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 $x$

$$p(x) = \sum_{k=1}^{K} \omega_k N(x | \mu_k, \Sigma_k)$$

- $K$: the number of Gaussians — they are called (mixture) components
- $\mu_k$ and $\Sigma_k$: mean and covariance matrix of the $k$-th component
- $\omega_k$: mixture weights – priors on each component that satisfy:
  $$\forall k, \omega_k > 0, \text{ and } \sum_{k} \omega_k = 1$$

- Given unlabeled data, $\mathcal{D} = \{x_n\}_{n=1}^{N}$, we must learn:
  - parameters of Gaussians
  - mixture components
Parameter estimation for GMMs: complete data

GMM Parameters

\[ \theta = \{\omega_k, \mu_k, \Sigma_k\}_{k=1}^K \]

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

\[ D' = \{x_n, z_n\}_{n=1}^N \]

MLE: Maximize the complete likelihood

\[ \theta = \arg \max \log D' = \sum_n \log p(x_n, z_n) \]

Easy problem to solve – intuitive, closed form solution exists!
Parameter estimation for GMMs: Incomplete data

GMM Parameters

\[ \theta = \{ \omega_k, \mu_k, \Sigma_k \}_{k=1}^K \]

Incomplete Data

Our data contains observed and unobserved data, and hence is incomplete
Parameter estimation for GMMs: Incomplete data

**GMM Parameters**

\[ \theta = \{ \omega_k, \mu_k, \Sigma_k \}_{k=1}^K \]

**Incomplete Data**

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

- Observed: \( \mathcal{D} = \{ x_n \} \)
- Unobserved (hidden): \( \{ z_n \} \)

**Goal** Obtain the maximum likelihood estimate of \( \theta \):

\[
\theta = \arg \max \ell(\theta) = \arg \max \sum_n \log p(x_n | \theta)
\]

\[
= \arg \max \sum_n \log \sum_{z_n} p(x_n, z_n | \theta)
\]

The objective function \( \ell(\theta) \) 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 $\theta$
- M-step: solve for new values of $\theta$ given imputed values for $z_n$ (maximize complete likelihood!)
E-step: Soft cluster assignments

We define $\gamma_{nk}$ as $p(z_n = k | x_n, \theta)$

- This is the posterior distribution of $z_n$ given $x_n$ and $\theta$
- Recall that in complete data setting $\gamma_{nk}$ was binary
- Now it’s a “soft” assignment of $x_n$ to $k$-th component, with $x_n$ assigned to each component with some probability

Given an estimate of $\theta = \{\omega_k, \mu_k, \Sigma_k\}_{k=1}^K$, we can compute $\gamma_{nk}$ as follows:

$$
\gamma_{nk} = p(z_n = k | x_n) = \frac{p(x_n | z_n = k) p(z_n = k)}{p(x_n)} = \frac{p(x_n | z_n = k) p(z_n = k)}{\sum_{k'=1}^K p(x_n | z_n = k') p(z_n = k')} 
$$
M-step: Maximize complete likelihood

Recall definition of complete likelihood from earlier:

\[
\sum_n \log p(x_n, z_n) = \sum_k \sum_n \gamma_{nk} \log \omega_k + \sum_k \left\{ \sum_n \gamma_{nk} \log N(x_n | \mu_k, \Sigma_k) \right\}
\]

Previously \( \gamma_{nk} \) was binary, but now we define \( \gamma_{nk} = p(z_n = k | x_n) \) (E-step)
M-step: Maximize complete likelihood

Recall definition of complete likelihood from earlier:

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

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

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

\[ \omega_k = \frac{\sum_n \gamma_{nk}}{\sum_k \sum_n \gamma_{nk}}, \quad \boldsymbol{\mu}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} \mathbf{x}_n \]
\[ \boldsymbol{\Sigma}_k = \frac{1}{\sum_n \gamma_{nk}} \sum_n \gamma_{nk} (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T \]

Intuition: Each point now contributes some fractional component to each of the parameters, with weights determined by \( \gamma_{nk} \)
EM procedure for GMM

