PCA: Unsupervised Dimensionality Reduction

CS 530
Chapman

Spring 2021

1

Take 🏠 message for rest of course: PCA

- What are dimensionality reduction methods?
- What are these methods good for?
- How are they carried out?
- When to use and not to use them?

2

TABLE OF CONTENTS

3

PRINCIPAL COMPONENT ANALYSIS (PCA)

4

Dimensionality Reduction

If our data are composed of many features (dimensions), we often want to reduce the number of dimensions of our data. This often improves prediction accuracy, interpretability, or enables better visual inspection of the data.

Dimensionality reduction methods include:

- Feature Selection
- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA; supervised)
  - Fischer's Linear Discriminant

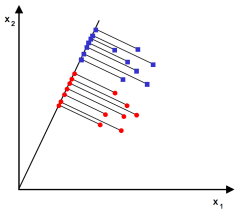


Image Source

5

Dimensionality Reduction

The term feature selection, as we saw for regression, refers to finding a subset of features—$x_{i_1}, x_{i_2} \ldots, x_{i_k}$—that best explain the variability in the response variable, y. We could look at the p-values of the variables or run a subset selection method.

For supervised learning problems, we seek the best subset according to which the dataset can be learned.

Feature selection is different from *dimensionality reduction*.

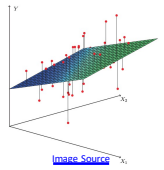


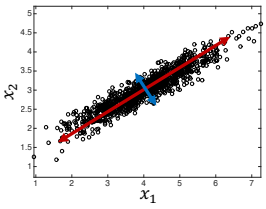
Image Source

6

Dimensionality Reduction

Principal Component Analysis (PCA) finds *linear combinations* of *orthogonal variables* that explain the maximal amount of variance in the data.

In other words, PCA finds an alternative vector basis for the data, where the 1st dimension explains the maximal amount of variance in the data, the 2nd dimension explains the maximal amount of variance left to explain (orthogonal to the 1st dimension), and so on.



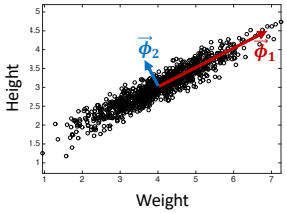
7

PCA: Explanatory Example

Say that you have a dataset with some population's height and weight.

The direction with most the variance in this data—*the 1st principal component*—is designated in burgundy, $\vec{\phi}_1$.

The remaining variance—*the second principal component* in this 2D data—is then designated in blue, $\vec{\phi}_2$.



8

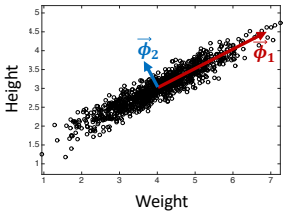
PCA: Explanatory Example

Here we have 1000 samples with 2 features (Height and Weight) each. So, we could describe these data as

$$\vec{x} = \begin{bmatrix} x_{1,1} & x_{1,2} \\ \vdots & \vdots \\ x_{1000,1} & x_{1000,2} \end{bmatrix}.$$

$\vec{x}$ is thus *samples* x *features,* and generally $n \times p$*:*

$$\mathbb{R}^{n\times p} \ni \vec{x} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,p} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$$



9

First Principal Component

For a given set of *features* (not samples), $\vec{x} = [\vec{x}_1, \vec{x}_2 \ldots, \vec{x}_p]$—so, each $\vec{x}_i = (x_{1,i}, x_{2,i} \ldots, x_{n,i})^T \in \mathbb{R}^n$ is a feature

$$\vec{x}_1, \vec{x}_2 \ldots, \vec{x}_p \in \mathbb{R}^n$$

the **first principal component**, $\vec{z}_1$, is the normalized linear combination of the features, $\vec{x}_1, \vec{x}_2 \ldots, \vec{x}_p$, resulting in the maximal variance:

$$\mathbb{R}^n \ni \vec{z}_1 = \phi_{1,1}\vec{x}_1 + \phi_{2,1}\vec{x}_2 + \cdots + \phi_{p,1}\vec{x}_p = \vec{x} \cdot \vec{\phi}_1.$$

Here $\vec{\phi}_1 = (\phi_{1,1}, \phi_{2,1} \ldots, \phi_{p,1})^T \in \mathbb{R}^p$ are the *loadings*.

How do we choose $\vec{z}_1$ that (naively) maximizes variance?

