PCA

Professor Ameet Talwalkar

Slide Credit: Professor Fei Sha

Outline

1. Administration

2. Review of last lecture

3. Course Evaluation

4. PCA

Announcements

- Graded HW5 available today
- HW6 released last Tuesday, due on Friday in section
- Project report guideline posted online

Upcoming Class Schedule

- Today: PCA – Last day of lecture!
- Thursday, 12/3: In-class office hours for project (9:00-11am)
  - You can sign up for a 10 minute slot on the Doodle poll
- Friday 12/4: Nikos section (covers midterm, HW6 questions)
  - Hand in HW6
- Friday, 12/11: Poster Presentation + Project Report

Outline

1. Administration

2. Review of last lecture
    - GMMs and Incomplete Data
    - EM Algorithm

3. Course Evaluation

4. PCA

Gaussian mixture models: formal definition

A Gaussian mixture model has the following density function for $\boldsymbol{x}$

$$p(\boldsymbol{x}) = \sum_{k=1}^{K} \omega_k N(\boldsymbol{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

- K: the number of Gaussians — they are called (mixture) components
- $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$: mean and covariance matrix of the k-th component
- ω_k: mixture weights – priors on each component that satisfy:

$$\forall\ k,\ \omega_k > 0, \quad \text{and} \quad \sum_k \omega_k = 1$$

- Given *unlabeled* data, $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$, we must learn:
  - *parameters* of Gaussians
  - *mixture components*

Parameter estimation for GMMs: complete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Complete Data: We (unrealistically) assume z is observed for every $\boldsymbol{x}$,

$$\mathcal{D}' = \{\boldsymbol{x}_n, z_n\}_{n=1}^{N}$$

MLE: Maximize the complete likelihood

$$\boldsymbol{\theta} = \arg\max \log \mathcal{D}' = \sum_n \log p(\boldsymbol{x}_n, z_n)$$

Easy problem to solve – intuitive, closed form solution exists!

Parameter estimation for GMMs: Incomplete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Incomplete Data

Our data contains observed and unobserved data, and hence is incomplete

Parameter estimation for GMMs: Incomplete data

GMM Parameters

$$\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^{K}$$

Incomplete Data

Our data contains observed and unobserved data, and hence is incomplete

- Observed: $\mathcal{D} = \{\boldsymbol{x}_n\}$
- Unobserved (hidden): $\{\boldsymbol{z}_n\}$

Goal Obtain the maximum likelihood estimate of $\boldsymbol{\theta}$:

$$\begin{aligned}\boldsymbol{\theta} = \arg\max \ell(\boldsymbol{\theta}) &= \arg\max \sum_n \log p(\boldsymbol{x}_n|\boldsymbol{\theta}) \\ &= \arg\max \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})\end{aligned}$$

The objective function $\ell(\boldsymbol{\theta})$ is called the *incomplete* log-likelihood.

Issue with Incomplete log-likelihood

No simple way to optimize the incomplete log-likelihood (exercise: try to take derivative with respect to parameters, set it to zero and solve)

EM algorithm provides a strategy for iteratively optimizing this function

Two steps as they apply to GMM:

- E-step: 'guess' values of the z_n using existing values of $\boldsymbol{\theta}$
- M-step: solve for new values of $\boldsymbol{\theta}$ given imputed values for z_n (maximize complete likelihood!)

E-step: Soft cluster assignments

We define γ_{nk} as $p(z_n = k|\boldsymbol{x}_n, \boldsymbol{\theta})$

- This is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}$
- Recall that in complete data setting γ_{nk} was binary
- Now it's a "soft" assignment of $\boldsymbol{x}_n$ to k-th component, with $\boldsymbol{x}_n$ assigned to each component with some probability

Given an estimate of $\boldsymbol{\theta} = \{\omega_k, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1}^K$, we can compute γ_{nk} as follows:

$$\begin{aligned}\gamma_{nk} &= p(z_n = k|\boldsymbol{x}_n) \\ &= \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{p(\boldsymbol{x}_n)} \\ &= \frac{p(\boldsymbol{x}_n|z_n = k)p(z_n = k)}{\sum_{k'=1}^K p(\boldsymbol{x}_n|z_n = k')p(z_n = k')}\end{aligned}$$

M-step: Maximimize complete likelihood

Recall definition of complete likelihood from earlier:

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Previously γ_{nk} was binary, but now we define $\gamma_{nk} = p(z_n = k | \boldsymbol{x}_n)$ (E-step)

M-step: Maximimize complete likelihood

Recall definition of complete likelihood from earlier:

$$\sum_n \log p(\boldsymbol{x}_n, z_n) = \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(\boldsymbol{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

Previously γ_{nk} was binary, but now we define $\gamma_{nk} = p(z_n = k | \boldsymbol{x}_n)$ (E-step)

We get the same simple expression for the MLE as before!

