

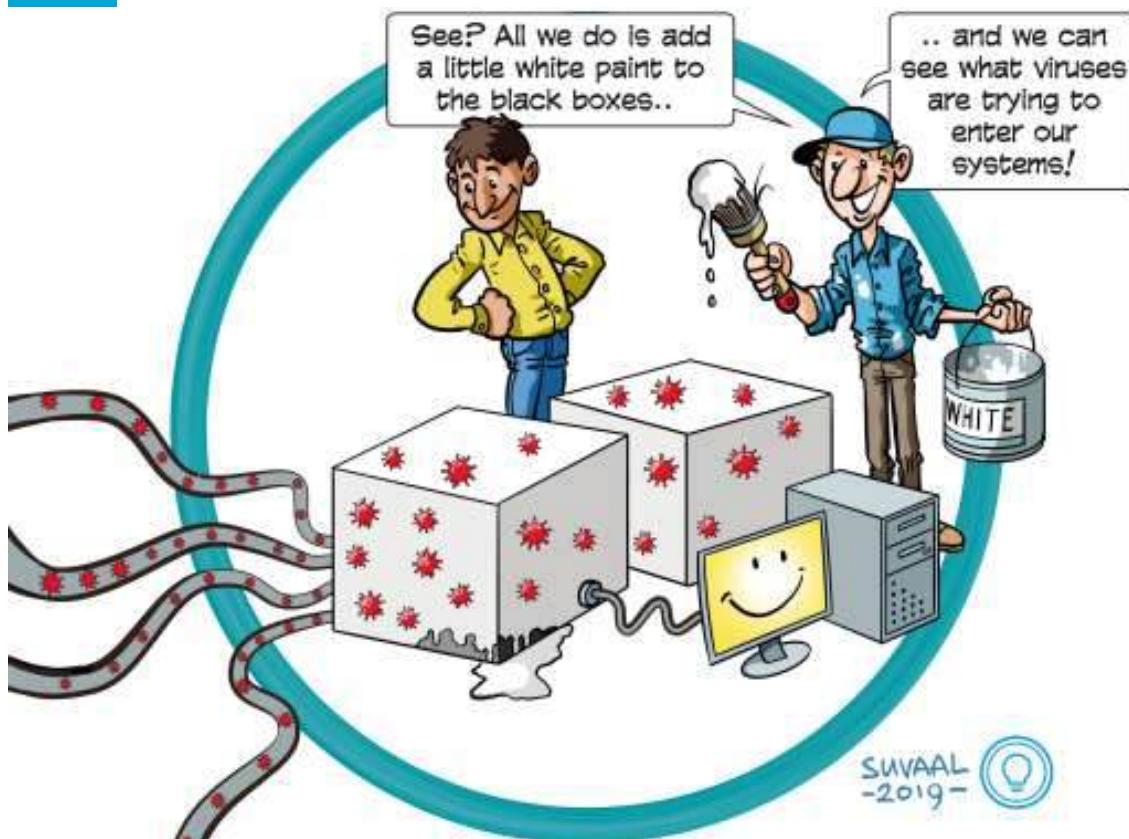
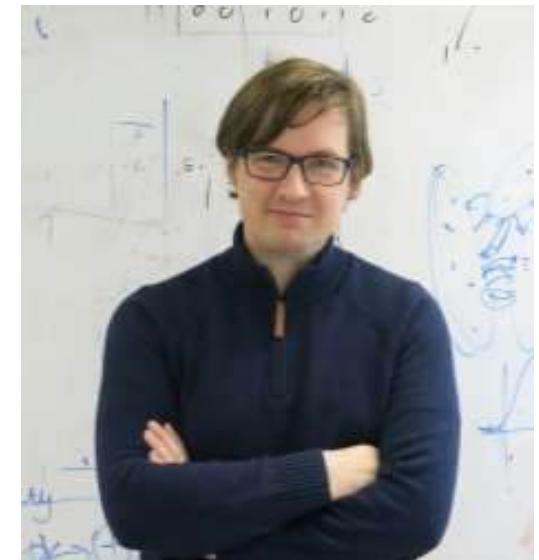
Datamining CSE2525

Nov 11, 2024



Sicco Verwer

Associate professor in Algorithms



<https://cyber-analytics.nl/>

- Research:
 - algorithm design for learning interpretable models
 - applications in cyber security
 - awards: Veni, Vidi grants, Test-of-Time award
- Teaching:
 - cyber data analytics
 - AI for software reverse engineering
 - data mining

Avishek Anand

- Associate Professor at the Web Information Systems (ST)
- Topics: Information retrieval, NLP, Explainable AI
- Teaching: Information Retrieval, NLP, Data mining
- Topics covered in this course
 - Text Data Mining
 - How do we mine massive collections of text data ?
 - Word embeddings, indexing text
 - Graph data mining
 - How do we mine large graphs ?
 - Graph embeddings, graph analysis



Nergis Tömen

- Assistant Professor at Intelligent Systems (INSY)
- Topics: Biologically-inspired machine vision, neuromorphic computing
- Labs:
 - [Computer Vision Lab](#) (member)
 - [Biomorphic Intelligence Lab](#) (director)
 - [Biomedical Intervention Optimisation Lab](#) (director)
- Teaching:
 - (MSc) Seminar Computer Vision by Deep Learning
 - (MSc) Machine Learning 2
 - (BSc) Data Mining
- Topics covered in this course:
 - Matrices, PCA, Matrix decomposition, Recommender systems



Today's goals

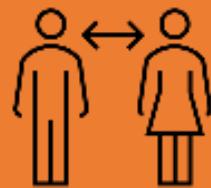
- Course goals
- Course logistics
- Course content overview
- What data mining is all about
- A word of caution

Course goals

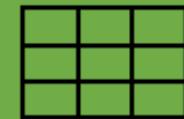
- In this course you learn about Data Mining:
 - Several **key algorithms**, you must know:
 - How to implement them
 - Their strengths and weaknesses in practice
 - Why and when to use these in practice
 - Core **concepts**:
 - What they are – including theory –
 - Which concepts exist for different data types
 - Practical **skills**:
 - What concept to use for a given problem
 - How to successfully apply algorithms in practice

Three Verticals

Distances



Matrices



Counting



Schedule	Week	Day	Date	Time	Topic	Lab Assignment Deadline
Lecture 1	2.1	Mon	Nov 11	13:45-15:45	Introduction	Lab 1: Anomaly detection
Lecture 2	2.1	Thu	Nov 14	10:45-12:45	Anomaly detection (DTW for Lab 1)	
Lecture 3	2.2	Mon	Nov 18	13:45-15:45	Distances (use case discussion)	
Lecture 4	2.2	Thu	Nov 21	10:45-12:45	Matrices (PCA for Lab 1)	
Lecture 5	2.3	Mon	Nov 25	13:45-15:45	Embeddings	
Lecture 6	2.3	Thu	Nov 28	10:45-12:45	Clustering (for Lab 2)	Lab 1 due date
Lecture 7	2.4	Mon	Dec 2	13:45-15:45	Discrimination (discussion)	Lab 2: Graph Clustering
Lecture 8	2.4	Thu	Dec 5	10:45-12:45	Invited lecture?	
Lecture 9	2.5	Wed	Dec 11	13:45-15:45	Graph Mining	
Lecture 10	2.5	Thu	Dec 12	10:45-12:45	MinHashing (for Lab 3)	
Lecture 11	2.6	Mon	Dec 16	13:45-15:45	Indexing	
Lecture 12	2.6	Thu	Dec 19	10:45-12:45	Sketching	Lab 2 due date
Lecture 13	2.7	Mon	Jan 6	13:45-15:45	NMF (for Lab 3)	Lab 3: Hashing/NMF
Lecture 14	2.7	Thu	Jan 9	10:45-12:45	Recommender systems (for Lab 3)	
Lecture 15	2.8	Mon	Jan 13	13:45-15:45	Manifold learning	
Lecture 16	2.8	Thu	Jan 16	10:45-12:45	Data Visualization (discussion)	
Lecture 17	2.9	Mon	Jan 20	13:45-15:45	Exam summary slides/Q&A	
Lecture 18	2.9	Thu	Jan 23	10:45-12:45	Mock exam answers/Q&A	Lab 3 due date
Exam	2.10	Mon	Jan 27	13:30-16:30	Weblab exam	

Teaching methods

- Lectures: 18
 - 13 content lectures
 - 2 invited lectures
 - 1 Intro
 - 2 Q&A
- Labs: 3
- Homework Assignments: 6

Lecture schedule

- Complete schedule: on Brightspace
- **Older lectures are recorded as backup at Collegerama that will be used sometimes in the flipped classroom**
- **E.g. On 24.11. - The lecture on distances will be flipped**
- *What is a flipped classroom ?*
 - *Please watch the video in Collegerama before come to class*
 - *In the lecture we do case studies – how do you apply what you have learnt in real-world scenarios ?*

Course material

- Required - Brightspace:
 - Lecture slides (after each class)
 - Lab exercises (beginning of the week)
 - Reading materials (book chapters and selected papers)
- Content from 2 books:
 - Mining of Massive Datasets
 - Data Mining
 - *Both are fully available through the TU Delft digital library!*
 - *Selected Chapters will be uploaded to Brightspace*

Lab sessions

- **Mandatory**

- 3 topics
- 9 sessions
- Lab sessions on Friday afternoon
- Assistance and feedback at lab session
 - Queue (<https://queue.tudelft.nl/requests>)
 - Mattermost (<https://mattermost.tudelft.nl/>)
 - Answers EWI with tag CSE2525 (<https://answers.ewi.tudelft.nl/>)
 - Kaggle (<https://www.kaggle.com>)
 - Weblab (<https://weblab.tudelft.nl/>)
 - Peer (<https://peer.tudelft.nl/>)
- Make sure you have a recent version of Python, including Numpy, Scipy, Pandas, Seaborn, Matplotlib on your own computer!

Lab sessions

- **Mandatory**

- 3 topics
- 9 sessions

- Lab sessions on Friday afternoon
- Assistance and feedback at lab session

- Queue (<https://queue.tudelft.nl/requests>)
- Mattermost (<https://mattermost.tudelft.nl/>)
- Answers EWI with tag CSE2525 (<https://answers.ewi.tudelft.nl/>)
- Kaggle (<https://www.kaggle.com>)
- Weblab (<https://weblab.tudelft.nl/>)
- Peer (<https://peer.tudelft.nl/>)

- Make sure you have a recent version of Python, including Numpy, Scipy, Pandas, Seaborn, Matplotlib on your own computer!

Please, do not use e-mail!
They will not be answered.

Lab sessions

- 3 topics
- Each lab topic has three components
 - **Algorithm implementation:** Distances, Matrices, Counting
 - **Building a pipeline:** Data transformation, analysis, and visualization
 - **Kaggle competition** [Bonus Points]
- Example - Topic 1 (this Friday): Anomaly detection
 - Algorithms to be implemented – DTW, PCA
 - Build an *anomaly detection pipeline* and evaluate its performance
 - Kaggle competition
- Labs in student pairs! Pair up as soon as possible, and register on Brightspace.



TI2736-C: Datamining Project 2018

Recommendation algorithm for movies.

#	Δpriv	Team Name	Kernel	Team Members	Score	Entries	Last
1	-	Boning Gong			0.81954	3	1y
2	-	Eksdie			0.82021	9	10mo
3	-	kamran			0.82161	1	1y
4	-	Niels de Bruin			0.82713	82	10mo
5	-	René van den Berg			0.82853	91	10mo
6	-	frenkvm			0.83135	33	10mo
7	-	Chris Mostert			0.83179	127	10mo
8	-	Alessandro Ariës			0.83460	76	10mo
9	▼ 1	Kaan Yilmaz			0.83539	53	10mo
10	▼ 1	mwlting			0.83552	127	10mo
11	▲ 2	Casper Boone			0.83556	89	10mo
12	▼ 3	Xilin			0.83665	41	10mo

Lab Evaluation

- Lab Evaluation – 30% of your final grade
- *Automatic evaluation* of the algorithmic component
- *Peer review* of the pipeline component
 - Please do your peer reviews, penalty if not completed
 - Your submissions will get 4 reviews
 - We will double-check the quality of the reviews

<https://peer.tudelft.nl/courses>

Lab Evaluation

- Lab Evaluation – 30% of your final grade
- *Automatic evaluation* of the algorithmic component
- *Peer review* of the pipeline component
- **Kaggle competition** – should beat our baselines to get bonus points
- No solutions! Ask for help during labs.
- Top 3 Kaggle submissions will be shared and asked to present

Lab Evaluation

- Lab Evaluation – 30% of your grade
- *Automatic evaluation* of the algorithmic component
- *Peer review* of the pipeline component
- **Kaggle competition** – should beat our baselines to get bonus points
- No solutions! Ask for help during labs.
- Top 3 Kaggle submissions will be shared and asked to present

No scikitlearn or other ML tool
will be used, everything is
build from scratch!

How does Peer Review work ?

The screenshot shows a web-based course management system. At the top, there is a green header bar with the text "Peer Review DATA MINING" and "ROLE: TEACHER". On the right side of the header, there are links for "Courses" and "Avishek Anand". Below the header, a navigation bar contains links for "Course Home", "Assignments", "Teacher Management", "TA Management", "Student Management", and "Statistics". The main content area has a light gray background. It displays the text "Assignments / Lab 1 - anomaly detection" and a green button on the right labeled "Edit assignment".

Homework Assignments

- 6 of them
- Idea: Mostly descriptive questions and problems
 - Reflects the type and hardness of questions you can expect in the final exam
- Solutions will be given
- NOT be graded or discussed in the lab

Final Exam

- No (partial) transfer from previous years
- WebLab (<https://weblab.tudelft.nl>) exam: 70%
 - One resit
- Wednesday Jan 31, 2024
 - Weblab exam (Osiris + weblab registration)
- Open and multiple-choice questions
- No programming questions this year!
- **Closed book** – calculator is allowed

*<https://mytimetable.tudelft.nl> is authoritative

Course changes

- We planned to remove 25% of the older content
- Removed content:
 - Graph cuts
 - Community detection
- New content:
 - High-dim data visualization

Expected prior knowledge

- Discrete mathematics:
 - sets, intersections, and unions
- Linear algebra:
 - matrix multiplication, projections, eigenanalysis
- Probability and statistics:
 - Gaussians, correlation, covariance
- Graph theory:
 - adjacency matrix, degree, clique, bipartite graph, shortest path
- Data structures:
 - hash tables and indexes
- Programming:
 - Python programming skills
- Machine learning:
 - basic algorithms: logistic regression, random forest, svm, ...

Prior courses

- CSE1100/TI1206 Object-oriented programming
CSE1305/TI1316 Algorithms and Data Structures
CSE1200/TI1106M Calculus
CSE1205/TI1206M Linear Algebra
CSE1210/TI2216M Probability Theory and Statistics
- CSE2510 Machine Learning
CSE2520/TI2736-B Big Data Processing
- Information only - not enforced
- You are responsible for your study success!

Feedback

- When:
 - Any time
- How:
 - E-mail: dm-cs-ewi@tudelft.nl
 - Anonymous evaluations (EvaSys/EvaTool)

Logistics summary

- Optional practicals
 - Questions through online tools and at sessions
- 3 mandatory Labs – peer review + automated tests + kaggle
- Closed-book Exam
- Only for feedback:
 - dm-cs-ewi@tudelft.nl

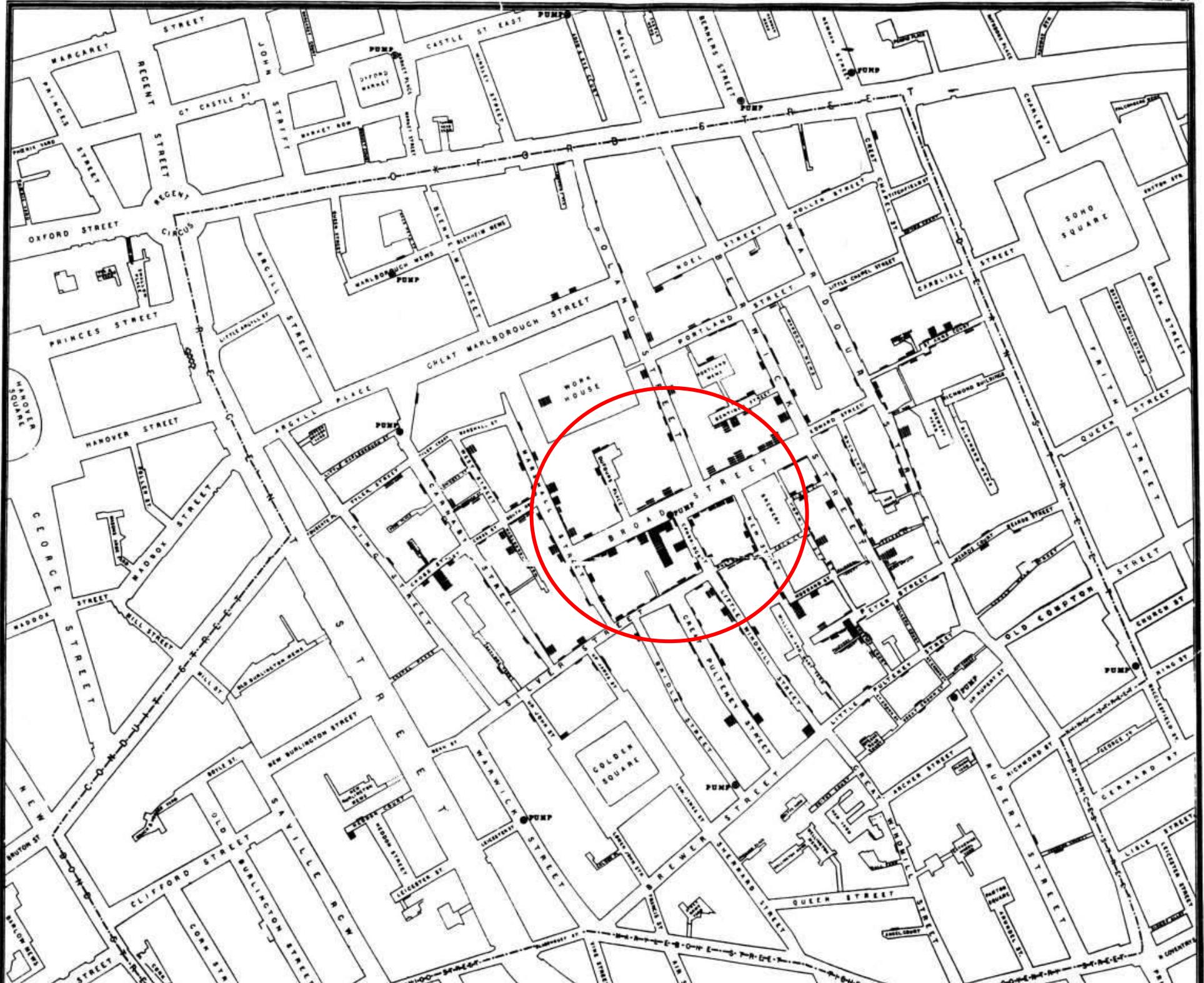


What is Data Mining?

What is data mining?

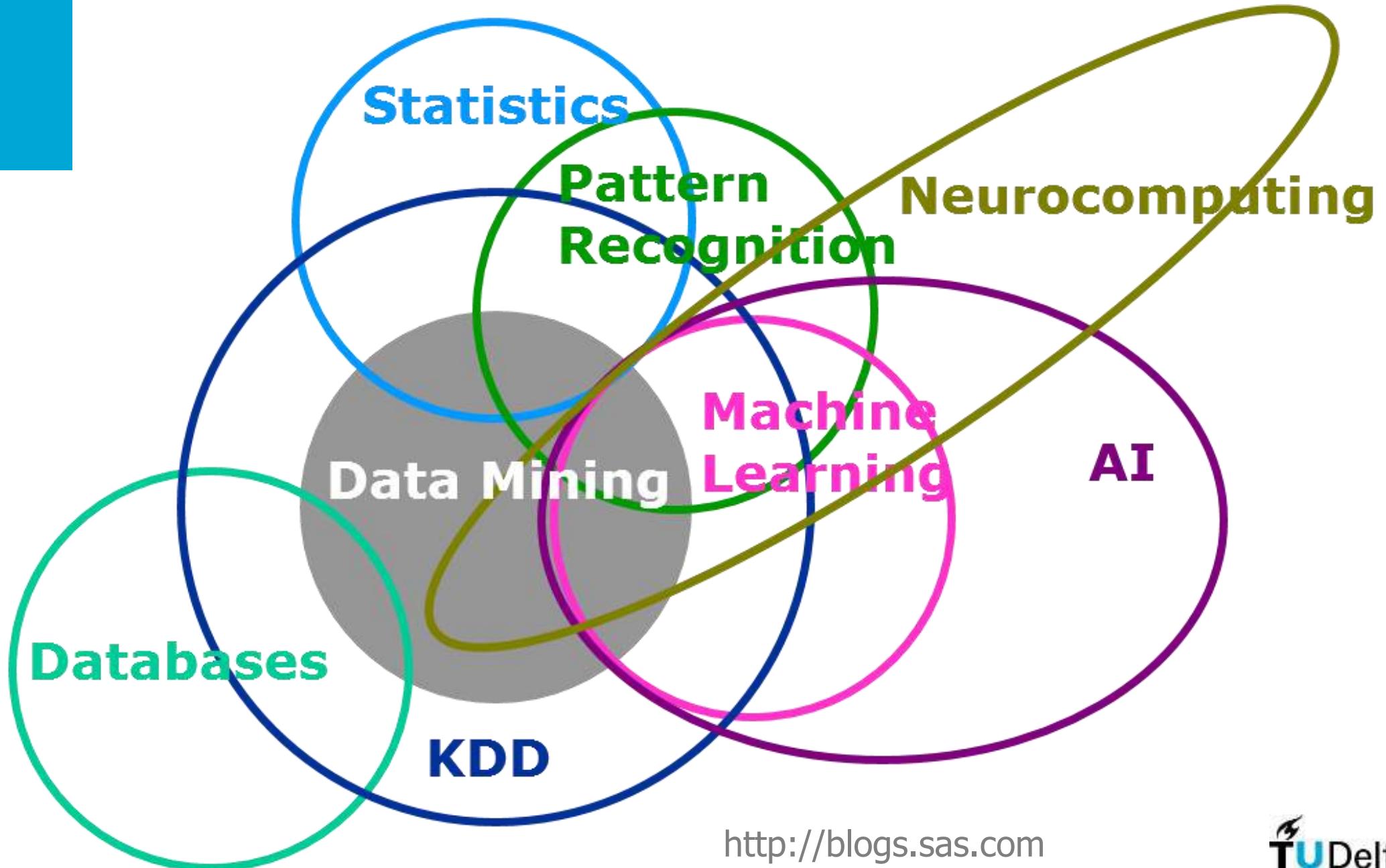
- The first data miner?
- John Snow (1813-1858)
- Plotted cholera cases on a map of London







Data mining in context



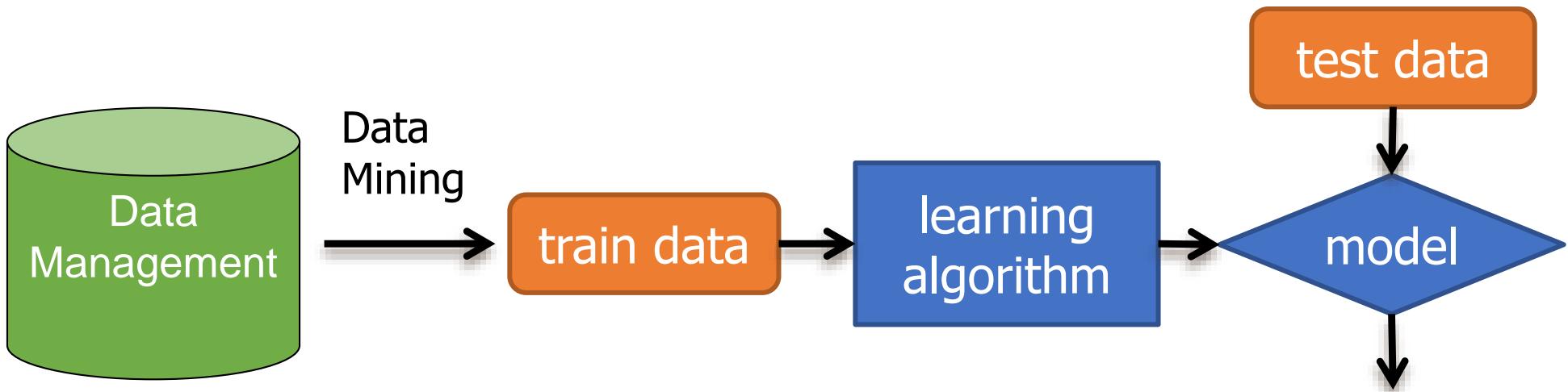
What is data mining?

- “..is the process of **searching and analyzing** a large batch of **raw data** in order to **identify** patterns and extract useful information”
- "...is the development of **models for data** in order to extract **information** from that data."
- "... is the process of **analyzing** data from different perspectives and summarizing it into **useful information**."
- "... is done by **humans**"

What is data mining?

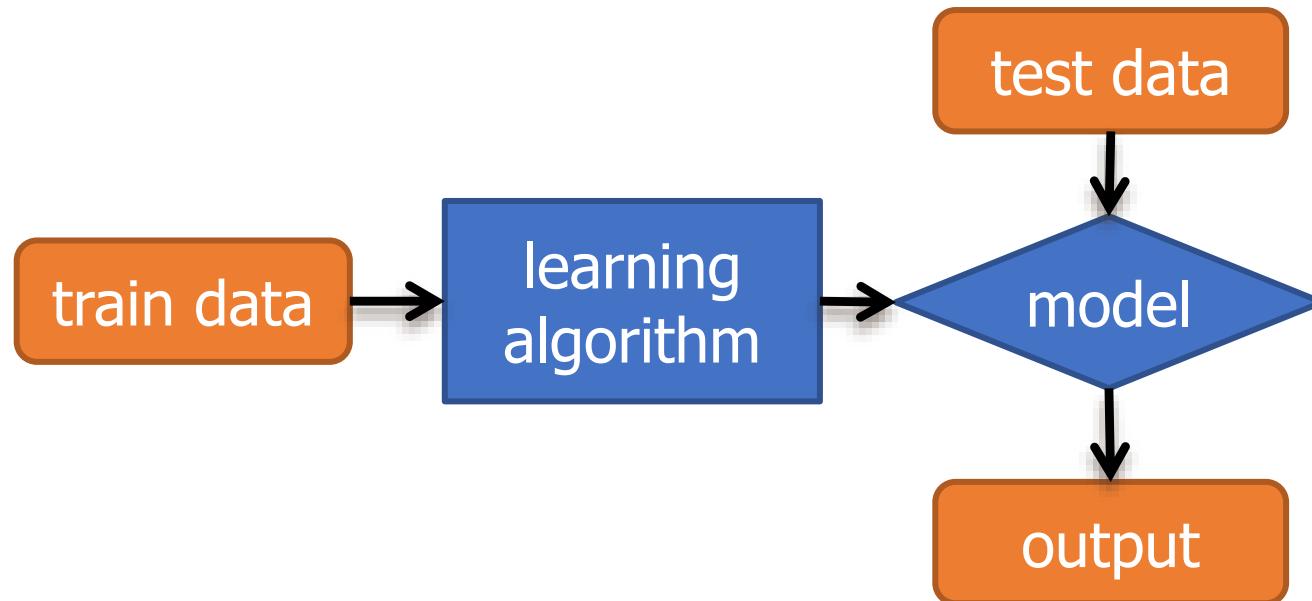
The primary goal of data mining is to extract useful information from a large volume of data and transform it into an understandable structure for further use.

Data Mining vs. Machine learning vs Data Management



Data Mining vs. Machine learning

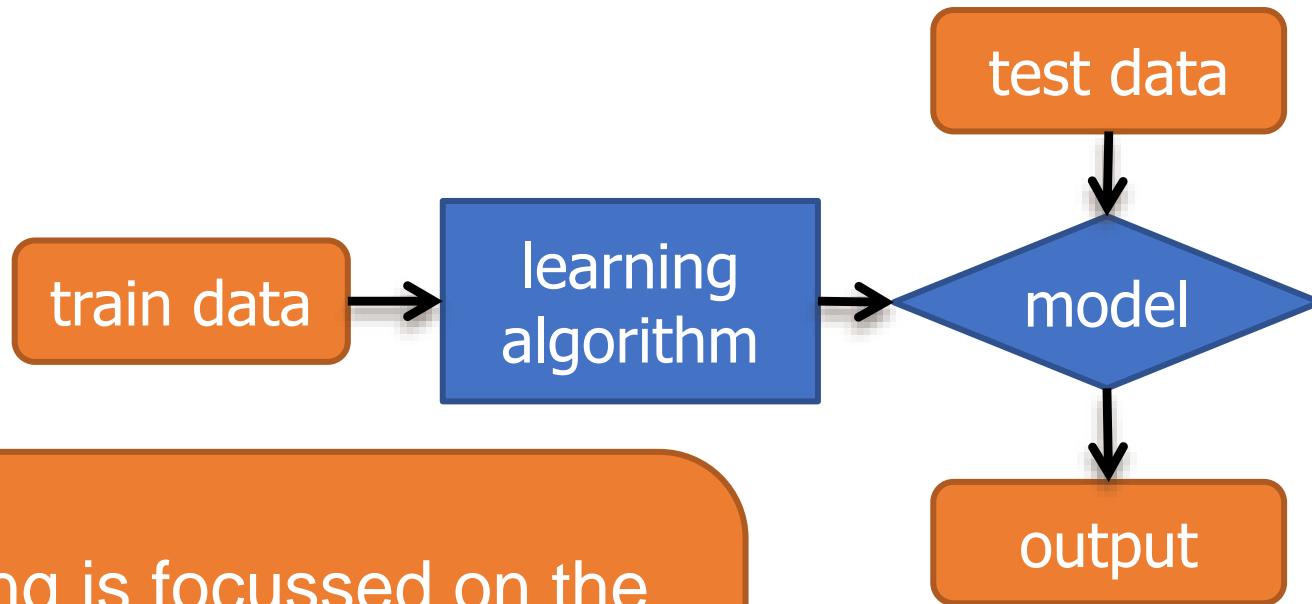
- Classic ML Approach:



- take a huge data set
- compute features
- train a classifier
- deploy the classifier on test

Data Mining vs. Machine learning

- Classic ML Approach:

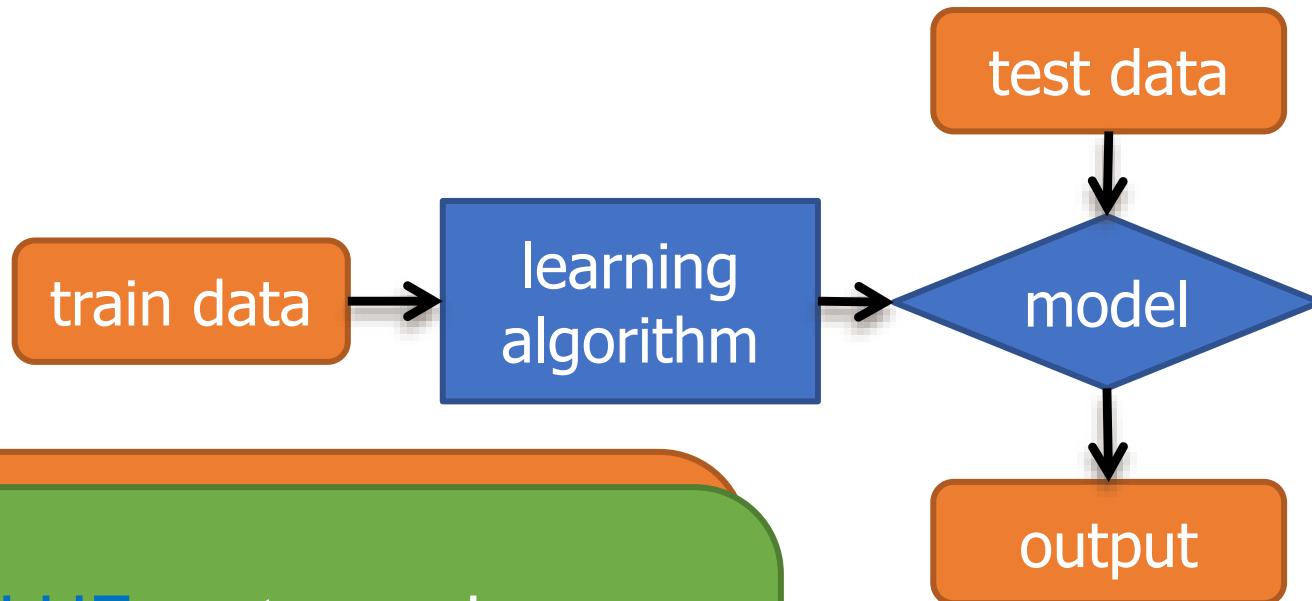


Data Mining is focussed on the
ORANGE parts:

What to do with input and output?

Data Mining vs. Machine learning

- Classic ML Approach:



For the **BLUE** parts we do care:

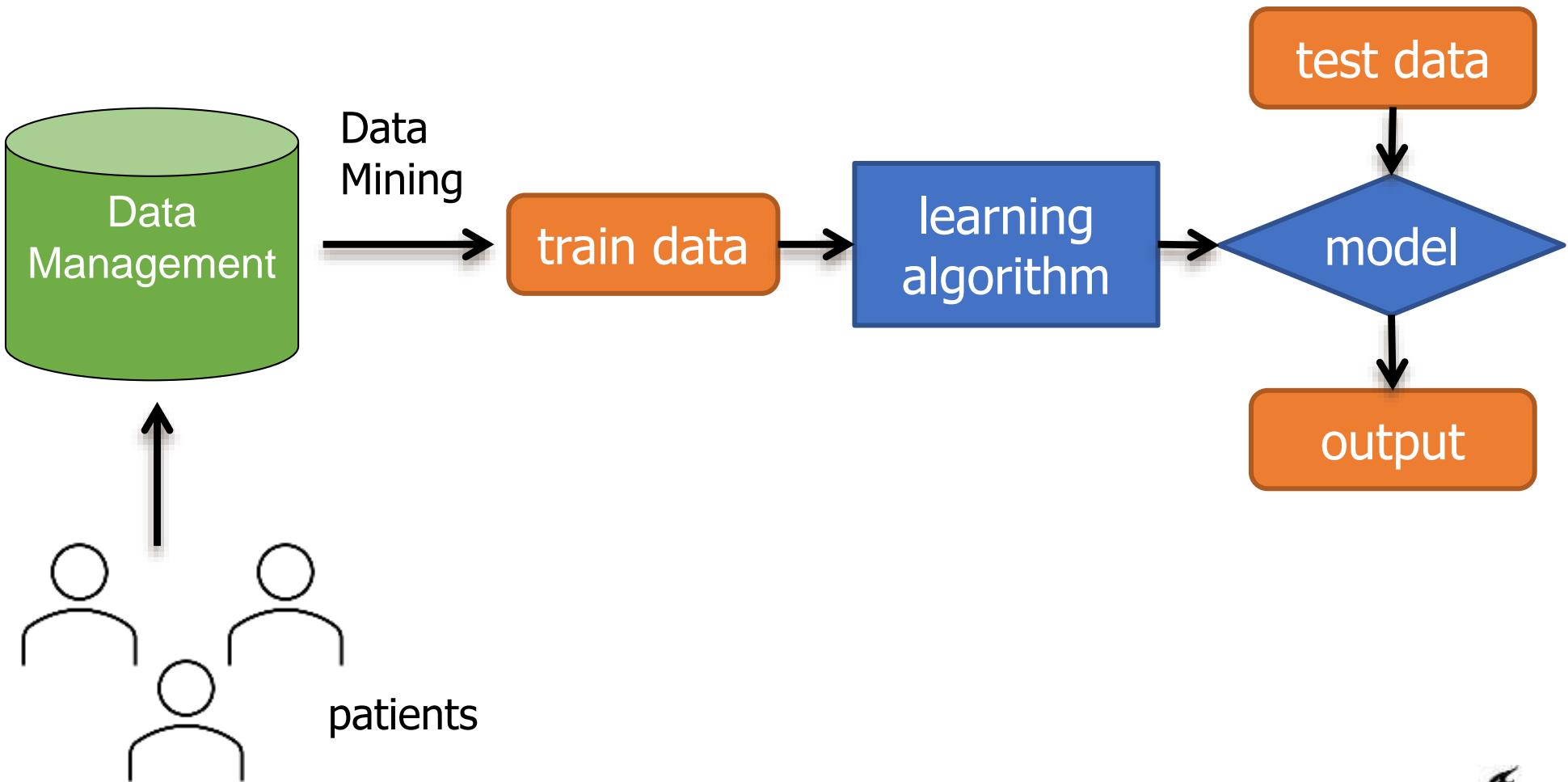
What algorithm to use and how to run it on our data?

Data Mining: Healthcare system

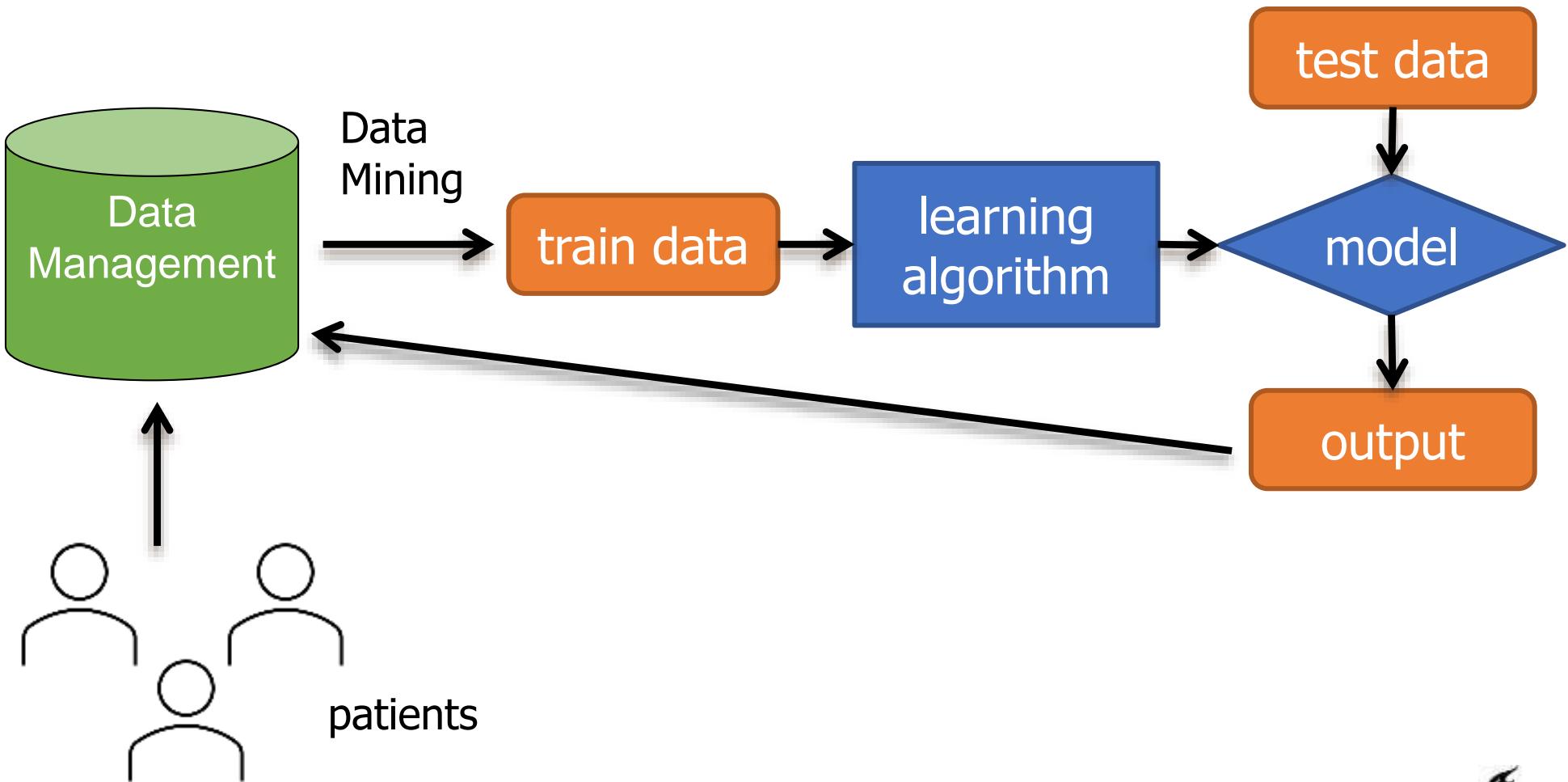
- **Objective:** To **find/uncover patterns** and correlations in patient data
- **Process:** The system analyzes a vast database of patient records, including symptoms, diagnostics, treatments, and outcomes.
 - Data mining techniques: clustering, association rule mining, and anomaly detection
- **Outcome:** The system identifies patterns such as **common symptoms** associated with particular diseases, **effective treatments for specific conditions**, and any anomalies like **rare side effects** of treatments

Patients with a certain combination of symptoms (e.g., **fever**, **cough**, and **shortness of breath**) often test positive for a **specific respiratory illness**.

Healthcare System



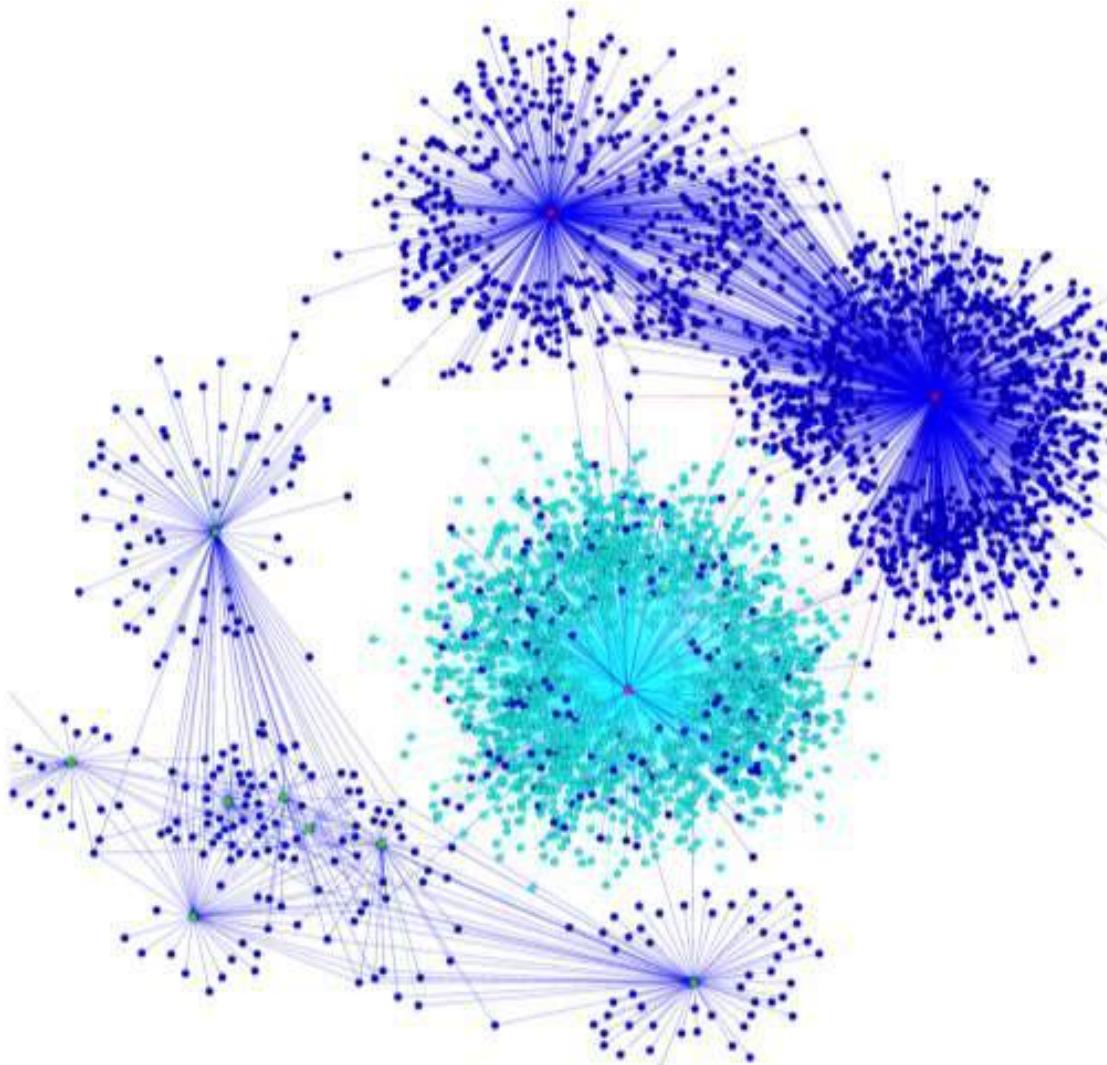
Healthcare System



Often learning is unsupervised



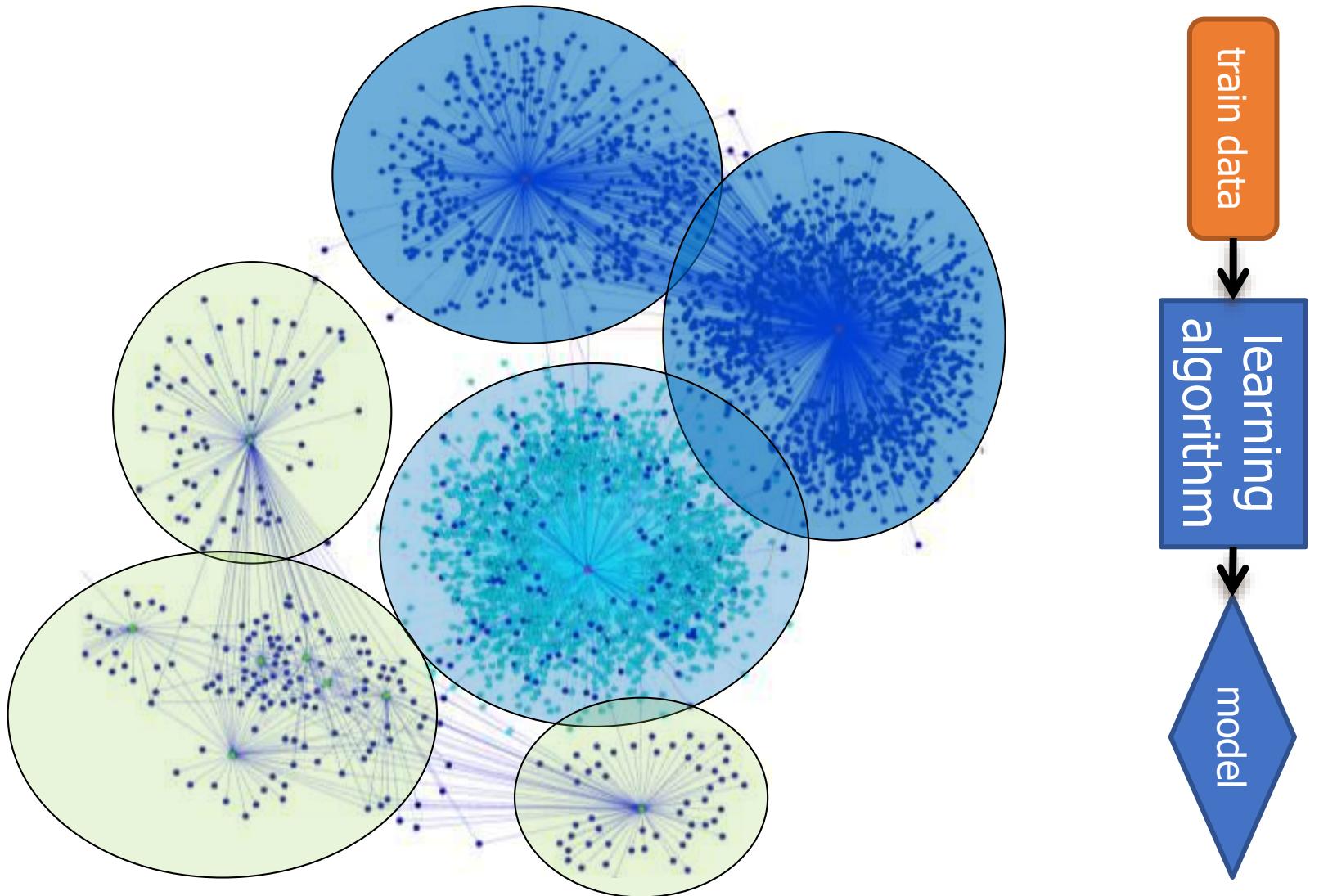
Example: Network/Graph data



Beyond Labeling: Using Clustering to Build Network Behavioral Profiles of Malware Families. Azqa Nadeem, Christian Hammerschmidt, Carlos H. Ganán, Sicco Verwer. In *Malware Analysis using Artificial Intelligence and Deep Learning*, Springer, 2020. (Forthcoming)

Hybrid Connection and Host Clustering for Community Detection in Spatial-temporal Network Data. Mark Patrick Roeling, Azqa Nadeem, Sicco Verwer. In *Machine Learning for Cybersecurity (MLCS)*, 2020

Community detection

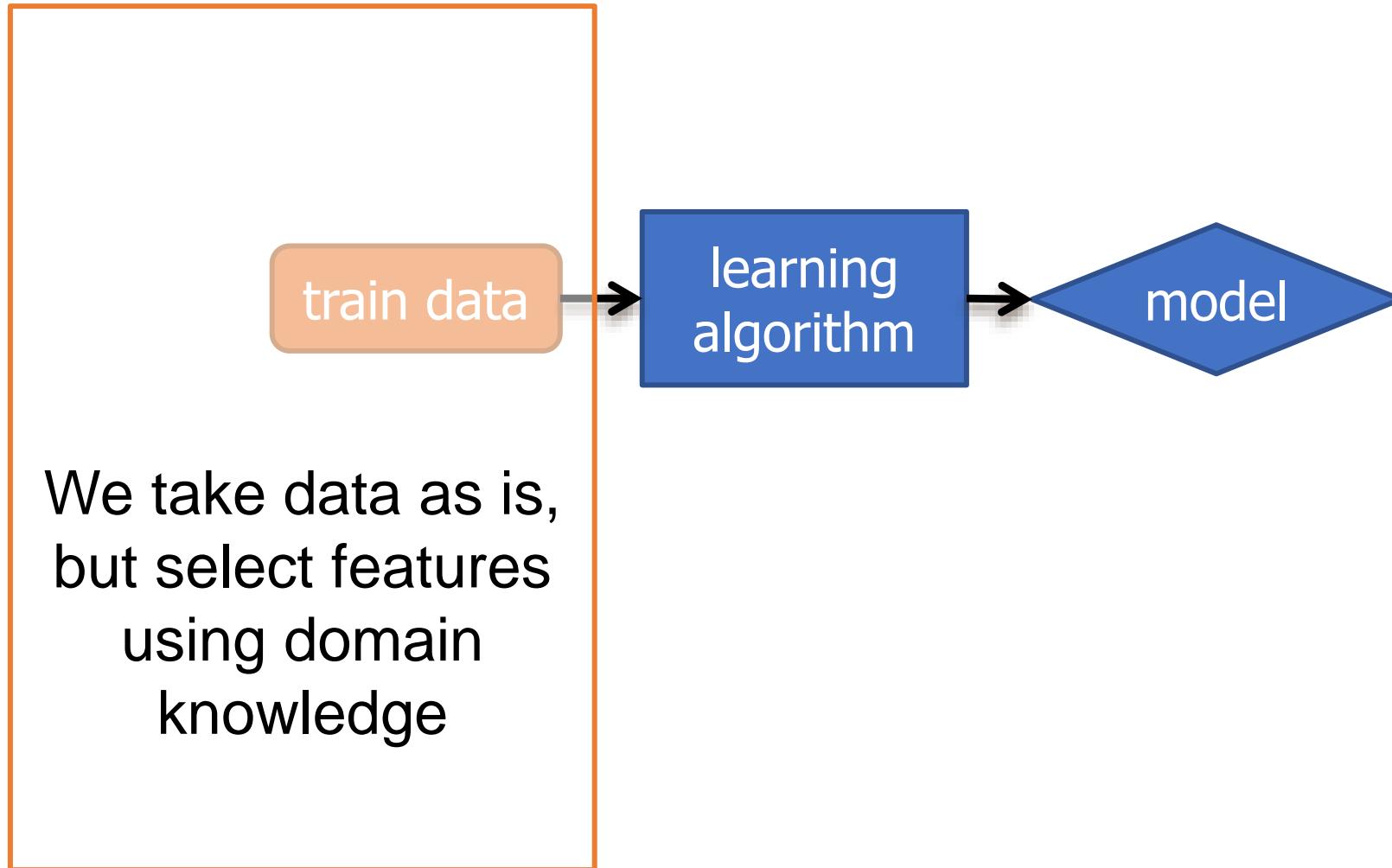


Use heatmaps to visualize groups

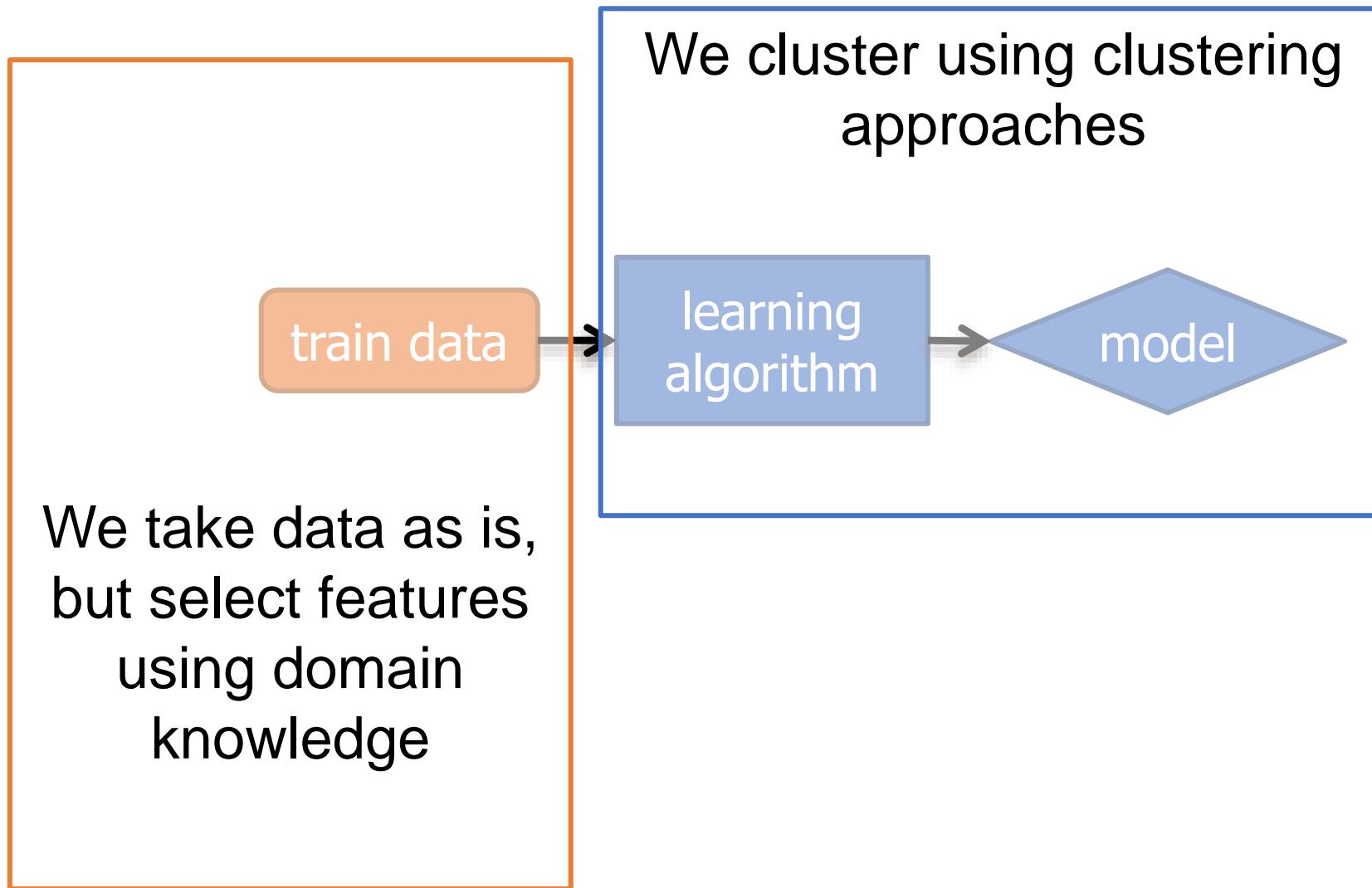


Row = connection
Column = time
Cell = Bytes transferred

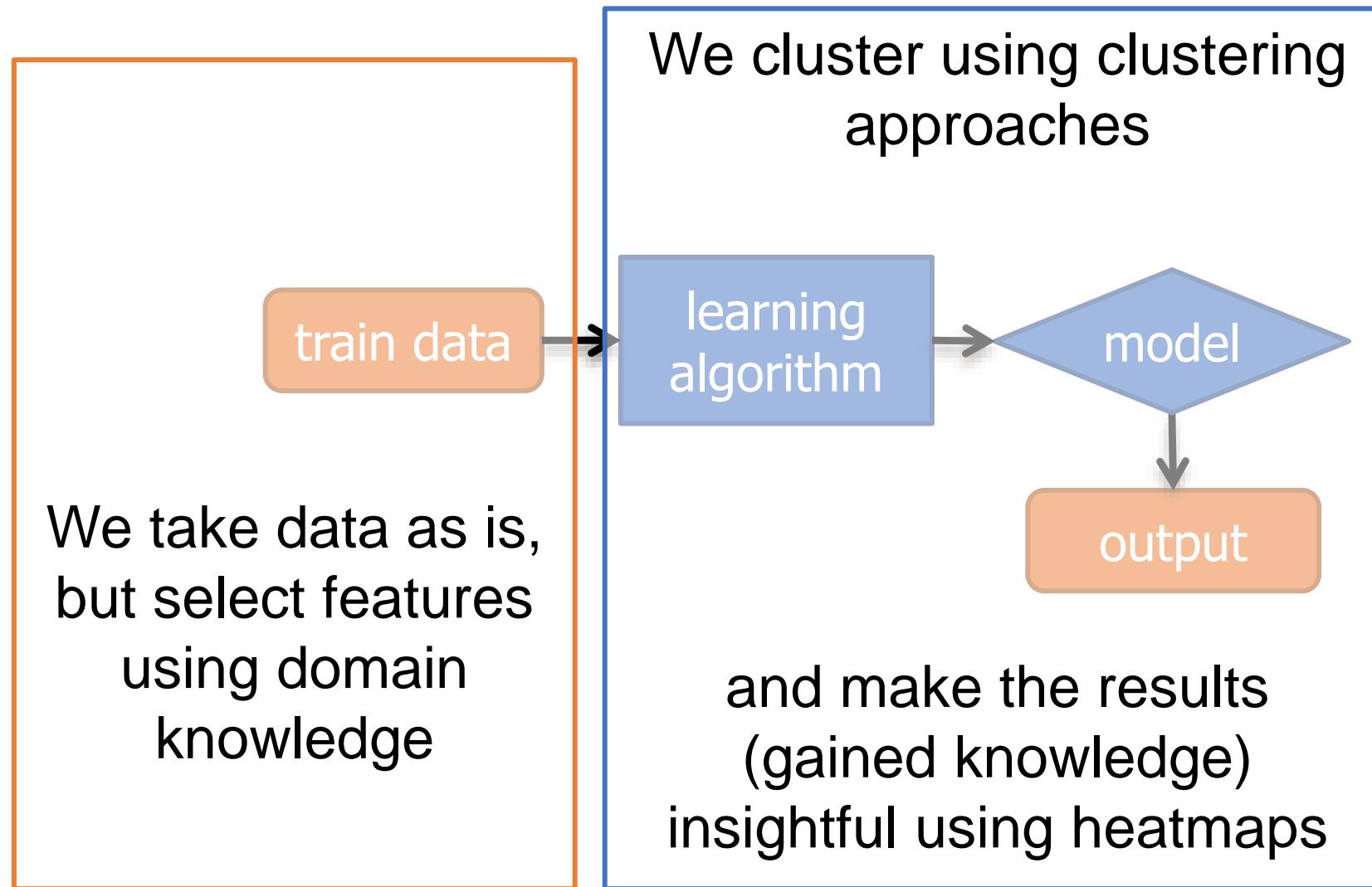
These are all Data Mining skills



These are all Data Mining skills



These are all Data Mining skills



Why is it hard ?

Volume

Velocity

Variety

Why is it hard ?

Volume

Velocity

Variety



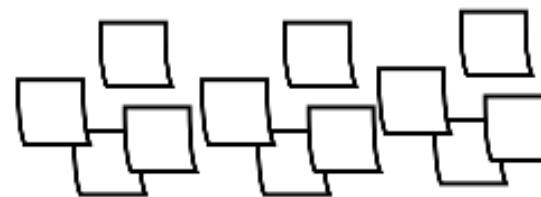
Need
scalable
algorithms

Why is it hard ?

Volume

Velocity

Variety



Need
approximate
yet accurate
estimates

Why is it hard ?

Volume

Velocity

Variety



Adapt to
different
datatypes,
distributions,
..

COURSE CONTENT

Distances, Matrices, Counting

Distances

- Similarity
- Metrics
- Computation
- DTW
- Text Embeddings
- Graph Embeddings

Matrices

- Representation
- Properties
- Operations
- Factorization
- Decompositions
- Dimensionality red.

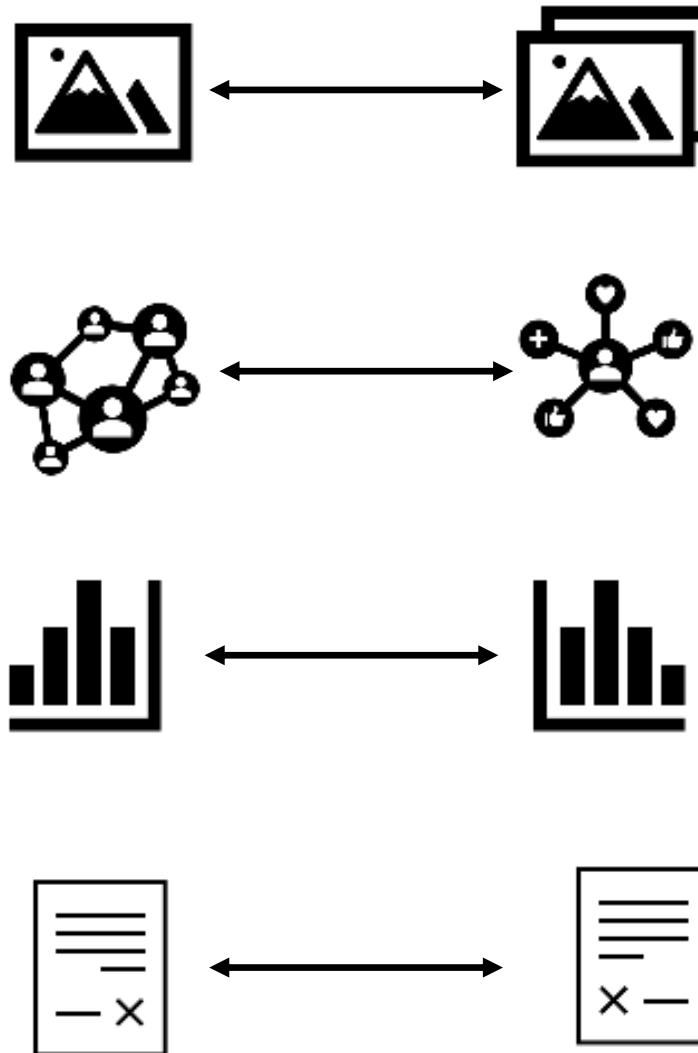
Counting

- Hashing
- Clustering
- Anomaly detection
- Sketching

Distances, Matrices, Counting

Distances

- Similarity
- Metrics
- Computation
- DTW
- Text Embeddings
- Graph Embeddings

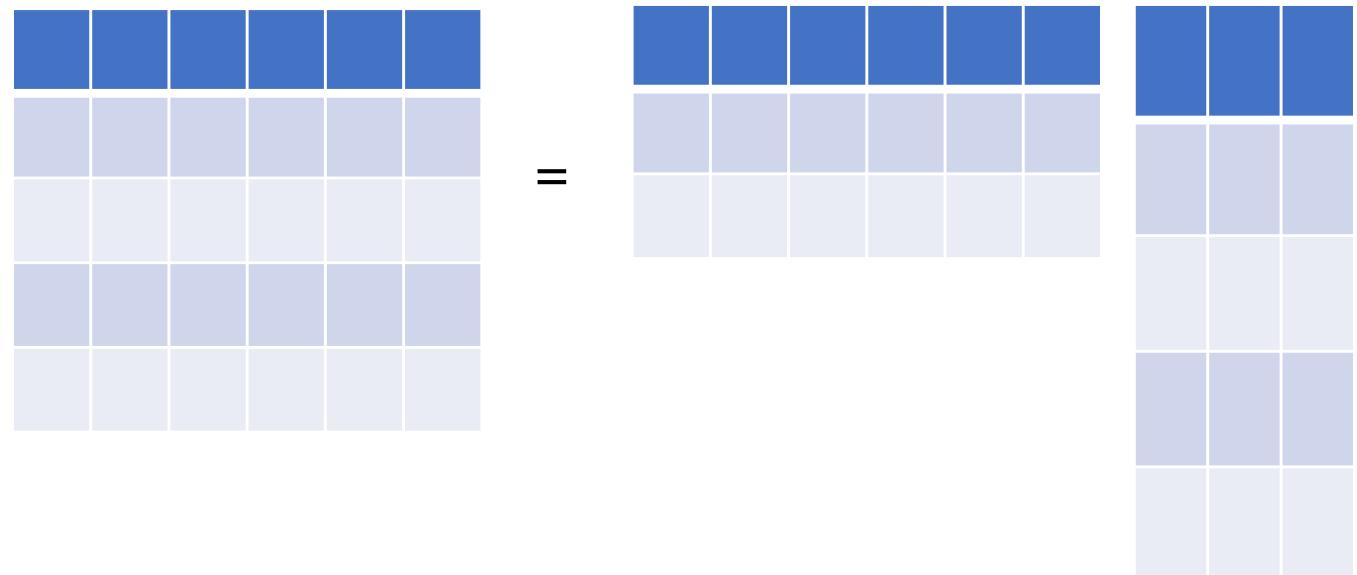


Distances, Matrices, Counting

Matrices

- Representation
- Properties
- Operations
- Factorization
- Decompositions
- Dimensionality
red.

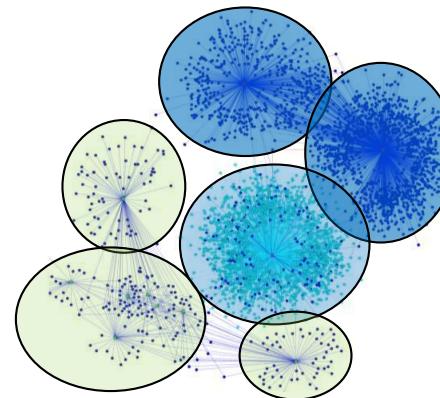
data →



Distances, Matrices, Counting

Counting

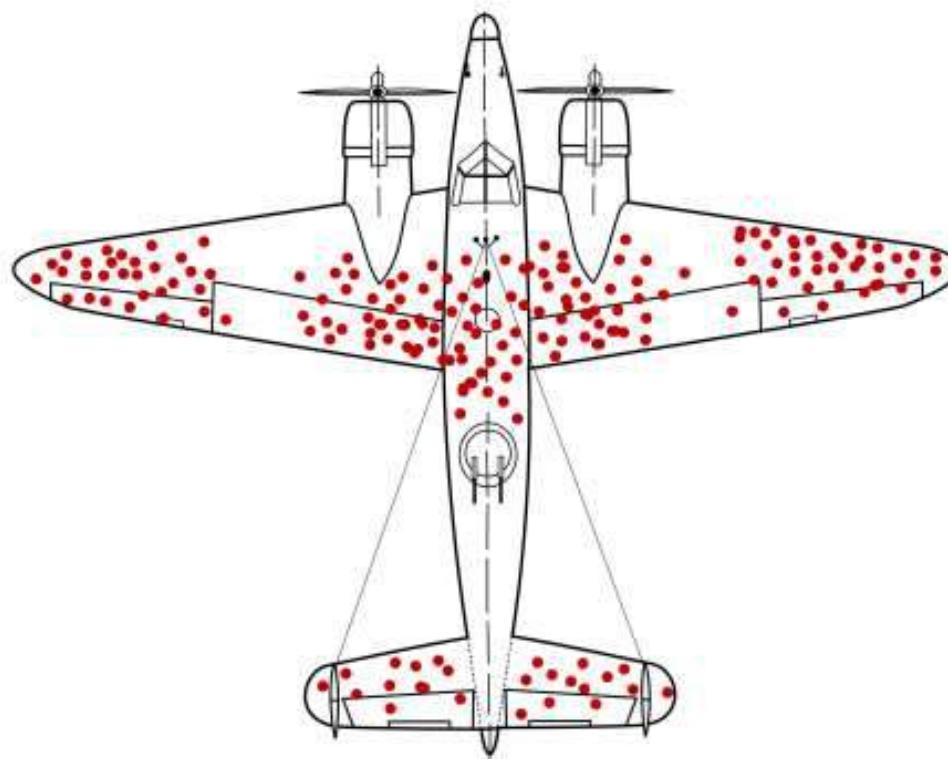
- Hashing
- Clustering
- Anomaly detection
- Sketching



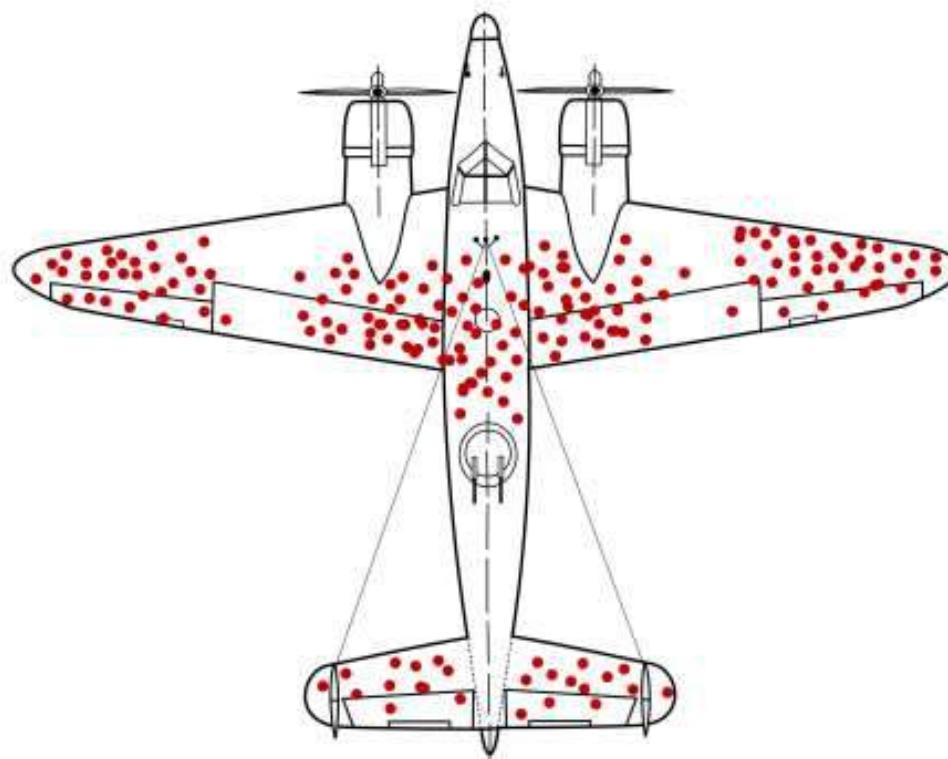
A word of caution

Don't fool yourself

Where should you reinforce the airplane ?



Survivorship bias



The Clever Hans Effect

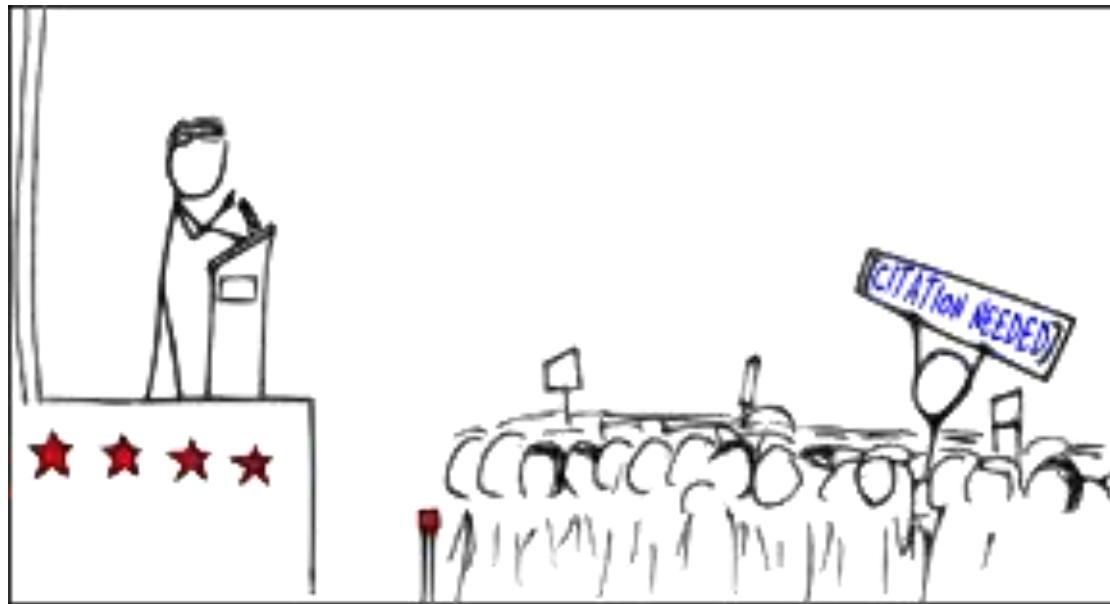


Citations in Wikipedia

The gunpowder store (Dutch: Kruithuis) was subsequently re-housed, a 'cannonball's distance away', outside the city, in a new building designed by architect [Pieter Post](#).^[14]



Citation needed in Wikipedia



... size or scope, and to pop up like a mushroom... (to appear unexpectedly and quickly). In reality, all species of mushrooms take several days to form primordial mushroom fruit bodies, though they do expand rapidly by the absorption of fluids. [citation needed]

Citations in Wikipedia

The gunpowder store (Dutch: Kruithuis) was subsequently re-housed, a 'cannonball's distance away', outside the city, in a new building designed by architect [Pieter Post](#). [\[14\]](#)

[Delft University of Technology](#) (TU Delft) is one of [four universities of technology](#) in the Netherlands. [\[23\]](#)

It was founded as an academy for civil engineering in 1842 by [King William II](#).

Today, well over 21,000 students are enrolled. [\[24\]](#)



How can we automatically provide citations for Wikipedia

- The neural network was trained, and was 100% accurate on the test set
- What should you be asking yourself ?

Its too good to be true

Citations in Wikipedia

The gunpowder store (Dutch: Kruithuis) was subsequently re-housed, a 'cannonball's distance away', outside the city, in a new building designed by architect [Pieter Post](#). [\[14\]](#)

[Delft University of Technology](#) (TU Delft) is one of [four universities of technology](#) in the Netherlands. [\[23\]](#)

It was founded as an academy for civil engineering in 1842 by [King William II](#).

Today, well over 21,000 students are enrolled. [\[24\]](#)

The gunpowder store (Dutch: Kruithuis) was subsequently re-housed, a 'cannonball's distance away', outside the city, in a new building designed by architect [Pieter Post](#).

[Delft University of Technology](#) (TU Delft) is one of [four universities of technology](#) in the Netherlands.

It was founded as an academy for civil engineering in 1842 by [King William II](#).

Today, well over 21,000 students are enrolled.

How can we automatically provide citations for Wikipedia

- The neural network was trained, and was 100% accurate on the test set
- When asked humans – they were < 55% accurate

Shortcuts in Learning

The gunpowder store (Dutch: Kruithuis) was subsequently re-housed, a 'cannonball's distance away', outside the city, in a new building designed by architect [Pieter Post](#). [\[14\]](#)

[Delft University of Technology](#) (TU Delft) is one of [four universities of technology](#) in the Netherlands. [\[23\]](#)

It was founded as an academy for civil engineer [William II](#).

If . ." " predict citation needed

Today, well over 21,000 students are enrolled. [\[24\]](#)

The gunpowder store (Dutch: Kruithuis) was subsequently re-housed, a 'cannonball's distance away', outside the city, in a new building designed by architect [Pieter Post](#).

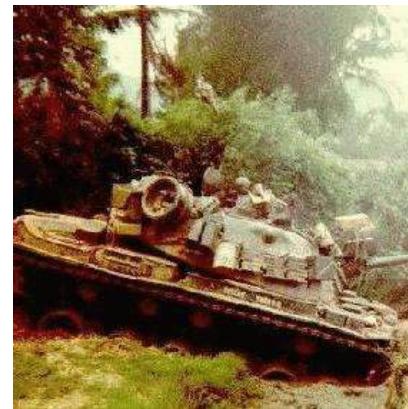
[Delft University of Technology](#) (TU Delft) is one of [four universities of technology](#) in the Netherlands.

It was founded as an academy for engineering in 1842 by King

Today, well over 21,000 students are enrolled.

Russian Tanks

- After much analysis, it turned out the network implemented the following:
 - The sky is blue if there is no tank on the horizon
 - When the sky is blue, there is no tank without otherwise there is



Shortcuts in image models



horse

Shortcuts in image models



If "copyright" predict horse

You can always find what you are looking for



JELLY BEANS
CAUSE ACNE!

SCIENTISTS!
INVESTIGATE!

BUT WE'RE
PLAYING
MINECRAFT!
... FINE.



WE FOUND NO
LINK BETWEEN
JELLY BEANS AND
ACNE ($P > 0.05$).



THAT SETTLES THAT.

I HEAR IT'S ONLY
A CERTAIN COLOR
THAT CAUSES IT.



WE FOUND NO
LINK BETWEEN
GREY JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
TAN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
CYAN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND A
LINK BETWEEN
MAUVE JELLY
BEANS AND ACNE
($P < 0.05$).



WE FOUND NO
LINK BETWEEN
GREEN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
ORANGE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
PURPLE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BROWN JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
PINK JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
BLUE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
TEAL JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
SALMON JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
RED JELLY
BEANS AND ACNE
($P > 0.05$).



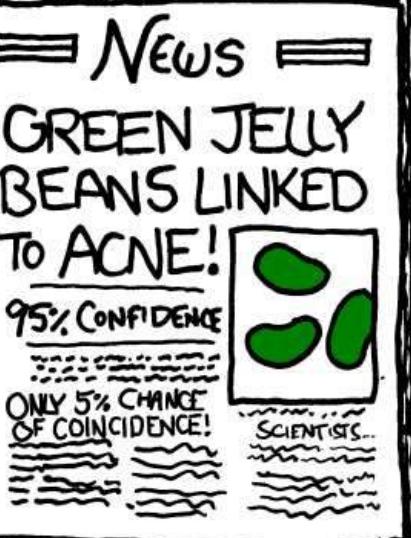
WE FOUND NO
LINK BETWEEN
TURQUOISE JELLY
BEANS AND ACNE
($P > 0.05$).



WE FOUND NO
LINK BETWEEN
MAGENTA JELLY
BEANS AND ACNE
($P > 0.05$).

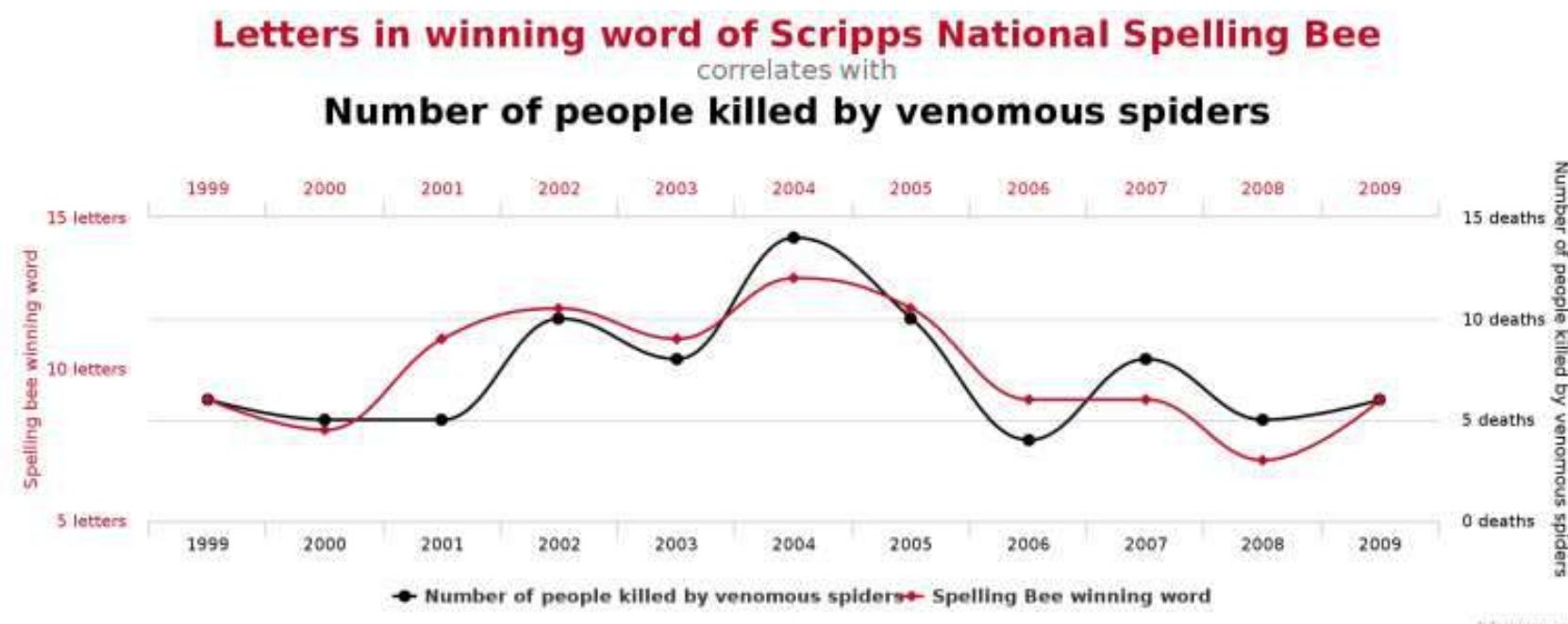


WE FOUND NO
LINK BETWEEN
YELLOW JELLY
BEANS AND ACNE
($P > 0.05$).



Multiple Comparisons Problem

- *The cause of many errors in data mining...*



https://youtu.be/HpjlcEH4zuY?si=3-KnR06RHipLk_UK

Summary

- **Data Mining:** Extract useful information from a large volume of data and transform it into an understandable structure
- Labs 30% (3 of them), Final exam 70%, Homeworks as examples for finals (6 of them)
- Content organization: Distances, Matrices, Counting
- Data mining as the skill of the 21st century
- But tread with caution

CSE2525 Anomaly detection

Anomaly detection challenge



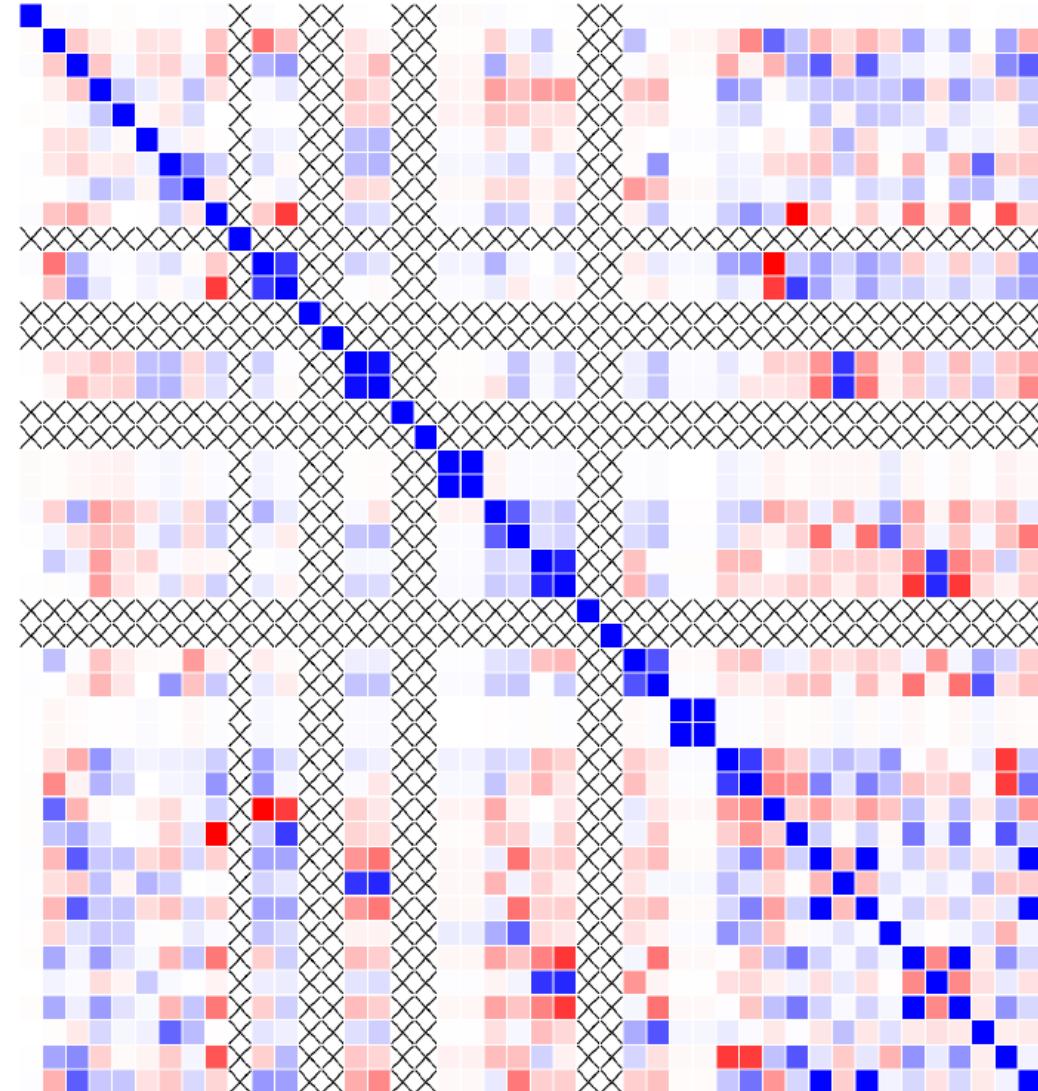
Sensor and actuator data
Highly discrete and predictable

Investigating data

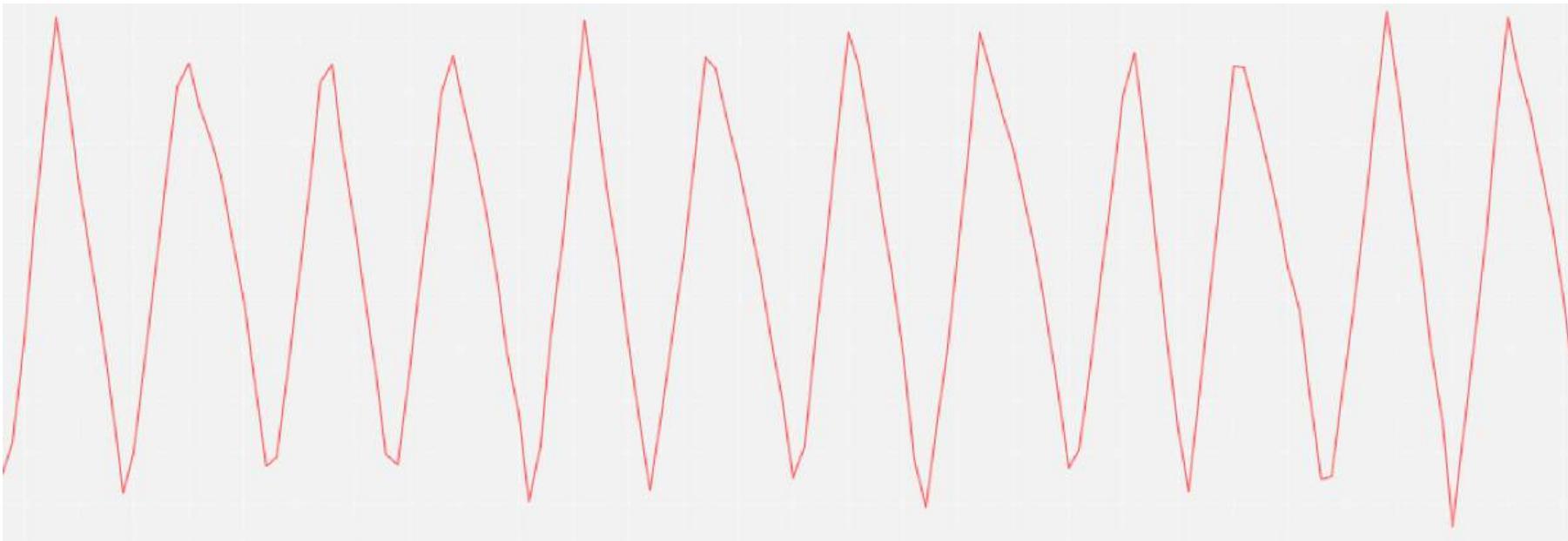
- Make plots and tables that show:
 - Dependence between features
 - *shows what kind of feature processing to use*
 - Dependence over time
 - *shows the type of temporal processing to use*
 - The difficulty of the problem
 - *shows what technique might be suitable*

Correlation between signals

- Red: negative
- Blue: positive
- What does it tell us?