10

First Principal Component

$\vec{z}_1 = \vec{x} \cdot \vec{\phi}_1$ is termed *normalized* linear combination because the coefficients, $\vec{\phi}_1$—the *loadings* of the first principal component—describe a unit vector.

$$\sum_{j=1}^{p} {\phi_{j,1}}^2 = 1 \quad \text{or} \quad \|\vec{\phi}_1\| = 1.$$

Remember, we want to maximize the variance of $\vec{z}_1$.

Why do we need the constraint? In other words, what would happen if we do not put the constraint $\|\vec{\phi}_1\| = 1$?

11

First Principal Component

$\vec{z}_1 = \vec{x} \cdot \vec{\phi}_1$ is termed *normalized* linear combination because the coefficients, $\vec{\phi}_1$—the *loadings* of the first principal component—describe a unit vector.

$$\sum_{j=1}^{p} {\phi_{j,1}}^2 = 1 \quad \text{or} \quad \|\vec{\phi}_1\| = 1.$$

Without the constraint that $\|\vec{\phi}_1\| = 1$, setting $\vec{\phi}_1$ arbitrarily large would make $\vec{z}_1 = \vec{x} \cdot \vec{\phi}_1$ arbitrarily large too (for given data, $\vec{x}$) and thus $\vec{z}_1$'s variance would be arbitrarily large as well. So, without limiting $\|\vec{\phi}_1\|$, the solution to our optimization problem would be $\|\vec{\phi}_1\| \to \infty$, and so $\|\vec{z}_1\| \to \infty$. And that is not the solution we are seeking.

12

First Principal Component

Below we assume that $\vec{x}$ was centered over each feature (column, dimension), and thus has a mean of 0.

If $\vec{x}$ is centered over each feature, what can we do?

13

First Principal Component

Below we assume that $\vec{x}$ was centered over each feature (column, dimension), and thus has a mean of 0. (If not, we subtract overall columns mean from all samples.)

We want to maximize the variance of the first principal component. We must solve the following constrained optimization problem:

$$\vec{\phi}_1 = \underset{\|\vec{\phi}_1\|=1}{\operatorname{argmax}}(\|\vec{z}_1\|^2) \quad (\text{i.e., subject to } \|\vec{\phi}_1\| = 1),$$

which is

$$\vec{\phi}_1 = \underset{\|\vec{\phi}_1\|=1}{\operatorname{argmax}} \|\vec{x} \cdot \vec{\phi}_1\|^2$$

$$= \underset{\|\vec{\phi}_1\|=1}{\operatorname{argmax}} \left\{ \vec{\phi}_1{}^T \vec{x}^T \vec{x} \vec{\phi}_1 \right\} \underset{\|\vec{\phi}_1\|=1}{=} \operatorname{argmax} \left\{ \frac{\vec{\phi}_1{}^T \vec{x}^T \vec{x} \vec{\phi}_1}{\vec{\phi}_1{}^T \vec{\phi}_1} \right\}.$$

14

Reminder: Positive Semi-Definite Matrix

Definition:

A symmetric square matrix, $\vec{a} \in \mathbb{R}^{d \times d}$, is *positive definite* if

$$\vec{v}^T \vec{a}\, \vec{v} > 0 \text{ for all } \vec{v} \in \mathbb{R}^d \setminus 0.$$

A symmetric square matrix, $\vec{a} \in \mathbb{R}^{d \times d}$, is *positive semi-definite* if

$$\vec{v}^T \vec{a}\, \vec{v} \geq 0 \text{ for all } \vec{v} \in \mathbb{R}^d.$$

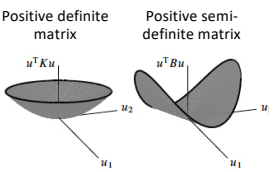


Image Source

15

First Principal Component

$\vec{C} = \vec{x}^T \vec{x}$ is positive semi-definite, so $\frac{\vec{\phi}_1{}^T \vec{x}^T \vec{x} \vec{\phi}_1}{\vec{\phi}_1{}^T \vec{\phi}_1} = \frac{\vec{\phi}_1{}^T \vec{C} \vec{\phi}_1}{\vec{\phi}_1{}^T \vec{\phi}_1}$ is the Rayleigh quotient of $\vec{C}$, which reaches a maximum at $\vec{C}$'s maximal eigenvalue, λ_{max}, with $\vec{v}_{max}$ being its corresponding eigenvector.

