Machine Learning I

Dimension reduction

Souhaib Ben Taieb

University of Mons

Table of contents

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

- One-dimensional projection
- Multiple projections

Computation of Principal Components

Example: National track records

Other dimension reduction methods

Recap: The dot product

Algebraic definition. The dot product of two vectors $\boldsymbol{a} = (a_1, \dots, a_p)^T$ and $\boldsymbol{b} = (b_1, \dots, b_p)^T$ is defined as

$$\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{a}^T \boldsymbol{b} = \boldsymbol{b}^T \boldsymbol{a} = \sum_{j=1}^{p} a_j b_j.$$

Geometric definition. The dot product of two Euclidean vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is defined as

$$\boldsymbol{a} \cdot \boldsymbol{b} = \|\boldsymbol{a}\| \|\boldsymbol{b}\| \cos(\theta),$$

where θ is the angle between $\boldsymbol{a}$ and $\boldsymbol{b}$. This implies that

$$\boldsymbol{a} \cdot \boldsymbol{a} = \|\boldsymbol{a}\|^2 = \boldsymbol{a}^T \boldsymbol{a}.$$

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

Computation of Principal Components

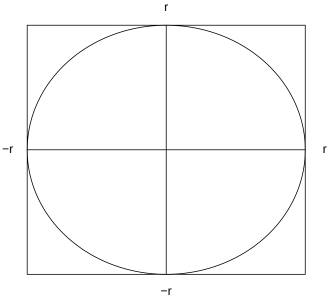
Example: National track records

Other dimension reduction methods

Hyper-cubes and hyper-spheres in high dimensions

Consider the **hyper-cube** $[-r, r]^p$ and the inscribed **hyper-sphere**. What does your intuition tell you about the relative sizes of these two objects as $p \to \infty$?

1. volume of sphere $>>$ volume of cube
2. volume of sphere $\approx$ volume of cube
3. volume of sphere $<<$ volume of cube

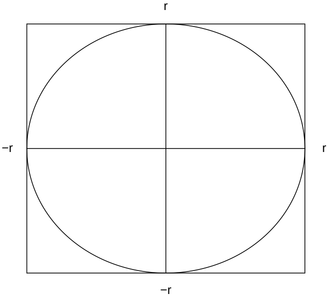


Fact 1

$$s_p = \frac{\text{Volume(hyper-sphere)}}{\text{Volume(hyper-cube)}} = \frac{\frac{r^p \pi^{p/2}}{\Gamma(\frac{p}{2}+1)}}{(2r)^p} = \left(\underbrace{\frac{\sqrt{\pi}}{2}}_{<1} \right)^p \frac{1}{\Gamma(\frac{p}{2}+1)},$$

where $\Gamma(\cdot)$ is the Gamma function.

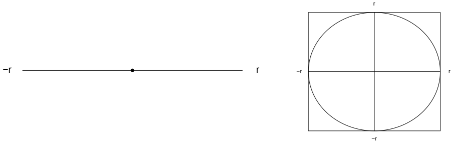
- s_p does not depend on r, just on p
- As the dimension increases, the volume of the sphere is much smaller (infinitesimal) than that of the cube
- How is this going against intuition?



Fact 1

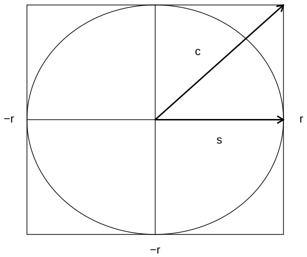
p	1	2	3	4	5	6
s_p	1	.785	.524	.308	.164	.080

- Not very surprising. we can see it even in lower dimensions. For $p = 1$, they have the same volume. For $p = 2$, area of the circle is already smaller.



- As the dimension increases, the volume of the area between the cube and the sphere becomes larger

Fact 2



- $\boldsymbol{c} = (r, r, \dots, r)^T \in \mathbb{R}^p$
- $\boldsymbol{s} = (r, 0, \dots, 0)^T \in \mathbb{R}^p$

- As d increases, $\boldsymbol{c}$ becomes **infinitely larger** than $\boldsymbol{s}$

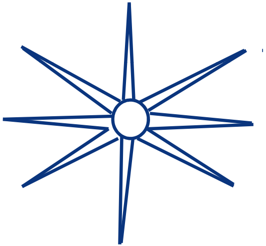
$$\frac{\|\boldsymbol{c}\|_2^2}{\|\boldsymbol{s}\|_2^2} = \frac{pr^2}{r^2} = p \overset{p\to\infty}{\longrightarrow} \infty,$$

- As d increases, $\boldsymbol{c}$ becomes **orthogonal** to $\boldsymbol{s}$

$$\cos(\theta) = \frac{\boldsymbol{c}^T\boldsymbol{s}}{\|\boldsymbol{c}\|_2\|\boldsymbol{s}\|_2} = \frac{r^2}{\sqrt{pr^2r^2}} = \frac{1}{\sqrt{p}} \overset{p\to\infty}{\longrightarrow} 0.$$