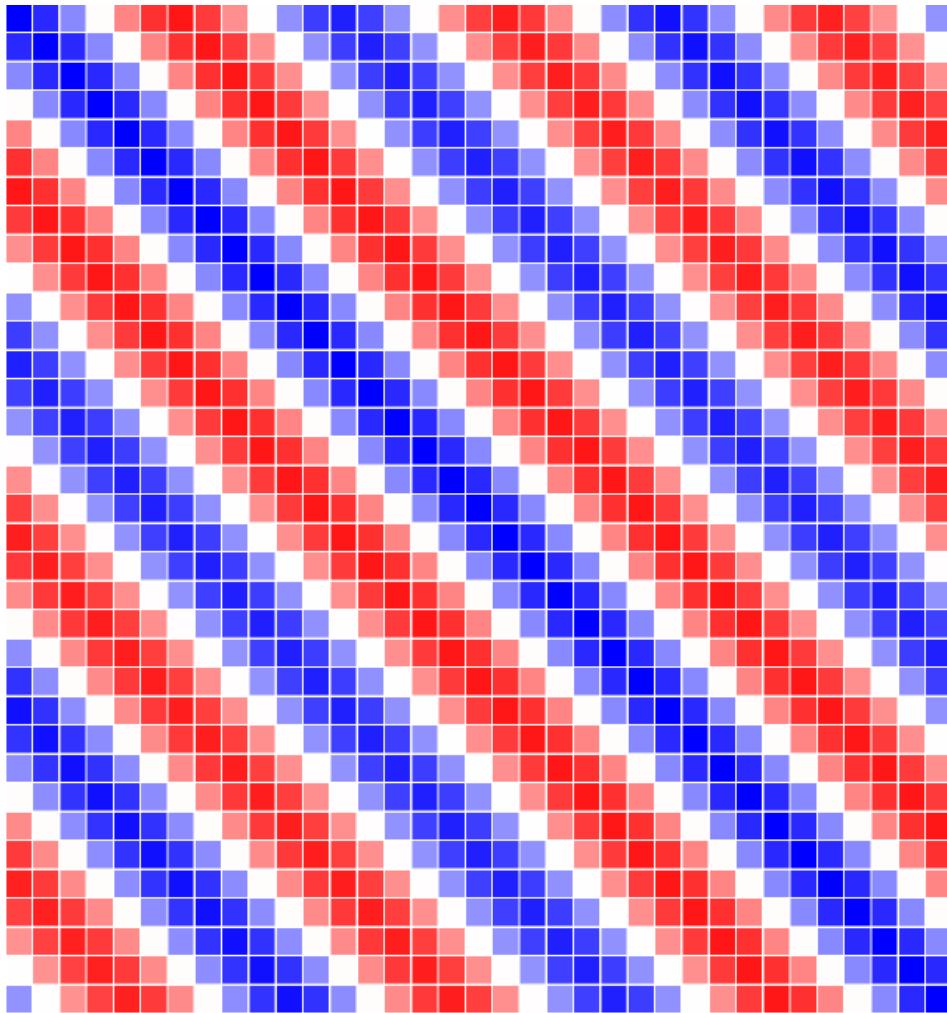


One signal plotted over time



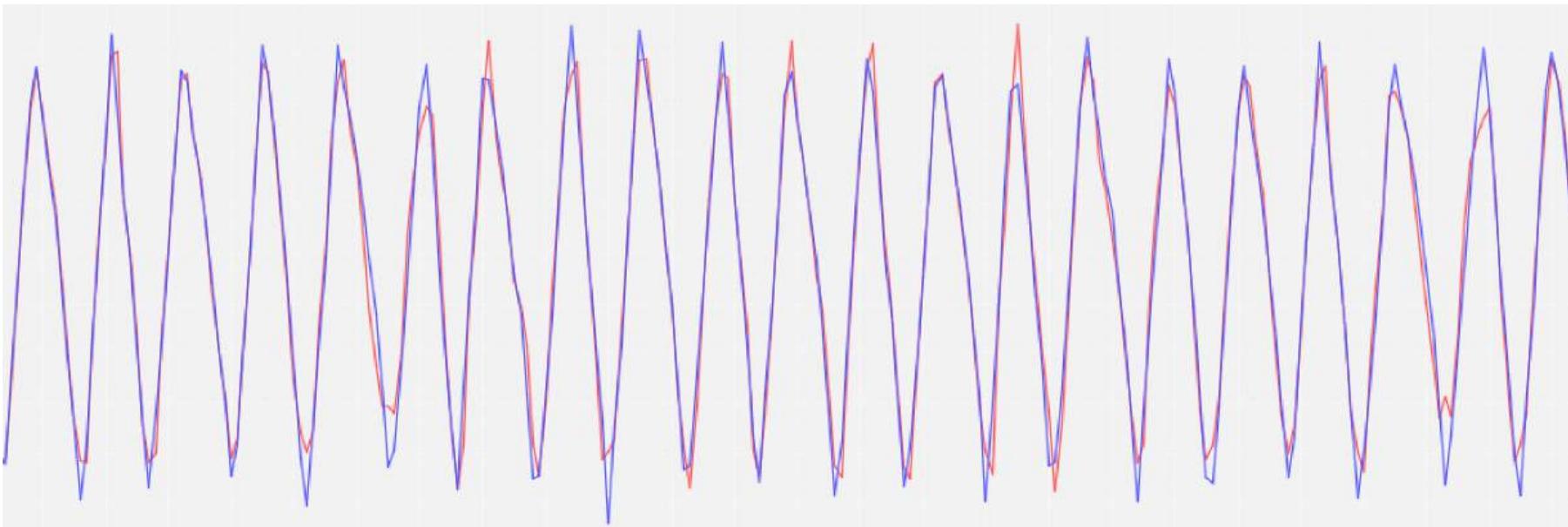
- Shows what?

A signal's self-correlation



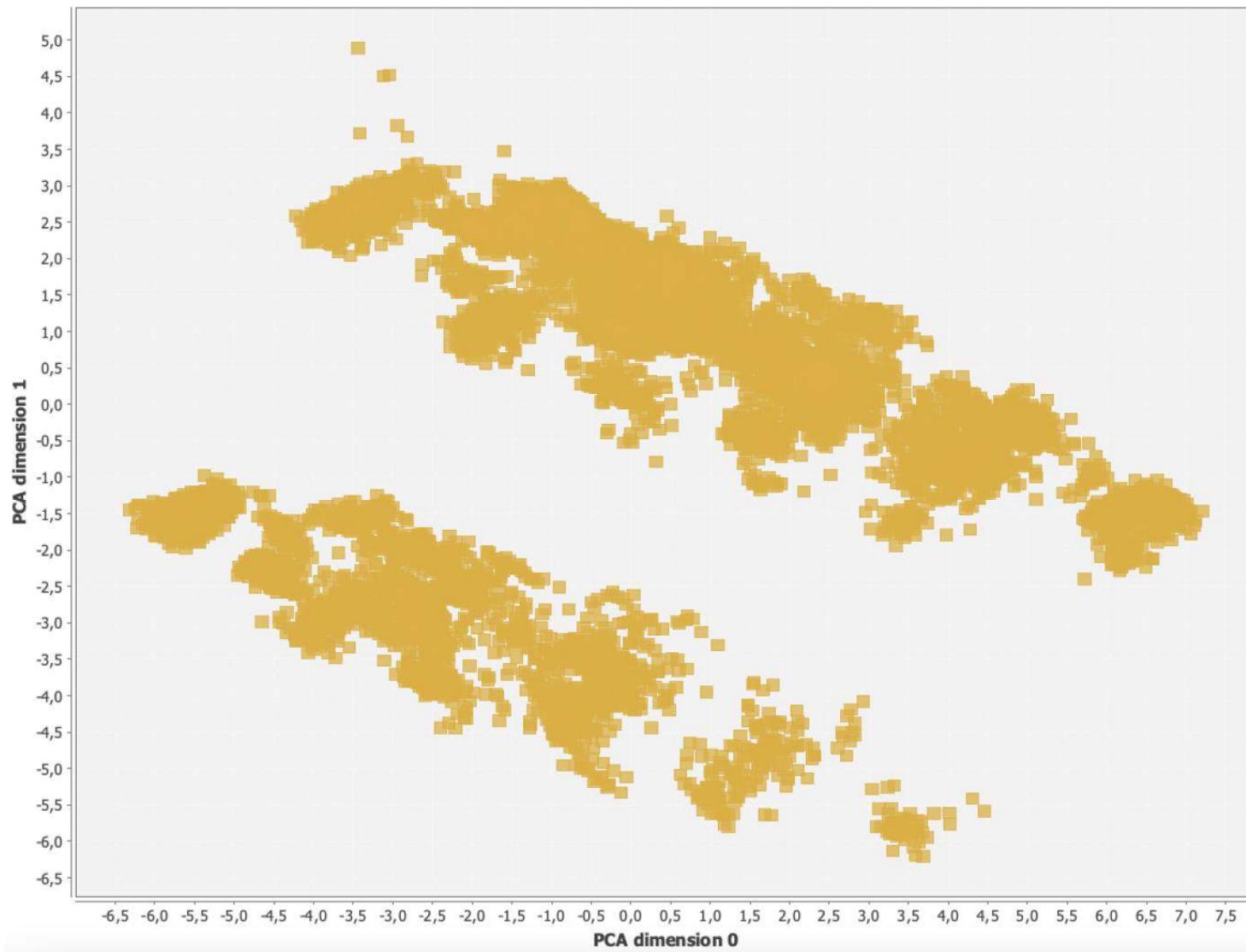
- Red: negative
- Blue: positive
- What to conclude?

Predict using linear regression

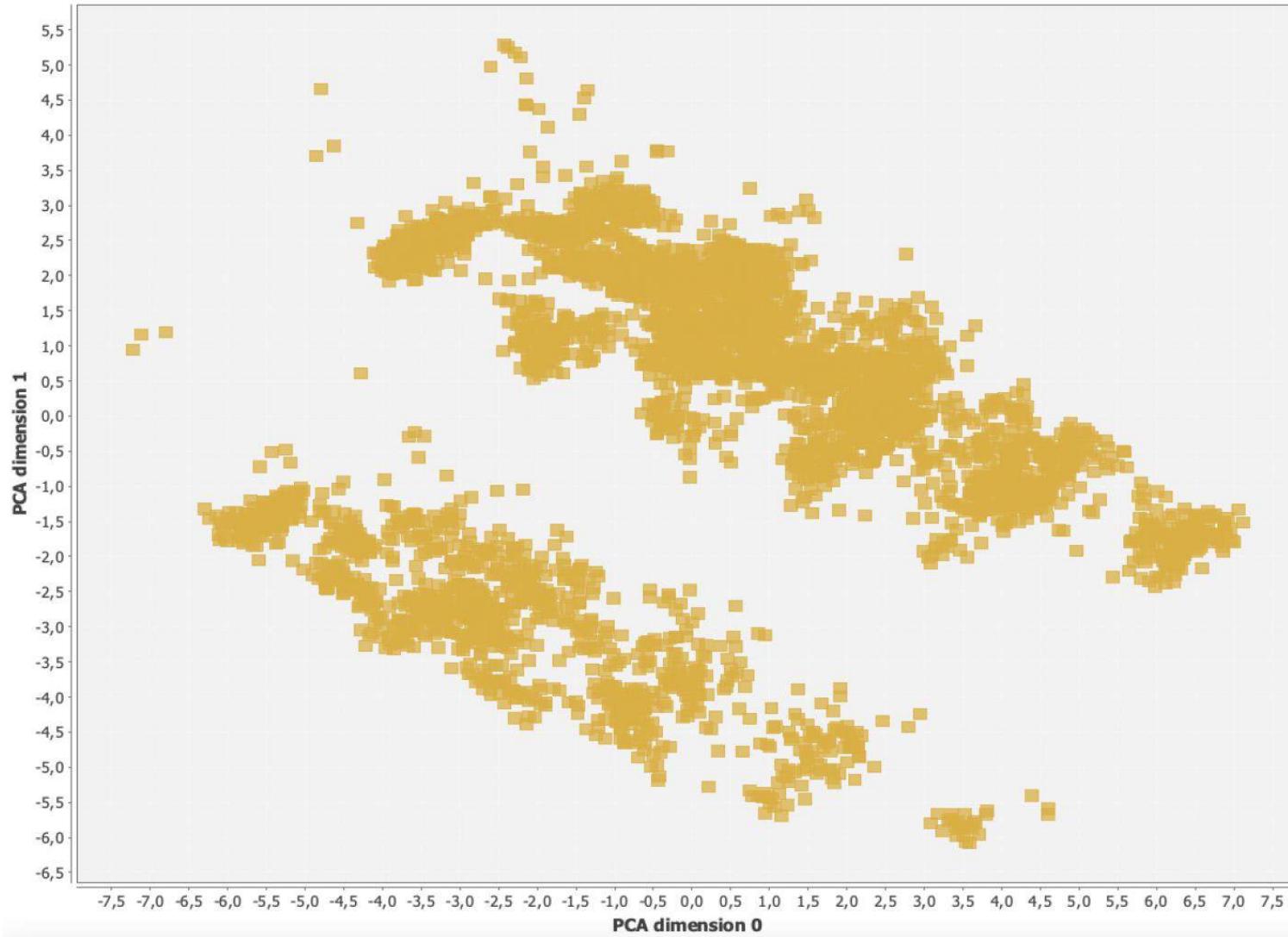


- Seems not too hard...

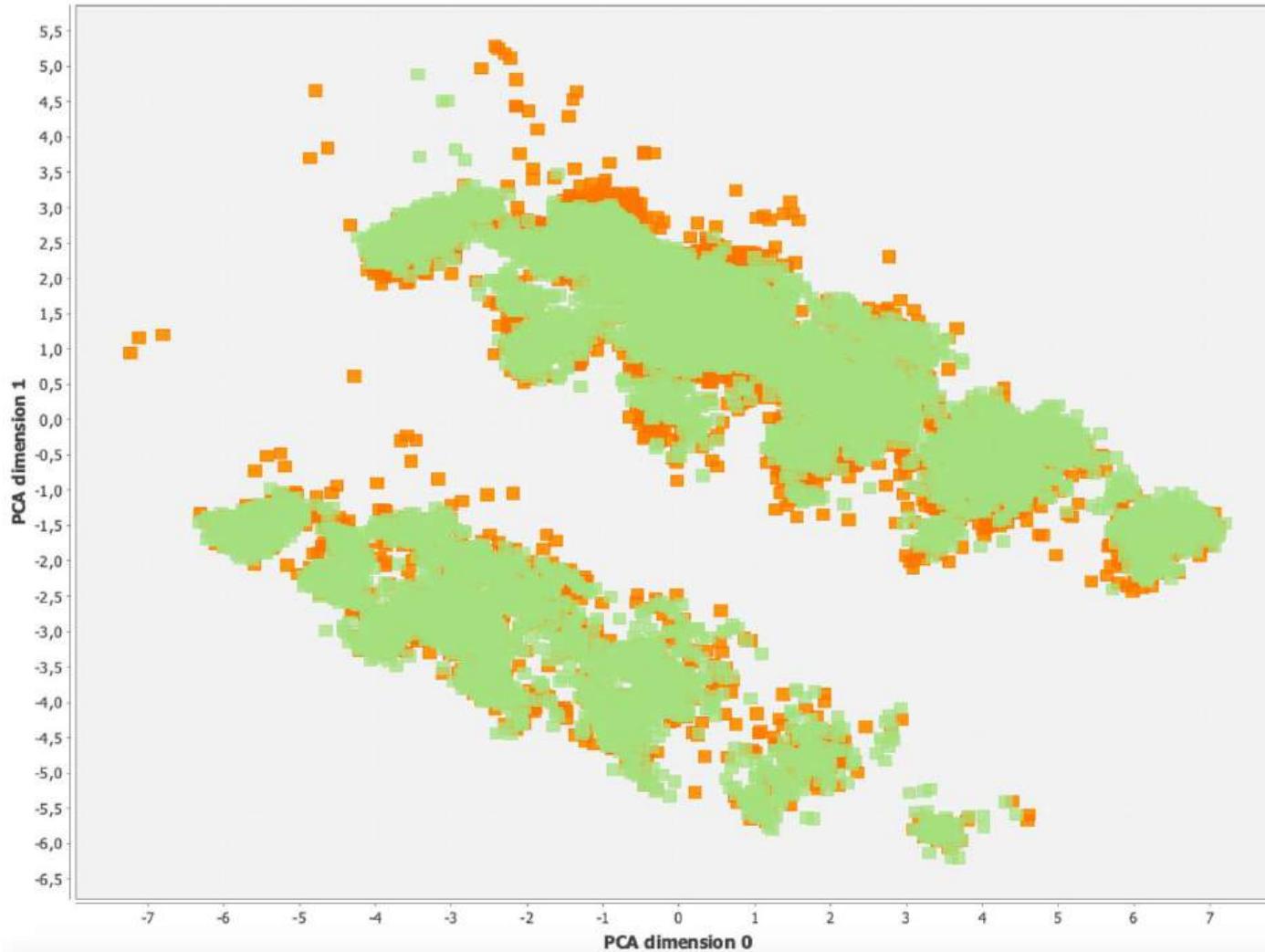
Embed into two dimensions



Embed also the test set!



Notice differences? Are they anomalies?



Wait a minute... Did I cheat?

- I looked at the test data!

Wait a minute... Did I cheat?

- I looked at the test data!
- Big question:
 - Are the test data available when learning a model?
 - Some people say yes, some say no ... it depends ...
- Traditional anomaly detection methods learn only from test data
- Outlier detection aims to find data points that are different
- In the lab, we have test data available, you may use it, but never ever use the test labels during model training

Wait a minute... Did I cheat?

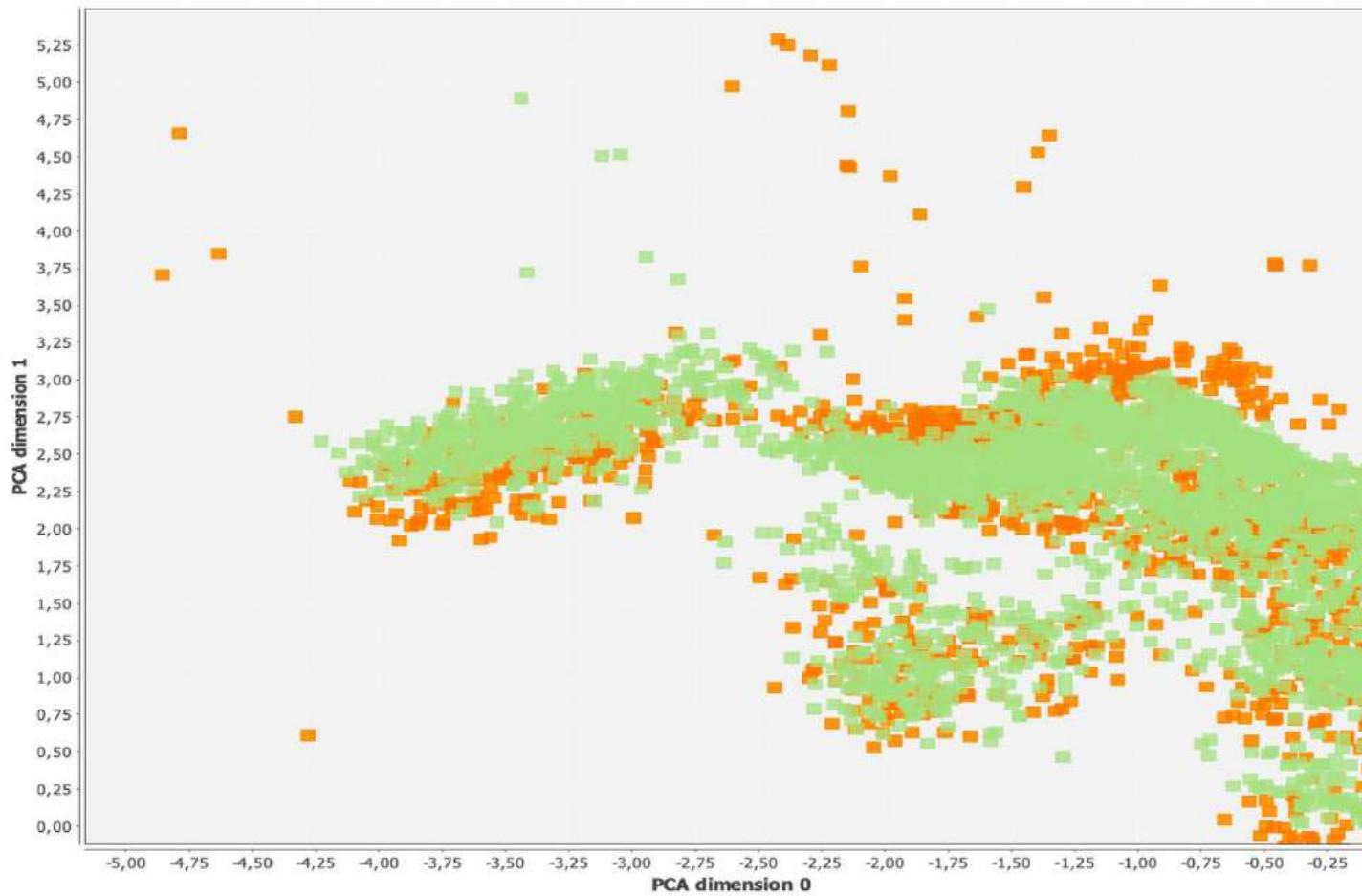
- I looked at the test data!
- Big question:
 - Are the test data available when learning a model?
 - Some people say yes, some say no ... it depends ...
- Traditional anomaly detection method:
 1. Team up
 2. Investigate data
 3. Transform if needed
 4. Learn models
 5. Detect deviations
 6. Raise alarms
- Outlier detection aims to find data points that are significantly different from others.
- In the lab, we have test data available, but we never ever use the test labels during training.

Today

- 3 types of anomalies
 - Point
 - Contextual
 - Collective
- Modeling context
 - Sliding Windows
 - Time Warping distance
- Popular anomaly detection methods
 - Classification
 - Nearest Neighbor
 - Spectral
 - Deep Learning

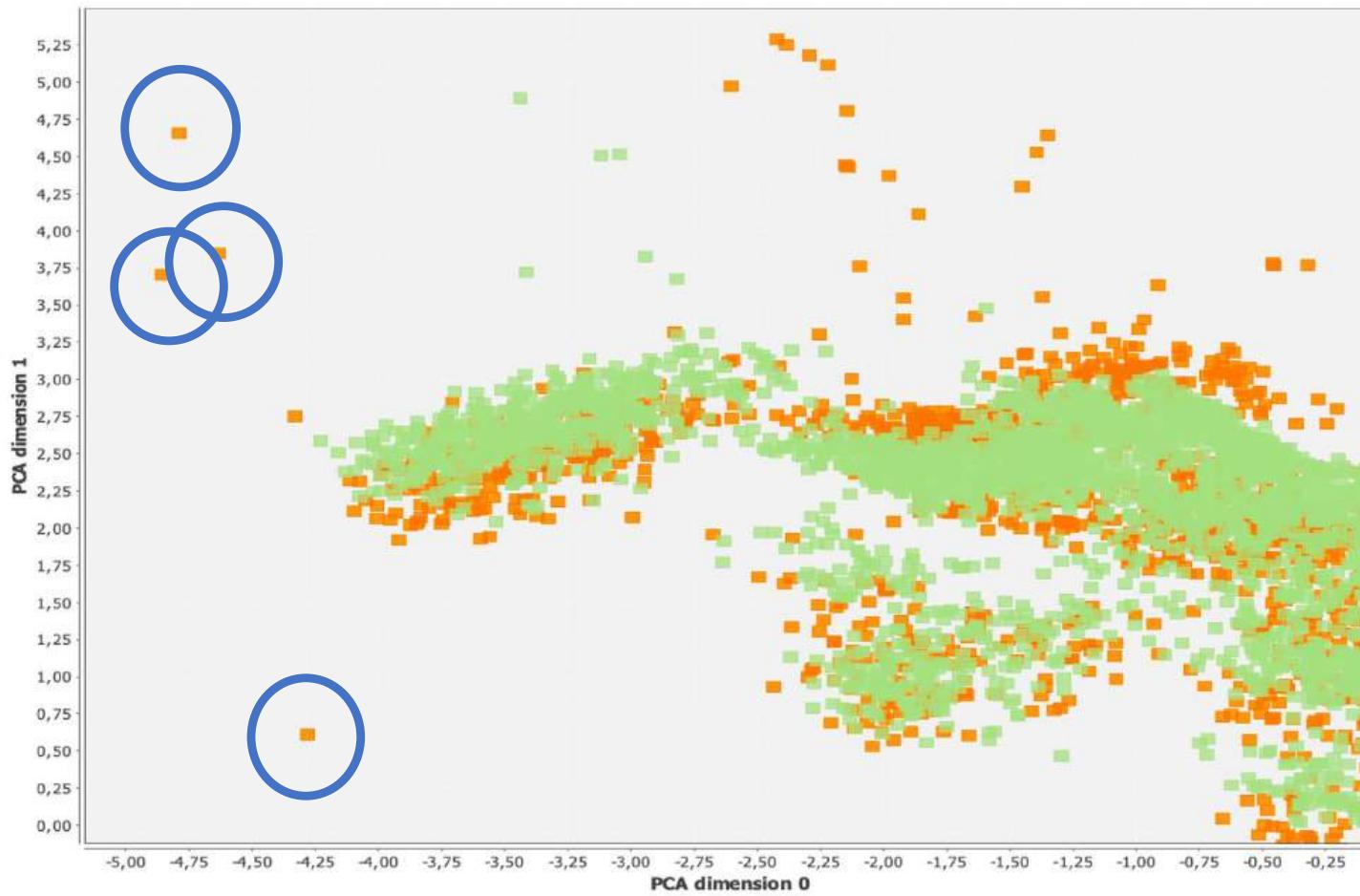
Point Anomalies

- An individual data instance is anomalous w.r.t. the data



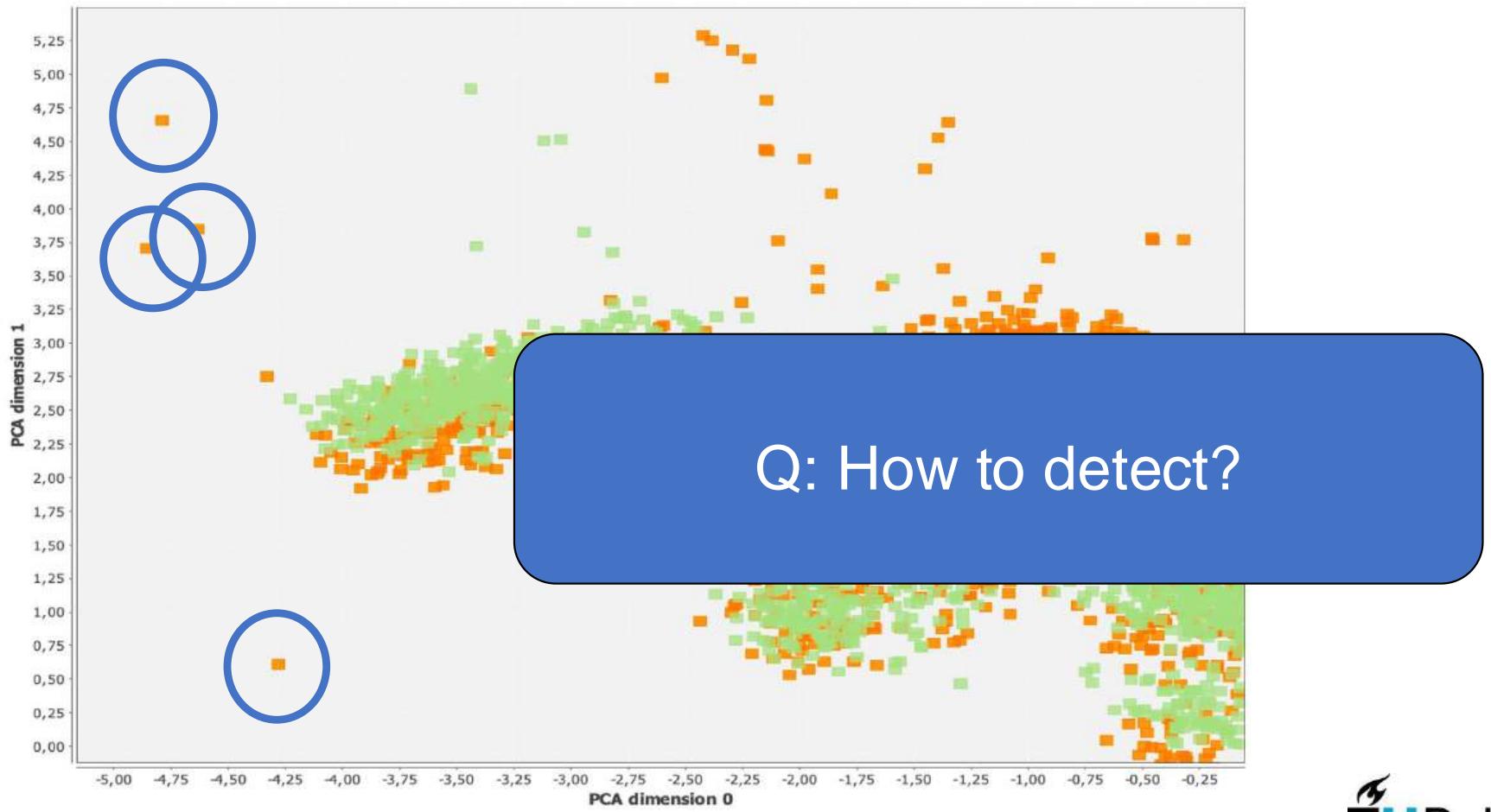
Point Anomalies

- An individual data instance is anomalous w.r.t. the data



Point Anomalies

- An individual data instance is anomalous w.r.t. the data

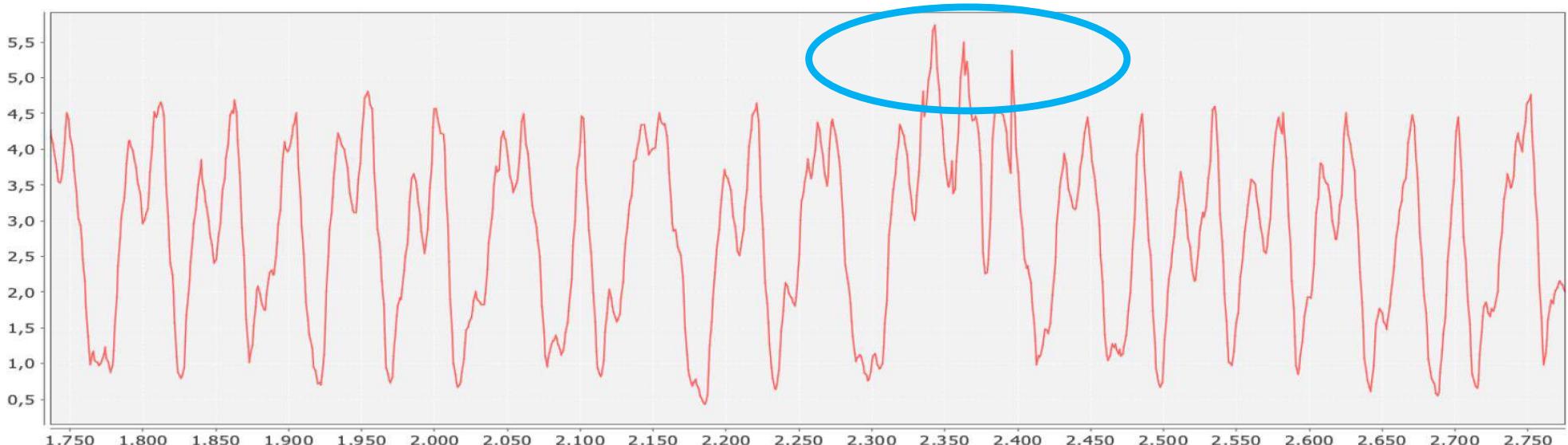


Detecting point anomalies

- Classification
 - learn from unlabeled data (one-class classification)
 - tests whether the point lies in an empty part of the input space
- Distance-based
 - tests whether the distance to the nearest neighbors is normal
- Data reconstruction
 - tests whether each feature value is normal given all other feature values
- In Lab 1, you will implement distance-based anomaly detection, and anomaly detection based on data reconstruction

Contextual Anomalies

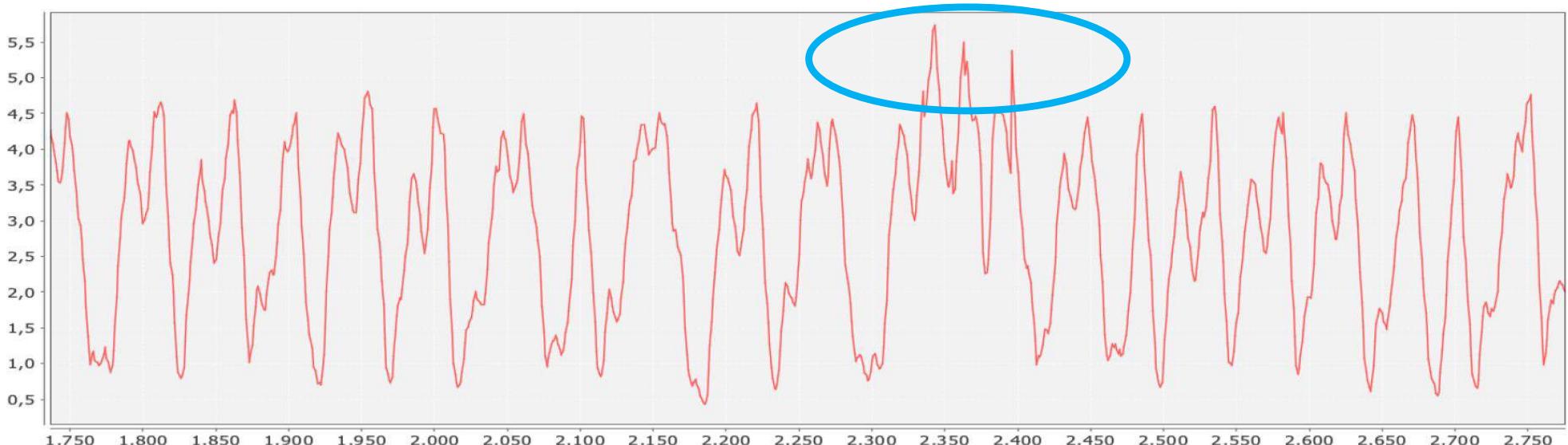
- An individual data instance is anomalous within a context
- *Context are other data points!*
 - surrounding data in time or space, but can be in other features



- Are these contextual anomalies?

Contextual Anomalies

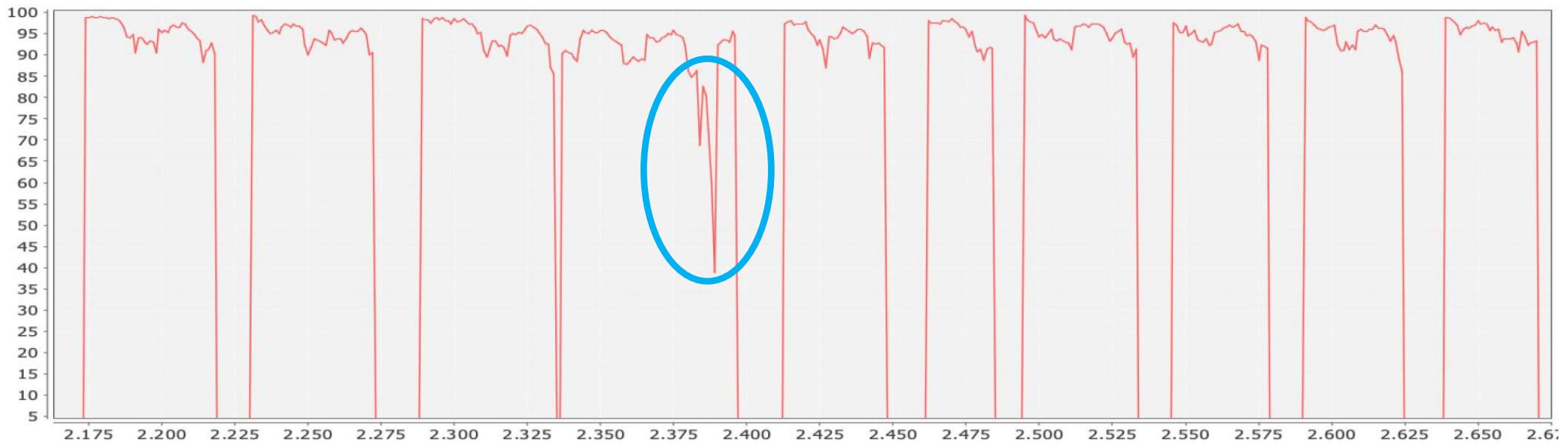
- An individual data instance is anomalous within a context
- *Context are other data points!*
 - surrounding data in time or space, but can be in other features



- No, they are out of normal range, thus point-anomalies

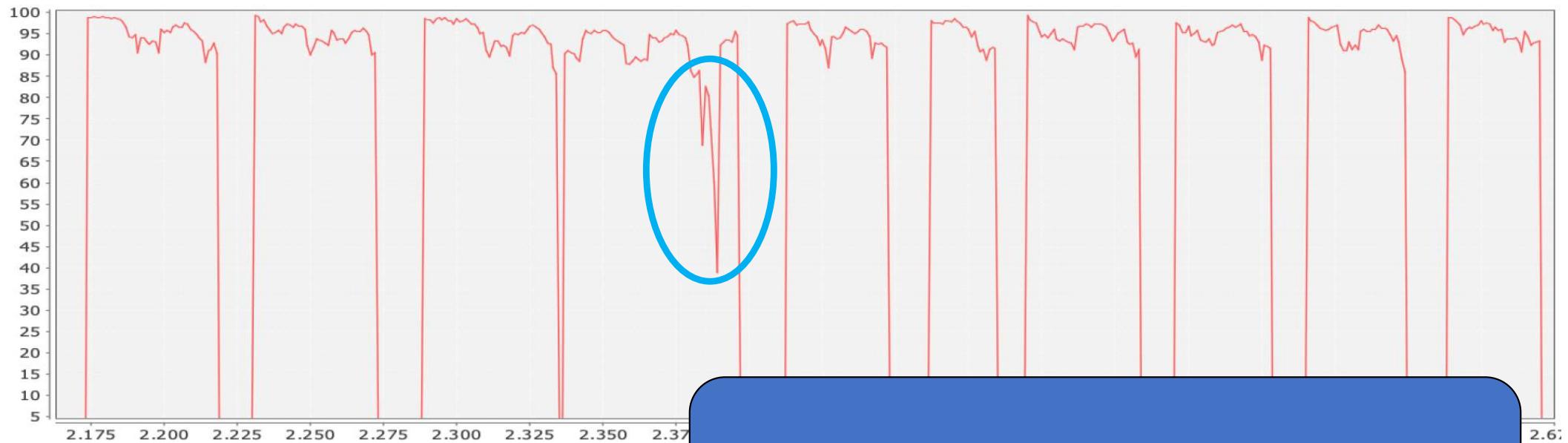
Contextual anomalies

- *Normal data points, but strange given the surrounding data*



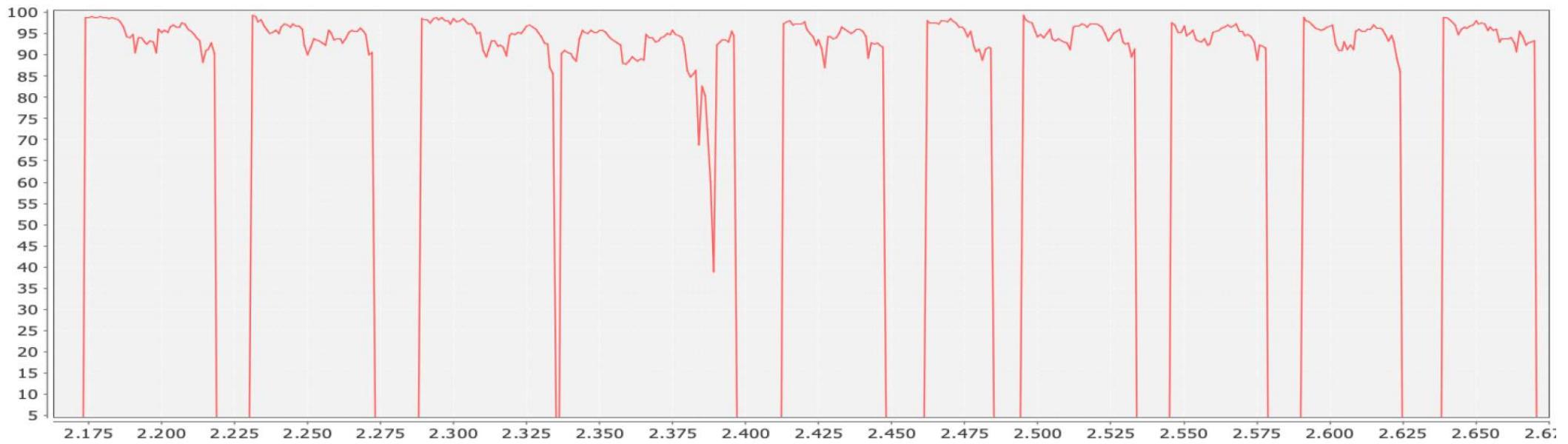
Contextual anomalies

- *Normal data points, but strange given the surrounding data*

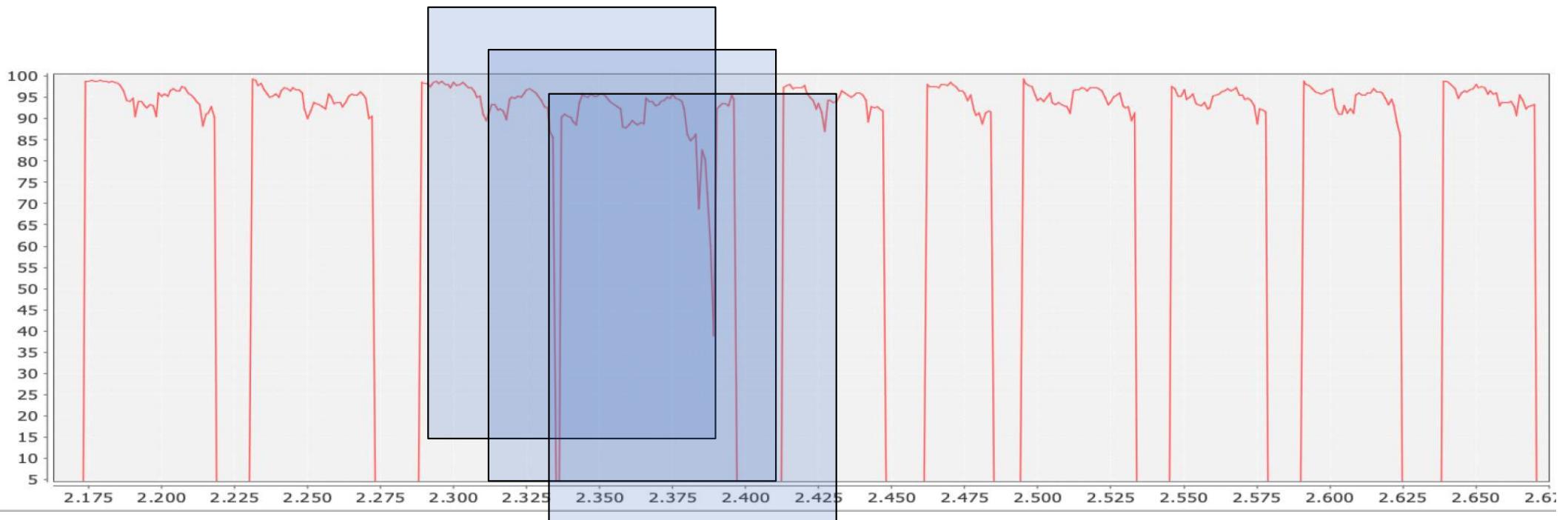


Q: How to detect?

What to use as context C?



What to use as context C?



The typical approach is to use sliding windows, and predict x_t from $x_{t-n} \dots x_{t-1}$

Does this detect the anomaly?

- I used sliding windows of length 5, predicting the last point using linear regression



Compute the residual

- Use sliding windows from training data to learn a model f for predicting the next value:

$$y_k = f(y_{k-1}) + \epsilon$$

- Compute the **expected next value**

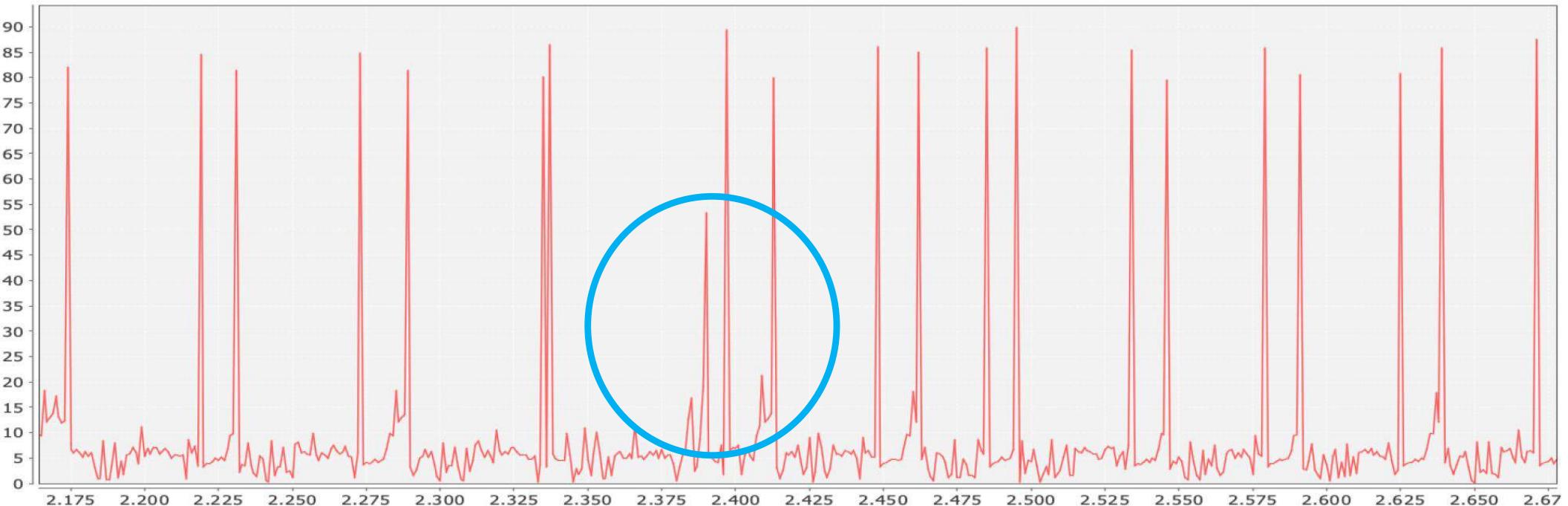
$$\hat{y}_{k|k-1} = f(y_{k-1})$$

- Evaluate the **residual** using the real next value

$$r_k = y_k - \hat{y}_{k|k-1}$$

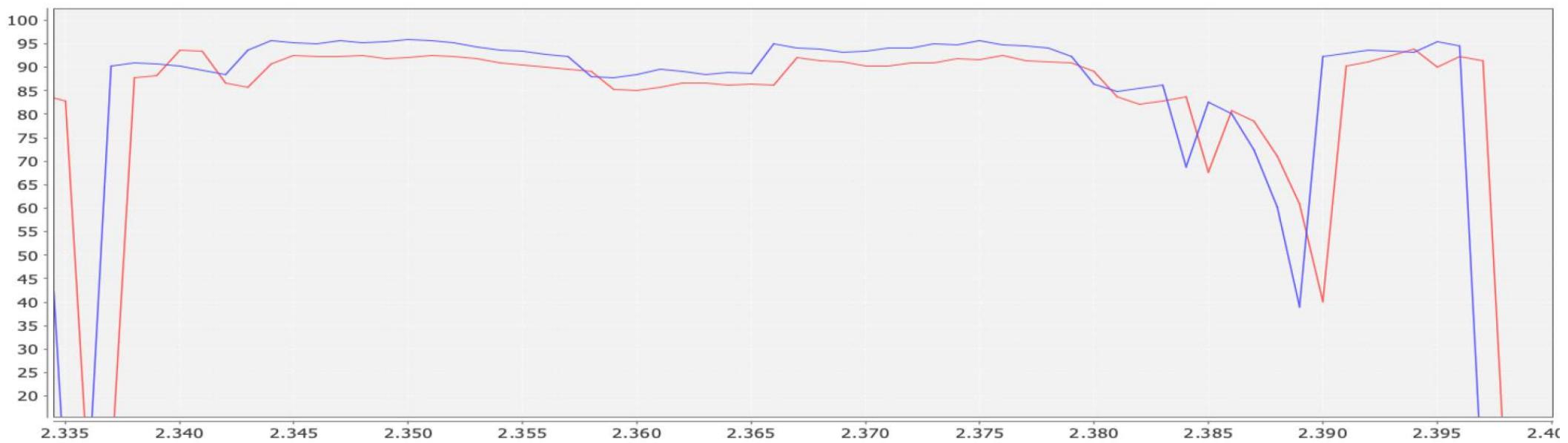
- Use a decision threshold, typical:
 - 2 or 3 times the standard error
 - or simply sort on residual error and return largest ones

The residual



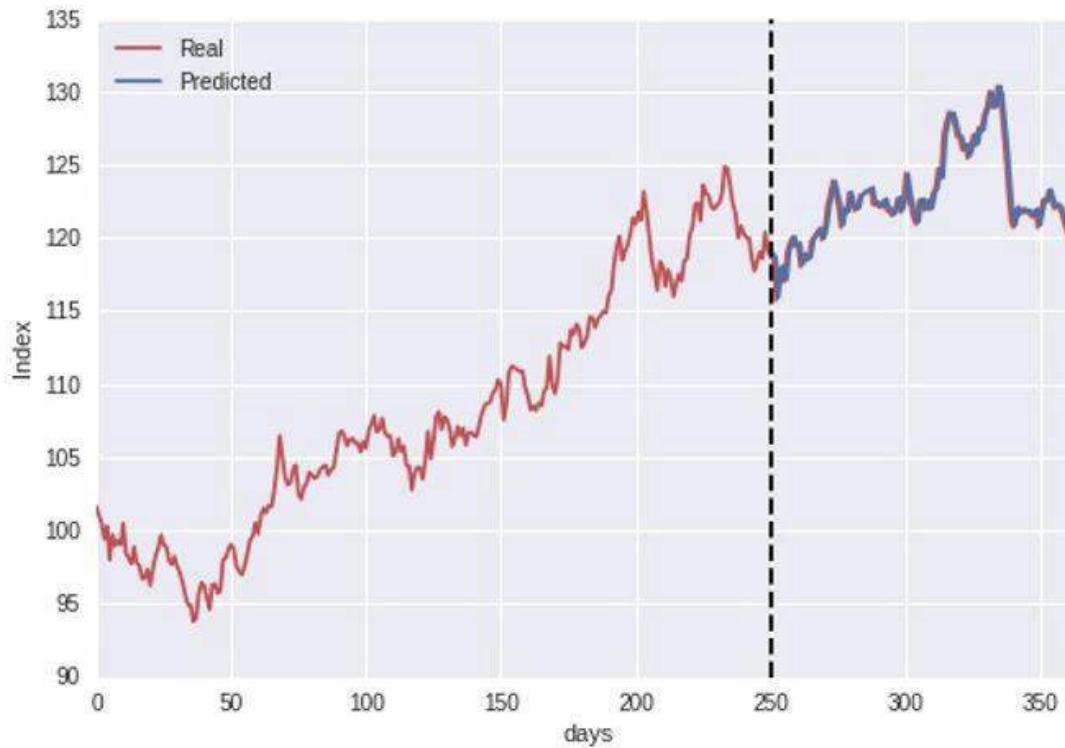
- No threshold would detect this, or give many false positives

Zoomed in



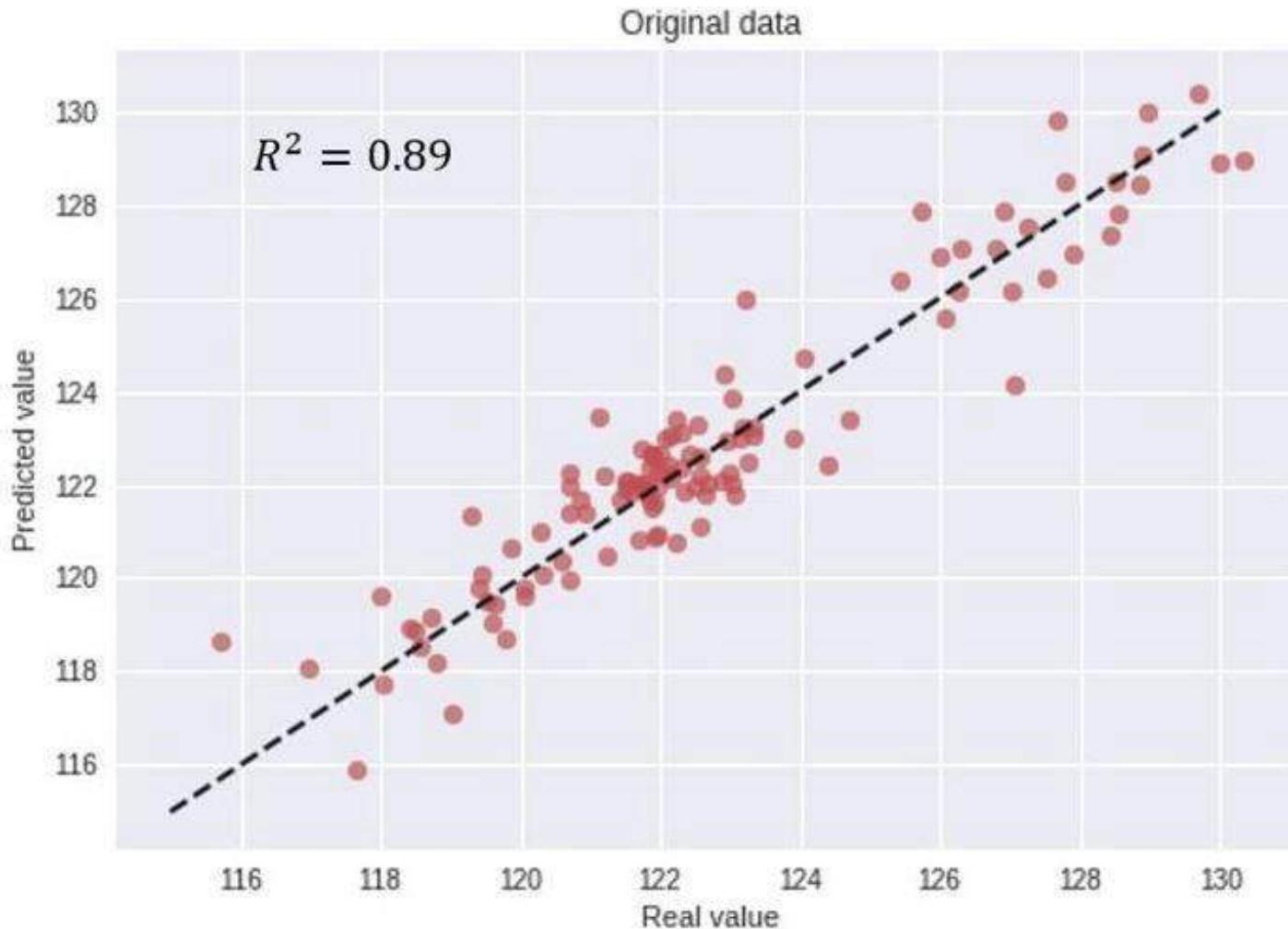
What is going on?

Pitfall: predictions



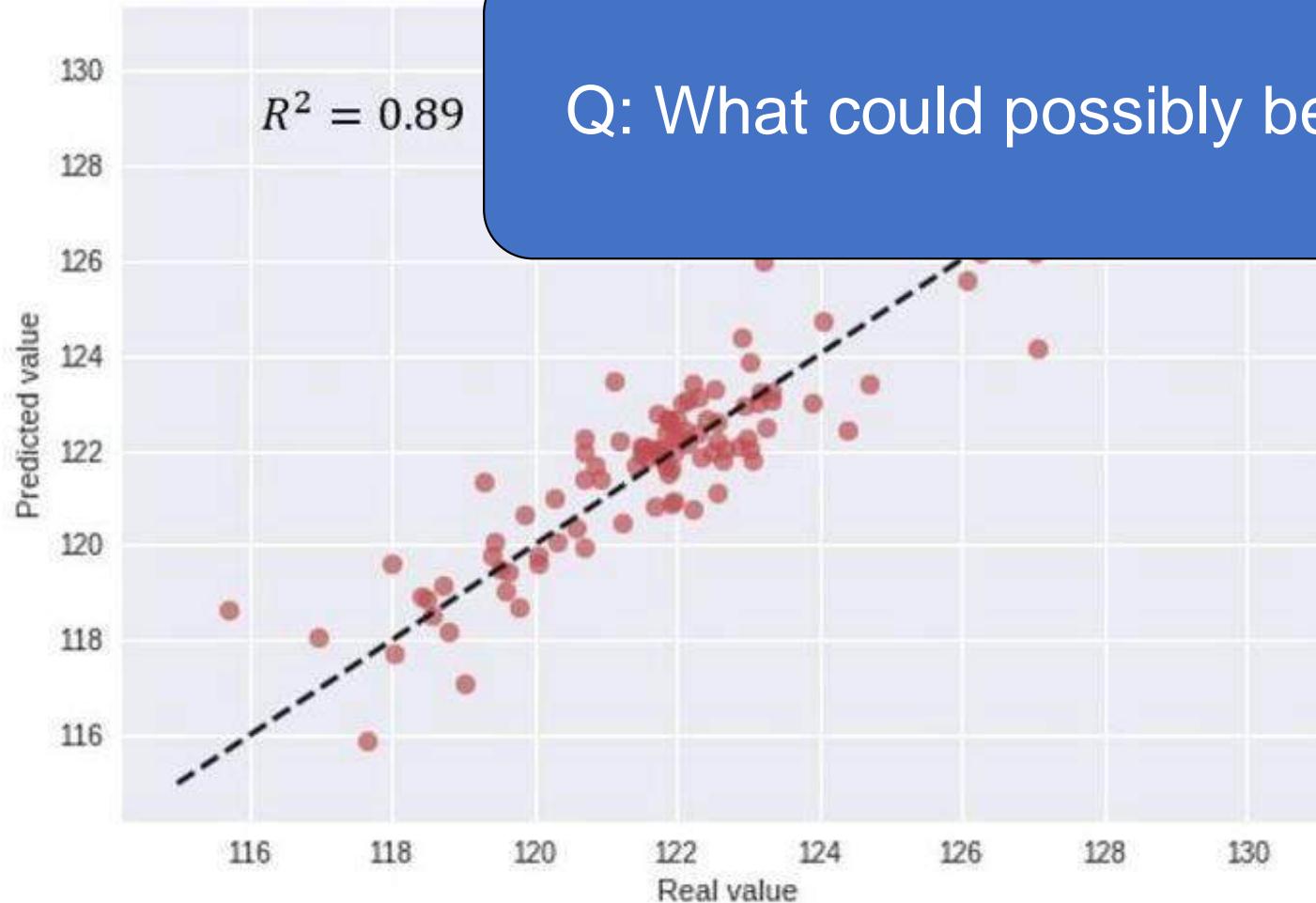
from <https://www.kdnuggets.com/2019/05/machine-learning-time-series-forecasting.html>

Pitfall: predictions



from <https://www.kdnuggets.com/2019/05/machine-learning-time-series-forecasting.html>

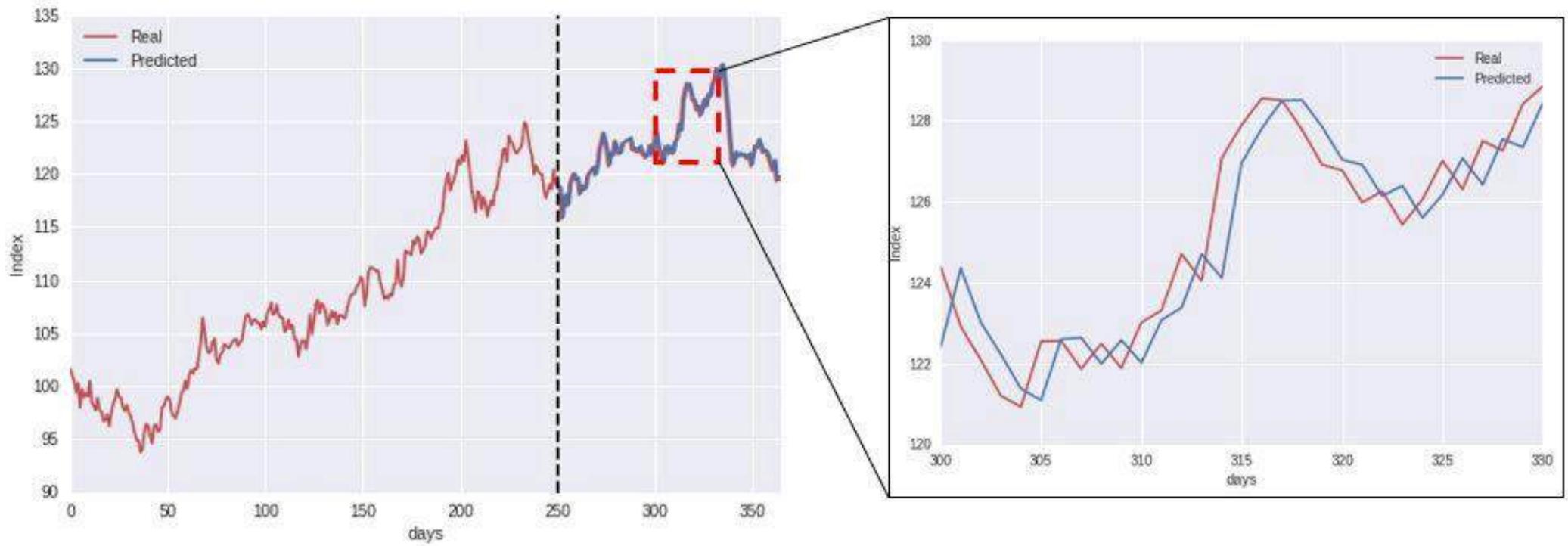
Pitfall: predictions



Q: What could possibly be wrong?

from <https://www.kdnuggets.com/2019/05/machine-learning-time-series-forecasting.html>

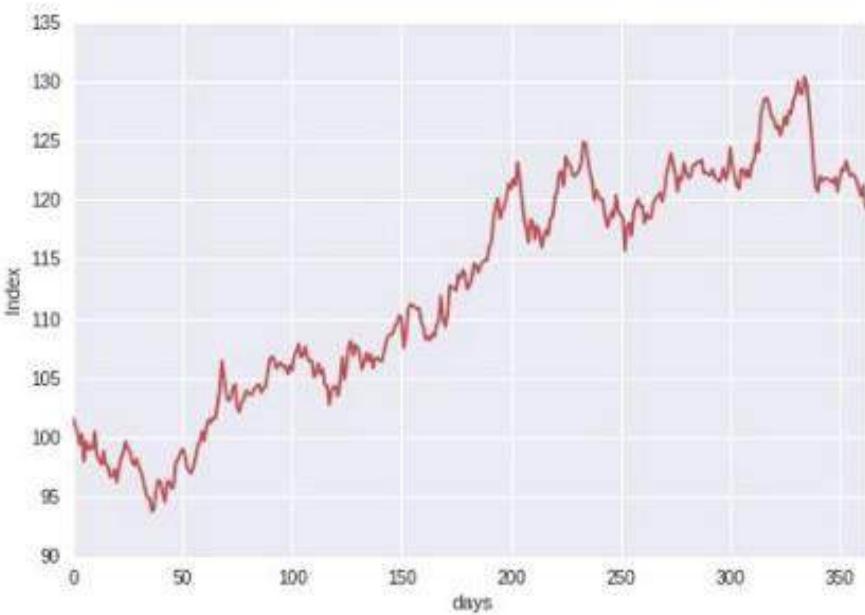
Results: zoomed in – persistance!



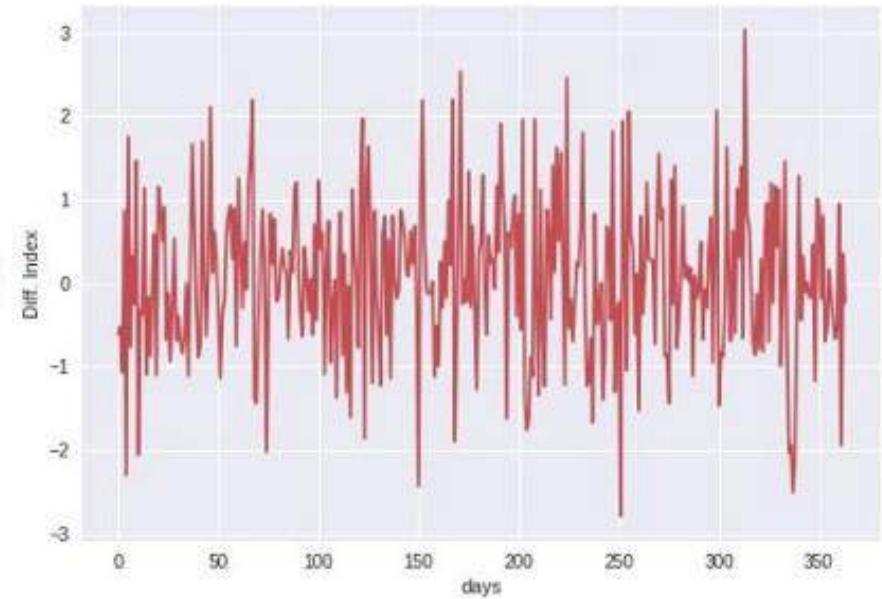
from <https://www.kdnuggets.com/2019/05/machine-learning-time-series-forecasting.html>

Pitfall: temporal correlation

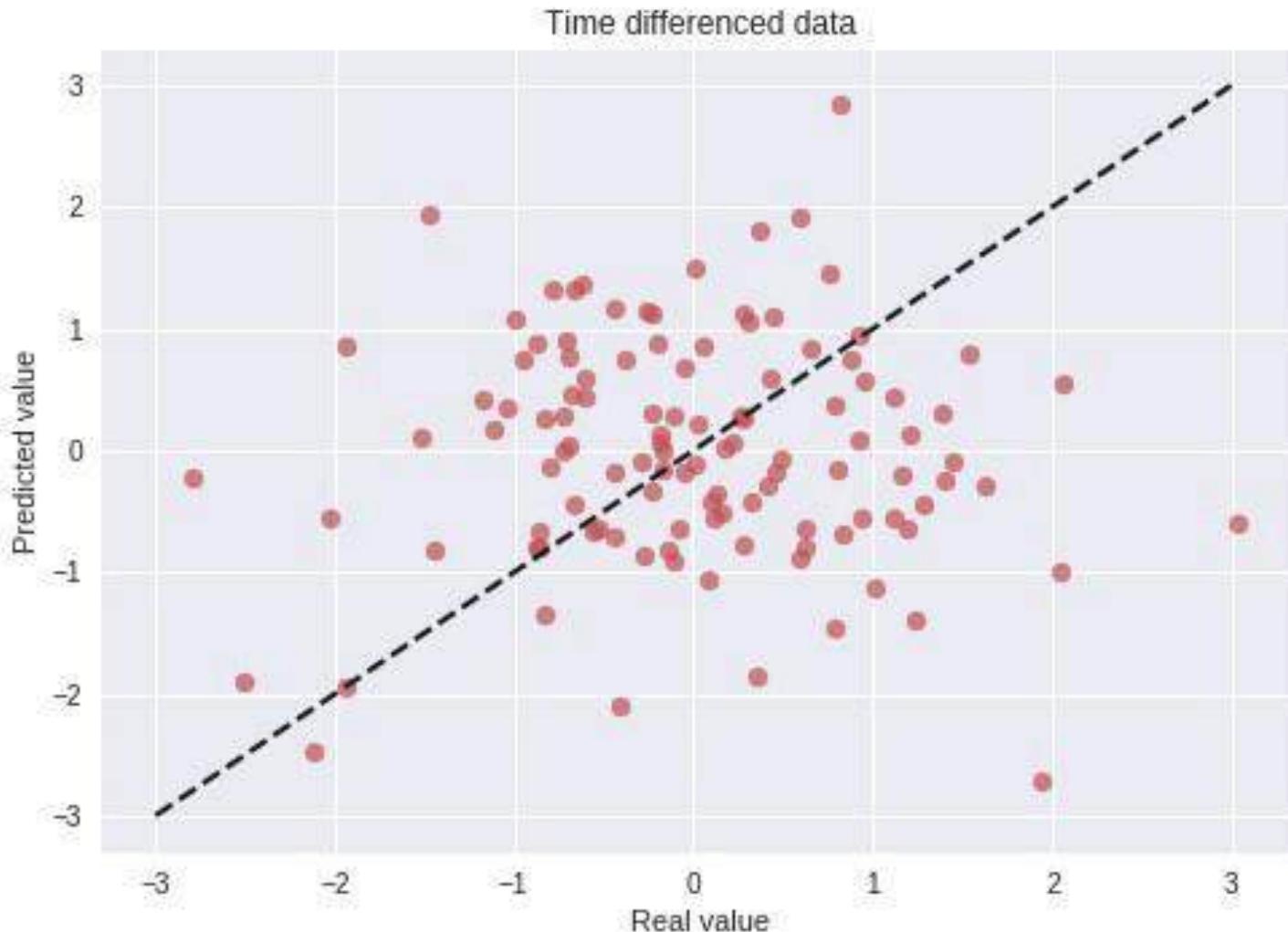
- x_t is quite likely close to x_{t-1}
- Solution:
 - Differencing - $x_t := x_t - x_{t-1}$



Time differencing
→

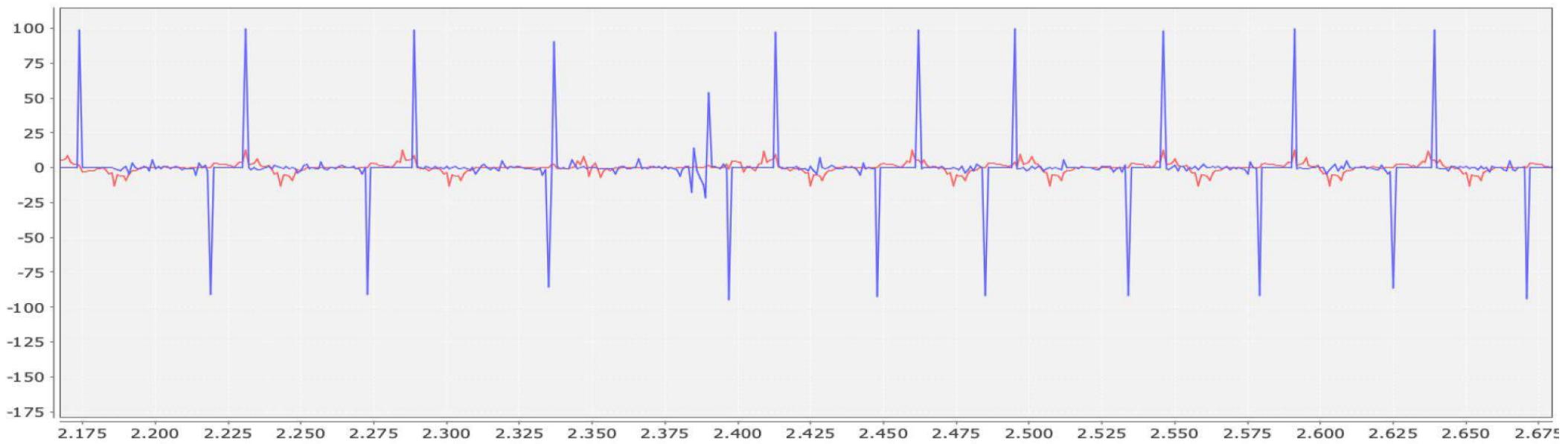


Results: not better than random (in fact, the data are random!)



So, what about our data?

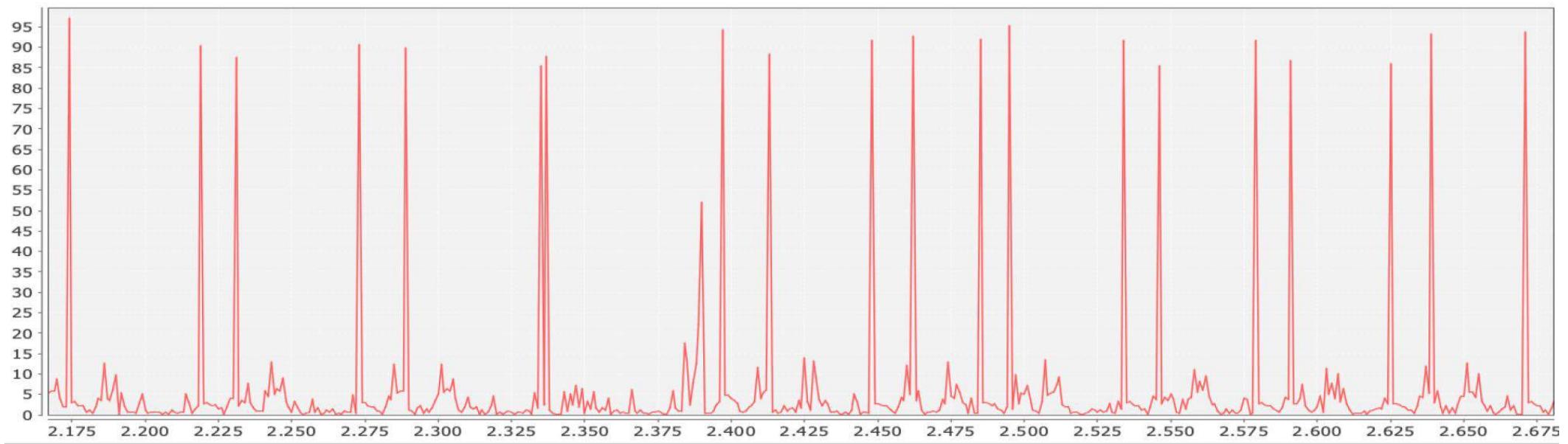
- Set sliding window a bit longer, length 20



- Although regular, predicting when a peak occurs is hard

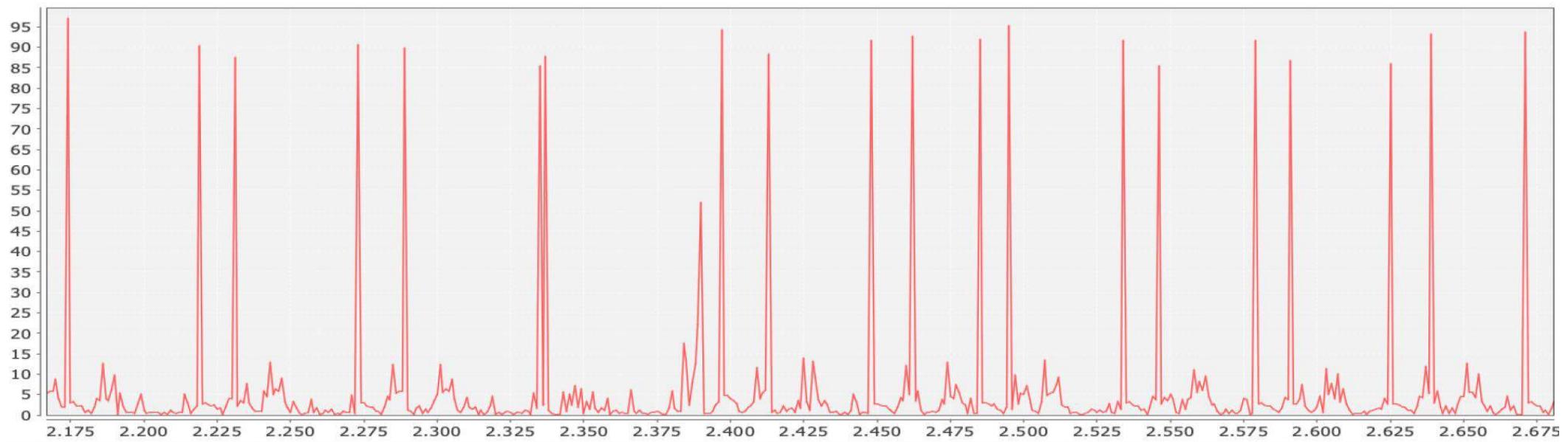
So, what about our data?

- This shows the absolute errors



So, what about our data?

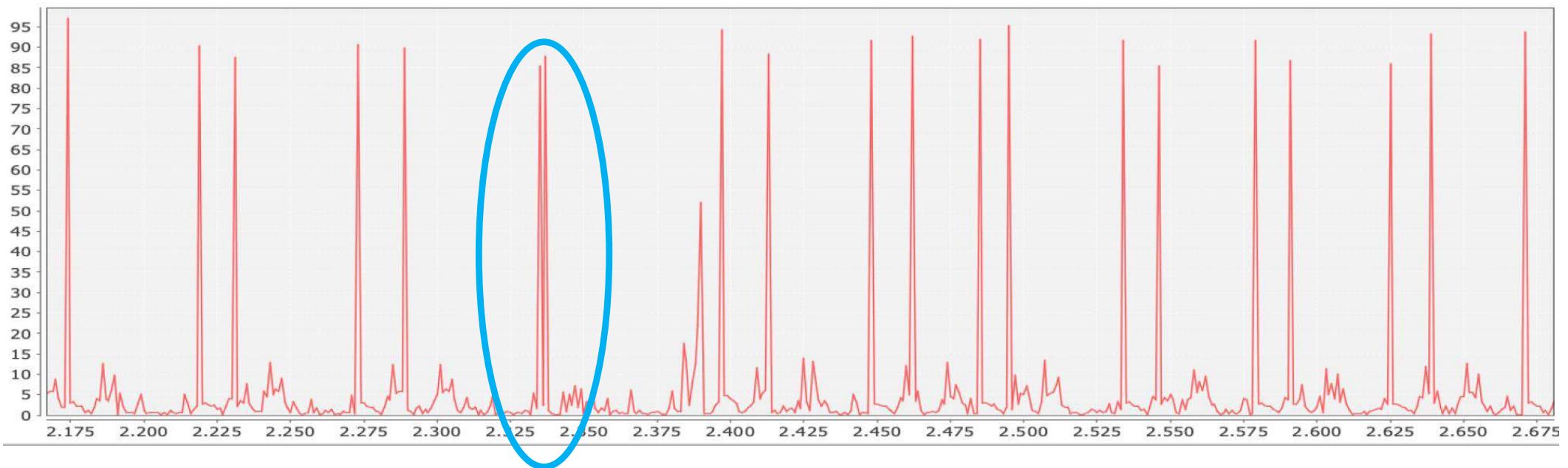
- This shows the absolute errors



Notice anything?

So, what about our data?

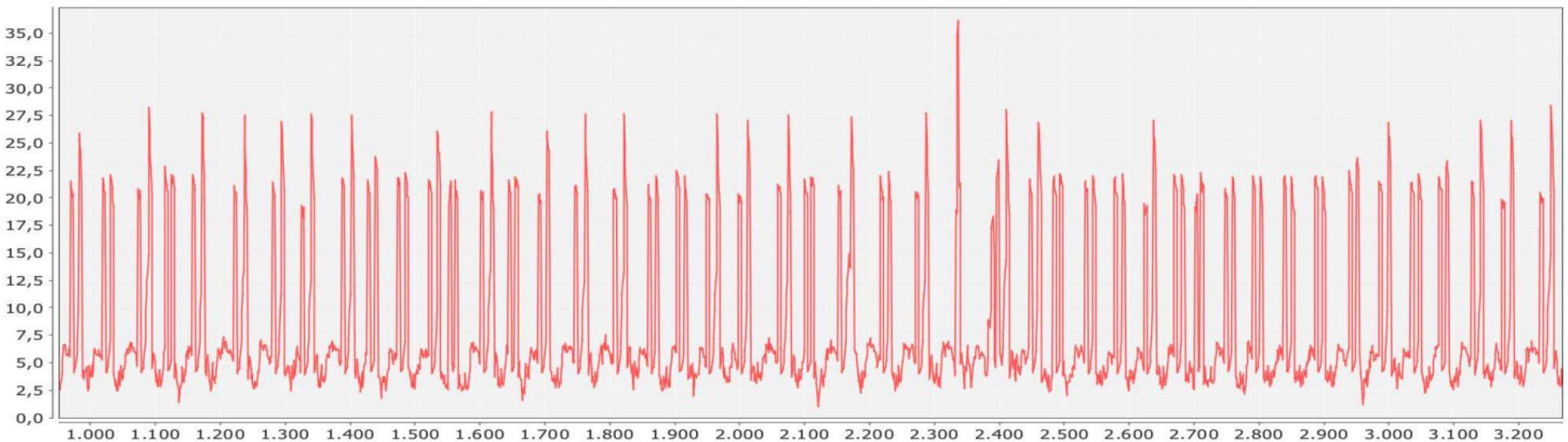
- This shows the absolute errors



These peaks are very near in time...

Some postprocessing

- Let's plot the mean error over a larger window, say size 5
- Remove differencing, as this seemed not useful (Keep it simple)



- Simple linear regression can still be used to detect this anomaly!

The initial data mining process

1. Visualize your data!
2. Try simple approaches. Is the problem difficult?

3. Look for something your system should be able to do
4. Look for problems in your setup, analyze it, fix if needed
5. Look for patterns, properties, features that could help
6. Add them to your system if helpful, but keep it simple!
7. Does it work? Goto 3
8. Does it not? Goto 4

9. Try and try again until satisfied, and hope it generalizes

The initial data mining process

1. Visualize your data!
2. Try simple approaches. Is the problem difficult?

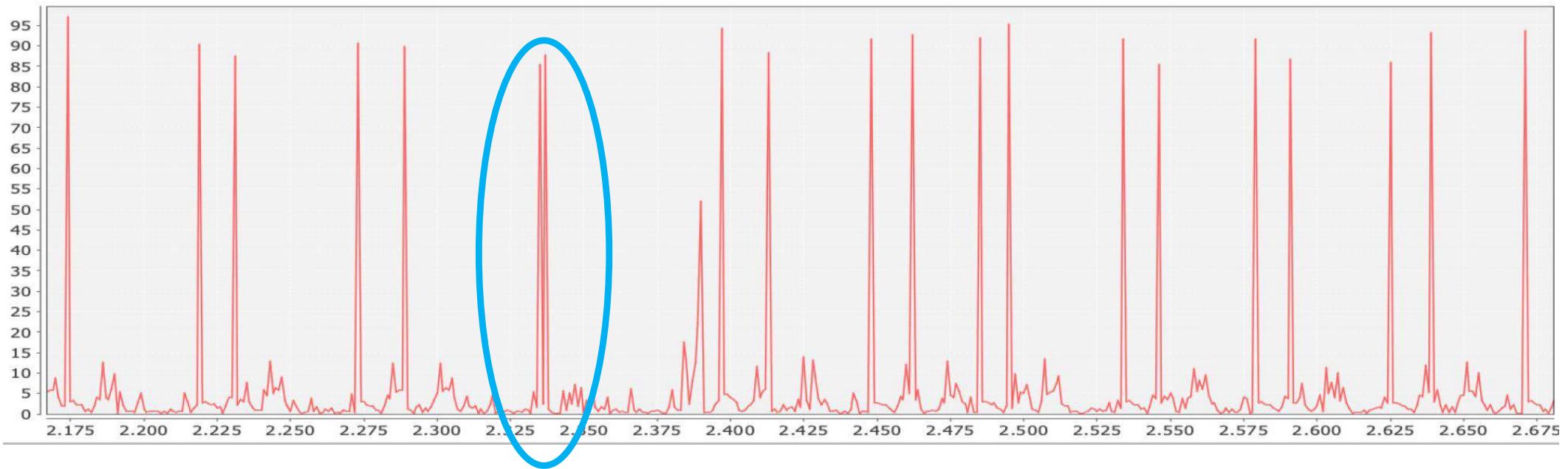
3. Look for something your system should be able to do
4. Look for problems in your setup, analyze it, fix if needed
5. Look for patterns, properties, features that could help
6. Add them to your system if helpful, but keep it simple!
7. Does it work? Goto 3
8. Does it not? Goto 4

9. Try and try again until satisfied, and hope it generalizes

This is your task for the first lab session in every assignment!

