

# Principle component analysis and covariance

Oliver W. Layton

CS251: Data analysis and visualization

Lecture 19, Spring 2019

Wednesday March 19

# MAP Estimates for height data

2D Gaussian Mean vector:

| parameter | mean (cm) |
|-----------|-----------|
| mu        | 154.61    |
| sigma     | 7.77      |

2D Gaussian Covariance matrix:

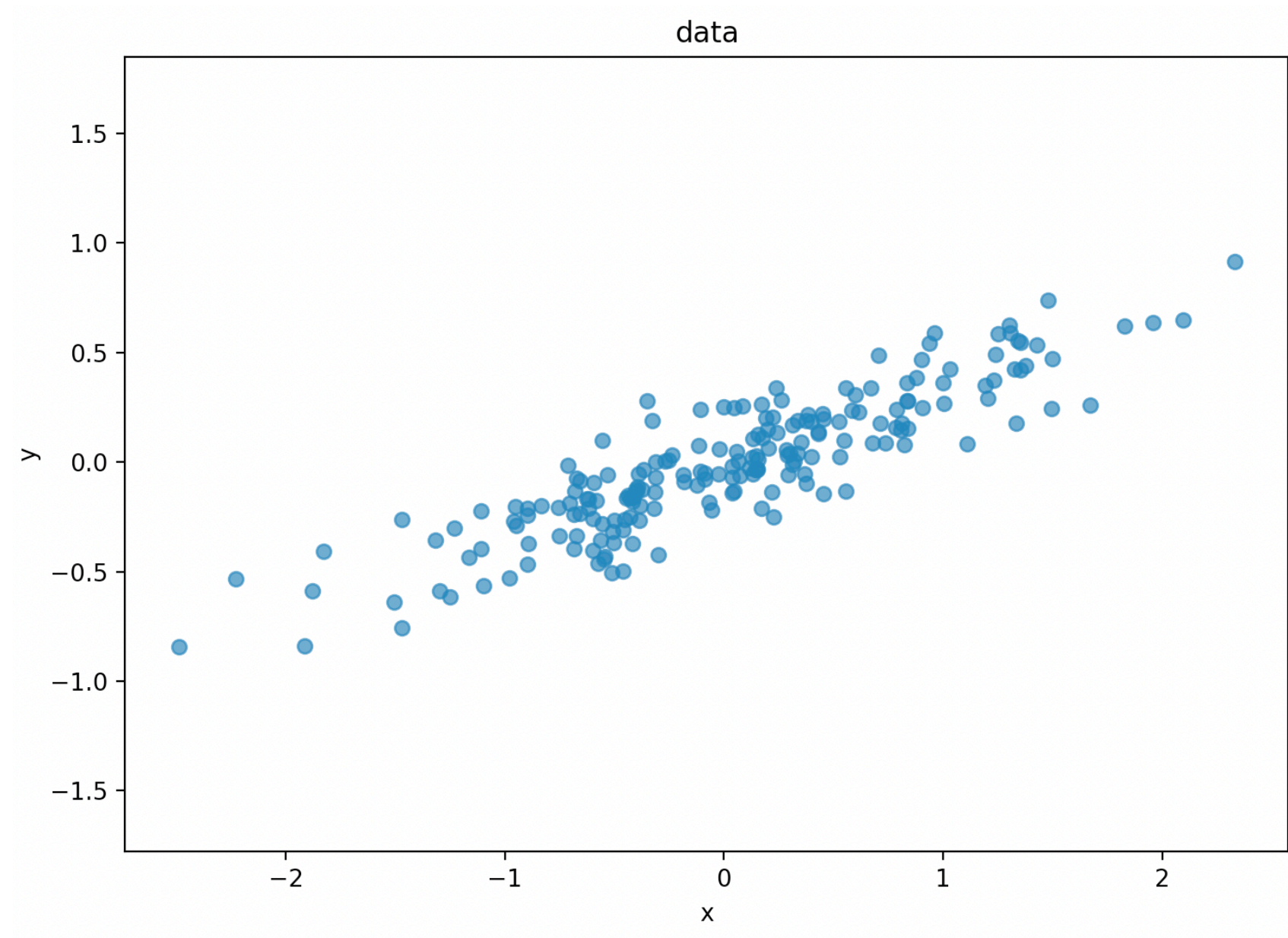
|       | mu       | sigma    |
|-------|----------|----------|
| mu    | 0.172456 | 0.001159 |
| sigma | 0.001159 | 0.086376 |

A 2D Gaussian with the above  $\mu$  and  $\Sigma$  matrix values captures the height population data.

