_2\|^2$$

In a similar way for $k > 2$

- Thus the subspace V_k is at least as good as W and hence is optimal.

Remarks

Every Matrix has a singular value decomposition

$$A = U\Sigma V^T$$

Remarks

Every Matrix has a singular value decomposition

$$A = U\Sigma V^T$$

Where

- The columns of U are an orthonormal basis for the column space.
- The columns of V are an orthonormal basis for the row space.

Remarks

Every Matrix has a singular value decomposition

$$A = U\Sigma V^T$$

Where

- The columns of U are an orthonormal basis for the column space.
- The columns of V are an orthonormal basis for the row space.
- The Σ is diagonal and the entries on its diagonal $\sigma_i = \Sigma_{ii}$ are positive real numbers, called the singular values of A.

Properties of the Singular Value Decomposition

First

- The eigenvalues of the symmetric matrix $A^T A$ are equal to the square of the singular values of A

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

Properties of the Singular Value Decomposition

First

- The eigenvalues of the symmetric matrix $A^T A$ are equal to the square of the singular values of A

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

Second

- The rank of a matrix is equal to the number of non-zero singular values.

Outline

Singular Value Decomposition as Sums

The singular value decomposition can be viewed as a sum of rank 1 matrices

$$A = A_1 + A_2 + ... + A_R \tag{28}$$

Singular Value Decomposition as Sums

The singular value decomposition can be viewed as a sum of rank 1 matrices

$$A = A_1 + A_2 + \ldots + A_R \tag{28}$$

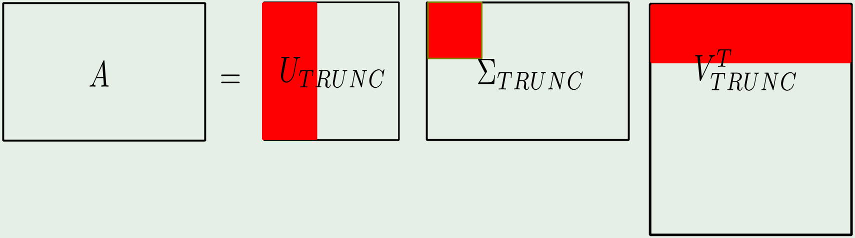
Why?

Decompose $A = U \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_R \end{pmatrix} V^T = \begin{pmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_R \end{pmatrix} \begin{pmatrix} \sigma_1 \boldsymbol{v}_1^T \\ \sigma_2 \boldsymbol{v}_2^T \\ \vdots \\ \sigma_R \boldsymbol{v}_R^T \end{pmatrix}$

$$= \sigma_1 \underbrace{\boldsymbol{u}_1 \boldsymbol{v}_1^T}_{A_1} + \sigma_2 \underbrace{\boldsymbol{u}_2 \boldsymbol{v}_2^T}_{A_2} + \cdots + \sigma_R \underbrace{\boldsymbol{u}_R \boldsymbol{v}_R^T}_{A_R}$$

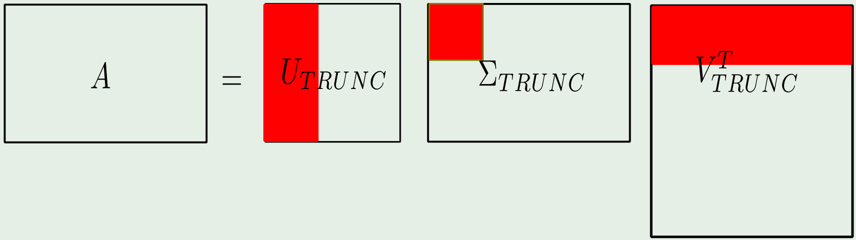
Truncating

Truncating the singular value decomposition allows us to represent the matrix with less parameters



Truncating

Truncating the singular value decomposition allows us to represent the matrix with less parameters

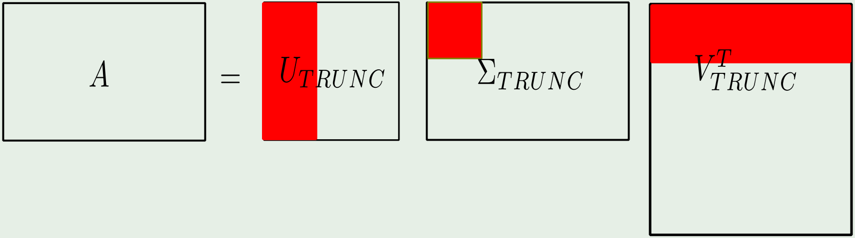


For a 512×512

- Full Representation $512 \times 512 = 262,144$
- Rank 10 approximation $512 \times 10 + 10 + 10 \times 512 = 10,250$

Truncating

Truncating the singular value decomposition allows us to represent the matrix with less parameters

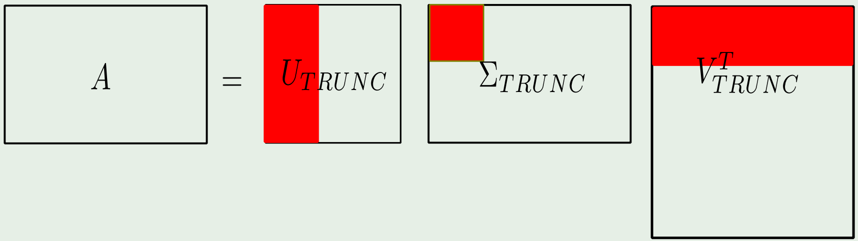


For a 512×512

- Full Representation $512 \times 512 = 262,144$
- Rank 10 approximation $512 \times 10 + 10 + 10 \times 512 = 10,250$
- Rank 40 approximation $512 \times 40 + 40 + 40 \times 512 = 41,000$

Truncating

Truncating the singular value decomposition allows us to represent the matrix with less parameters



For a 512×512

- Full Representation $512 \times 512 = 262,144$
- Rank 10 approximation $512 \times 10 + 10 + 10 \times 512 = 10,250$
- Rank 40 approximation $512 \times 40 + 40 + 40 \times 512 = 41,000$
- Rank 80 approximation $512 \times 80 + 80 + 80 \times 512 = 82,000$