Collective anomalies

- Detection based on multiple data points -> collective



- In this case, we use post-processing, but we can also add pre-processing features such as moving averages to detect such anomalies

Collective anomalies

- Detection based on multiple data points -> collective



- In this case, we use post-processing, but we can also add pre-processing features such as moving averages to detect such anomalies

Three kinds of anomalies

- Point anomalies:
 - *an individual strange data points*
- Contextual anomalies:
 - *a data point that is strange given a set of data points as context*
- Collective anomalies:
 - *a set of data points that together are strange*

Anomaly detection

- Different techniques for different anomaly types
- Point:
 - Classification
 - Distance
 - Reconstruction
- Contextual:
 - Prediction
 - Conditional probability
- Collective:
 - Probability distribution
 - Feature engineering

Anomaly detection

- Different techniques for different anomaly types
- Point:
 - Classification
 - Distance
 - Reconstruction

For data point x , and set of context points C ,
find a function that returns label y :

$$f(x) \rightarrow y$$
- Contextual:
 - Prediction
 - Conditional probability
$$f(x,C) \rightarrow y$$
- Collective:
 - Probability distribution
 - Feature engineering
$$f(C) \rightarrow y$$

Anomaly detection

- Different techniques for different anomaly types
 - Point:
 - Classification
 - Distance
 - Reconstruction

For data point x , and set of points C ,
find a function that returns label y :

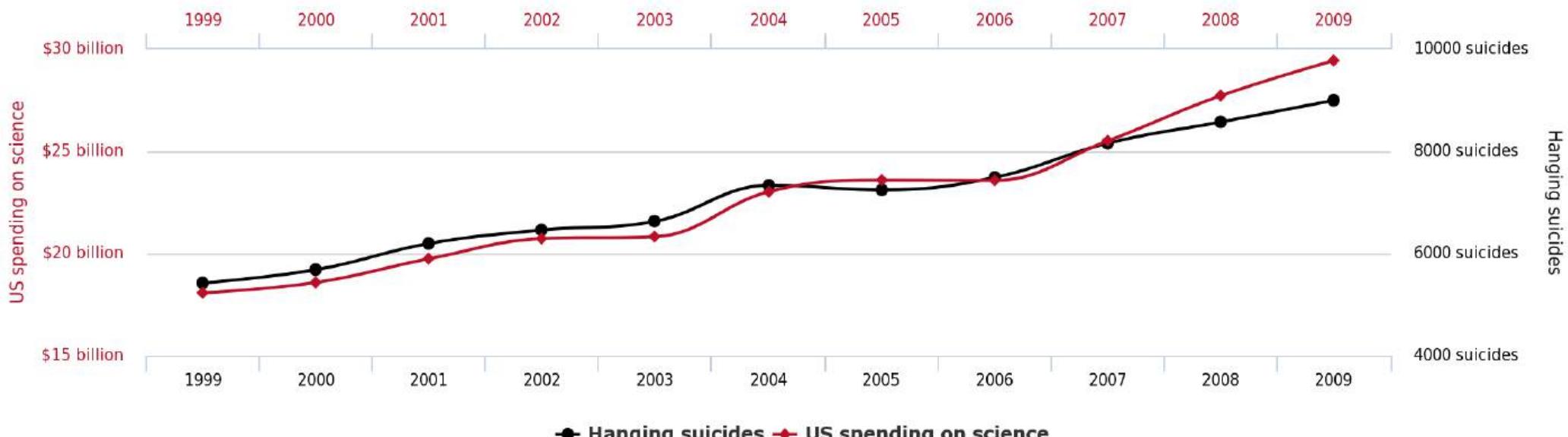
$$f(x) \rightarrow y$$
 - Context:
 - Preprocessing
 - Contextual
 - Collective:
 - Preprocessing
 - Feature
- The focus in lab 1 is on detecting point anomalies
- Use preprocessing and/or postprocessing to
detect collective ones

Today

- 3 types of anomalies
 - Point
 - Contextual
 - Collective
- Modeling context
 - Sliding Windows
 - Time Warping distance
- Popular anomaly detection methods
 - Classification
 - Nearest Neighbor
 - Spectral
 - Deep Learning

Shape often matters most

US spending on science, space, and technology
correlates with
Suicides by hanging, strangulation and suffocation



tylervigen.com

Are these series similar? Their y-ranges are very different...
check out: <http://www.tylervigen.com/spurious-correlations>

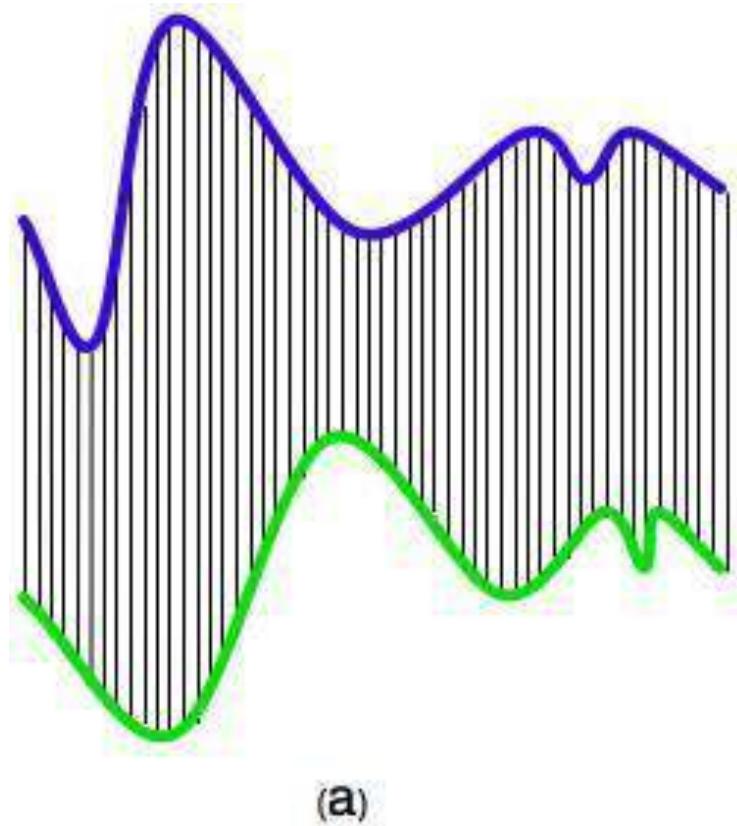
Scaling time series

- To compare shapes, we need to scale the series:

$$c'_i = \frac{c_i - \mu(C)}{\sigma(C)}$$

- for every data point c_i , where μ is the mean and σ is the standard deviation of the series
- After normalization, we can compute distances between sliding windows

Euclidean distance



(a)

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

Euclidean distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10
$(s1 - s2)^2$	100	225	400	0	625	100	100	100

$$\text{Distance} = \sqrt{100+225+400+0+625+100+100+100} = 40.62$$

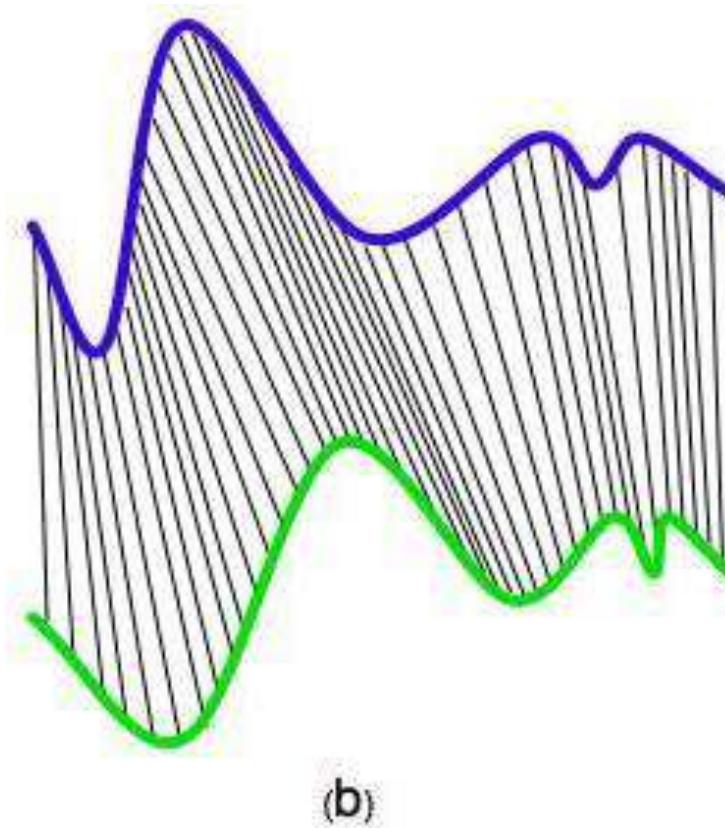
Euclidean distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10
$(s1 - s2)^2$	100	225	400	0	625	100	100	100

$$\text{Distance} = \sqrt{100+225+400+0+625+100+100+100} = 40.62$$

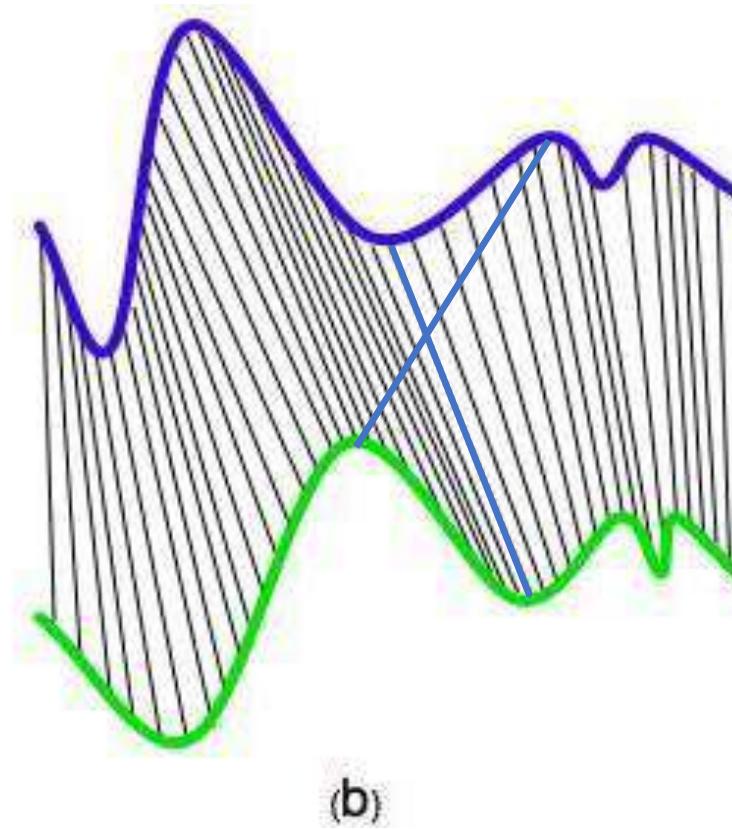
Signals seem out of phase...

Dynamic Time Warping or sequence alignment



Aligns time-series non-linearly in time,
finds the best match

Dynamic Time Warping or sequence alignment



Aligns

Lines are not allowed to cross,
keeping shape intact

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100							
v2		225						
v3		.	400					
v4				0				
v5					625			
v6						100		
v7							100	
v8								100

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Find a minimum cost path from left top to bottom right

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Without going up or left (crossing)

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Without

For instance...

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Distance = $\sqrt{100+25+0+0+0+25+0+25+100} = 16.58$

Time-warping distance

Before alignment

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

After alignment

v/v	1/1	2/1	3/1	4/2	5/3	6/4	6/5	7/6	8/7	8/8
s1	10	15	20	30	0	25	25	10	20	10
s2	20	20	20	30	0	30	25	15	20	10

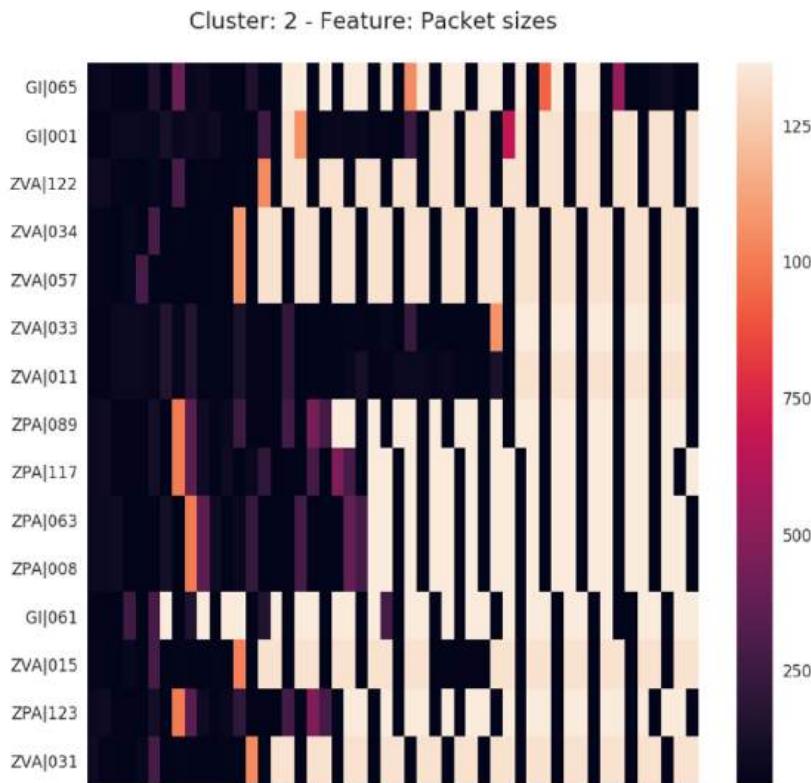
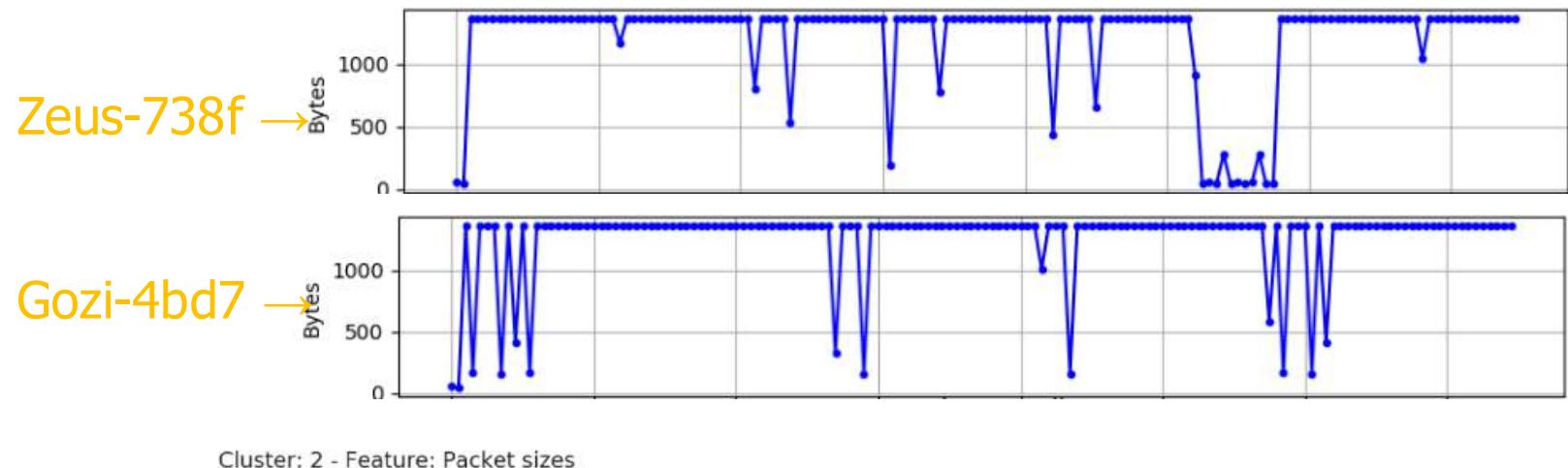
Although we consider more points,
their synchronisation gives a smaller distance!

Applications in real world

- Extensive use in
 - Speech recognition
 - Gesture recognition
 - Handwriting recognition
 - ...
- But now, also in
 - Clustering malware behavior

No.	Source	Destination	Protocol	Length	Info
40	192.168.1.2	192.168.1.110	ICMP	82	Redirect (Redirect for host)
41	CzNicZSP_00:0...	PcsCompu_7c:9...	ARP	60	192.168.1.1 is at d8:58:d7:00:0f:72
42	192.168.1.110	203.153.165.21	TCP	182	49191 → 8343 [PSH, ACK] Seq=1 Ack=1 Win=65700 Len=128
43	203.153.165.21	192.168.1.110	TCP	60	8343 → 49191 [ACK] Seq=1 Ack=129 Win=15744 Len=0
44	203.153.165.21	192.168.1.110	TCP	1188	8343 → 49191 [PSH, ACK] Seq=1 Ack=129 Win=15744 Len=1134
45	192.168.1.110	203.153.165.21	TCP	380	49191 → 8343 [PSH, ACK] Seq=129 Ack=1135 Win=64564 Len=326
46	192.168.1.2	192.168.1.110	ICMP	408	Redirect (Redirect for host)
47	203.153.165.21	192.168.1.110	TCP	113	8343 → 49191 [PSH, ACK] Seq=1135 Ack=455 Win=16768 Len=59
48	fd2d:ab8c:225...	fd2d:ab8c:225...	DNS	110	Standard query 0xb554 A www.download.windowsupdate.com

Example: Clustering malware



- DTW on network traffic with clustering identifies malware families

Important for lab

- **Sliding windows:**
 - obtain fixed length rows by moving a window over the series
 - learn a model from the obtained data table
- **Distances with sequential orders:**
 - first normalize if shape matters
 - Euclidean distance is OK for synchronized series
 - Time warping/sequence alignment handles out-of-sync series
- More on distances next week....

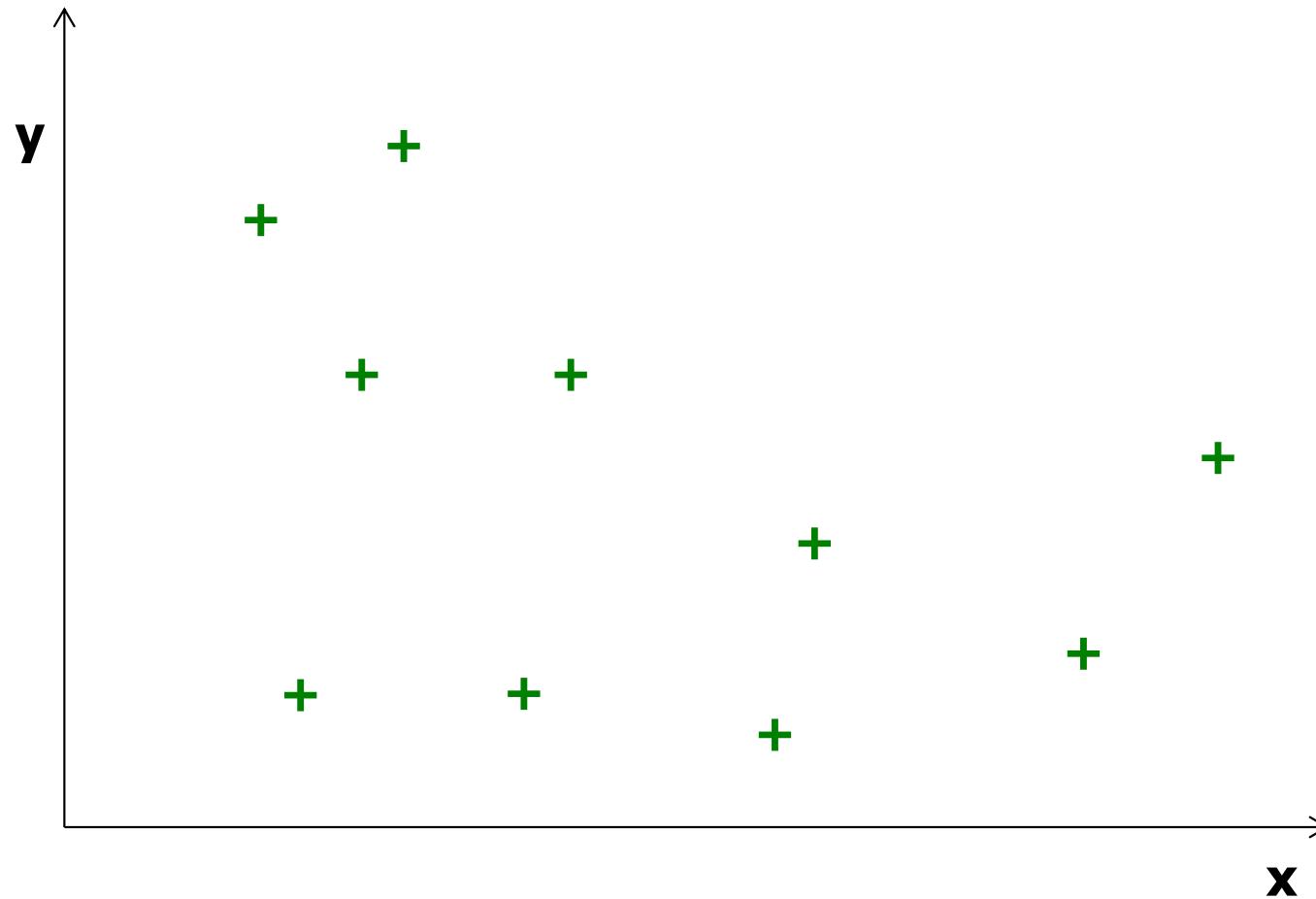
Today

- 3 types of anomalies
 - Point
 - Contextual
 - Collective
- Modeling context
 - Sliding Windows
 - Time Warping distance
- Popular anomaly detection methods
 - Classification
 - Nearest Neighbor
 - Spectral
 - Deep Learning

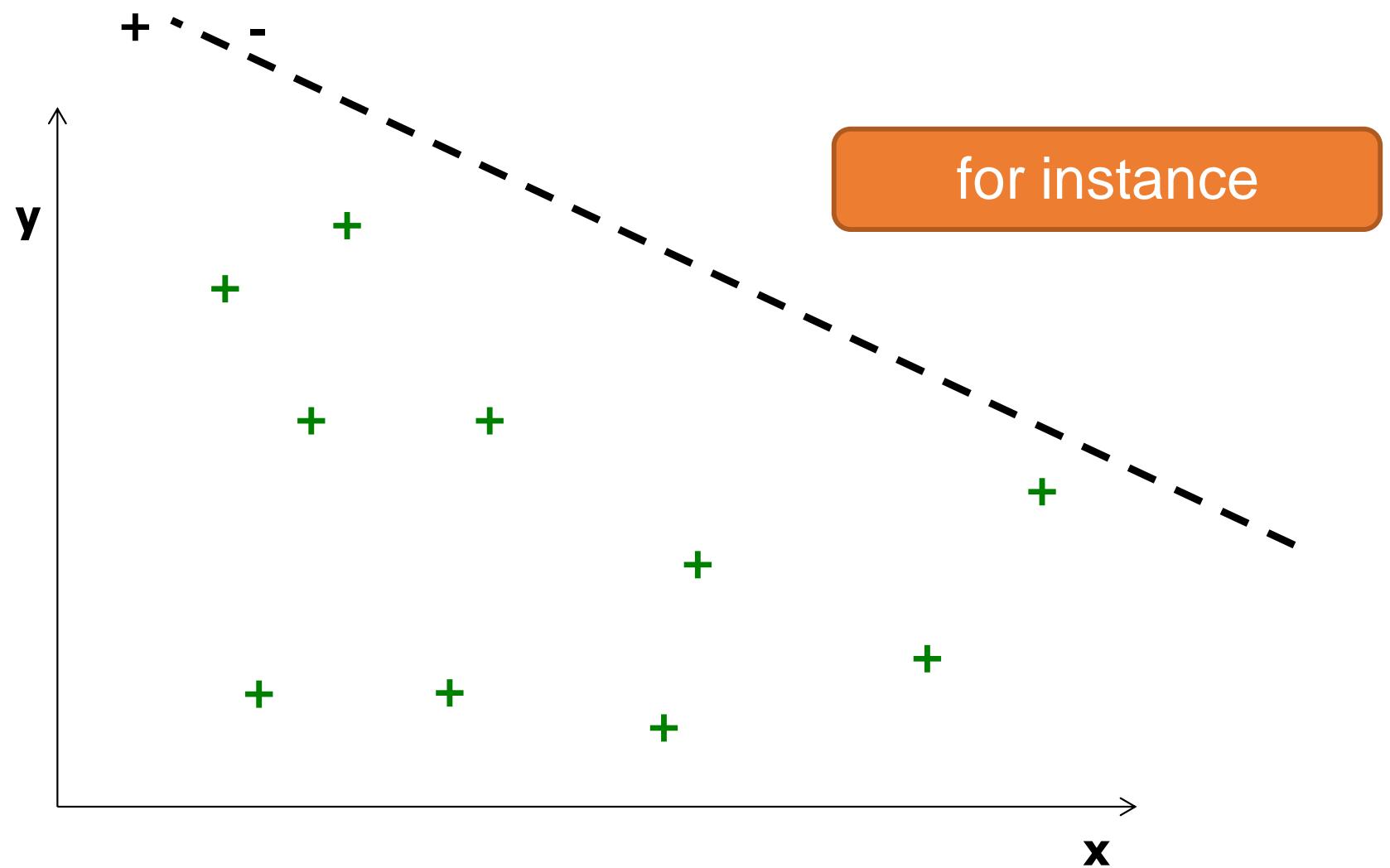
Anomaly detection

- A mess of possible methods:
 - Clustering
 - (One-class) Classification
 - Nearest Neighbors
 - Statistical
 - Spectral
 - ...
- ***Key ingredient: assumption of what is an anomaly***

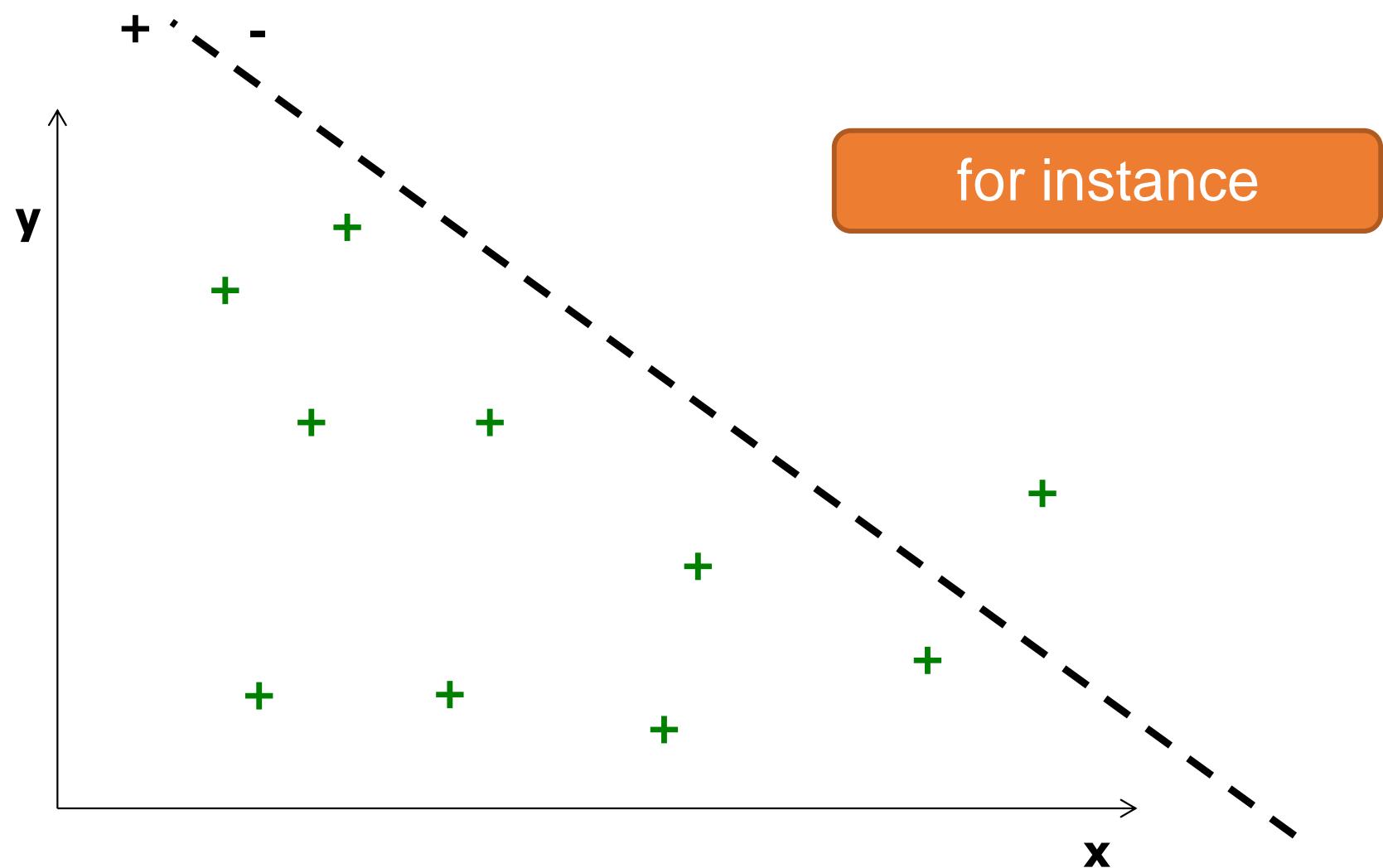
Where to put the decision boundary?



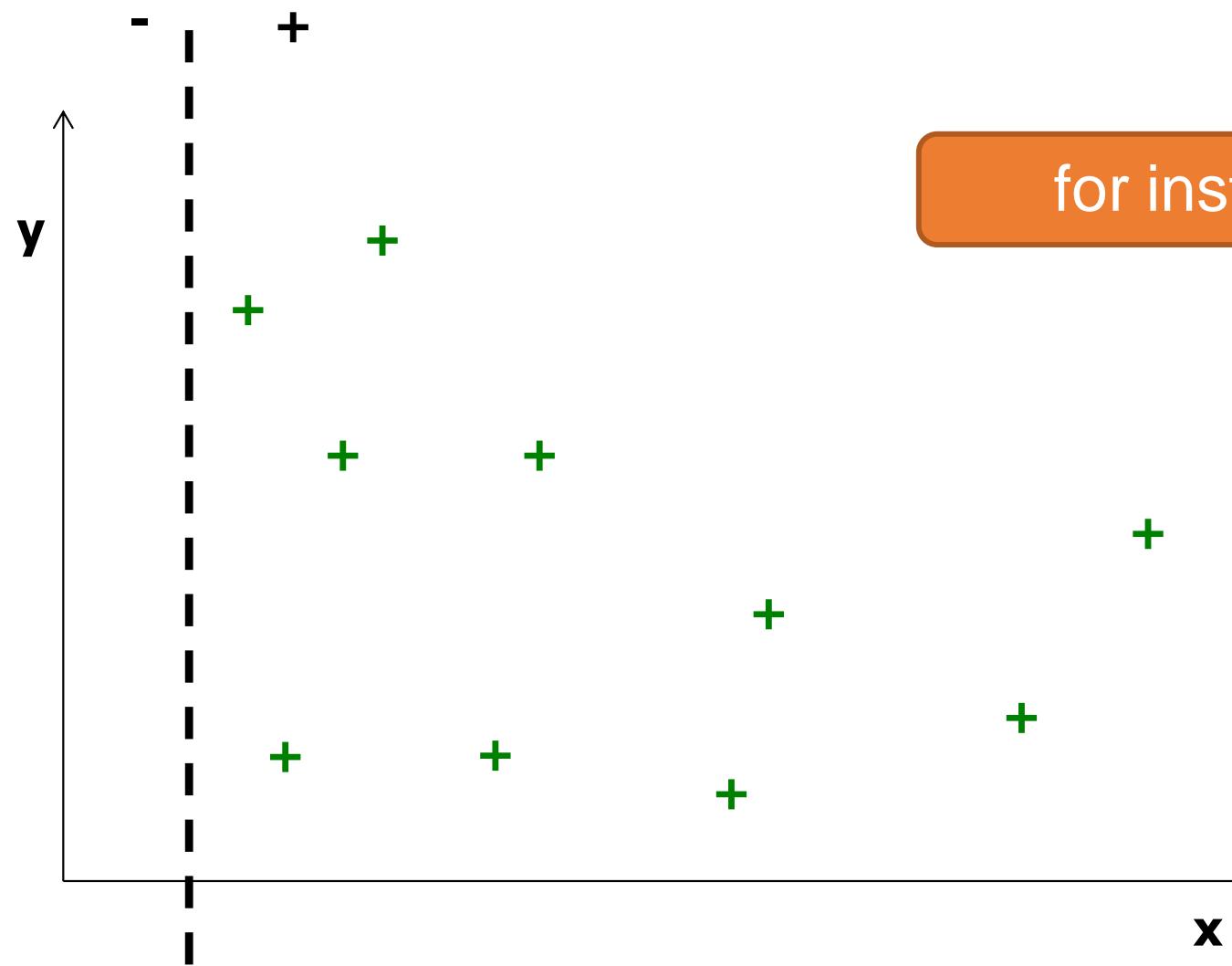
Where to put the decision boundary?



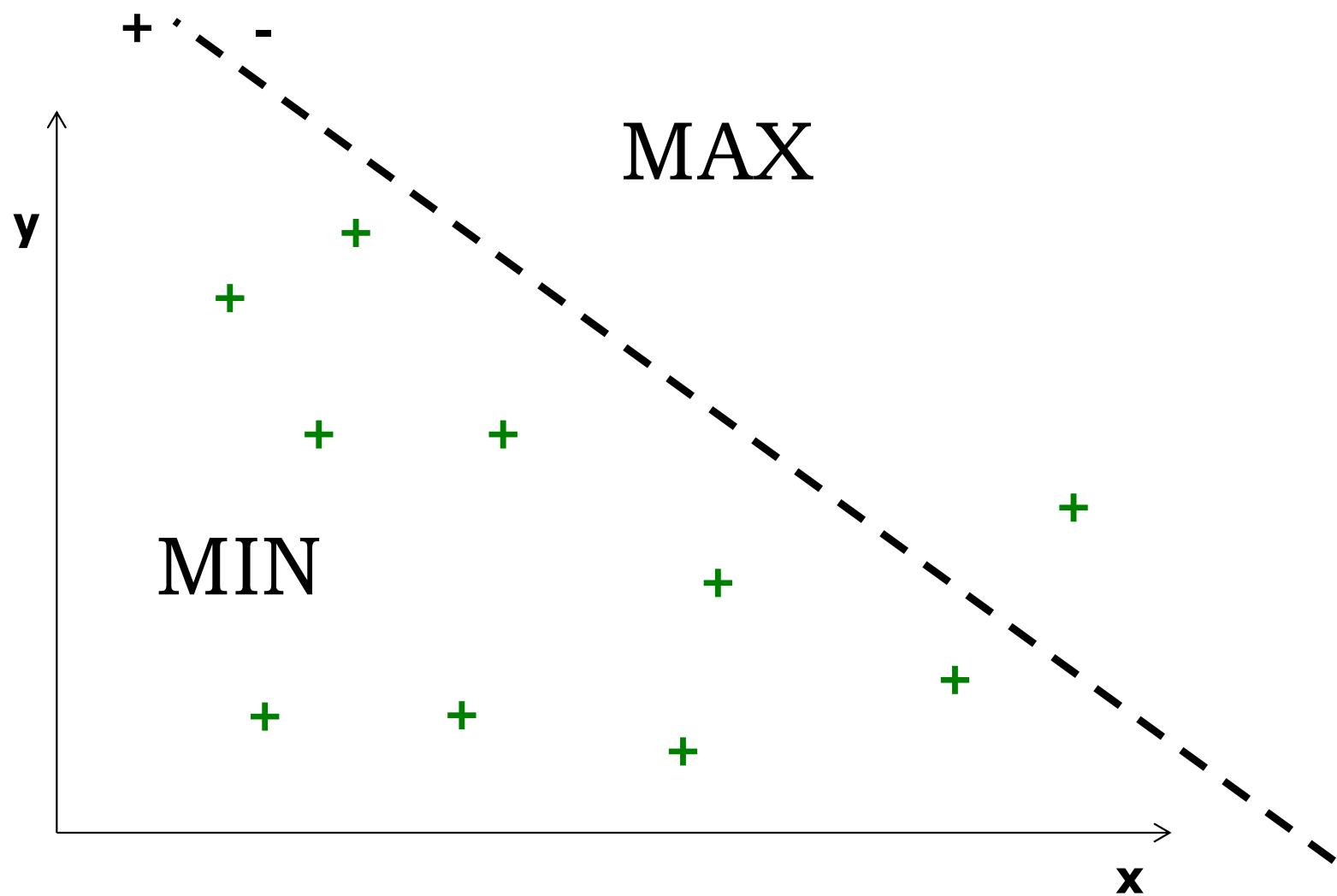
Where to put the decision boundary?



Where to put the decision boundary?



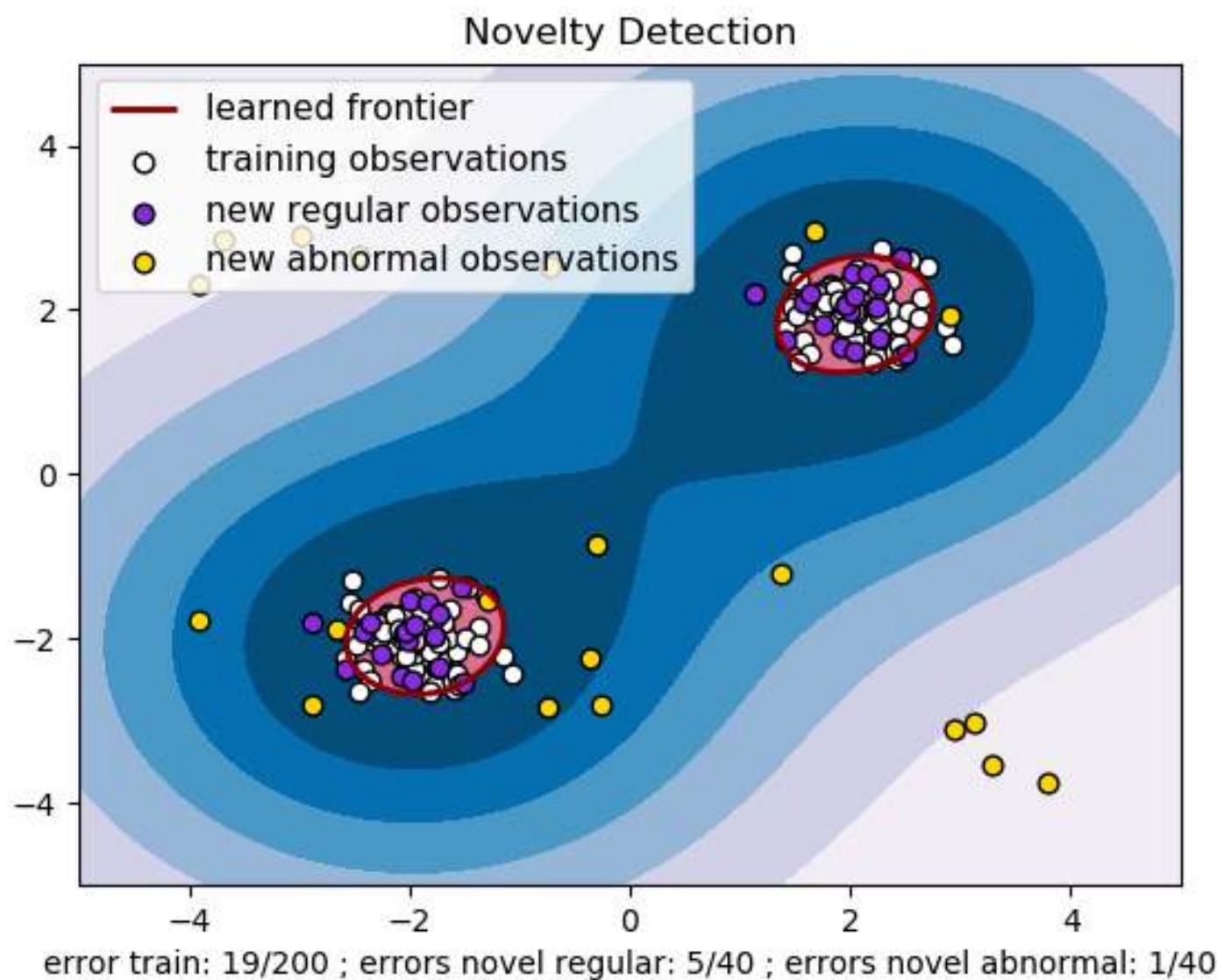
for instance



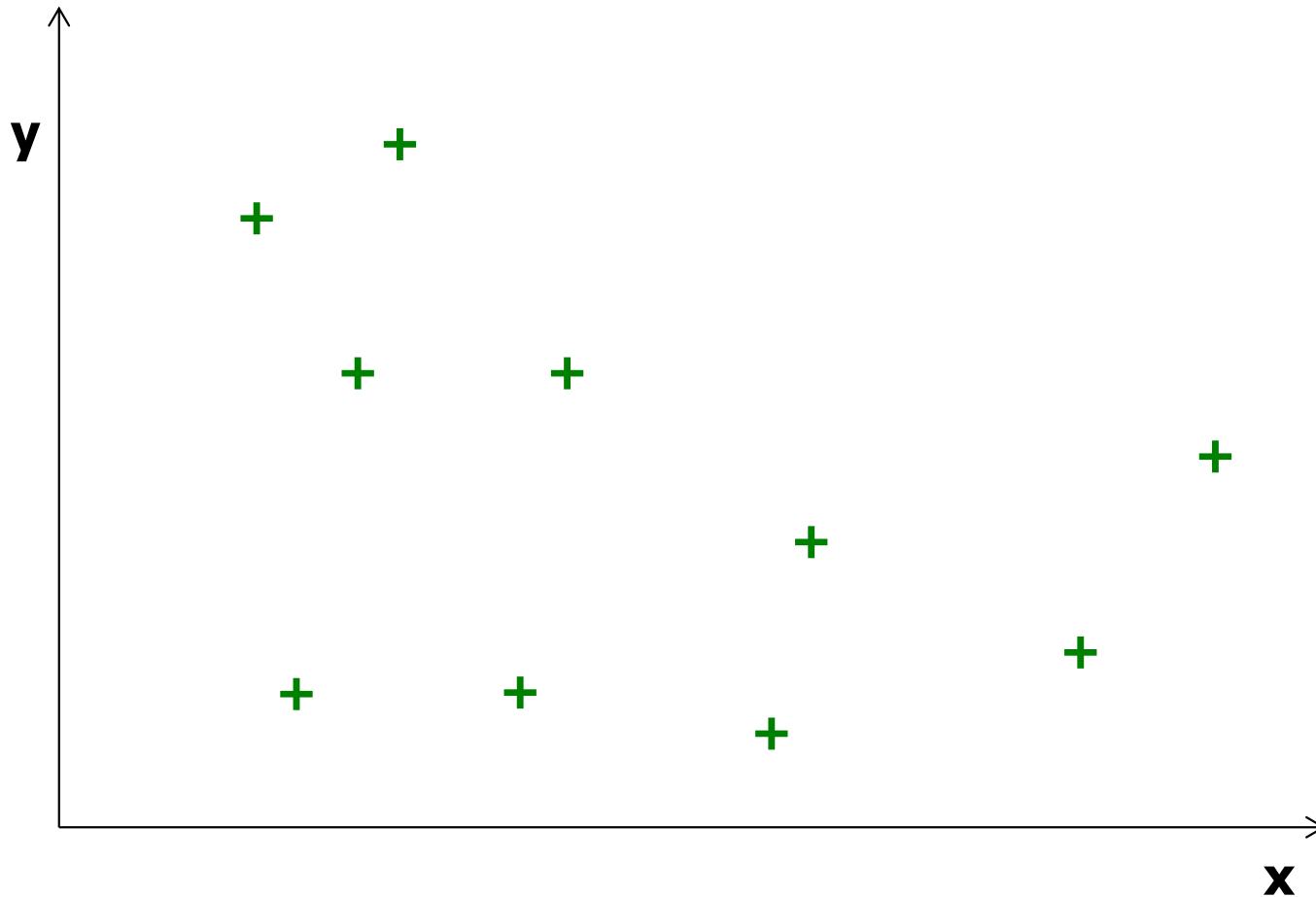
Classification based methods: OSVM

- Only positive data:
separate positive data from remaining input space
- Assumptions:
 - Further away from origin is anomalous (one-class SVM)
 - Close to the origin is anomalous (different one-class SVM)
 - Close to origin is also anomalous (improved one-class SVM)
 - Further from centroid is anomalous (non-linear one-class SVM)
 - ...
- Key ingredient:
 - *Maximize negative/outlier space*
 - *Minimize positive/normal space*

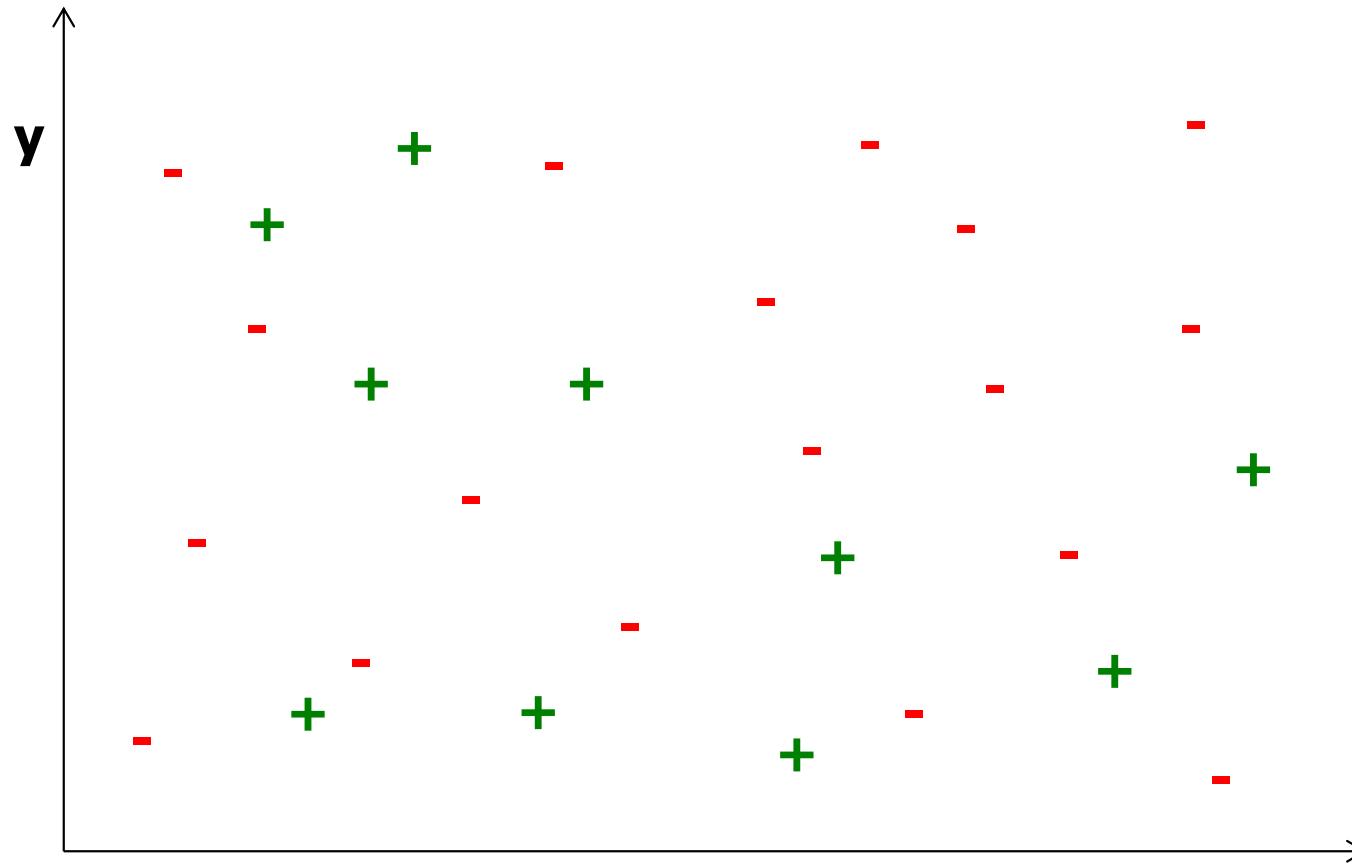
RBF one-class SVM



Different ways to use classification?

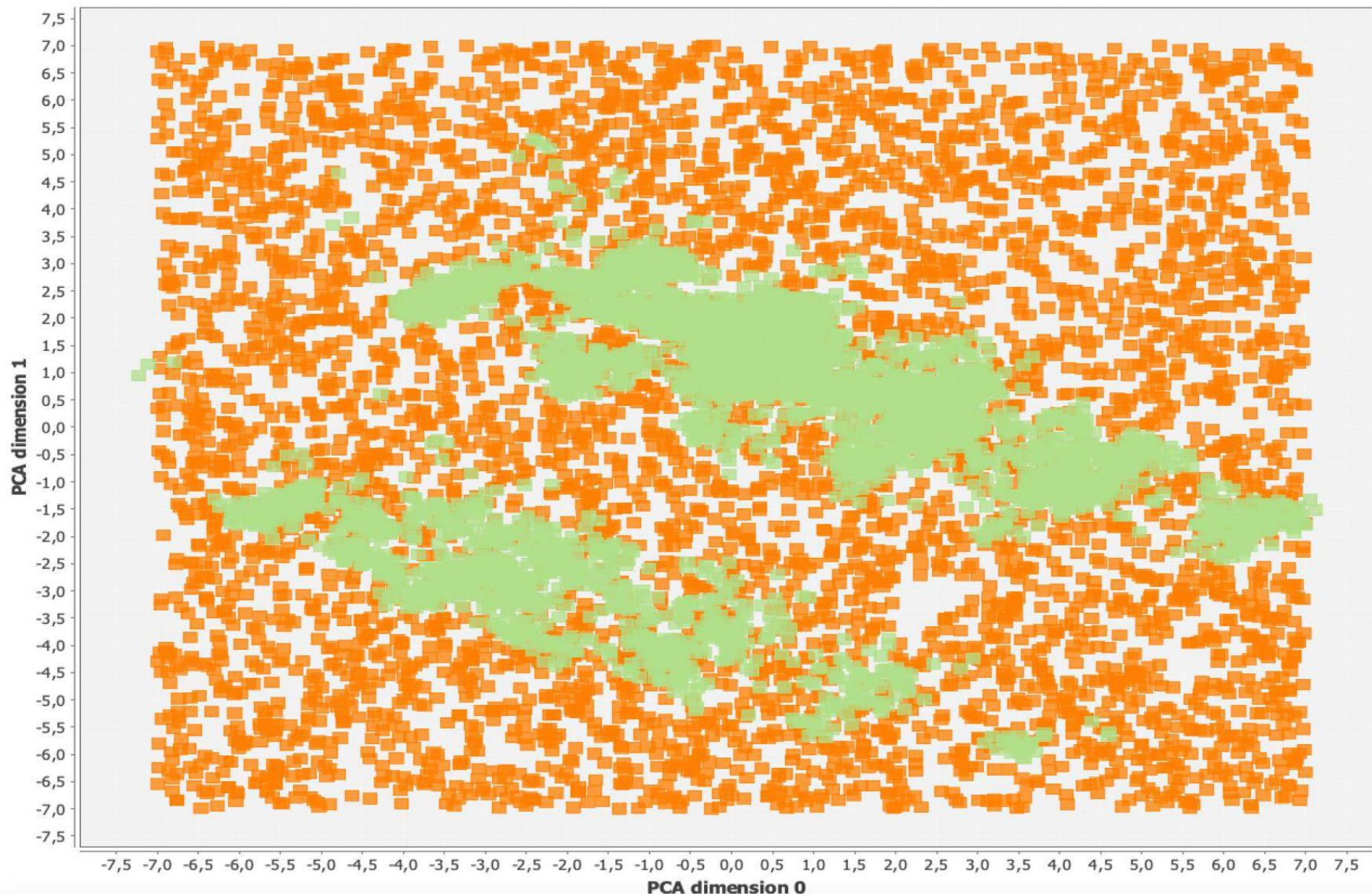


Different ways to use classification?

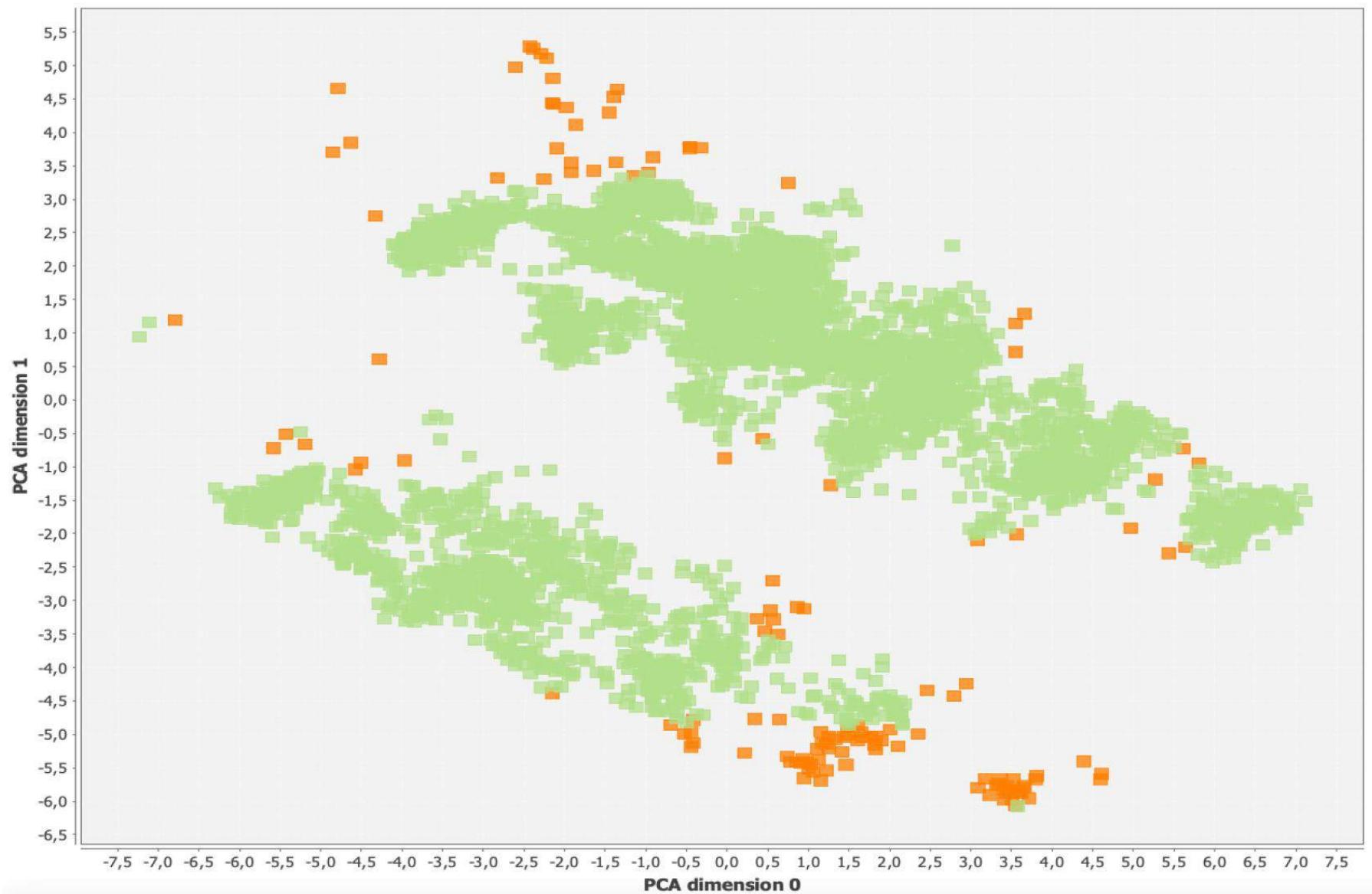


Add artificial negative samples (e.g., uniformly at random)

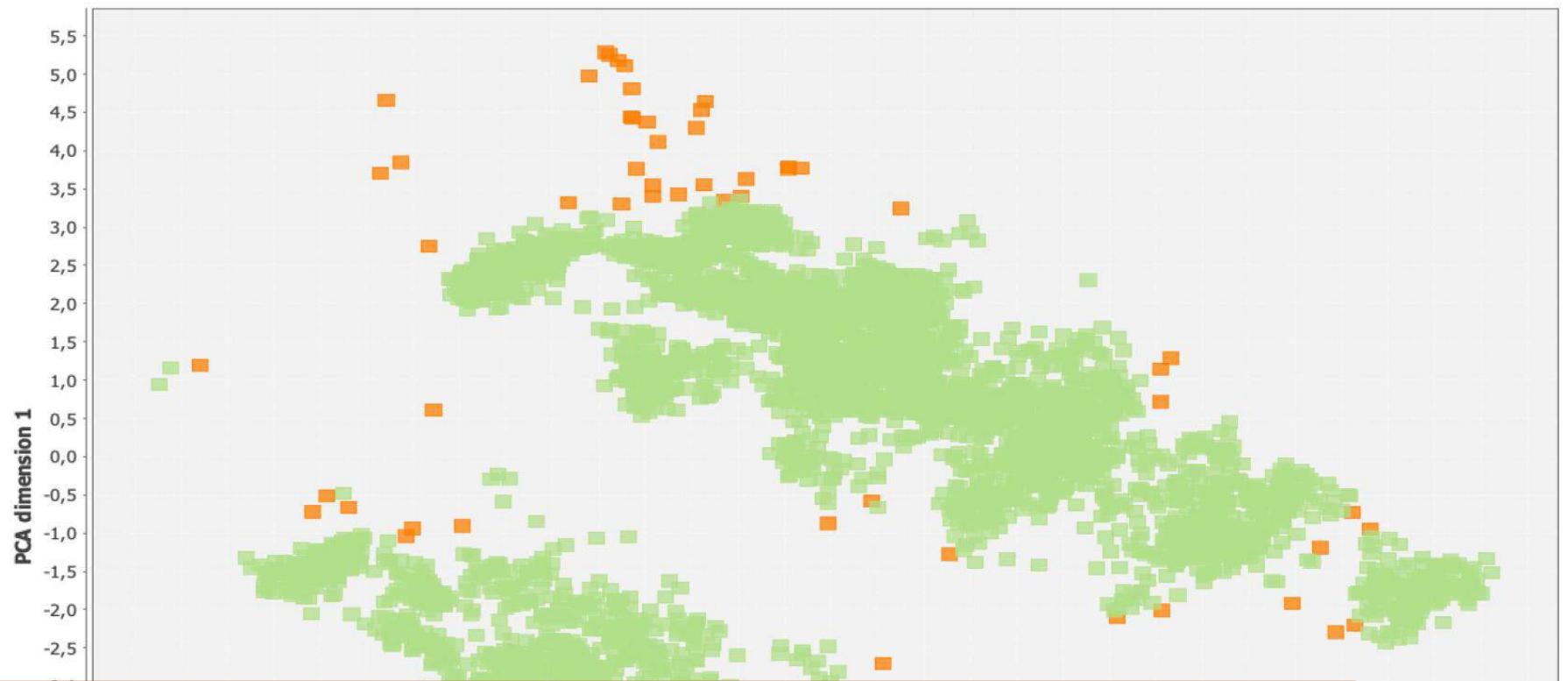
On our 2D data, uniform (-7,7)



Random forest, depth 8 trees

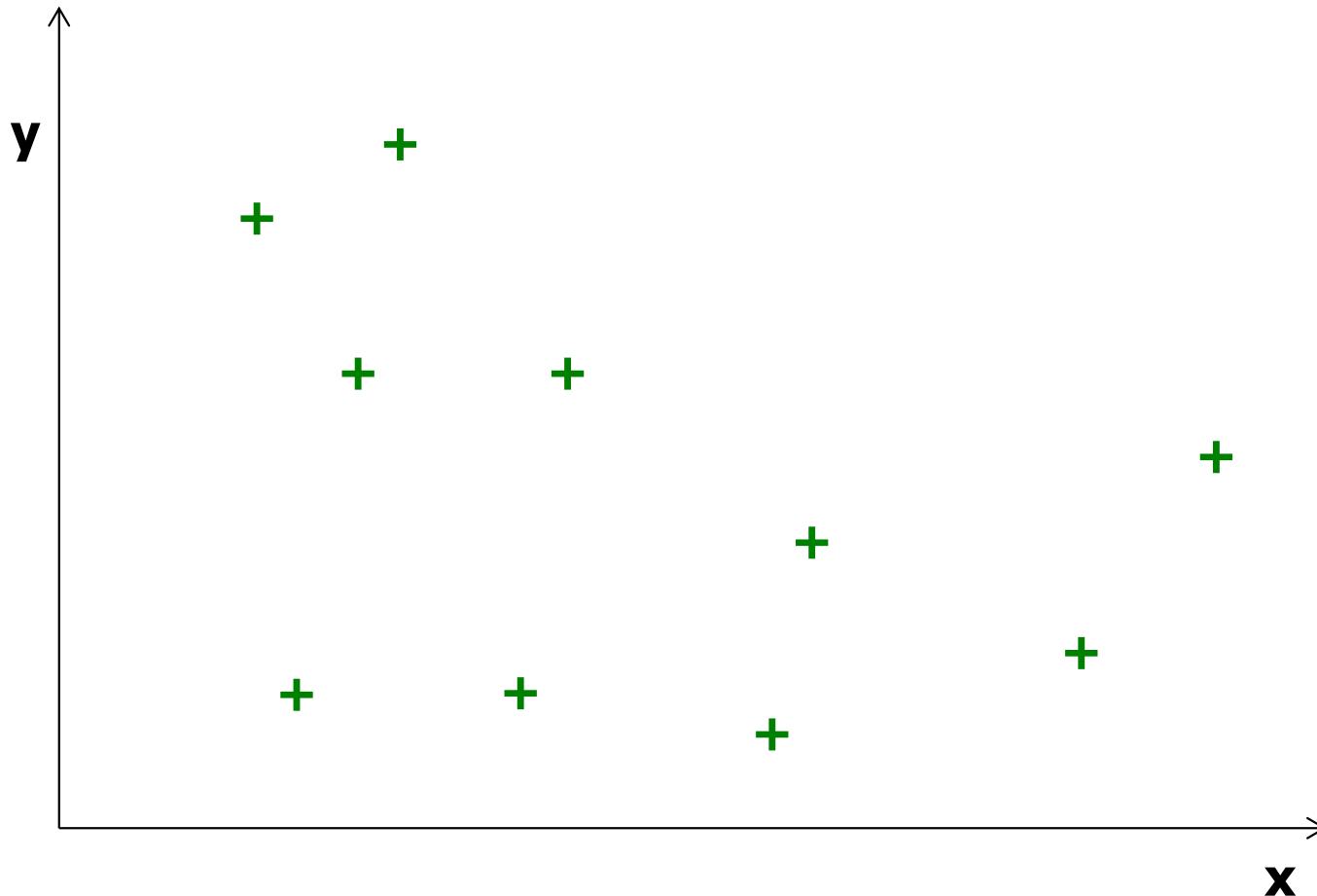


Random forest, depth 8 trees



Assuming the train set is larger than the test set
You could also learn to separate train data from test data
Avoid overfitting
The points that are believed to be in the test data are outliers

Different ways to use classification?



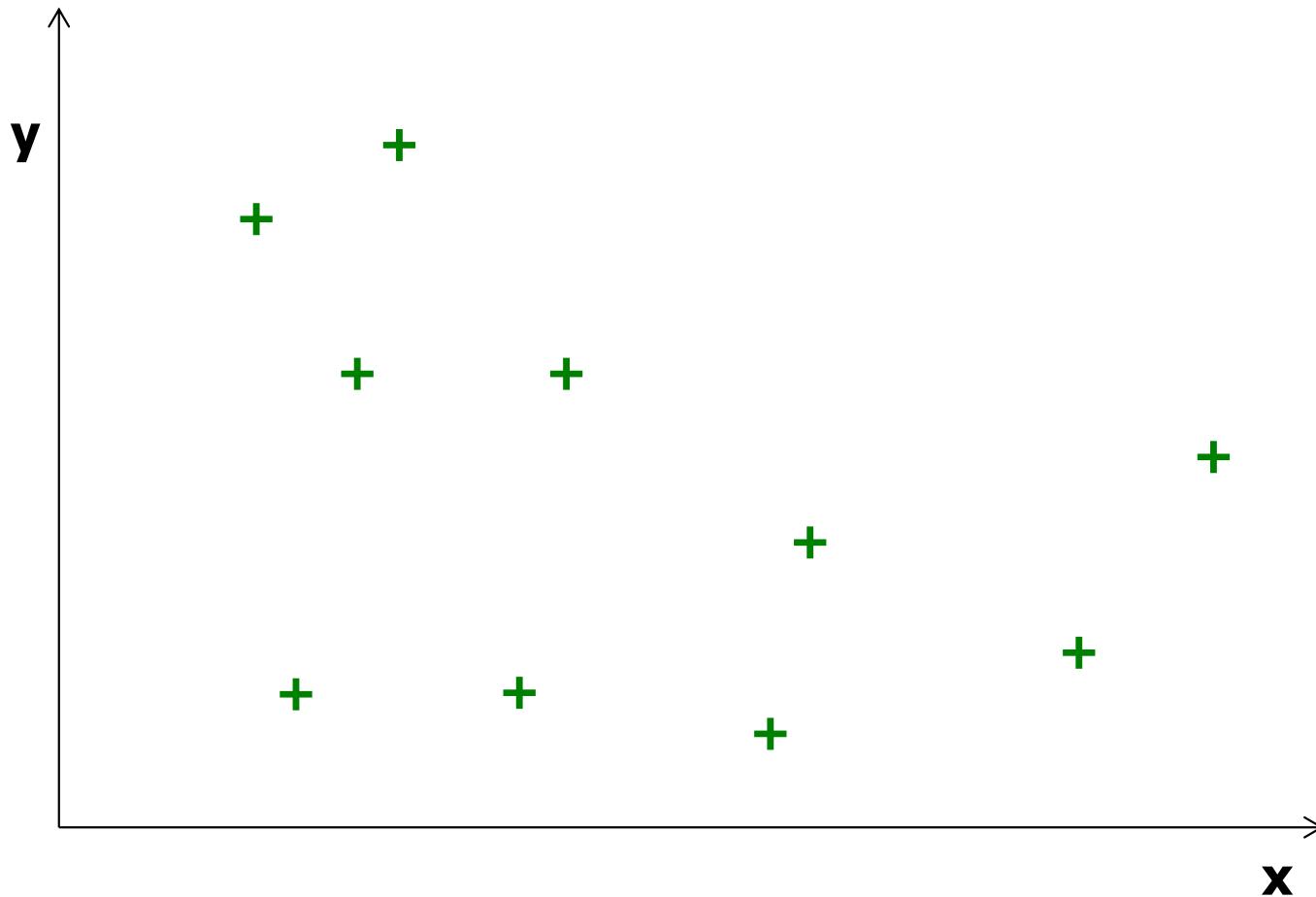
Isolation Forest

- Repeat N times:
 - Randomly pick a feature f
 - Split the f uniformly at randomly between [min,max]
 - Continue until all leafs contain singletons
- The path length to reach a leaf is the isolation score
- *Q: What is the intuition behind this score?*

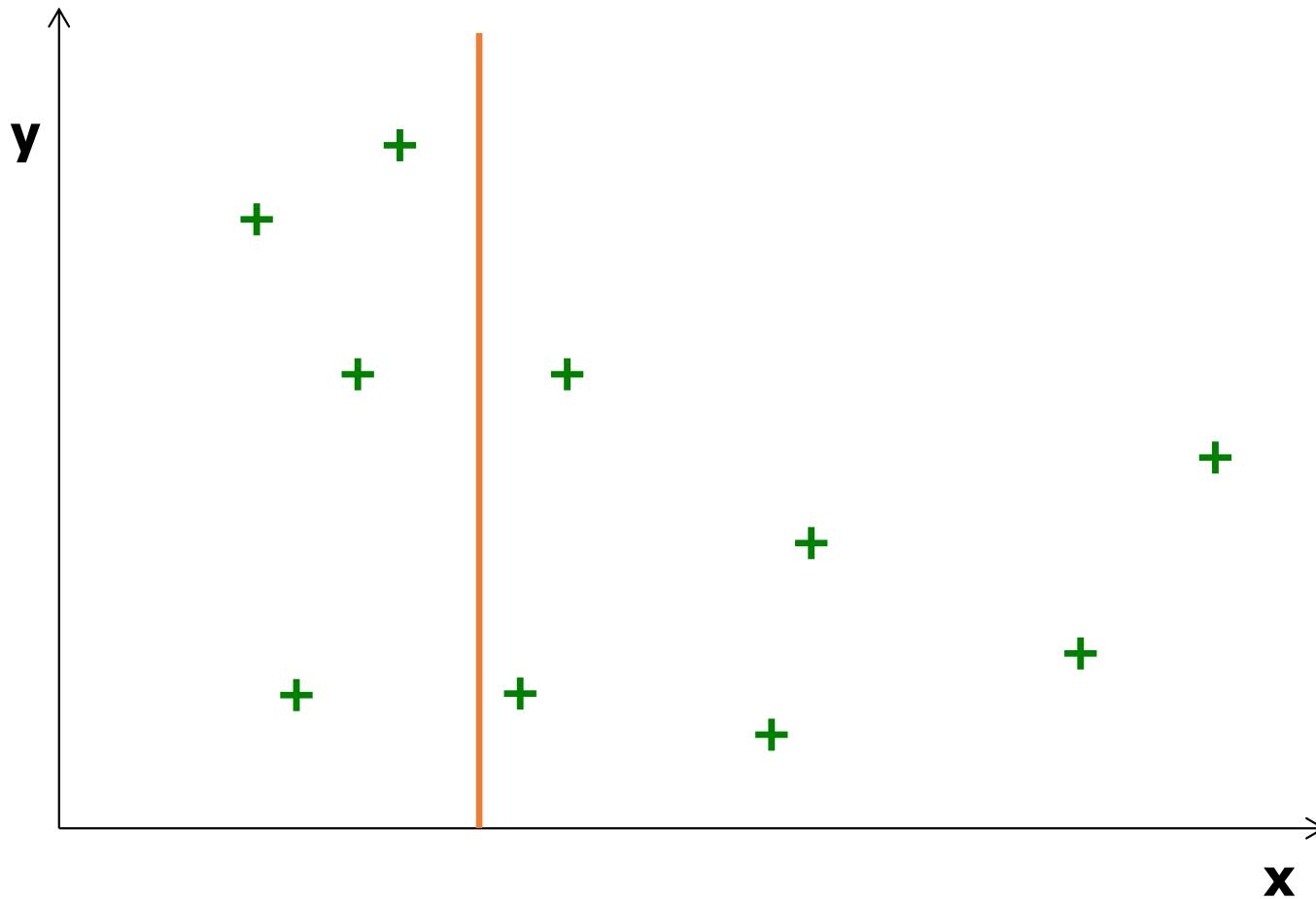
Isolation Forest

- Repeat N times:
 - Randomly pick a feature f
 - Split the f uniformly at randomly between [min,max]
 - Continue until all leafs contain singletons
- The path length to reach a leaf is the isolation score
- Average this length over all trees to get the anomaly score
- Intuition:
 - isolating anomalies is easier because only a few conditions are needed to separate those cases from the normal observations

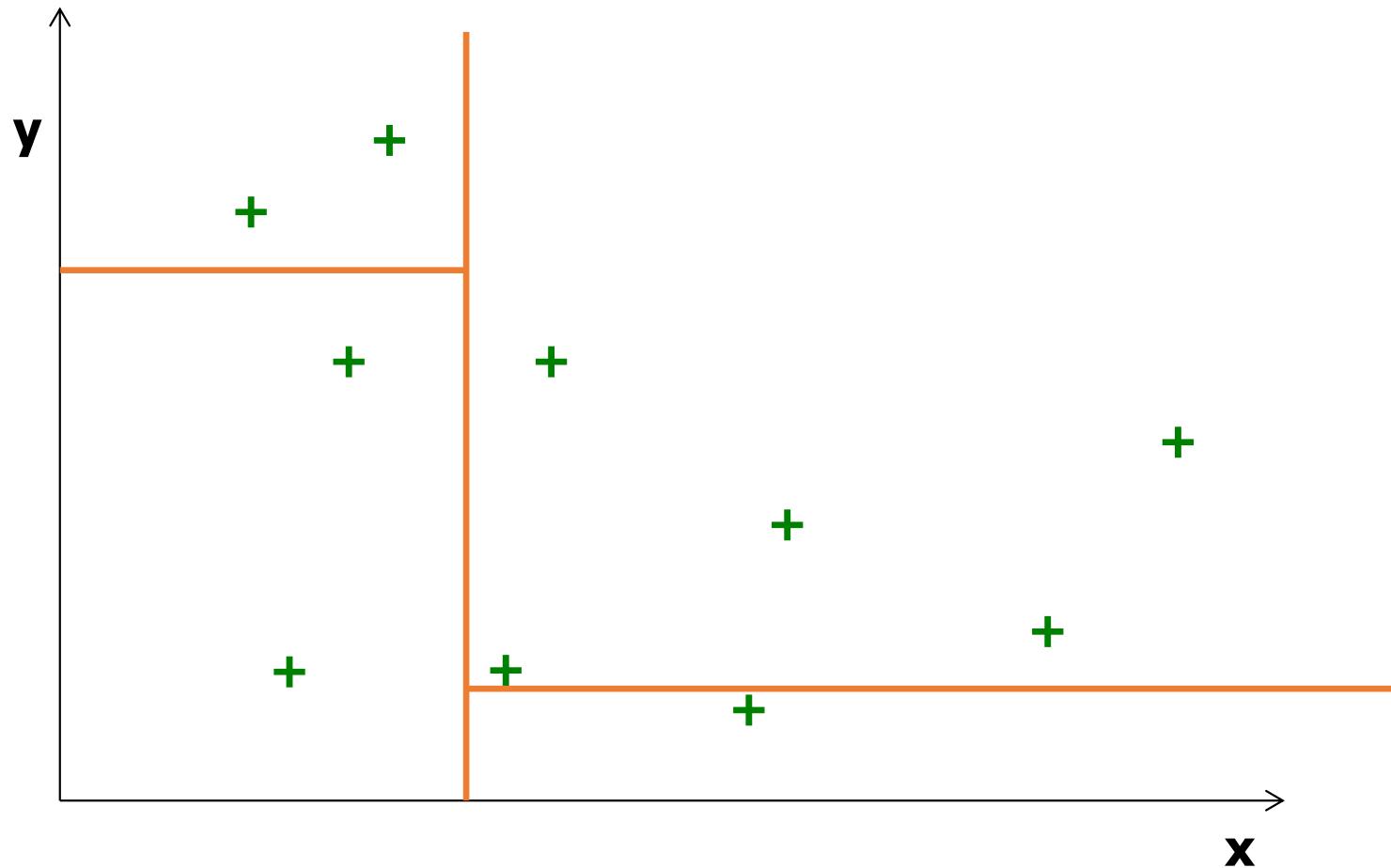
Isolation Forest



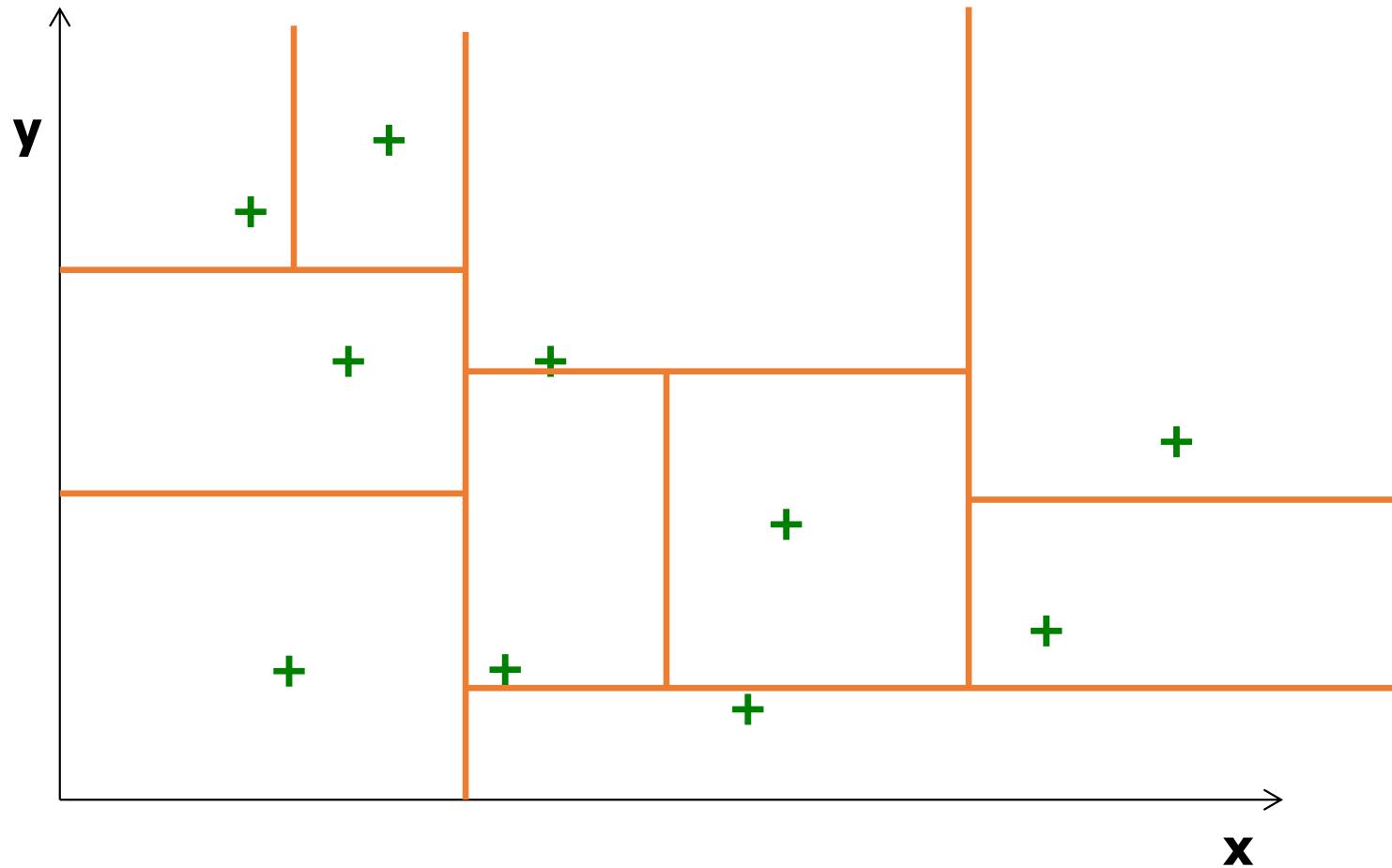
Isolation Forest



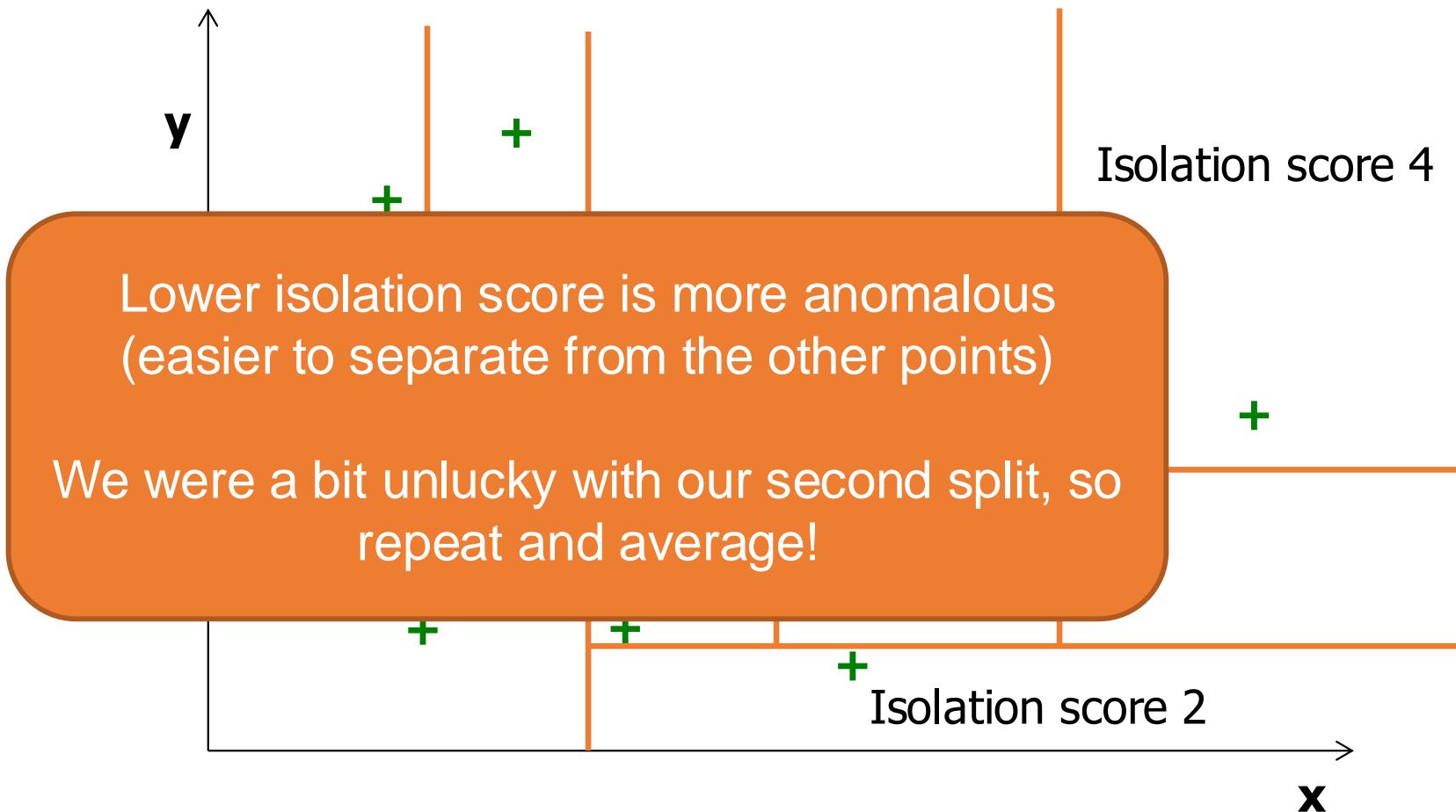
Isolation Forest



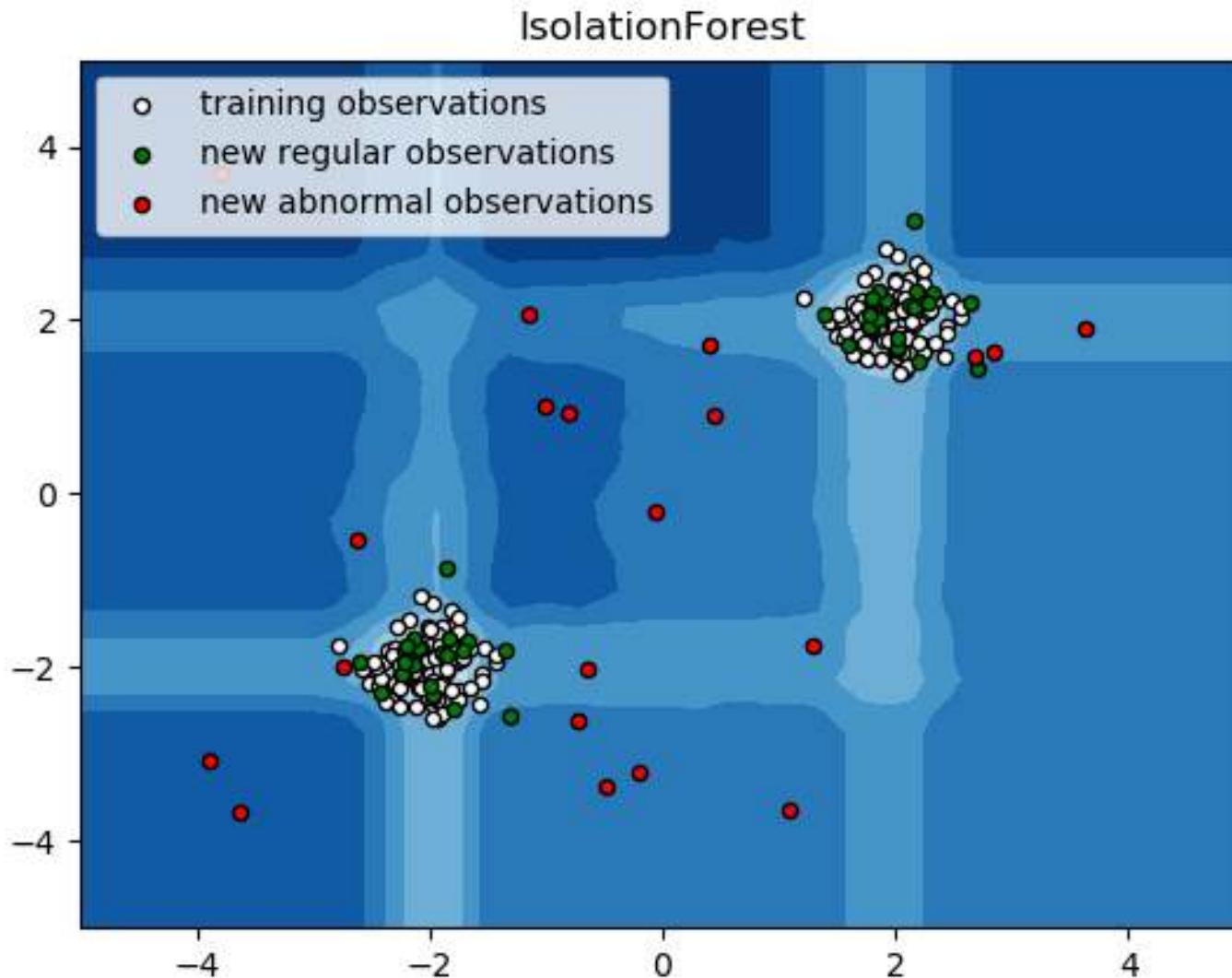
Isolation Forest



Isolation Forest



Isolation Forest



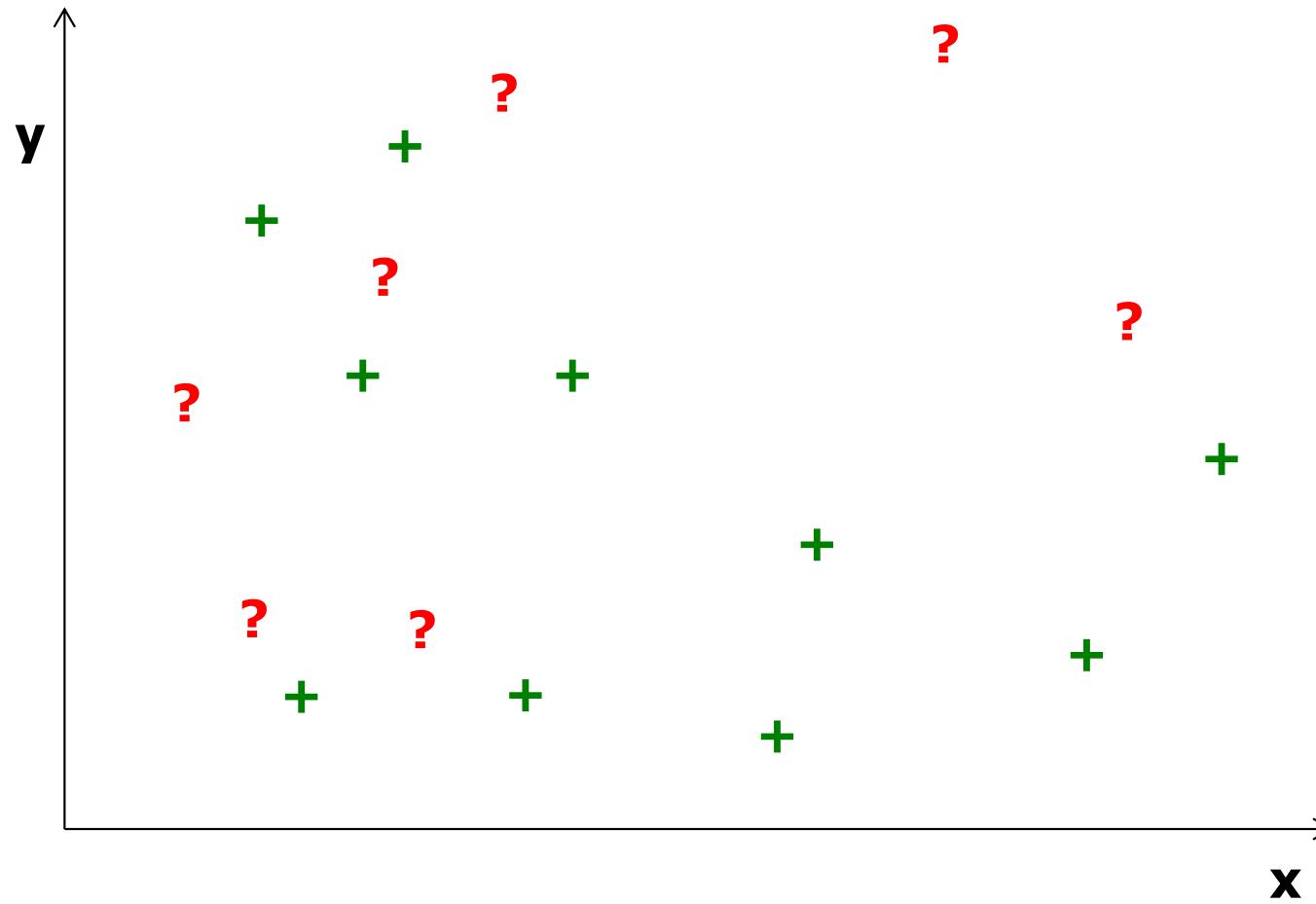
Classification Based Techniques

- Advantage
 - Gives access to powerful classifiers
 - Some models are interpretable
 - Computationally inexpensive when testing
- Drawback
 - Make assumptions about data distribution
 - Where is the origin? Is it normal or anomalous?