$$\omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \boldsymbol{x}_n$$

$$\boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\boldsymbol{x}_n - \boldsymbol{\mu}_k)(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^{\mathrm{T}}$$

Intuition: Each point now contributes some fractional component to each of the parameters, with weights determined by γ_{nk}

EM procedure for GMM

Alternate between estimating γ_{nk} and estimating $\boldsymbol{\theta}$

- Initialize $\boldsymbol{\theta}$ with some values (random or otherwise)
- Repeat
  - E-Step: Compute γ_{nk} using the current $\boldsymbol{\theta}$
  - M-Step: Update $\boldsymbol{\theta}$ using the γ_{nk} we just computed
- Until Convergence

EM procedure for GMM

Alternate between estimating γ_{nk} and estimating $\boldsymbol{\theta}$

- Initialize $\boldsymbol{\theta}$ with some values (random or otherwise)
- Repeat
  - E-Step: Compute γ_{nk} using the current $\boldsymbol{\theta}$
  - M-Step: Update $\boldsymbol{\theta}$ using the γ_{nk} we just computed
- Until Convergence

Remaining questions

- How does GMM relate to K-means?
- Is this procedure reasonable, i.e., are we optimizing a sensible criterion?
- Will this procedure converge?

GMMs and K-means

GMMs provide probabilistic interpretation for K-means

GMMs reduce to K-means under the following assumptions (in which case EM for GMM parameter estimation simplifies to K-means):

- Assume all Gaussians have $\sigma^2 \boldsymbol{I}$ covariance matrices
- Further assume $\sigma \rightarrow 0$, so we only need to estimate $\boldsymbol{\mu}_k$, i.e., means

K-means is often called "hard" GMM or GMMs is called "soft" K-means

The posterior γ_{nk} provides a probabilistic assignment for $\boldsymbol{x}_n$ to cluster k

EM algorithm: motivation and setup

- General procedure to estimate parameters for probabilistic models with hidden/latent variables
- Suppose the model is given by a joint distribution

$$p(\boldsymbol{x}|\boldsymbol{\theta}) = \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}|\boldsymbol{\theta})$$

- Given incomplete data $\mathcal{D} = \{\boldsymbol{x}_n\}$ our goal is maximimize incomplete log likelihood, $\ell(\boldsymbol{\theta}) = \sum_n \log \sum_{\boldsymbol{z}_n} p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})$
- log-sum form of incomplete log-likelihood is difficult to work with
- EM: construct lower bound on $\ell(\boldsymbol{\theta})$ (E-step) and optimize it (M-step)

A lower bound

- Jensen's inequality: $f(\mathbb{E}X) \geq \mathbb{E}f(X)$ for concave function f
- Using Jensen's, we can show that for any distribution $q(\boldsymbol{z})$ over $\boldsymbol{z}$:

$$\ell(\boldsymbol{\theta}) \geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

A lower bound

- Jensen's inequality: $f(\mathbb{E}X) \geq \mathbb{E}f(X)$ for concave function f
- Using Jensen's, we can show that for any distribution $q(\boldsymbol{z})$ over $\boldsymbol{z}$:

$$\ell(\boldsymbol{\theta}) \geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- We want a *tight* lower bound, and given some current estimate $\boldsymbol{\theta}^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- To achieve $f(\mathbb{E}X) = \mathbb{E}f(X)$ it is sufficient for X to be a constant random variable, i.e.,

A lower bound

- Jensen's inequality: $f(\mathbb{E}X) \geq \mathbb{E}f(X)$ for concave function f
- Using Jensen's, we can show that for any distribution $q(\boldsymbol{z})$ over $\boldsymbol{z}$:

$$\ell(\boldsymbol{\theta}) \geq \sum_n \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta})}{q(\boldsymbol{z}_n)}$$

- We want a *tight* lower bound, and given some current estimate $\boldsymbol{\theta}^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\boldsymbol{\theta}^t$
- To achieve $f(\mathbb{E}X) = \mathbb{E}f(X)$ it is sufficient for X to be a constant random variable, i.e., choose $q(\boldsymbol{z}_n) \propto p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^t)$!
- We can show that $q(\boldsymbol{z}_n) = p(\boldsymbol{z}_n | \boldsymbol{x}_n; \boldsymbol{\theta}^t)$, which is the posterior distribution of z_n given $\boldsymbol{x}_n$ and $\boldsymbol{\theta}^t$

E and M Steps

Our simplified expression

$$
\begin{aligned}
\ell(\boldsymbol{\theta}^t) &= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)} \\
&= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t) - p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)
\end{aligned}
$$

E-Step: For all n, compute $q(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$

E and M Steps

Our simplified expression

$$\begin{aligned}\ell(\boldsymbol{\theta}^t) &= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\log\frac{p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)} \\ &= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta}^t) - p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\log p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\end{aligned}$$