**Alternate between estimating $\gamma_{nk}$ and estimating $\theta$**

- Initialize $\theta$ with some values (random or otherwise)
- Repeat
  - E-Step: Compute $\gamma_{nk}$ using the current $\theta$
  - M-Step: Update $\theta$ using the $\gamma_{nk}$ 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?
EM procedure for GMM

Alternate between estimating $\gamma_{nk}$ and estimating $\theta$

- Initialize $\theta$ with some values (random or otherwise)
- Repeat
  - E-Step: Compute $\gamma_{nk}$ using the current $\theta$
  - M-Step: Update $\theta$ using the $\gamma_{nk}$ 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 $\sigma^2 I$ covariance matrices
- Further assume $\sigma \to 0$, so we only need to estimate $\mu_k$, i.e., means

K-means is often called “hard” GMM or GMMs is called “soft” K-means

The posterior $\gamma_{nk}$ provides a probabilistic assignment for $x_n$ 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

\[ p(x|\theta) = \sum_z p(x, z|\theta) \]

- Given incomplete data \( D = \{x_n\} \) our goal is maximize incomplete log likelihood, \( \ell(\theta) = \sum_n \log \sum_{z_n} p(x_n, z_n|\theta) \)
- Log-sum form of incomplete log-likelihood is difficult to work with
- EM: construct lower bound on \( \ell(\theta) \) (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 $q(z)$ over $z$:

$$\ell(\theta) \geq \sum_n \sum_{z_n} q(z_n) \log \frac{p(x_n, z_n | \theta)}{q(z_n)}$$
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 $q(z)$ over $z$:

\[
\ell(\theta) \geq \sum_n \sum_{z_n} q(z_n) \log \frac{p(x_n, z_n|\theta)}{q(z_n)}
\]

- We want a *tight* lower bound, and given some current estimate $\theta^t$, we will pick $q(\cdot)$ such that our lower bound holds *with equality* at $\theta^t$
- 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 $q(z)$ over $z$:

$$
\ell(\theta) \geq \sum_n \sum_{z_n} q(z_n) \log \frac{p(x_n, z_n|\theta)}{q(z_n)}
$$

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

Our simplified expression

\[
\ell(\theta^t) = \sum_n \sum_{z_n} p(z_n|\mathbf{x}_n; \theta^t) \log \frac{p(\mathbf{x}_n, z_n|\theta^t)}{p(z_n|\mathbf{x}_n; \theta^t)}
\]

\[
= \sum_n \sum_{z_n} p(z_n|\mathbf{x}_n; \theta^t) \log p(\mathbf{x}_n, z_n|\theta^t) - p(z_n|\mathbf{x}_n; \theta^t) \log p(z_n|\mathbf{x}_n; \theta^t)
\]

**E-Step**: For all \( n \), compute \( q(z_n) = p(z_n|\mathbf{x}_n; \theta^t) \)
E and M Steps

Our simplified expression

\[
\ell(\theta^t) = \sum_n \sum_{z_n} p(z_n | x_n; \theta^t) \log \frac{p(x_n, z_n | \theta^t)}{p(z_n | x_n; \theta^t)}
\]

\[
= \sum_n \sum_{z_n} p(z_n | x_n; \theta^t) \log p(x_n, z_n | \theta^t) - p(z_n | x_n; \theta^t) \log p(z_n | x_n; \theta^t)
\]

E-Step: For all \( n \), compute \( q(z_n) = p(z_n | x_n; \theta^t) \)

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

\[
Q(\theta | \theta^t) = \sum_n \sum_{z_n} p(z_n | x_n; \theta^t) \log p(x_n, z_n | \theta) = \mathbb{E}_q \sum_n \log p(x_n, z_n | \theta)
\]
E and M Steps

Our simplified expression

\[ \ell(\theta^t) = \sum_n \sum_{z_n} p(z_n|x_n; \theta^t) \log \frac{p(x_n, z_n|\theta^t)}{p(z_n|x_n; \theta^t)} = \sum_n \sum_{z_n} p(z_n|x_n; \theta^t) \log p(x_n, z_n|\theta^t) - p(z_n|x_n; \theta^t) \log p(z_n|x_n; \theta^t) \]