Picture in High Dimensions

In high dimensions the picture you should have in mind is

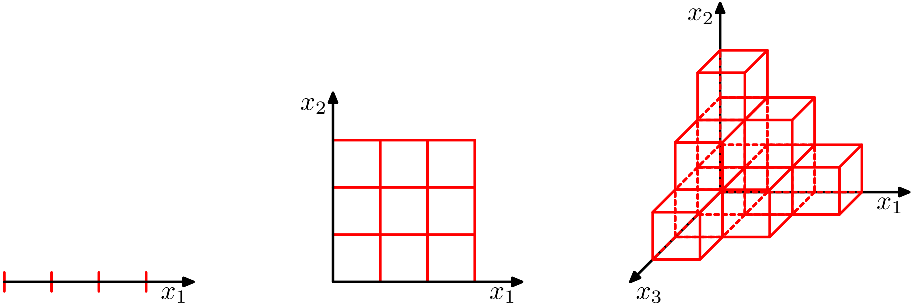


→ In high dimensions, all the volume of the cube is in “the spikes”

Other facts

- High-dimensional balls (the space bounded by a sphere) have a **vanishing volume**.
- The volume of a high-dimensional ball is **concentrated** in its crust
  - Most data points are closer to the boundary of the sample space than to any other data point
  - Prediction is much more difficult near the edges of the training sample: extrapolation vs interpolation
- All the points are at a **similar distance** one from the others
  - The notion of nearest-points vanishes
- High dimensional spaces are strange → **never trust your intuition in high dimensions!**

Other facts



If we divide a region of a space into regular cells, then the number of such cells grows exponentially with the dimensionality of the space $\implies$ We would need an exponentially large quantity of training data points in order to ensure that the cells are not empty.

The curse of dimensionality

"The **curse of dimensionality** *refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience. The expression was coined by Richard E. Bellman..."*.

Wikipedia, May, 2023.

Can we learn in high dimension?

Yes!

- First, real data will often be confined to a region of the space having **lower intrinsic dimensionality**. The data *lives* in a low dimensional subspace.
- Second, real data will typically exhibit some **smoothness** properties (at least locally)

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

Computation of Principal Components

Example: National track records

Other dimension reduction methods

Dimensionality reduction

Mapping data to a low-dimensional space is called dimensionality reduction
A mapping to a space that's easier to manipulate/visualize is called a representation, and learning such a mapping is representation learning
Why dimensionality reduction?

- Curse of dimensionality
- Intrinsic dimensionality
- Visualization (in two/three dimensions)
- Reduce computation and storage

We avoid unnecessary dimensions/variables by checking if:

- variables are **not useful** (for my learning problem)
- variables are **not independent** (redundancy)

Dimension reduction methods

- Variable selection vs feature extraction
- Unsupervised vs supervised
- Linear vs nonlinear
- ...

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

- One-dimensional projection
- Multiple projections

Computation of Principal Components

Example: National track records

Other dimension reduction methods

Principal components analysis

Principal components analysis (PCA) is an unsupervised linear feature extraction method.

PCA produces a **low-dimensional representation** of a dataset (with continuous variables). It finds a sequence of **linear combinations of the variables** that have **maximal variance**, and are **mutually uncorrelated**. Since PCA is a linear model, the mapping will be a **projection**.

PCA is one of the **oldest** dimension reduction technique, and has been rediscovered many times in many fields, so it is also known as the **Karhunen-Loève** transformation, the **Hotelling** transformation, the method of **empirical orthogonal functions**, etc.

Principal components analysis

- We start with p-dimensional vectors, and want to summarize them by **projecting** down into a q-dimensional subspace. Our summary will be the projection of the original vectors on to q directions, the **principal components**, which span the subspace.
- There are several **equivalent way**s of deriving the principal components mathematically.
- **Maximum variance formulation**: We look for the projection which maximize the variance.
- **Minimum error reconstruction formulation**: We look for the projection with the smallest average mean-squared distance between the original vectors and their projections on to the principal components.

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

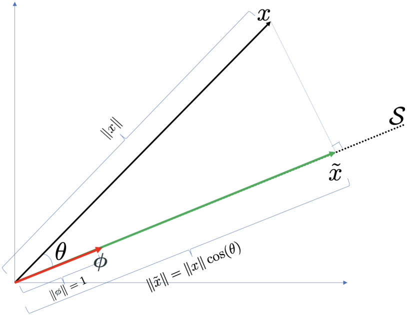
- One-dimensional projection
- Multiple projections