E-Step: For all n, compute $q(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$

We can view it as computing the *expected (complete) log-likelihood*:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta}) = \mathbb{E}_q \sum_n \log p(\boldsymbol{x}_n,\boldsymbol{z}_n|\boldsymbol{\theta})$$

E and M Steps

Our simplified expression

$$\begin{aligned}\ell(\boldsymbol{\theta}^t) &= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t)}{p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)} \\ &= \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}^t) - p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)\end{aligned}$$

E-Step: For all n, compute $q(\boldsymbol{z}_n) = p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t)$

We can view it as computing the *expected (complete) log-likelihood*:

$$Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t) = \sum_n \sum_{\boldsymbol{z}_n} p(\boldsymbol{z}_n|\boldsymbol{x}_n;\boldsymbol{\theta}^t) \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta}) = \mathbb{E}_q \sum_n \log p(\boldsymbol{x}_n, \boldsymbol{z}_n|\boldsymbol{\theta})$$

M-Step: Maximize $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$, i.e., $\boldsymbol{\theta}^{t+1} = \arg\max_{\boldsymbol{\theta}} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^t)$

Example: applying EM to GMMs

What is the E-step in GMM?

$$\gamma_{nk} = p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^{(t)})$$

What is the M-step in GMM? The Q-function is

$$\begin{aligned} Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \sum_n \sum_k p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^{(t)}) \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\ &= \sum_n \sum_k \gamma_{nk} \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \end{aligned}$$

Example: applying EM to GMMs

What is the E-step in GMM?

$$\gamma_{nk} = p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^{(t)})$$

What is the M-step in GMM? The Q-function is

$$\begin{aligned}
Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) &= \sum_n \sum_k p(z = k|\boldsymbol{x}_n; \boldsymbol{\theta}^{(t)}) \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\
&= \sum_n \sum_k \gamma_{nk} \log p(\boldsymbol{x}_n, z = k|\boldsymbol{\theta}) \\
&= \sum_k \sum_n \gamma_{nk} \log p(z = k)p(\boldsymbol{x}_n|z = k) \\
&= \sum_k \sum_n \gamma_{nk} \left[\log \omega_k + \log N(\boldsymbol{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)\right]
\end{aligned}$$

We have recovered the parameter estimation problem for GMMs that we previously discussed

Iterative and monotonic improvement

- We can show that $\ell(\boldsymbol{\theta}^{t+1}) \geq \ell(\boldsymbol{\theta}^{t})$
- Recall that we chose $q(\cdot)$ in the E-step such that:

$$\ell(\boldsymbol{\theta}^{t}) = \sum_{n} \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^{t})}{q(\boldsymbol{z}_n)}$$

Iterative and monotonic improvement

- We can show that $\ell(\boldsymbol{\theta}^{t+1}) \geq \ell(\boldsymbol{\theta}^{t})$
- Recall that we chose $q(\cdot)$ in the E-step such that:

$$\ell(\boldsymbol{\theta}^{t}) = \sum_{n} \sum_{\boldsymbol{z}_n} q(\boldsymbol{z}_n) \log \frac{p(\boldsymbol{x}_n, \boldsymbol{z}_n | \boldsymbol{\theta}^{t})}{q(\boldsymbol{z}_n)}$$

- However, in the M-step, $\boldsymbol{\theta}^{t+1}$ is chosen to maximize the right hand side of the equation with respect to $\boldsymbol{\theta}$, thus proving our desired result
- Note: the EM procedure converges but only to a local optimum

Outline

1. Administration
2. Review of last lecture
3. Course Evaluation
4. PCA

Instructions for filling out online course evaluation

- You should have recieved an email with a direct link
- You can also access the evaluation from MyUCLA
- `www.oid.ucla.edu/assessment/eip/onlineeval/studentaccess`

Outline

1 Administration

2 Review of last lecture

3 Course Evaluation

4 PCA
- PCA Basics
- PCA Algorithm / Derivation

Raw data can be Complex, High-dimensional

To understand a phenomenon we measure various related quantities

If we knew what to measure or how to represent our measurements we might find simple relationships

Raw data can be Complex, High-dimensional

To understand a phenomenon we measure various related quantities

If we knew what to measure or how to represent our measurements we might find simple relationships

But in practice we often *measure redundant signals*, e.g., US and European shoe sizes

Raw data can be Complex, High-dimensional

To understand a phenomenon we measure various related quantities

If we knew what to measure or how to represent our measurements we might find simple relationships

But in practice we often *measure redundant signals*, e.g., US and European shoe sizes

We also *represent data via the method by which it was gathered*, e.g., pixel representation of brain imaging data

Dimensionality Reduction

Issues

- *Measure redundant signals*
- *Represent data via the method by which it was gathered*

Goal: Find a ‘better’ representation for data

- To visualize and discover hidden patterns
- Preprocessing for supervised task

Dimensionality Reduction

Issues

- *Measure redundant signals*
- *Represent data via the method by which it was gathered*

Goal: Find a ‘better’ representation for data

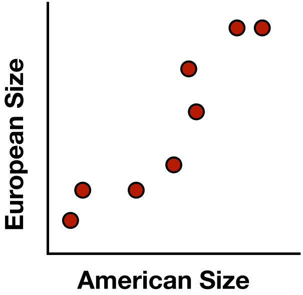
- To visualize and discover hidden patterns
- Preprocessing for supervised task

How do we define ‘better’?