Nearest Neighbour Based Techniques

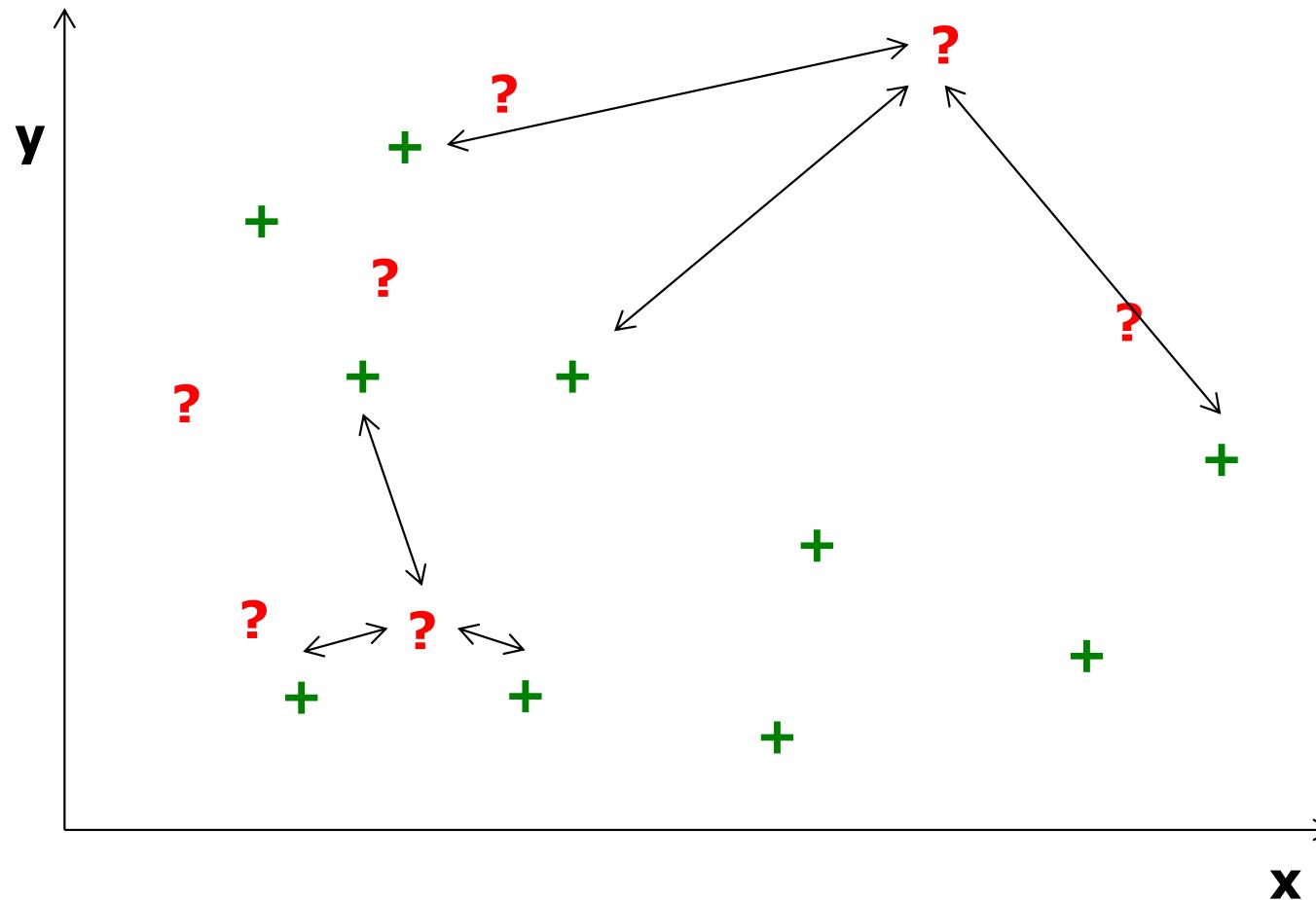
- Key assumption:
normal points have close neighbours while anomalies are located far from other points
- Two-step approach
 1. Compute neighbourhood for each data record
 2. Analyse the neighbourhood to determine whether data record is anomaly or not

How to use neighbors?

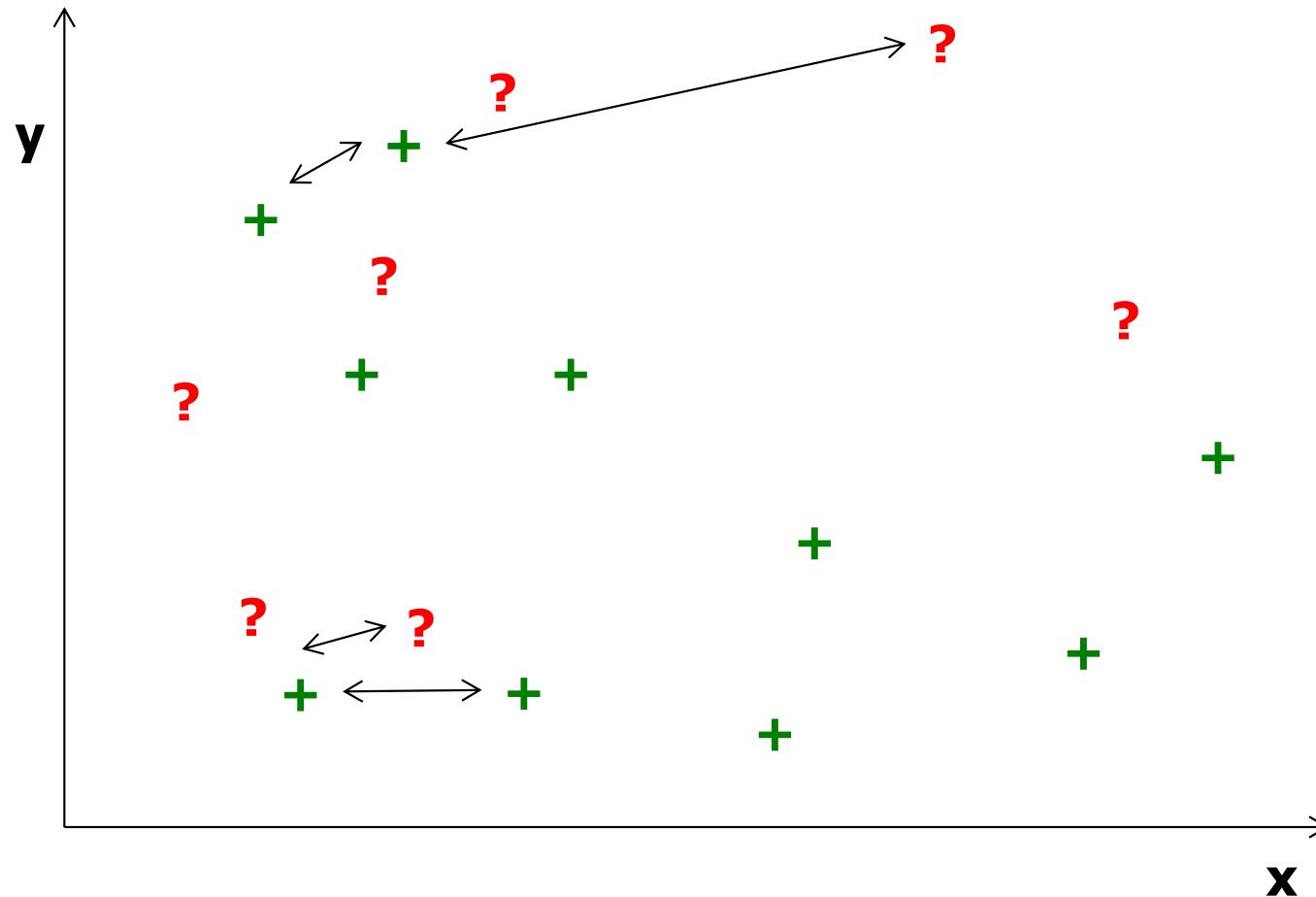


How to use neighbors?

Use Distances



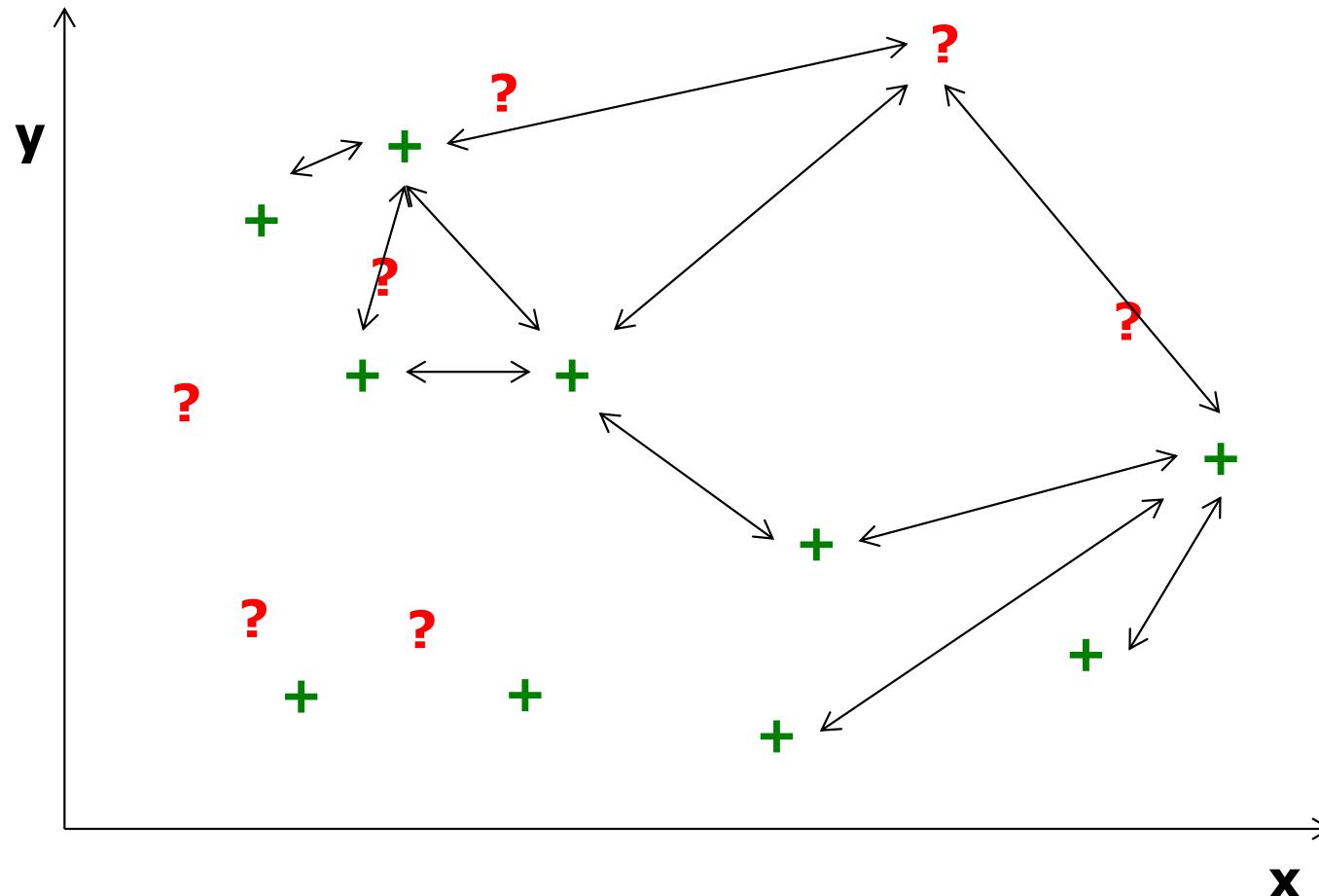
How to use neighbors? Compare Distances



Anomaly if distance to nearest neighbor n compared to
distance from n to nearest neighbor is above threshold

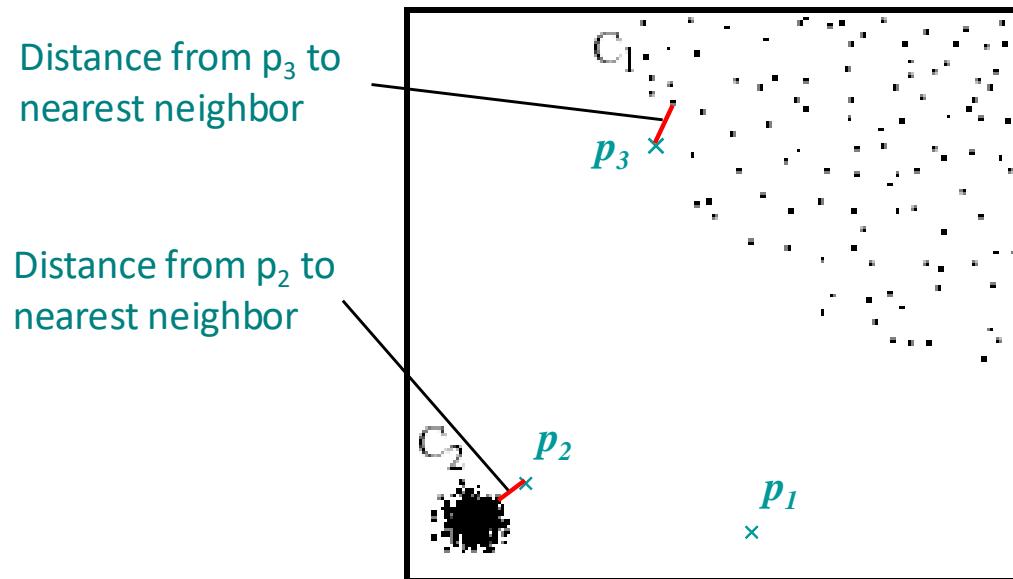
How to use neighbors?

Compare Densities



Anomaly if density is substantially lower than neighbor's density,
or average density

A famous one: Local Outlier Factor

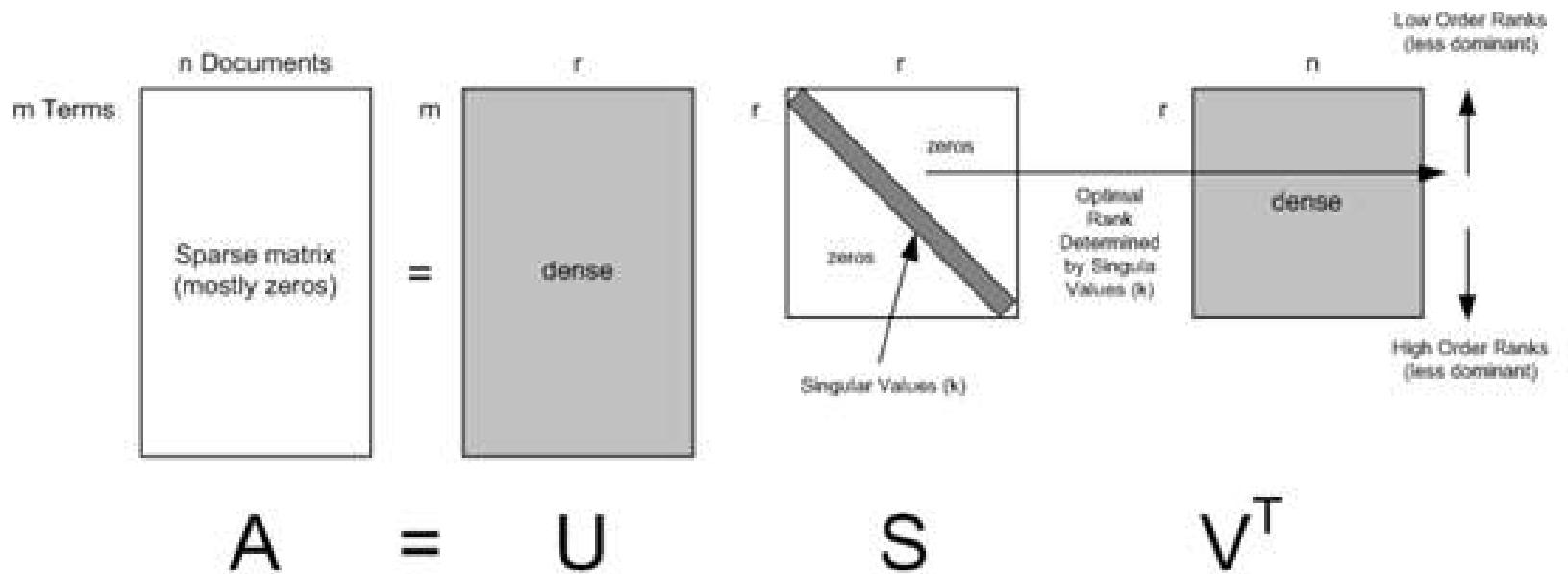


In the *NN* approach, p_2 is not considered as outlier, while the *LOF* approach find both p_1 and p_2 as outliers

NN approach may consider p_3 as outlier, but LOF approach does not

Spectral Techniques

- Analysis based on Eigen decomposition of data
- PCA (Principal Component Analysis)
 - Orthogonal transformation to reduce dimension
 - Most data patterns are captured by the several principal vectors

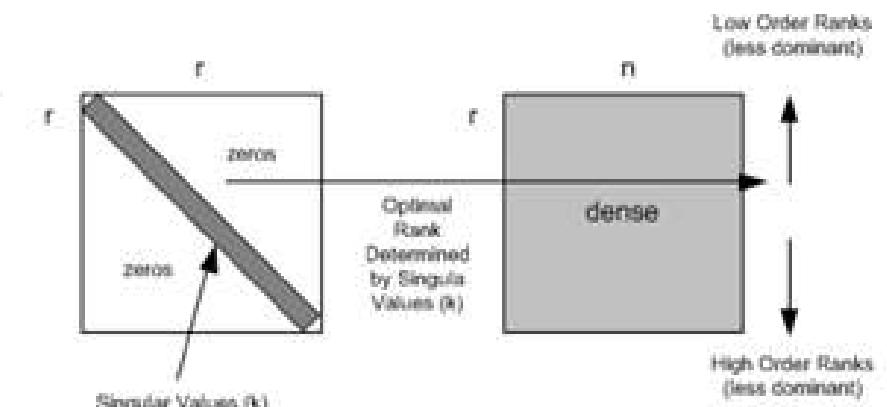


Spectral Techniques

- Analysis based on Eigen decomposition of data
- PCA (Principal Component Analysis)
 - Orthogonal transformation to reduce dimension
 - Most (linear) data patterns are captured by several principal vectors

How to use this for anomaly detection?

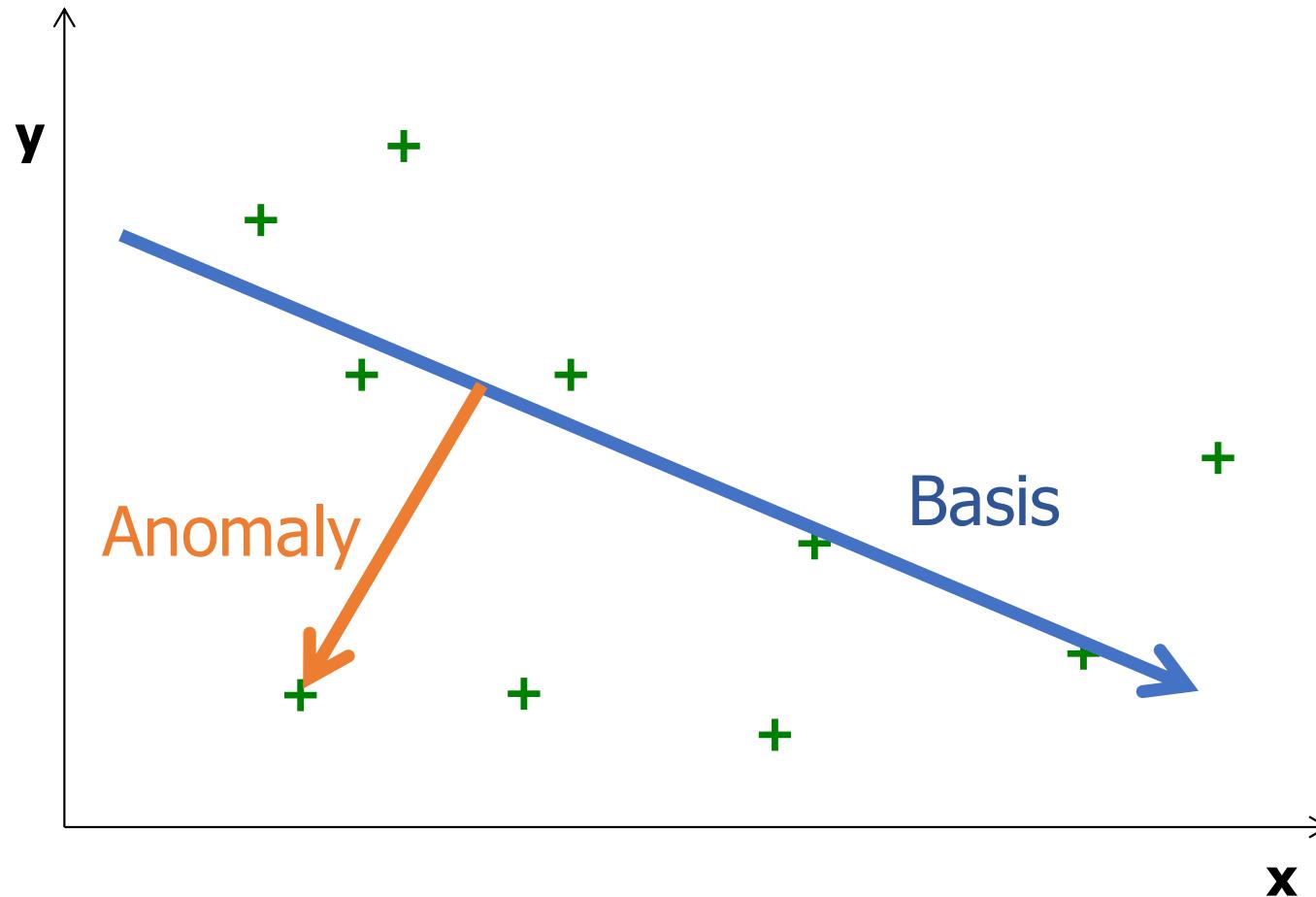
$$A = U S V^T$$



Spectral Techniques

- Key Idea
 - Find combination of attributes that capture bulk of variability
 - Reduced set of attributes can explain normal data well
 - **But do not necessarily explain the outliers**
- Several methods use Principal Component Analysis
 - Top few principal components capture variability in normal data
 - Smallest principal component should have constant values
 - Outliers have variability in the smallest component

How does PCA work?



It finds a new set of features, that explain most of the variance
Datapoints that vary in unexplained dimensions are anomalous

How does PCA work? (intuition)

value	amount1	amount2
4	2000	2500
4	2250	2750
8	3000	3500
10	3300	3800
11	3400	3900
6	3500	3900
8	3800	4300
10	4000	4500
14	4100	4600
12	4200	4700

How does PCA work? (intuition)

value	amount1	amount2
4	2000	2500
4	2250	2750
8	3000	3500
10	3300	3800
11	3400	3900
6	3500	3900
8	3800	4300
10	4000	4500
14	4100	4600
12	4200	4700

How to reconstruct amount2 from amount1?

How does PCA work? (intuition)

value	amount1
4	2000
4	2250
8	3000
10	3300
11	3400
6	3500
8	3800
10	4000
14	4100
12	4200

Reconstruction is perfect
after reducing the space

$$\text{amount2} = \text{amount1} + 500$$

How does PCA work? (intuition)

For two new points:

value	amount1	amount2
6	3600	4100
8	2800	3000

How does PCA work? (intuition)

For two new points:

value	amount1	amount2
6	3600	4100
8	2800	3000

After reduction:

value	amount1
6	3600
8	2800

How does PCA work? (intuition)

For two new points:

value	amount1	amount2
6	3600	4100
8	2800	3000

After reduction:

value	amount1
6	3600
8	2800

After reconstruction:

value	amount1	amount2
6	3600	4100
8	2800	3300

How does PCA work? (intuition)

For two new points:

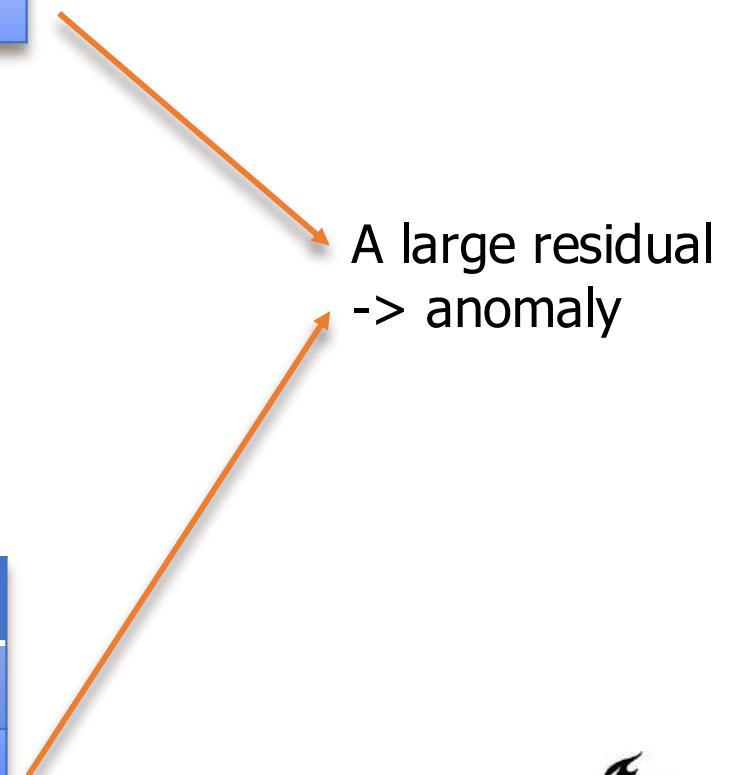
value	amount1	amount2
6	3600	4100
8	2800	3000

After reduction:

value	amount1
6	3600
8	2800

After reconstruction:

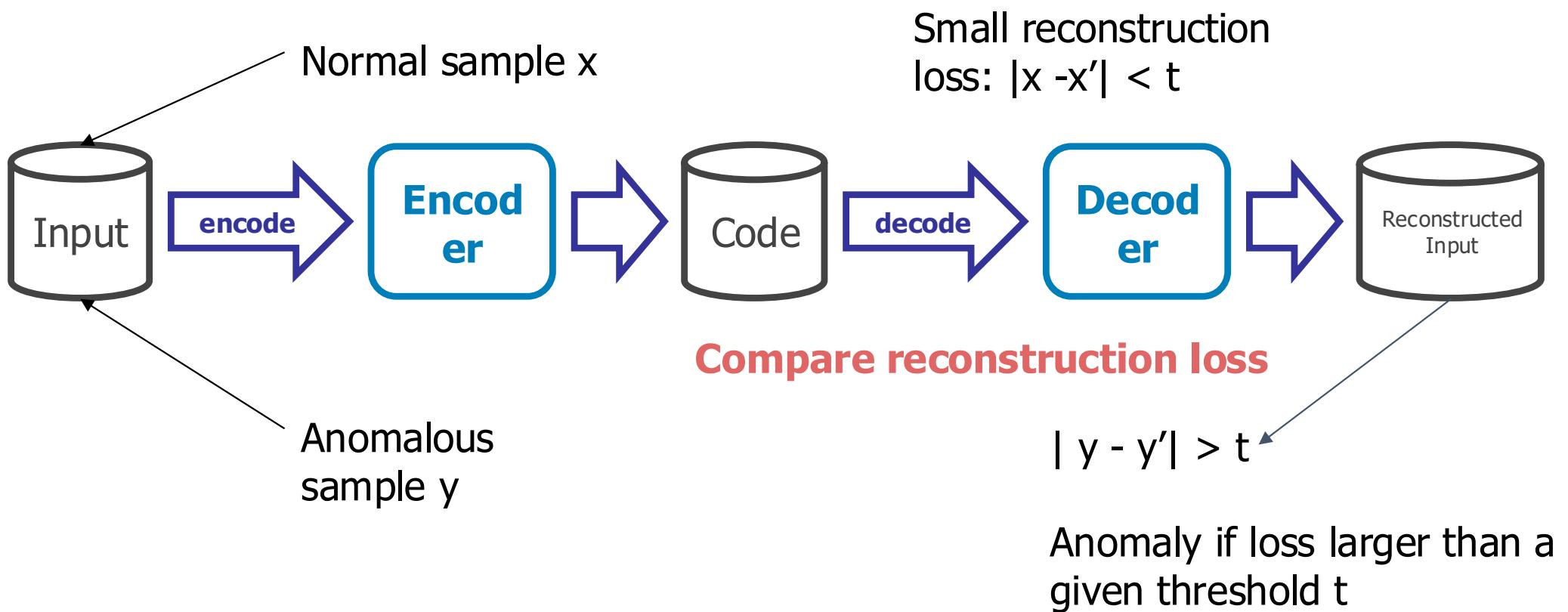
value	amount1	amount2
6	3600	4100
8	2800	3300



Spectral Techniques

- Advantage
 - Useful for modeling feature interactions (multivariate data)
 - Computationally efficient (use graphic cards!)
- Disadvantage
 - Based on the assumption that anomalies and normal instances are distinguishable in the reduced space
 - Does not take context into account
 - PCA is sensitive to outliers...

Autoencoder for anomaly detection



Implementations:

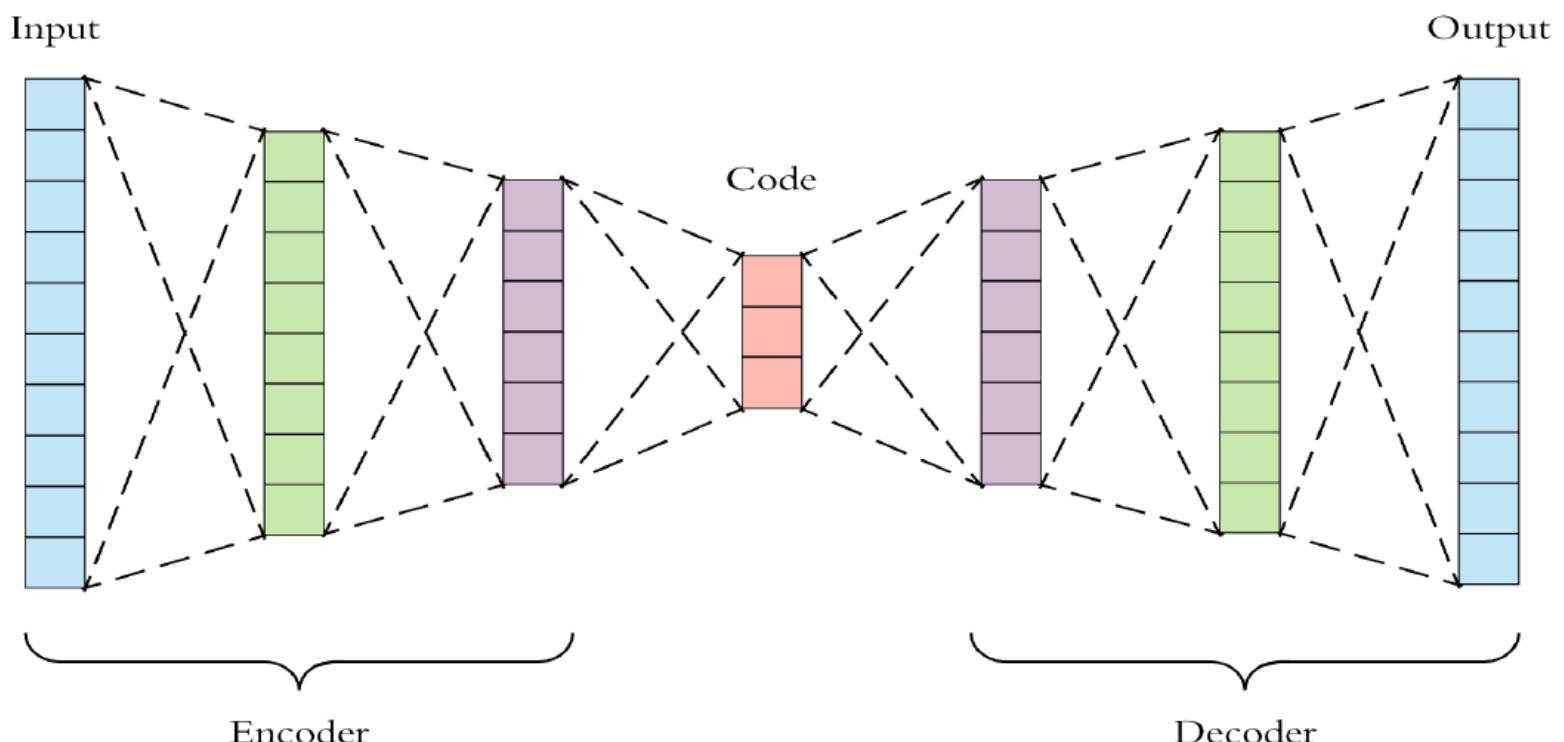
<https://towardsdatascience.com/extreme-rare-event-classification-using-autoencoders-in-keras-a565b386f098>

<https://towardsdatascience.com/lstm-autoencoder-for-extreme-rare-event-classification-in-keras-ce209a224cfb>

Further reading on autoencoder types: <https://medium.com/datadriveninvestor/deep-learning-different-types-of-autoencoders-41d4fa5f7570>

Deep Learning: use auto-encoder or GAN or ... (think of PCA)

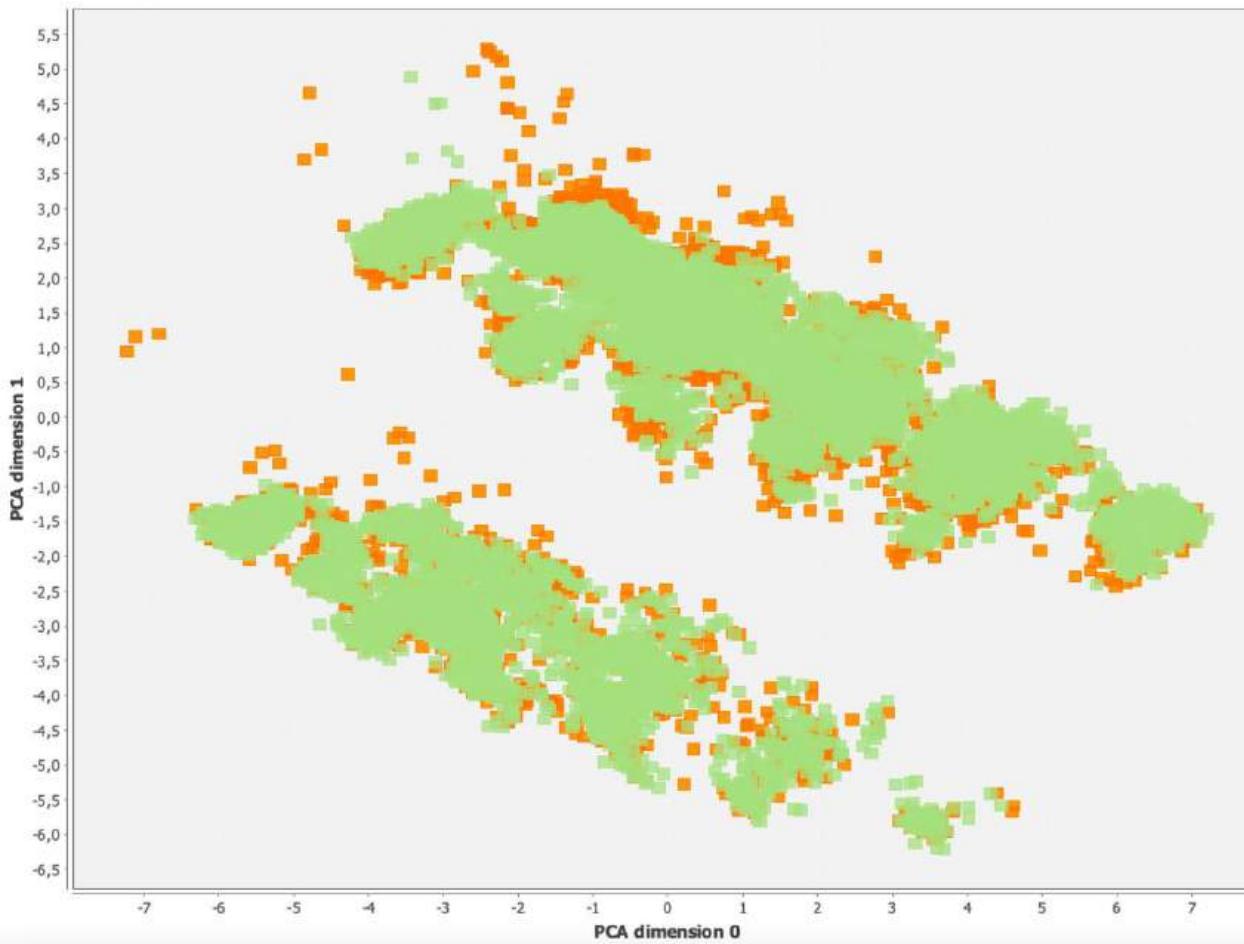
- Find a low dimensional representation of normal data
- Abnormal data will map to abnormal code
- Reconstruction from that code will be bad
 - The model has only learned to map back to normal data



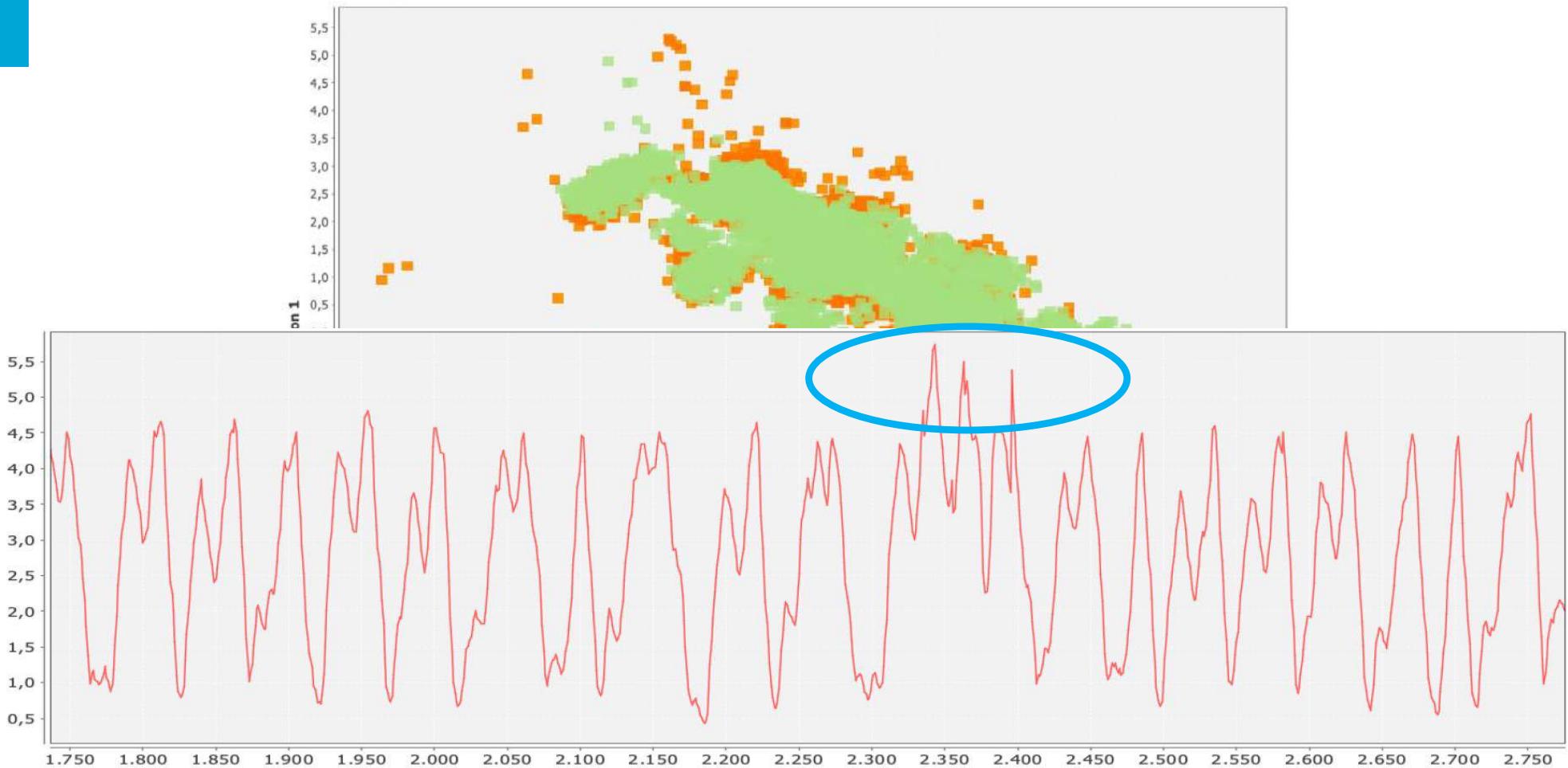
Today

- 3 types of anomalies
 - Point – *point x is strange*
 - Contextual – *point x is strange given set Y*
 - Collective – *set Y is strange*
- Modeling context
 - Sliding windows – *to model sequential context*
 - Time Warping distance – *to align sequences*
- Popular anomaly detection methods
 - Classification – *minimize positive space*
 - Nearest Neighbor – *density or distance*
 - Spectral – *remember small code (dimensions)*

How to evaluate performance?



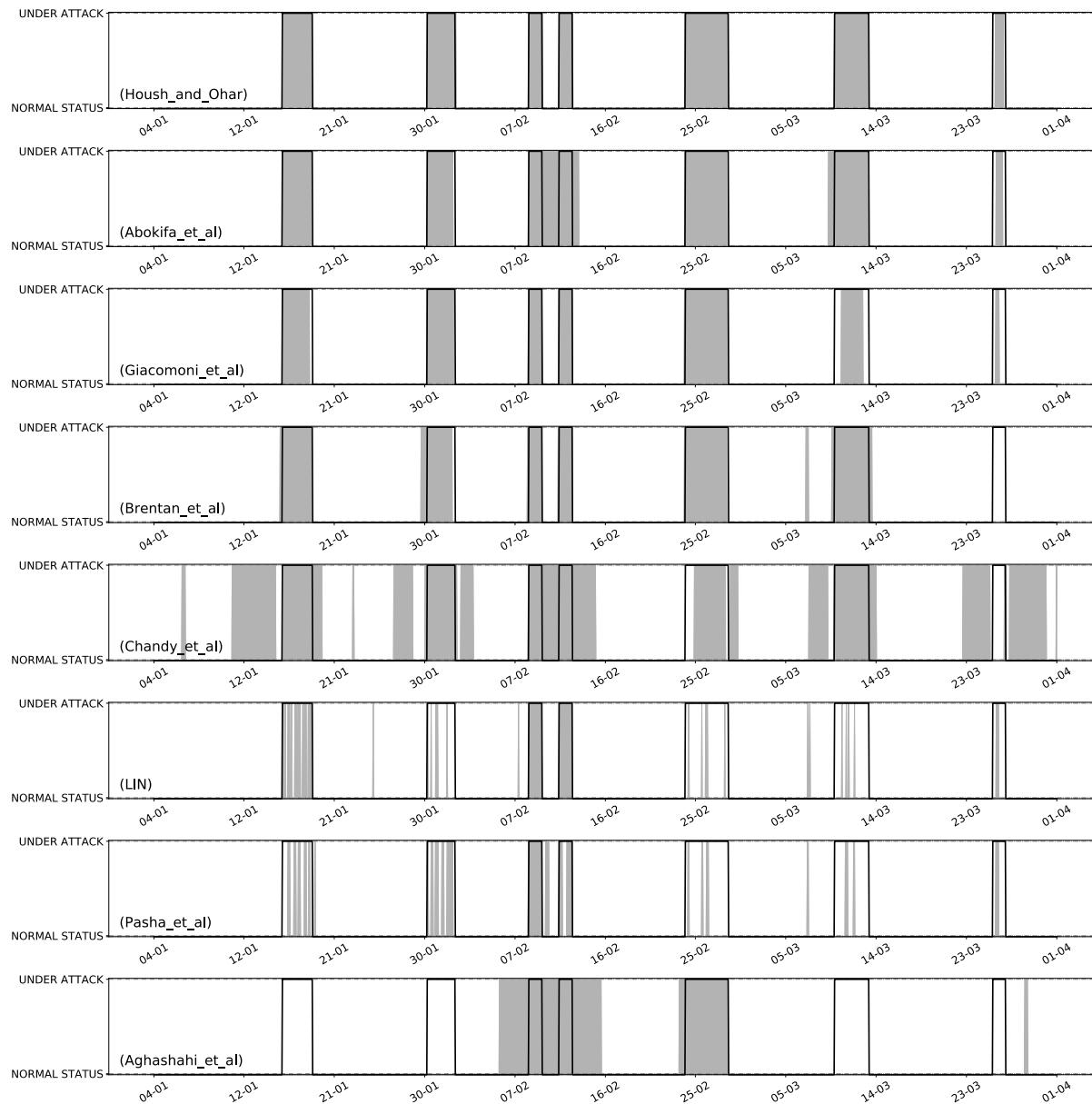
How to evaluate performance?



Anomaly detection is hard to evaluate

- Often little/no information on positives
 - Rely on quality of clustering, no clear quality measure exists
 - Good distances are often hard to find
- Anomalies are usually time periods instead of points
 - An attack starts and stops
 - Is every detection within that period a true positive?
- Unclear how to count positives
 - Many alarms are raised in a few seconds, is this a single positive?
 - Should we group them over time?
 - ..
- For the Kaggle challenge, we simply use point-based F1...

An attempt in BATADAL



Exam material

- Concepts

- Three types of anomalies & how to detect them
- Distance/density based anomalies
- Matching shapes in sequences

- Skills

- Sliding windows
- Minimize positive space
- Temporal correlation

- Algorithms

- Dynamic Time Warping
- Isolation Forest

CSE2525 Distances & Dimensionality

How to compute and apply them

Today

- Two parts:

1. Distances
2. Dimensionality

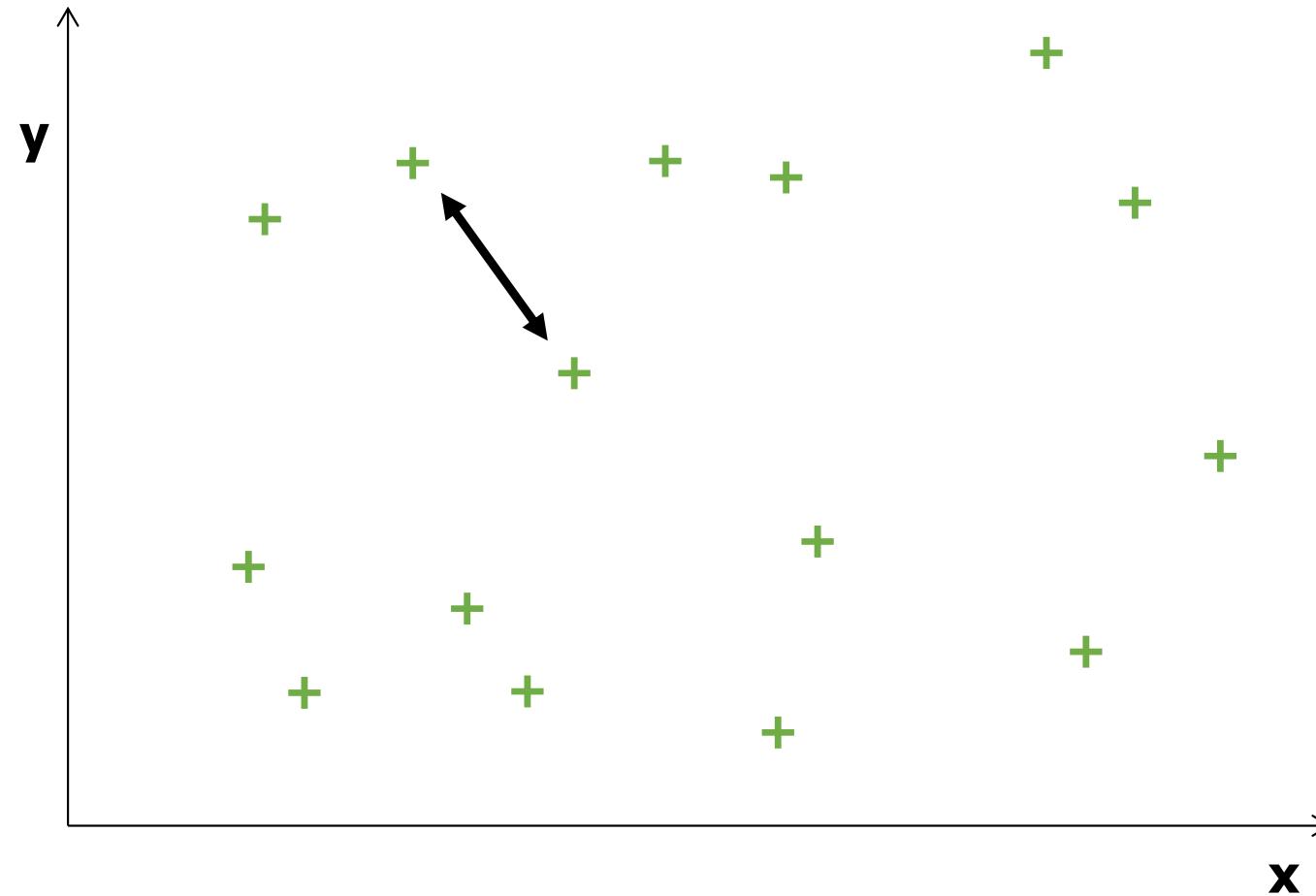
- By Nergis Tömen, assistant professor in deep learning for computer vision, neuromorphic computing and neuroscience
- biologically-inspired computer vision, deep network models of biological circuits



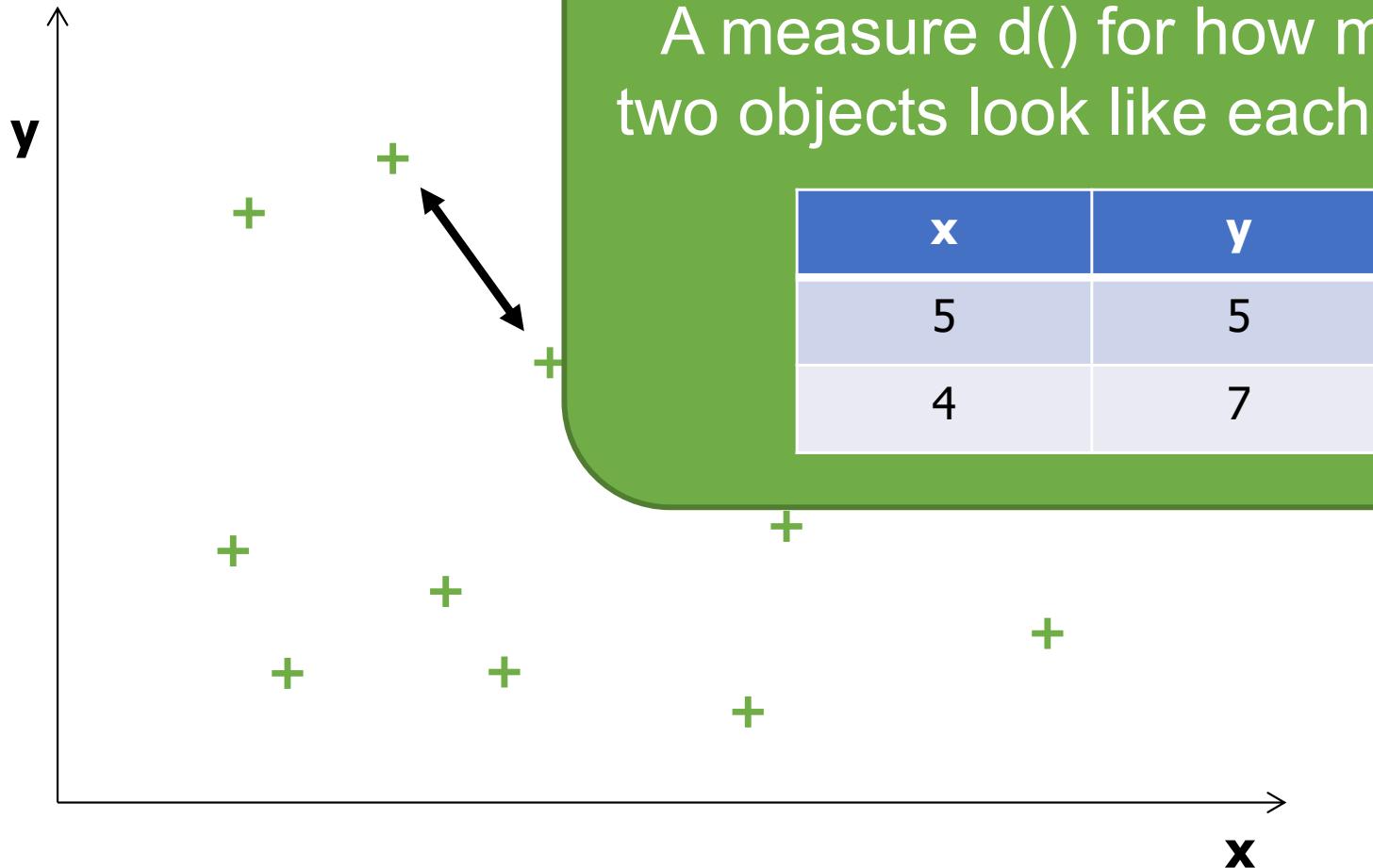
What is a distance?



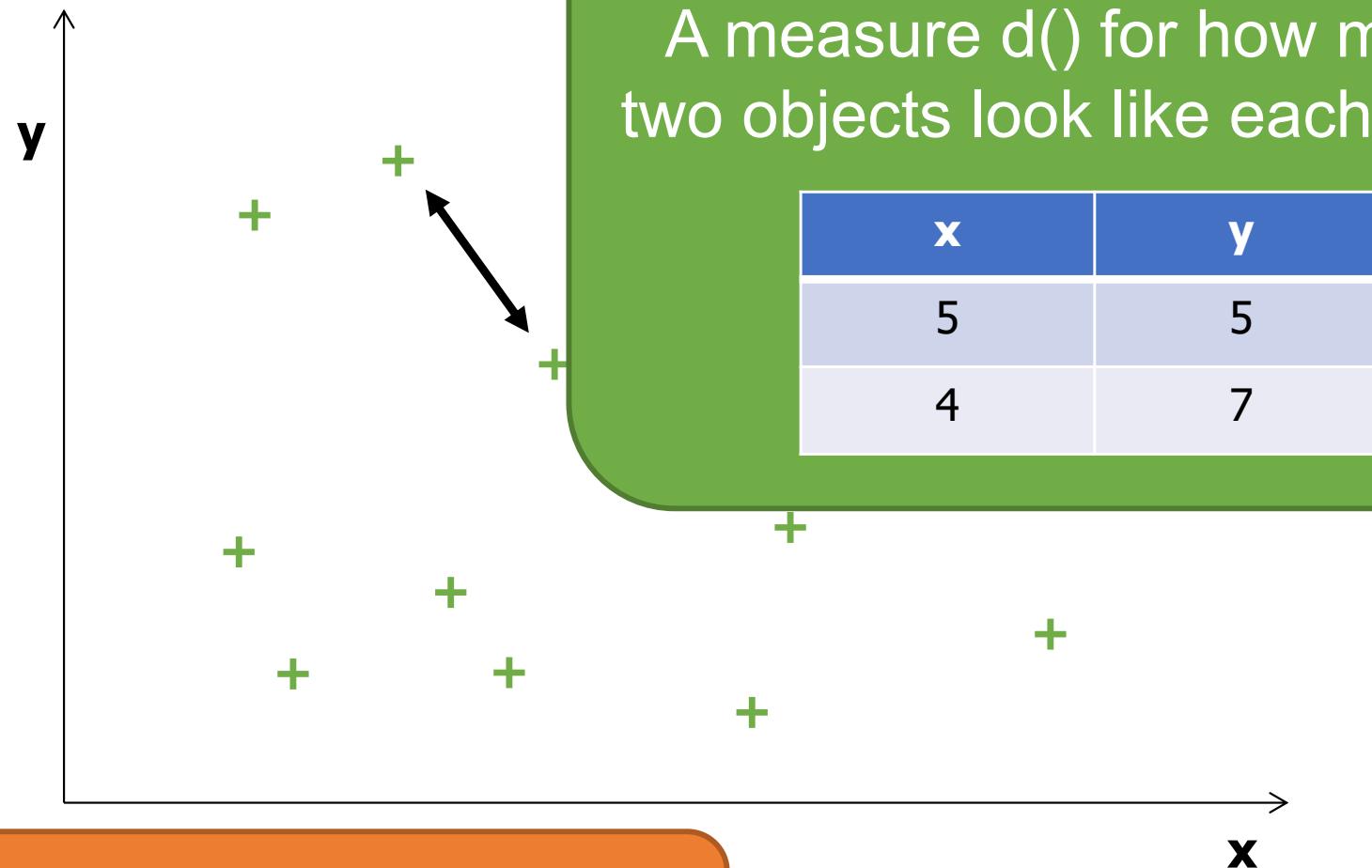
What is a distance?



What is a distance?

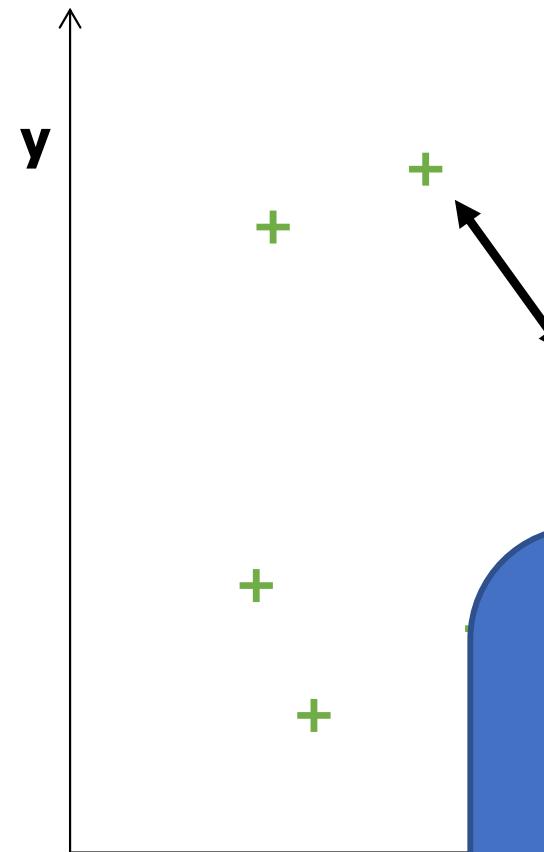


What is a distance?



What should their distance be?

What is a distance?



A measure $d()$ for how much two objects look like each other

x	y
5	5
4	7

Max: 2

Min: 1

Sum: 3

Euclidean: 2.24

Cosine: 0.035

Overlap: 2

..

What should their dista

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:
 - Non-negativity: $d(A, B) \geq 0$

Distance cannot be negative

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

- Non-negativity:

$$d(A, B) \geq 0$$

- Identity:

$$d(A, B) = 0 \text{ iff } A = B$$

Distance is only 0 between
identical objects

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

- Non-negativity:

$$d(A, B) \geq 0$$

- Identity:

$$d(A, B) = 0 \text{ iff } A = B$$

- Symmetry:

$$d(A, B) = d(B, A)$$

Direction does not matter for
the distance

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

- Non-negativity:

$$d(A, B) \geq 0$$

- Identity:

$$d(A, B) = 0 \text{ iff } A = B$$

- Symmetry:

$$d(A, B) = d(B, A)$$

- Triangle inequality:

$$d(A, C) \leq d(A, B) + d(B, C)$$

Taking a detour through a
third point is never shorter

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

- Non-negativity:

$$d(A, B) \geq 0$$

- Identity:

$$d(A, B) = 0 \text{ iff } A = B$$

- Symmetry:

$$d(A, B) = d(B, A)$$

- Triangle inequality:

$$d(A, C) \leq d(A, B) + d(B, C)$$

Taking a detour through a
third point is never shorter

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

- Non-negativity: $d(A, B) \geq 0$

- Identity of indiscernibles: $d(A, B) = 0 \iff A = B$

These properties make $d()$ “mathematically nicer”. Gives closure properties, makes problems easier (can cut parts of the search space), etc.

It is important to know whether an algorithm uses these properties!

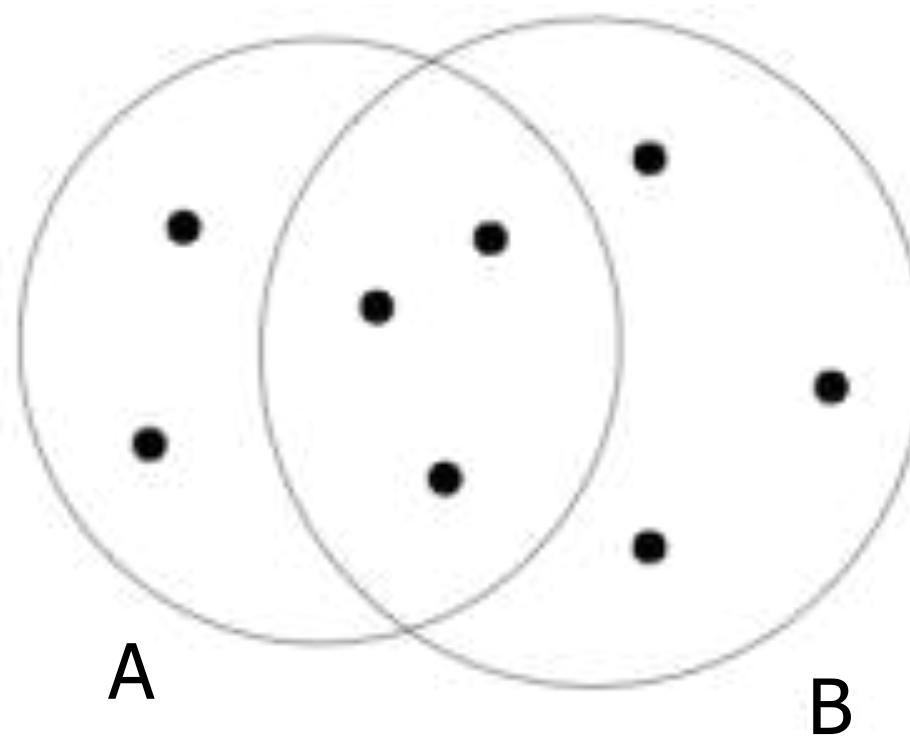
Overview

- Distances between
 - sets
 - Jaccard
 - vectors
 - Euclidean, L_p norms
 - Cosine
 - Hamming
 - ISOMAP
 - distributions –
 - Kullback-Leibler, Jensen-Shannon
 - sequences
 - DTW
 - Edit (Levenshtein)

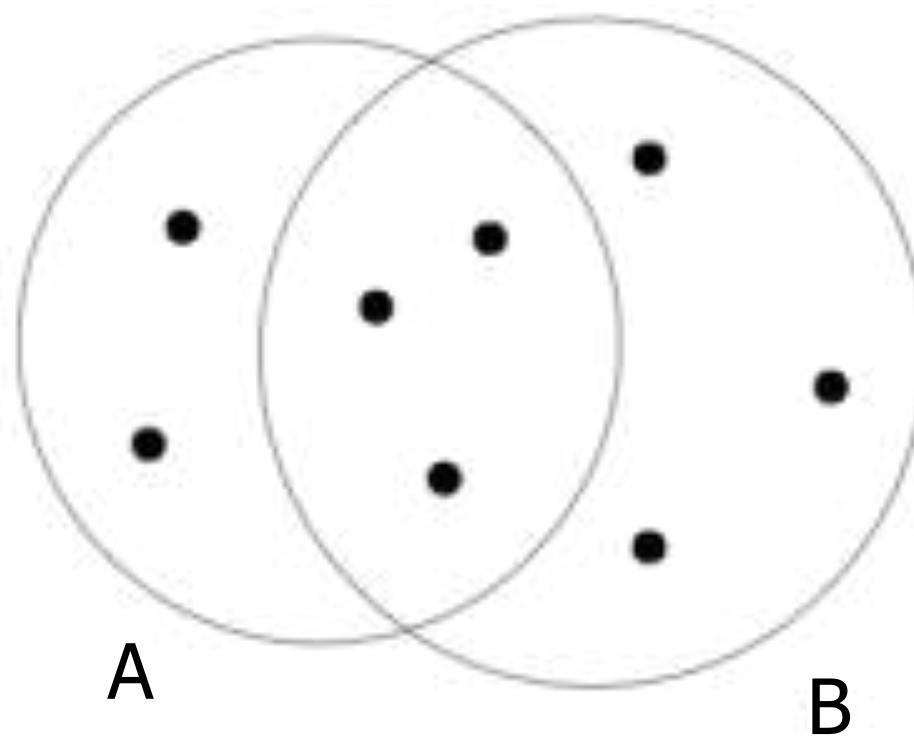
Jaccard similarity

- The Jaccard similarity measures the overlap between two sets A and B

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$



Jaccard similarity example



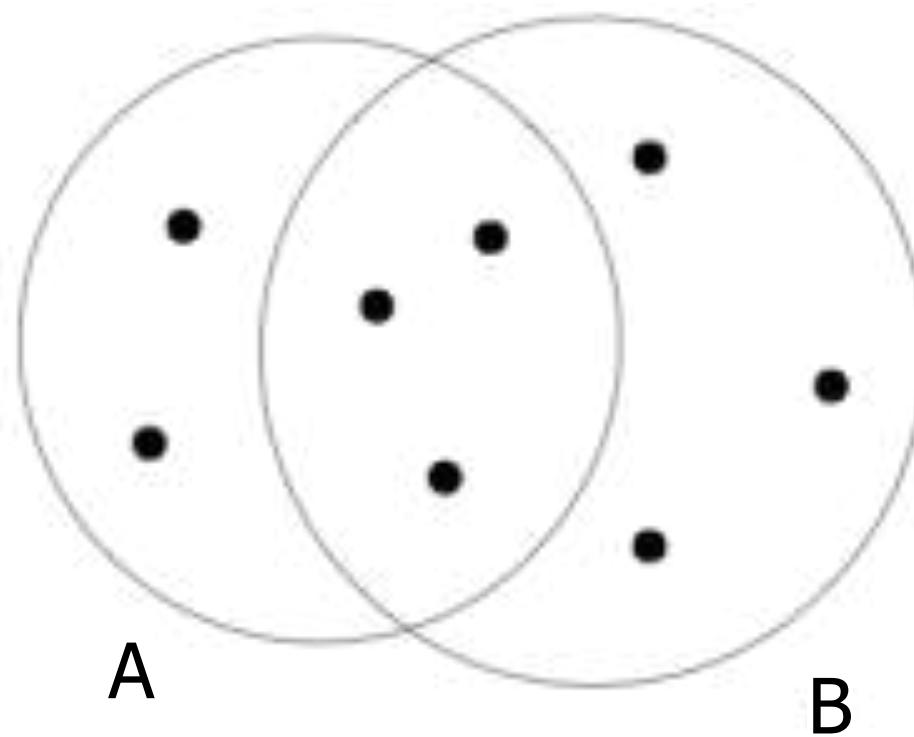
$$\frac{|A \cap B|}{|A \cup B|}$$

3/8

Jaccard distance

- The Jaccard distance is $1 - \text{Jaccard similarity}$

$$d(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}$$



Jaccard similarity: Applications

- Finding out whether students plagiarized their reports:
 - If you copy parts of text, Jaccard similarity will be high even after many edits
- Making sure that Google does not show duplicate results:
 - Remove items that have a Jaccard similarity of nearly one
- Finding similar pieces of malware:
 - Collect the presence of system calls, cluster using Jaccard

Jaccard similarity: collaborative filtering

- Making recommendations on what to buy on Amazon (collaborative filtering):
 - Find users that bought similar items
 - Recommend items those users bought but you didn't
 - Combine with clustering
- Movie ratings
 - Movie similarity vs. user similarity
 - Ratings --> binary

Euclidean distance

- The Euclidean distance between two vectors \mathbf{a} and \mathbf{b} is defined as:

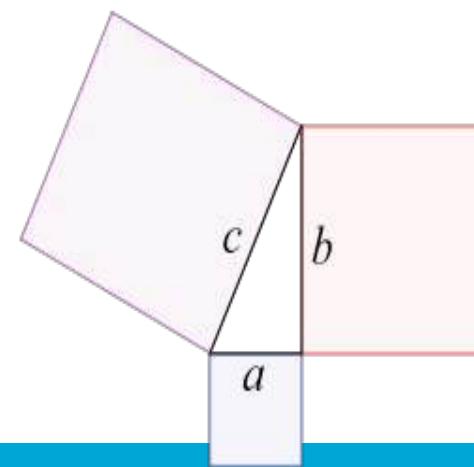
$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{d=1}^D (a_d - b_d)^2} = \|\mathbf{a} - \mathbf{b}\|$$

Euclidean distance

- The Euclidean distance between two vectors \mathbf{a} and \mathbf{b} is defined as:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{d=1}^D (a_d - b_d)^2} = \|\mathbf{a} - \mathbf{b}\|$$

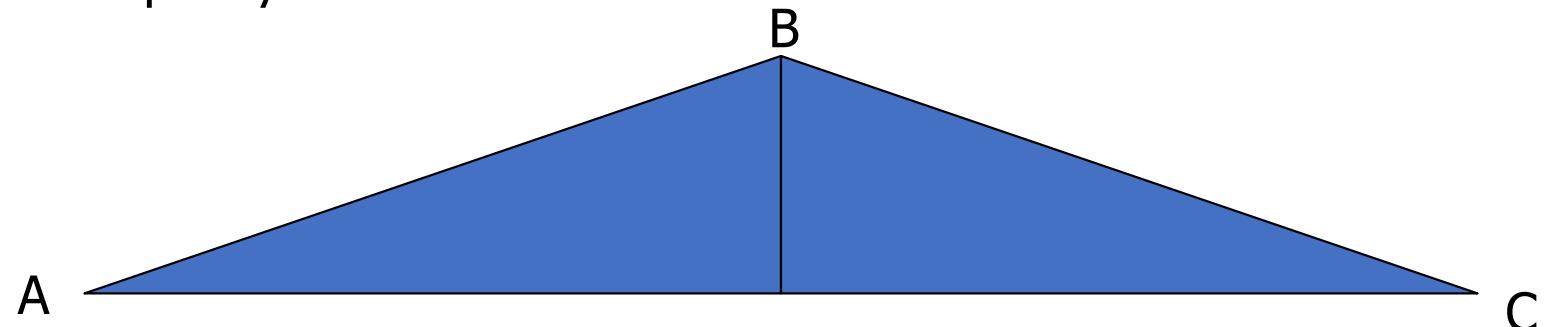
- This expression is a generalization of the Pythagorean theorem:



Euclidean distance

- Is the Euclidean distance a distance metric?

- Non-negativity: $\forall d : (a_d - b_d)^2 \geq 0$
- Identity: $\forall d : (a_d - b_d) = 0$ iff $a_d = b_d$
- Symmetry: $\forall d : (a_d - b_d)^2 = (b_d - a_d)^2$
- Triangle inequality:



Euclidean distance

- When to use Euclidean distance?

1. Continuous data
2. Not too many columns
3. Magnitude matters

Euclidean distance

- When to use Euclidean distance?

1. Continuous data
2. Not too many columns
3. Magnitude matters

Keep in mind: when used on maps it is a lower bound on the actual (traveling) distance!



Euclidean distance

- When to avoid Euclidean distance?
 1. When dealing with high dimensional data
 - Curse of dimensionality
 2. When dealing with sparse data
 - Zero values are treated as any other
 3. When magnitudes matter
 - It is not scale invariant! ($\text{dist}(a,b) \neq \text{dist}(a*c, b*c)$ for constant c)
 4. In the case of discrete data

Euclidean distance

- When to avoid Euclidean distance?
 1. When dealing with high dimensional data
 - Curse of dimensionality
 2. When dealing with sparse data
 - Zero values are treated as equal
 3. When magnitudes matter
 - It is not scale invariant
 4. In the case of discrete data

Note that it is often the default distance

Do not fall into the trap to using defaults without thinking!

How to decide on a distance?

- It is often useful to consider distance invariants/properties
- Euclidean is
 - a metric
 - invariant to translation
 - dependent on scale
 - sensitive to outliers
 - sensitive to dimensionality
 - sensitive to sparsity
 - ...

How to decide on a distance?

There is no clear recipe, do what makes sense:

- It is often Euclidean
- Euclidean distances
 - a reasonable choice
 - invariant to scaling
 - deceptively simple
 - sensitive to outliers
 - sensitive to noise
 - ...
- 1. Compute a distance D for your data
- 2. Test whether near neighbors should be close to each other
- 3. Perform manipulations on your data that D should be invariant to
- 4. Test for this invariance
- 5. Perform manipulations on your data that D should be sensitive to
- 6. Test for this sensitivity

The curse of dimensionality

- Demo notebook

In general: L_p -norm – generalized Minkowski distance

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}.$$

L_0 = Non-zeros

L_1 = Manhattan distance

L_2 = Euclidean distance

L_{∞} = Maximum distance

In general: L_p -norm – generalized Minkowski distance

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}.$$

Optional weight w_i

L_0 = Non-zeros

L_1 = Manhattan distance

L_2 = Euclidean distance

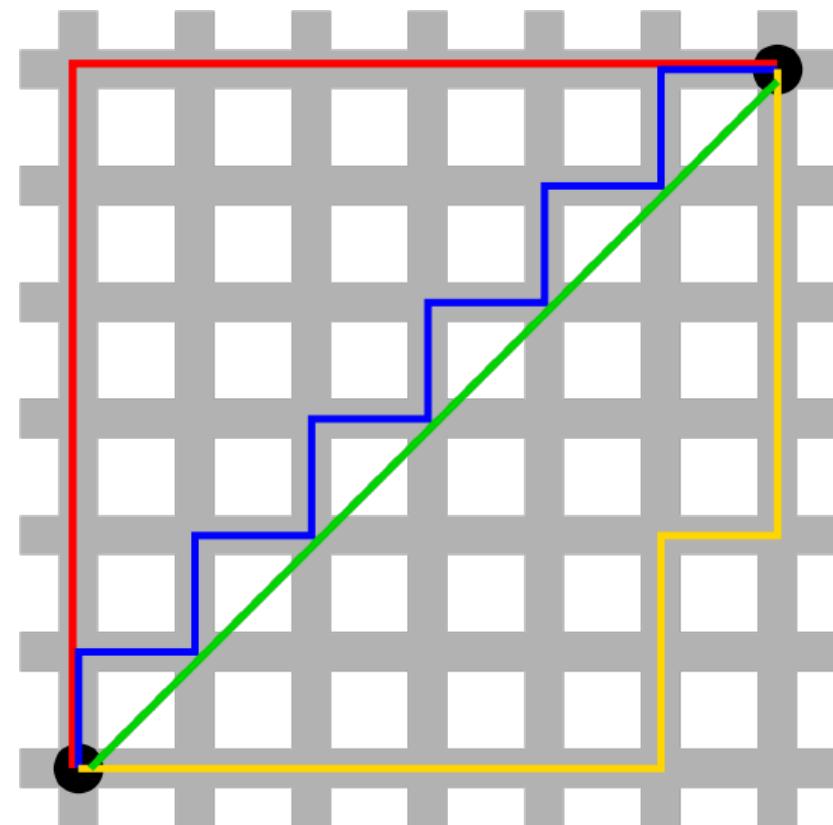
L_{∞} = Maximum distance

Manhattan distance

- aka, absolute difference:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- When would you use this?

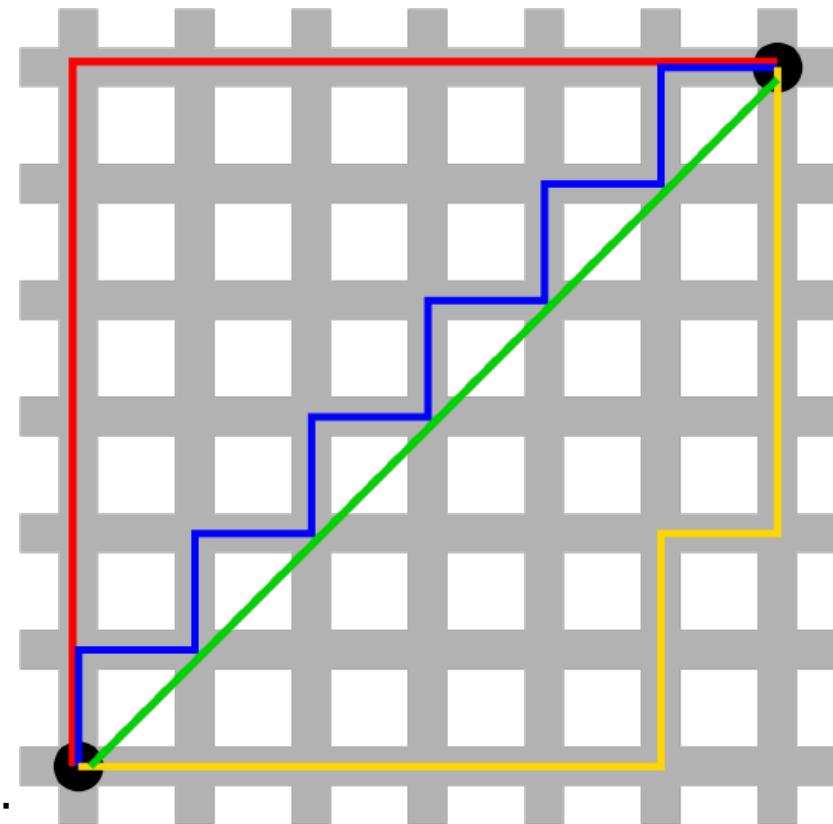


Manhattan distance

- aka, absolute difference:

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

- When would you use this?
 - When outliers are a problem
 - When features are incomparable
 - When you have many dimensions...
- *Keep in mind: when used on maps it is an upper bound on the actual (traveling) distance!*



Maximum (Chebyshev) distance

- aka chessboard distance –

- $d(x, y) = \max_i |x_i - y_i|$

- When to use max distance?

- Not so clear...
 - When it makes sense
 - When features are incomparable

- When not?

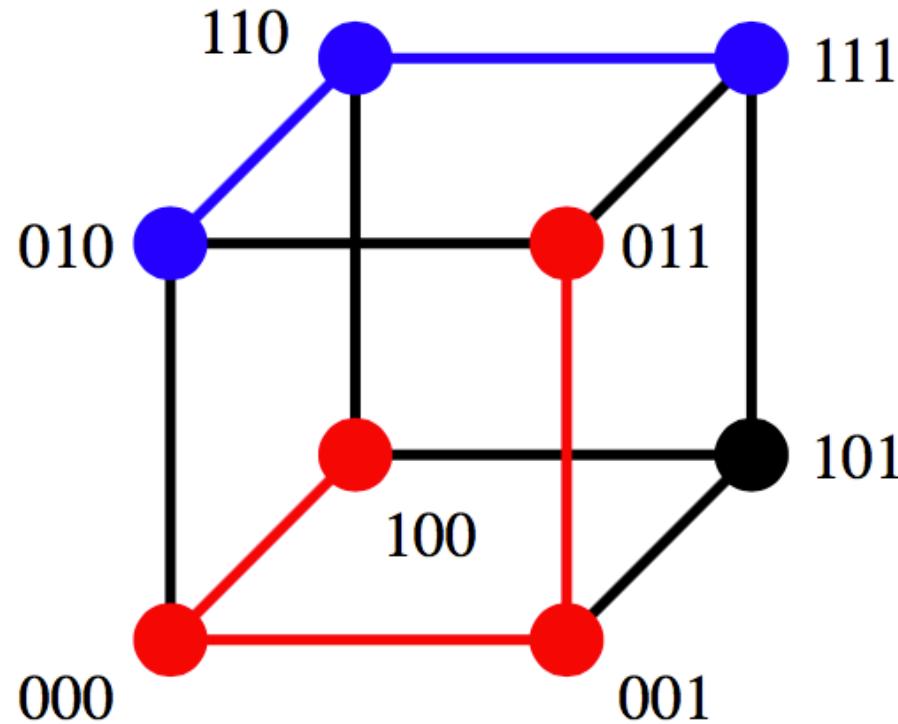
- Outliers are clearly a problem
 - ...

- Keep in mind: when used on maps it is an (extreme) lower bound on the actual (traveling) distance!*

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1	1	1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

Hamming distance

- Hamming distances compare two bit (or character) sequences:
 - Hamming distance counts the number of dissimilar bits (or characters)



Hamming distance: Example

- Hamming distance between bit strings A and B

A	1	1	0	0	1	0	0	1
B	1	0	0	1	1	1	0	1

Hamming distance = 3

Match-based distances

- The problem of high dimensionality is for a large part due to different noise in different dimensions
- This effect becomes less with lower values for p
- Another approach to lessen this effect is to only perform local distance computations per dimension
- We divide each dimension into k equal sized bins with range [m, n], then if two points are in the same bin for dimension d compute:

$$PSelect(\overline{X}, \overline{Y}, k_d) = \left[\sum_{i \in \mathcal{S}(\overline{X}, \overline{Y}, k_d)} \left(1 - \frac{|x_i - y_i|}{m_i - n_i} \right)^p \right]^{1/p}$$

- otherwise return 0

Match-based distances

- The problem of high dimensionality is for a large part due to different noise in different dimensions
- This effect becomes less with lower values for p
- Another approach to lessen this effect is to only perform local distance computations per dimension
- We divide each dimension into k equal sized bins with range $[m, n]$, then if two points are in the same bin for dimension d compute:

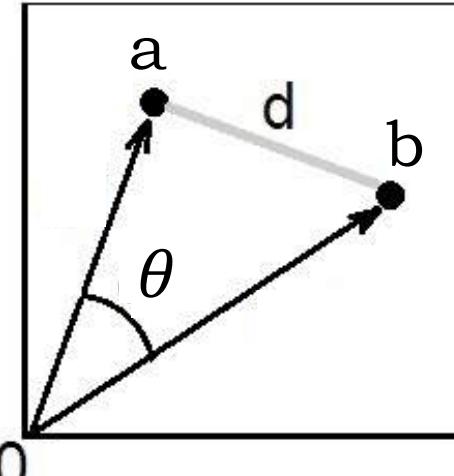
$$PSelect(\overline{X}, \overline{Y}, k_d) = \left[\sum_{i \in \mathcal{S}(\overline{X}, \overline{Y}, k_d)} \left(1 - \frac{|x_i - y_i|}{m_i - n_i} \right)^p \right]^{1/p}$$

- otherwise return 0

Essentially an extension of Hamming distance, we will see similar ideas in sketching/hashing...

Cosine distance

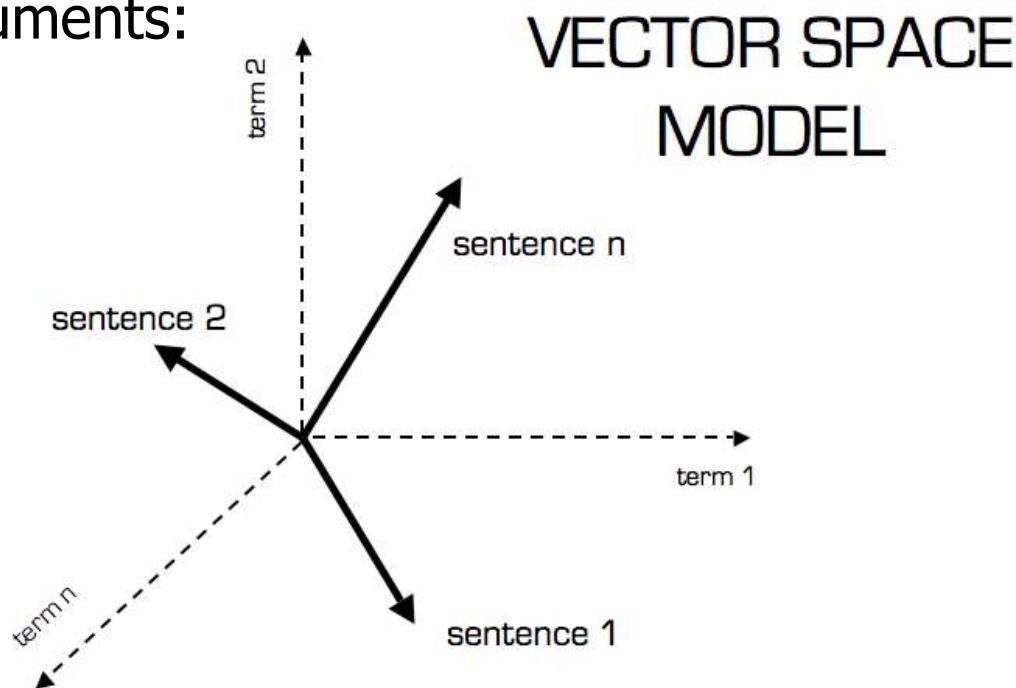
- The cosine distance measures the angle between two vectors:

$$d(\mathbf{a}, \mathbf{b}) = 1 - \cos(\theta) = 1 - \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$$

$$\frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

only looks at direction of vectors,
not at the length of these vectors!

Cosine distance: Applications

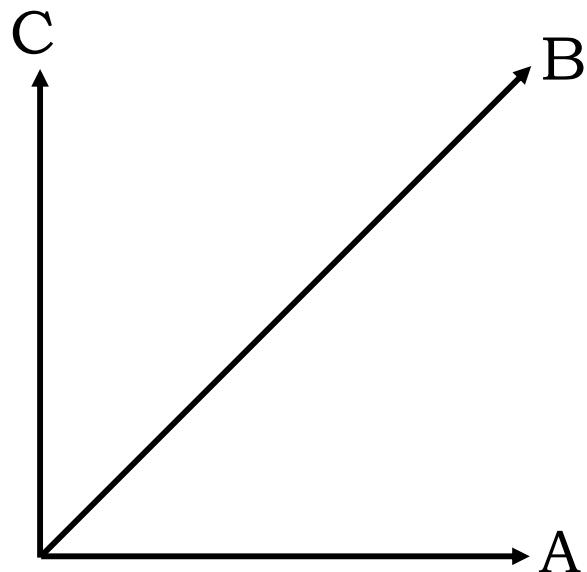
- Cosine distance is often used to compare bag-of-word vectors of documents:



- Main advantage: distance does not depend on length of documents! scale invariant!
- But is is not translation invariant!

Cosine distance

- The cosine distance violates identity of indiscernibles and triangle inequality
- Example of a violation of the triangle inequality:



$$d(A, B) = 1 - \cos\left(\frac{1}{4}\pi\right) = 1 - \frac{1}{2}\sqrt{2}$$

$$d(B, C) = 1 - \cos\left(\frac{1}{4}\pi\right) = 1 - \frac{1}{2}\sqrt{2}$$

$$d(A, C) = 1 - \cos\left(\frac{1}{2}\pi\right) = 1$$

$$1 > 2 - \sqrt{2}$$

Angular distance*

- So technically, the cosine “distance” is not a proper distance metric
- The triangle inequality violation may be resolved by taking the arc-cosine:

$$d(\mathbf{a}, \mathbf{b}) = \arccos(\cos(\theta)) = \arccos\left(\frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}\right)$$

* To confuse you, the book refers to the angular distance as the cosine distance!

Angular distance*

- So technically, the cosine “distance” is not a proper distance metric
- The triangle inequality violation may be resolved by taking the arc-cosine:

$$d(\mathbf{a}, \mathbf{b}) = \arccos(\cos(\theta)) = \arccos\left(\frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}\right)$$

- Looking at the counterexample for cosine:

$$d(A, B) = 1 - \cos\left(\frac{1}{4}\pi\right) = 1 - \frac{1}{2}\sqrt{2}$$

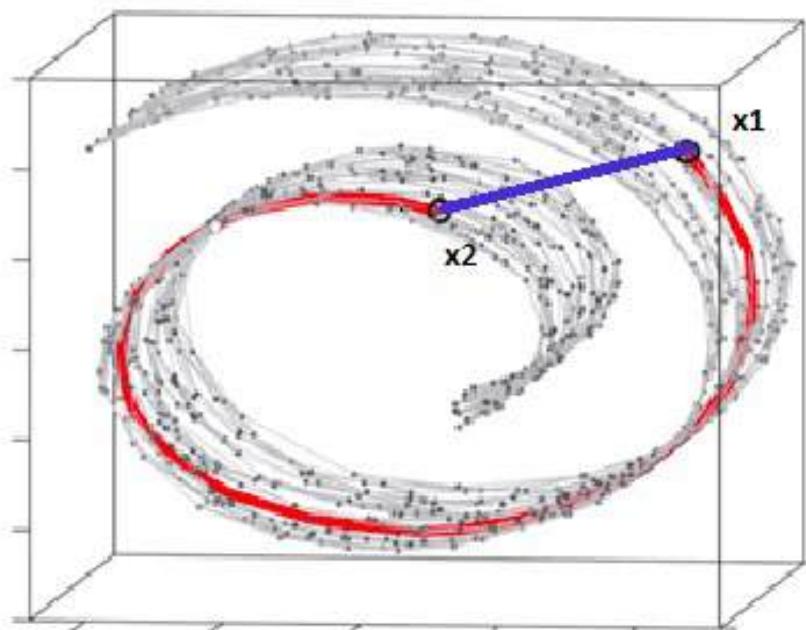
$$d(B, C) = 1 - \cos\left(\frac{1}{4}\pi\right) = 1 - \frac{1}{2}\sqrt{2}$$

$$d(A, C) = 1 - \cos\left(\frac{1}{2}\pi\right) = 1$$

- obviously, $\frac{1}{4} + \frac{1}{4} = \frac{1}{2}$

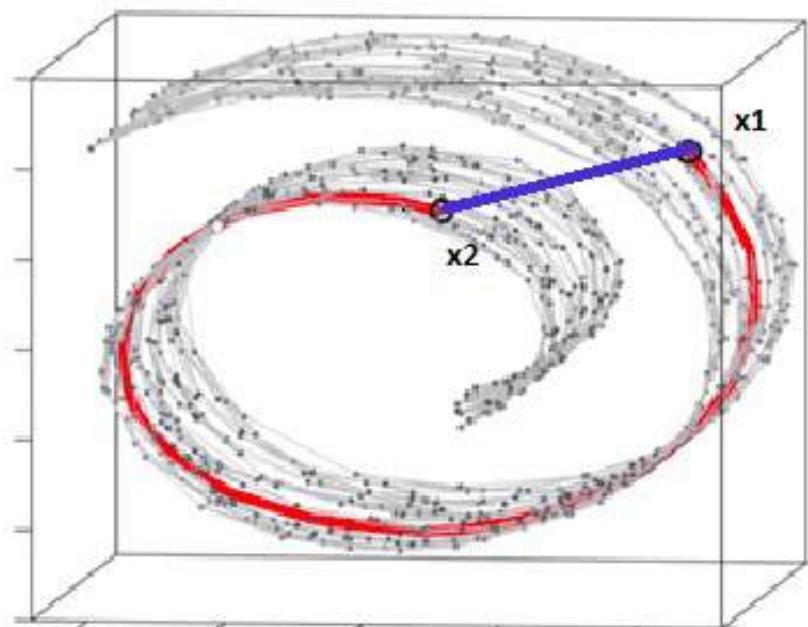
ISOMAP distance

- Compute kNN for each data point
- Construct weighted graph: weights = distance
- Set:
 - ISOMAP distance(p,q) = weight of shortest path from p to



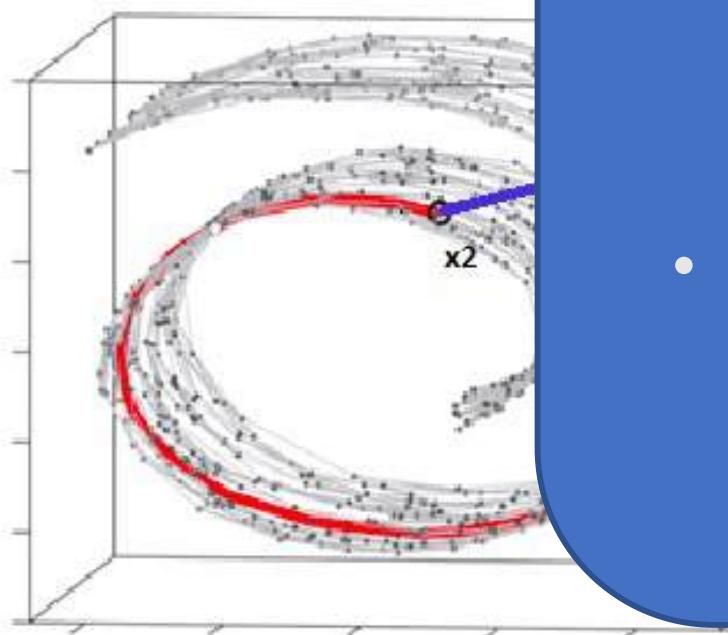
ISOMAP distance

- Compute kNN for each data point
- Construct weighted graph: weights = distance
- Set:
 - ISOMAP distance(p,q) = weight of shortest path from p to



ISOMAP distance

- Compute kNN for each point
- Construct weighted graph
- Set:
 - ISOMAP distance(p, q)



An “improvement” of any existing metric, cannot cross parts of the space without data (roads)

But other issues:

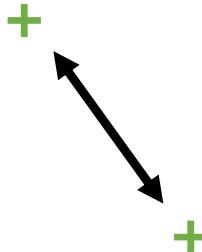
- Expensive
- Disconnected components
- Very sensitive to noise
- ...