Computation of Principal Components

Example: National track records

Other dimension reduction methods

Projection onto a one-dimensional subspace



- $\tilde{\boldsymbol{x}}$ is the projection of $\boldsymbol{x}$ on the subspace $\mathcal{S}$ (denoted $\text{Proj}_S(\boldsymbol{x})$), i.e. a line along the unit vector ϕ.
- $\tilde{\boldsymbol{x}} = \text{Proj}_S(\boldsymbol{x}) = \underbrace{(\boldsymbol{x} \cdot \phi)}_{\text{length}} \underbrace{\phi}_{\text{direction}} = (\boldsymbol{x}^T\phi)\phi = \|\tilde{\boldsymbol{x}}\|\phi$
- The (orthogonal) projection can be represented by a projection matrix $\boldsymbol{P} = \phi\phi^T$. In fact, we have $\tilde{\boldsymbol{x}} = (\boldsymbol{x}^T\phi)\phi = \phi(\phi^T\boldsymbol{x}) = \phi\phi^T\boldsymbol{x}$.

One-dimensional projection

Let us consider a dataset composed of n p-dimensional data points $\boldsymbol{x}_i \in \mathbb{R}^p$, where $i = 1, \ldots, n$. We will assume that the data has been **centered**, so that every variable has mean zero, i.e. $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = \mathbf{0}_p$.

We want to **project** them on to a line through the origin, specified by a unit vector along it, $\phi \in \mathbb{R}^p$ $(\|\phi\|_2 = 1)$.

Recall that the projection of $\boldsymbol{x}_i$ on to the line is given by the **dot product** $\boldsymbol{x}_i \cdot \phi$, and the actual coordinate of the point in p-dimensional space is given by $\tilde{\boldsymbol{x}}_i = (\boldsymbol{x}_i \cdot \phi)\phi$.

One-dimensional projection - Minimize reconstruction error

How big is the difference between the **projected vectors** and the **original vectors**? For a given direction ϕ and any $\boldsymbol{x}_i$, we have

$$\begin{aligned}\|\boldsymbol{x}_i - (\boldsymbol{x}_i \cdot \phi)\phi\|^2 &= (\boldsymbol{x}_i - (\boldsymbol{x}_i \cdot \phi)\phi) \cdot (\boldsymbol{x}_i - (\boldsymbol{x}_i \cdot \phi)\phi) \\ &= \boldsymbol{x}_i \cdot \boldsymbol{x}_i - \boldsymbol{x}_i \cdot (\boldsymbol{x}_i \cdot \phi)\phi - (\boldsymbol{x}_i \cdot \phi)\phi \cdot \boldsymbol{x}_i \\ &+ (\boldsymbol{x}_i \cdot \phi)\phi \cdot (\boldsymbol{x}_i \cdot \phi)\phi \\ &= \boldsymbol{x}_i \cdot \boldsymbol{x}_i - (\boldsymbol{x}_i \cdot \phi)^2 \\ &= \|\boldsymbol{x}_i\|^2 - (\boldsymbol{x}_i \cdot \phi)^2\end{aligned}$$

The total reconstruction error is

$$\frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - (\boldsymbol{x}_i \cdot \phi)\phi\|^2 = \frac{1}{n}\left(\sum_{i=1}^{n} \|\boldsymbol{x}_i\|^2 - \sum_{i=1}^{n}(\boldsymbol{x}_i \cdot \phi)^2\right)$$

The first term does not depend on ϕ. Hence, *minimizing the total reconstruction* error is **equivalent** to *maximize the second term*.

One-dimensional projection - Maximize variance

Note that since $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i = \mathbf{0}_p$, the mean of the projections will be zero. In fact, we have

$$\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i \cdot \phi)\phi = (\boldsymbol{x}_1 \cdot \phi)\phi + \cdots + (\boldsymbol{x}_n \cdot \phi)\phi = \left(\left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i\right) \cdot \phi\right)\phi = \mathbf{0}_p$$

Let $z_i = \boldsymbol{x}_i \cdot \phi$ and $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} z_i$. We can write

$$\begin{aligned}\frac{1}{n}\sum_{i=1}^{n}(z_i - \bar{z})^2 &= \frac{1}{n}\sum_{i=1}^{n} z_i^2 \quad (\text{since } \bar{z} = 0)\\ &= \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i \cdot \phi)^2\end{aligned}$$