In other words, it can be shown that

$$\frac{\vec{\phi}_1{}^T \vec{C} \vec{\phi}_1}{\vec{\phi}_1{}^T \vec{\phi}_1} \leq \lambda_{max}.$$

And

$$\frac{\vec{v}_{max}{}^T \vec{C} \vec{v}_{max}}{\vec{v}_{max}{}^T \vec{v}_{max}} = \lambda_{max}.$$

Where (again) λ_{max} and $\vec{v}_{max}$ are the maximum eigenvalue of $\vec{C}$ and its corresponding eigenvalue

16

Eigenvectors and Eigenvalues

Every square (non-pathological) matrix, $\vec{A}$, corresponds to a linear transformation. So, typically, it rotates and scales a vector, $\vec{y}$:

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 2y_1 + y_2 \\ y_1 + 2y_2 \end{bmatrix}.$$

But, for $\vec{A}$'s eigenvectors $(1,1)^T$ and $(1,-1)^T$:

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3 \\ 3 \end{bmatrix} = 3 \cdot \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 3 \\ -3 \end{bmatrix} = 3 \cdot \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

So, $\vec{A}$ only scales each eigenvector $\vec{v}$ by its corresponding eigenvalue λ:

$$\vec{A}\vec{v} = \lambda \vec{v}.$$

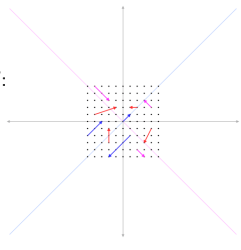
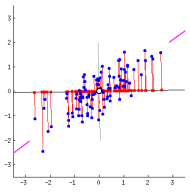


Image Source

17

Visualizing the direction of 1st principal component

- What is the first principal component in the dataset below?



- The direction of maximal variance (magenta lines)

18

Further Principal Components

So $\vec{\phi}_1$ are the loadings of the 1st principal component. The m'th principal component is calculated by finding the first $(m-1)$ principal component of matrix, and then computing

$$\vec{\vec{x}}_{(m)} = \vec{\vec{x}} - \sum_{i=1}^{m-1} \vec{\vec{x}}\,\vec{\phi}_m \vec{\phi}_m^{\,T},$$

in the same manner as we found the first principal component of $\vec{\vec{x}}$:

$$\vec{\phi}_m = \underset{\|\vec{\phi}_m\|=1}{\operatorname{argmax}} \left\| \vec{\vec{x}}_{(m)} \cdot \vec{\phi}_m \right\|^2 = \underset{\|\vec{\phi}_m\|=1}{\operatorname{argmax}} \left\{ \frac{\vec{\phi}_m^{\,T} \vec{\vec{x}}_{(m)}^{\,T} \vec{\vec{x}}_{(m)} \vec{\phi}_m}{\vec{\phi}_m^{\,T} \vec{\phi}_m} \right\}.$$

So, $\vec{\phi}_m$ are the loadings of the m'th principal component .

It turns out that $\vec{\phi}_2, \ldots, \vec{\phi}_p$ are simply the remaining eigenvectors of $\vec{\vec{x}}^T \vec{\vec{x}} = \vec{\vec{C}}$.

19

All Principal Components

In matrix form, we write

$$\vec{\vec{z}} = \vec{\vec{x}}\vec{\vec{\phi}},$$

where

$$\mathbb{R}^{p\times p} \ni \vec{\vec{\phi}} = [\vec{\phi}_1, \vec{\phi}_2 \ldots, \vec{\phi}_\mathrm{p}],$$

and $\vec{\phi}_\mathrm{k}$ is the k'th eigenvector of $\vec{\vec{x}}^T\vec{\vec{x}} = \vec{\vec{C}}$ (the covariance matrix of $\vec{\vec{x}}$).

So, all together,

$$\vec{\phi}_1, \ldots, \vec{\phi}_\mathrm{p}$$

are the eigenvectors of $\vec{\vec{x}}^T\vec{\vec{x}}$ associated with eigenvalues

$$|\lambda_1| > |\lambda_2| > \cdots > |\lambda_p|.$$

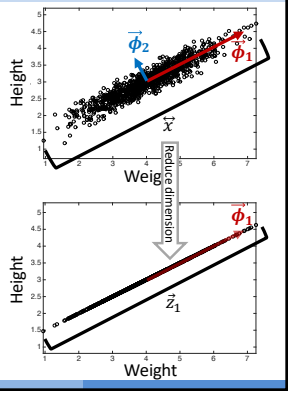
20

Principal Components

Geometrically, $\vec{\phi}_1$, which are the loadings of the first principal component, $\vec{z}_1$, designate the direction along which the data, $\vec{\vec{x}}$, vary the most.

$\vec{z}_1$ is then a reduced-dimension (1D) representation of the 2D $\vec{\vec{x}}$ data, along the most variable direction of $\vec{\vec{x}}$.