E-Step: For all \( n \), compute \( q(z_n) = p(z_n|x_n; \theta^t) \)

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

\[ Q(\theta|\theta^t) = \sum_n \sum_{z_n} p(z_n|x_n; \theta^t) \log p(x_n, z_n|\theta) = \mathbb{E}_q \sum_n \log p(x_n, z_n|\theta) \]

M-Step: Maximize \( Q(\theta|\theta^t) \), i.e., \( \theta^{t+1} = \arg \max_\theta Q(\theta|\theta^t) \)
Example: applying EM to GMMs

What is the E-step in GMM?

\[ \gamma_{nk} = p(z = k | x_n; \theta^{(t)}) \]

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

\[
Q(\theta | \theta^{(t)}) = \sum_n \sum_k p(z = k | x_n; \theta^{(t)}) \log p(x_n, z = k | \theta) \\
= \sum_n \sum_k \gamma_{nk} \log p(x_n, z = k | \theta)
\]
Example: applying EM to GMMs

What is the E-step in GMM?

\[ \gamma_{nk} = p(z = k | x_n; \theta^{(t)}) \]

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

\[
Q(\theta | \theta^{(t)}) = \sum_n \sum_k p(z = k | x_n; \theta^{(t)}) \log p(x_n, z = k | \theta)
\]

\[
= \sum_n \sum_k \gamma_{nk} \log p(x_n, z = k | \theta)
\]

\[
= \sum_k \sum_n \gamma_{nk} \log p(z = k) p(x_n | z = k)
\]

\[
= \sum_k \sum_n \gamma_{nk} [\log \omega_k + \log N(x_n | \mu_k, \Sigma_k)]
\]

We have recovered the parameter estimation problem for GMMs that we previously discussed.
Iterative and monotonic improvement

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

$$
\ell(\theta^t) = \sum_n \sum_{z_n} q(z_n) \log \frac{p(x_n, z_n|\theta^t)}{q(z_n)}
$$
Iterative and monotonic improvement

- We can show that $\ell(\theta^{t+1}) \geq \ell(\theta^t)$
- Recall that we chose $q(\cdot)$ in the E-step such that:
  $$\ell(\theta^t) = \sum_n \sum_{z_n} q(z_n) \log \frac{p(x_n, z_n | \theta^t)}{q(z_n)}$$
- However, in the M-step, $\theta^{t+1}$ is chosen to maximize the right hand side of the equation with respect to $\theta$, 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 received 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
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
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

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

\[ Z = X \begin{bmatrix} P \end{bmatrix} \]
PCA Formulation

PCA: find lower-dimensional representation of raw data

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

Linearity assumption ($Z = XP$) 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

\[
\begin{bmatrix}
\mathbf{Z}
\end{bmatrix}
= \begin{bmatrix}
\mathbf{X}
\end{bmatrix}
\begin{bmatrix}
\mathbf{P}
\end{bmatrix}
\]
Given \( n \) training points with \( d \) features:

- \( \mathbf{X} \in \mathbb{R}^{n \times d} \): matrix storing points
- \( x_{j}^{(i)} \): \( j \)th feature for \( i \)th point
- \( \mu_{j} \): mean of \( j \)th feature
Given $n$ training points with $d$ features:

- $\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix storing points
- $x^{(i)}_j$: $j$th feature for $i$th point
- $\mu_j$: mean of $j$th feature

Variance of 1st feature (assuming zero mean)

$$\sigma_1^2 = \frac{1}{n} \sum_{i=1}^{n} \left( x^{(i)}_1 - \mu_1 \right)^2$$

Variance of 1st feature

$$\sigma_1^2 = \frac{1}{n} \sum_{i=1}^{n} \left( x^{(i)}_1 \right)^2$$
Given $n$ training points with $d$ features:
- $\mathbf{X} \in \mathbb{R}^{n \times d}$: matrix storing points
- $x_{j}^{(i)}$: $j$th feature for $i$th point
- $\mu_{j}$: mean of $j$th 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 $\rightarrow$ uncorrelated
- Large magnitude $\rightarrow$ (anti) correlated / redundant
- $\sigma_{12} = \sigma_{1}^{2} = \sigma_{2}^{2} \rightarrow$ features are the same
Covariance Matrix