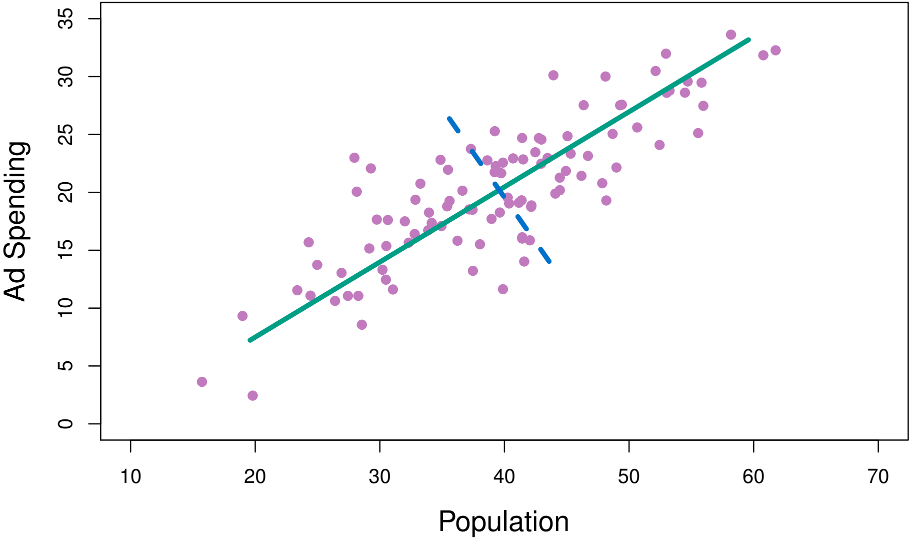
Since $\bar{z} = 0$, maximizing the RHS is equivalent to maximize the LHS, which represents the **variance** of the projections.

One-dimensional projection - Reconstruction error

Note that we can also write

$$\begin{aligned}\|\boldsymbol{x}_i - (\boldsymbol{x}_i \cdot \phi)\phi\|^2 &= \left\|\boldsymbol{x}_i - \phi\phi^T\boldsymbol{x}_i\right\|^2 \\ &= (\boldsymbol{x}_i - \phi\phi^T\boldsymbol{x}_i)^T(\boldsymbol{x}_i - \phi\phi^T\boldsymbol{x}_i) \\ &= \boldsymbol{x}_i^T\boldsymbol{x}_i - 2\boldsymbol{x}_i^T\phi\phi^T\boldsymbol{x}_i + \boldsymbol{x}_i^T\phi\phi^T\phi\phi^T\boldsymbol{x}_i \\ &= \boldsymbol{x}_i^T\boldsymbol{x}_i - \boldsymbol{x}_i^T\phi\phi^T\boldsymbol{x}_i \\ &= \|\boldsymbol{x}_i\|^2 - (\boldsymbol{x}_i \cdot \phi)^2\end{aligned}$$

PCA Example



One-dimensional projection - Maximize variance

$$\underset{\phi\in\mathbb{R}^p,\|\phi\|=1}{\text{min.}} \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i-(\boldsymbol{x}_i\cdot\phi)\phi\|^2 \equiv \underset{\phi\in\mathbb{R}^p,\|\phi\|=1}{\text{max.}} \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i\cdot\phi)^2$$

If we stack our n data points into an $n \times p$ matrix, $\boldsymbol{X}$, then the projections are given by $\boldsymbol{X}\phi$, which is an $n \times 1$ matrix. We have

$$\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i\cdot\phi)^2 = \frac{1}{n}(\boldsymbol{X}\phi)^T\boldsymbol{X}\phi = \frac{1}{n}\phi^T\boldsymbol{X}^T\boldsymbol{X}\phi = \phi^T\boldsymbol{C}\phi,$$

where $\boldsymbol{C} = \frac{\boldsymbol{X}^T\boldsymbol{X}}{n}$ is the covariance matrix of the data (since the variables are centered).

One-dimensional projection - Optimization problem

The problem reduces to the following constrained optimization problem:

$$\underset{\phi \in \mathbb{R}^p, \|\phi\|=1}{\text{max.}} \phi^T \boldsymbol{C} \phi.$$

It can be shown that the solution is given by $\phi = \boldsymbol{v}$ where $\boldsymbol{v}$ is the eigenvector associated with the largest eigenvalue of $\boldsymbol{C}$.

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

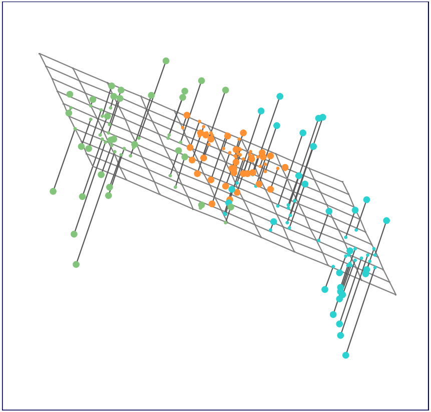
- One-dimensional projection
- Multiple projections

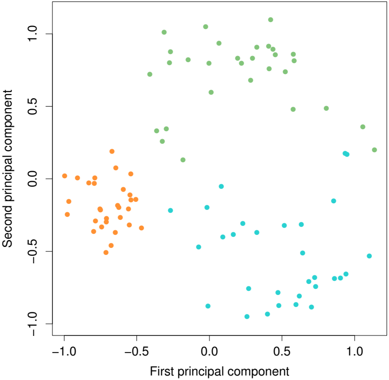
Computation of Principal Components

Example: National track records

Other dimension reduction methods

Multiple projections





Multiple projections

In general, we want to project on **multiple principal components**. If those components are orthogonal and have the unit vectors $\phi_1, \ldots, \phi_M$, then the image of $\boldsymbol{x}_i$ is its projection into the space spanned by these vectors,

$$\sum_{m=1}^{M} (\boldsymbol{x}_i \cdot \phi_m) \phi_m.$$

Let $\boldsymbol{\Phi} = [\phi_1 \cdots \phi_M] \in \mathbb{R}^{p \times M}$ be the matrix with all vectors ϕ_m stacked. Note that we have $\boldsymbol{\Phi}^T \boldsymbol{\Phi} = \boldsymbol{I}_M$.