# MAP Correlation matrix (from covariance matrix)

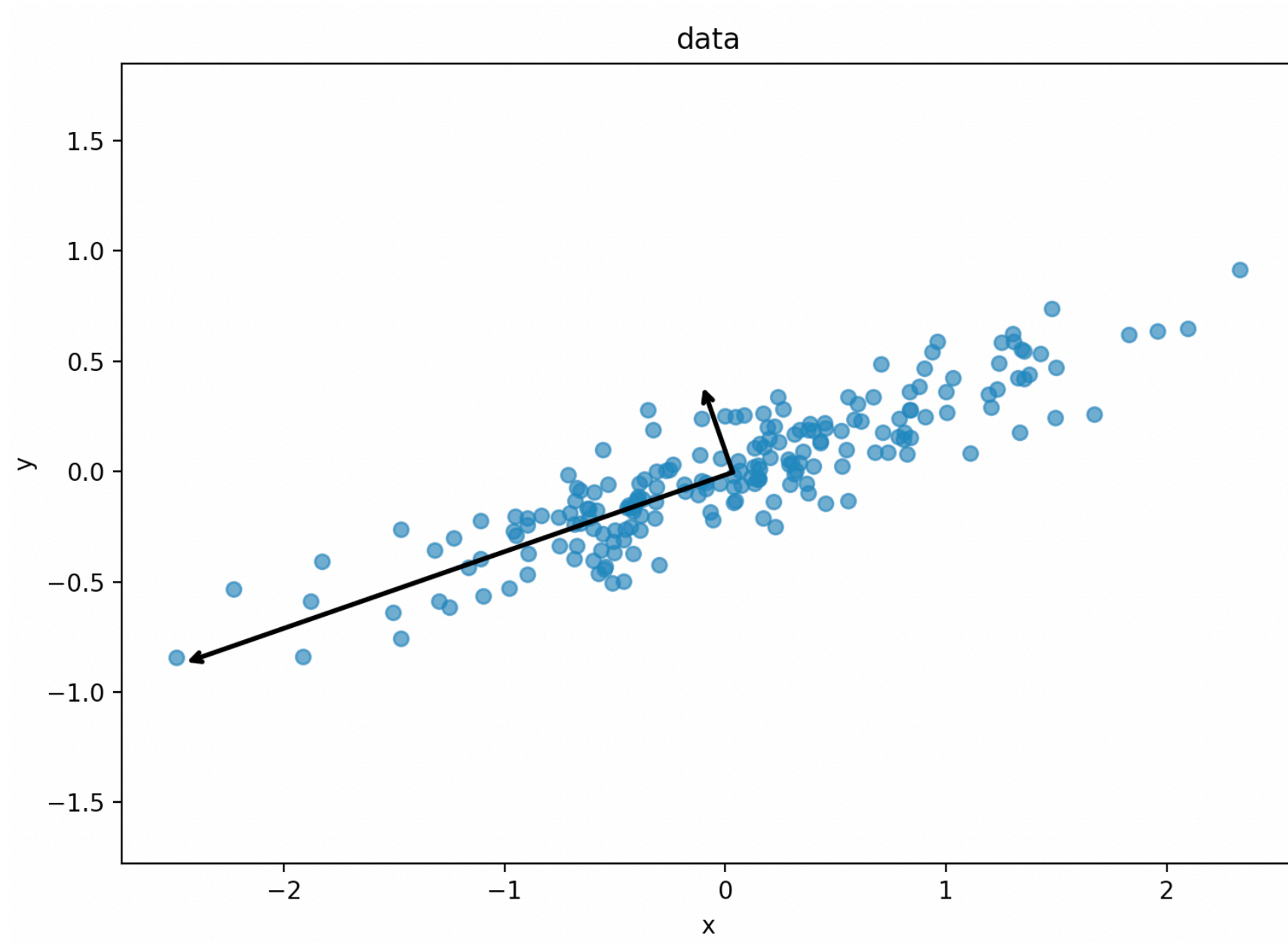
|       | <b>mu</b> | <b>sigma</b> |
|-------|-----------|--------------|
| mu    | 1.0       | 0.009494     |
| sigma | 0.009494  | 1.0          |

Tells us that knowing either mu or sigma tells us next to nothing about the other value. They are independent.



- Are these data really 2D?
- What does the covariance matrix look like?