E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation



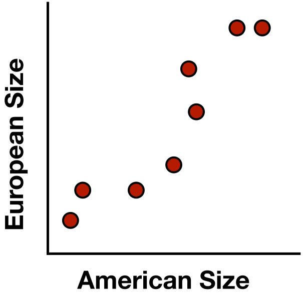
E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation

How can we do ‘better’, i.e., find a simpler, compact representation?

- Pick a direction and project onto this direction



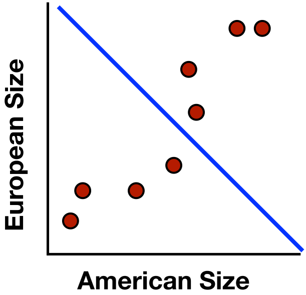
E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation

How can we do ‘better’, i.e., find a simpler, compact representation?

- Pick a direction and project onto this direction



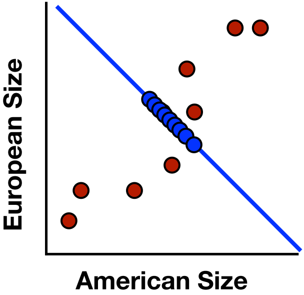
E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation

How can we do ‘better’, i.e., find a simpler, compact representation?

- Pick a direction and project onto this direction



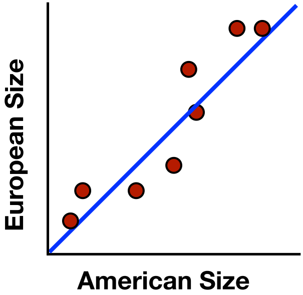
E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation

How can we do 'better', i.e., find a simpler, compact representation?

- Pick a direction and project onto this direction



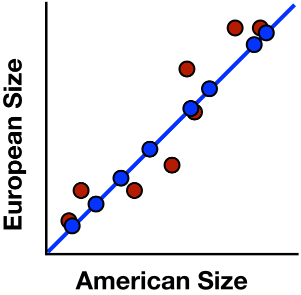
E.g., Shoe Size

We take noisy measurements on European and American scale

- Modulo noise, we expect perfect correlation

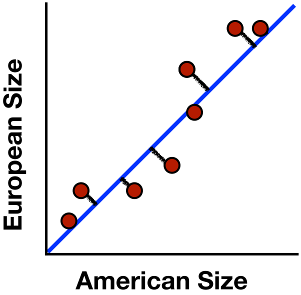
How can we do ‘better’, i.e., find a simpler, compact representation?

- Pick a direction and project onto this direction



Goal: Minimize Reconstruction Error

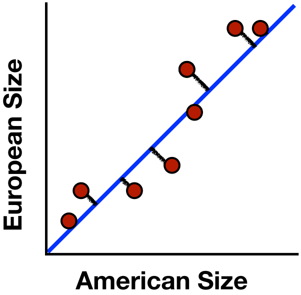
Minimize Euclidean distances between original points and their projections



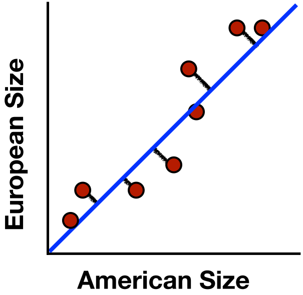
Goal: Minimize Reconstruction Error

Minimize Euclidean distances between original points and their projections

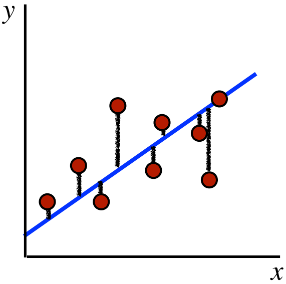
PCA solution solves this problem!