The mean of the projection on to each component is still zero. We can also show that minimizing the reconstruction error is equivalent to maximimize the sum of the variances of the projections onto the components.

Multiple projections - Minimize reconstruction error

With multiple principal comonents, the total reconstruction error is

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}_i - \sum_{m=1}^{M}(\boldsymbol{x}_i \cdot \phi_{\boldsymbol{m}})\phi_{\boldsymbol{m}}\right\|^2 &= \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}_i - \sum_{m=1}^{M}\phi_{\boldsymbol{m}}\phi_{\boldsymbol{m}}^T\boldsymbol{x}_i\right\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\left\|\boldsymbol{x}_i - \Phi\Phi^T\boldsymbol{x}_i\right\|^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \Phi\Phi^T\boldsymbol{x}_i)^T(\boldsymbol{x}_i - \Phi\Phi^T\boldsymbol{x}_i) \\
&= \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^T\boldsymbol{x}_i - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i^T\Phi\Phi^T\boldsymbol{x}_i \\
&= \frac{1}{n}\sum_{i=1}^{n}\|\boldsymbol{x}_i\|^2 - \frac{1}{n}\sum_{i=1}^{n}\left\|\Phi^T\boldsymbol{x}_i\right\|^2
\end{aligned}$$

Multiple projections - Maximize variance

We want to maximize

$$\begin{aligned}
\frac{1}{n}\sum_{i=1}^{n}\left\|\Phi^{T}\boldsymbol{x}_i\right\|^2 &= \frac{1}{n}\sum_{i=1}^{n}\sum_{m=1}^{M}(\boldsymbol{x}_i\cdot\phi_m)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}\sum_{m=1}^{M}(\phi_m^T\boldsymbol{x}_i)(\boldsymbol{x}_i^T\phi_m) \\
&= \frac{1}{n}\sum_{m=1}^{M}\sum_{i=1}^{n}\phi_m^T\boldsymbol{x}_i\boldsymbol{x}_i^T\phi_m \\
&= \sum_{m=1}^{M}\phi_m^T\frac{\boldsymbol{X}^T\boldsymbol{X}}{n}\phi_m \\
&= \sum_{m=1}^{M}\phi_m^T\boldsymbol{C}\phi_m
\end{aligned}$$

Multiple projections - Optimization problem

The problem reduces to a constrained optimization problem:

$$\max_{\mathbf{\Phi}=[\phi_1 \cdots \phi_M] \in \mathbb{R}^{pxM}, \mathbf{\Phi}^T \mathbf{\Phi} = \boldsymbol{I}_M} \sum_{m=1}^{M} \phi_m^T \boldsymbol{C} \phi_m.$$

We can show that the solution is given by $\mathbf{\Phi} = \boldsymbol{V}$ where $\boldsymbol{V}$ is the square matrix whose *m*th column is the *m*th eigenvector of $\boldsymbol{C}$ (the columns of $\boldsymbol{V}$ are orthonormal, i.e. $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}$).

Geometry of PCA

- The first **loading vector** ϕ_1 defines a direction in feature space along which the data vary the most.
- If we project the n data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ onto a direction ϕ, the projected values, $z_i = \boldsymbol{x}_i \cdot \phi, i = 1, \ldots, n$, are the **principal component scores**.
- The first **principal component** is the linear combination of the features that has maximal variance.
- The second **principal component** is the linear combination of the features that has maximal variance among all linear combinations that are **uncorrelated** with the first principal component.
- And so on.
- There are at most $M = \min(n-1, p)$ principal components.

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

Computation of Principal Components

Example: National track records

Other dimension reduction methods

Computation of PCs (method I)

1. Compute the covariance matrix (after centering the columns of $\boldsymbol{X}$)

$$\boldsymbol{C} = \frac{\boldsymbol{X}^T \boldsymbol{X}}{n}$$

2. Find eigenvalues and eigenvectors:

$$\boldsymbol{C} = \boldsymbol{V} \boldsymbol{D} \boldsymbol{V}^T$$

   where $\boldsymbol{V}$ is the square matrix whose *m*th column is the *m*th eigenvector of $\boldsymbol{C}$ (the columns of $\boldsymbol{V}$ are orthonormal, i.e. $\boldsymbol{V}^T \boldsymbol{V} = \boldsymbol{I}$), and $\boldsymbol{D}$ is a diagonal matrix whose diagonal elements are the corresponding eigenvalues.

