ASSIGNMENT 2B: PRINCIPLE COMPONENT ANALYSIS

Institute for Machine Learning

JOHANNES KEPLER
UNIVERSITY LINZ

Institute for
Machine Learning

Contact

Heads:
Thomas Adler,
Philipp Renz,
Andreas Radler

Institute for Machine Learning
Johannes Kepler University
Altenberger Str. 69
A-4040 Linz

E-Mail: `{adler,renz,radler}@ml.jku.at`
Institute Homepage

Copyright statement:

Agenda

- PCA Intuition
- PCA Derivation

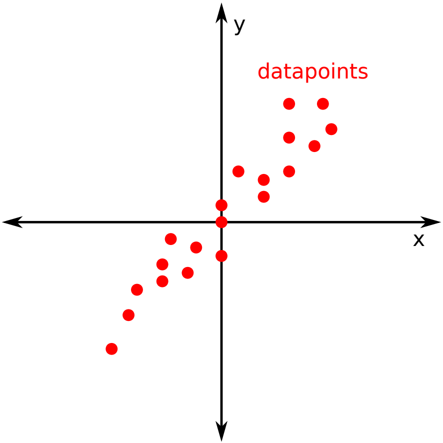
\- Lecture notes (Hochreiter, 2014)
\- Mathematics for Machine Learning (Deisenroth et al., 2018)

Principal Component Analysis (PCA)

- Idea: Reduce dimensionality of the dataset, while still preserving as much information as possible.
- Method: Use variance as measure of information.

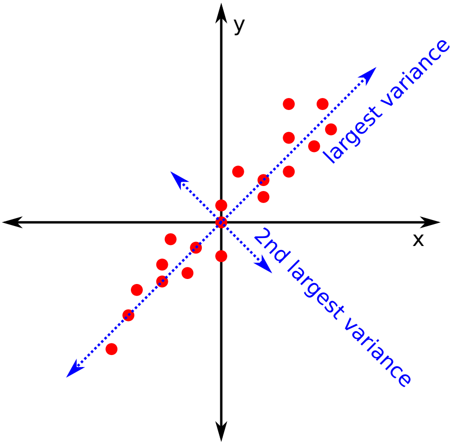
PCA – Intuition

- Starting point: we have some (centered) two dimensional data with coordinates $(x_1 = x, x_2 = y)$
- Note: usually we have high dimensional data; 2D for visualization



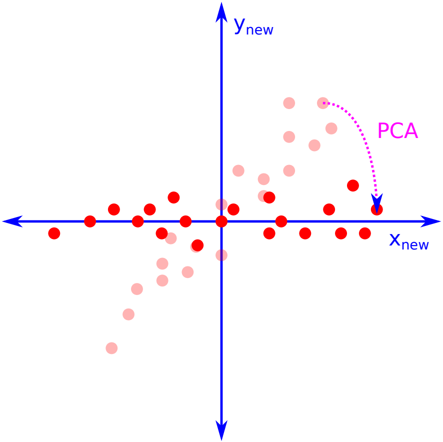
PCA – Intuition

- Sometimes data is correlated $\rightarrow$ find direction of largest variance (i.e. 1st principal component), then orthogonal direction of 2nd largest variance (i.e. 2nd principal component), and so on for higher dim.



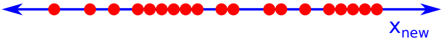
PCA – Intuition

- We can use these (orthogonal) directions of the largest variances as axes for a new coordinate system (= PCA):



PCA – Intuition

- Now we may omit axes with smaller variance, i.e. down-project our data. This can be useful for compression of our data for further processing or provide a means of visualization:



PCA – Derivation

- We use the **variance** as measure of information.
- Require that new dimensions are orthogonal to each other (so they are uncorrelated).
- Assume we are given n data points $\mathbf{x}_i = (x_i^{(1)}, ..., x_i^{(d)})$, $i = 1, ...n$, each of dimension d.
- Write data into in data matrix $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)^T \in \mathbb{R}^{n \times d}$.
- Rows of the data matrix $\mathbf{X}$ contain the observations; columns contain the features.
- If not already the case, shift data such that it is centered, i.e. has mean = 0 for every feature/dimension: $\frac{1}{n}\sum_{i=1}^{n} x_i^{(j)} = 0$ for every $j = 1, ..., d$.
- Compute the (symmetric and positive definite) sample covariance matrix: $\mathbf{C} = \mathrm{Covar}(\mathbf{X}) = \frac{1}{n}\mathbf{X}^T\mathbf{X} \in \mathbb{R}^{d \times d}$.