Covariance matrix generalizes this idea for many features

- $i$th diagonal entry equals variance of $i$th feature
- $ij$th entry is covariance between $i$th and $j$th features
- Symmetric (makes sense given definition of covariance)

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

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

PCA: find lower-dimensional representation of raw data

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

PCA: find lower-dimensional representation of raw data

- \( X \) is \( n \times d \) (raw data)
- \( Z = XP \) is \( n \times k \) (reduced representation, PCA ‘scores’)
- \( P \) 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
- \( X \) is \( n \times d \) (raw data)
- \( Z = XP \) is \( n \times k \) (reduced representation, PCA ‘scores’)
- \( P \) 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 \( C_Z \) are zero
- Rank-ordered features by variance, i.e., sorted diagonals of \( C_Z \)
PCA Formulation

PCA: find lower-dimensional representation of raw data
• $X$ is $n \times d$ (raw data)
• $Z = XP$ is $n \times k$ (reduced representation, PCA ‘scores’)
• $P$ is $d \times k$ (columns are $k$ principal components)
• Variance / Covariance constraints

$P$ equals the top $k$ eigenvectors of $C_X$

$$Z = X P$$
PCA Solution

All covariance matrices have an eigendecomposition

- $C_x = U \Lambda U^T$ (eigendecomposition)
- $U$ is $d \times d$ (column are eigenvectors, sorted by their eigenvalues)
- $\Lambda$ is $d \times d$ (diagonals are eigenvalues, off-diagonals are zero)
PCA Solution

All covariance matrices have an eigendecomposition

- $C_X = U \Lambda U^T$ (eigendecomposition)
- $U$ is $d \times d$ (column are eigenvectors, sorted by their eigenvalues)
- $\Lambda$ 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 $\lambda_1$)
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

\[
\begin{align*}
\mathbf{a} &= \begin{bmatrix} 1 & 0 \end{bmatrix}^\top \\
\mathbf{b} &= \begin{bmatrix} 0 & 1 \end{bmatrix}^\top \\
\mathbf{c} &= \begin{bmatrix} 1 & 1 \end{bmatrix}^\top \\
\mathbf{d} &= \begin{bmatrix} 2 & 0 \end{bmatrix}^\top 
\end{align*}
\]

Orthonormal vectors are orthogonal and have unit norm
- \( \mathbf{a} \) and \( \mathbf{b} \) are orthonormal, but \( \mathbf{b} \) and \( \mathbf{d} \) 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

![Graph showing the relationship between American and European sizes with a line representing the direction of max variance.](image)
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 $i$th feature in new representation
Eigendecomposition

All covariance matrices have an eigendecomposition
- $C_x = U \Lambda U^T$ (eigendecomposition)
- $U$ is $d \times d$ (column are eigenvectors, sorted by their eigenvalues)
- $\Lambda$ is $d \times d$ (diagonals are eigenvalues, off-diagonals are zero)

Eigenvector / Eigenvalue equation: $C_x u = \lambda u$
- By definition $u^T u = 1$ (unit norm)

Example:
- $\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- eigenvector: $u = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$
- eigenvalue: $\lambda = 1$
PCA Formulation

PCA: find lower-dimensional representation of raw data

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

\[
\begin{bmatrix}
Z \\
\end{bmatrix}
= 
\begin{bmatrix}
X \\
\end{bmatrix}
\begin{bmatrix}
P \\
\end{bmatrix}
\]
PCA Formulation, $k = 1$

PCA: find **one-dimensional** representation of raw data

- **$X$** is $n \times d$ (raw data)
- **$z = Xp$** is $n \times 1$ (reduced representation, PCA ‘scores’)
- **$p$** is $d \times 1$ (columns are $k$ principal components)
- **Variance constraint**

\[
\sigma_z^2 = \frac{1}{n} \sum_{i=1}^{n} \left( z^{(i)} \right)^2 = ||z||_2^2
\]
PCA Formulation, \( k = 1 \)