For instance, $\vec{\phi}_1$might be (0.8 Weight) + (0.6 Height). $0.8^2 + 0.6^2 = 1$, so the vector is properly normalized. If $\vec{\phi}_1 = (0.8,0.6)$, $\vec{\phi}_2$ must be perpendicular to $\vec{\phi}_1$. So, $\vec{\phi}_2 = \pm(-0.6,0.8)$.



21

Computing Principal Components

Solving

$$\vec{\phi}_1 = \underset{\|\vec{\phi}_1\|=1}{\operatorname{argmax}} \|\vec{z}_1\|^2$$

can be done in two ways:

1. Find the eigenvectors and eigenvalues of the covariance matrix, $\vec{\vec{C}}$, of the data, $\vec{\vec{x}}$.
2. Use Singular Value Decomposition (SVD) on the data matrix, $\vec{\vec{x}}$.

22

Computing Principal Components I

The covariance matrix, $\vec{\vec{C}}$, is defined elementwise as

$$C_{ij} = \begin{cases} \text{variance of } x_i & i = j \\ \text{covariance of } x_i \text{ and } x_j & i \neq j \end{cases}$$

We can calculate $\vec{\vec{C}}$ using $\vec{\vec{C}} = \frac{1}{n-1}\vec{\vec{x}}^T\vec{\vec{x}}$. The eigenvectors of $\vec{\vec{C}}$ are the directions of greatest variance in $\vec{\vec{x}}$. Ordering these eigenvectors by the absolute values of their eigenvalues, we get the 1st, 2nd, 3rd, … principal component directions. Diagonalizing $\vec{\vec{C}}$ (which is positive semidefinite):

$$\vec{\vec{C}} = \vec{\vec{V}}\vec{\vec{\Lambda}}\vec{\vec{V}}^T$$

Then $\vec{\vec{V}} = \begin{bmatrix} | & | & & | \\ v_1 & v_2 & \cdots & v_p \\ | & | & & | \end{bmatrix}$ and $\vec{\vec{\Lambda}} = \begin{bmatrix} \lambda_1 & & & 0 \\ & \lambda_2 & & \\ & & \ddots & \\ 0 & & & \lambda_p \end{bmatrix}$,

with $|\lambda_1| > |\lambda_2| > \cdots > |\lambda_p|$.

23

Computing Principal Components II

We can also compute the principal components directly using Singular Value Decomposition (SVD).
We can express any (sane…) matrix, including $\vec{\vec{x}}$, as the product of 3 matrices:

$$\vec{\vec{x}} = \vec{\vec{U}} \cdot \vec{\vec{\Sigma}} \cdot \vec{\vec{V}}^T$$

Here $\vec{\vec{U}}$ and $\vec{\vec{V}}$ are unitary matrices—i.e., $\vec{\vec{U}}^T = \vec{\vec{U}}^{-1}$ and $\vec{\vec{V}}^T = \vec{\vec{V}}^{-1}$, so $\vec{\vec{V}} \cdot \vec{\vec{V}}^T = \vec{\vec{V}} \cdot \vec{\vec{V}}^{-1} = \vec{\vec{I}}$ and the same for $\vec{\vec{U}}$, hence their columns form orthonormal bases; and $\vec{\vec{\Sigma}}$ is diagonal with the "singular values" of $\vec{\vec{X}}$ in descending order.

24

Computing Principal Components II

Plugging the SVD decomposition into the formula for $\vec{C}$:

$$\vec{C} = \frac{1}{n-1}\vec{X}^T\vec{X}$$
$$= \frac{1}{n-1}\vec{V}\vec{\Sigma}\vec{U}^T\vec{U}\vec{\Sigma}\vec{V}^T$$
$$= \frac{1}{n-1}\vec{V}\vec{\Sigma}^2\vec{V}^T$$
$$= \vec{V}\left(\frac{\vec{\Sigma}^2}{n-1}\right)\vec{V}^T$$

So, the columns of $\vec{V}$ are the principal directions, and

$$\frac{\Sigma_{i,i}^2}{n-1} = \lambda_i$$

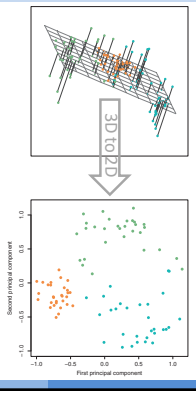
ranks them in ascending order.
Geometrically, $\vec{v}_1$ corresponds to the direction in $\vec{X}$ with highest variance. $\vec{v}_2$ is the direction with highest variance remaining (perpendicular to $\vec{v}_1$); and so on.