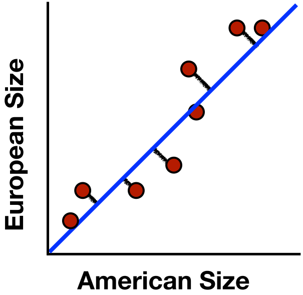
PCA — reconstruct 2D data via 2D data with single degree of freedom. Evaluate reconstructions (represented by blue line) by **Euclidean** distances



Linear Regression — predict y from x. Evaluate accuracy of predictions (represented by blue line) by **vertical** distances between points and the line

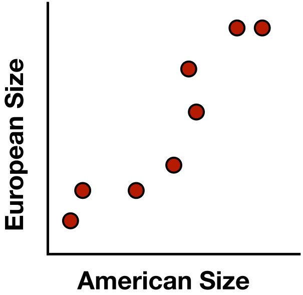


PCA — reconstruct 2D data via 2D data with single degree of freedom. Evaluate reconstructions (represented by blue line) by **Euclidean** distances



Another Goal: Maximize Variance

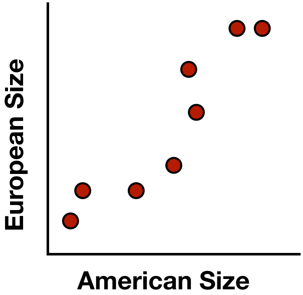
To identify patterns we want to study variation across observations



Another Goal: Maximize Variance

To identify patterns we want to study variation across observations

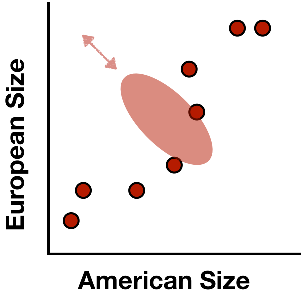
Can we do 'better', i.e., find a compact representation that captures variation?



Another Goal: Maximize Variance

To identify patterns we want to study variation across observations

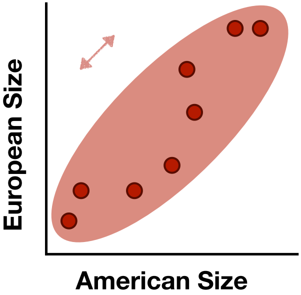
Can we do ‘better’, i.e., find a compact representation that captures variation?



Another Goal: Maximize Variance

To identify patterns we want to study variation across observations

Can we do ‘better’, i.e., find a compact representation that captures variation?

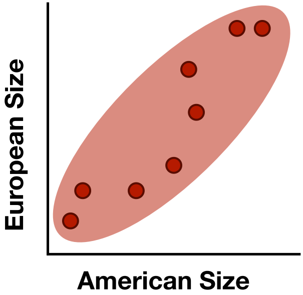


Another Goal: Maximize Variance

To identify patterns we want to study variation across observations

Can we do ‘better’, i.e., find a compact representation that captures variation?

PCA solution finds directions of maximal variance!



PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)

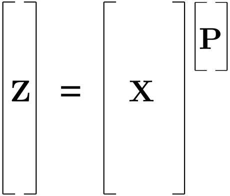
$$\begin{bmatrix} \\ \mathbf{Z} \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \quad \mathbf{X} \quad \\ \\ \end{bmatrix} \begin{bmatrix} \mathbf{P} \end{bmatrix}$$

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{XP}$ is $n \times k$ (reduced representation, PCA ‘scores’)
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)

Linearity assumption ($\mathbf{Z} = \mathbf{XP}$) simplifies problem

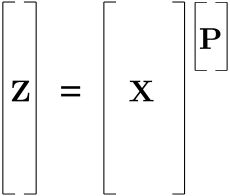


PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{XP}$ is $n \times k$ (reduced representation, PCA ‘scores’)
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance constraints

Linearity assumption ($\mathbf{Z} = \mathbf{XP}$) simplifies problem



Given n training points with d features:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix storing points
- $x_j^{(i)}$: jth feature for ith point
- μ_j : mean of jth feature

Given n training points with d features:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix storing points
- $x_j^{(i)}$: jth feature for ith point
- μ_j : mean of jth feature

Variance of 1st feature

$$\sigma_1^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x_1^{(i)} - \mu_1 \right)^2$$

Variance of 1st feature (assuming zero mean)

$$\sigma_1^2 = \frac{1}{n} \sum_{i=1}^{n} \left(x_1^{(i)} \right)^2$$

Given n training points with d features:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix storing points
- $x_j^{(i)}$: jth feature for ith point
- μ_j : mean of jth feature

Covariance of 1st and 2nd features (assuming zero mean) $\sigma_{12} = \frac{1}{n} \sum_{i=1}^{n} x_1^{(i)} x_2^{(i)}$

- Symmetric: $\sigma_{12} = \sigma_{21}$
- Zero → uncorrelated
- Large magnitude → (anti) correlated / redundant
- $\sigma_{12} = \sigma_1^2 = \sigma_2^2$ → features are the same

Covariance Matrix

Covariance matrix generalizes this idea for many features

$d \times d$ covariance matrix with zero mean features

$$\mathbf{C_X} = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$$

- ith diagonal entry equals variance of ith feature
- ijth entry is covariance between ith and jth features
- Symmetric (makes sense given definition of covariance)

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{X}\mathbf{P}$ is $n \times k$ (reduced representation, PCA ‘scores’)
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance / Covariance constraints

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{X}\mathbf{P}$ is $n \times k$ (reduced representation, PCA 'scores')
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance / Covariance constraints

What constraints make sense in reduced representation?