PCA – Derivation (2)

- Let's consider the first principal component[1]: vector $\mathbf{u}$ such that $\mathbf{X}\mathbf{u}$ retains as much variance as possible.
- Variance of projection:

$$\mathrm{Covar}(\mathbf{X}\,\mathbf{u}) = \mathbf{u}^T \mathbf{C}\,\mathbf{u}$$

- If we want $\max_{\mathbf{u}} \mathrm{Covar}(\mathbf{X}\,\mathbf{u})$, we have to constrain $\mathbf{u}$, otherwise trivial solution: $\mathbf{u} = \infty$
- Constraint: $\mathbf{u}$ must be a unit vector: $\|\mathbf{u}\| = 1$.

[1] We'll write $\mathbf{u}$ instead of $\mathbf{u}_1$ for now

PCA – Derivation (3)

- With Lagrange multipliers we can find the extrema of a function of several variables subject to one or more constraints.
- Given the following optimization problem:

$$\begin{aligned} \max_{x} \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \\ & h(x) = 0 \end{aligned}$$

- The Lagrangian is given by

$$\mathcal{L} = f(x) - \lambda\, g(x) - \phi\, h(x)$$

where λ and ϕ are Lagrange multipliers.

PCA – Derivation (4)

$$\max_{\mathbf{u}} \operatorname{Covar}(\mathbf{X}\,\mathbf{u})$$
$$\text{s.t. } ||\mathbf{u}|| = 1$$

- Lagrangian:

$$\mathcal{L} = \mathbf{u}^T \mathbf{C}\,\mathbf{u} - \lambda\,(\mathbf{u}^T \mathbf{u} - 1)$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = 2\,\mathbf{C}\,\mathbf{u} - 2\,\lambda\,\mathbf{u}$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = 0 \;\Leftrightarrow\; \mathbf{C}\,\mathbf{u} = \lambda\,\mathbf{u}$$

- So we need to find $\mathbf{u}$ such that $\mathbf{C}\,\mathbf{u} = \lambda\,\mathbf{u}$
- Have you seen such an equation before?

PCA – Derivation (4)

$$\max_{\mathbf{u}} \mathrm{Covar}(\mathbf{X}\,\mathbf{u})$$
$$\text{s.t. } ||\mathbf{u}|| = 1$$

- Lagrangian:

$$\mathcal{L} = \mathbf{u}^T \mathbf{C}\,\mathbf{u} - \lambda\,(\mathbf{u}^T \mathbf{u} - 1)$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = 2\,\mathbf{C}\,\mathbf{u} - 2\,\lambda\,\mathbf{u}$$
$$\frac{\partial \mathcal{L}}{\partial \mathbf{u}} = 0 \;\Leftrightarrow\; \mathbf{C}\,\mathbf{u} = \lambda\,\mathbf{u}$$

- So we need to find $\mathbf{u}$ such that $\mathbf{C}\,\mathbf{u} = \lambda\,\mathbf{u}$
- Have you seen such an equation before?
- λ is an Eigenvalue of $\mathbf{C}$, and $\mathbf{u}$ is its Eigenvector

PCA – Derivation (5)

- Which of the Eigenvalues of $\mathbf{C}$ do we want?
- Don't forget what we want to maximize:

$$\mathbf{u}^T\mathbf{C}\mathbf{u} = \mathbf{u}^T\lambda\mathbf{u} = \lambda\mathbf{u}^T\mathbf{u} = \lambda$$

- The constraint was: $\mathbf{u}^T\mathbf{u} = 1$
- So we want to find the largest Eigenvalue

PCA – Derivation (6)

- What's the next Principal Component, $\mathbf{u}_2$?
- $\mathbf{u}_2$ must be uncorrelated/orthogonal with $\mathbf{u}_1$: $\mathbf{u}_2 \cdot \mathbf{u}_1 = 0$.
- Adds a new constraint to the Lagrangian:

$$\mathcal{L} = \mathbf{u}_2^T \mathbf{C} \mathbf{u}_2 - \lambda(\mathbf{u}_2^T \mathbf{u}_2 - 1) - \phi(\mathbf{u}_2 \cdot \mathbf{u}_1) \cdots \Rightarrow \mathbf{C}\mathbf{u}_2 = \lambda \mathbf{u}_2$$

- We just look for the 2nd largest Eigenvalue and its Eigenvector.
- Derivation works just the same for all the following PCs as well.
- PCA is unique up to the directions (signs) of the eigenvectors.