3. Compute PCs:
   - $\Phi = \boldsymbol{V}$
   - $\boldsymbol{Z} = \boldsymbol{X} \Phi$.

Computation of PCs (method II)

Singular Value Decomposition

$$\boldsymbol{X} = \boldsymbol{U}\Lambda\boldsymbol{V}^T$$

- $\boldsymbol{X}$ is $n \times p$ matrix
- $\boldsymbol{U}$ is $n \times r$ matrix with orthonormal columns ($\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$). The columns of $\boldsymbol{U}$ are called the left-singular vectors of $\boldsymbol{X}$.
- Λ is $r \times r$ diagonal matrix with non-negative elements (called singular values).
- $\boldsymbol{V}$ is $p \times r$ matrix with orthonormal columns ($\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$). The columns of $\boldsymbol{V}$ are called the right-singular vectors of $\boldsymbol{X}$

It is always possible to **uniquely** decompose a matrix in this way.

Computation of PCs (method II)

1. Compute SVD: $\boldsymbol{X} = \boldsymbol{U}\Lambda\boldsymbol{V}^T$.
2. Compute PCs: $\Phi = \boldsymbol{V}$. $\boldsymbol{Z} = \boldsymbol{X}\Phi$.

Relationship with covariance:

$$\boldsymbol{C} = \boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\Lambda\boldsymbol{U}^T\boldsymbol{U}\Lambda\boldsymbol{V}^T = \boldsymbol{V}\Lambda^2\boldsymbol{V}^T = \boldsymbol{VDV}^T$$

- Eigenvalues of $\boldsymbol{C}$ are squares of singular values of $\boldsymbol{X}$.
- Eigenvectors of $\boldsymbol{C}$ are the right-singular vectors of $\boldsymbol{X}$.
- The PC directions $\phi_{\mathbf{1}}, \phi_2, \phi_3, \ldots, \phi_M$ are the right-singular vectors of $\boldsymbol{X}$.

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

Computation of Principal Components

Example: National track records

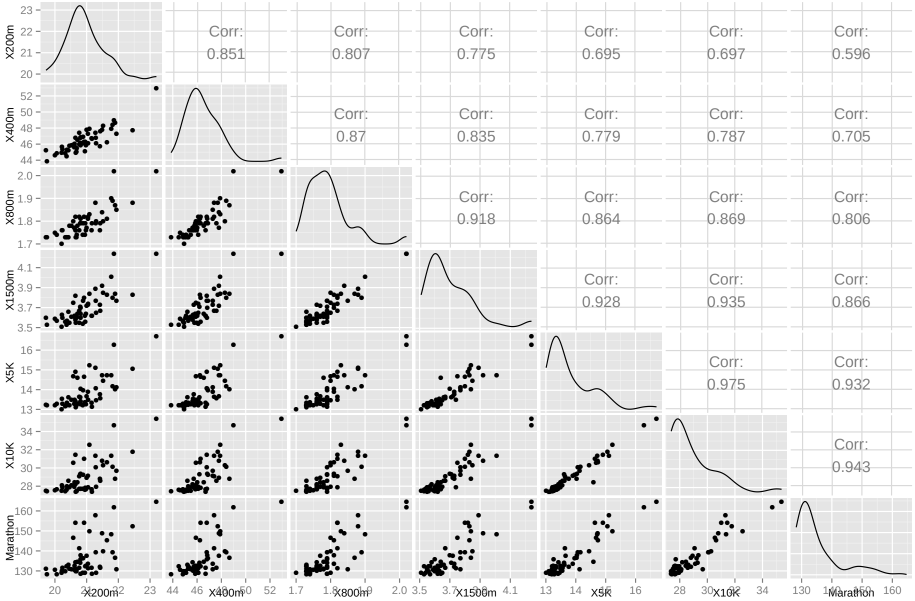
Other dimension reduction methods

Example: National track records

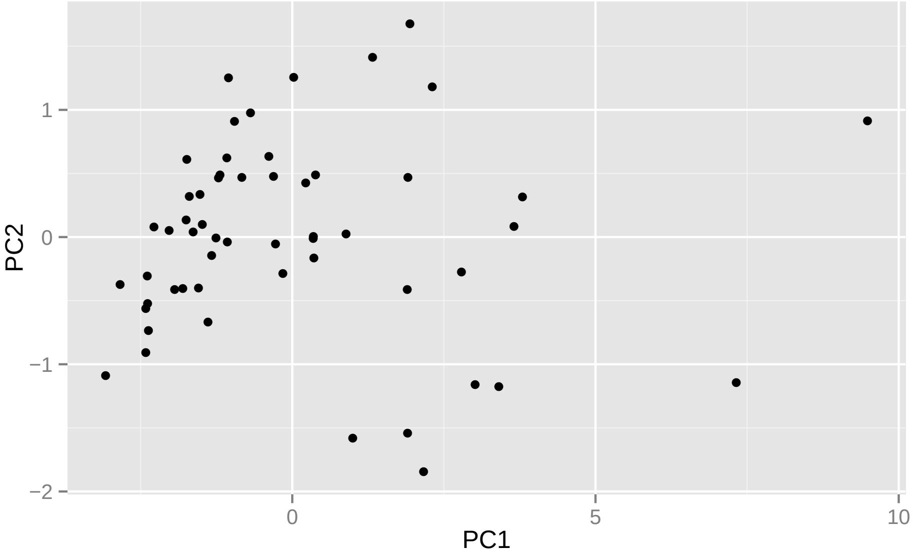
The data on national track records for men are listed in the following table (as at 1984):