Overview

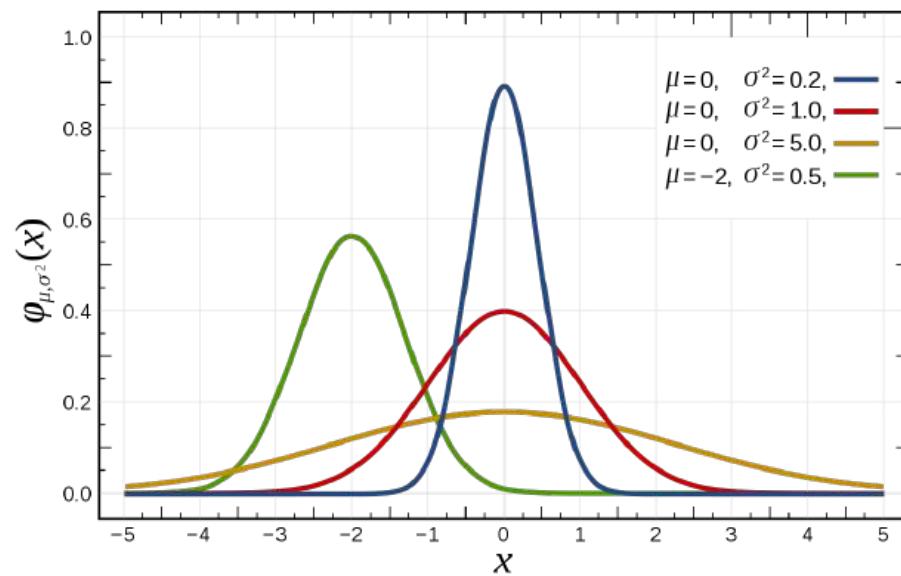
- Distances between
 - sets
 - Jaccard
 - vectors
 - Euclidean, L_p norms
 - Cosine
 - Hamming
 - ISOMAP
 - distributions –
 - Kullback-Leibler, Jensen-Shannon
 - sequences
 - DTW
 - Edit (Levenshtein)

Which distance to use and why?



- Demo notebook + slido
- **Slido.com #2053073**

Statistical distances



Kullback-Leibler divergence (not distance!)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

- Measures “surprise” when using Q instead of P
- The expected value of the loglikelihood ratio between P and Q
- The number of extra bits needed to represent samples from P using Q
- Information gain or relative entropy
- ...

Kullback-Leibler divergence (not distance!)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

- In practice, compute KL for:

P	0.01	0.1	0.39	0.5
Q	0.1	0.5	0.3	0.1

approx. $-0.03 - 0.23 + 0.15 + 1.16 = 1.05$

Kullback-Leibler divergence (not distance!)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

- In practice, compute **reverse KL** (from Q to P) for:

P	0.01	0.1	0.39	0.5
Q	0.1	0.5	0.3	0.1

Kullback-Leibler divergence (not distance!)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

- In practice, compute **reverse KL** (from Q to P) for:

P	0.01	0.1	0.39	0.5
Q	0.1	0.5	0.3	0.1

approx. $0.33 + 1.16 - 0.11 - 0.23 = 1.15$

Kullback-Leibler divergence (not distance!)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)},$$

Not symmetric!

- In practice,

P	0.01	0.1	0.39	0.5
Q	0.1	0.5	0.3	0.1

$$P||Q \quad \text{approx. } -0.03 - 0.23 + 0.15 + 1.16 = 1.05$$

$$Q||P \quad \text{approx. } 0.33 + 1.16 - 0.11 - 0.23 = 1.15$$

Kullback-Leibler divergence (not distance!)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

- In practice, largest mass matters most

P	0.01	0.1	0.39	0.5
Q	0.1	0.5	0.3	0.1

$$P||Q \quad \text{approx. } -0.03 - 0.23 + 0.15 + 1.16 = 1.05$$

$$Q||P \quad \text{approx. } 0.33 + 1.16 - 0.11 - 0.23 = 1.15$$

Kullback-Leibler divergence (not distance!)

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$$

$$D_{KL}(P||Q) = \int P(x) \log \frac{P(x)}{Q(x)} dx$$

- In practice, largest mass matters most

P	0.01	0.1	0.39	0.5
Q	0.1	0.5	0.3	0.1

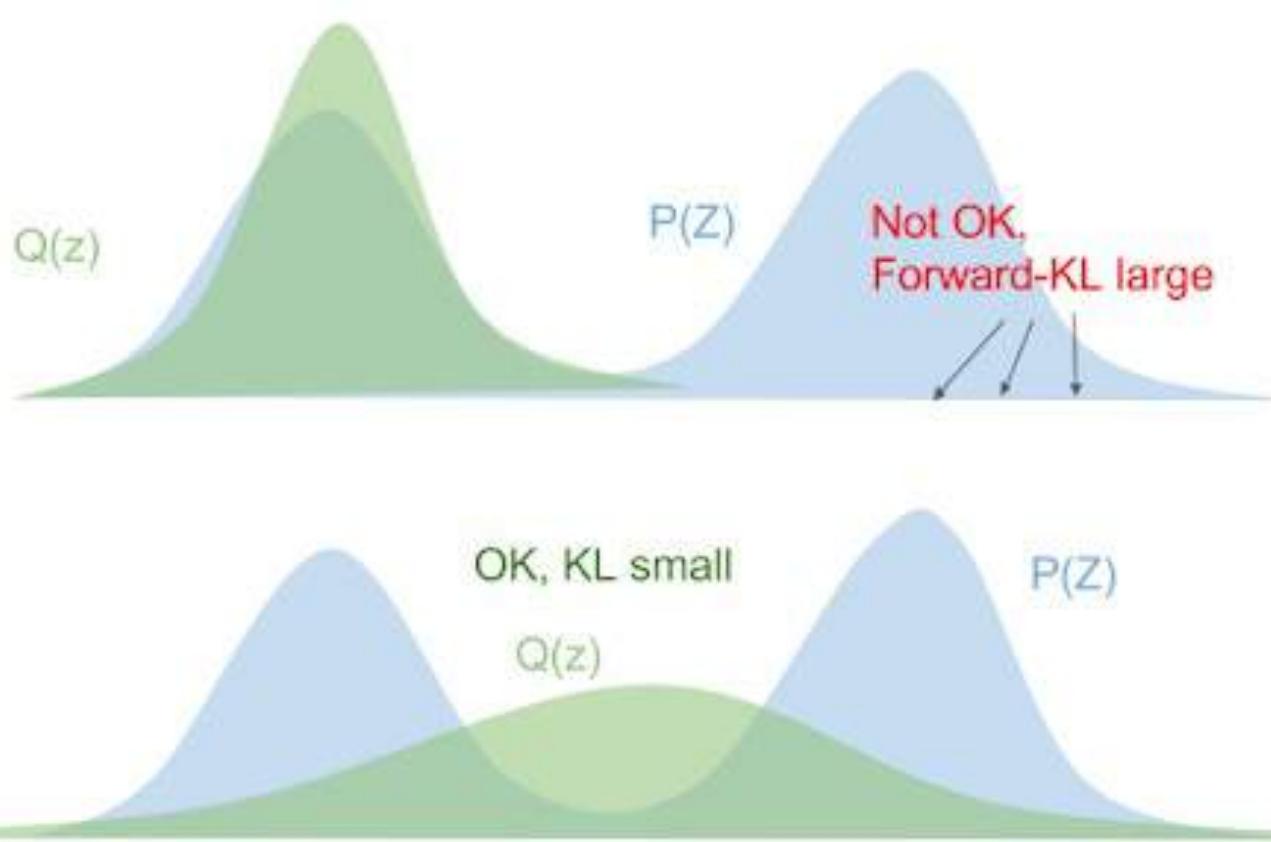
$$P||Q \quad \text{approx. } -0.03 - 0.23 + 0.15 + 1.16 = 1.05$$

$$Q||P \quad \text{approx. } 0.33 + 1.16 - 0.11 - 0.23 = 1.15$$

small differences can have large effect

Forward $\text{KL}(P \parallel Q)$

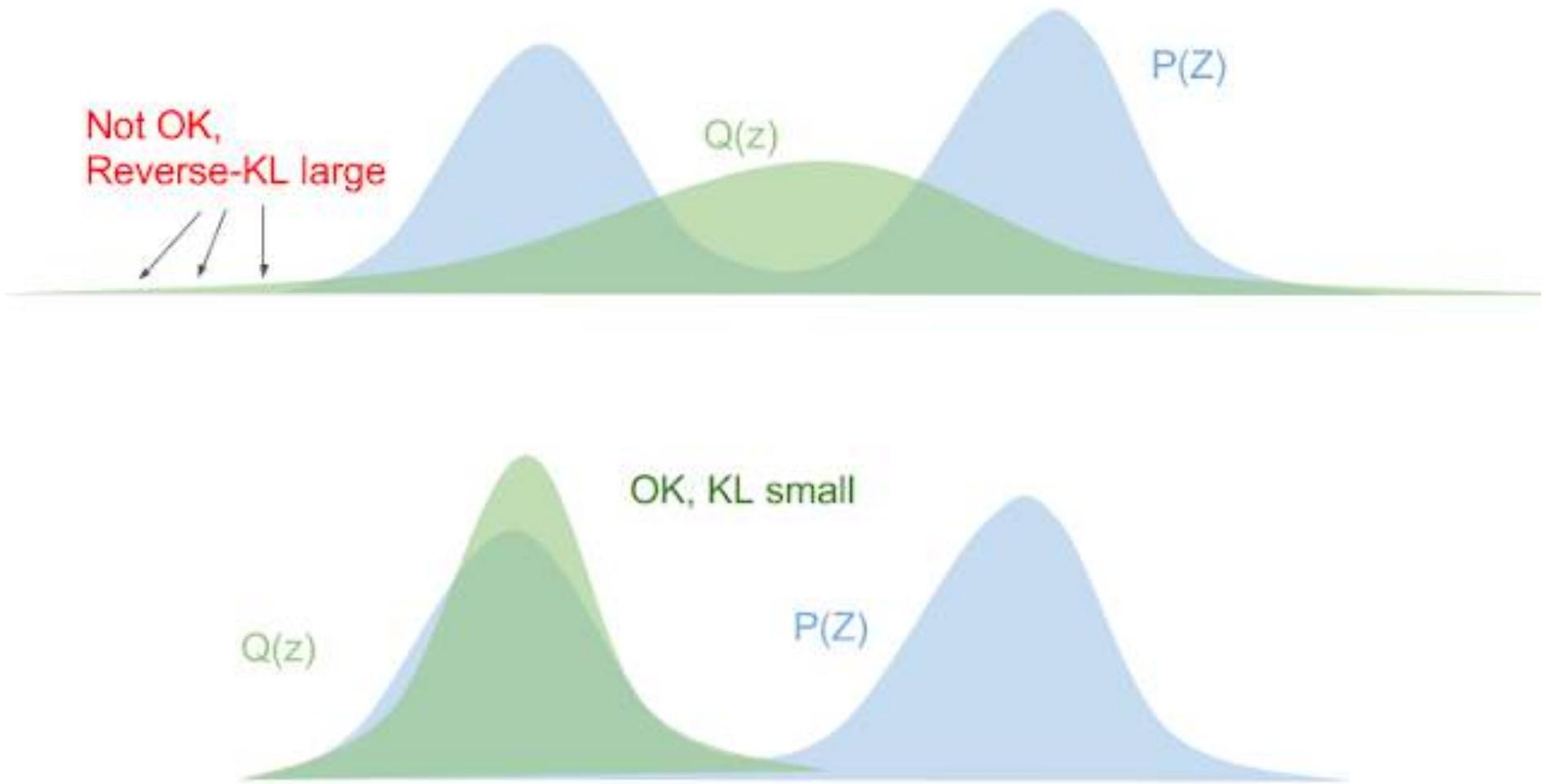
when P is high, Q should be high



source: <https://blog.evjang.com/2016/08/variational-bayes.html>

Backward $\text{KL}(Q \parallel P)$

when Q is high, P should be high



source: <https://blog.evjang.com/2016/08/variational-bayes.html>

Jensen-Shannon divergence

- A symmetrized, smoothed version of KL divergence
 - $JS(P||Q) = \frac{1}{2} * KL(P||M) + \frac{1}{2} * KL(Q||M)$
 - with $M = \frac{1}{2} * (P + Q)$
- Unlike KL-divergence, it is a metric
- Moreover, it will not be infinite when P or Q becomes 0

In practice

- Computing statistical distance can be hard, we often:
 - Sample data X
 - Compute $P(X)$ and $Q(X)$
 - Normalize these to sum to 1
 - Compute KL-divergence
- To avoid 0-probabilities, and thus infinite distances, often some type of smoothing is applied
 - e.g., adding a small probability to all outcomes before normalization, or adding 1 to every count
 - as a consequence, statistical distances can depend more on the used normalization/smoothing than on the learning algorithm!

In practice

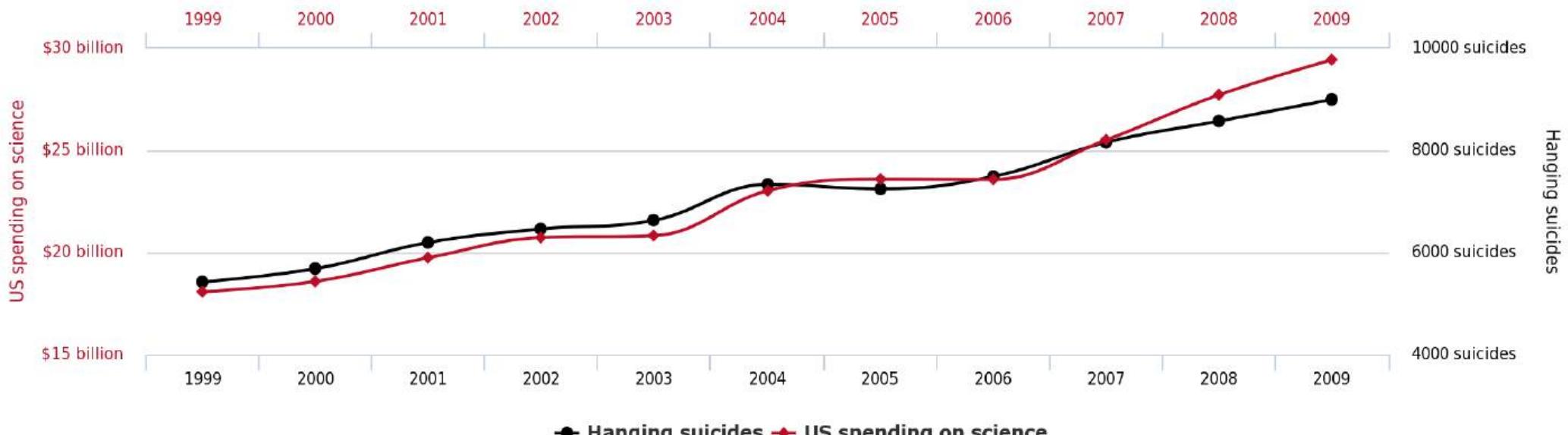
- Computing statistical distance can be hard, we often:
 - Sample data X
 - Compute $P(X)$ and $Q(X)$
 - Normalize these to sum to 1
 - Compute KL-divergence
- To avoid 0-probabilities, some type of smoothing:
 - e.g., adding a small value to all probabilities, or normalization, or
 - as a consequence, the used normalization

You can also compute regular distances between the obtained probabilities, but all statistical distances I know have problems..

Sequential distances

Shape often matters most

US spending on science, space, and technology
correlates with
Suicides by hanging, strangulation and suffocation



Are these series similar? Their y-ranges are very different...
check out: <http://www.tylervigen.com/spurious-correlations>

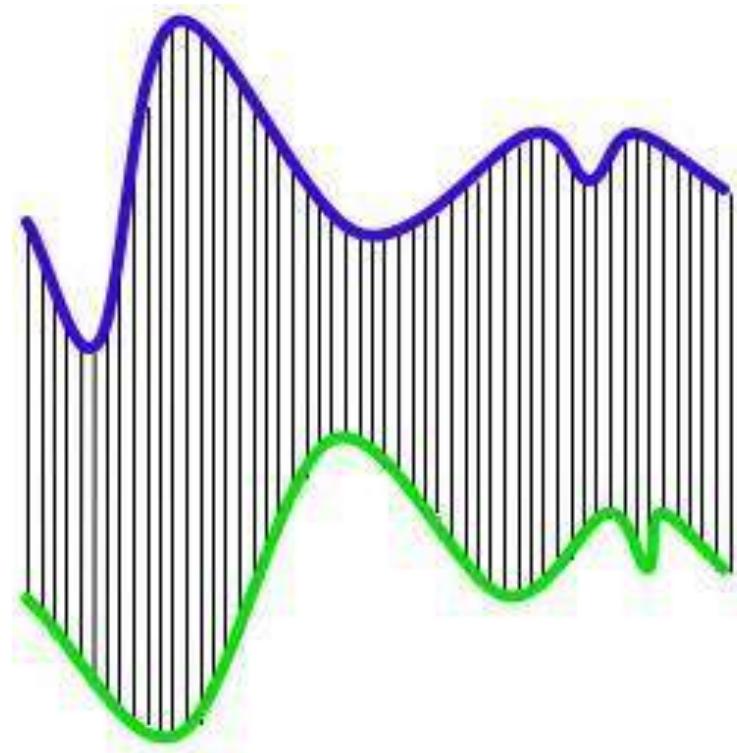
Scaling time series

- In order to compare shapes, we need to scale the series:

$$c'_i = \frac{c_i - \mu(C)}{\sigma(C)}$$

- for every data point c_i , where μ is the mean and σ is the standard deviation of the series
- After normalization, we can compute distances between sliding windows

Euclidean distance



(a)

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}.$$

Euclidean distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10
$(s1-s2)^2$	100	225	400	0	625	100	100	100

$$\text{Distance} = \sqrt{100+225+400+0+625+100+100+100} = 40.62$$

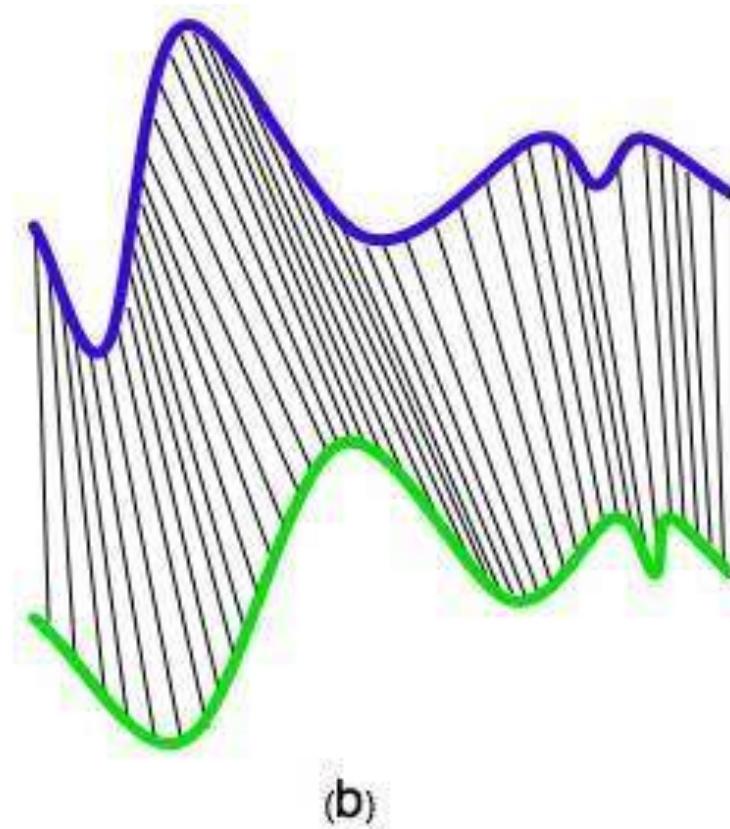
Euclidean distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10
(s1 - s2) ²	100	225	400	0	625	100	100	100

$$\text{Distance} = \sqrt{100+225+400+0+625+100+100+100} = 40.62$$

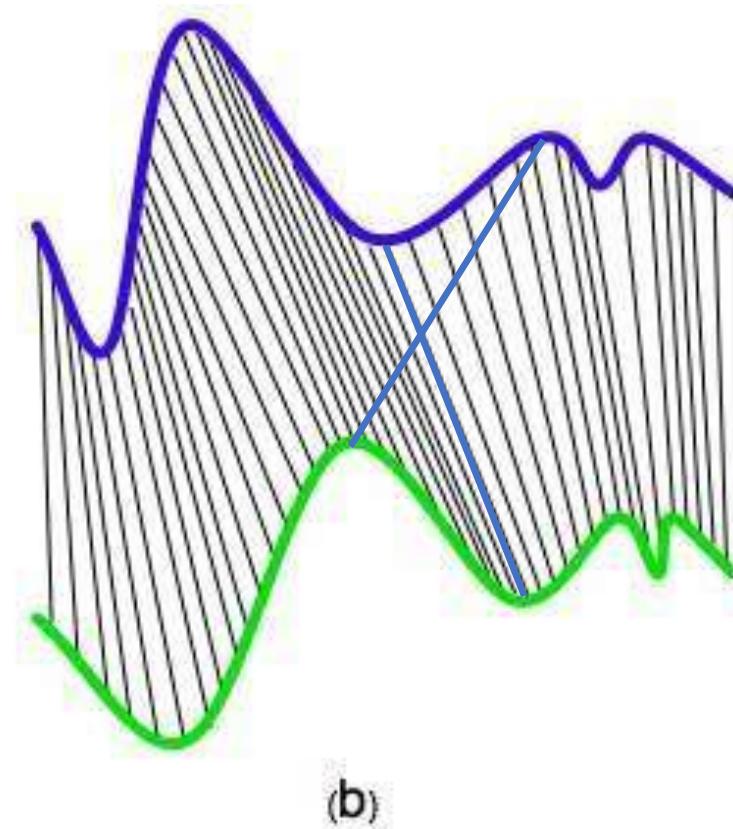
Signals seem out of phase...

Dynamic Time Warping or sequence alignment



Aligns time-series non-linearly in time,
finds the best match

Dynamic Time Warping or sequence alignment



Aligns

Lines are not allowed to cross,
keeping shape intact

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100							
v2		225						
v3		.	400					
v4				0				
v5					625			
v6						100		
v7							100	
v8								100

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Find a minimum cost path from left top to bottom right

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Without going up or left (crossing)

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Without

For instance...

Time-warping distance

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100	400	225	25	100	0
v2	25	225	225	225	100	0	25	25
v3	0	100	400	100	25	25	0	100
v4	100	0	900	0	25	225	100	400
v5	400	900	0	900	625	225	400	100
v6	25	25	625	25	0	100	25	225
v7	100	400	100	400	225	25	100	0
v8	0	100	400	100	25	25	0	100

Distance = $\sqrt{100+25+0+0+0+25+0+25+100} = 16.58$ 

Time-warping distance

Before alignment

	v1	v2	v3	v4	v5	v6	v7	v8
s1	10	15	20	30	0	25	10	20
s2	20	30	0	30	25	15	20	10

After alignment

v/v	1/1	2/1	3/1	4/2	5/3	6/4	6/5	7/6	8/7	8/8
s1	10	15	20	30	0	25	25	10	20	10
s2	20	20	20	30	0	30	25	15	20	10

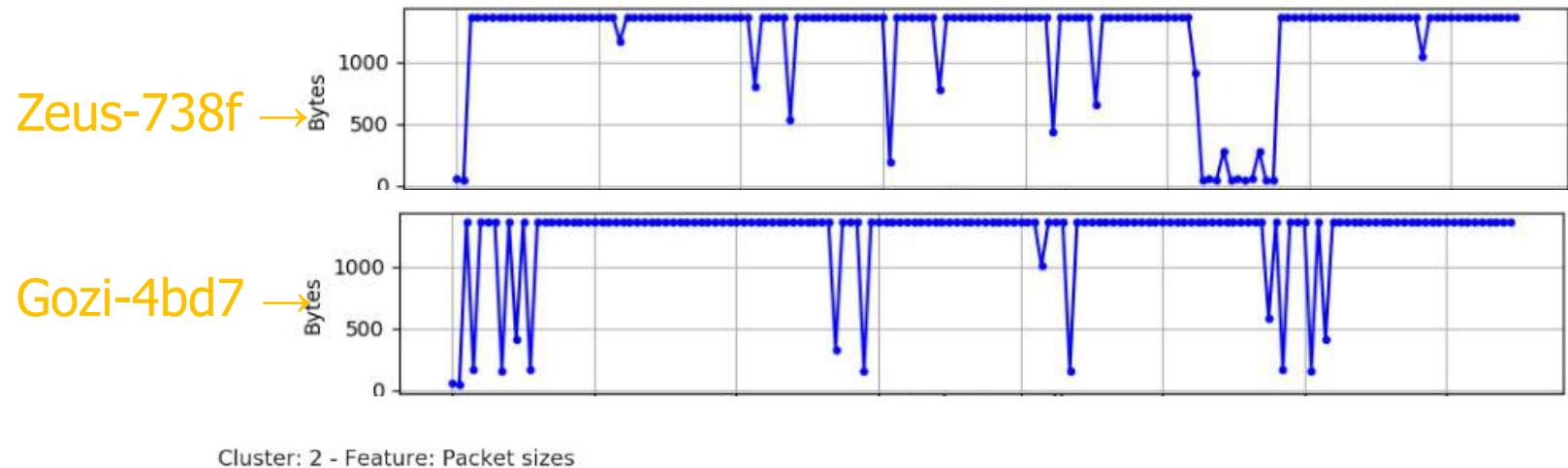
Although we consider more points,
their synchronisation gives a smaller distance!

Applications in real world

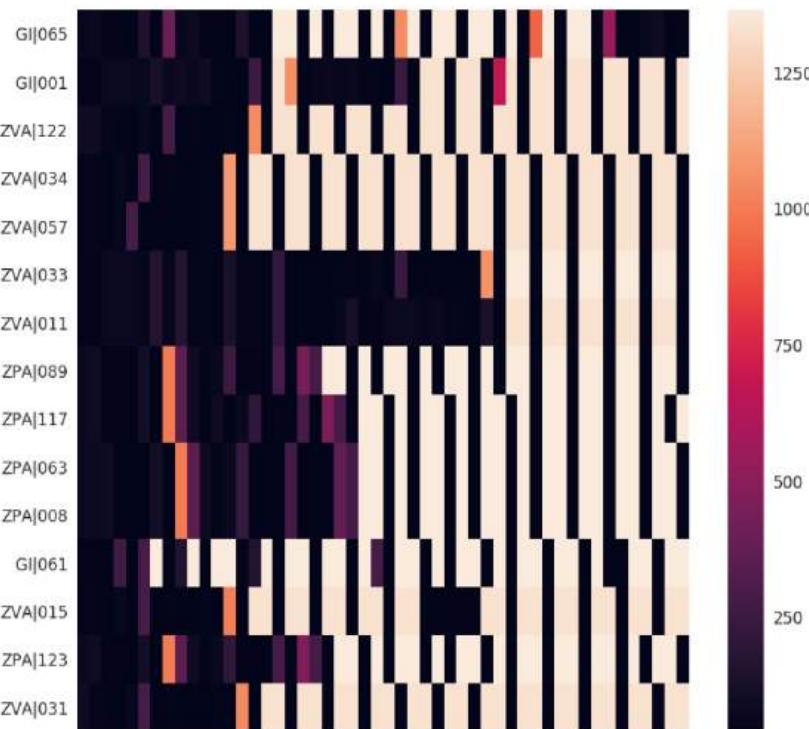
- Extensive use in
 - Speech recognition
 - Gesture recognition
 - Handwriting recognition
 - ...
- But now, also in
 - Clustering malware behavior

No.	Source	Destination	Protocol	Length	Info
40	192.168.1.2	192.168.1.110	ICMP	82	Redirect (Redirect for host)
41	CzNicZSP_00:0...	PcsCompu_7c:9...	ARP	60	192.168.1.1 is at d8:58:d7:00:0f:72
42	192.168.1.110	203.153.165.21	TCP	182	49191 → 8343 [PSH, ACK] Seq=1 Ack=1 Win=65700 Len=128
43	203.153.165.21	192.168.1.110	TCP	60	8343 → 49191 [ACK] Seq=1 Ack=129 Win=15744 Len=0
44	203.153.165.21	192.168.1.110	TCP	1188	8343 → 49191 [PSH, ACK] Seq=1 Ack=129 Win=15744 Len=1134
45	192.168.1.110	203.153.165.21	TCP	380	49191 → 8343 [PSH, ACK] Seq=129 Ack=1135 Win=64564 Len=326
46	192.168.1.2	192.168.1.110	ICMP	408	Redirect (Redirect for host)
47	203.153.165.21	192.168.1.110	TCP	113	8343 → 49191 [PSH, ACK] Seq=1135 Ack=455 Win=16768 Len=59
48	fd2d:ab8c:225...	fd2d:ab8c:225...	DNS	110	Standard query 0xb554 A www.download.windowsupdate.com

Example: Clustering malware



Cluster: 2 - Feature: Packet sizes



- DTW on network traffic with clustering identifies malware families

Sequence alignment

- Discrete variant of time-warping
- Build matrix using (dynamic programming):

$$H(i, 0) = 0, \quad 0 \leq i \leq m,$$

$$H(0, j) = 0, \quad 0 \leq j \leq n,$$

and

$$H(i, j) = \max \begin{cases} 0 \\ H(i - 1, j - 1) + \text{Sim}(a_i, b_j) \\ \max_{k \geq 1} \{H(i - k, j) + W_k\} \\ \max_{l \geq 1} \{H(i, j - l) + W_l\} \end{cases},$$

$1 \leq i \leq m, 1 \leq j \leq n$

- where W is a gap penalty

Sequence alignment

- Discrete variant of time-warping
- Build matrix using (dynamic programming):

$$H(i, 0) = 0, \quad 0 \leq i \leq m,$$

$$H(0, j) = 0, \quad 0 \leq j \leq n,$$

and

$$H(i, j) = \max \left\{ \begin{array}{l} H(i - 1, j - 1) + \underset{0}{\text{Sim}(a_i, b_j)} \\ \max_{k \geq 1} \{H(i - k, j) + W_k\} \\ \max_{l \geq 1} \{H(i, j - l) + W_l\} \end{array} \right\}, \quad 1 \leq i \leq m, 1 \leq j \leq n$$

Diagonal

Right

Down

- where W is a gap penalty

Sequence alignment

- Discrete variant of time-warping
- Build matrix using:

Very popular in bioinformatics:

Histone H1 (residues 120-180)



Sequence alignment

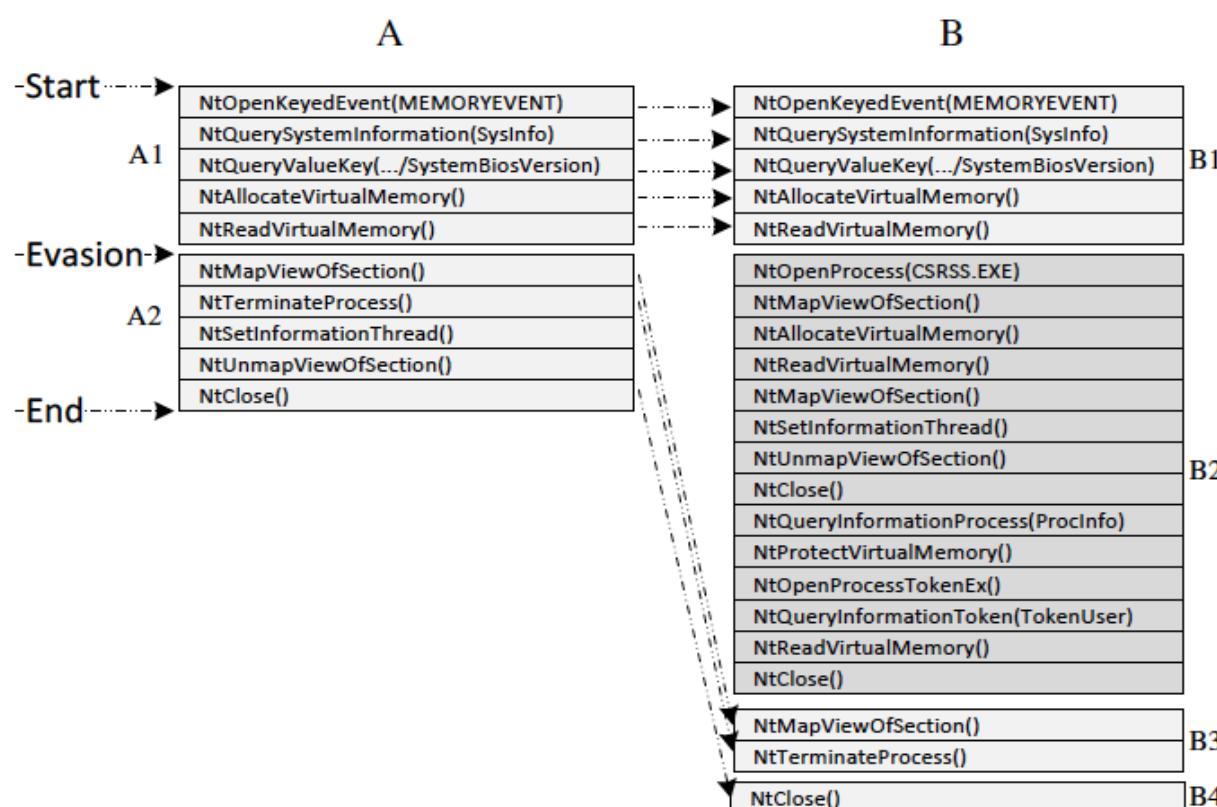
Also used for system call traces:

- Discrete
- Build

Very popular

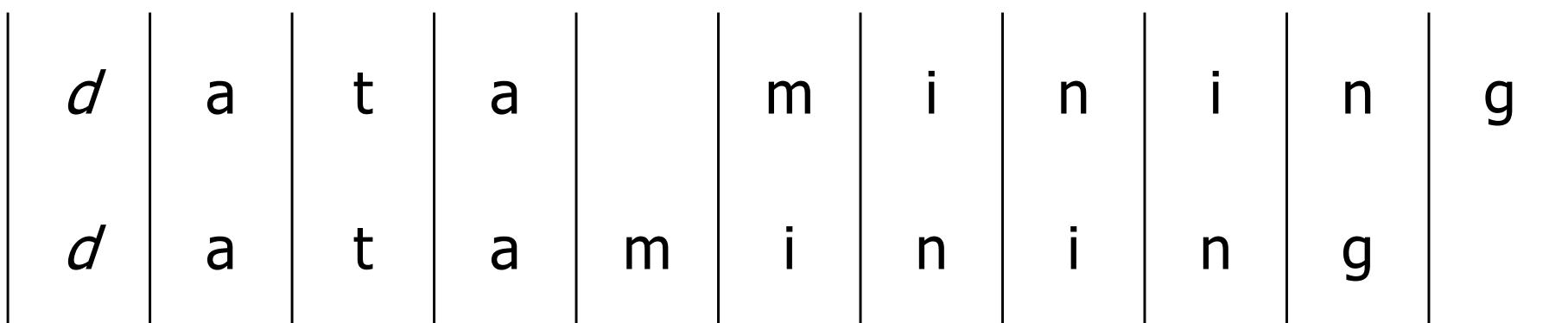
HUMAN KK
MOUSE KK
RAT KK
COW KK
CHIMP KK
**

NON-CONSERVED
AMINO ACIDS



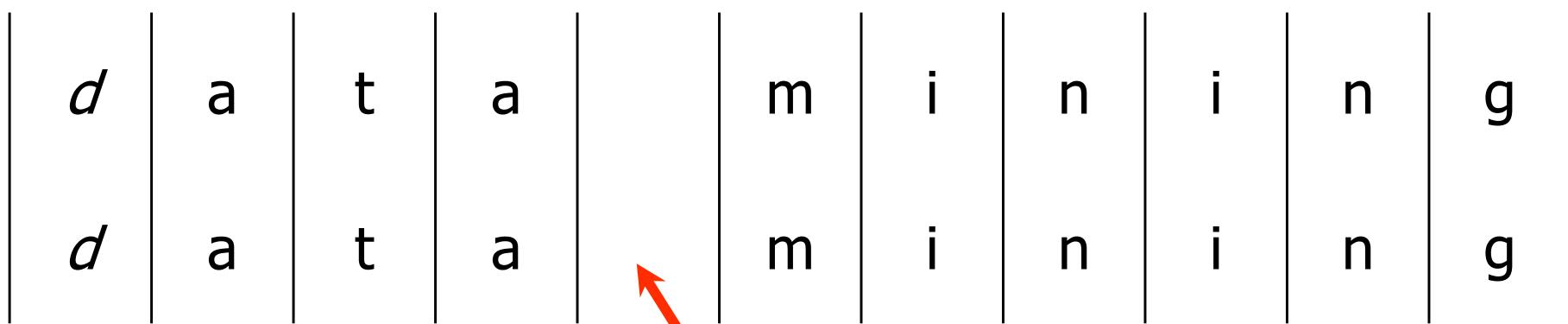
Edit distance

- The edit distance counts the number of operations to go from A to B:
 - Substitution of characters
 - Insertion of characters
 - Deletion of characters



Edit distance

- The edit distance counts the number of operations to go from A to B:
 - Substitution of characters (like Hamming)
 - Insertion of characters
 - Deletion of characters



insert whitespace edit distance = 1

Edit distance: Assignment

- What is the edit distance between the following two strings?

<i>d</i>	a	t	a		m	i	n	i	n	g
<i>d</i>	a	t	a	m	i		n	i	n	g

insert whitespace delete character

edit distance = 2

Is DTW a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:
 - Non-negativity: $d(A, B) \geq 0$
 - Identity: $d(A, B) = 0$ iff $A = B$
 - Symmetry: $d(A, B) = d(B, A)$
 - Triangle inequality: $d(A, C) \leq d(A, B) + d(B, C)$
- Does not satisfy Triangle inequality:
 - $x = [0, 1, 1, 2], y = [0, 1, 2], z = [0, 2, 2]$
 - Then $d(x, y) = 0, d(x, z) = 2$, and $d(y, z) = 1$
 - Counterexample: $d(x, z) > d(x, y) + d(y, z)$

Is edit distance a metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:
 - Non-negativity: $d(A, B) \geq 0$
 - Identity: $d(A, B) = 0$ iff $A = B$
 - Symmetry: $d(A, B) = d(B, A)$
 - Triangle inequality: $d(A, C) \leq d(A, B) + d(B, C)$
- Yes!
- But there exist versions that do not

Is edit distance a metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

These properties are **desirable**

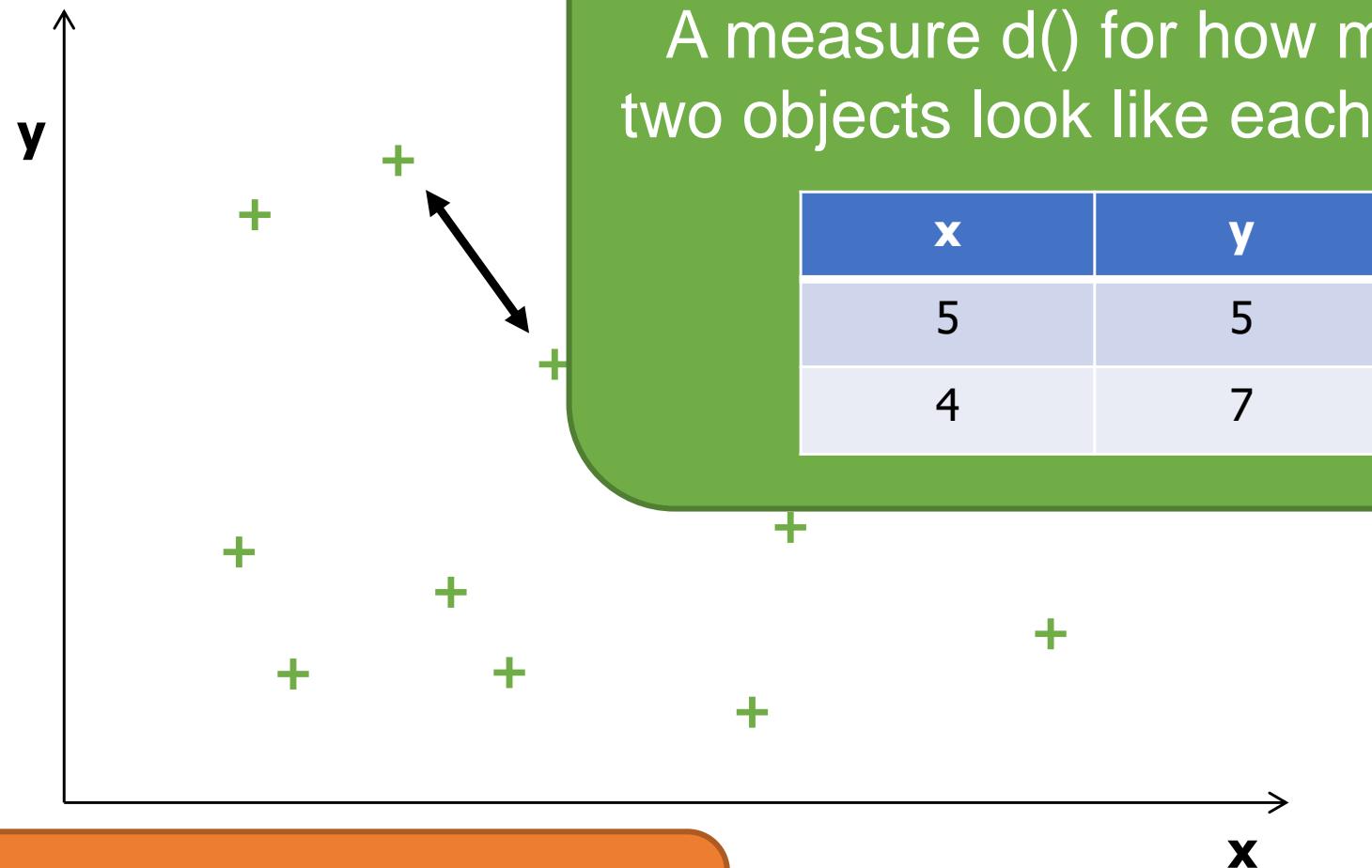
- Generalizations exist: quasi-metrics,
pseudo-metrics, semi-metrics, ...

Many **useful** distances are non-metric

CSE2525 Distances Case studies

How to compute and apply them

What is a distance?



What should their distance be?

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

- Non-negativity:

$$d(A, B) \geq 0$$

- Identity:

$$d(A, B) = 0 \text{ iff } A = B$$

- Symmetry:

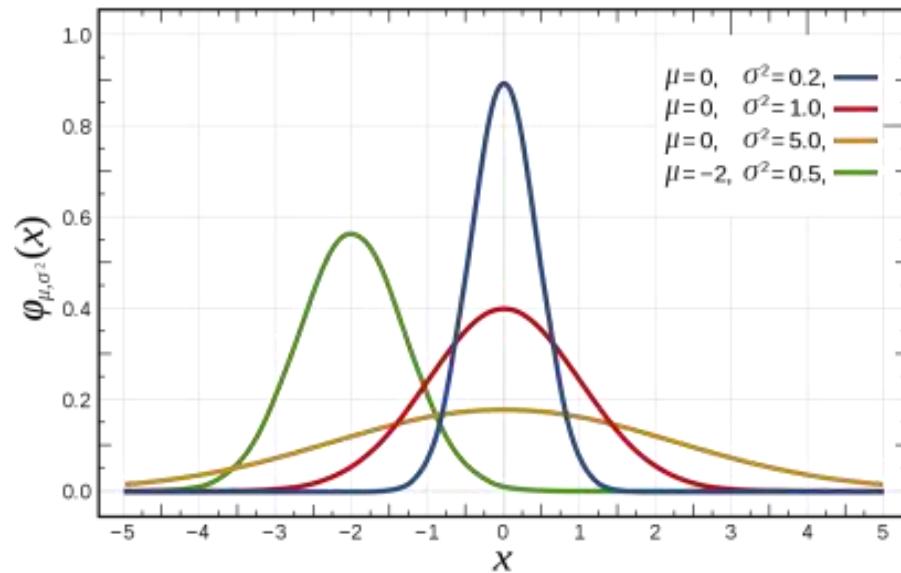
$$d(A, B) = d(B, A)$$

- Triangle inequality:

$$d(A, C) \leq d(A, B) + d(B, C)$$

Case Study – 1

Finding similar items



I want to suggest similar items...

Toys & Games > Dolls & Accessories > Dollhouses

Sponsored

LEGO 10787 Gabby's Dollhouse Kitty Fee's Garden Party, Playhouse and Animals Toys with Gabby and Pandy Poek Figures Plus Treehouse, Swing, Slide and Carousel, Gift for Children from 4 Years

Visit the LEGO Store

4.8 ★★★★★ 80 ratings

€21³⁶
0% claimed
prime One-Day
FREE delivery Tomorrow, 18 November. Order within 4 hrs 12 mins
Deliver to Avishek - Delft 2614

Page 1 of 28

Product Image	Name	Price	Rating
	LEGO 43224 Disney Wish Castle of King Magnifico Wish Film Toy Set with Mini Dolls	€82.85 prime	★★★★★ 2
	LEGO 43223 Disney Wish Asha in Town Rosas Buildable Toy Set of the Wish Movie	€17.99 prime	★★★★★ 97
	LEGO 10786 Gabby's Dollhouse Coddling Ship of Gabby and Meerminkat Spa and...	€14.94 prime	★★★★★ 97
	LEGO 10991 DUPLO City Dream Playground, Toy for Children from 2 Years with Whale an...	€42.98 prime	★★★★★ 19
	LEGO 43189 Disney Frozen 2 Elsa and the Nokk LEGO Book Portable Playset,...	€14.99 prime	★★★★★ 5,106
	LEGO 76992 Sonic the Hedgehog Amy's Animal Island Playset, Buildable Toy with 6 Cha...	€34.99 prime	★★★★★ 14

Products related to this item

Page 1 of 28

Sponsored



LEGO 43224 Disney Wish Castle of King Magnifico Wish Film Toy Set with Mini Dolls
€82.85



LEGO 43223 Disney Wish Asha in Town Rosas Buildable Toy Set of the Wish Movie
★★★★★ 2
€17.99



LEGO 10786 Gabby's Dollhouse Coddling Ship of Gabby and Meerminkat Spa and...
★★★★★ 97
 €14.94
List: €20.99 (29% off)



LEGO 10991 DUPLO City Dream Playground, Toy for Children from 2 Years with Whale an...
★★★★★ 19
€42.98



LEGO 43189 Disney Frozen 2 Elsa and the Nokk LEGO Book Portable Playset,...
★★★★★ 5,106
€14.99



LEGO 76992 Sonic the Hedgehog Amy's Animal Island Playset, Buildable Toy with 6 Cha...
★★★★★ 14
€34.99

Different from

Customers who bought this item also bought

Page 1 of 8



LEGO 10786 Gabby's Dollhouse Coddling Ship of Gabby and Meerminkat Spa and...
★★★★★ 109
 €14.94
RRP: €20.99
 2% Claimed



LEGO 10785 Gabby's Dollhouse Baking with Cakey, Kitchen Set with Gabby and Cakey Cat Figures, Includes...
★★★★★ 201
€8.95



Trefl - Gabby's Dollhouse Gabi Cat House Puzzle with 100 Pieces Colorful Puzzles with Fairy Tale Characters Creative Fu...
★★★★★ 32
€8.49



Gabby's Dollhouse - Gabby's Magic Cat Ears Headband with Light & Sound
★★★★★ 7
 €14.99



Gabby's Dollhouse - Jumbling Tower Game
★★★★★ 36
 €7.08
RRP: €10.00
 13% Claimed



Clementoni - Puzzle 3X48 Pieces Gabby's Dollhouse, Children's Puzzles, 5-7 Years, 25290
★★★★★ 6
€6.99



Gabby's Dollhouse, Carlita's Vehicle with Pandi Poek
€13.99



What are the features ?

Toys & Games > Dolls & Accessories > Dollhouses

Sponsored



Roll over image to zoom in



LEGO 10787 Gabby's Dollhouse Kitty Fee's Garden Party, Playhouse and Animals Toys with Gabby and Pandy Poek Figures Plus Treehouse, Swing, Slide and Carousel, Gift for Children from 4 Years

Visit the LEGO Store

4.8 ★★★★★ 80 ratings

Amazon's Choice for "gabby poppenhuis"

100+ bought in past month

Black Friday Deal

-29% €21³⁶

RRP: €29.99

✓prime One-Day

All prices include VAT.

Brand LEGO

Manufacturer 48

Minimum Age
(MONTHS)

Material Plastic

Colour Multicolour

Educational Role Play

€21³⁶

0% claimed

✓prime One-Day

FREE delivery Tomorrow, 18 November. Order within 4 hrs
12 mins

Deliver to Avishek - Delft 2614

In stock

Quantity: 1

Add to Basket

Buy Now

Dispatches from Amazon

Sold by Amazon

Returns Returnable until

Jan 31, 2024

Payment Secure transaction

Add gift options

Add to List

Amazon

Toys & Games > Dolls & Accessories > Dollhouses

Sponsored



Roll over image to zoom in

4+

VISUALS

VIDEO

LEGO 10787 Gabby's Dollhouse Kitty Fee's Garden Party, Playhouse and Animals Toys with Gabby and Pandy Poek Figures Plus Treehouse, Swing, Slide and Carousel, Gift for Children from 4 Years

Visit the LEGO Store

4.8 ★★★★★ 80 ratings

Amazon's Choice for "gabby poppenhuis"

100+ bought in past month

Black Friday Deal

-29% €21³⁶

RRP: €29.99

prime One-Day

All prices include VAT.

Brand	LEGO
Manufacturer	48
Minimum Age (MONTHS)	
Material	Plastic
Colour	Multicolour
Educational	Role Play

€21³⁶

0% claimed

prime One-Day

FREE delivery Tomorrow, 18 November. Order within 4 hrs 12 mins.

Deliver to Avishek - Delft 2614

In stock

Quantity: 1

Add to Basket

Buy Now

Dispatches from Amazon

Sold by Amazon

Returns Returnable until Jan 31, 2024

Payment Secure transaction

Add gift options

Add to List

General Recipe -- Use Distances

On which feature should we compute the distance ?

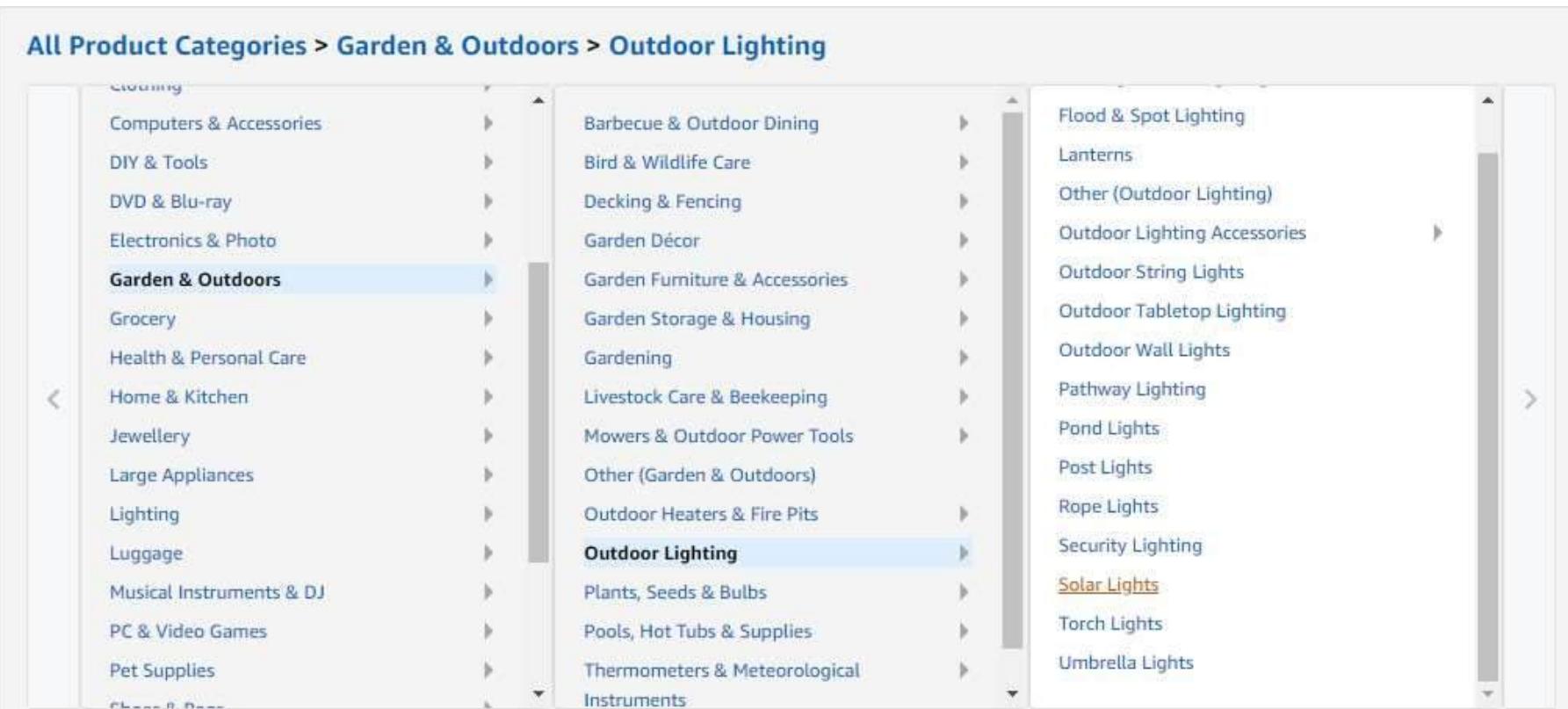
General Recipe -- Use Distances

		Products					
		i	j				
				0.34			

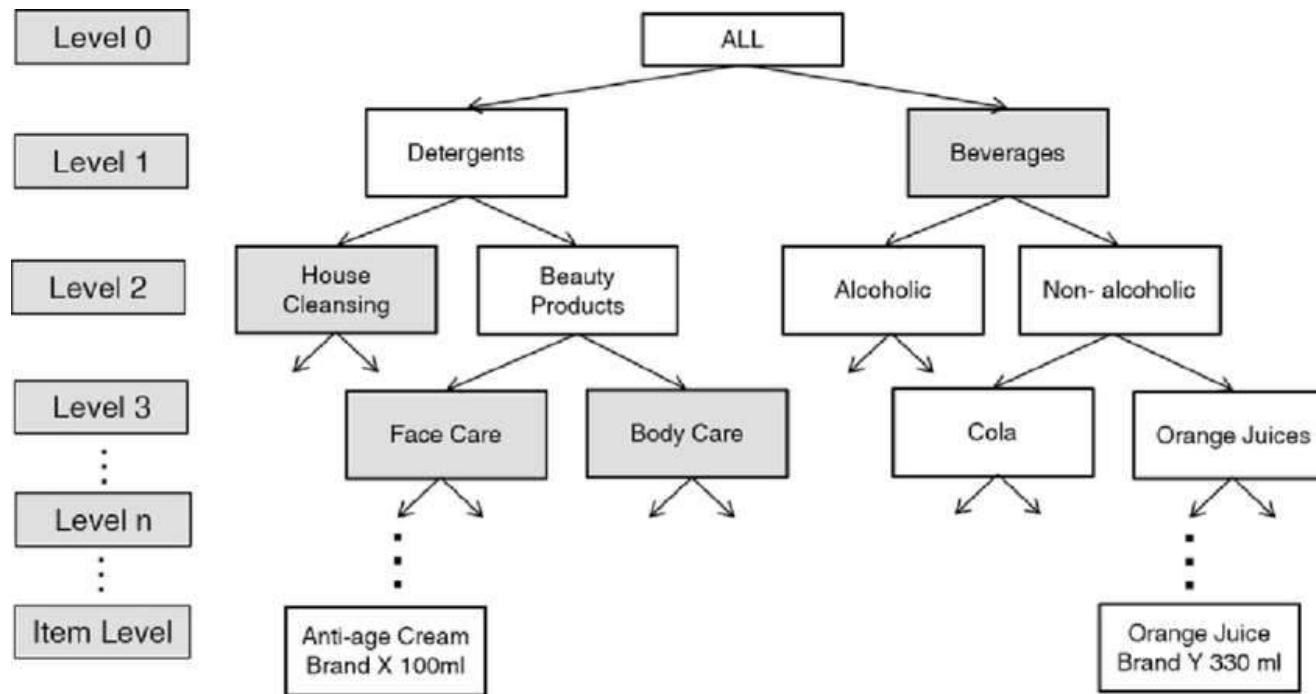
On which feature should we compute the distance ?

How do we choose the best feature ?

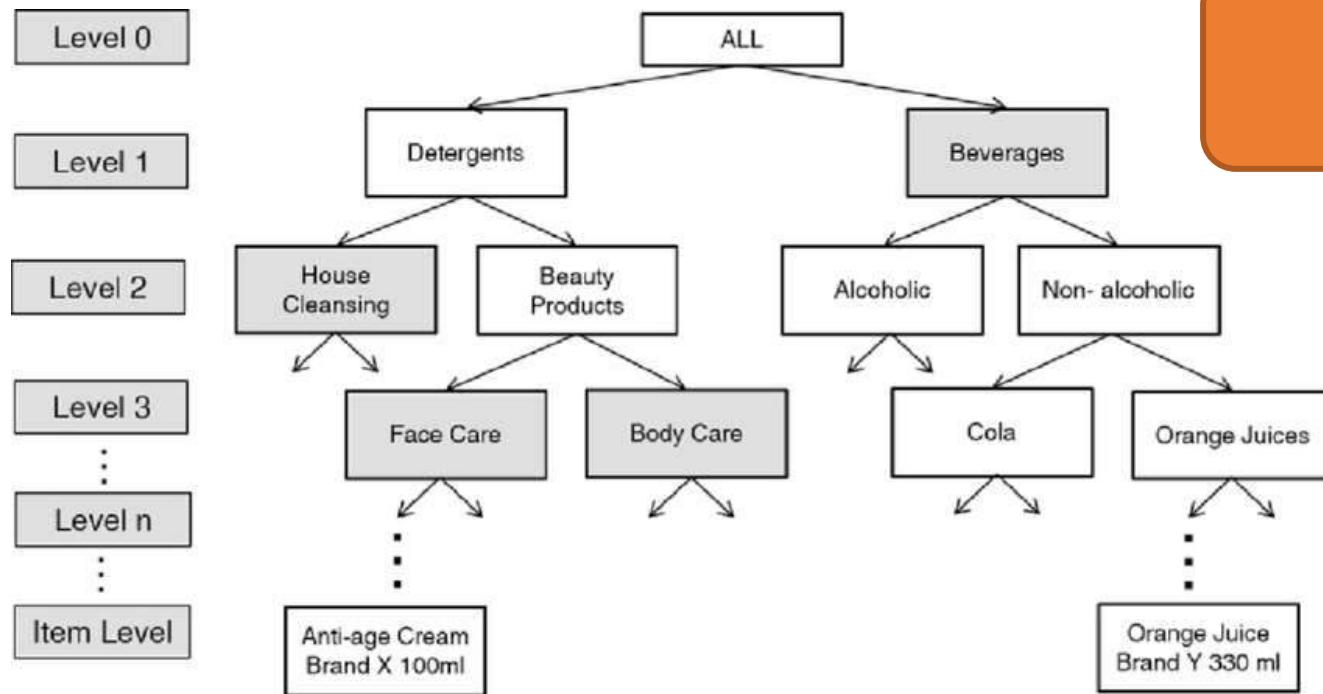
Product Taxonomy



What is the distance ?



What is the distance ?



Shortest path

Is it a distance metric ?

What is a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:

- Non-negativity:

$$d(A, B) \geq 0$$

- Identity:

$$d(A, B) = 0 \text{ iff } A = B$$

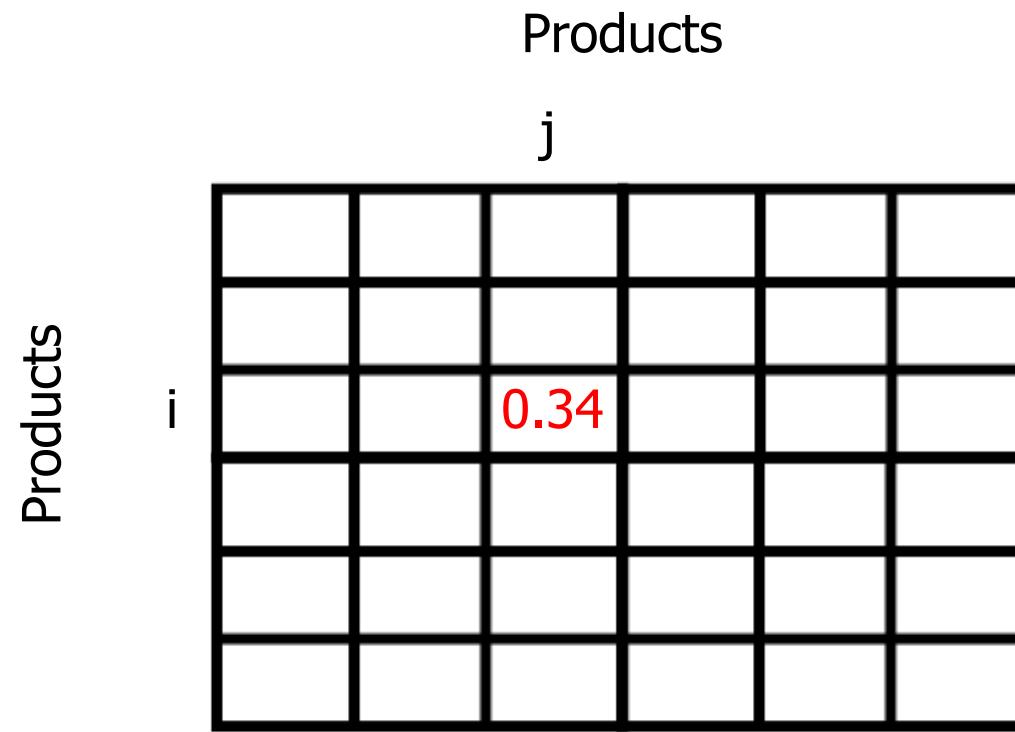
- Symmetry:

$$d(A, B) = d(B, A)$$

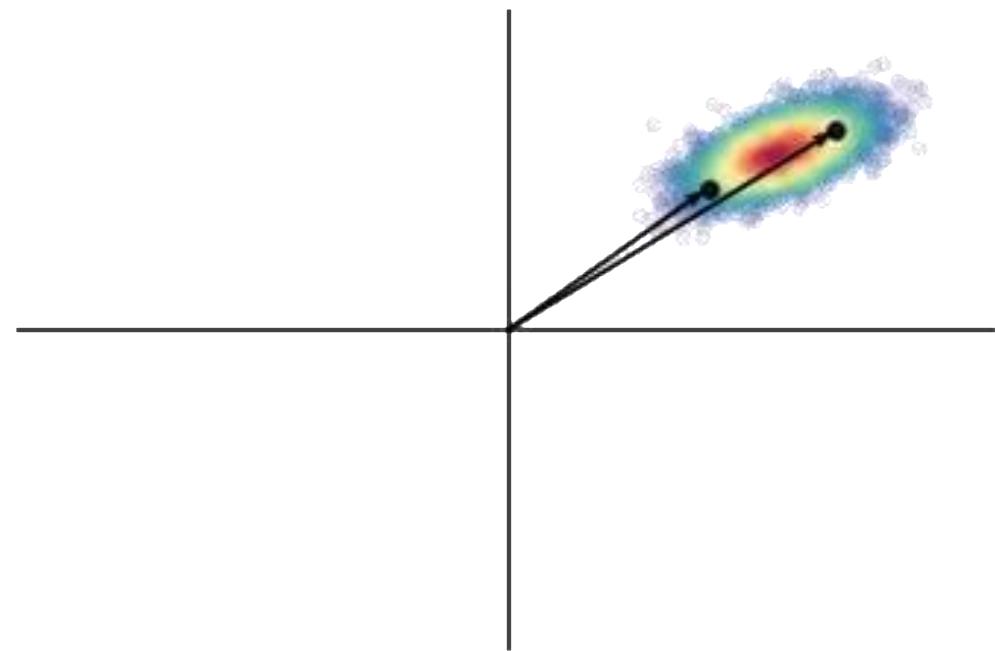
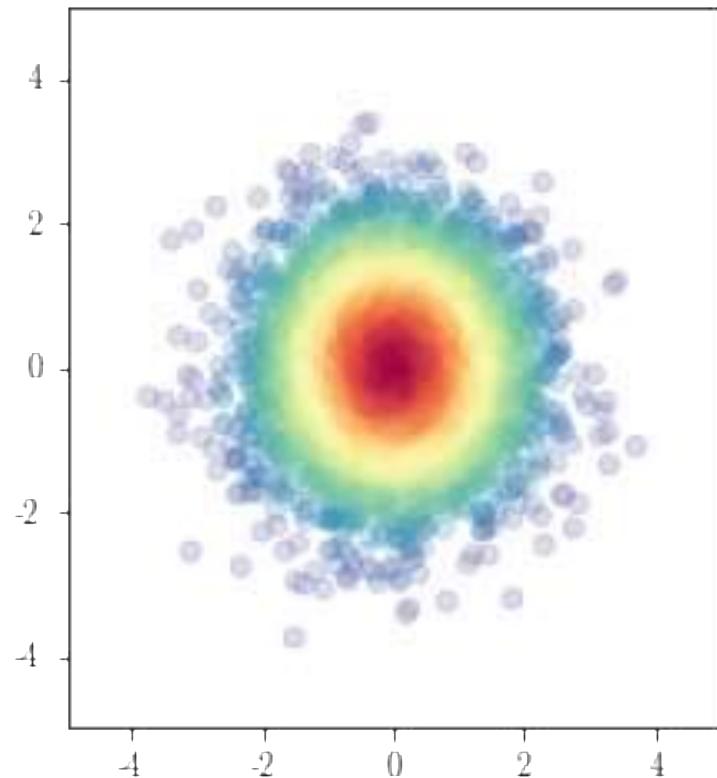
- Triangle inequality:

$$d(A, C) \leq d(A, B) + d(B, C)$$

Text based distance



Text based distance – Cosine sim.

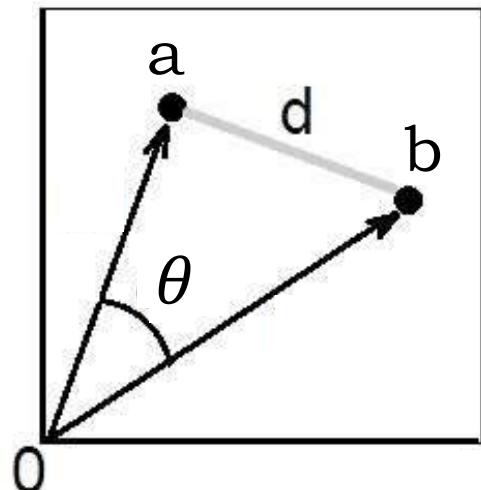


Distances should be discriminative

Cosine distance

- The cosine distance measures the angle between two vectors:

$$d(\mathbf{a}, \mathbf{b}) = 1 - \cos(\theta) = 1 - \frac{\mathbf{a}^\top \mathbf{b}}{\|\mathbf{a}\| \|\mathbf{b}\|}$$
$$\frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$



only looks at direction of vectors,
not at the length of these vectors!

Other features

- What is the distance between two items based on
 - Price
 - Ratings

Price: Euclidean distance

- The Euclidean distance between two vectors \mathbf{a} and \mathbf{b} is defined as:

$$d(\mathbf{a}, \mathbf{b}) = \sqrt{\sum_{d=1}^D (a_d - b_d)^2} = \|\mathbf{a} - \mathbf{b}\|$$

When is price not good feature ?

MEGA Barbie Color Reveal Dream House Building Set with Over 25 Surprises, 5 Micro Dolls and 6 Animals, Gift Set for Children from 5 Years, HHM01

Brand: MEGA

4.6  477 ratings

Use normalized distances
with mean , and std. dev

Hughdy Ceramic Neti Pot, 250ml Ceramic Neti Pot Nose Wash Comfortable Spout Pot for Nose Cleaning Green

Brand: Hughdy

= 20000 Euros

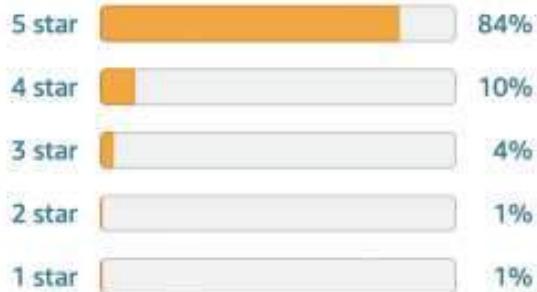
The outlier problem

Reviews

Customer reviews

★★★★★ 4.8 out of 5

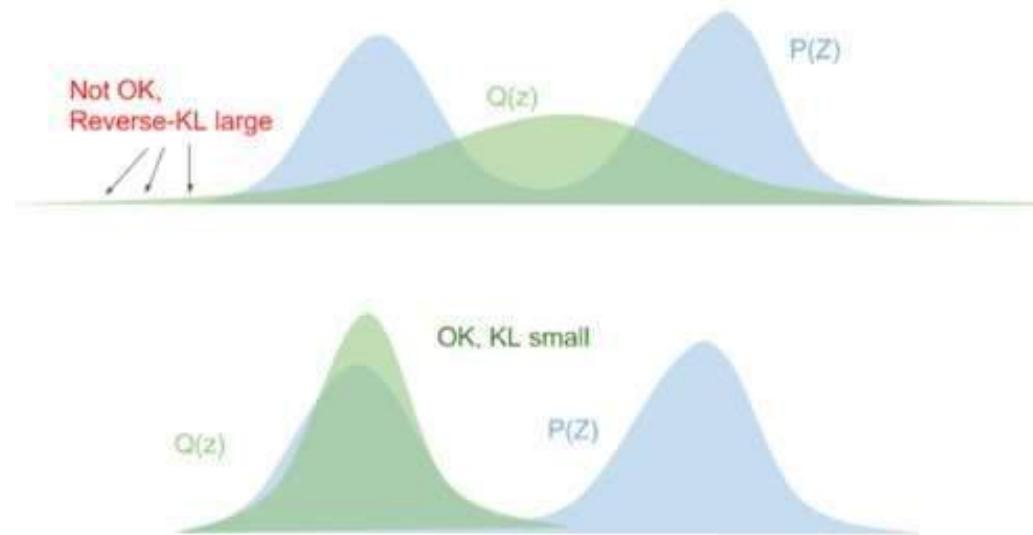
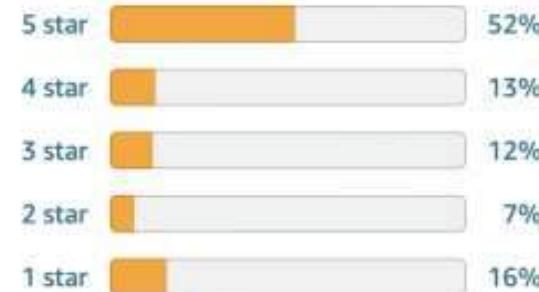
187 global ratings



Customer reviews

★★★★☆ 3.8 out of 5

47 global ratings



When are reviews not good feature ?

MEGA Barbie Color Reveal Dream House Building Set with Over 25 Surprises, 5 Micro Dolls and 6 Animals, Gift Set for Children from 5 Years, HHM01

Brand: MEGA

4.6  477 ratings

Hughdy Ceramic Neti Pot, 250ml Ceramic Neti Pot Nose Wash Comfortable Spout Pot for Nose Cleaning Green

Brand: Hughdy

€20⁵⁰

The cold start problem

How to choose a feature ?

Products related to this item

Sponsored



Reviews



Categorical



How to choose a feature ?

Customers who bought this item also bought

Page 1 of 8

LEGO 10786 Gabby's Dollhouse Coddling Ship of Gabby and Meeminkat Spa and...
★★★★★ 109
29% off Black Friday Deal
€14.94
RRP: €20.99
prime FREE delivery
2% claimed

LEGO 10785 Gabby's Dollhouse Baking with Caky, Kitchen Set with Gabby and Caky Cat Figures, Includes...
★★★★★ 201
€8.95
prime FREE delivery

Treff - Gabby's Dollhouse Gabi Cat House Puzzle with 100 Pieces Colorful Puzzles with Fairy Tale Characters Creative Fu...
★★★★★ 32
€8.49
prime FREE delivery

Gabby's Dollhouse - Gabby's Magic Cat Ears Headband with Light & Sound
★★★★★ 7
#1 Best Seller in Musical Toy Instruments
€14.99
prime FREE delivery

Gabby's Dollhouse - Jumbling Tower Game
★★★★★ 36
29% off Black Friday Deal
€7.08
RRP: €10.00
prime FREE delivery
13% claimed

Clementoni - Puzzle 3X48 Pieces Gabby's Dollhouse, Children's Puzzles, 5-7 Years, 25290
★★★★★ 6
€6.99
prime FREE delivery

Gabby's Dollhouse, Carlita's Vehicle with Pandy Poek
€13.99
prime FREE delivery

Playmobil 9466 City Action Fire Terrain Vehicle, 34.8 x 25 x 12.5 cm, Multicolor
★★★★★ 5,843
Black Friday Deal
€32.10 prime
List: €53.99 (41% off)

LEGO 43212 Disney: Disney Party Train Set with Vaiama, Woody, Peter Pan and Tinker...
★★★★★ 363
€27.99 prime

This LEGO Disney and Pixar buildable toy includes the iconic 'Up' house with balloon...
★★★★★ 1,342
€41.99 prime

LEGO 41738 Friends Dog Rescue Bike, Animal Grooming Playset for Children from 6 Yea...
★★★★★ 802
€8.95 prime

Lexibook, Barbie Luminous Microphone for Kids, Musical Toy, Built-in Speaker, Lumin...
★★★★★ 5
€19.99 prime

Lexibook Barbie K704BB Children's Electronic Piano with Light Effects, Microphone D...
★★★★★ 1
€34.99 prime

PAW Patrol - Watchtower playset with vehicle thrower 2 Chase action figures Chase p...
★★★★★ 332
Black Friday Deal
€36.99 prime
List: €59.99 (38% off)

Lessons about the design space

1. Heterogeneous Data Types: Product attributes include

1. numerical values (e.g., price, rating),
2. categorical data (e.g., brand, category),
3. and text (e.g., reviews).

Lessons about the design space

1. Heterogeneous Data Types: Product attributes include

1. numerical values (e.g., price, rating),
2. categorical data (e.g., brand, category),
3. and text (e.g., reviews).

2. Make sure to choose the **correct way to define distances**

1. Is a metric or not ?
2. Normalizing, missing values, ..
3. Equally spread around and not equally distant

Lessons about the design space

1. Heterogeneous Data Types: Product attributes include

1. numerical values (e.g., price, rating),
2. categorical data (e.g., brand, category),
3. and text (e.g., reviews).

2. Make sure to choose the **correct way to define distances**

1. Is a metric or not ?
2. Normalizing, missing values, ..
3. Equally spread around and not equally distant

3. Be mindful of the **actual objective**

1. Are you optimizing for clicks
2. Are you optimizing for buys

Pipeline

- **Objective:** to show k-related items
- What feature/s should be used to compute similarity ?
 - What are some features that make sense?
 - Non-sparse, reliable, predictive, and sensitive to outliers, ..
- What is the best distance function to use ?
 - **E.g.;** Text: Jaccard, edit-distance, cosine-similarity, ..
 - Do we use distance transformations, scaling, normalization ?
 - What does the distribution look like ?
 - What is computationally feasible ?
- What is the objective function ?
 - Algorithmic hyperparameters: we use **k-nearest** neighbors

Case Study – 2

Near Duplicate Detection

Youtube

An average of **2,500 new videos** are uploaded to YouTube **every minute**, amounting to **183 hours of video content**, with an average video length of **4.4 minutes**.



How do we find near-duplicates given a query video (upload) ?

Youtube

In 2007, Viacom sued youtube for over 1 billion dollars

Followed by class action law suits by many sports channels



How do we find near-duplicates given a query video (upload) ?

Youtube

Why do we care about ***near duplicates*** ? What about exact exact duplicates ?

- Exact duplicates are easier to deal
 - Use hash functions on the entire bit stream
 - For distinctness one can even use cryptographic hash
 - **The avalanche effect:** *a slight change in input will cause a significant change in its hash*
- **Near duplicates:** When two inputs are similar, we hope the hashing function should generate similar outputs as well

Lets try some fancy hashing...



Perceptual video hashing

1001000100011100

1101000110011100

What distance function ?

$$\mathbf{h}_v = H(V, K)$$

- Some properties:
 - Highly unlikely to recover the video given the hash
 - Hash value should be much concise
 - All videos give you say a **binary hash** of length 16 or 32

1001000100011100

1101000110011100

Hamming distance: Example

- Hamming distance between bit strings A and B

A	1	1	0	0	1	0	0	1
B	1	0	0	1	1	1	0	1

Hamming distance = 3

Norm. Hamming distance = 3/8

What distance function ?

$$\mathbf{h}_v = H(V, K)$$

- Some properties:
 - Highly unlikely to recover the video given the hash
 - Hash value should be much concise
 - All videos give you say a **binary hash** of length 16 or 32
- Normalized hamming distance = hamming distance/length
 - **Robustness:**
 - $d(v, v)$ is almost 0
 - **Fragility, unpredictability**
 - For same key $d(v, v') = 0.5$, if v and v' are different
 - For diff. keys $d(v_{k1}, v_{k2}) = 0.5$, for even same v

How do find near dups now ?

- Given a video you are uploading q, find all the near duplicates
- Can use p-hash
 - But what are the limitations given it's a blackbox
 - Also given the properties

What should you be bothered about ?

- Given a video you are uploading q , find all the near duplicates
- Can use p-hash
 - But what are the limitations given it's a blackbox
 - Also given the properties
- Short vs long videos ?
- Do we care about missing returning duplicates ?
- Do we care more about good queries identified as duplicates ?

How do design a pipeline ?

- Use pHash as a **fast filtering** stage
 - Calibrate the distance function – what is the threshold of similarity ?
 - Domain knowledge
 - Visualization, ...
 - Candidate output videos = $c_1, c_2, c_3, \dots, c_k$
- **Slow double-check** phase
 - Annotate data
 - Turn it into a ML task
 - Given **(q,cand)** learn a ML model
 - to predict {**near dup**, **NOT near dup**}
- Note that you only do the second phase on the output of phase one

How do design a pipeline ?

- Use pHash as a **fast filtering** stage
 - Calibrate the distance function – what is the threshold of similarity ?
 - Domain knowledge
 - Visualization
 - Candidate output videos = $c_1, c_2, c_3, \dots, c_k$
- **Slow double-check** phase
 - Annotate data
 - Turn it into a ML task
 - Given $(q, cand)$ learn a ML model
 - to predict if c_i is a near-duplicate of q
- Note that you only do the second phase on the output of phase one

General recipe in many DM tasks

- **Web Search:** Use a **fast filtering** stage like lexical matching
 - Given a query **q**, find all documents that contain the query terms
 - Facilitated by an index (lecture on indexing text)
- **Slow double-check phase**
 - Annotate data
 - Turn it into a ML task
 - Given **(q,cand)** learn a ML model
 - to predict {**relevant** , **NOT relevant**}
- Note that you only do the second phase on the output of phase one

General recipe in many DM tasks

- **Question Answering:** Use a **fast filtering** stage like lexical matching
 - Given a question q , find all documents that contain the query terms
 - Facilitated by an index (lecture on indexing text)
- **Slow double-check phase**
 - Read each result document for answer containment
 - Given $(q, cand)$ learn a ML model to find answer span
- Note that you only do the second phase on the output of phase one

Thanks

The diagram illustrates a matrix $A_{m \times n}$. It is represented by a large bracketed rectangular frame. On the left side, a vertical brace indicates the number of rows, labeled from 1 to m . Above the matrix, a horizontal brace indicates the number of columns, labeled from 1 to n . The matrix itself contains elements a_{ij} , where i represents the row index and j represents the column index. The elements are arranged in rows and columns, with ellipses indicating continuation.

$$\left[\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{array} \right] = A_{m \times n}$$

Matrices

Data Mining CSE2525

Nergis Tomen

21.11.2024

Introduction



Nergis (n.tomen [at] tudelft.nl)



Intelligent Systems Department

Co-directing...



Biomorphic Intelligence Lab -
Biologically-inspired aerial robotics



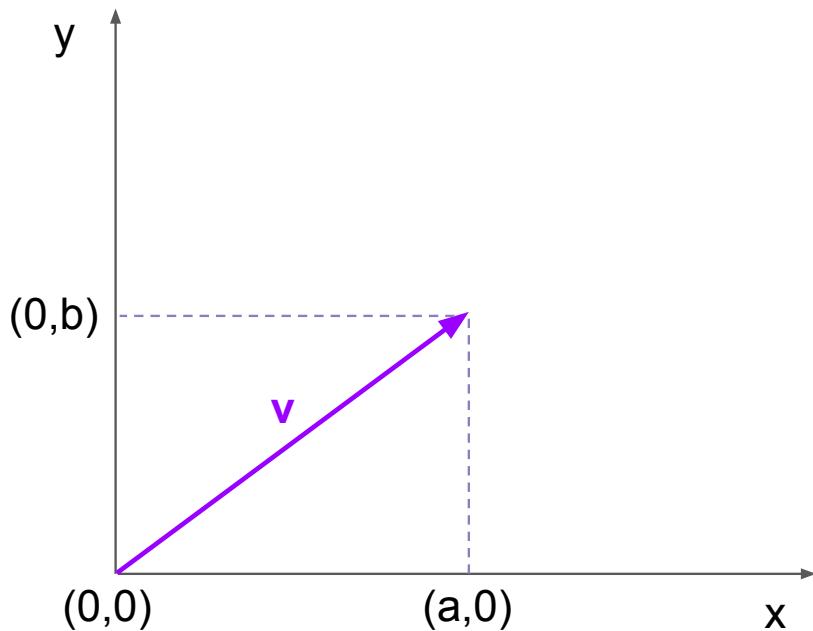
BIOLab -
Biomedical Intervention Optimization

Recall: Linear algebra

Question:

What is a vector?

What is a vector?



- Geometric object with:
 - magnitude
 - direction
 - In Cartesian coordinates:
$$\mathbf{v} = [a,b]$$
typically represents a vector \mathbf{v} from the origin to the point (a,b).
 - In this example, the vector \mathbf{v} exists in 2-dimensional Euclidean space:
$$\mathbf{v} \in \mathbb{R}^2$$
- Or we can define a vector \mathbf{w} in N-dimensions:
- $$\mathbf{w} \in \mathbb{R}^N$$

What is a vector?

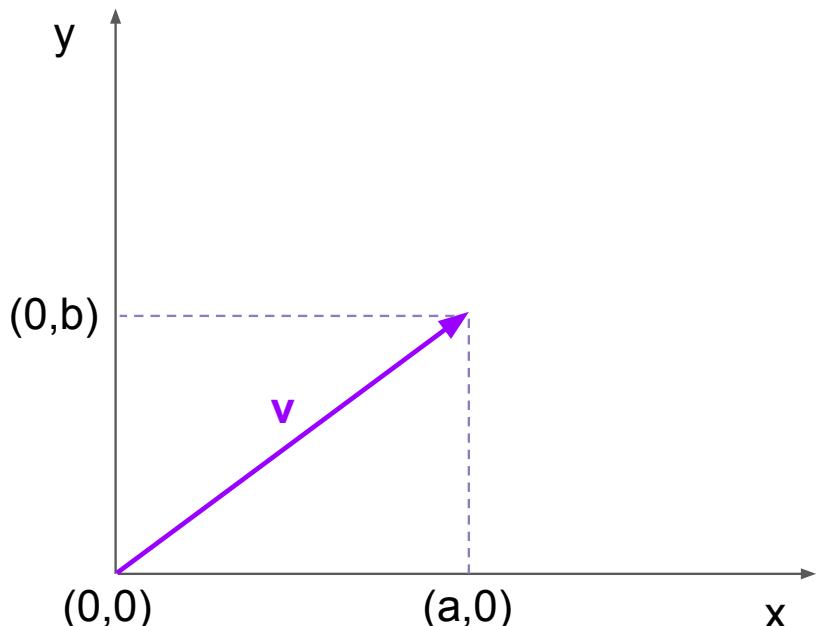
Some properties:

- Vector addition: $\vec{V} + \vec{W}$

$$\begin{bmatrix} 3 \\ -5 \end{bmatrix} + \begin{bmatrix} 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 3+2 \\ -5+1 \end{bmatrix}$$

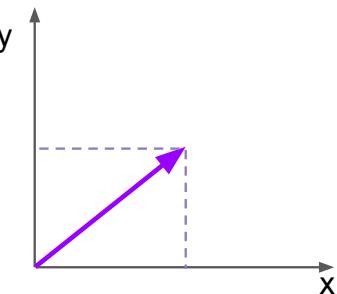
- Scalar multiplication: $2 \vec{V}$

$$2 \begin{bmatrix} 3 \\ -5 \end{bmatrix} = \begin{bmatrix} 2(3) \\ 2(-5) \end{bmatrix}$$



We can 'interpret' a vector as:

- A geometric object:



- A list of numbers: $\mathbf{w} = [x_1, x_2, x_3, \dots, x_N]$, with $\mathbf{w} \in \mathbb{R}^N$
- A formally defined mathematical object with some properties, e.g. vector addition and scalar multiplication



Image source: <https://www.amazon.com/Matrix-Keanu-Reeves/dp/B00000K19E>

Question:

What is a matrix?