PCA – Derivation (7)

- PCA is the singular value decomposition of the data matrix $\mathbf{X}$:

$$\mathbf{X} = \mathbf{V}\,\mathbf{D}\,\mathbf{U}^T$$

  with $n > d$, $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\mathbf{V} \in \mathbb{R}^{n\times n}$ orthogonal, $\mathbf{D} \in \mathbb{R}^{n\times d}$ diagonal (rectangular) with positive diagonal entries $\sqrt{\lambda_j}$, $\mathbf{U} \in \mathbb{R}^{d\times d}$ orthogonal.
- Equivalently, PCA is the eigenvalue decomposition of the covariance matrix $\mathbf{C}$:

$$\mathbf{C} = \frac{1}{n}\,\mathbf{U}\,\mathbf{D}_d\,\mathbf{U}^T$$

  with $\mathbf{C} \in \mathbb{R}^{d\times d}$ symmetric and positive definite, $\mathbf{U} \in \mathbb{R}^{d\times d}$ orthogonal, $\mathbf{D}_d \in \mathbb{R}^{d\times d}$ diagonal with positive diagonal entries λ_j.

PCA – Derivation (8)

- $\mathbf{D}_d$ is a diagonal $d \times d$ matrix of the (ordered) eigenvalues of $n\,\mathbf{C} = \mathbf{X}^T\mathbf{X}$:

$$\mathbf{D}_d = \begin{pmatrix} \lambda_1 & 0 & 0 & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ 0 & 0 & \lambda_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \lambda_d \end{pmatrix}$$

  with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$.

- $\mathbf{U}$ is an orthogonal (in fact orthonormal) $d \times d$ matrix of the eigenvectors:

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 & \dots & \mathbf{u}_d \end{pmatrix}$$

  with $\mathbf{u}_j^T \mathbf{u}_k = \delta_{j,k}$.

PCA – Derivation (9)

- PCA projection (onto new representation of the data):

$$\mathbf{Y} = \mathbf{X}\,\mathbf{U}$$

- In case you want to downproject onto a smaller-dimensional space of dimension $k < d$, construct $\mathbf{W} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k) \in \mathbb{R}^{d \times k}$ of only the first k eigenvectors and compute

$$\mathbf{Y} = \mathbf{X}\,\mathbf{W}$$

Explained Variance

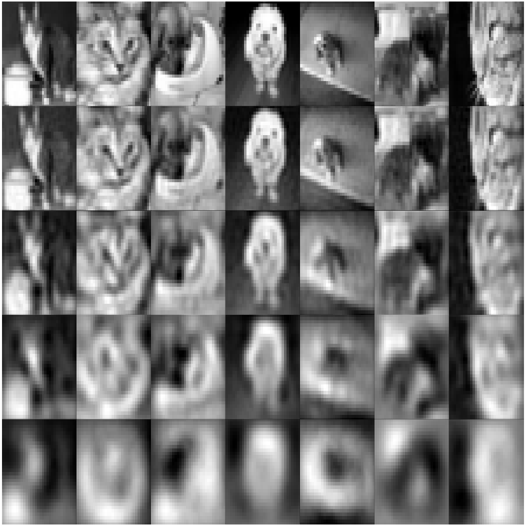
- PCA defines a new space for the data, where each dimension explains less of the original variance than the last.
- Amount of variance explained can be seen from the Eigenvectors:

$$\mathrm{Covar}(\mathbf{X}) = \mathbf{C} = \frac{1}{n}\sum_{j=1}^{d} \lambda_j \mathbf{u}_j \mathbf{u}_j^T$$

- The amount of “explained variance” by principal component j is

$$v_j = \frac{\lambda_j}{\sum_{k=1}^{d} \lambda_k}$$

Example



Recapitulation: Pros and Cons of PCA

- We provided some intuition and a proof how to derive PCA

Recapitulation: Pros and Cons of PCA

- We provided some intuition and a proof how to derive PCA
- PCA is used to visualize data: downproject data to a small number of dimensions and plot it

  (sometimes insightful even for very high-dimensional data)
- In Machine Learning, PCA is often used to reduce dimensionality:
  use enough components to explain 75 %, 90 % or 95 % of the variance:
  - + often drastically reduces amount of data $\rightarrow$ faster algorithms, less memory needed
  - + often performs (much) better, since less overfitting/noise
  - – destroys sparseness
  - – no guarantees (maybe you throw away important information)