Country	**100m**	**200m**	**400m**	**800m**	**1500m**	**5000m**	**10000m**	**Marathon**
	(s)	*(s)*	*(s)*	*(min)*	*(min)*	*(min)*	*(min)*	*(min)*
Argentina	10.39	20.81	46.84	1.81	3.70	14.04	29.36	137.72
Australia	10.31	20.06	44.84	1.74	3.57	13.28	27.66	128.30
Austria	10.44	20.81	46.82	1.79	3.60	13.26	27.72	135.90
Belgium	10.34	20.68	45.04	1.73	3.60	13.22	27.45	129.95
Bermuda	10.28	20.58	45.91	1.80	3.75	14.68	30.55	146.62
Brazil	10.22	20.43	45.21	1.73	3.66	13.62	28.62	133.13
⋮								
Turkey	10.71	21.43	47.60	1.79	3.67	13.56	28.58	131.50
USA	9.93	19.75	43.86	1.73	3.53	13.20	27.43	128.22
USSR	10.07	20.00	44.60	1.75	3.59	13.20	27.53	130.55
W.Samoa	10.82	21.86	49.00	2.02	4.24	16.28	34.71	161.83

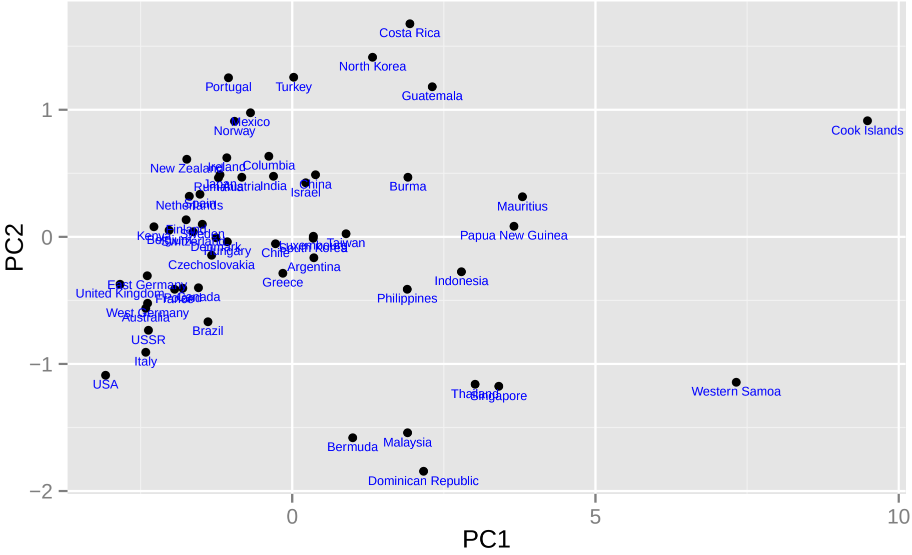
Example: National track records



Example: National track records



Example: National track records



Example: National track records

```
> pca
Standard deviations:
[1] 2.573 0.937 0.399 0.352 0.283 0.261 0.215 0.150

Rotation:
PC1     PC2    PC3     PC4    PC5     PC6      PC7       PC8
X100m    0.318  0.5669  0.332 -0.1276  0.263 -0.5937  0.13624
X200m    0.337  0.4616  0.361  0.2591 -0.154  0.6561 -0.11264
X400m    0.356  0.2483 -0.560 -0.6523 -0.218  0.1566 -0.00285
X800m    0.369  0.0124 -0.532  0.4800  0.540 -0.0147 -0.23802
X1500m   0.373 -0.1398 -0.153  0.4045 -0.488 -0.1578  0.61001
X5K      0.364 -0.3120  0.190 -0.0296 -0.254 -0.1413 -0.59130
X10K     0.367 -0.3069  0.182 -0.0801 -0.133 -0.2190 -0.17687
Marathon 0.342 -0.4390  0.263 -0.2995  0.498  0.3153  0.39882
```