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{XP}$ is $n \times k$ (reduced representation, PCA 'scores')
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance / Covariance constraints

What constraints make sense in reduced representation?

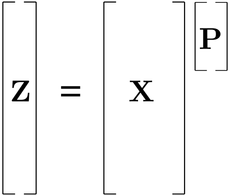
- No feature correlation, i.e., all off-diagonals in $\mathbf{C_Z}$ are zero
- Rank-ordered features by variance, i.e., sorted diagonals of $\mathbf{C_Z}$

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{XP}$ is $n \times k$ (reduced representation, PCA 'scores')
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance / Covariance constraints

$\mathbf{P}$ equals the top k eigenvectors of $\mathbf{C_X}$



PCA Solution

All covariance matrices have an eigendecomposition

- $\mathbf{C}_{\mathbf{X}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$ (eigendecomposition)
- $\mathbf{U}$ is $d \times d$ (column are eigenvectors, sorted by their eigenvalues)
- $\mathbf{\Lambda}$ is $d \times d$ (diagonals are eigenvalues, off-diagonals are zero)

PCA Solution

All covariance matrices have an eigendecomposition

- $\mathbf{C}_{\mathbf{X}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^{\top}$ (eigendecomposition)
- $\mathbf{U}$ is $d \times d$ (column are eigenvectors, sorted by their eigenvalues)
- $\boldsymbol{\Lambda}$ is $d \times d$ (diagonals are eigenvalues, off-diagonals are zero)

The d eigenvectors are orthonormal directions of max variance

- Associated eigenvalues equal variance in these directions
- 1st eigenvector is direction of max variance (variance is λ_1)

Choosing k

How should we pick the dimension of the new representation?

Choosing k

How should we pick the dimension of the new representation?

Visualization: Pick top 2 or 3 dimensions for plotting purposes

Choosing k

How should we pick the dimension of the new representation?

Visualization: Pick top 2 or 3 dimensions for plotting purposes

Other analyses: Capture 'most' of the variance in the data

- Recall that eigenvalues are variances in the directions specified by eigenvectors, and that eigenvalues are sorted
- Fraction of retained variance: $\frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$

Can choose k such that we retain some fraction of the variance, e.g., 95%

Other Practical Tips

PCA assumptions (linearity, orthogonality) not always appropriate

- Various extensions to PCA with different underlying assumptions, e.g., manifold learning, Kernel PCA, ICA

Other Practical Tips

PCA assumptions (linearity, orthogonality) not always appropriate

- Various extensions to PCA with different underlying assumptions, e.g., manifold learning, Kernel PCA, ICA

Centering is crucial, i.e., we must preprocess data so that all features have zero mean before applying PCA

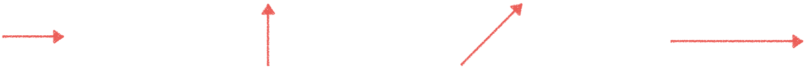
PCA results dependent on scaling of data

- Data is sometimes rescaled in practice before applying PCA

Orthogonal and Orthonormal Vectors

Orthogonal vectors are **perpendicular** to each other

- Equivalently, their dot product equals zero
- $\mathbf{a}^\top \mathbf{b} = 0$ and $\mathbf{d}^\top \mathbf{b} = 0$, but $\mathbf{c}$ isn't orthogonal to others



$\mathbf{a} = \begin{bmatrix}1 & 0\end{bmatrix}^\top$ $\quad$ $\mathbf{b} = \begin{bmatrix}0 & 1\end{bmatrix}^\top$ $\quad$ $\mathbf{c} = \begin{bmatrix}1 & 1\end{bmatrix}^\top$ $\quad$ $\mathbf{d} = \begin{bmatrix}2 & 0\end{bmatrix}^\top$

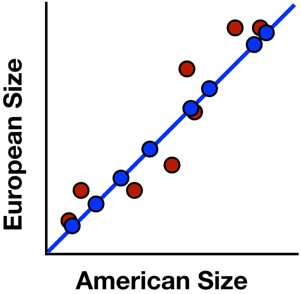
Orthonormal vectors are orthogonal and have unit norm

- $\mathbf{a}$ are $\mathbf{b}$ are orthonormal, but $\mathbf{b}$ are $\mathbf{d}$ are not orthonormal

PCA Iterative Algorithm

$k = 1$: Find direction of max variance, project onto this direction

- Locations along this direction are the new 1D representation



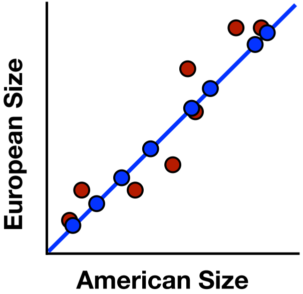
PCA Iterative Algorithm

$k = 1$: Find direction of max variance, project onto this direction

- Locations along this direction are the new 1D representation

More generally, for i in $\{1, \ldots, k\}$:

- Find direction of max variance that is *orthonormal* to previously selected directions, project onto this direction
- Locations along this direction are the ith feature in new representation



Eigendecomposition

All covariance matrices have an eigendecomposition

- $\mathbf{C_X} = \mathbf{U \Lambda U}^\top$ (eigendecomposition)
- $\mathbf{U}$ is $d \times d$ (column are eigenvectors, sorted by their eigenvalues)
- Λ is $d \times d$ (diagonals are eigenvalues, off-diagonals are zero)

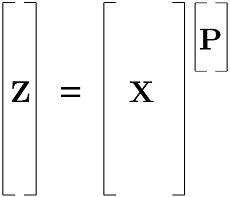
Eigenvector / Eigenvalue equation: $\mathbf{C_x u} = \lambda \mathbf{u}$

- By definition $\mathbf{u}^\top \mathbf{u} = 1$ (unit norm)
- Example: $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \implies$ eigenvector: $\mathbf{u} = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top$ eigenvalue: $\lambda = 1$

PCA Formulation

PCA: find lower-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{Z} = \mathbf{XP}$ is $n \times k$ (reduced representation, PCA 'scores')
- $\mathbf{P}$ is $d \times k$ (columns are k principal components)
- Variance / Covariance constraints



PCA Formulation, $k = 1$

PCA: find one-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{z} = \mathbf{X}\mathbf{p}$ is $n \times 1$ (reduced representation, PCA 'scores')
- $\mathbf{p}$ is $d \times 1$ (columns are k principal components)
- Variance constraint

$$\sigma_{\mathbf{z}}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(z^{(i)} \right)^2 = ||\mathbf{z}||_2^2$$

PCA Formulation, $k = 1$

PCA: find one-dimensional representation of raw data

- $\mathbf{X}$ is $n \times d$ (raw data)
- $\mathbf{z} = \mathbf{X}\mathbf{p}$ is $n \times 1$ (reduced representation, PCA 'scores')
- $\mathbf{p}$ is $d \times 1$ (columns are k principal components)
- Variance constraint

$$\sigma_{\mathbf{z}}^2 = \frac{1}{n} \sum_{i=1}^{n} \left(z^{(i)} \right)^2 = ||\mathbf{z}||_2^2$$

Goal: Maximizes variance, i.e., $\max_{\mathbf{p}} \sigma_{\mathbf{z}}^2$ where $||\mathbf{p}||_2 = 1$

Goal: Maximizes variance, i.e., $\max_{\mathbf{p}} \sigma_{\mathbf{z}}^2$ where $||\mathbf{p}||_2 = 1$

$$\sigma_{\mathbf{z}}^2 = \frac{1}{n}||\mathbf{z}||_2^2$$

Goal: Maximizes variance, i.e., $\max\limits_{\mathbf{p}} \sigma_{\mathbf{z}}^2$ where $||\mathbf{p}||_2 = 1$

Relationship between Euclidean distance and dot product

$$\begin{aligned}\sigma_{\mathbf{z}}^2 &= \frac{1}{n}||\mathbf{z}||_2^2 \\ &= \frac{1}{n}\mathbf{z}^\top \mathbf{z}\end{aligned}$$

Goal: Maximizes variance, i.e., $\max_{\mathbf{p}} \sigma_{\mathbf{z}}^2$ where $||\mathbf{p}||_2 = 1$

$$\sigma_{\mathbf{z}}^2 = \frac{1}{n}||\mathbf{z}||_2^2$$

Relationship between Euclidean distance and dot product

$$= \frac{1}{n}\mathbf{z}^\top \mathbf{z}$$

Definition: $\mathbf{z} = \mathbf{X}\mathbf{p}$

$$= \frac{1}{n}(\mathbf{X}\mathbf{p})^\top (\mathbf{X}\mathbf{p})$$

Goal: Maximizes variance, i.e., $\max\limits_{\mathbf{p}} \sigma_{\mathbf{z}}^2$ where $||\mathbf{p}||_2 = 1$

$$\sigma_{\mathbf{z}}^2 = \frac{1}{n}||\mathbf{z}||_2^2$$

Relationship between Euclidean distance and dot product

$$= \frac{1}{n}\mathbf{z}^\top\mathbf{z}$$

Definition: $\mathbf{z} = \mathbf{X}\mathbf{p}$

$$= \frac{1}{n}(\mathbf{X}\mathbf{p})^\top(\mathbf{X}\mathbf{p})$$

Transpose property: $(\mathbf{X}\mathbf{p})^\top = \mathbf{p}^\top\mathbf{X}^\top$; associativity of multiply

$$= \frac{1}{n}\mathbf{p}^\top\mathbf{X}^\top\mathbf{X}\mathbf{p}$$

Goal: Maximizes variance, i.e., $\max_{\mathbf{p}} \sigma_{\mathbf{z}}^2$ where $||\mathbf{p}||_2 = 1$

$$\sigma_{\mathbf{z}}^2 = \frac{1}{n}||\mathbf{z}||_2^2$$

Relationship between Euclidean distance and dot product $$= \frac{1}{n}\mathbf{z}^\top\mathbf{z}$$