Question:

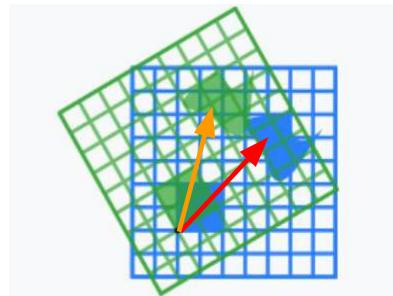
What is a matrix?



What is a matrix?

Just like a vector, there are multiple interpretations!

- A linear map:



- A table of numbers (organized in m rows and n columns): $\mathbf{A}_{(m \times n)}$
- A formally defined mathematical object with some properties, e.g. addition, multiplication and transposition

What is a matrix?

Different fields provide different useful perspectives, e.g.

In **natural science** or **physics**: A vector can represent a force, velocity, etc.

A matrix defines a **linear map** between vector spaces.

In **computer science**:

A matrix can be a "**data table**":

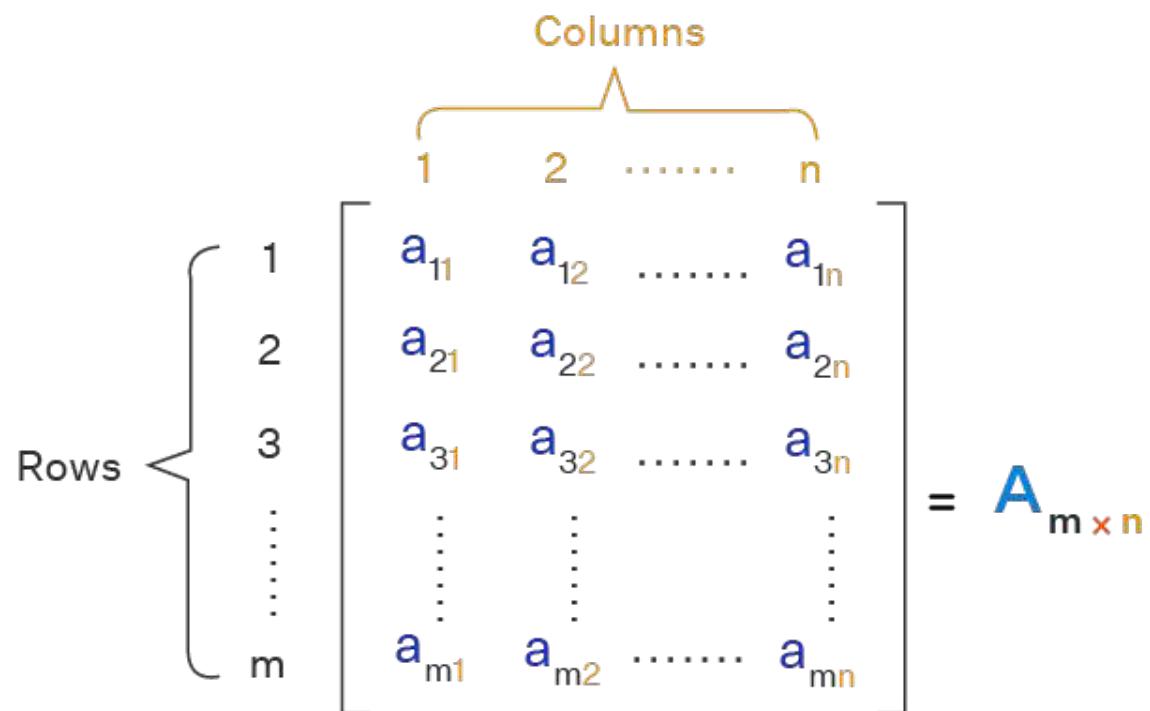
m samples (e.g. houses) with n features (e.g. number of rooms, size and price of a house, etc.).

Mathematics:

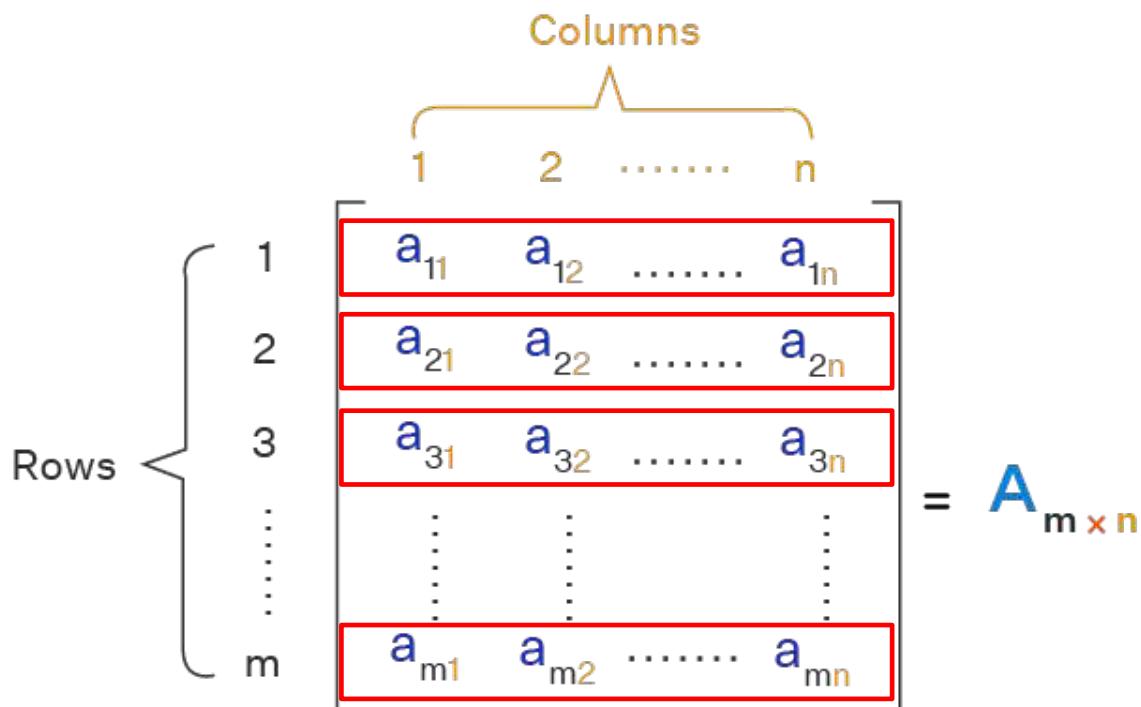
A matrix has formally defined **properties** (e.g. linearity):
Important to keep in mind for applications

What is a matrix?

- Rectangular array of numbers



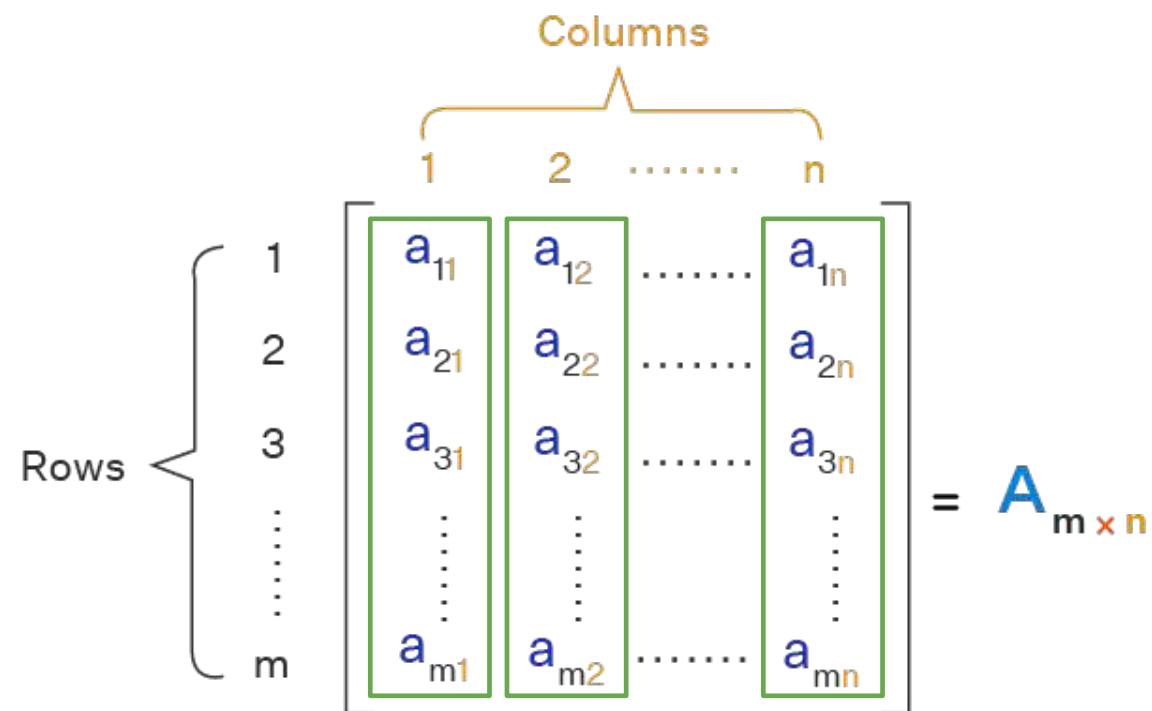
What is a matrix?



- Rectangular array of numbers

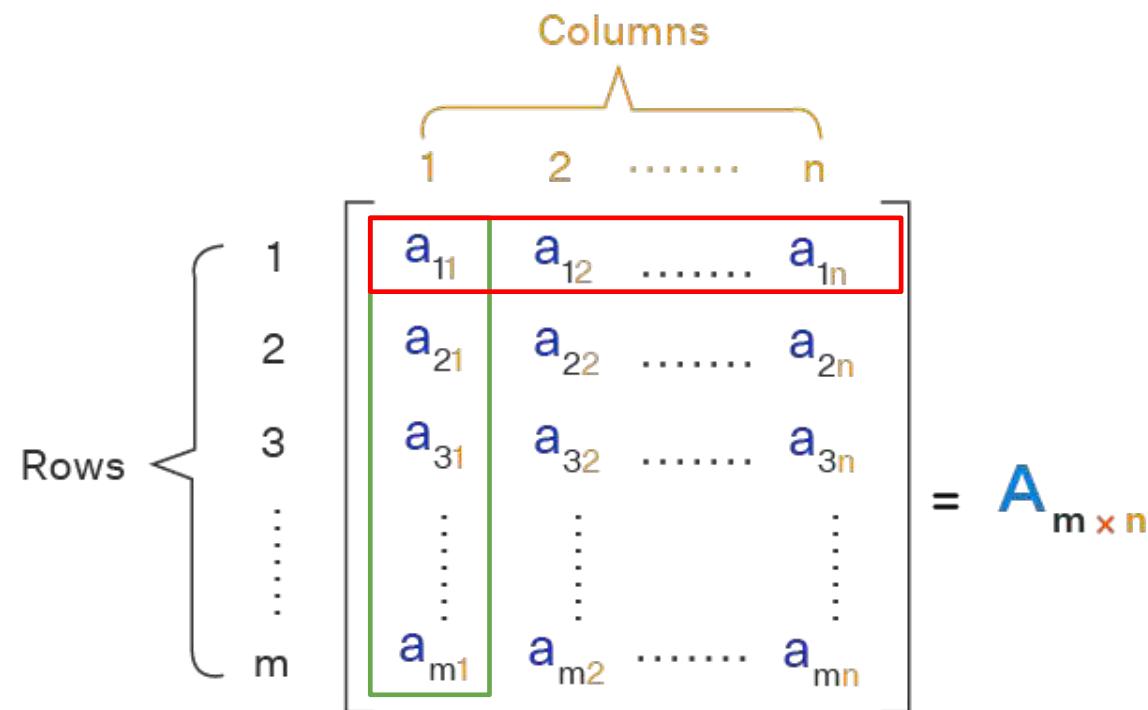
- We can build a matrix by e.g.:
 - Stacking rows of vectors

What is a matrix?



- Rectangular array of numbers
- We can build a matrix by e.g.:
 - Stacking rows of vectors
 - Stacking columns of vectors

What is a matrix?



- Rectangular array of numbers
- We can build a matrix by e.g.:
 - Stacking rows of vectors
 - Stacking columns of vectors
- A row vector is a $1 \times n$ matrix, a column vector is a $m \times 1$ matrix.

Basics of linear algebra

Plan:

- Matrices as **linear transformations** (linear maps)
- Matrices as **data tables** (the "data matrix")
- Eigenvalues, eigenvectors and **Principal Component Analysis** (PCA)

Important applications of matrices in this course:

- Lab assignments:
 - detecting anomalies in the SCADA system
 - building a recommender system for profile matching
 - clustering data in graphs

Questions?

Matrix as a linear transformation

Question:

What is a linear
transformation?

Matrix as a linear transformation

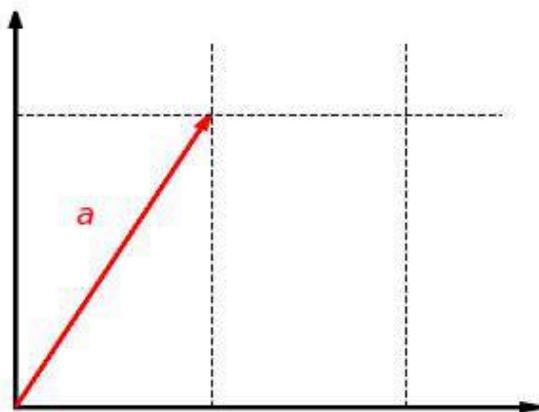
Definition:

Given two vectors \mathbf{a} and \mathbf{b} , and a scalar λ , a function f is a linear map or linear transformation if:

1) **Additivity** $\rightarrow f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$

2) **Scalar multiplication** $\rightarrow f(\lambda \mathbf{a}) = \lambda f(\mathbf{a})$

Matrix as a linear transformation

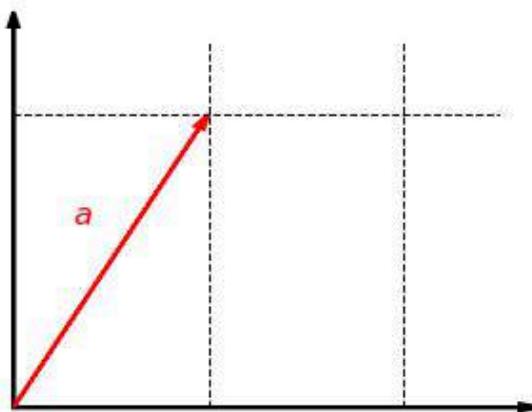


$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 2x \\ y \end{bmatrix}$$

M a $f(a)$

Image source: https://en.wikipedia.org/wiki/Linear_map

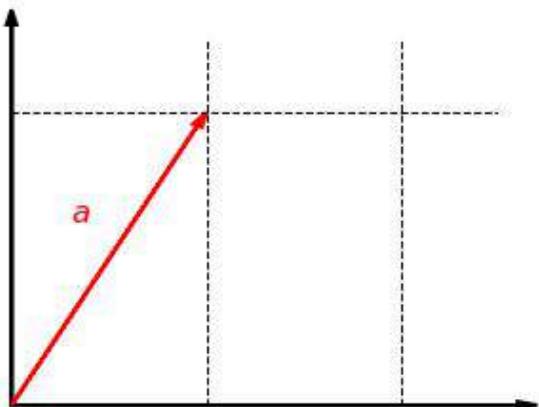
Matrix as a linear transformation



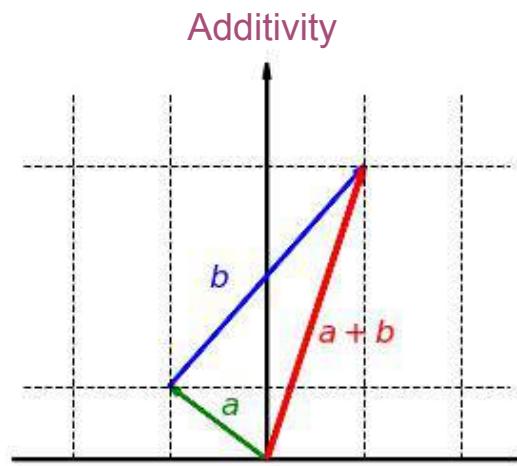
The function $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with
 $f(x, y) = (2x, y)$ is a linear map.

$$\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} M \begin{bmatrix} x \\ y \end{bmatrix} a = \begin{bmatrix} 2x \\ y \end{bmatrix} f(a)$$

Matrix as a linear transformation



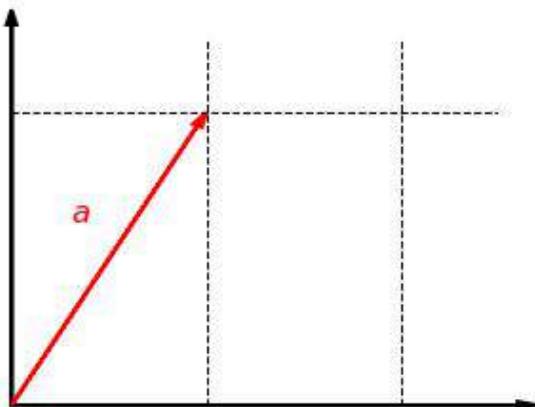
The function $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with $f(x, y) = (2x, y)$ is a linear map.



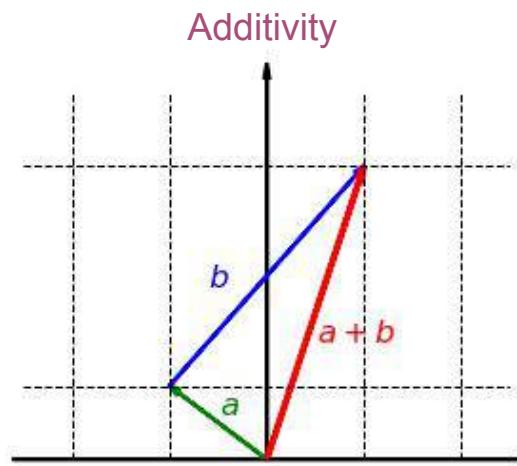
The function $f(x, y) = (2x, y)$ is additive:

$$f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$$

Matrix as a linear transformation

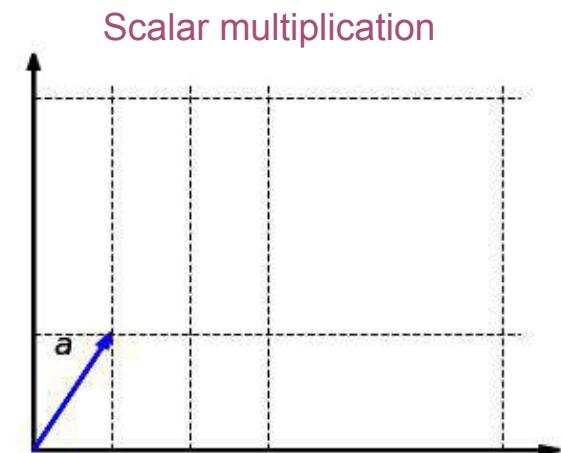


The function $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ with $f(x, y) = (2x, y)$ is a linear map.



The function $f(x, y) = (2x, y)$ is additive:

$$f(\mathbf{a} + \mathbf{b}) = f(\mathbf{a}) + f(\mathbf{b})$$



The function $f(x, y) = (2x, y)$ is homogeneous:

$$f(\lambda \mathbf{a}) = \lambda f(\mathbf{a})$$

Matrix as a linear transformation

1) **Additivity** $\rightarrow f(\mathbf{a}+\mathbf{b})=f(\mathbf{a})+f(\mathbf{b})$

2) **Scalar multiplication** $\rightarrow f(\lambda \mathbf{a})=\lambda f(\mathbf{a})$



Any linear map (between finite-dimensional vector spaces) can be represented as a matrix multiplied with a vector: $f(\mathbf{a})=\mathbf{M}\mathbf{a}$

Matrix as a linear transformation

1) **Additivity** $\rightarrow f(\mathbf{a}+\mathbf{b})=f(\mathbf{a})+f(\mathbf{b})$

2) **Scalar multiplication** $\rightarrow f(\lambda \mathbf{a})=\lambda f(\mathbf{a})$

} Any linear map (between finite-dimensional vector spaces) can be represented as a matrix multiplied with a vector: $f(\mathbf{a})=\mathbf{M}\mathbf{a}$

Question:

Is translation a linear map?

$$\text{e.g. } f(\mathbf{a}) = \mathbf{a} + \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

Matrix as a linear transformation

1) **Additivity** $\rightarrow f(\mathbf{a}+\mathbf{b})=f(\mathbf{a})+f(\mathbf{b})$

2) **Scalar multiplication** $\rightarrow f(\lambda \mathbf{a})=\lambda f(\mathbf{a})$

} Any linear map (between finite-dimensional vector spaces) can be represented as a matrix multiplied with a vector: $f(\mathbf{a})=\mathbf{M}\mathbf{a}$

Question:

Is translation a linear map?

$$\text{e.g. } f(\mathbf{a}) = \mathbf{a} + \begin{bmatrix} 5 \\ 5 \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 5 \\ 5 \end{bmatrix}$$

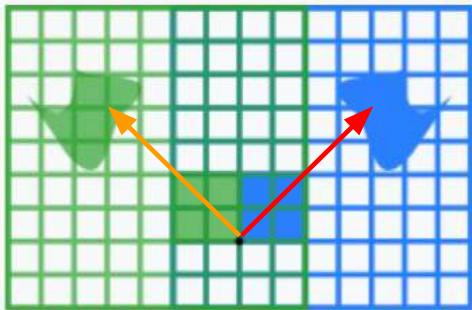
Spoiler: No.
Exercises!

Questions?

Examples of transformation matrices

Reflection through the vertical axis

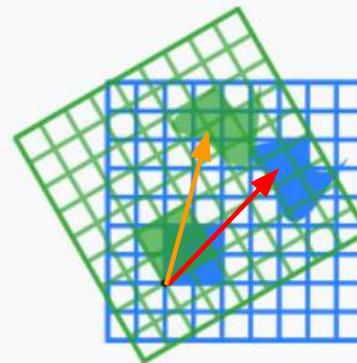
$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \mathbf{v} = \mathbf{w}$$



Rotation

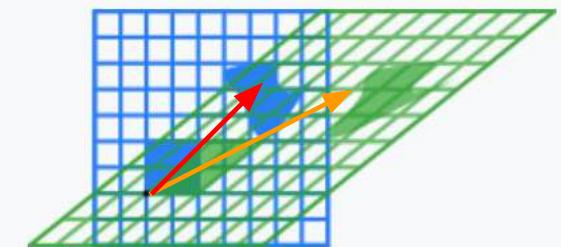
by $\pi/6 = 30^\circ$

$$\begin{bmatrix} \cos\left(\frac{\pi}{6}\right) & -\sin\left(\frac{\pi}{6}\right) \\ \sin\left(\frac{\pi}{6}\right) & \cos\left(\frac{\pi}{6}\right) \end{bmatrix} \cdot \mathbf{v} = \mathbf{w}$$



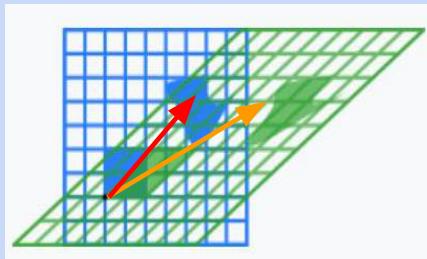
Horizontal shear
with $m = 1.25$.

$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \cdot \mathbf{v} = \mathbf{w}$$



Special case

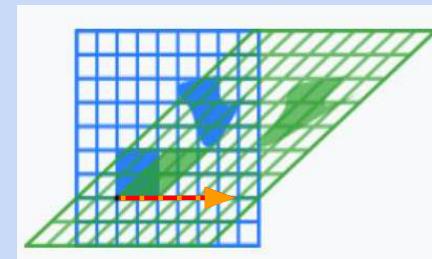
Shear example 1



$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -4 \end{bmatrix} = \begin{bmatrix} -3 \\ -4 \end{bmatrix}$$

M **v** **w**

Shear example 2

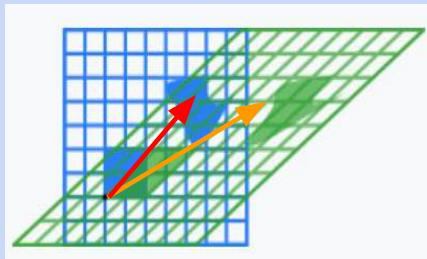


$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

M **v** **w**

Special case

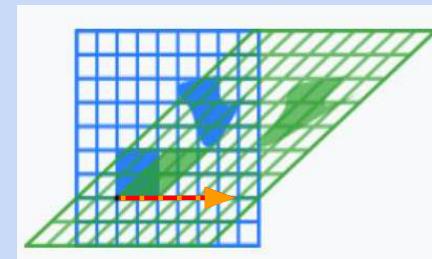
Shear example 1



$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ -4 \end{bmatrix} = \begin{bmatrix} -3 \\ -4 \end{bmatrix}$$

M **v** **w**

Shear example 2



$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

M **v** **w**

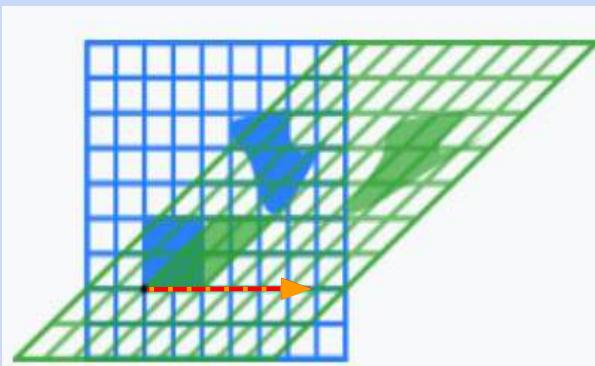
Eigenvectors!

Eigenvalues and eigenvectors

Shear example

$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

M **v** **w**



- Vector **v** is unaffected by the transformation **M**:

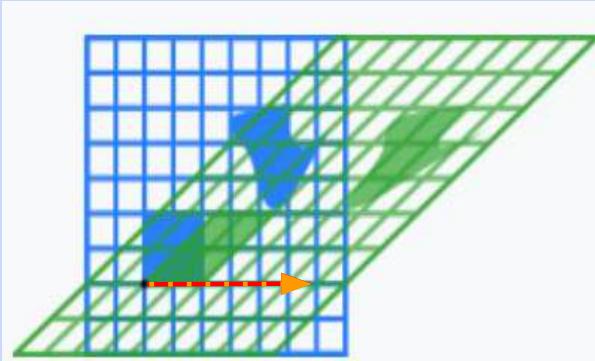
$$\mathbf{Mv} = \mathbf{v}$$

Eigenvalues and eigenvectors

Shear example

$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

M **v** **w**



- Vector **v** is unaffected by the transformation **M**:

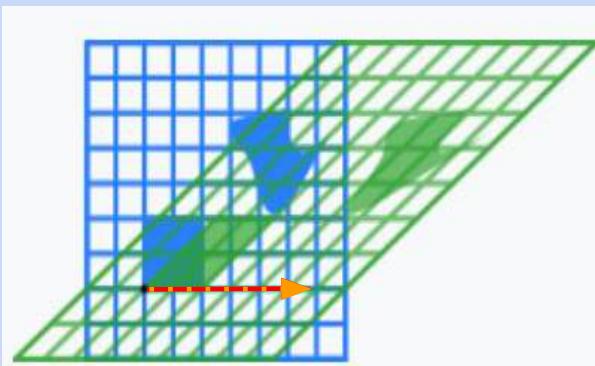
$$\mathbf{M}\mathbf{v} = \mathbf{v}$$

- This is surprising behaviour: look at how the whole space is 'bent' by the shear (blue→green)!

Eigenvalues and eigenvectors

Shear example

$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$



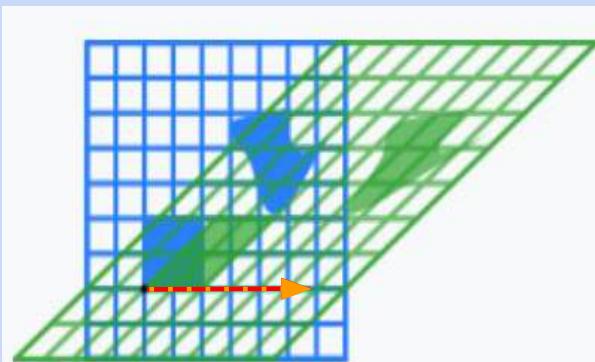
- Vector \mathbf{v} is unaffected by the transformation \mathbf{M} :
$$\mathbf{M}\mathbf{v} = \mathbf{v}$$
 - This is surprising behaviour: look at how the whole space is 'bent' by the shear (blue → green)!
 - Generalized, for a scalar λ :

$$\mathbf{M} \mathbf{v} = \lambda \mathbf{v}$$

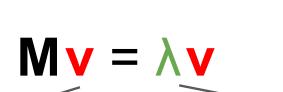
Eigenvalues and eigenvectors

Shear example

$$\begin{bmatrix} 1 & 1.25 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$



- Vector \mathbf{v} is unaffected by the transformation \mathbf{M} :
$$\mathbf{M}\mathbf{v} = \mathbf{v}$$
 - This is surprising behaviour: look at how the whole space is 'bent' by the shear (blue \rightarrow green)!
 - Generalized, for a scalar λ :
$$\mathbf{M}\mathbf{v} = \lambda\mathbf{v}$$


eigenvector \mathbf{v} eigenvalue λ
 - In this example $\mathbf{v}=[2,0]$ is an eigenvector, and the corresponding eigenvalue is $\lambda=1$.

Questions?

Matrices as data tables

A **data matrix** is an n -by- d matrix which has n samples, and d features (or "attributes"). Each sample is a $1 \times d$ row vector.

d features				
n samples	sepal length	sepal width	petal length	petal width
	5.1	3.5	1.4	0.2
	4.9	3	1.4	0.2
	6.5	3.2	5.1	2
	6.4	2.7	5.3	1.9
	6.8	3	5.5	2.1
	6.7	3.1	4.4	1.4
	5.6	3	4.5	1.5
	5.8	2.7	4.1	1

The "data matrix" (Data Mining The Textbook: Chapter 1.4)

Covariance matrix

n samples

x	y
3	5
7	2
-1	3
6	0
2	-1
5	4
7	2
6	3

Data matrix

$$\begin{matrix} & \begin{matrix} x & y \end{matrix} \\ \begin{matrix} x \\ y \end{matrix} & \begin{bmatrix} var(x) & cov(x, y) \\ cov(x, y) & var(y) \end{bmatrix} \end{matrix}$$

Covariance matrix

x	y	z
3	5	0
7	2	3
-1	3	4
6	0	1
2	-1	6
5	4	-2
7	2	2
6	3	4

Data matrix

$$\begin{matrix} & \begin{matrix} x & y & z \end{matrix} \\ \begin{matrix} x \\ y \\ z \end{matrix} & \begin{bmatrix} var(x) & cov(x, y) & cov(x, z) \\ cov(x, y) & var(y) & cov(y, z) \\ cov(x, z) & cov(y, z) & var(z) \end{bmatrix} \end{matrix}$$

Covariance matrix

Covariance matrix

Definition: Given variables x and y , and their means μ_x and μ_y , their covariance is:

$$\text{cov}(x, y) = \langle (x_i - \mu_x)(y_i - \mu_y) \rangle_i$$

where the brackets $\langle \cdot \rangle$ denote the empirical mean over all samples $i = 1, \dots, n$.

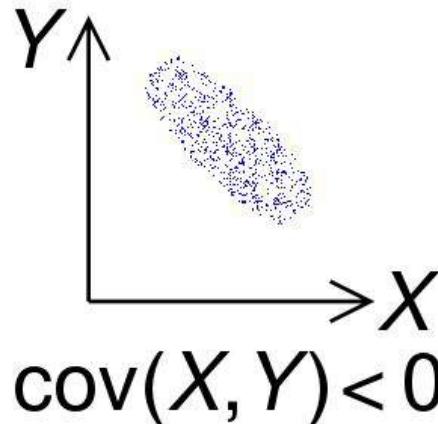
Similarly,

$$\text{cov}(x, x) = \langle (x_i - \mu_x)^2 \rangle_i = \text{var}(x)$$

Covariance matrix is **symmetric**:

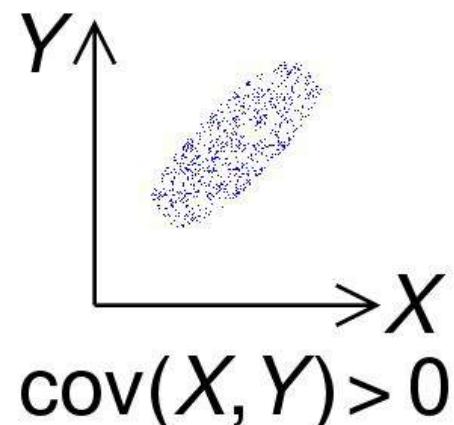
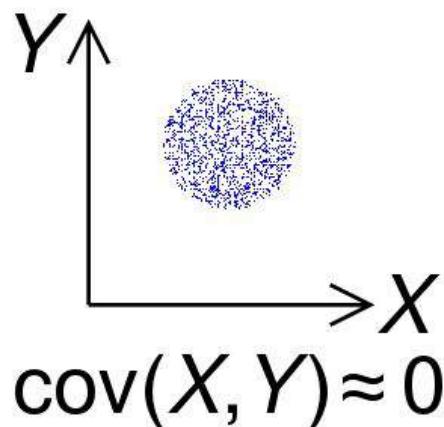
$$\begin{matrix} & x & y \\ x & \left[\begin{matrix} \text{var}(x) & \text{cov}(x, y) \\ \text{cov}(x, y) & \text{var}(y) \end{matrix} \right] \\ y & \end{matrix}$$

Covariance matrix



$$\begin{matrix} & \begin{matrix} x & y \end{matrix} \\ \begin{matrix} x \\ y \end{matrix} & \begin{bmatrix} \text{var}(x) & \text{cov}(x, y) \\ \text{cov}(x, y) & \text{var}(y) \end{bmatrix} \end{matrix}$$

Covariance matrix



Property:

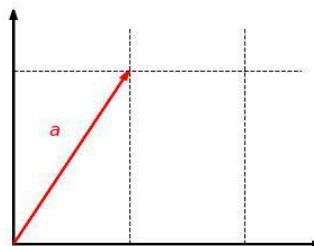
Covariance matrix is symmetric:

- Eigenvectors are orthogonal.
- Eigenvector with the largest eigenvalue points in the direction of largest variance in the data matrix.

Important in PCA algorithm.

Summary

- Matrices are useful!
- Matrices are **linear transformations** with additivity and scalar multiplication properties.



- When multiplied with vectors, matrices can provide geometric transformations such as rotation.
- Matrices can be used as **data tables** to store and manipulate important information.

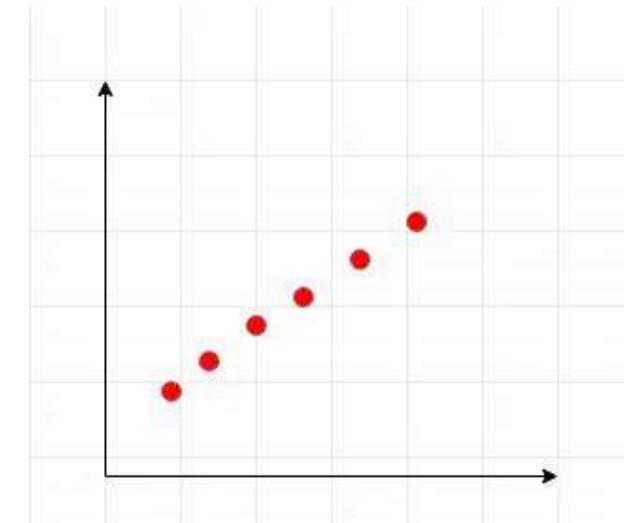
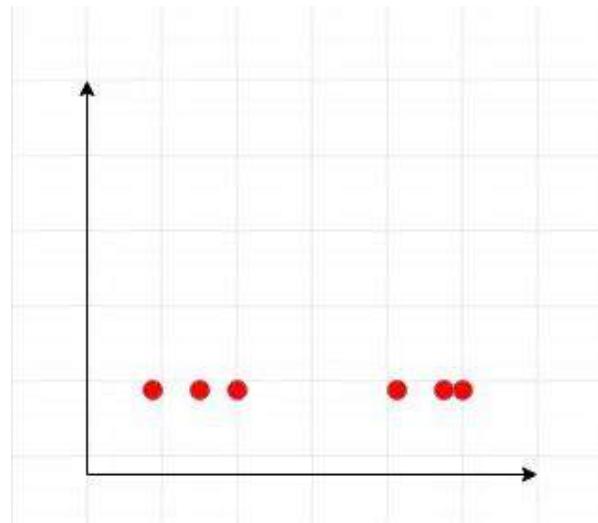
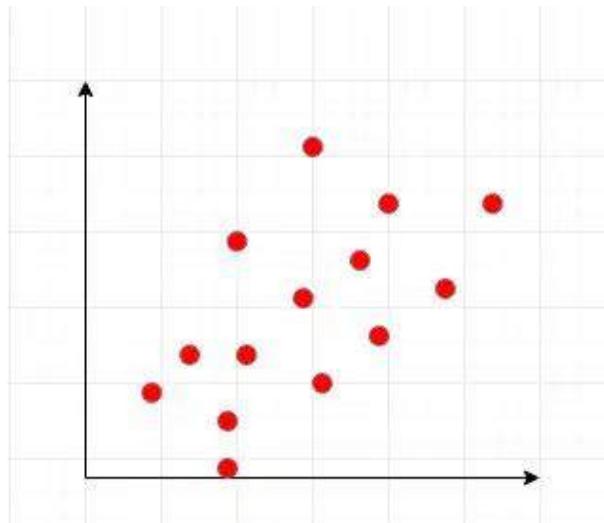
d features				
n samples	sepal length	sepal width	petal length	petal width
	5.1	3.5	1.4	0.2
	4.9	3	1.4	0.2
	6.5	3.2	5.1	2
	6.4	2.7	5.3	1.9
	6.8	3	5.5	2.1
	6.7	3.1	4.4	1.4
	5.6	3	4.5	1.5
	5.8	2.7	4.1	1

Questions?

Principal component analysis (PCA)

PCA

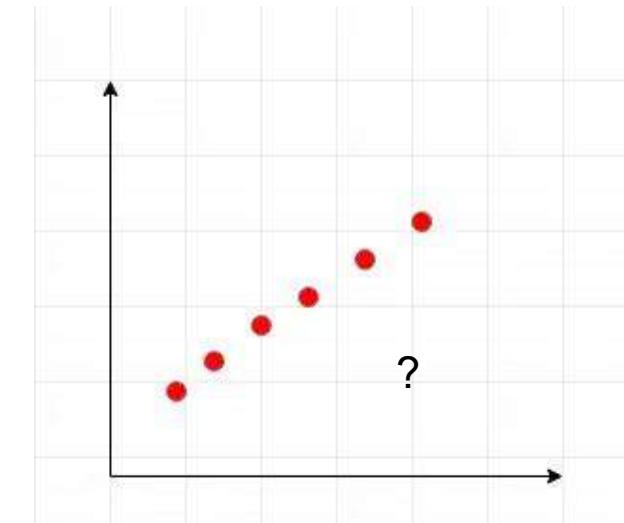
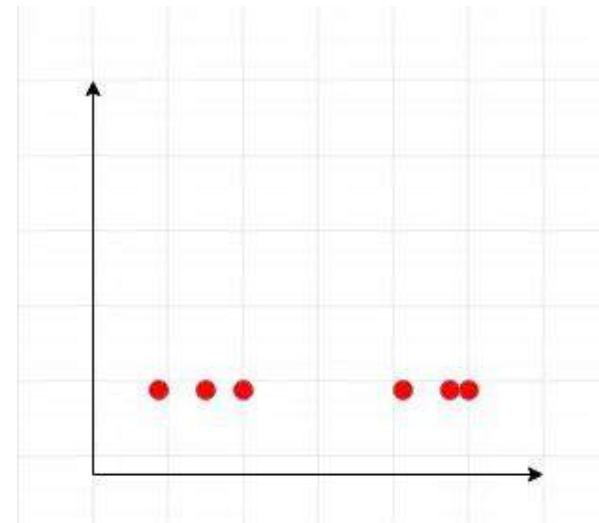
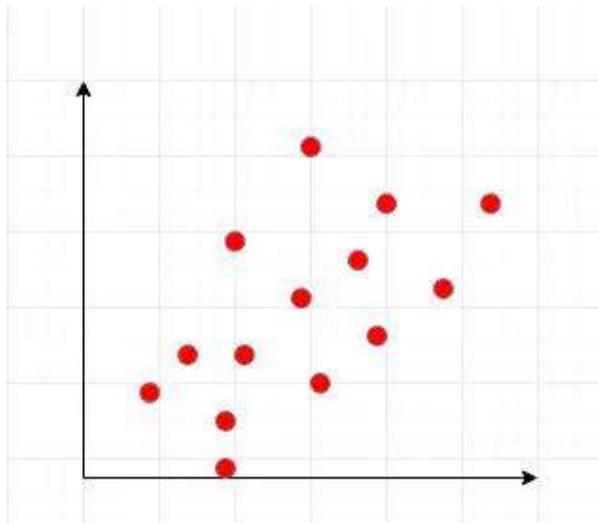
What's the dimensionality of the following datasets?



PCA

What's the dimensionality of the following datasets?

What would you say are the "intrinsic" dimensions?

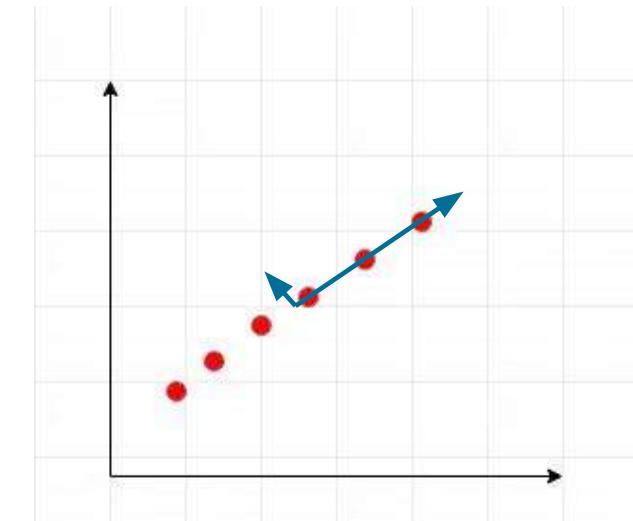
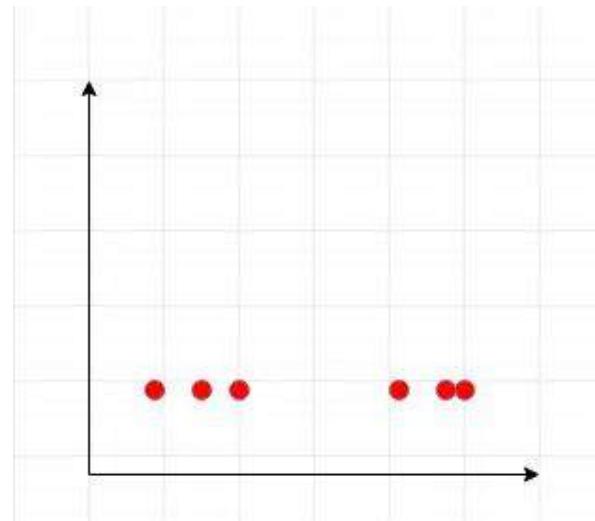
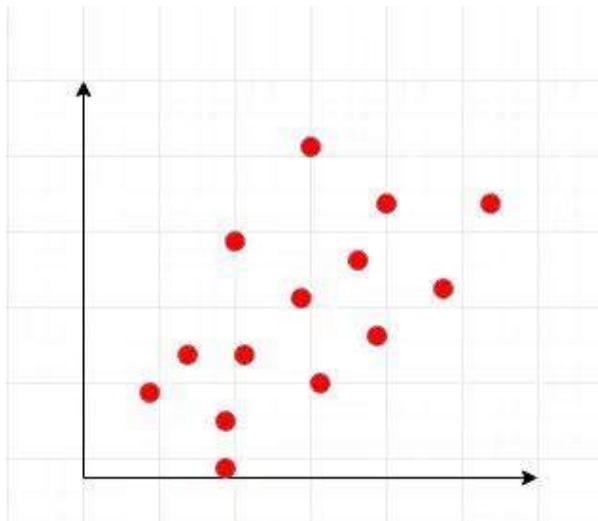


Often, when there is little variance in one dimension/variable of the dataset, it is redundant, i.e. not very informative.

PCA

What's the dimensionality of the following datasets?

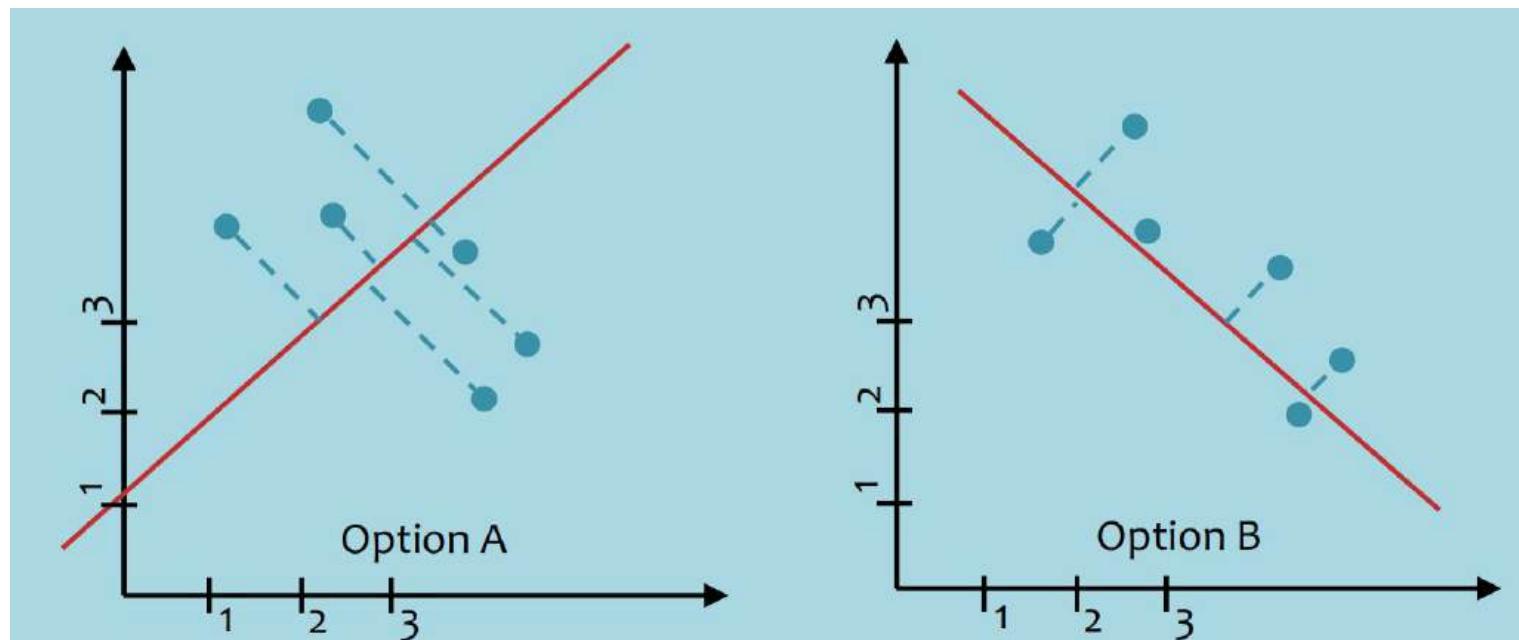
What would you say are the "intrinsic" dimensions?



Often, when there is little variance in one dimension/variable of the dataset, it is redundant, i.e. not very informative.

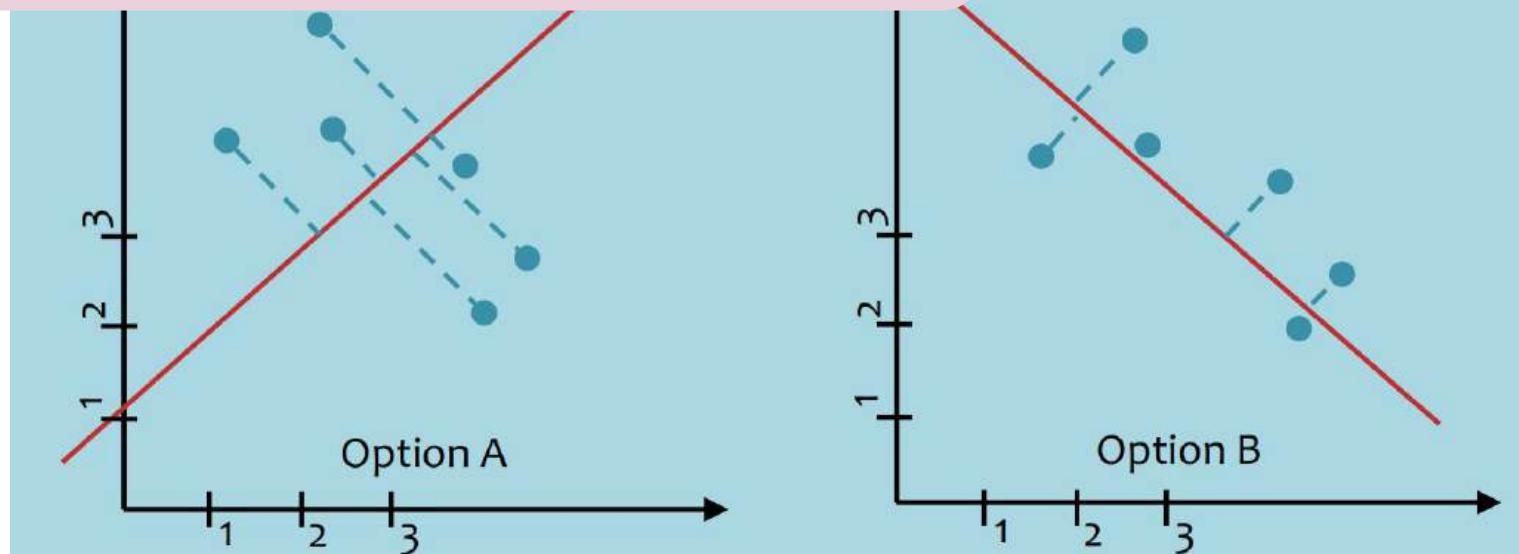
Project all samples (2D) onto a line (1D)

Which red line would you pick?



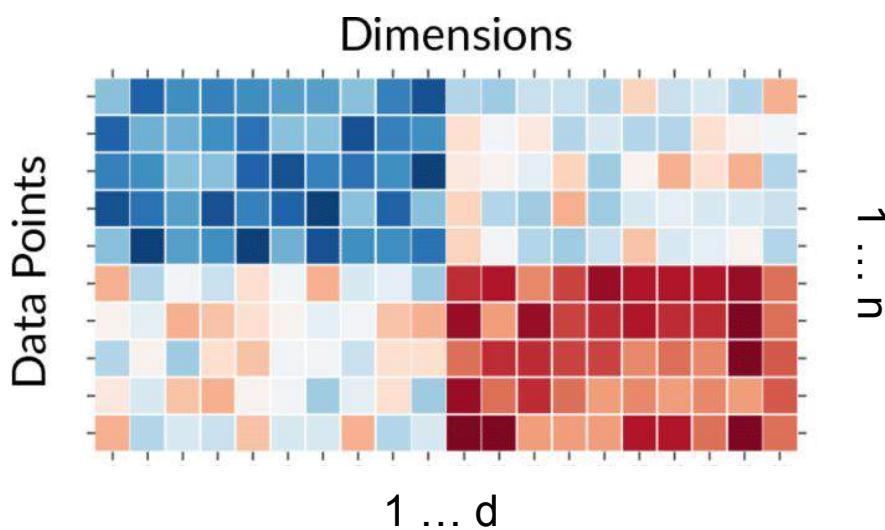
Project all samples (2D) onto a line (1D)

Minimizing the reprojection error is equivalent
to **maximizing** variance over the projection line!



PCA - Algorithm

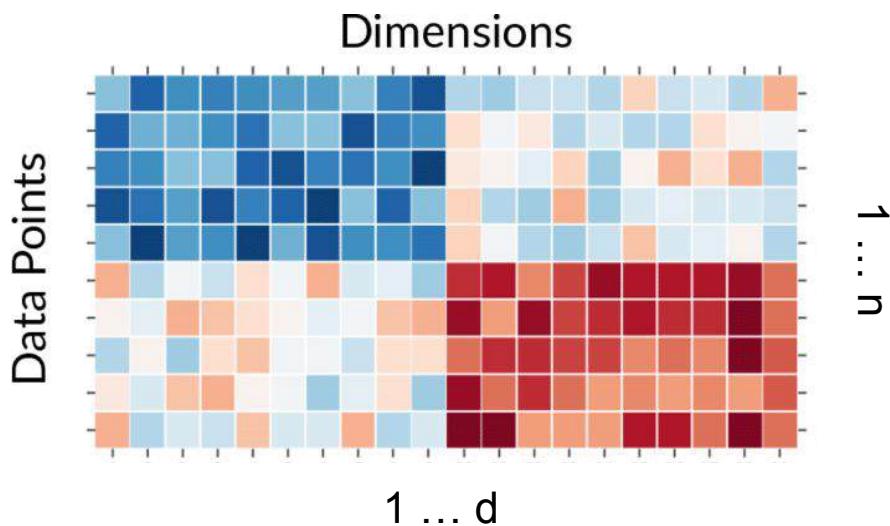
Data matrix \mathbf{X} , with n data points,
each with d features



- 1) Normalize the data
- 2) Compute the d -by- d covariance matrix (with elements $\text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ for each pair of columns of the data matrix \mathbf{X}).
- 3) Compute the eigenvectors of the covariance matrix (principal components), and the corresponding eigenvalues.
- 4) Order the eigenvectors by the size of their eigenvalues, in descending order, to construct a transformation matrix \mathbf{K} .
- 5) Use the transformation matrix \mathbf{K} to project the data points to component space, i.e. matrix multiplication $\mathbf{X}\mathbf{K}$.
- 5.5) For dimensionality reduction, we can choose only the first $m < d$ eigenvectors to construct \mathbf{K} .
- 6) Compute the explained variance for each principle component.
- 7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

PCA - Algorithm

Data matrix \mathbf{X} , with n data points,
each with d features



1) Normalize the data

2 approaches:

1.a) Mean-shifted data \mathbf{x}' :

subtract from each column vector \mathbf{x} its mean μ_x

$$\mathbf{x}'_i = \mathbf{x}_i - \mu_x$$

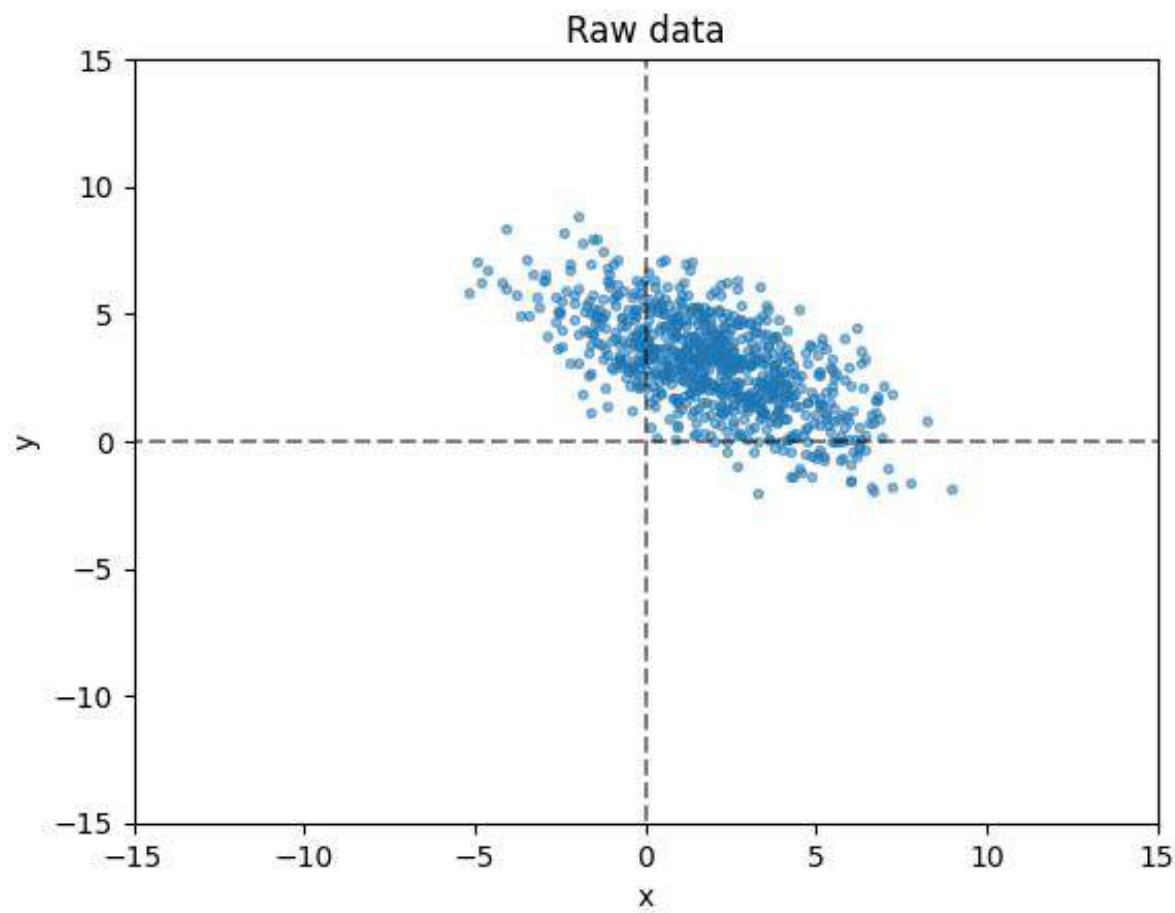
1.b) z-score normalized data \mathbf{x}' :

compute the z-score for each column vector \mathbf{x} , using
its mean μ_x and standard deviation σ_x

$$\mathbf{x}'_i = (\mathbf{x}_i - \mu_x)/\sigma_x$$

Example

800-by-2 data matrix



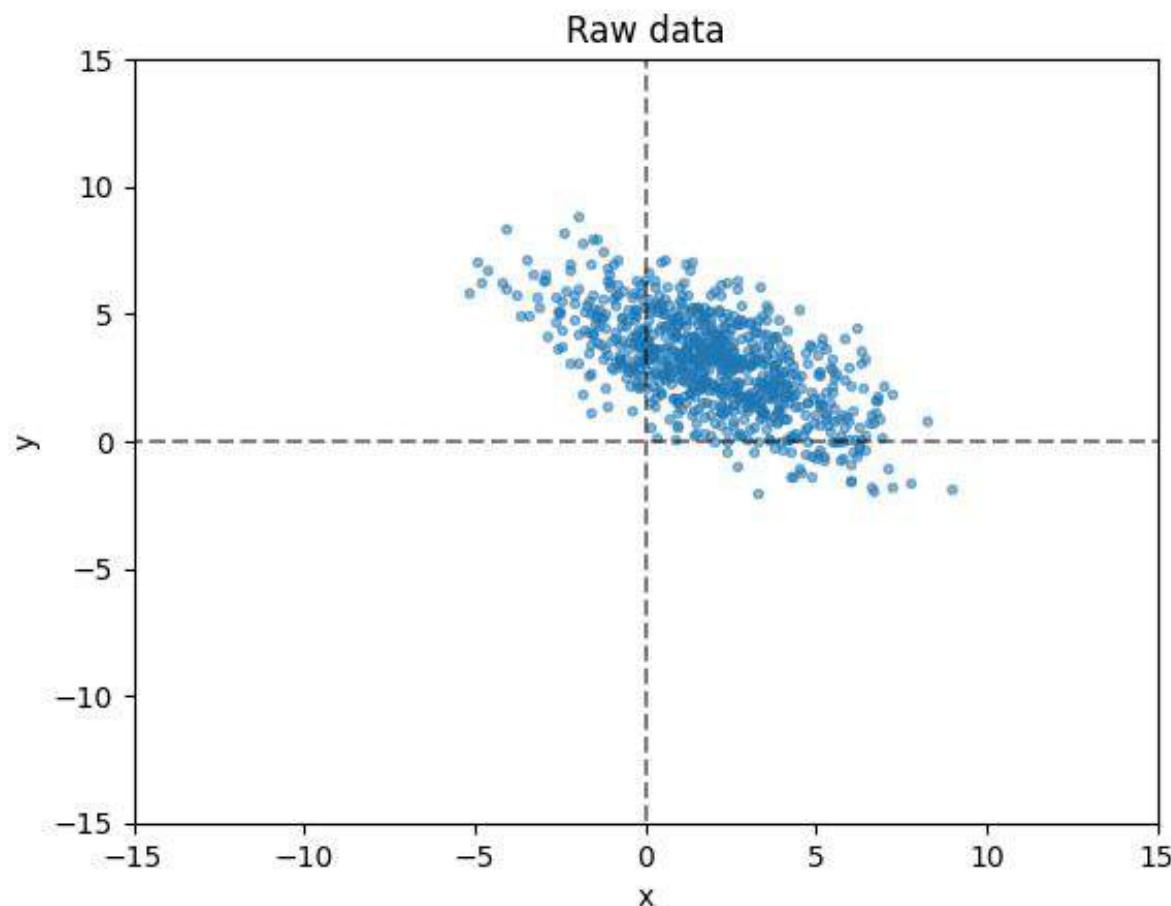
Example

800-by-2 data matrix

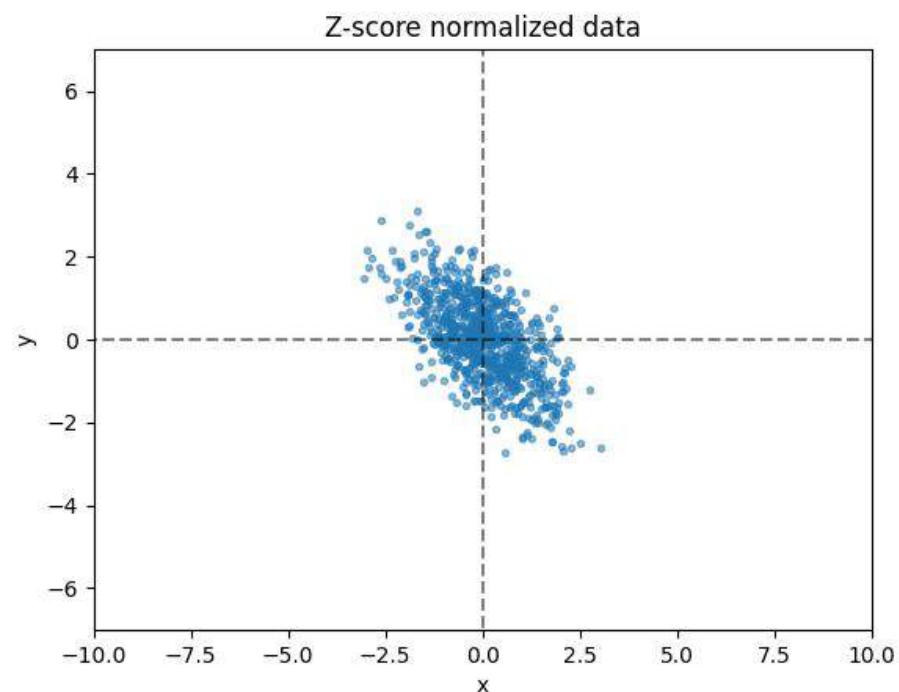
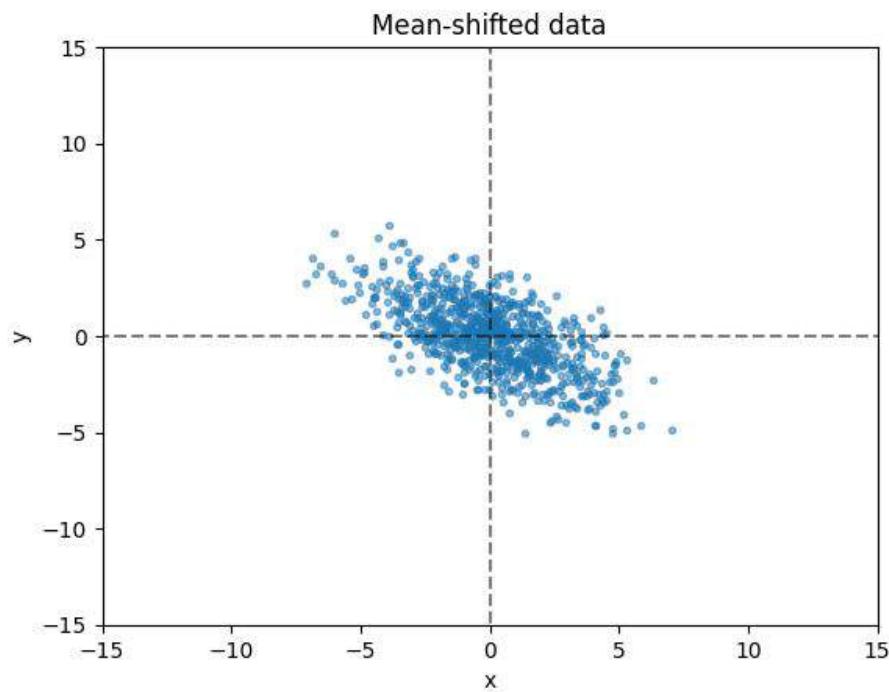
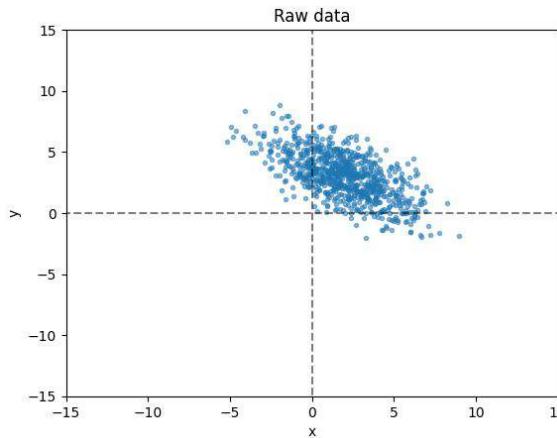
Step 1: Shift the data center to the origin.

Remember: Translation is not a linear map! (So we can't do it by matrix multiplication!)

Shift each column manually!

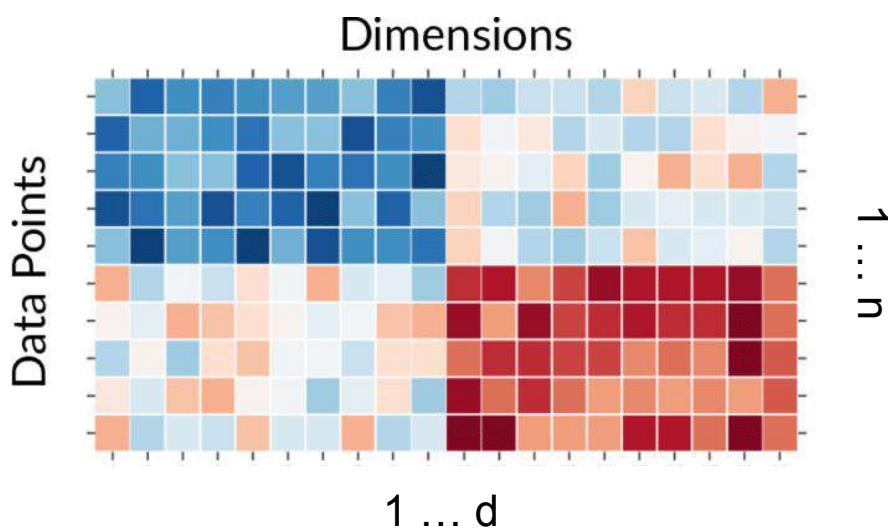


Example



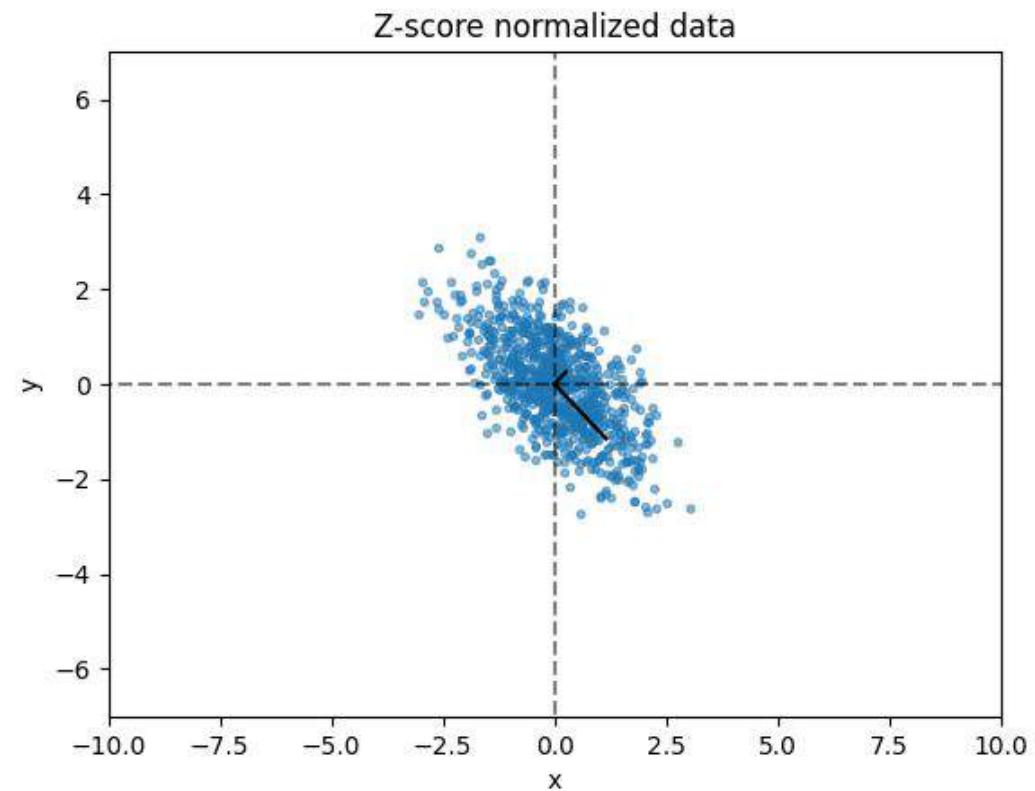
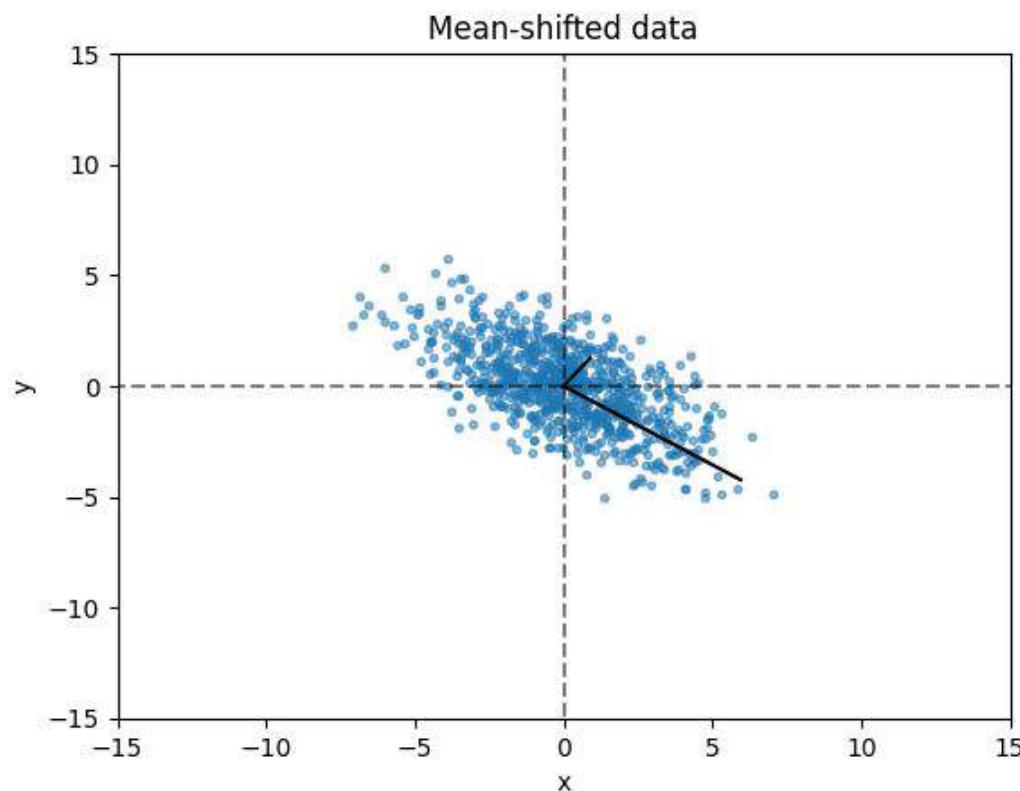
PCA - Algorithm

Data matrix \mathbf{X} , with n data points,
each with d features



- 1) Normalize the data
- 2) Compute the d -by- d covariance matrix (with elements $\text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ for each pair of columns of the data matrix \mathbf{X}).
- 3) Compute the eigenvectors of the covariance matrix (principal components), and the corresponding eigenvalues.
- 4) Order the eigenvectors by the size of their eigenvalues, in descending order, to construct a transformation matrix \mathbf{K} .
- 5) Use the transformation matrix \mathbf{K} to project the data points to component space, i.e. matrix multiplication $\mathbf{X}\mathbf{K}$.
- 5.5) For dimensionality reduction, we can choose only the first $m < d$ eigenvectors to construct \mathbf{K} .
- 6) Compute the explained variance for each principle component.
- 7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

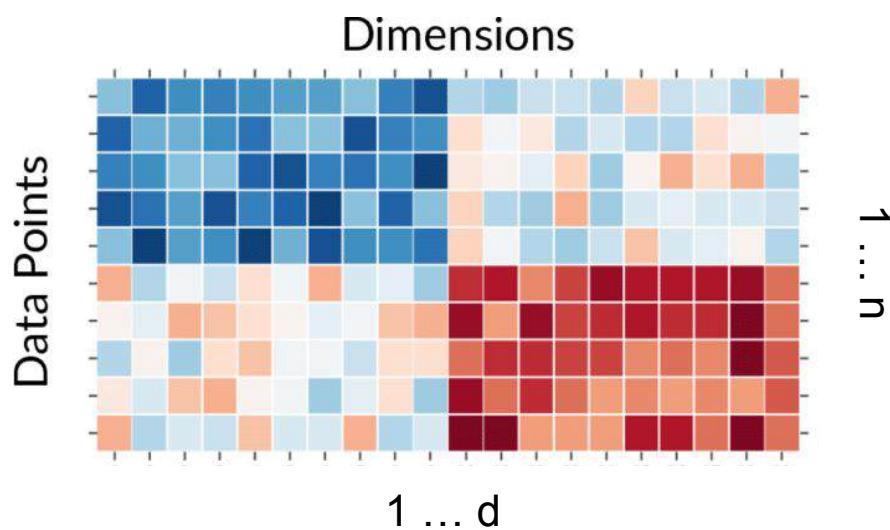
Example



Eigenvectors are orthogonal and the one with the largest eigenvalue points in the direction of largest variance.

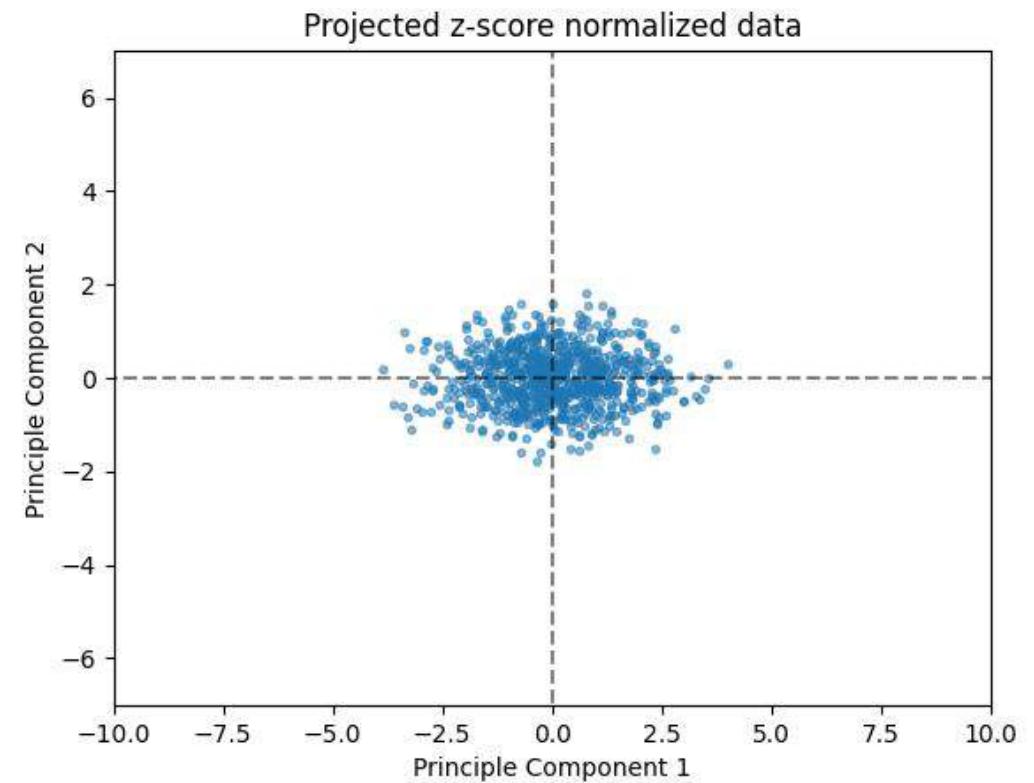
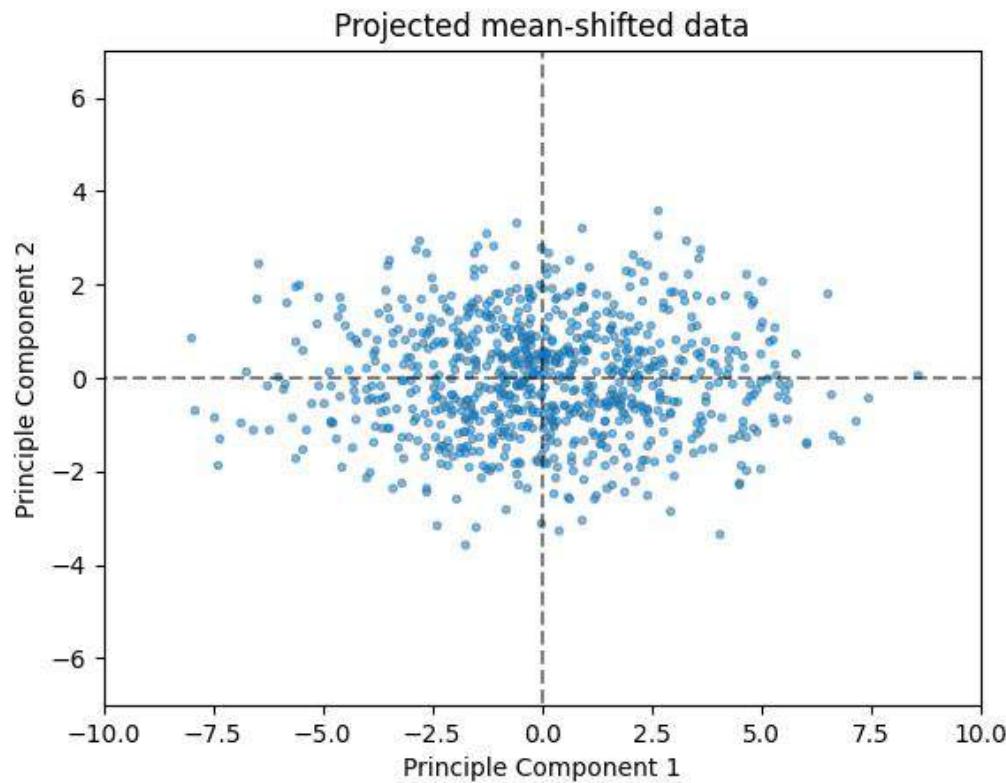
PCA - Algorithm

Data matrix \mathbf{X} , with n data points,
each with d features



- 1) Normalize the data
- 2) Compute the d -by- d covariance matrix (with elements $\text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ for each pair of columns of the data matrix \mathbf{X}).
- 3) Compute the eigenvectors of the covariance matrix (principal components), and the corresponding eigenvalues.
- 4) Order the eigenvectors by the size of their eigenvalues, in descending order, to construct a transformation matrix \mathbf{K} .
- 5) Use the transformation matrix \mathbf{K} to project the data points to component space, i.e. matrix multiplication $\mathbf{X}\mathbf{K}$.
- 5.5) For dimensionality reduction, we can choose only the first $m < d$ eigenvectors to construct \mathbf{K} .
- 6) Compute the explained variance for each principle component.
- 7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

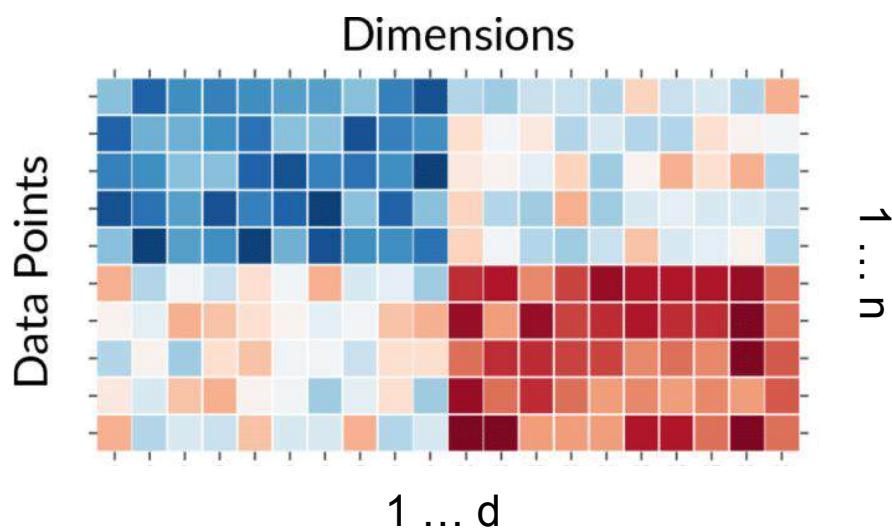
Example



Data is rotated onto the orthogonal principal components.

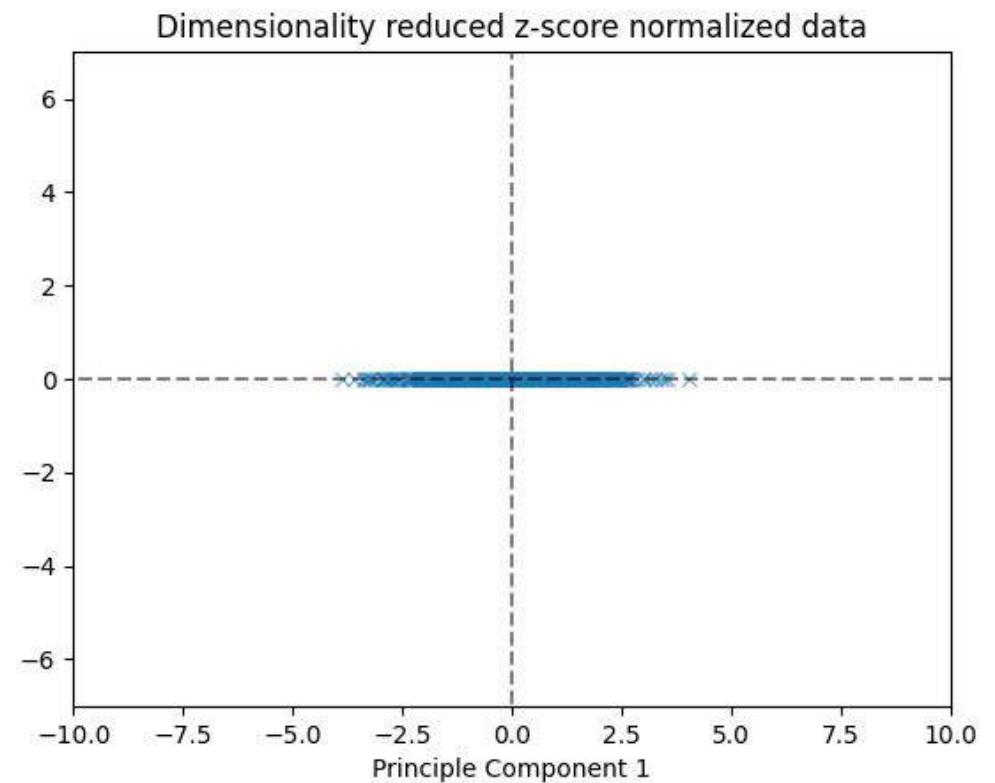
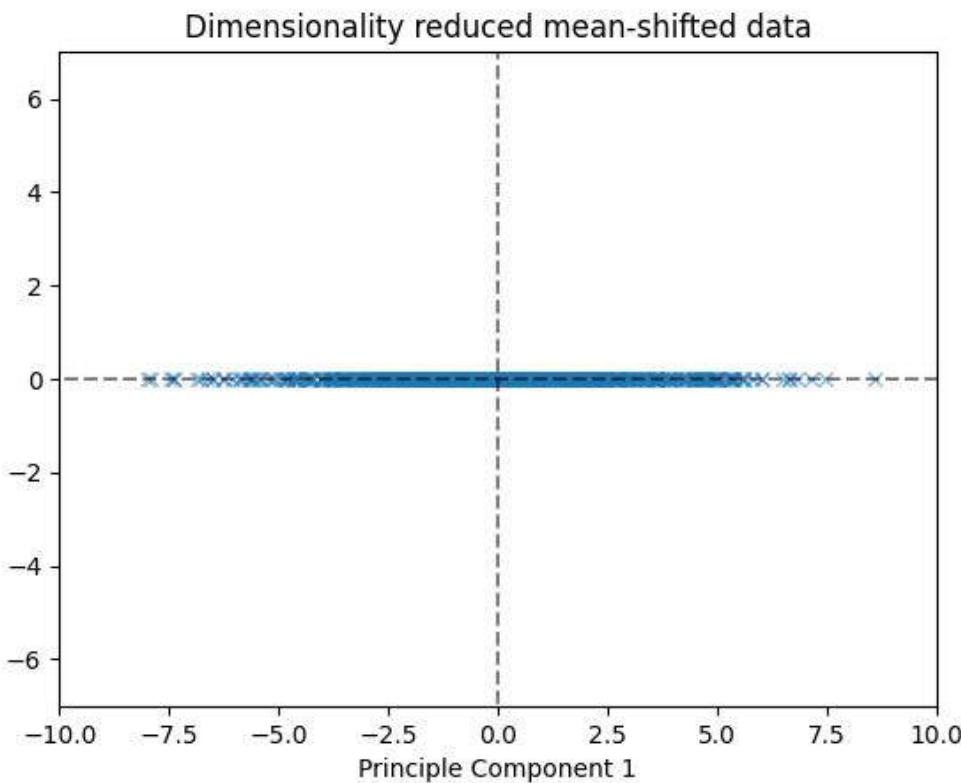
PCA - Algorithm

Data matrix \mathbf{X} , with n data points,
each with d features



- 1) Normalize the data
- 2) Compute the d -by- d covariance matrix (with elements $\text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ for each pair of columns of the data matrix \mathbf{X}).
- 3) Compute the eigenvectors of the covariance matrix (principal components), and the corresponding eigenvalues.
- 4) Order the eigenvectors by the size of their eigenvalues, in descending order, to construct a transformation matrix \mathbf{K} .
- 5) Use the transformation matrix \mathbf{K} to project the data points to component space, i.e. matrix multiplication $\mathbf{X}\mathbf{K}$.
- 5.5)** For dimensionality reduction, we can choose only the first $m < d$ eigenvectors to construct \mathbf{K} .
- 6) Compute the explained variance for each principle component.
- 7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

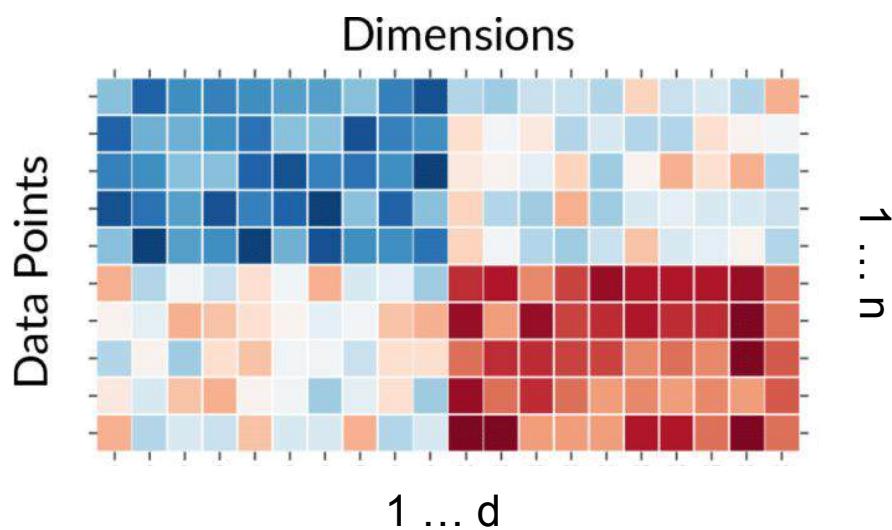
Example



Use only the first principle component (K is d -by-1), to reduce data dimensionality to 1.