Proportion of variance explained

Total variance in data (assuming variables centered at 0):

$$\mathrm{TV} = \sum_{j=1}^{p} \mathrm{Var}(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=1}^{n} x_{ij}^2$$

Variance explained by mth PC:

$$V_m = \mathrm{Var}(Z_m) = \frac{1}{n} \sum_{i=1}^{n} z_{im}^2$$

$$\mathrm{TV} = \sum_{m=1}^{M} V_m \qquad \text{where } M = \min(n-1, p).$$

Proportion of variance explained:

$$\mathrm{PVE}_m = V_m / TV$$

Scree plots and biplots

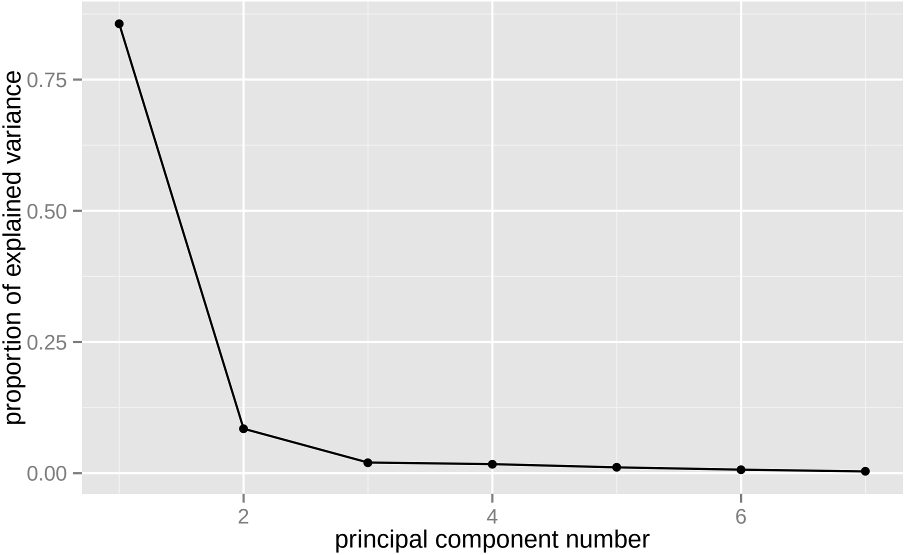
Scree plot

Plot of variance explained by each component vs number of component.

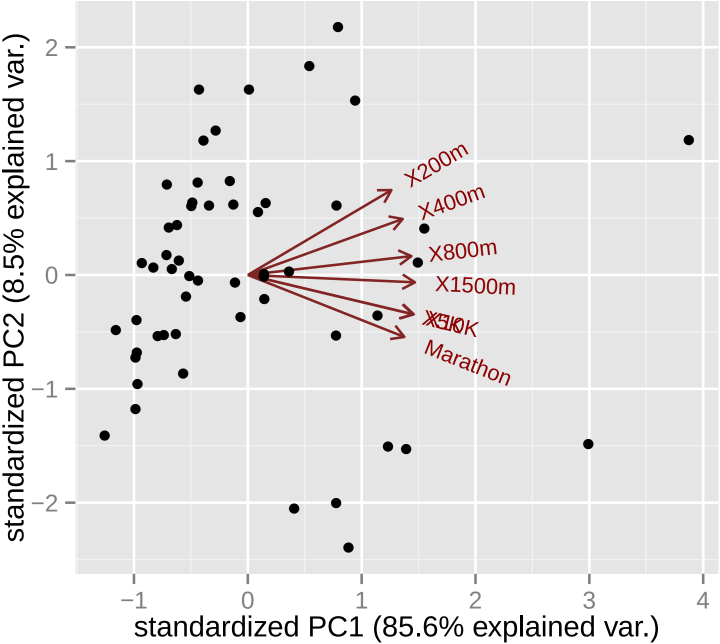
Biplot

Plot of PC2 vs PC1, overlaid with directions of the loading vectors (ϕ_1, ϕ_2).

Scree plot


proportion of explained variance
0.75
0.50
0.25
0.00
2
4
6
principal component number

Biplot


X200m
X400m
X800m
X1500m
X5K
X10K
Marathon
2
1
0
−1
−2
−1
0
1
2
3
4
standardized PC1 (85.6% explained var.)
standardized PC2 (8.5% explained var.)

Scaling

- If the variables are in different units, scaling each to have standard deviation equal to one is recommended.
- If they are in the same units, you might or might not scale the variables.

Outline

The strange geometry in high dimensions

Dimensionality reduction

Principal components analysis

Computation of Principal Components

Example: National track records

Other dimension reduction methods

Other dimension reduction methods

- Nonlinear PCA, Kernel PCA, Sparse PCA, etc.
- Multidimensional scaling (MDS), Independent component analysis (ICA), etc.
- Partial least squares (PLS), Canonical correlation analysis (CCA), Factor analysis (FA), etc.
- Isomap, diffusion maps, t-SNE, autoencoders, etc.