Definition: $\mathbf{z} = \mathbf{X}\mathbf{p}$ $$= \frac{1}{n}(\mathbf{X}\mathbf{p})^\top(\mathbf{X}\mathbf{p})$$

Transpose property: $(\mathbf{X}\mathbf{p})^\top = \mathbf{p}^\top\mathbf{X}^\top$; associativity of multiply $$= \frac{1}{n}\mathbf{p}^\top\mathbf{X}^\top\mathbf{X}\mathbf{p}$$

Definition: $\mathbf{C}_{\mathbf{X}} = \frac{1}{n}\mathbf{X}^\top\mathbf{X}$ $$= \mathbf{p}^\top\mathbf{C}_{\mathbf{X}}\mathbf{p}$$

Goal: Maximizes variance, i.e., $\max_{\mathbf{p}} \sigma_{\mathbf{z}}^2$ where $||\mathbf{p}||_2 = 1$

	$\sigma_{\mathbf{z}}^2 = \frac{1}{n}\|\|\mathbf{z}\|\|_2^2$
Relationship between Euclidean distance and dot product	$= \frac{1}{n}\mathbf{z}^\top \mathbf{z}$
Definition: $\mathbf{z} = \mathbf{X}\mathbf{p}$	$= \frac{1}{n}(\mathbf{X}\mathbf{p})^\top (\mathbf{X}\mathbf{p})$
Transpose property: $(\mathbf{X}\mathbf{p})^\top = \mathbf{p}^\top \mathbf{X}^\top$; associativity of multiply	$= \frac{1}{n}\mathbf{p}^\top \mathbf{X}^\top \mathbf{X}\mathbf{p}$
Definition: $\mathbf{C}_{\mathbf{X}} = \frac{1}{n}\mathbf{X}^\top \mathbf{X}$	$= \mathbf{p}^\top \mathbf{C}_{\mathbf{X}} \mathbf{p}$

Restated Goal: $\max_{\mathbf{p}} \mathbf{p}^\top \mathbf{C}_{\mathbf{x}} \mathbf{p}$ where $||\mathbf{p}||_2 = 1$

Connection to Eigenvectors

Recall eigenvector / eigenvalue equation: $\mathbf{C}_\mathbf{x}\mathbf{u} = \lambda\mathbf{u}$

- By definition $\mathbf{u}^\top\mathbf{u} = 1$, and thus $\mathbf{u}^\top\mathbf{C}_\mathbf{x}\mathbf{u} = \lambda$

Restated Goal: $\max_{\mathbf{p}} \mathbf{p}^\top\mathbf{C}_\mathbf{x}\mathbf{p}$ where $||\mathbf{p}||_2 = 1$

Connection to Eigenvectors

Recall eigenvector / eigenvalue equation: $\mathbf{C}_{\mathbf{x}}\mathbf{u} = \lambda\mathbf{u}$

- By definition $\mathbf{u}^\top\mathbf{u} = 1$, and thus $\mathbf{u}^\top\mathbf{C}_{\mathbf{x}}\mathbf{u} = \lambda$
- But this is the expression we're optimizing, and thus maximal variance achieved when $\mathbf{p}$ is top eigenvector of $\mathbf{C}_{\mathbf{X}}$

Restated Goal: $\max\limits_{\mathbf{p}} \mathbf{p}^\top\mathbf{C}_{\mathbf{x}}\mathbf{p}$ where $||\mathbf{p}||_2 = 1$

Connection to Eigenvectors

Recall eigenvector / eigenvalue equation: $\mathbf{C}_{\mathbf{x}}\mathbf{u} = \lambda\mathbf{u}$

- By definition $\mathbf{u}^\top\mathbf{u} = 1$, and thus $\mathbf{u}^\top\mathbf{C}_{\mathbf{x}}\mathbf{u} = \lambda$
- But this is the expression we're optimizing, and thus maximal variance achieved when $\mathbf{p}$ is top eigenvector of $\mathbf{C}_{\mathbf{X}}$

Similar arguments can be used for $k > 1$

Restated Goal: $\max_{\mathbf{p}} \mathbf{p}^\top\mathbf{C}_{\mathbf{x}}\mathbf{p}$ where $||\mathbf{p}||_2 = 1$