PCA - Algorithm

Data matrix \mathbf{X} , with n data points,
each with d features



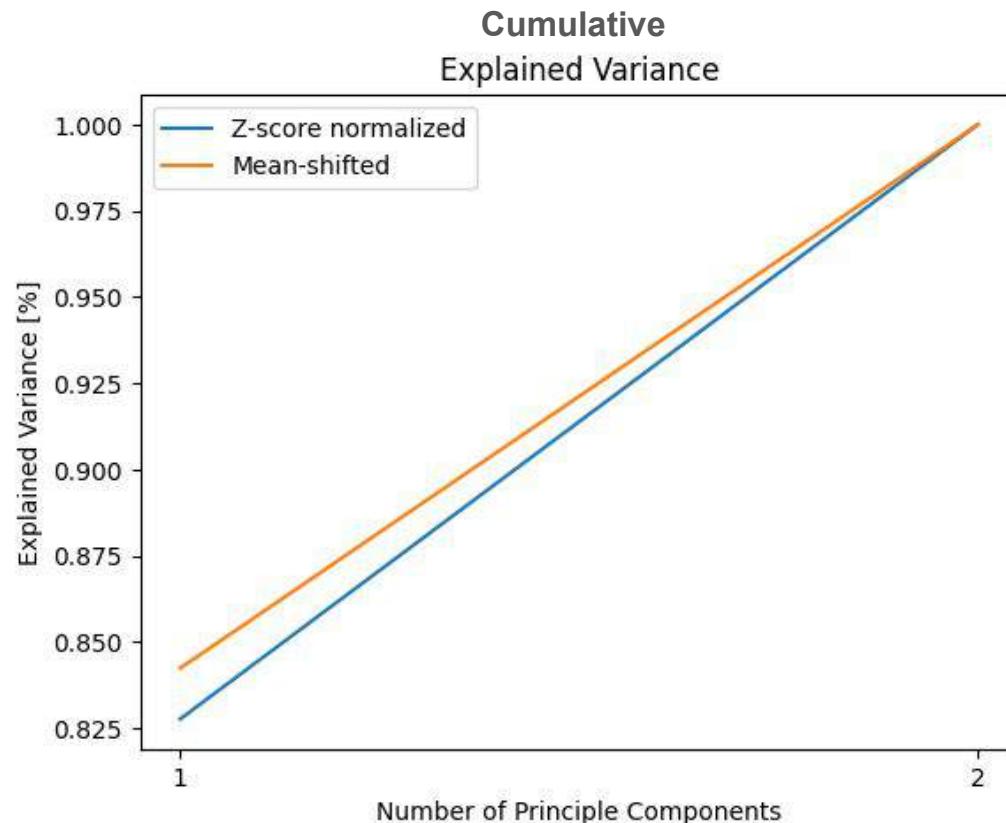
- 1) Normalize the data
- 2) Compute the d -by- d covariance matrix (with elements $\text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ for each pair of columns of the data matrix \mathbf{X}).
- 3) Compute the eigenvectors of the covariance matrix (principal components), and the corresponding eigenvalues.
- 4) Order the eigenvectors by the size of their eigenvalues, in descending order, to construct a transformation matrix \mathbf{K} .
- 5) Use the transformation matrix \mathbf{K} to project the data points to component space, i.e. matrix multiplication $\mathbf{X}\mathbf{K}$.
- 5.5) For dimensionality reduction, we can choose only the first $m < d$ eigenvectors to construct \mathbf{K} .
- 6) Compute the explained variance for each principle component.
- 7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

Example

Explained variance for each principle component (PC) i :

$\text{eigenvalue}_i / \sum_i (\text{eigenvalue}_i)$

is the ratio of the eigenvalue for that PC to the sum of all eigenvalues.



There isn't much difference in explained variance between the 2 normalization methods for this data, i.e. they both preserve variance well when projecting data to 1-dimensional space.

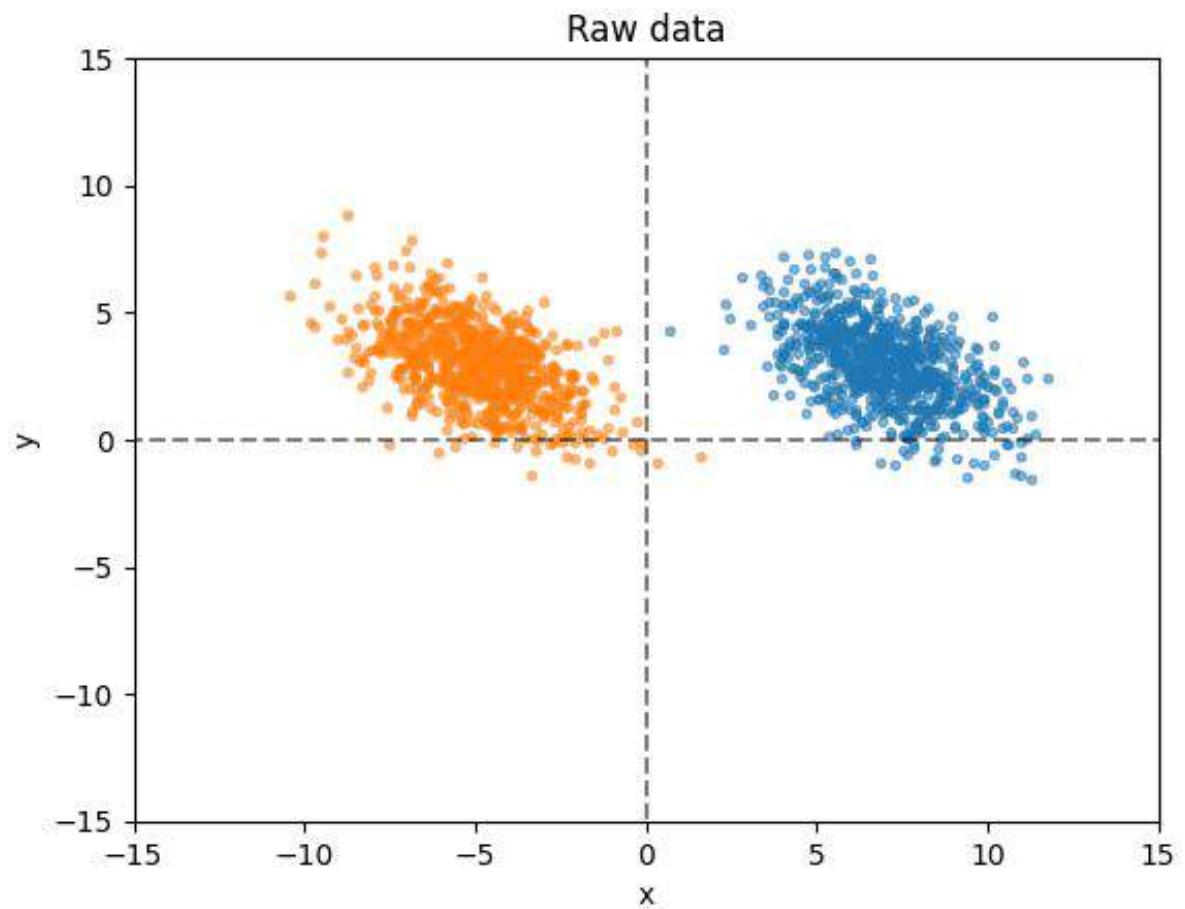
Questions?

Example 2: 2 Data clouds

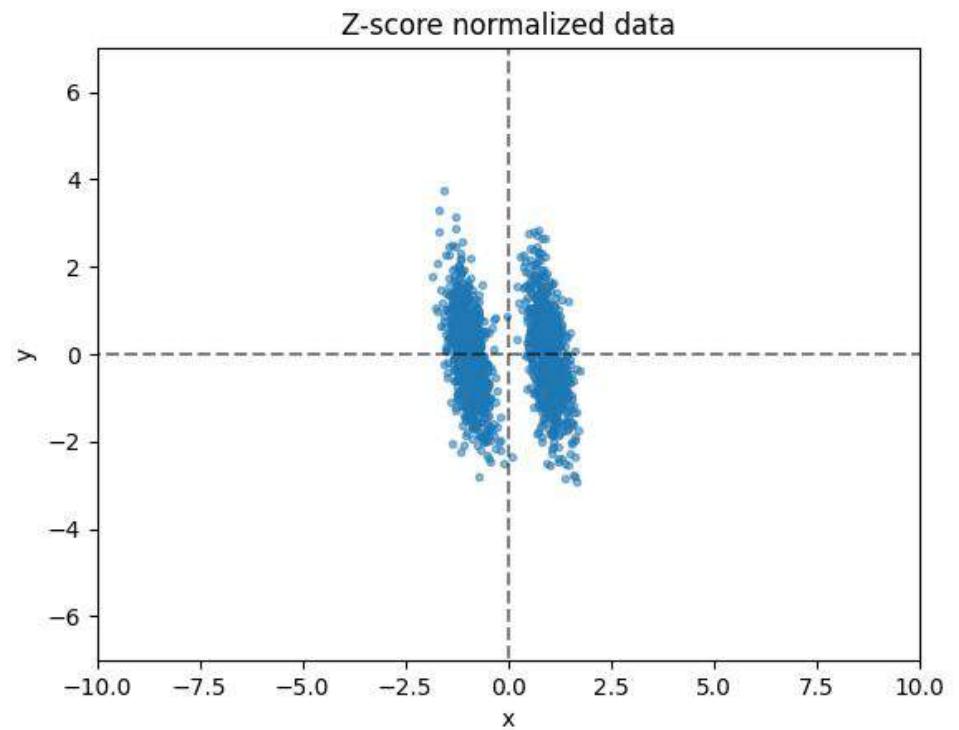
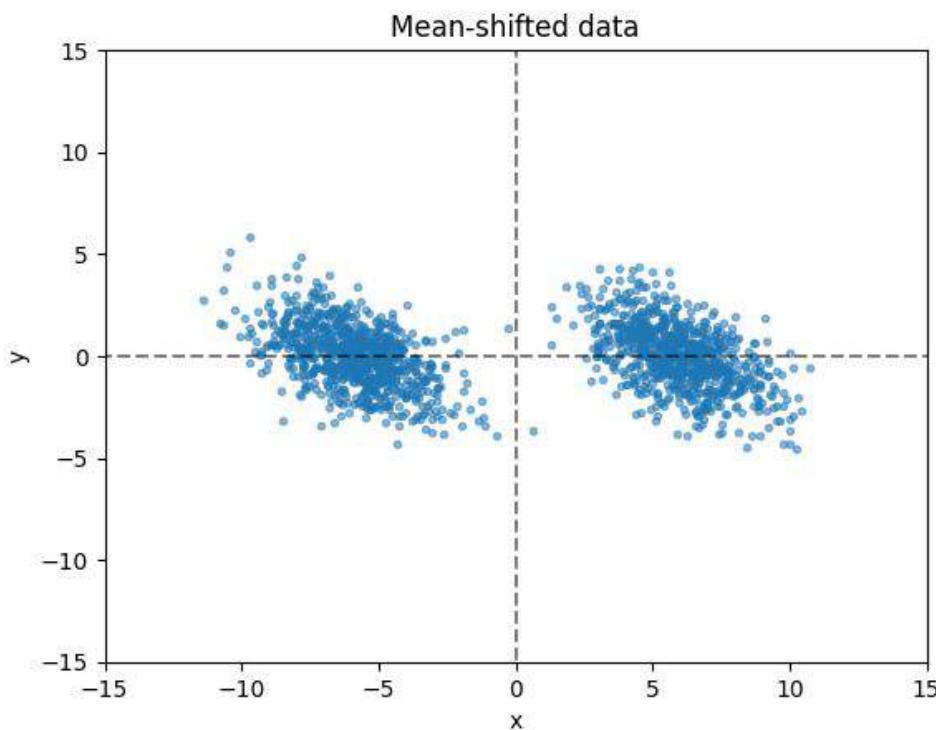
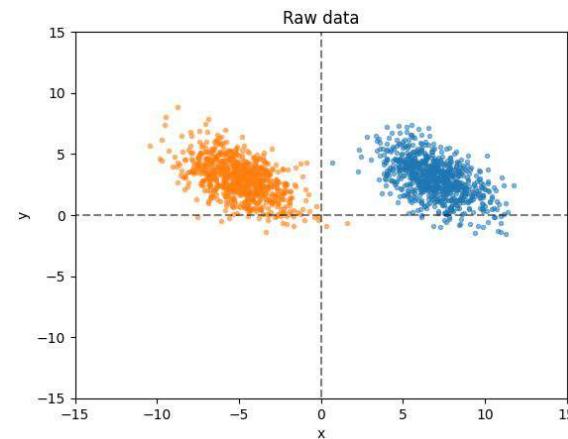
1600-by-2 data matrix

2 distinct clusters of 800 points each

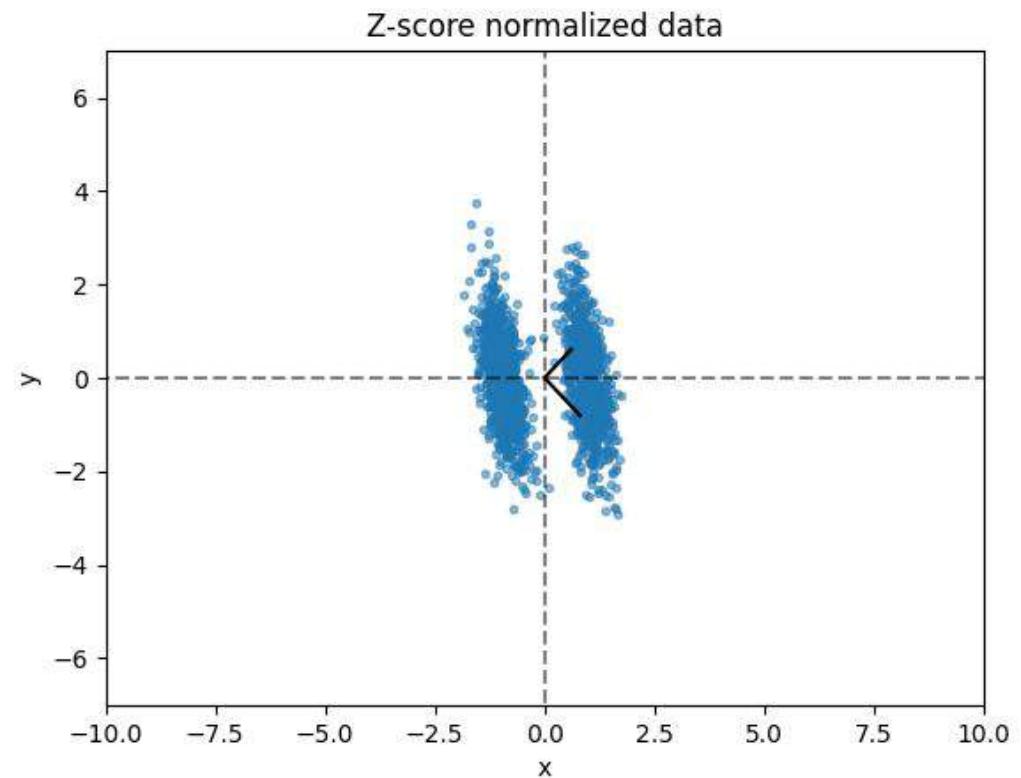
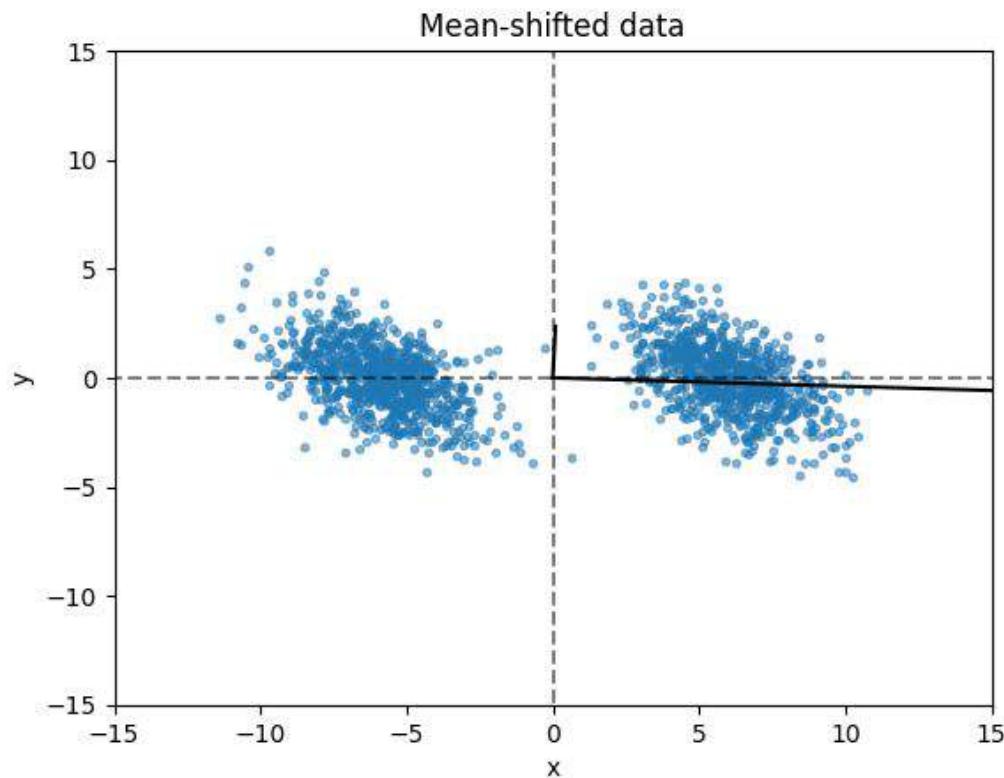
(No labels, colors just for visualization)



Example 2: 2 Data clouds

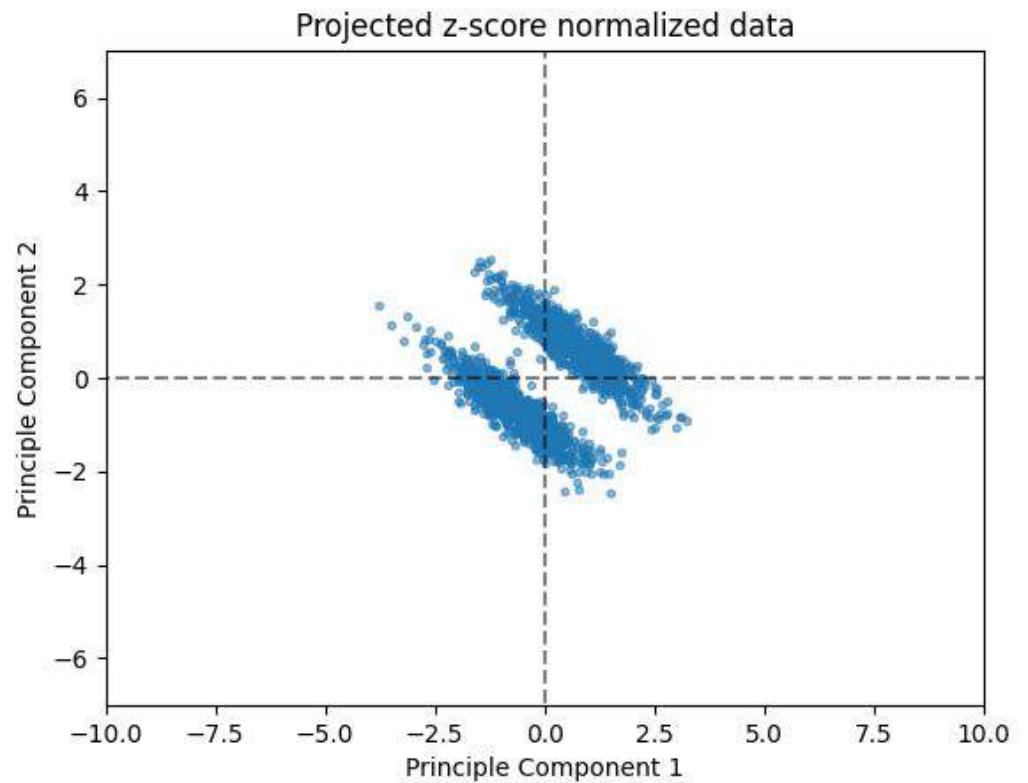
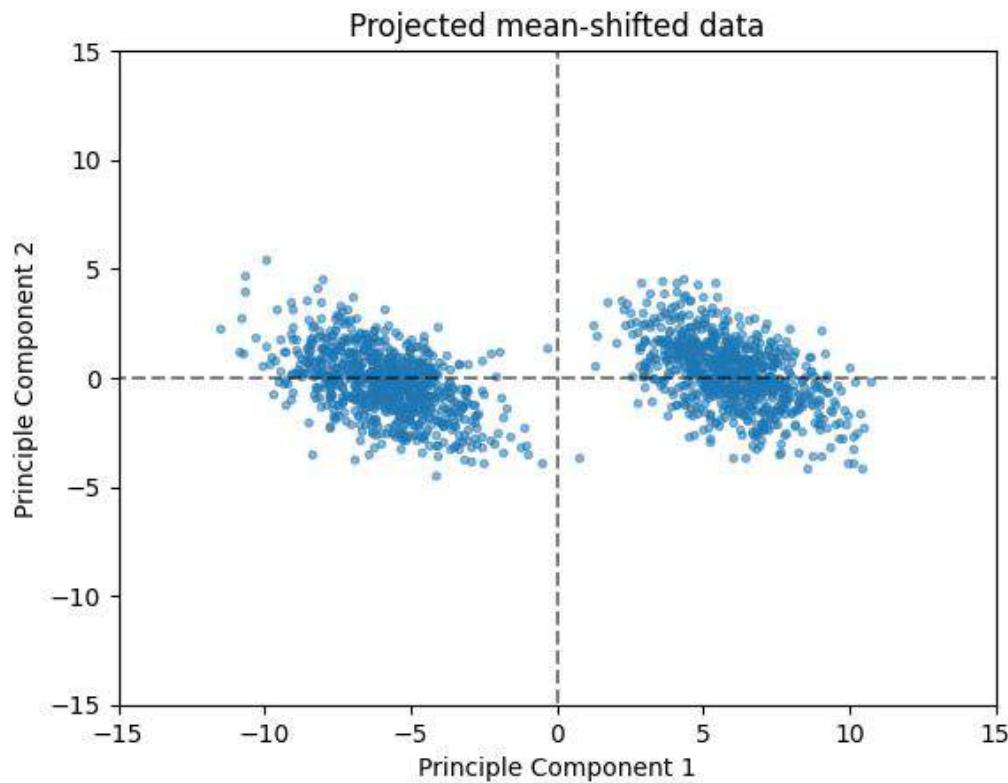


Example 2: 2 Data clouds



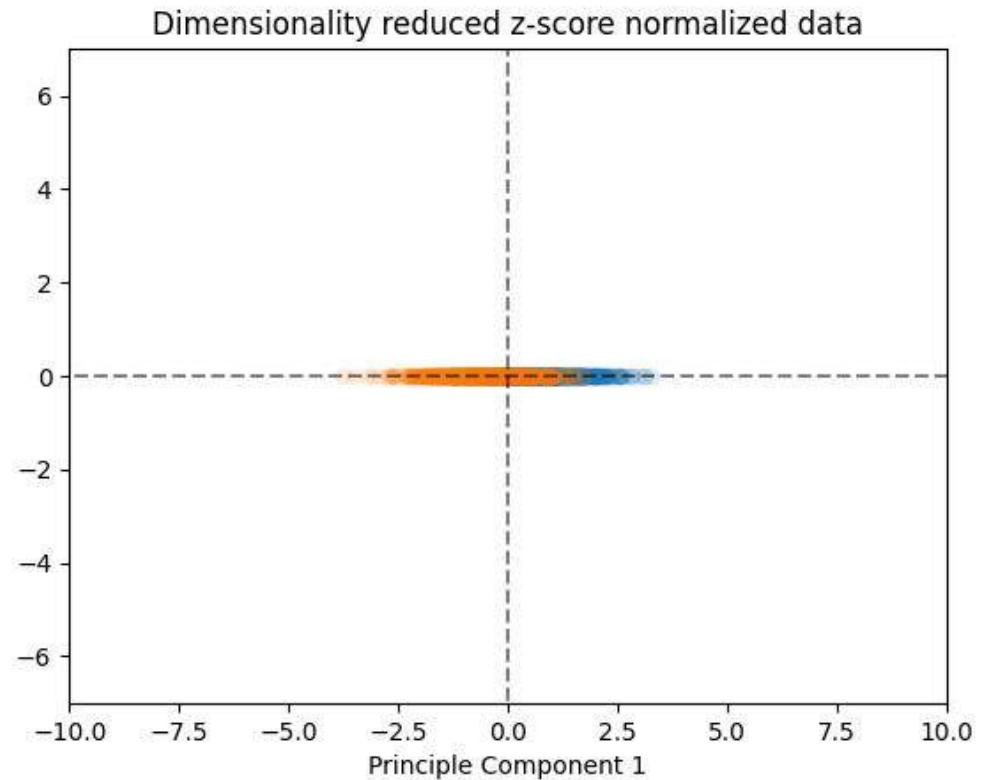
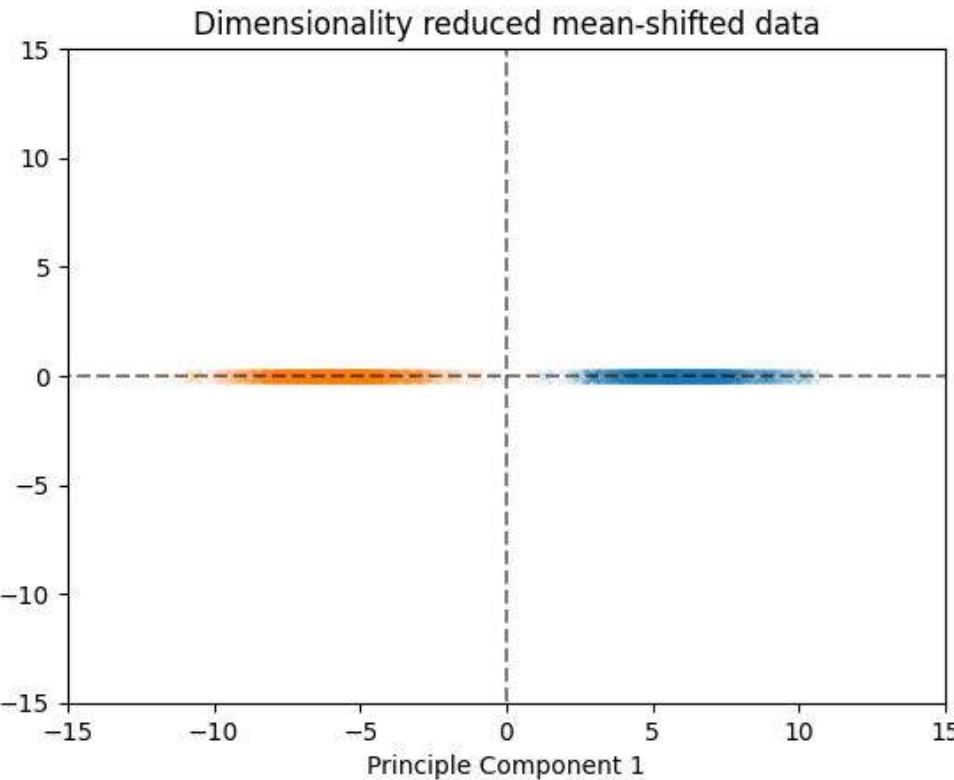
Eigenvectors are orthogonal and the one with the largest eigenvalue points in the direction of largest variance.

Example 2: 2 Data clouds



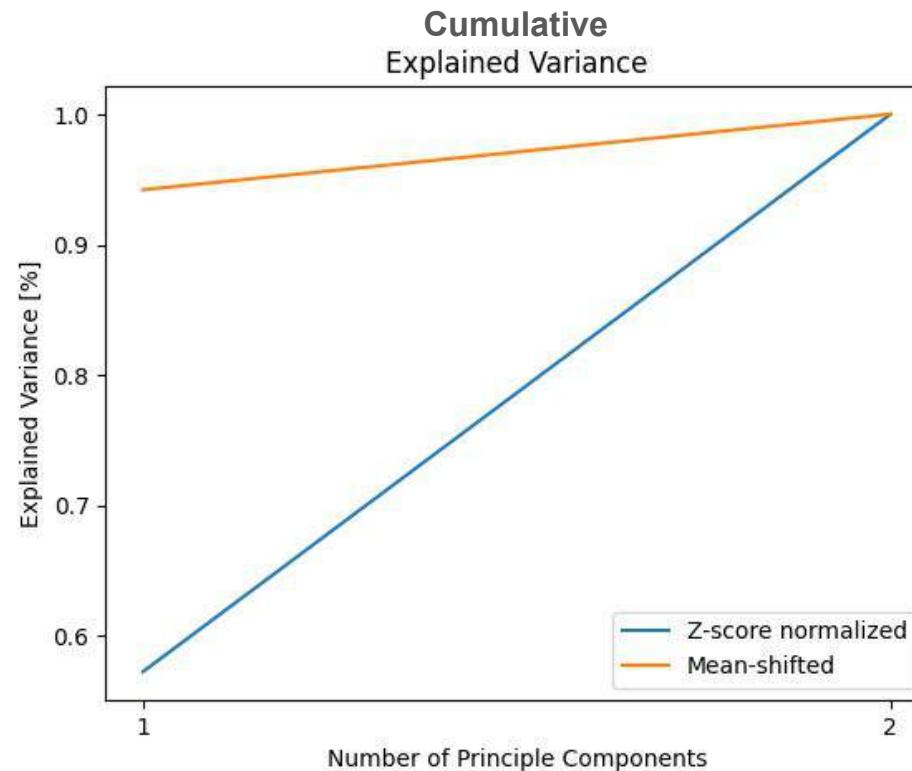
Data is rotated onto the orthogonal principal components.

Example 2: 2 Data clouds



When dimensionality is reduced, z-score normalization does not preserve the clusters!

Example 2: 2 Data clouds



Mean-shift normalization preserves more of the variance in the first PC.

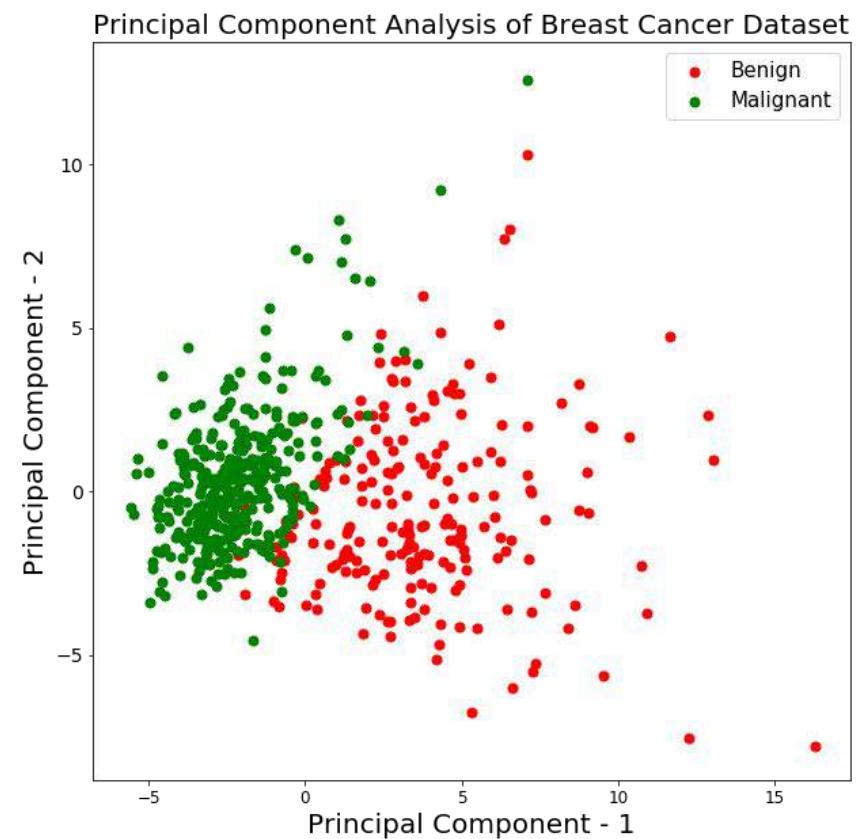
High dimensional example

Breast cancer dataset

569-by-30 data matrix

(example features: tumour area, perimeter, smoothness, symmetry, ...)

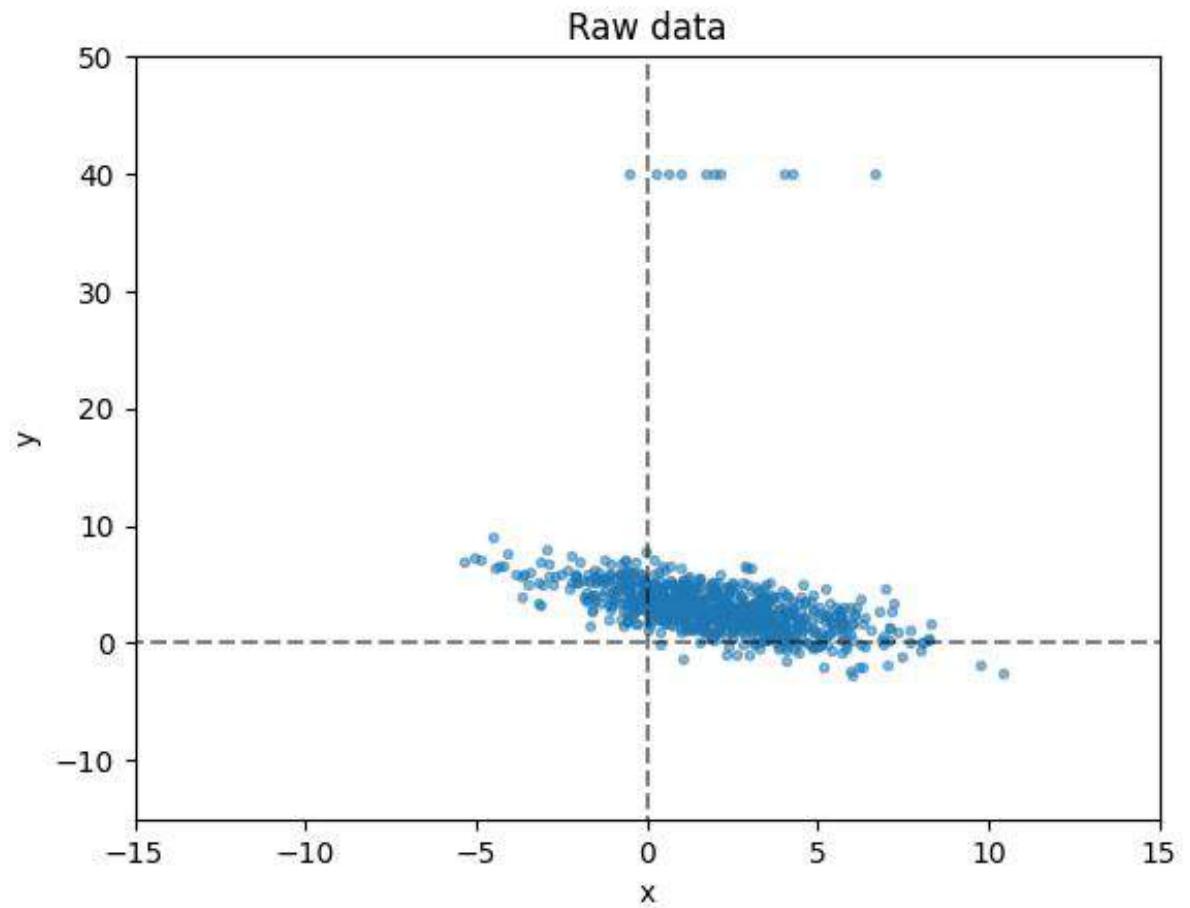
z-score normalization and PCA for dimensionality reduction to 2-D space for visualization.



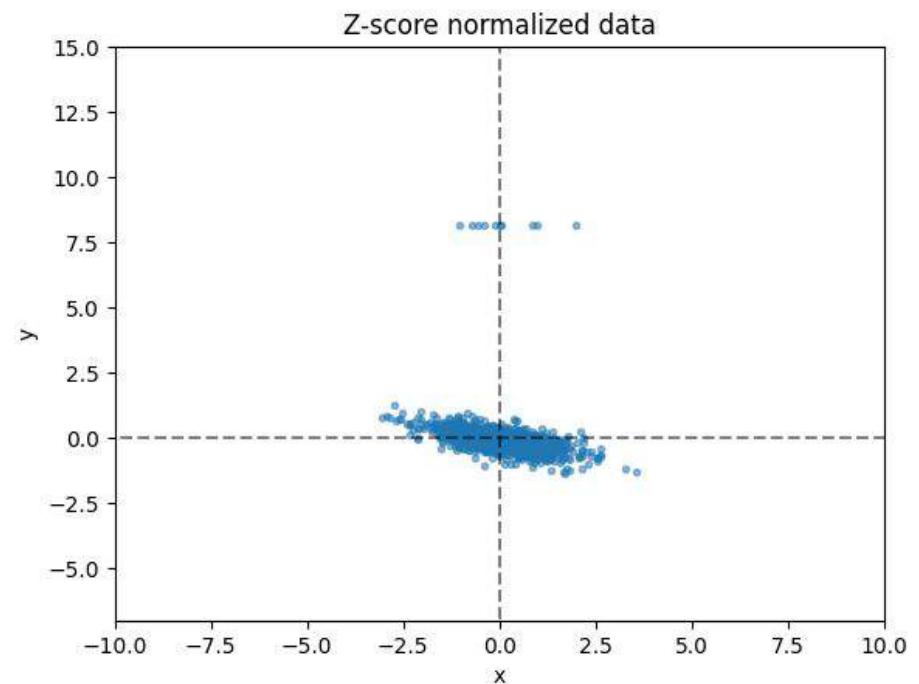
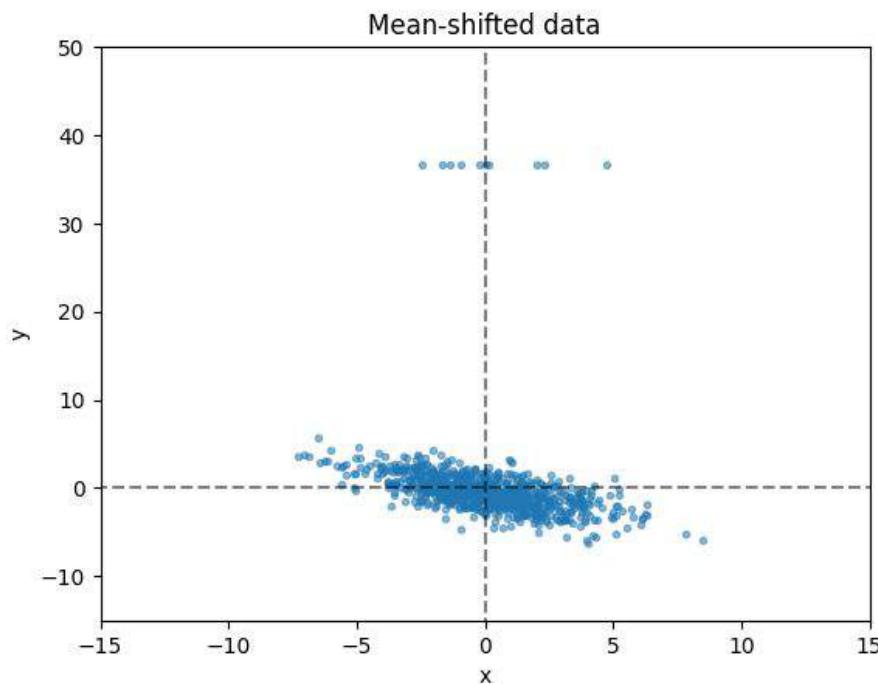
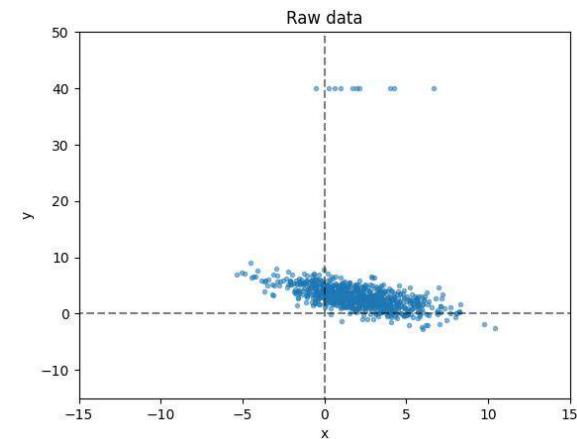
Example 3: Anomaly detection (without training data)

800-by-2 data matrix

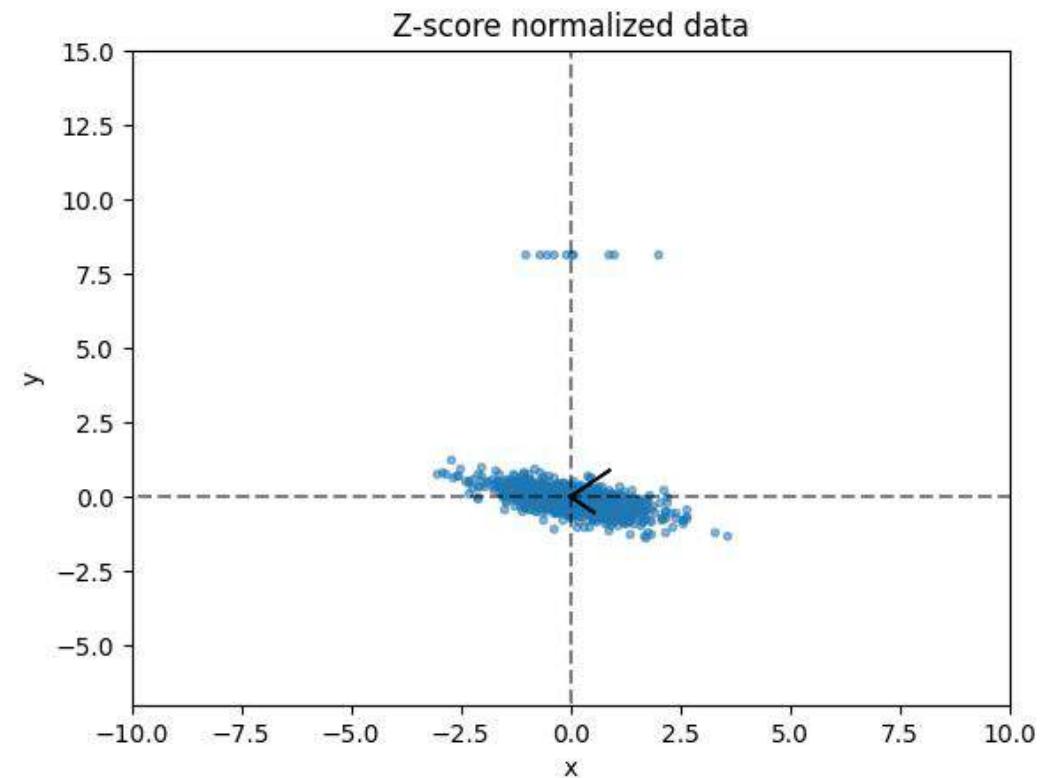
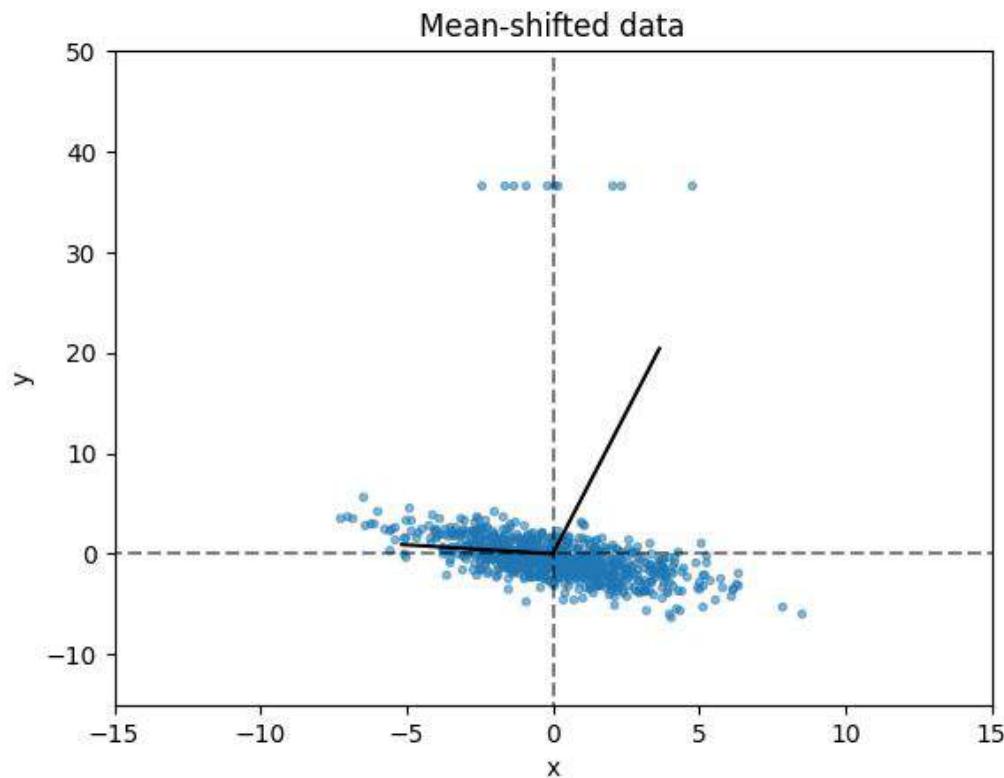
10 "anomalous" points



Example 3: Anomaly detection

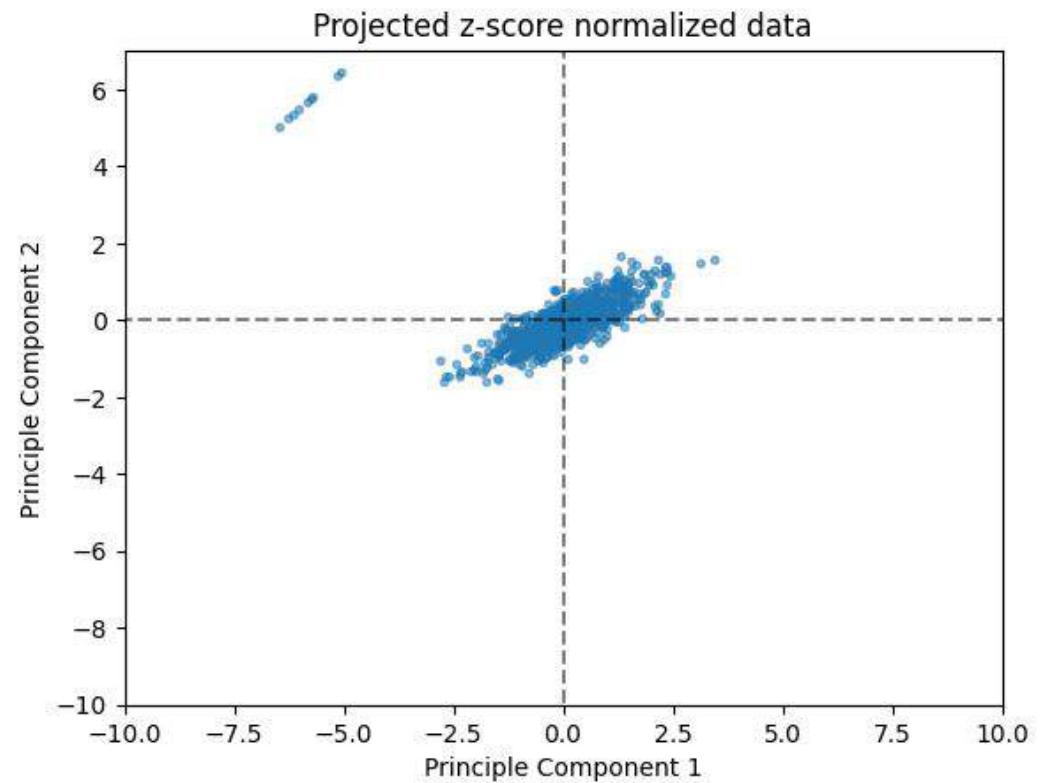
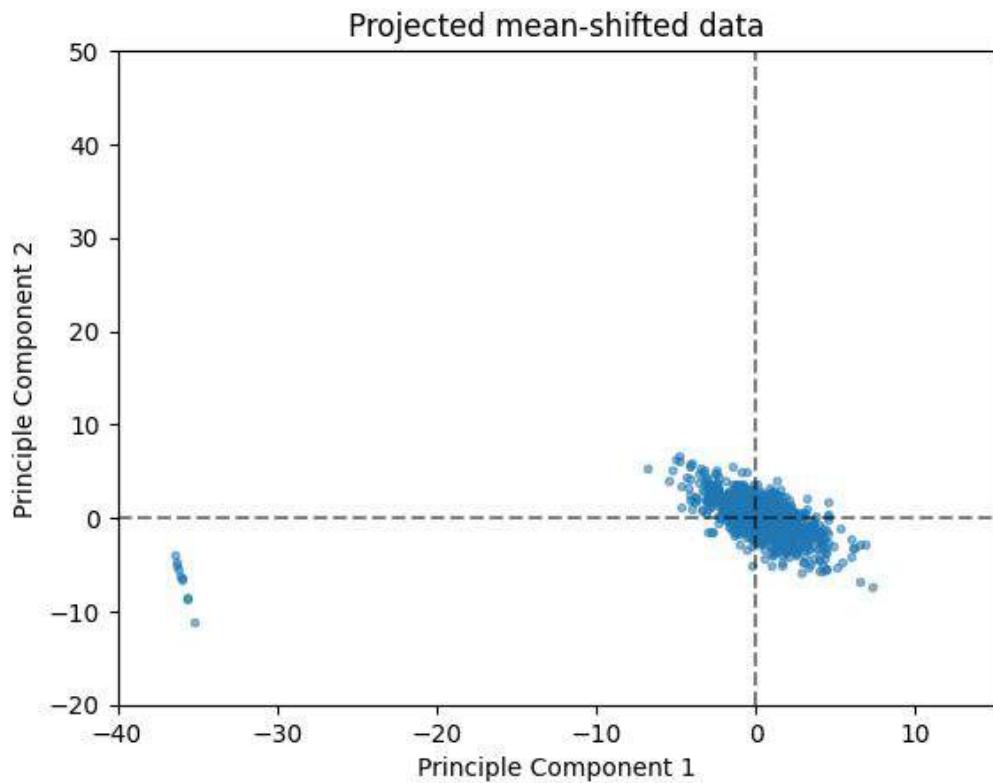


Example 3: Anomaly detection



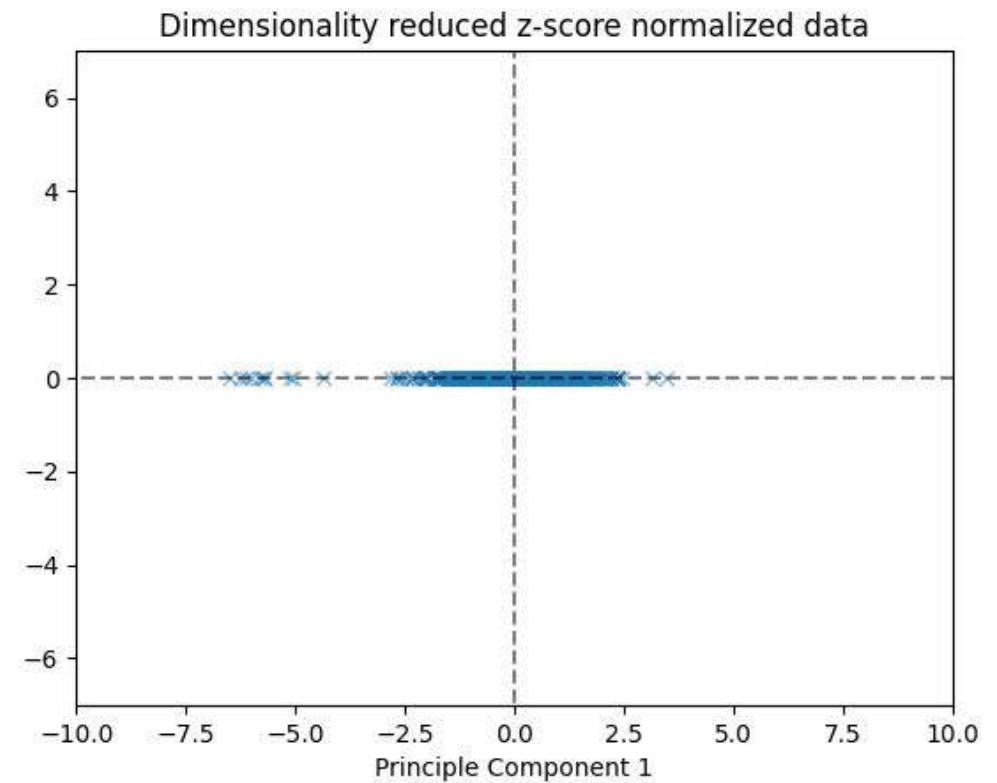
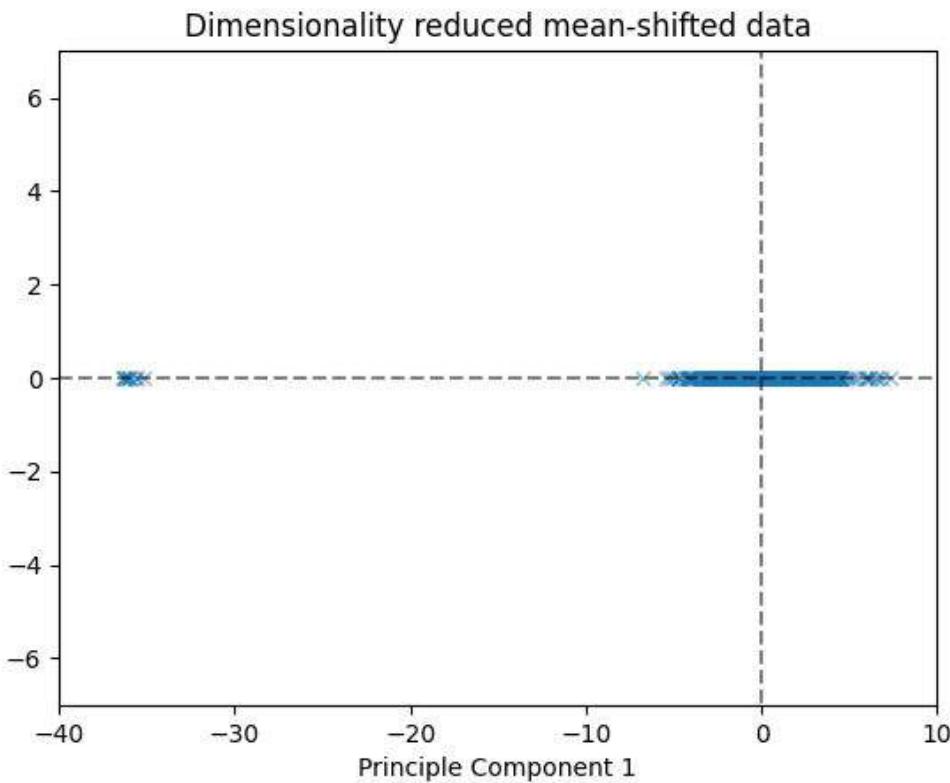
Eigenvectors affected by outliers! (Not aligned with direction of largest variance of the 'cloud'.)

Example 3: Anomaly detection



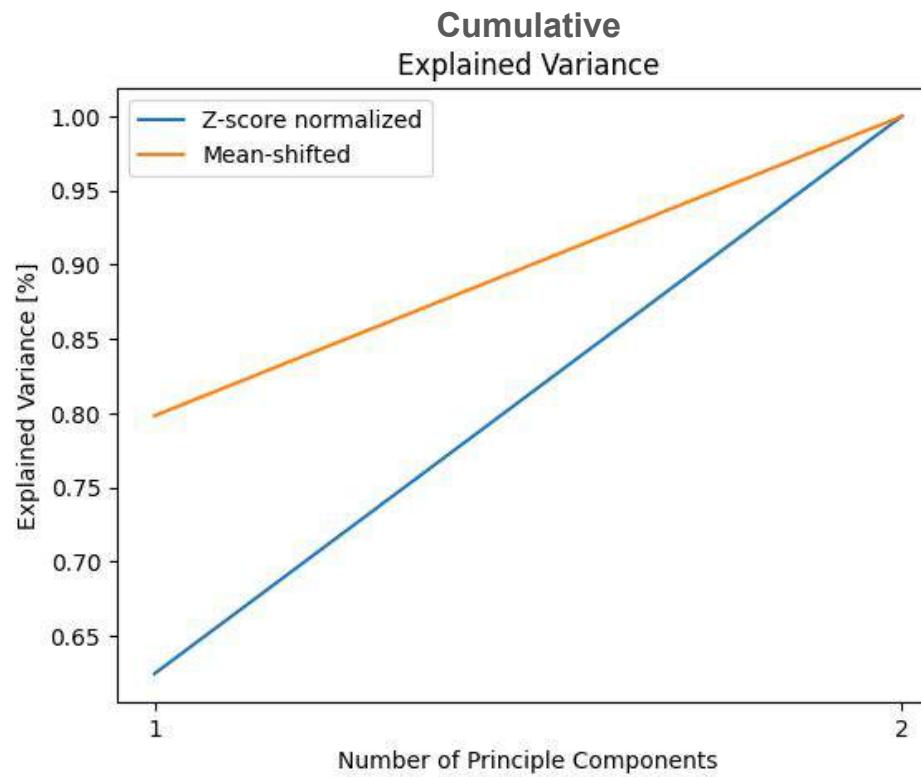
Projections on to the component space, also not aligned with the variance of the 'cloud'.

Example 3: Anomaly detection



Mean-shift normalization preserves the anomalies better.

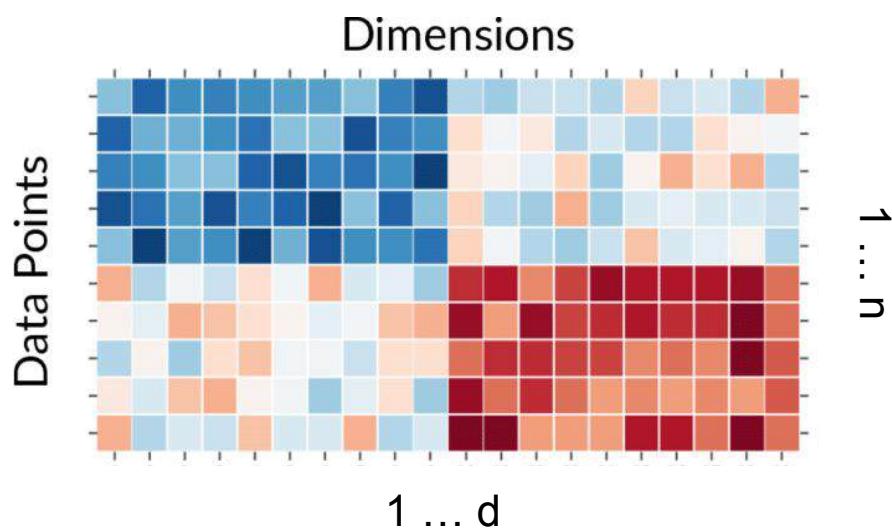
Example 3: Anomaly detection



Mean-shift normalization preserves more of the variance in the first PC.

PCA - Algorithm

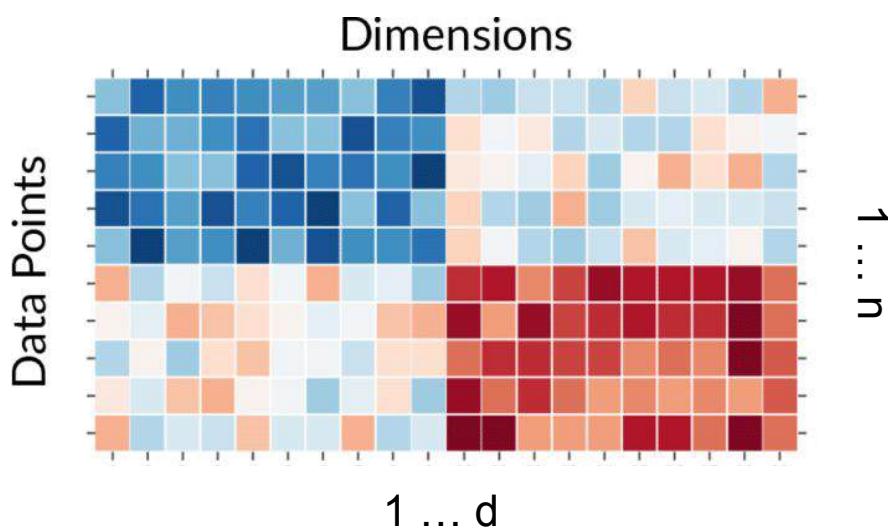
Data matrix \mathbf{X} , with n data points,
each with d features



- 1) Normalize the data
- 2) Compute the d -by- d covariance matrix (with elements $\text{cov}(\mathbf{x}_i, \mathbf{x}_j)$ for each pair of columns of the data matrix \mathbf{X}).
- 3) Compute the eigenvectors of the covariance matrix (principal components), and the corresponding eigenvalues.
- 4) Order the eigenvectors by the size of their eigenvalues, in descending order, to construct a transformation matrix \mathbf{K} .
- 5) Use the transformation matrix \mathbf{K} to project the data points to component space, i.e. matrix multiplication $\mathbf{X}\mathbf{K}$.
- 5.5) For dimensionality reduction, we can choose only the first $m < d$ eigenvectors to construct \mathbf{K} .
- 6) Compute the explained variance for each principle component.
- 7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

PCA - Algorithm

Data matrix \mathbf{X} , with n data points,
each with d features



7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

In Step 5, to project data \mathbf{X} onto component space, we used matrix multiplication $\mathbf{X}\mathbf{K} = \mathbf{Y}$.

\mathbf{K} is the matrix of eigenvectors of the covariance matrix.

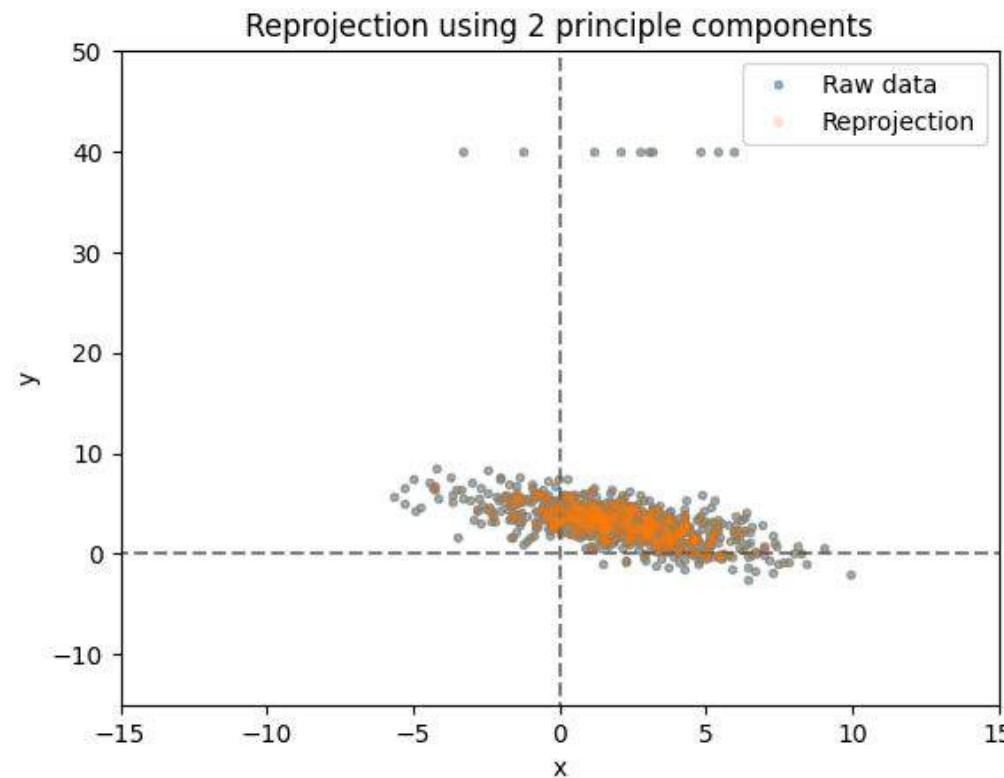
Remember: Covariance matrix is symmetric, \mathbf{K} is orthonormal! Its inverse is its transpose.

So to reproject \mathbf{Y} back into raw data space:

$$\mathbf{X}\mathbf{K}\mathbf{K}^{-1} = \mathbf{Y}\mathbf{K}^{-1} \rightarrow \mathbf{X}' = \mathbf{Y}\mathbf{K}^{-1} = \mathbf{Y}\mathbf{K}^T$$

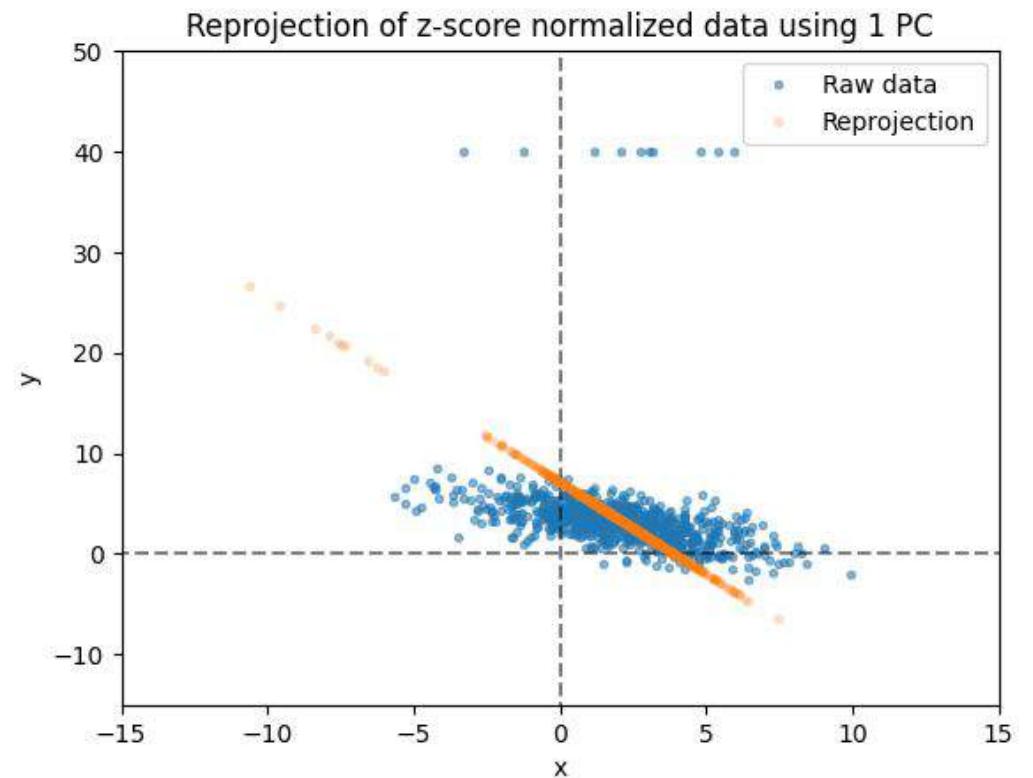
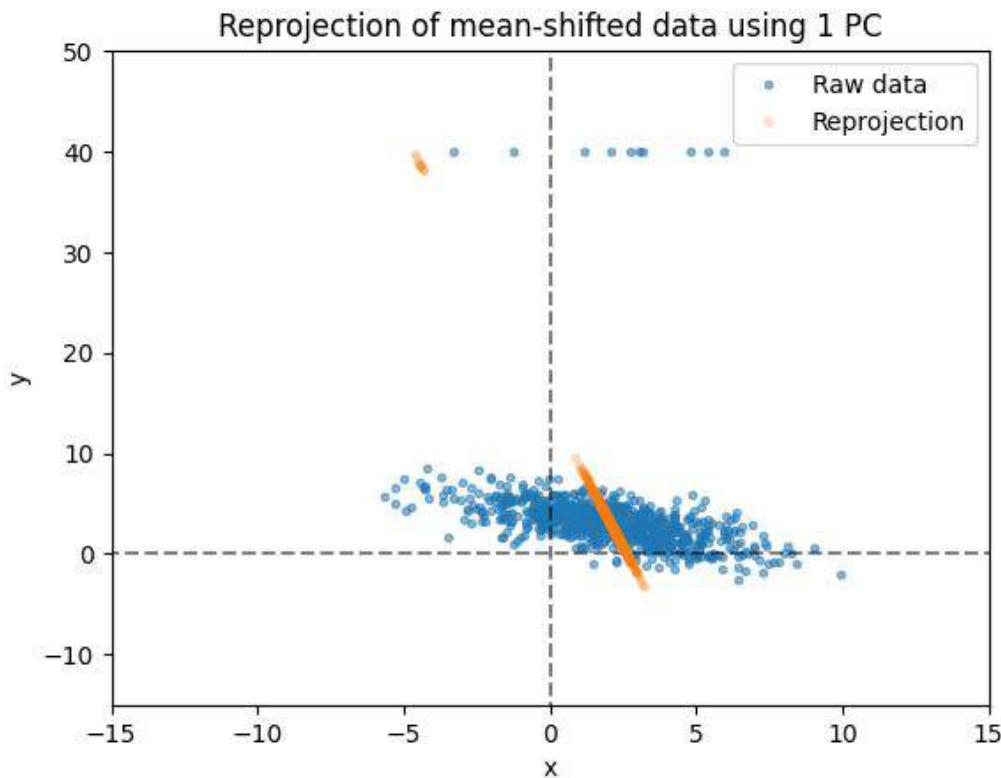
Note: Don't forget to also 'undo' the normalization!

Example 3: Anomaly detection



2 PCs capture all the variance for 2-dimensional data.

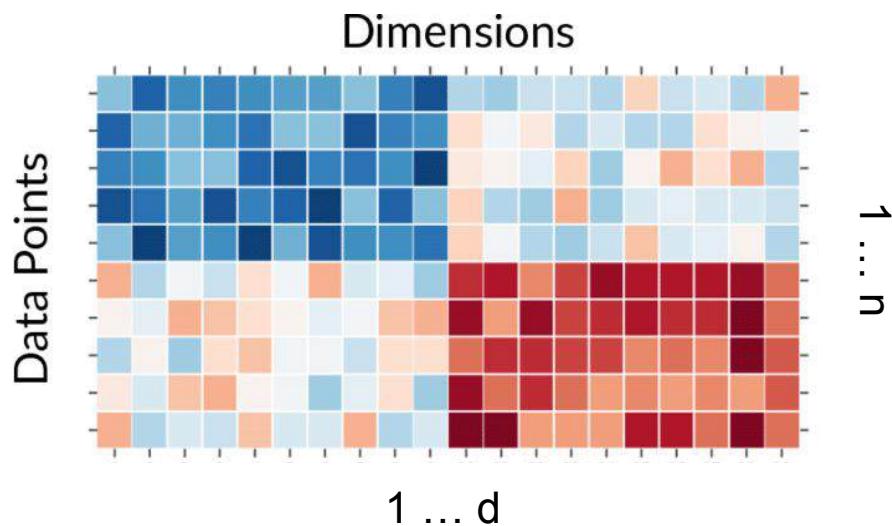
Example 3: Anomaly detection



Mean-shifted data preserves variance better in the first PC, however, also preserves the anomalies in the re-projection.

PCA - Algorithm

Data matrix \mathbf{X} , with n data points,
each with d features



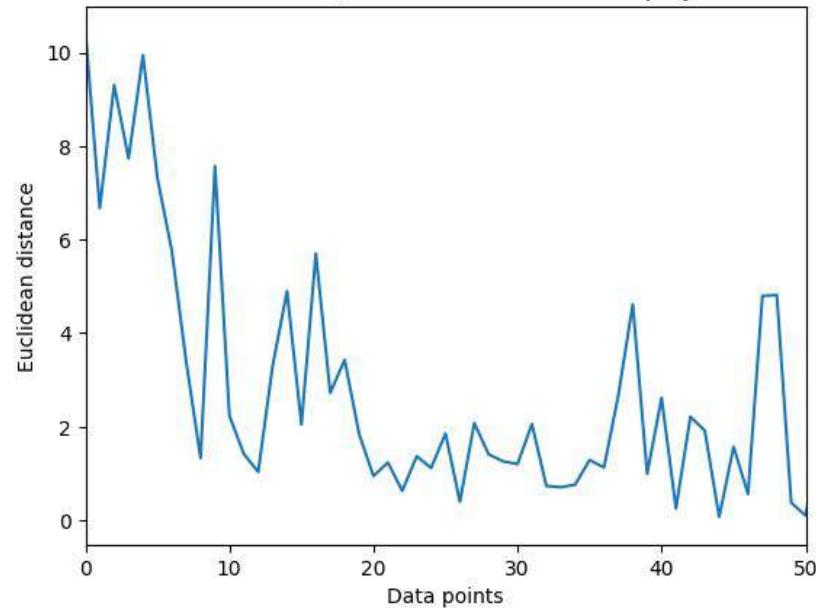
7) Reproject the data from component space back to raw data space. Compute pointwise distance to raw data (reprojection error).

Reprojection error (a.k.a. residuals):

'Reprojection error' or 'residuals' are computed as the average pairwise distance (here I used Euclidean) between the raw data and the reprojected points.

Example 3: Anomaly detection

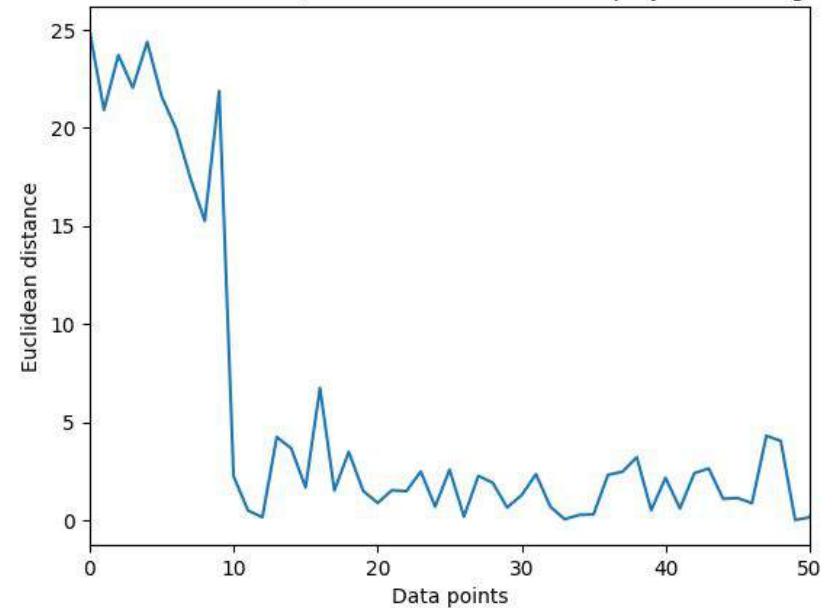
Residuals mean-shifted data (between raw data and reprojection using 1 PC)



Average residual anomalies: 6.95

Average residual the rest of the data points: 1.74

Residuals z-score data (between raw data and reprojection using 1 PC)



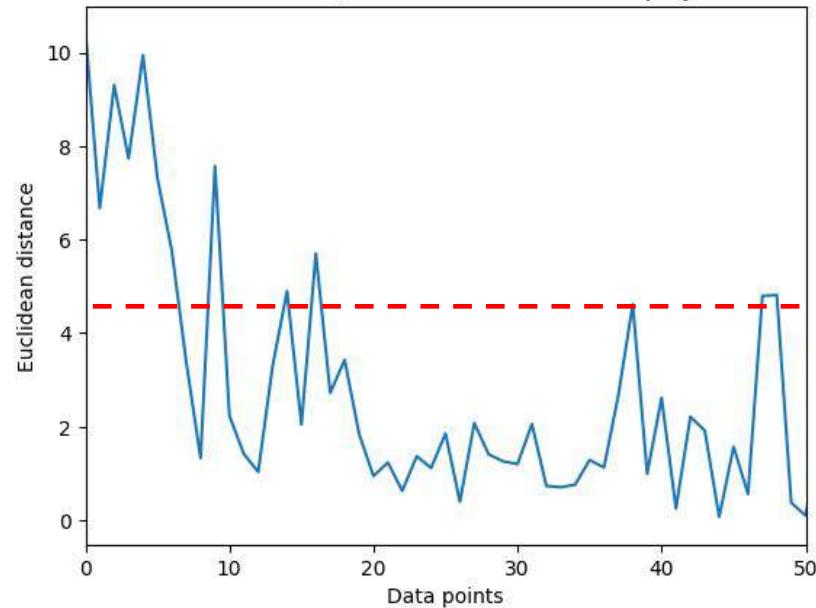
Average residual anomalies: 21.21

Average residual the rest of the data points: 1.60

First 10 data points are the anomalies.

Example 3: Anomaly detection

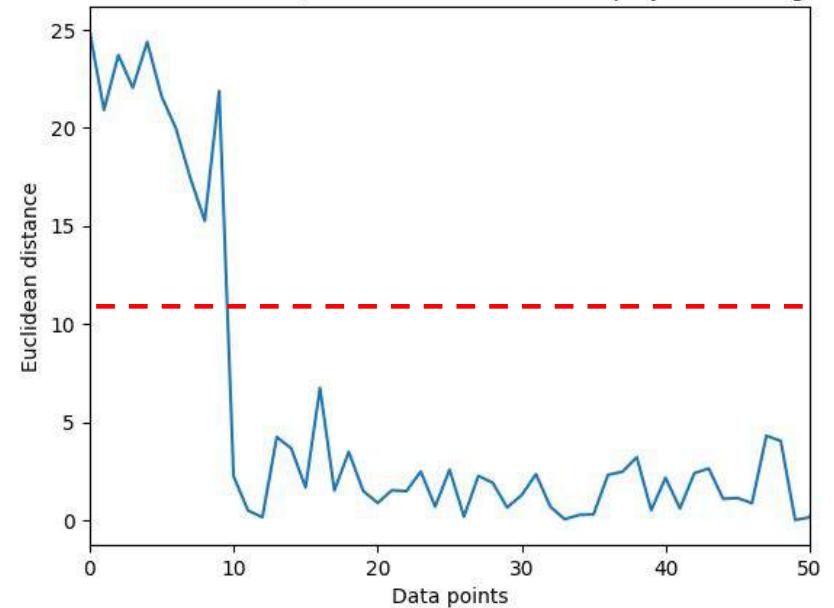
Residuals mean-shifted data (between raw data and reprojection using 1 PC)



Average residual anomalies: 6.95

Average residual the rest of the data points: 1.74

Residuals z-score data (between raw data and reprojection using 1 PC)



Average residual anomalies: 21.21

Average residual the rest of the data points: 1.60

First 10 data points are the anomalies.