# Axes of variation of the data



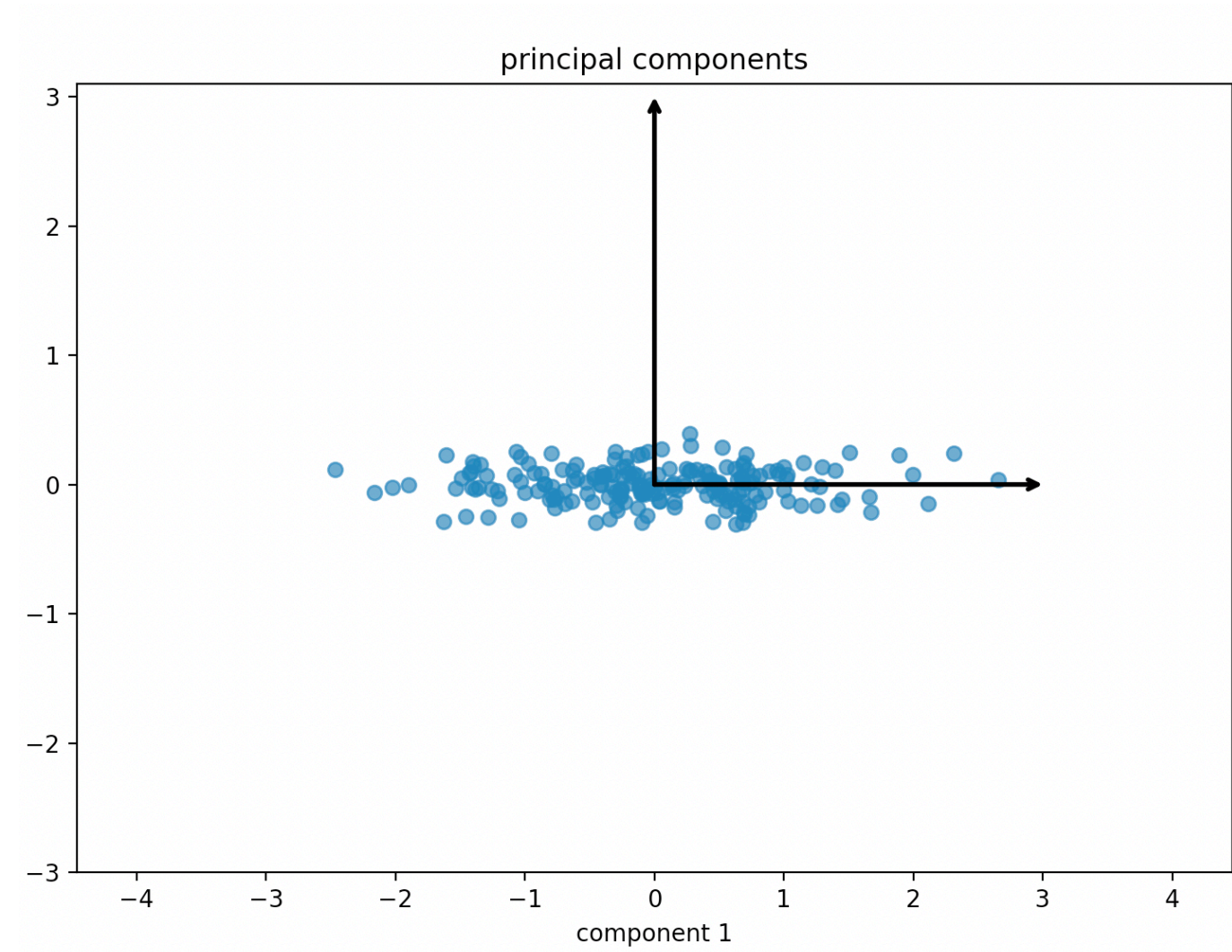
# Principle component analysis (PCA)

- We can identify the "real" axes of the data.
- Leverages covariance matrix  $\Sigma$ . Why?
  - Quantifies how much of the variation spans each data axis.
- Compute **spectral decomposition** of  $\Sigma$  (via linear algebra).
  - **Eigenvectors**: Axes of variation of data (vectors on previous slide).
  - **Eigenvalues**: Proportional to how much variance lies in each eigenvector direction (length of vectors on previous slide).
- We can project the data into **PCA space** where the eigenvectors<sup>1</sup> become the "new x-y axes".

---

<sup>1</sup> They form an orthogonal basis. Aka it's a rotated, data-aligned, valid coordinated system we can use.

# Data in PCA space



Instead of using default "Euclidean" axes, PCA use axes specifically tailored/oriented to the data variation.

# Dimensionality reduction

- Imagine this example in higher dimensions (e.g. 100) and having the variance spread out only in 2-3 main directions. What a waste!
- If we compute the 100 eigenvalues and eigenvectors, what can we do to compress the number of dimensions we use?
- We can decide whether to keep a dimension of data based on how big the eigenvalues are.
  - Threshold/throw out dimensions if the associated eigenvalues are "too small" relative to others.
  - This would mean we could **compress** 100 dimensional data into its 2 "real" dimensions! **We can actually visualize it too!**

# PCA workflow (1/2)

1. Center each data axis (Make mean =  $\vec{0}$ ).
2. (optional; at your own risk) normalize each variable separately (each has range =  $[0, 1]$ ).
  1. Data *homogenous*: variables have same units. Usually don't want to normalize separately (distorts relative relationships). Might want to *globally* normalize ALL variables.
  2. Data *heterogenous*: variables have different units. Usually more defensible to normalize separately.

# PCA workflow (2/2)

- 3) Compute covariance matrix  $\Sigma$ .
- 4) Compute eigenvectors and eigenvalues of  $\Sigma$  (e.g. via spectral decomposition or other linear algebra techniques).
- 5) Sort eigenvalues high-to-low (e.g. 3.2, 1.1, 0.03)
- 6) Compute portion variance in each eigen-dimension by successive sum of sorted eigenvalues, divided by sum of all eigenvalues (e.g.  $3.2/(3.2+1.1+0.03)$ ,  $(3.2+1.1)/(3.2+1.1+0.03)$ )
- 7) Threshold dimensions based on the variance you want to keep in your data (e.g. 90%)
- 8) Project data into **PCA space** determined by the eigenvectors of the surviving dimensions.

Let's apply the workflow to a simple  
example on the board