25

Reducing Dimensions

So far, we saw how to represent the data, $\vec{X}$, according to its "intrinsic directions", as $\vec{Z}$.

If we want to reduce dimensionality from p to q, we take only the first (highest) q eigenvalues and their corresponding eigenvectors, creating truncated versions of $\vec{V}$,$\vec{\Sigma}$, and $\vec{U}$: $\vec{V}^q$, $\vec{\Sigma}^q$, and $\vec{U}^q$.



26

Reducing Dimensions

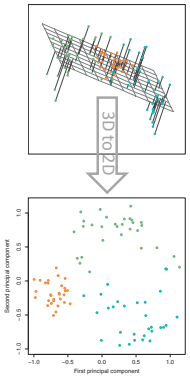
The reduced-dimension representation of $\vec{X}$ is $\vec{Z}^q$. $\vec{Z}^q$ results in a linear combination of all the features for each $\vec{\phi}_i$ (e.g., $\vec{\phi}_1 = 0.8\,Weight + 0.6\,Height$) and not in feature selection.

Often, the first few dimensions represent much of the overall variance of the data (given overall p dimensions):

$$\frac{\sum_{i=1}^{q}\lambda_i}{\sum_{i=1}^{p}\lambda_i}.$$

In that case,

$$\vec{X} \approx \vec{Z}^q \cdot \vec{\Phi}^{q\,-1}.$$



27

Scree Plot

But which approximation (i.e., which value of q) should we choose? Reducing to how many dimensions is good enough? We can decide how many principal components to use based on the cumulative proportion of variance explained, using a scree plot like the one on the right.

Generally, we set q to the "elbow" of the curve, where the curve makes a sharp turn. In the example on the right, perhaps $q = 2$ or 3.

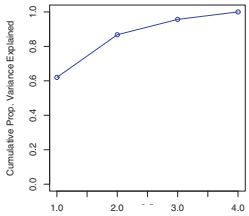


Image Source

28

PCA Pros & Cons

Advantages of Principal Component Analysis:

- Unsupervised preprocessing technique is labels agnostic, hence does not lead to double dipping (if used properly)
- Dimensionality reduction may lead to more interpretable data and data that is easier to visualize
- Running PCA does not rest on any special assumptions

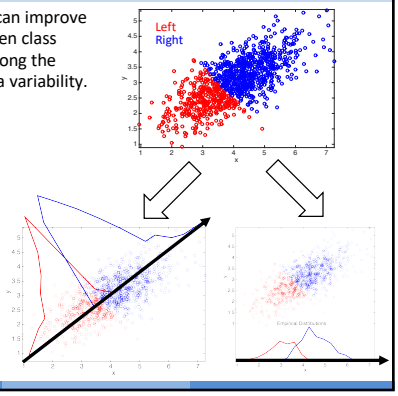
Disadvantages of Principal Component Analysis:

- Principal components (linear combinations) can also sometimes be difficult to interpret directly
- Nonlinear structure can be hard to model with PCA (though nonlinear PCA methods with kernels and other extensions do exist)

29

PCA for Classification Preprocessing

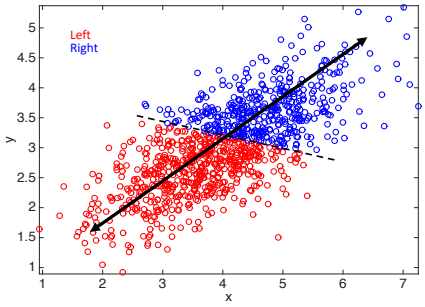
PCA before classification can improve classification accuracy when class differentiation is better along the directions of maximal data variability.



30

PCA for Classification Preprocessing

But PCA is not exposed to the labels. So, it will not do well when optimal class differentiation is not along the directions of maximal variability.



31

CONCLUSIONS

- Dimensionality reduction methods either remove some features of the data (feature selection) or create combinations of features and leave only ones optimal in some sense (e.g., PCA)
  - As a preprocessing step, this can potentially improve classification
  - It can also assist in visualizing the data
- Principal Component Analysis (PCA) is an unsupervised method (agnostic of labels) that finds "intrinsic" directions where the data's variance is maximized along the 1st direction, the remaining variance is maximized along the 2nd direction, and so on

33

Take 🏠 message for rest of course: PCA

- Dimensionality reduction methods attempt to simplify the data by reducing its number of features
- Preprocessing: they may reduce classification error
  - PCA: class differentiability along maximal variance direction
- PCA can improve data visualizability

34