PCA: find \textbf{one-dimensional} representation of raw data

- \( X \) is \( n \times d \) (raw data)
- \( z = Xp \) is \( n \times 1 \) (reduced representation, PCA ‘scores’)
- \( p \) is \( d \times 1 \) (columns are \( k \) principal components)
- Variance constraint

\[
\sigma_z^2 = \frac{1}{n} \sum_{i=1}^{n} (z^{(i)})^2 = \|z\|^2
\]

**Goal:** Maximizes variance, i.e., \( \max_p \sigma_z^2 \) where \( \|p\|_2 = 1 \)
Goal: Maximizes variance, i.e., $\max_{\mathbf{p}} \sigma^2_z$ where $||\mathbf{p}||_2 = 1$

$$\sigma^2_z = \frac{1}{n} ||\mathbf{z}||_2^2$$
Goal: Maximizes variance, i.e., \( \max_p \sigma_z^2 \) where \( \|p\|_2 = 1 \)

\[
\sigma_z^2 = \frac{1}{n} \|z\|_2^2 \\
= \frac{1}{n} z^\top z
\]

Relationship between Euclidean distance and dot product
Goal: Maximizes variance, i.e., \( \max_p \sigma_z^2 \) where \( \|p\|_2 = 1 \)

\[
\sigma_z^2 = \frac{1}{n} \|z\|_2^2 = \frac{1}{n} z^\top z = \frac{1}{n} (Xp)^\top (Xp)
\]

Relationship between Euclidean distance and dot product

Definition: \( z = Xp \)
Goal: Maximizes variance, i.e., $\max_p \sigma_z^2$ where $\|p\|_2 = 1$

Relationship between Euclidean distance and dot product

Definition: $z = Xp$

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

\[
\sigma_z^2 = \frac{1}{n} \|z\|_2^2
= \frac{1}{n} z^\top z
= \frac{1}{n} (Xp)^\top (Xp)
= \frac{1}{n} p^\top X^\top Xp
\]
Goal: Maximizes variance, i.e., \( \max_p \sigma^2_z \) where \( ||p||_2 = 1 \)

Relationship between Euclidean distance and dot product

Definition: \( z = Xp \)

Transpose property: \((Xp)^\top = p^\top X^\top\); associativity of multiply

Definition: \( C_X = \frac{1}{n} X^\top X \)
Goal: Maximizes variance, i.e., \( \max_{\mathbf{p}} \sigma^2_{\mathbf{z}} \) where \( \|\mathbf{p}\|_2 = 1 \)

\[
\sigma^2_{\mathbf{z}} = \frac{1}{n} \|\mathbf{z}\|_2^2 = \frac{1}{n} \mathbf{z}^\top \mathbf{z}
\]

Relation between Euclidean distance and dot product

Definition: \( \mathbf{z} = \mathbf{Xp} \)

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

Definition: \( \mathbf{C}_\mathbf{X} = \frac{1}{n} \mathbf{X}^\top \mathbf{X} \)

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: $C_x u = \lambda u$

- By definition $u^T u = 1$, and thus $u^T C_x u = \lambda$

Restated Goal: $\max_p p^T C_x p$ where $\|p\|_2 = 1$
Connection to Eigenvectors

Recall eigenvector / eigenvalue equation: $C_x u = \lambda u$

• By definition $u^T u = 1$, and thus $u^T C_x u = \lambda$

• But this is the expression we’re optimizing, and thus maximal variance achieved when $p$ is top eigenvector of $C_x$

Restated Goal: $\max_p p^T C_x p$ where $\|p\|_2 = 1$
Connection to Eigenvectors

Recall eigenvector / eigenvalue equation: $C_x u = \lambda u$

- By definition $u^\top u = 1$, and thus $u^\top C_x u = \lambda$
- But this is the expression we’re optimizing, and thus maximal variance achieved when $p$ is top eigenvector of $C_x$

Similar arguments can be used for $k > 1$

Restated Goal: $\max_p p^\top C_x p$ where $\|p\|_2 = 1$