Example 4: Anomaly detection (with training data)

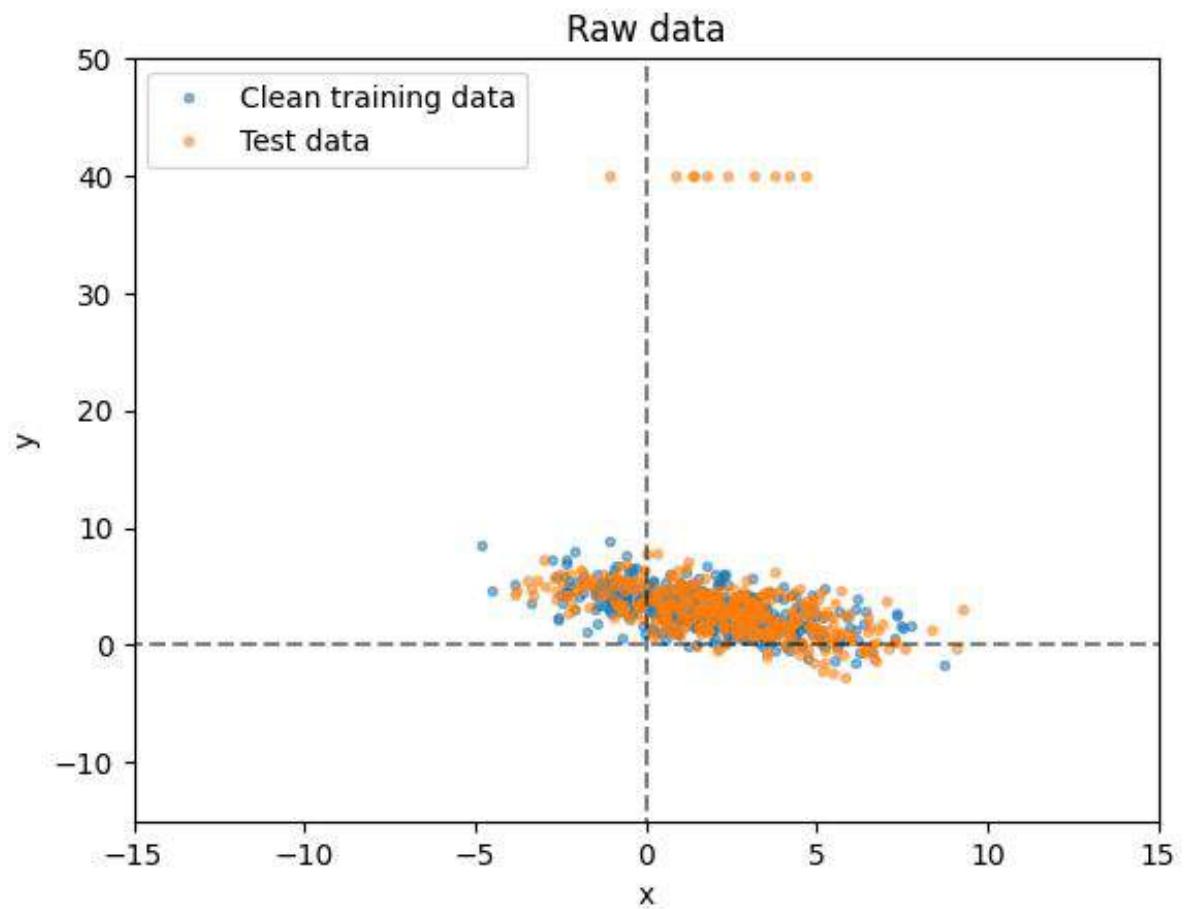
'Clean' training data:

400-by-2 data matrix

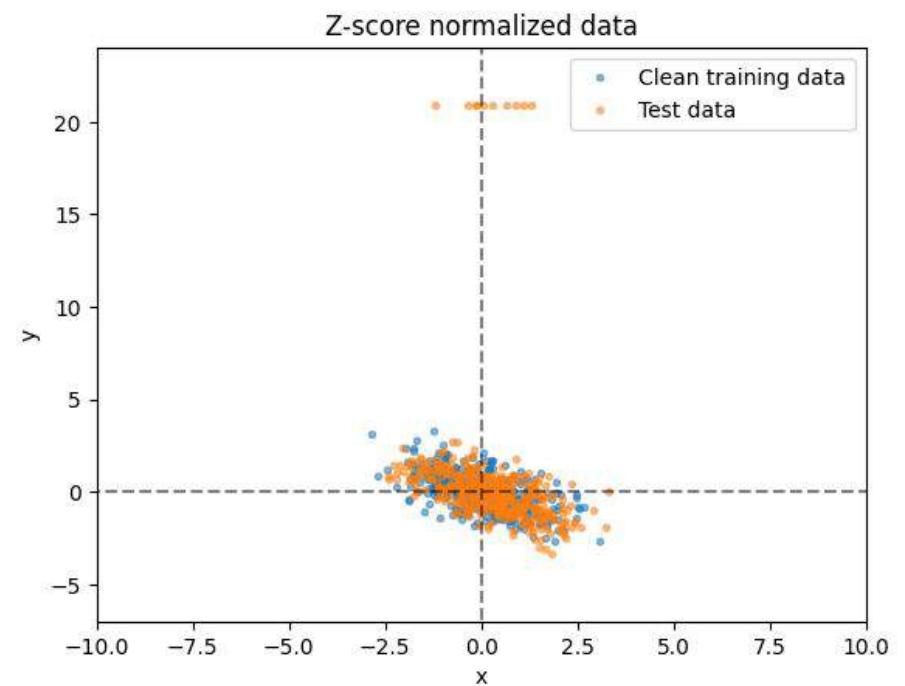
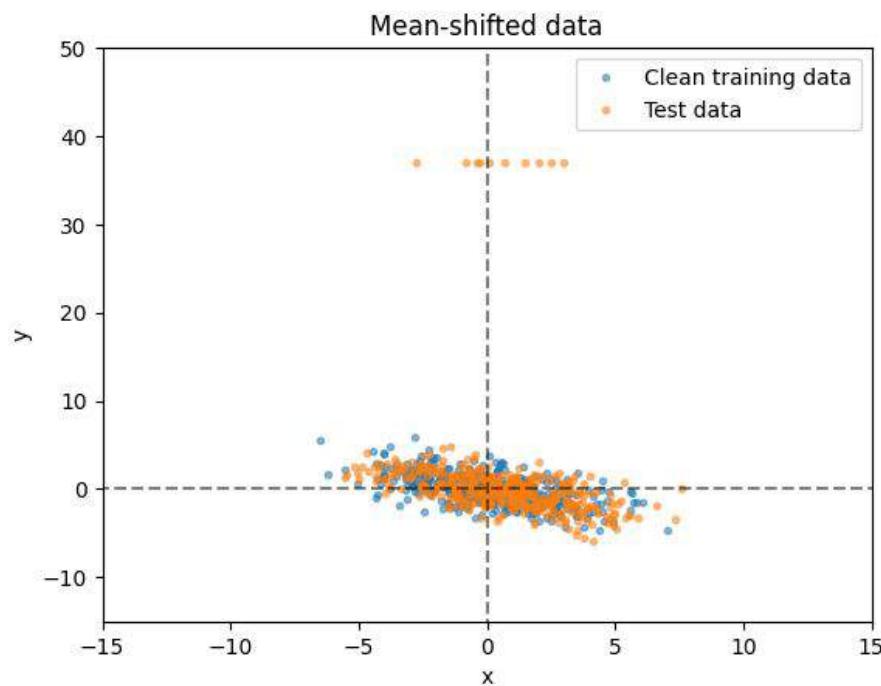
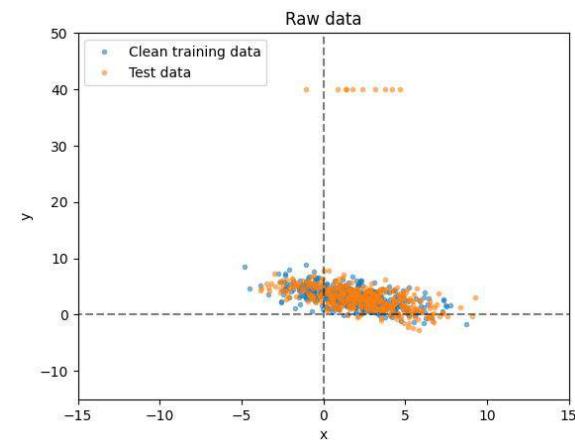
Unknown test time data:

400 samples

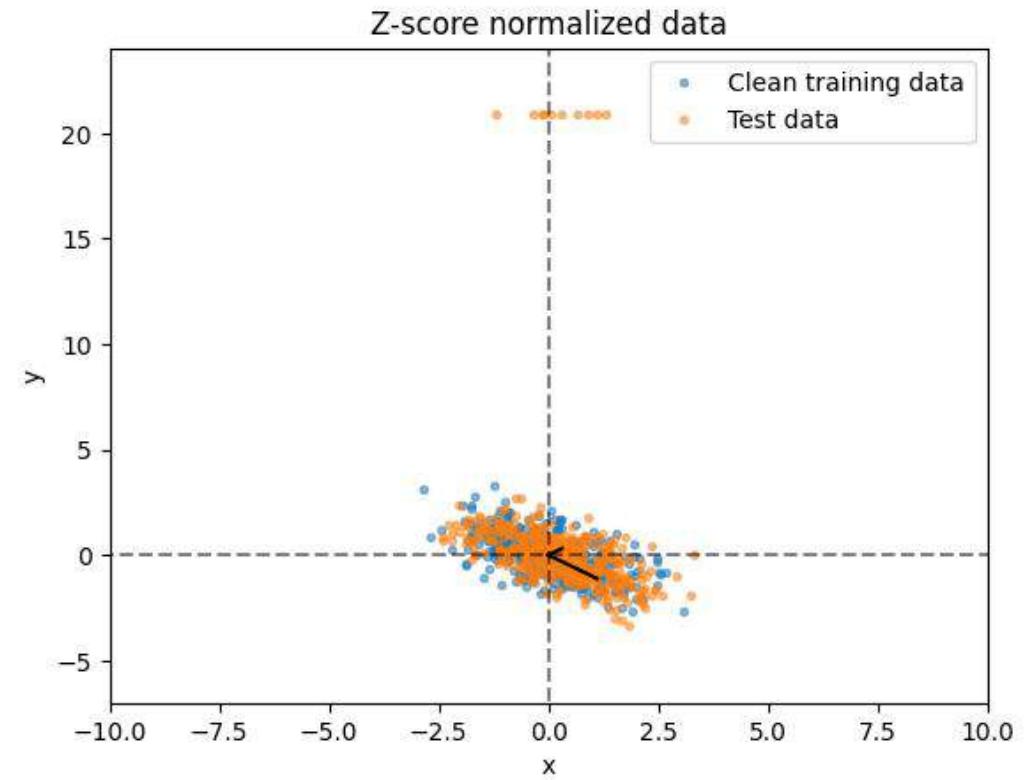
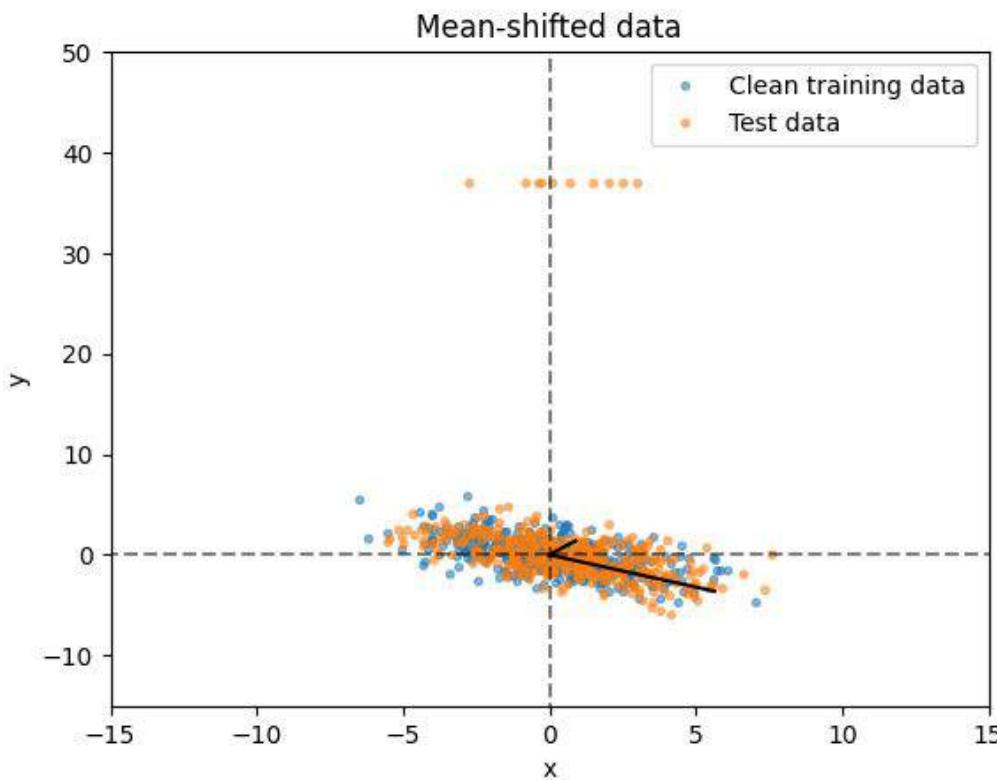
with 10 "anomalous" samples



Example 4: Anomaly detection (with training data)

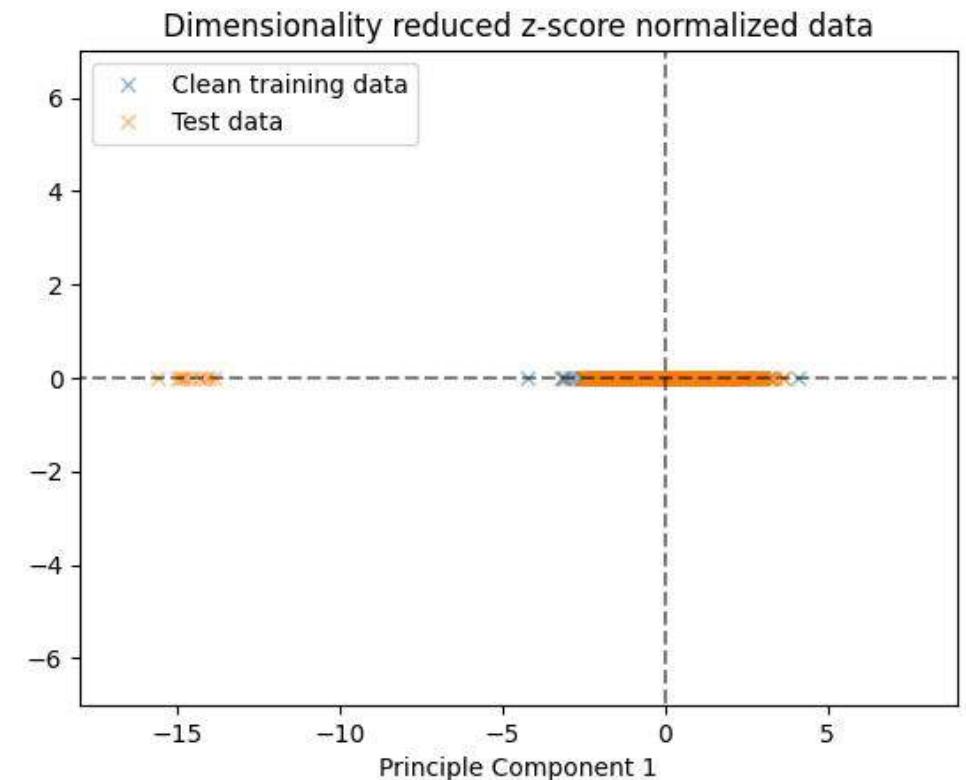
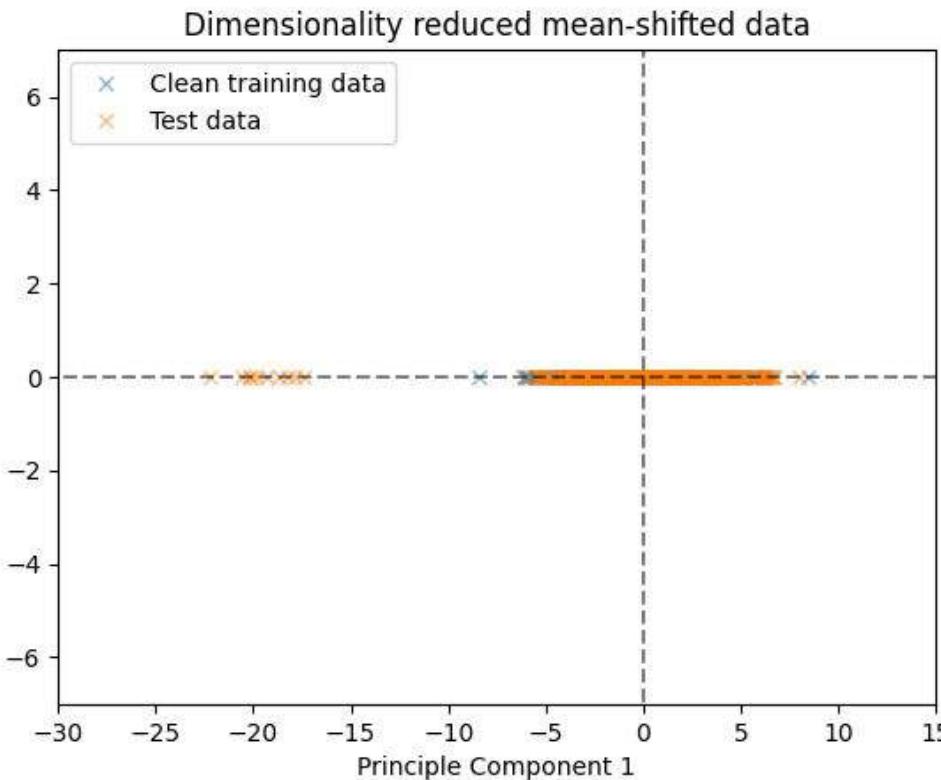


Example 4: Anomaly detection (with training data)



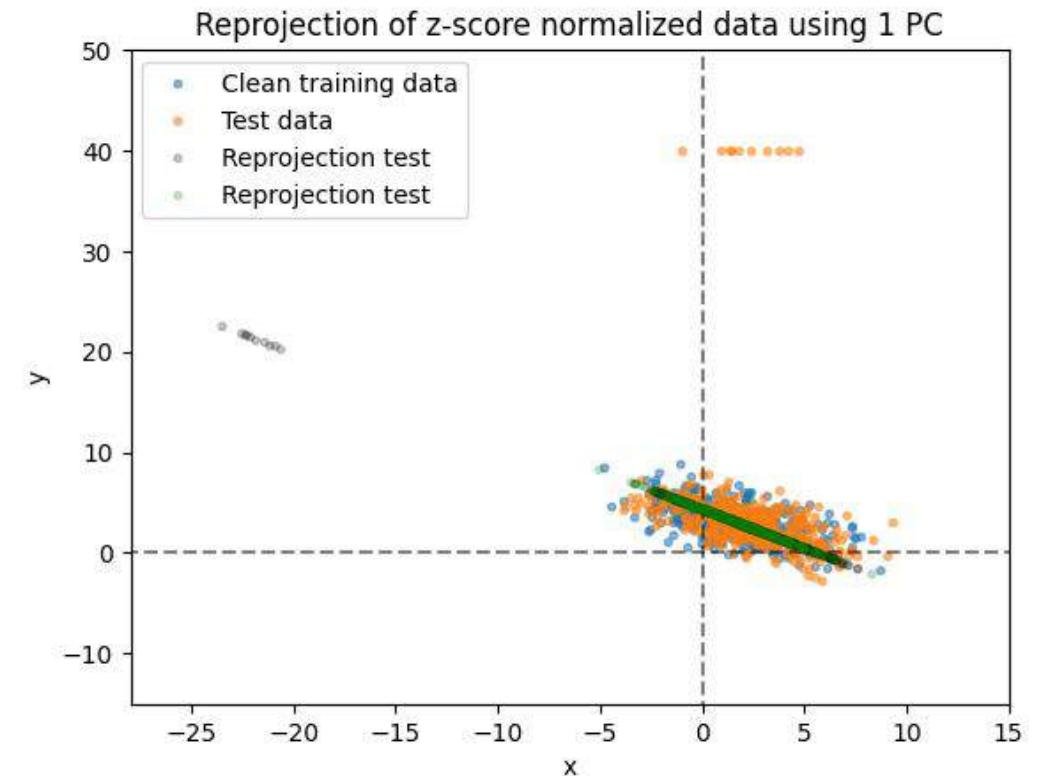
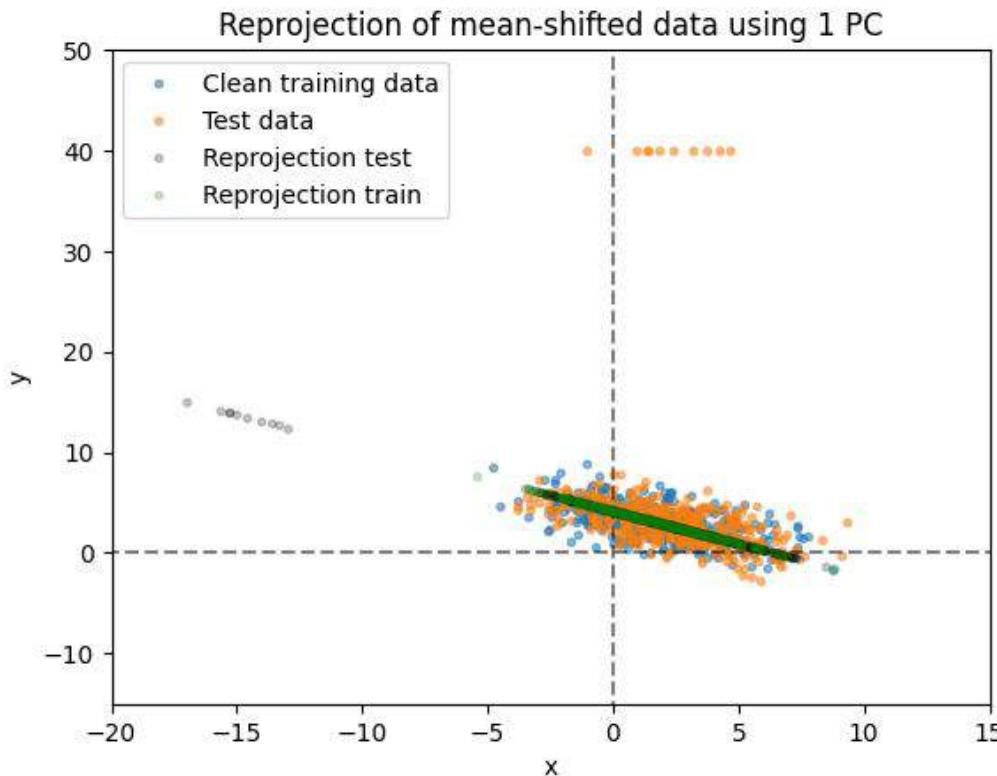
Eigenvectors computed from training data only: Projection of data not affected by outliers.

Example 4: Anomaly detection (with training data)



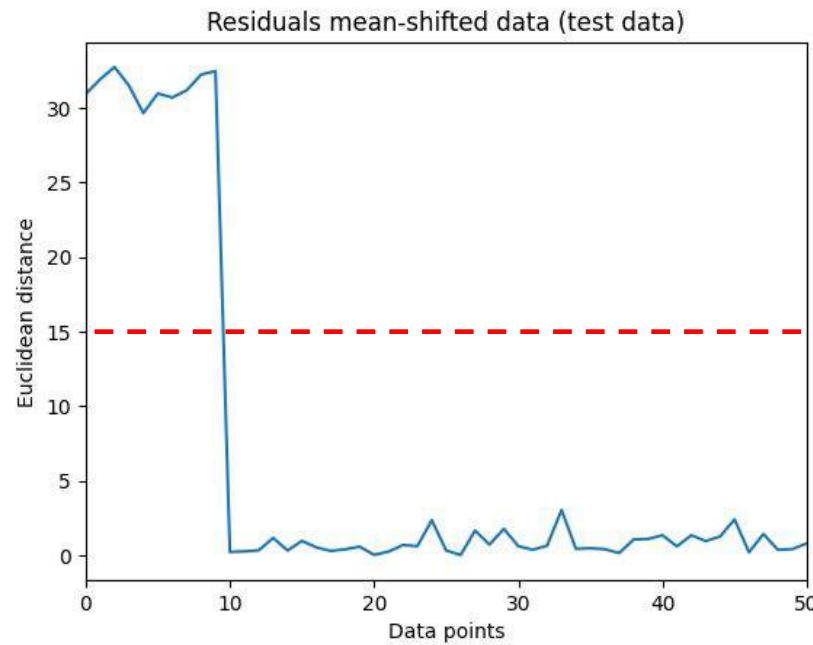
Both normalizations preserve the anomalies.

Example 4: Anomaly detection (with training data)

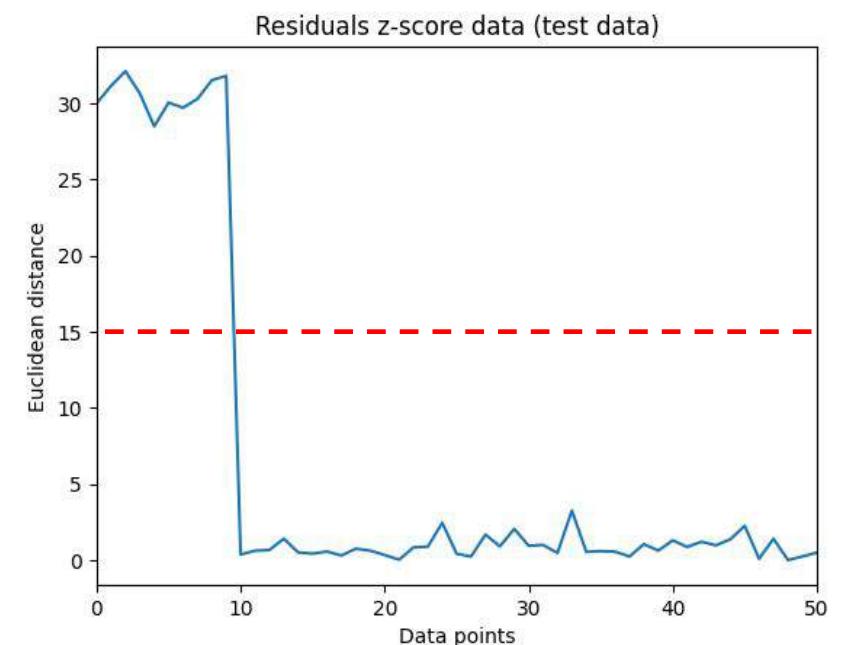


For both normalizations, anomalies have much larger reprojection error.

Example 4: Anomaly detection (with training data)



Average residual anomalies: 31.41
Average residual the rest of the data points: 1.01

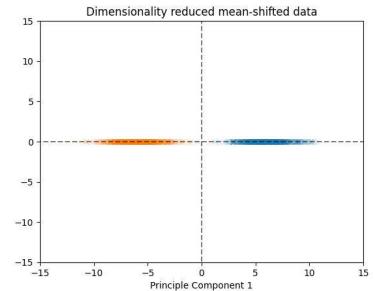


Average residual anomalies: 30.55
Average residual the rest of the data points: 1.07

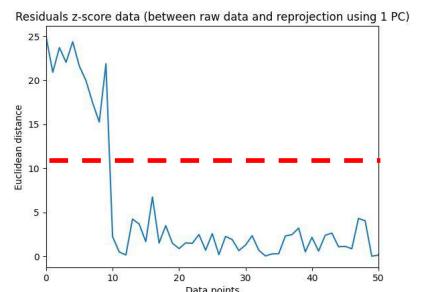
When training data is available, there is a more distinct separation of anomalies.

Summary and outlook

- PCA is computed using the eigenvectors of the covariance matrix.
- PCA can be helpful for dimensionality reduction and anomaly detection.

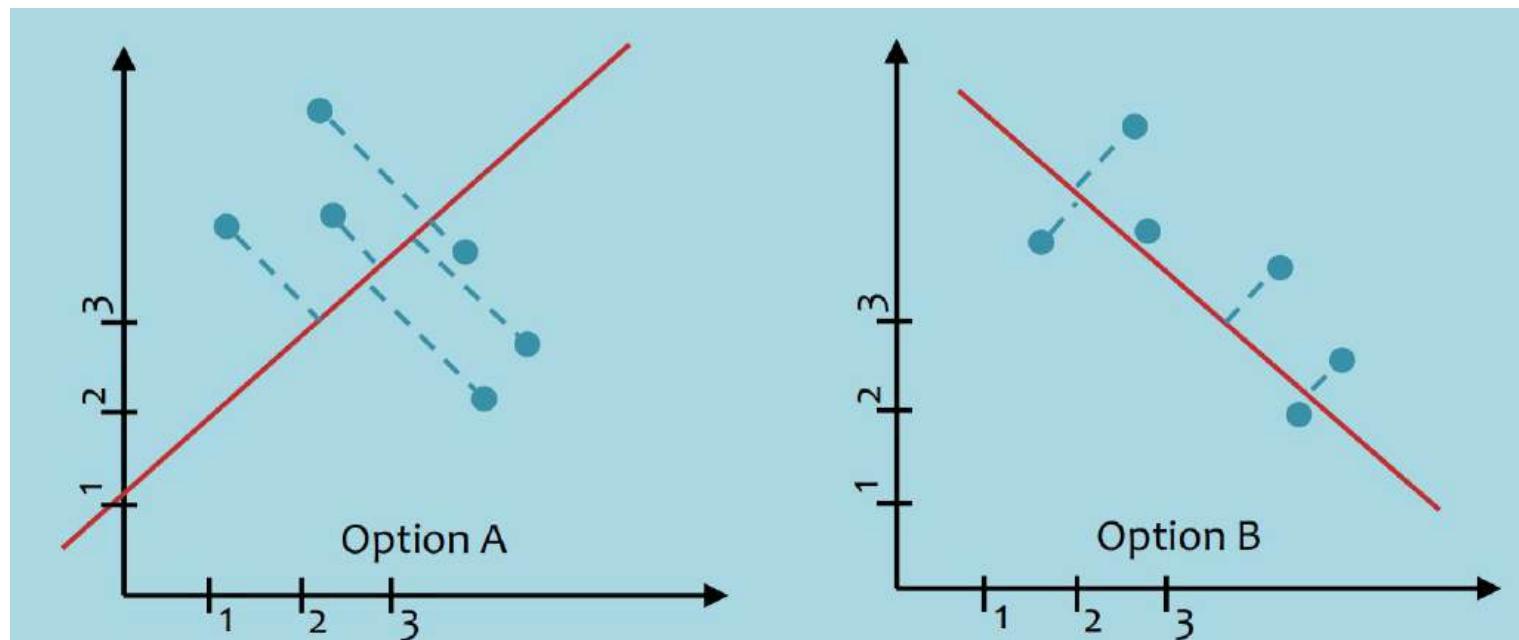


- **Normalization can be crucial** for a given dataset and task!
 - e.g. A good method for clustering might be bad for anomaly detection.
- **Outlook:** PCA is used for the first lab assignment. You can compute it following the above steps, but make sure the functions you use are within the allowed libraries.



Project all samples (2D) onto a line (1D)

Equivalence of **Maximizing** Variance and **Minimizing** Reprojection Error



PCA

Equivalence of
Maximizing Variance
and **Minimizing**
Reprojection Error

Proof: First, note that:

*Derivation not
part of the exam

$$\text{Reprojection error} \rightarrow \| \mathbf{x}^{(i)} - (\mathbf{v}^T \mathbf{x}^{(i)}) \mathbf{v} \|^2 = \| \mathbf{x}^{(i)} \|^2 - (\mathbf{v}^T \mathbf{x}^{(i)})^2 \quad (1)$$

$$\text{since } \mathbf{v}^T \mathbf{v} = \| \mathbf{v} \|^2 = 1.$$

Substituting into the minimization problem, and removing the extraneous terms, we obtain the maximization problem.

$$\mathbf{v}^* = \underset{\mathbf{v}: \| \mathbf{v} \|^2 = 1}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \| \mathbf{x}^{(i)} - (\mathbf{v}^T \mathbf{x}^{(i)}) \mathbf{v} \|^2 \quad (2)$$

$$= \underset{\mathbf{v}: \| \mathbf{v} \|^2 = 1}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \| \mathbf{x}^{(i)} \|^2 - (\mathbf{v}^T \mathbf{x}^{(i)})^2 \quad (3)$$

$$= \underset{\mathbf{v}: \| \mathbf{v} \|^2 = 1}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^N (\mathbf{v}^T \mathbf{x}^{(i)})^2 \quad (4)$$

Resources

1. Data Mining the Textbook: Chapters 1.3, 1.4 and 2.4.3

2. Recap Linear Algebra:

https://www.youtube.com/watch?v=fNk_zzaMoSs&list=RDCMUCYO_jab_esuFRV4b17AJtAw

Extra information - Matrix definitions

Recap: Matrix Properties

Addition

$$\begin{bmatrix} 1 & 3 & 1 \\ 1 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 5 \\ 7 & 5 & 0 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 & 1+5 \\ 1+7 & 0+5 & 0+0 \end{bmatrix}$$

Scalar multiplication

$$2 \cdot \begin{bmatrix} 1 & 8 & -3 \\ 4 & -2 & 5 \end{bmatrix} = \begin{bmatrix} 2 \cdot 1 & 2 \cdot 8 & 2 \cdot -3 \\ 2 \cdot 4 & 2 \cdot -2 & 2 \cdot 5 \end{bmatrix}$$

Transposition

$$(\mathbf{A}^T)_{i,j} = \mathbf{A}_{j,i}$$

For example:

$$\begin{bmatrix} 1 & 2 & 3 \\ 0 & -6 & 7 \end{bmatrix}^T = \begin{bmatrix} 1 & 0 \\ 2 & -6 \\ 3 & 7 \end{bmatrix}$$

Recap: Matrix Properties

Matrix multiplication

If \mathbf{A} is an m -by- n matrix and \mathbf{B} is an n -by- p matrix, then their *matrix product* \mathbf{AB} is the m -by- p matrix, with $1 \leq i \leq m$ and $1 \leq j \leq p$:

$$[\mathbf{AB}]_{i,j} = a_{i,1}b_{1,j} + a_{i,2}b_{2,j} + \cdots + a_{i,n}b_{n,j} = \sum_{r=1}^n a_{i,r}b_{r,j}$$

For example:

$$\begin{bmatrix} 2 & 3 & 4 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \frac{1000}{100} \\ 1 & \frac{100}{10} \\ 0 & \underline{\frac{10}{10}} \end{bmatrix} = \begin{bmatrix} 3 & \frac{2340}{1000} \\ 0 & 1000 \end{bmatrix}$$

Non-commutative:

$$\mathbf{AB} \neq \mathbf{BA}$$

Associative:

$$(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$$

Distributive:

$$(\mathbf{A} + \mathbf{B})\mathbf{C} = \mathbf{AC} + \mathbf{BC}$$

Recap: Inverse matrices

- Multiplication of a matrix \mathbf{A} with its inverse \mathbf{A}^{-1} is the identity matrix:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

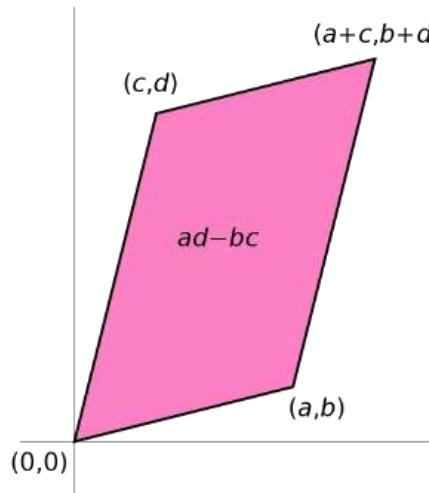
- We can solve for \mathbf{X} using the inverse:

$$\mathbf{X}\mathbf{A} = \mathbf{B} \rightarrow \mathbf{X} = \mathbf{B}\mathbf{A}^{-1}$$

- A square n -by- n matrix is invertible if and only if the determinant $\det(\mathbf{A}) \neq 0$ (i.e. the rows and columns of \mathbf{A} are linearly independent).

Recap: Determinant

- The determinant $\det(\mathbf{A})$ represents the scale factor by which areas (2D) or volumes (more than 2D) are transformed by \mathbf{A}



The determinant of a 2×2 matrix is

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc,$$

and the determinant of a 3×3 matrix is

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + bfg + cdh - ceg - bdi - afh.$$

The area of the parallelogram is the absolute value of the determinant of the matrix formed by the vectors representing the parallelogram's sides.

Image source: <https://en.wikipedia.org/wiki/Determinant>

Eigenvalues and eigenvectors

- Assume we know matrix \mathbf{M} , how can we find the eigenvalues?:

$$\mathbf{M}\mathbf{v} = \lambda\mathbf{v}$$

Eigenvalues and eigenvectors

- Assume we know matrix \mathbf{M} , how can we find the eigenvalues?:

$$\underbrace{\mathbf{M} \mathbf{v}}_{\text{matrix multiplication}} = \underbrace{\lambda \mathbf{v}}_{\text{scalar multiplication}}$$

Eigenvalues and eigenvectors

- Assume we know matrix \mathbf{M} , how can we find the eigenvalues?:

$$\underbrace{\mathbf{M}\mathbf{v}}_{\text{matrix multiplication}} = \underbrace{\lambda\mathbf{v}}_{\text{scalar multiplication}}$$

- Rewrite both sides as matrix multiplication with identity matrix I :

$$\mathbf{M}\mathbf{v} = (\lambda I)\mathbf{v}$$

Eigenvalues and eigenvectors

- Assume we know matrix \mathbf{M} , how can we find the eigenvalues?:

$$\underbrace{\mathbf{M}\mathbf{v}}_{\text{matrix multiplication}} = \underbrace{\lambda\mathbf{v}}_{\text{scalar multiplication}}$$

- Rewrite both sides as matrix multiplication with identity matrix I :

$$\mathbf{M}\mathbf{v} = (\lambda I)\mathbf{v}$$

- Rearrange, where $\mathbf{0}$ is the zero-vector:

$$\mathbf{M}\mathbf{v} - (\lambda I)\mathbf{v} = \mathbf{0} \rightarrow (\mathbf{M} - \lambda I)\mathbf{v} = \mathbf{0}$$

Eigenvalues and eigenvectors

- Assume we know matrix \mathbf{M} , how can we find the eigenvalues?:

$$\underbrace{\mathbf{M}\mathbf{v}}_{\text{matrix multiplication}} = \underbrace{\lambda\mathbf{v}}_{\text{scalar multiplication}}$$

- Rewrite both sides as matrix multiplication with identity matrix I :

$$\mathbf{M}\mathbf{v} = (\lambda I)\mathbf{v}$$

- Rearrange, where $\mathbf{0}$ is the zero-vector:

$$\mathbf{M}\mathbf{v} - (\lambda I)\mathbf{v} = \mathbf{0} \rightarrow (\mathbf{M} - \lambda I)\mathbf{v} = \mathbf{0}$$

↓

$$\begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix} \mathbf{v} = \mathbf{0}$$

Eigenvalues and eigenvectors

- Property: Assuming v is not the zero-vector,

$$\begin{bmatrix} a-\lambda & b \\ c & d-\lambda \end{bmatrix} v = 0$$

is true if and only if $\det(M - \lambda I) = 0$.

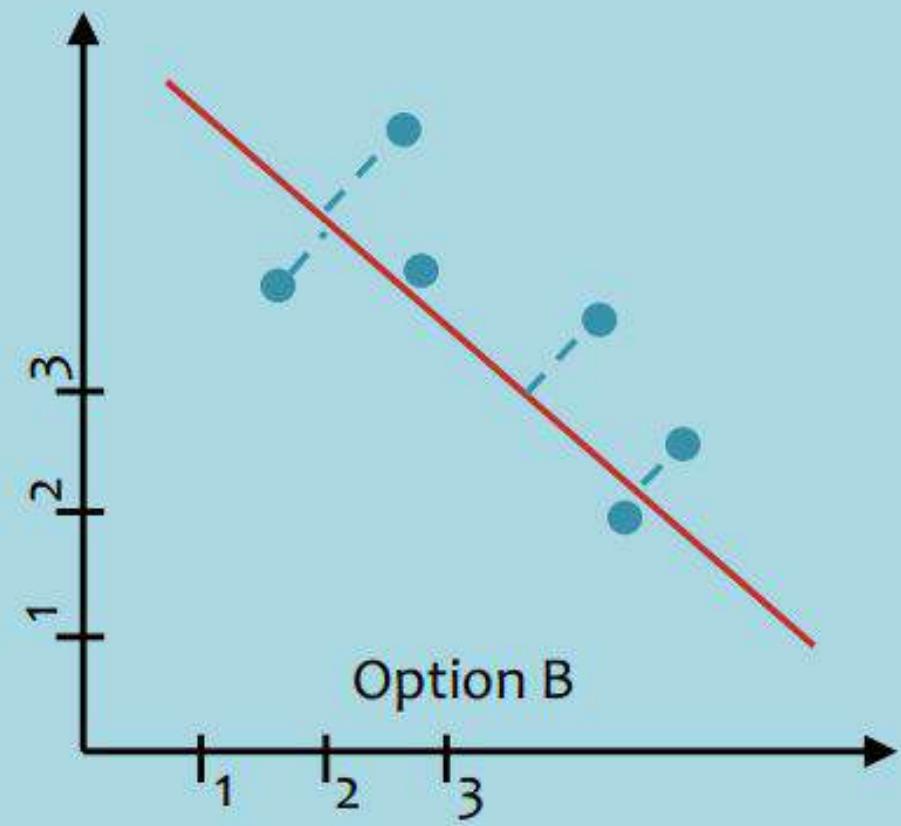
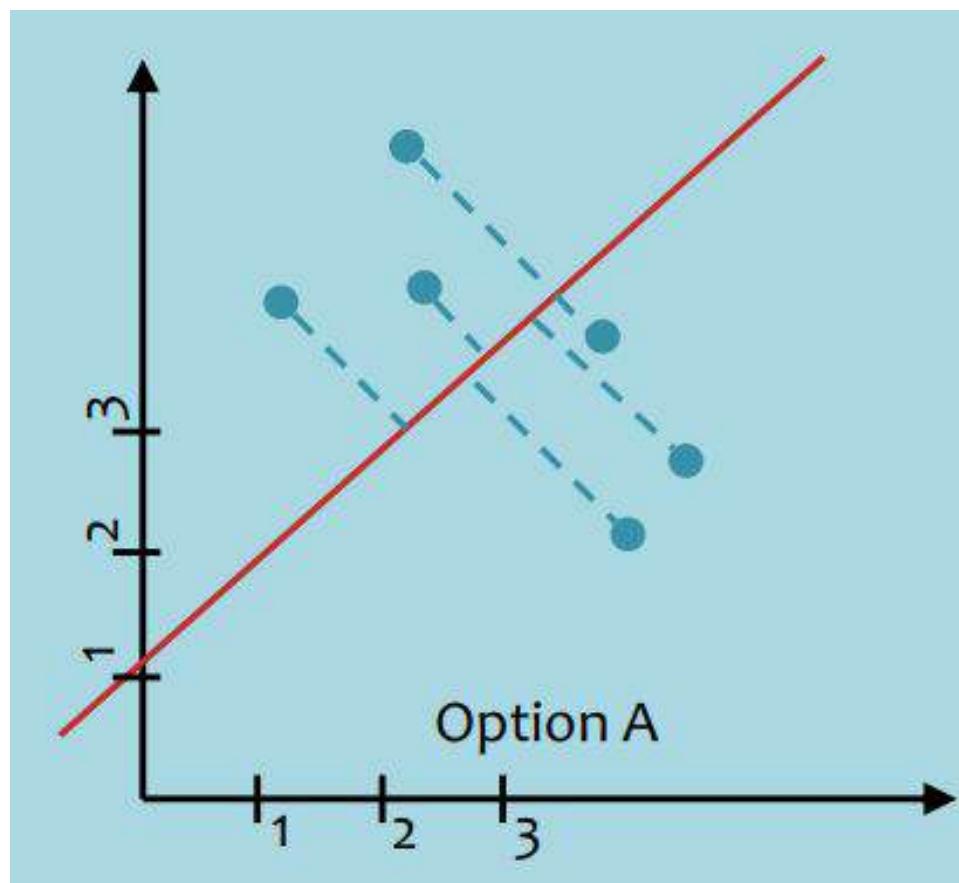
- Using the determinant → Solve for the eigenvalues
- Plug in the eigenvalues, e.g. if $\lambda_1 = 1$, to solve for the eigenvector:

$$\begin{bmatrix} a-1 & b \\ c & d-1 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Eigenvalues and eigenvectors

or

```
eigenvalues, eigenvectors = np.linalg.eig(my_matrix)
```



Equivalence of Maximizing Variance and Minimizing Reconstruction Error

PCA

Claim: Minimizing the reconstruction error is equivalent to maximizing the variance.

Proof: First, note that:

$$\|\mathbf{x}^{(i)} - (\mathbf{v}^T \mathbf{x}^{(i)})\mathbf{v}\|^2 = \|\mathbf{x}^{(i)}\|^2 - (\mathbf{v}^T \mathbf{x}^{(i)})^2 \quad (1)$$

since $\mathbf{v}^T \mathbf{v} = \|\mathbf{v}\|^2 = 1$.

Substituting into the minimization problem, and removing the extraneous terms, we obtain the maximization problem.

$$\mathbf{v}^* = \underset{\mathbf{v}: \|\mathbf{v}\|^2=1}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - (\mathbf{v}^T \mathbf{x}^{(i)})\mathbf{v}\|^2 \quad (2)$$

$$= \underset{\mathbf{v}: \|\mathbf{v}\|^2=1}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)}\|^2 - (\mathbf{v}^T \mathbf{x}^{(i)})^2 \quad (3)$$

$$= \underset{\mathbf{v}: \|\mathbf{v}\|^2=1}{\operatorname{argmax}} \frac{1}{N} \sum_{i=1}^N (\mathbf{v}^T \mathbf{x}^{(i)})^2 \quad (4)$$

Lab 1 – Lab 2

- Tomorrow final day lab 1
- Deadline is next week Thursday
- Kaggle is open until next week Friday
- Peer opens tomorrow, make sure you registered your group in Brightspace! And both groupmembers register on peer!
- Lab 2 starts next week Friday
- Peer review for Lab 1 the week after on Monday

Speeding up Algorithms in data mining

DTW can be slow

- How to fix it?

DTW can be slow

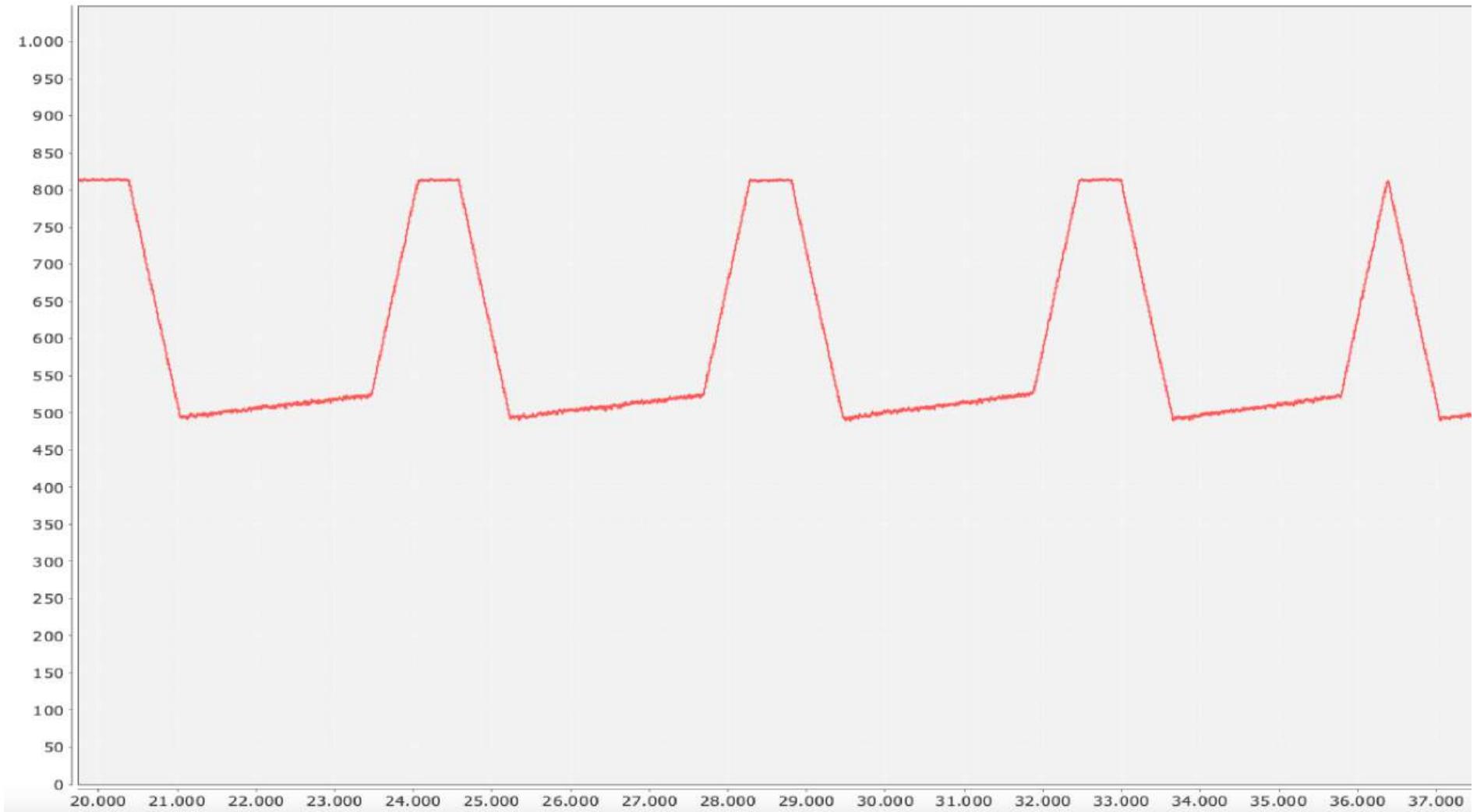
- How to fix it?
 - Do we need an exact solution?
 - Is all data really necessary?
 - Make it as simple as possible

Fast DTW

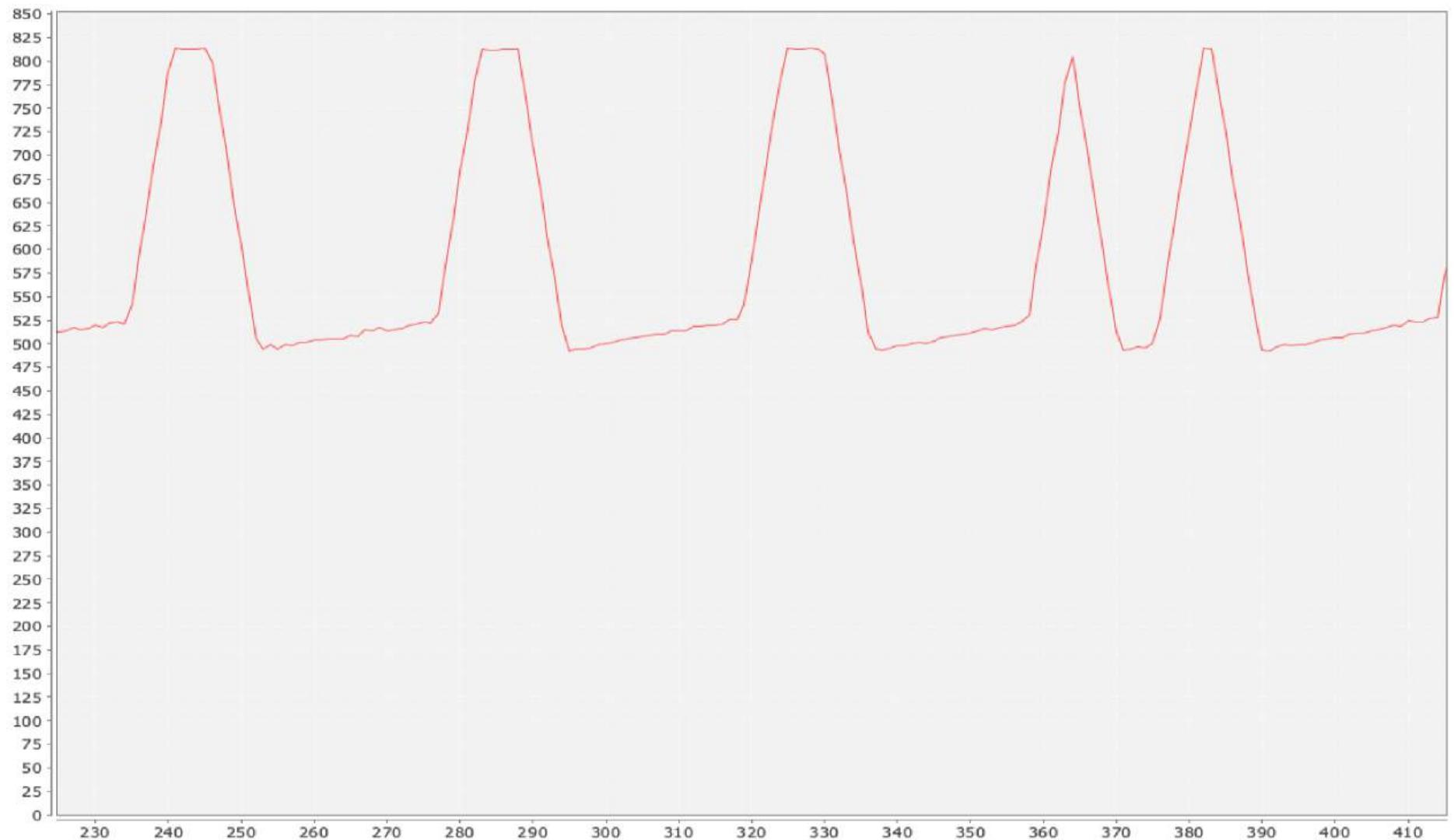
- Simply compute only k steps around the diagonal

s1/s2	v1	v2	v3	v4	v5	v6	v7	v8
v1	100	400	100					
v2	25	225	225	225				
v3	0	100	400	100	25			
v4		0	900	0	25	225		
v5			0	900	625	225	400	
v6				25	0	100	25	225
v7					225	25	100	0
v8						25	0	100

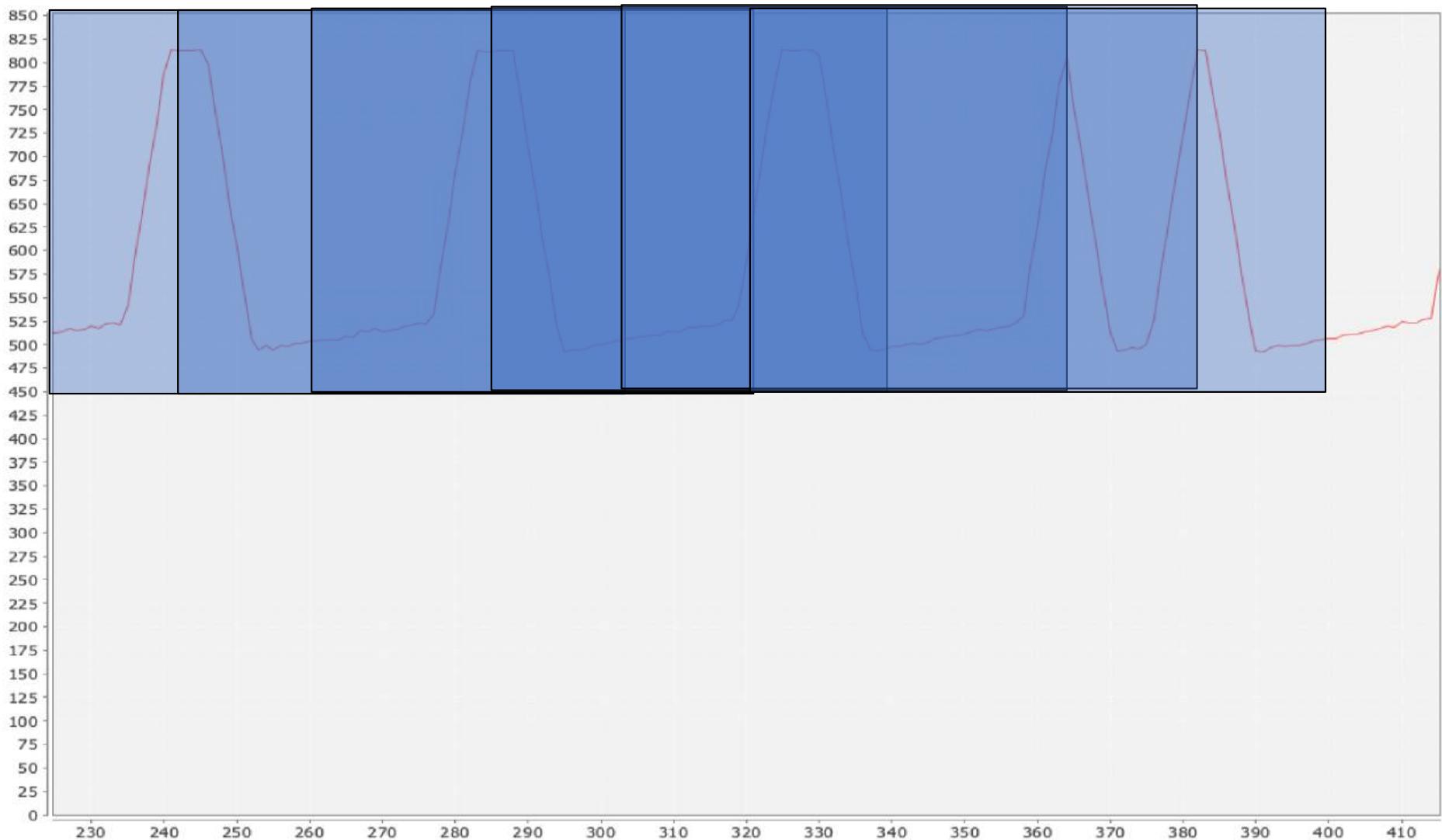
Too many points?



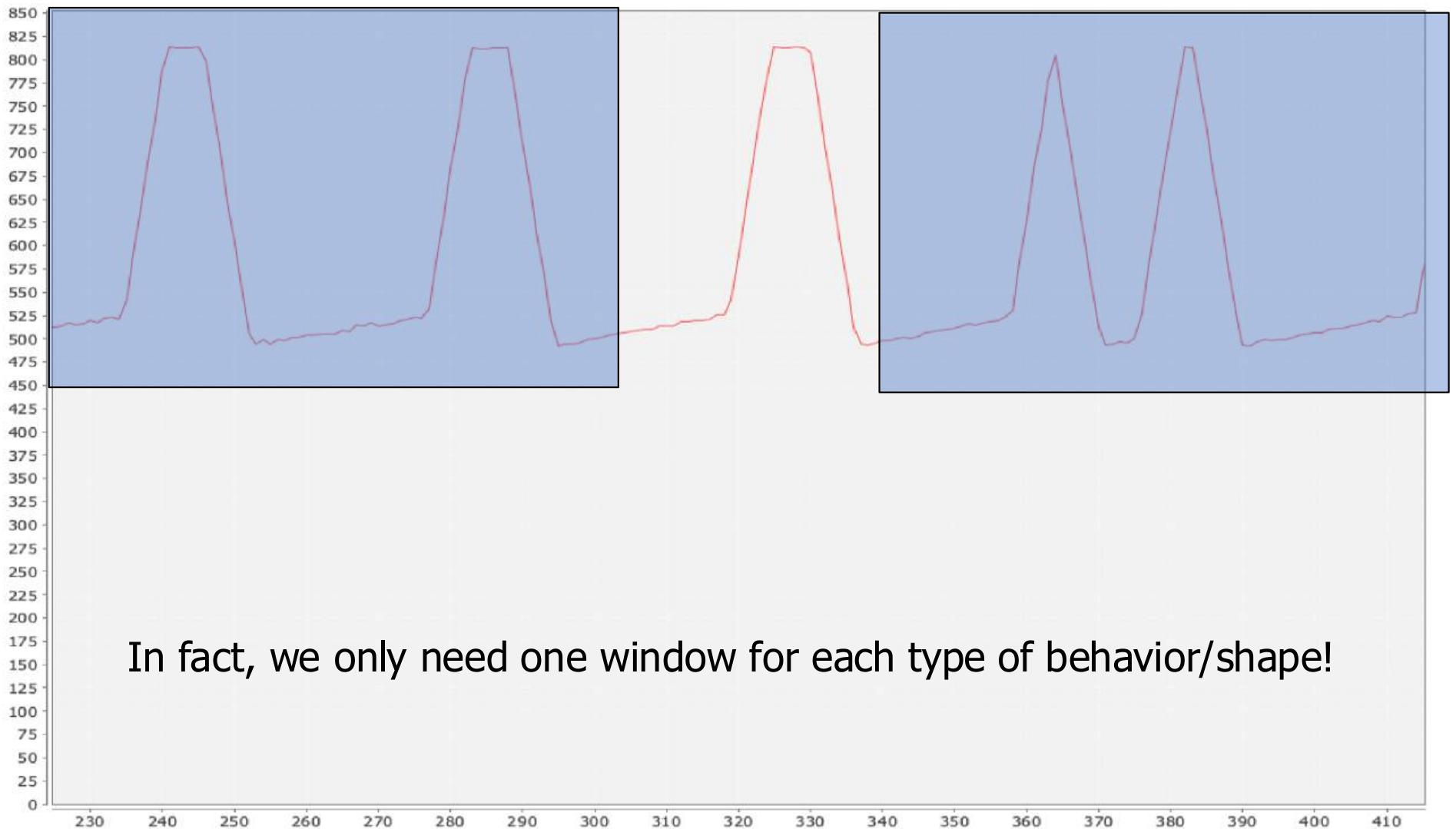
Too many points? -> sample!



To many windows?



No need to keep all train windows...



In fact, we only need one window for each type of behavior/shape!

Speeding up DTW

- Approximate (Fast) DTW
- Sample points
- Select representative windows -> prototyping
- Advantages of DTW:
 - Windows can be different sized
 - Windows do not have to be synchronized
 - Can be multi-variate...
 - ...
 - But Euclidean probably also works...

Data mining is trial-and-error

- What makes a good data scientist?

Data mining is trial-and-error

- What makes a good data scientist?
 - Has an intuitive understanding of ML and statistics
 - Knows how to apply techniques to data
 - Knows the required assumptions and what to do when violated
 - Knows how to transform data to fit with techniques
 - Can evaluate results, even without ground truth information
 - Can speed up learning algorithms
 - ...

Data mining is trial-and-error

- What makes a good data scientist?
 - Has an intuitive understanding of ML and statistics
 - Knows how to apply techniques to data
 - Knows the required assumptions and what to do when violated
 - Knows how to transform data to fit with techniques
 - Can evaluate results, even without ground truth information
 - Can speed up learning algorithms
 - ...
 - Never uses ML as a black box
 - Prefers simple over complex
 - Knows solutions are never perfect (be careful of overly tuning)
 - Understands business needs (not in this course)

Clustering - continued

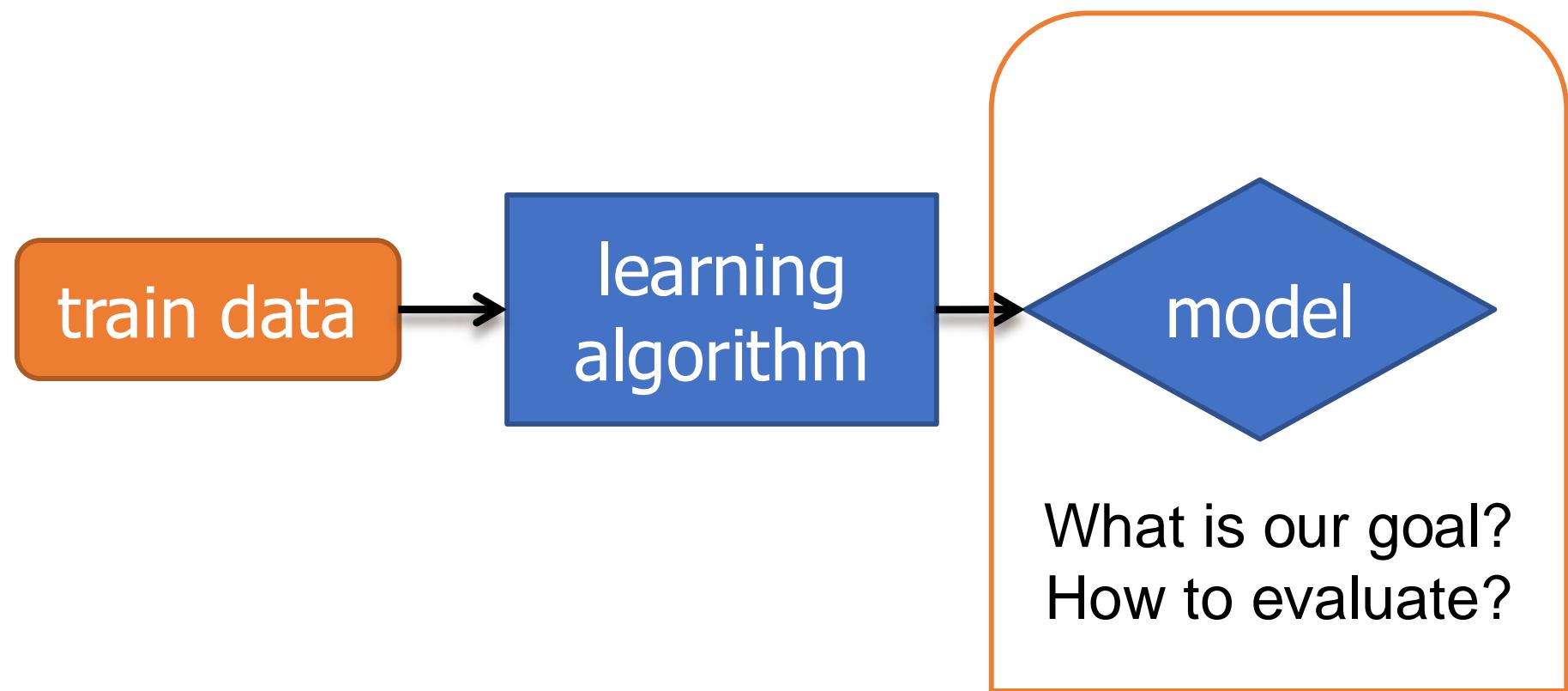
Today

- Clustering
 - Recap on k-Means & hierarchical
 - Density – DBScan
- Evaluation of unsupervised learning... *is hard!*
- Clustering large datasets
 - Batches
 - Prototyping
 - Sufficient statistics

Unsupervised learning



Unsupervised learning

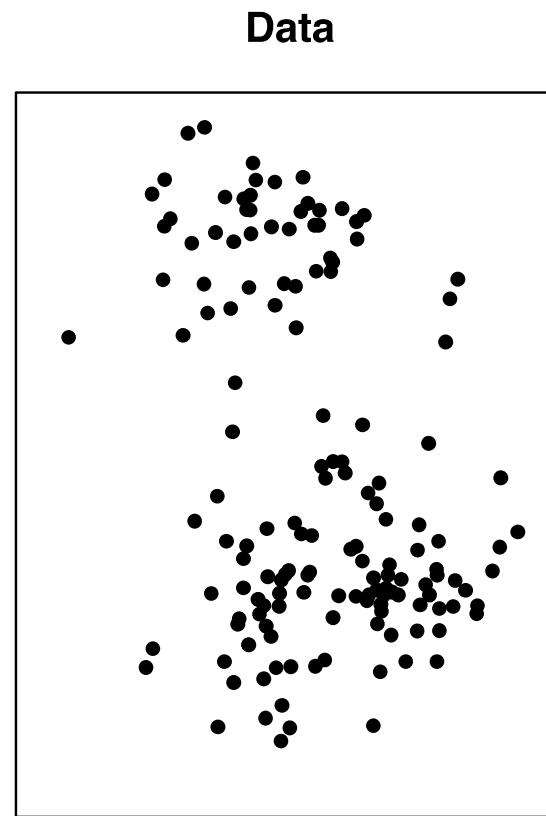


Clustering algorithms: three famous ones

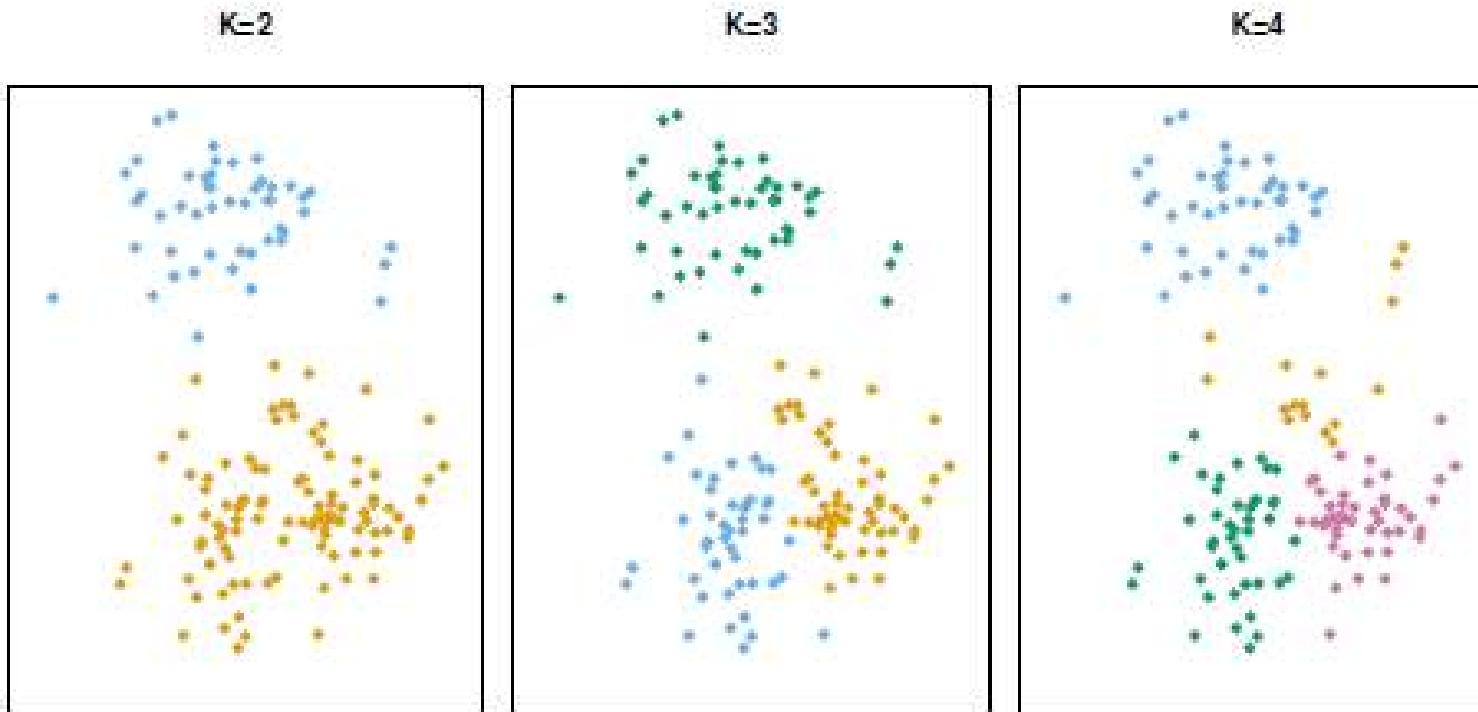
- K-means clustering
 - Group points based on distance to average cluster point (centroid).
- Hierarchical clustering
 - Group points based on distances between subgroups.
- DBScan
 - Group points based on distance and density to neighbors.

Clustering illustration

- How many clusters would you choose?



K-means clustering



- A simulated data set with 150 observations in 2-dimensional space. Panels show the results of applying K-means clustering with different values of K, the number of clusters.

K-means clustering: algorithm (recap)

- Randomly assign a number, from 1 to K, to each of the observations.
- Iterate until the cluster assignments stop changing
 - For each of the K clusters, compute the cluster centroid. The kth cluster centroid is the vector of the p feature means for the observations in the kth cluster.
 - Assign each observation to the cluster with center closest to the observation

K-means clustering: details

- The idea behind K-means clustering:
 - a good clustering is one for which the within-cluster variation (WCV) is as small as possible.
- The WCV for cluster k : measure the amount by which the observations within a cluster differ from each other.
- K-means clustering is to solve this **optimization problem**:

$$\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$$

K-means clustering: details

- The idea behind K-means clustering:
 - a good clustering is one for which the within-cluster variation (WCV) is as small as possible.
- The WCV for cluster k : measure the amount by which the observations within a cluster differ from each other.
- K-means clustering is to solve this **optimization problem**:

$$\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k}$$

k-means requires
Euclidean distance, why?

K-means clustering: details

- The algorithm is guaranteed to decrease the value of the objective at each iteration
 - Why?

$$\frac{1}{|C_k|} \sum_{i,i' \in C_k}$$

k-means requires Euclidean distance,
why?

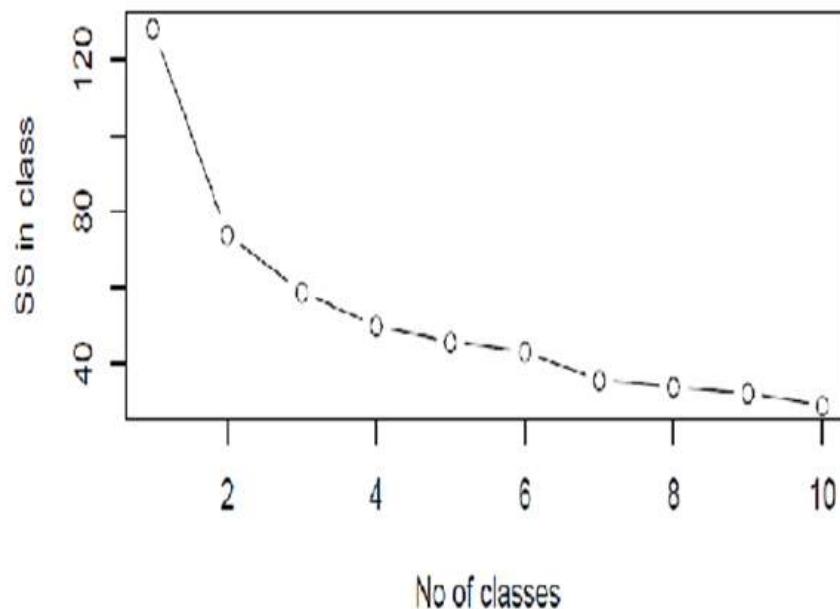
- Where cluster

$$\bar{x}_{kj} = \frac{1}{|C_k|} \sum_{i \in C_k} x_{ij}$$

*Minimizes squared error
Without Euclidean space, the
centroid is meaningless!*

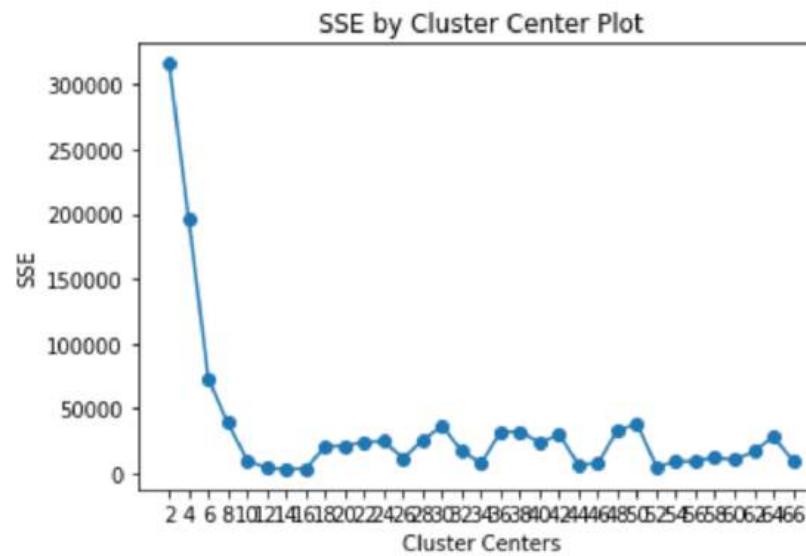
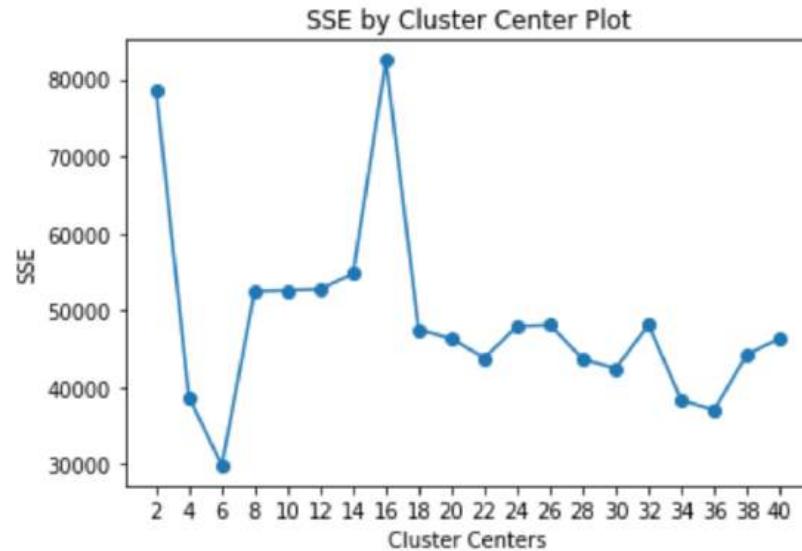
Determine the number of clusters in Kmeans

- Based on the sum of squares within the clusters (WCV) for a solution against the number of clusters
- The “optimal” number of clusters can be determined by the “elbow criterion”.



- This "elbow" cannot always be unambiguously identified ...

Practice...



K-means for other distances?

- What to do for Manhattan (sum) distance?

K-means for other distances?

- What to do for Manhattan (sum) distance?
 - Use PAM: partition around medoids, a medoid is the median data point
 - Minimizes absolute error instead of squared
 - More expensive than k-means (medoid selection is discrete)

K-means for other distances?

- What to do for Manhattan (sum) distance?
 - Use PAM: partition around medoids, a medoid is the median data point
 - Minimizes absolute error instead of squared
 - More expensive than k-means (medoid selection is discrete)
- What for cosine?

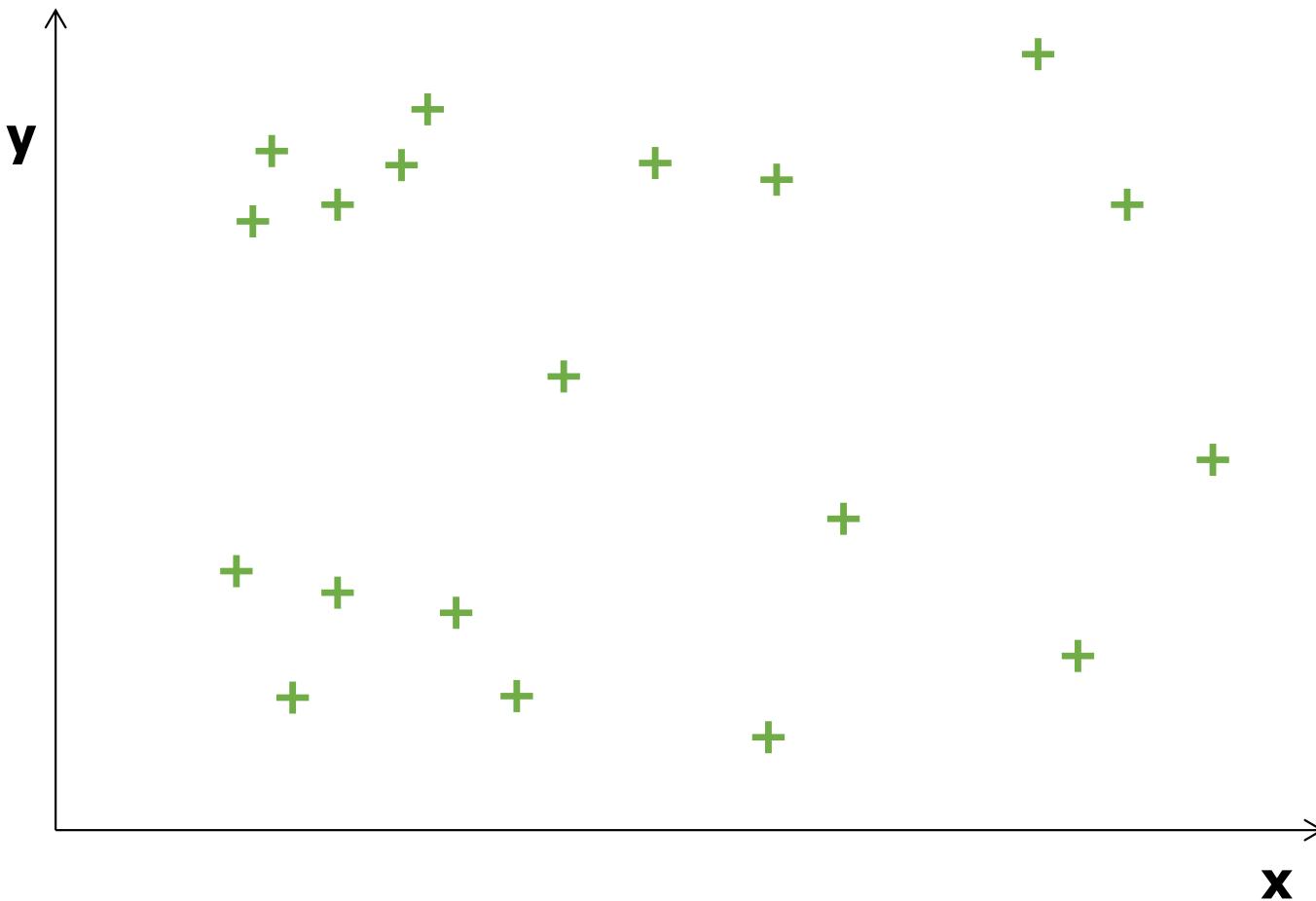
K-means for other distances?

- What to do for Manhattan (sum) distance?
 - Use PAM: partition around medoids, a medoid is the median data point
 - Minimizes absolute error instead of squared
 - More expensive than k-means (medoid selection is discrete)
- What for cosine?
 - Is **non-metric**, hence the mean is really odd...
 - Can we fix it?

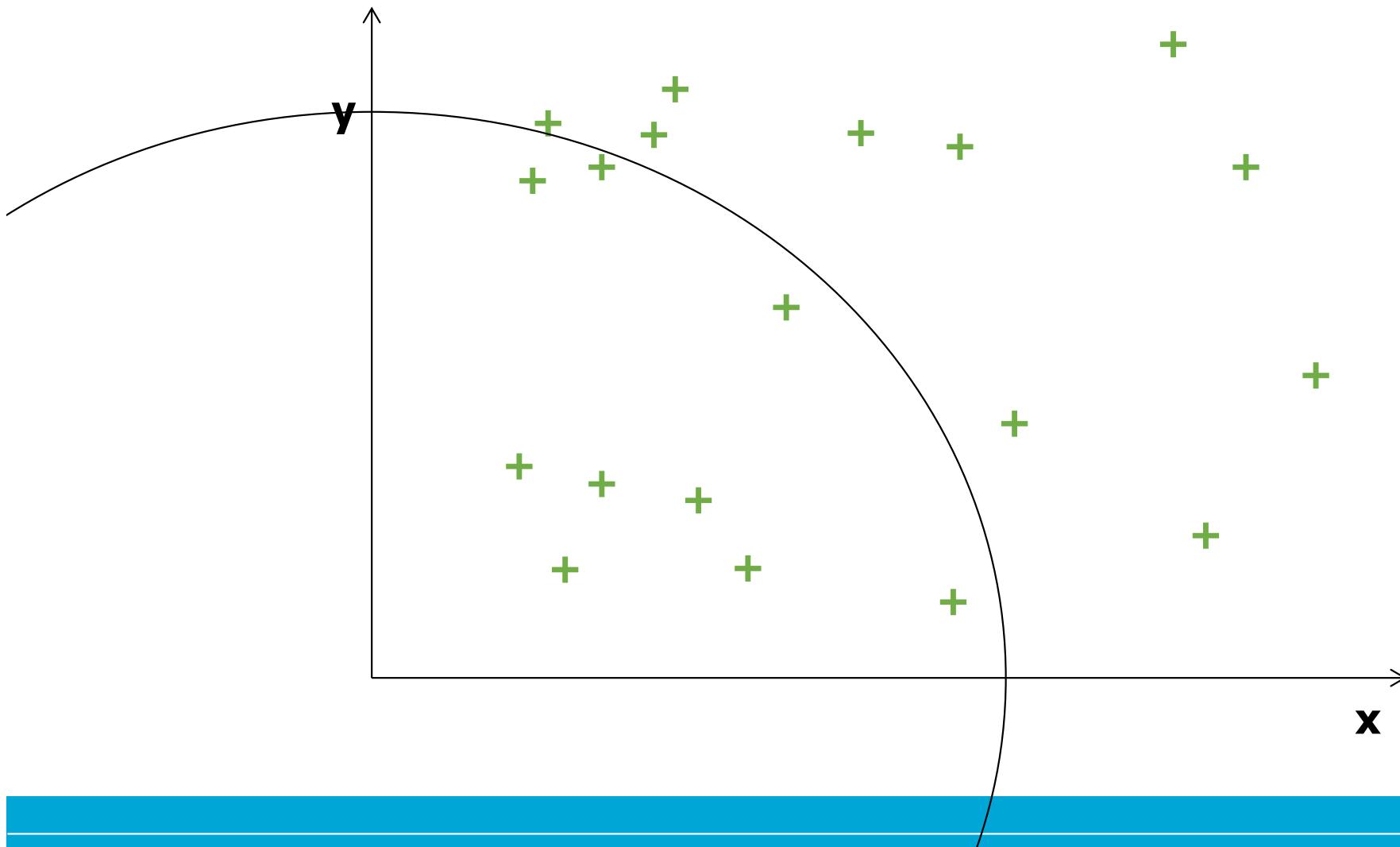
K-means for other distances?

- What to do for Manhattan (sum) distance?
 - Use PAM: partition around medoids, a medoid is the median data point
 - Minimizes absolute error instead of squared
 - More expensive than k-means (medoid selection is discrete)
- What for cosine?
 - Is **non-metric**, hence the mean is really odd...
 - In practice, you can normalize every data row and simply apply standard k-means, why?

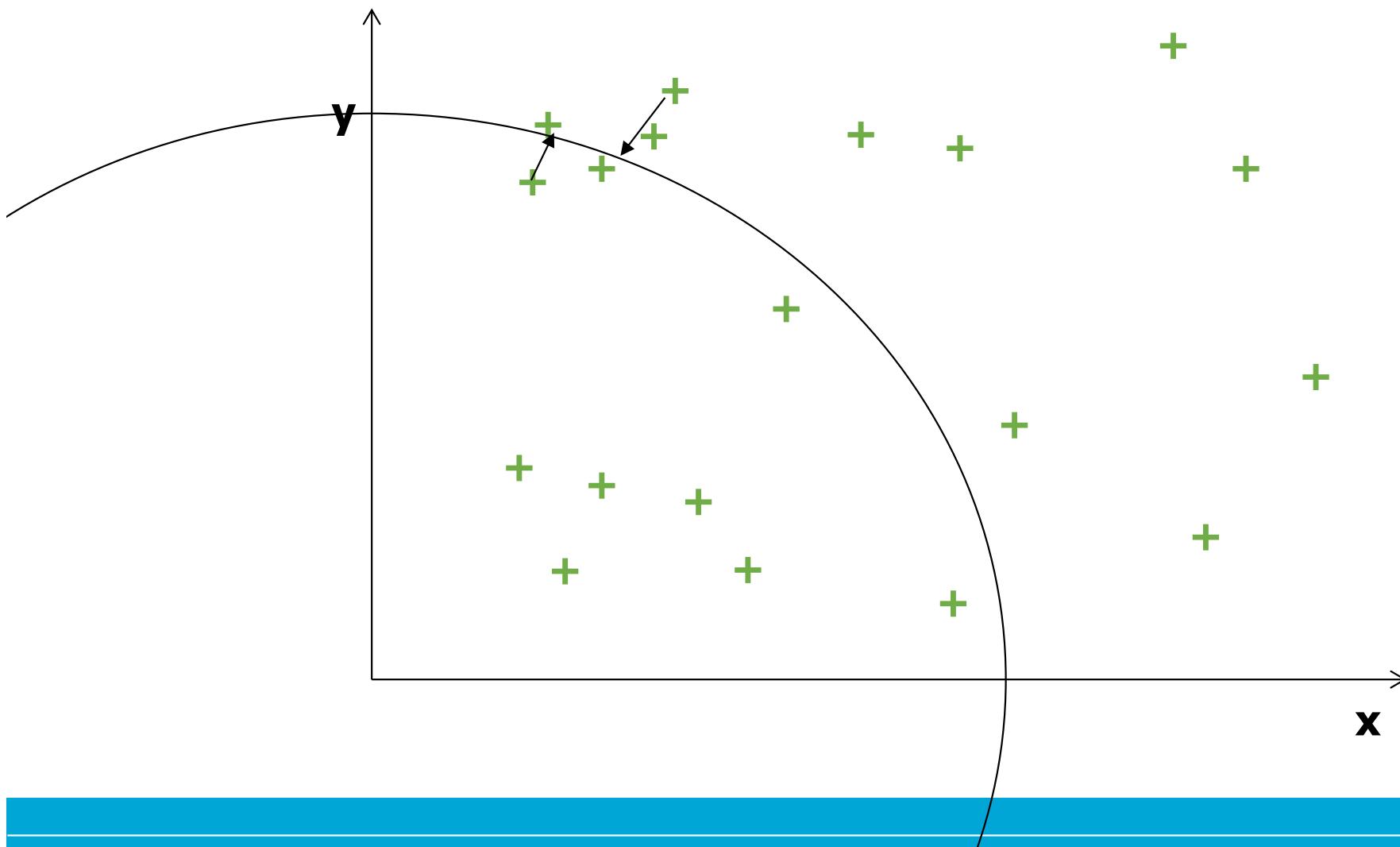
Row normalization



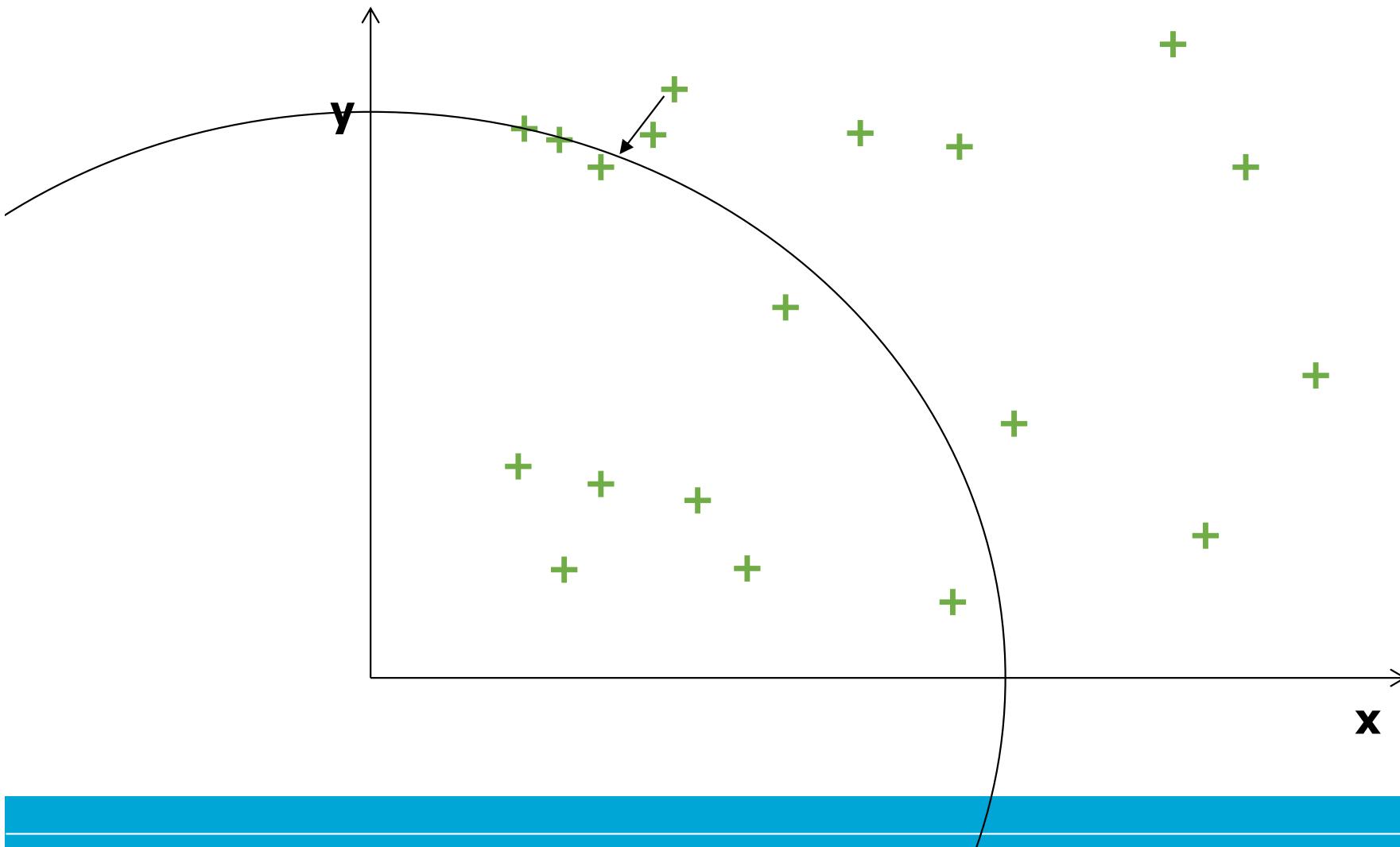
Row normalization



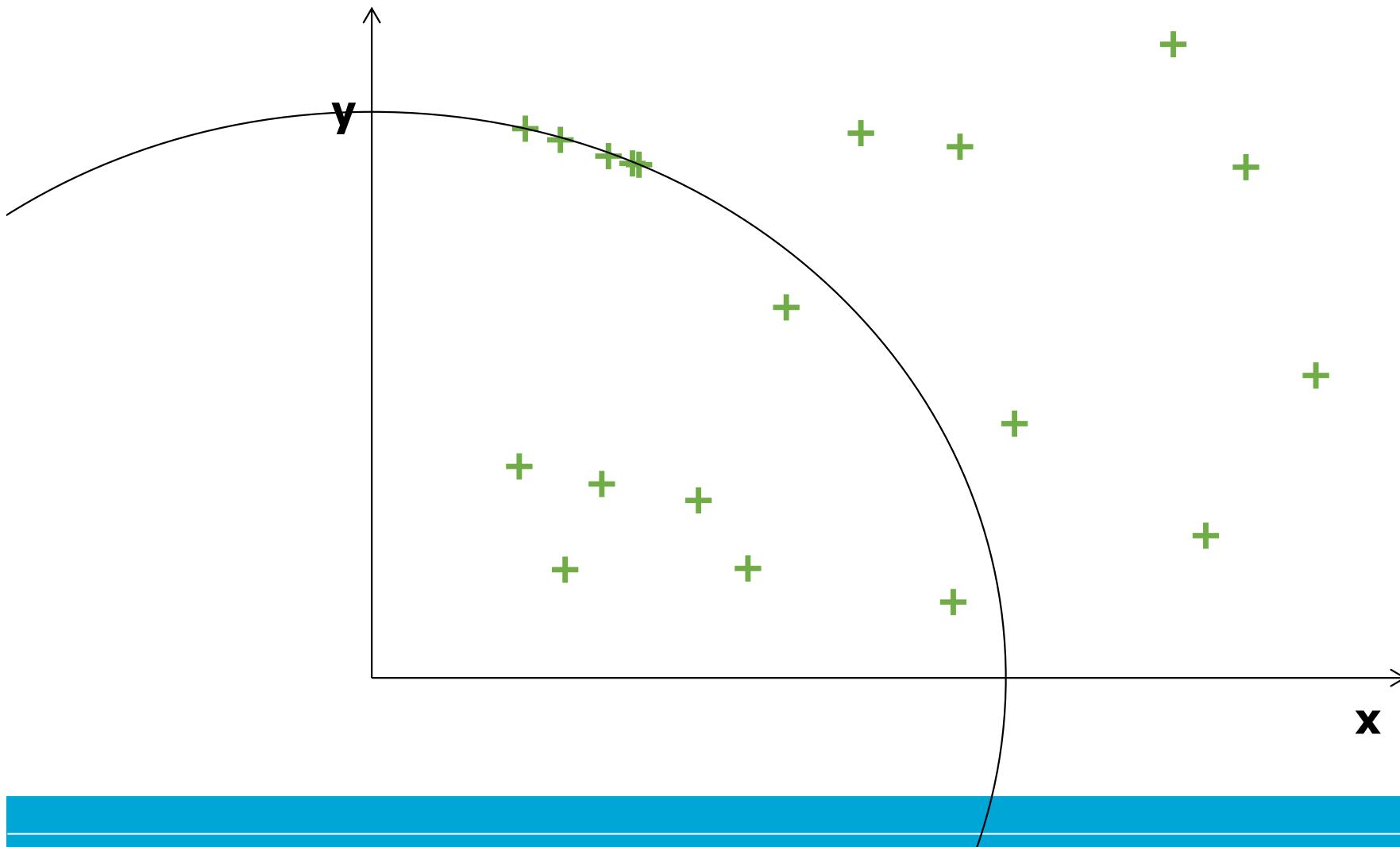
Row normalization



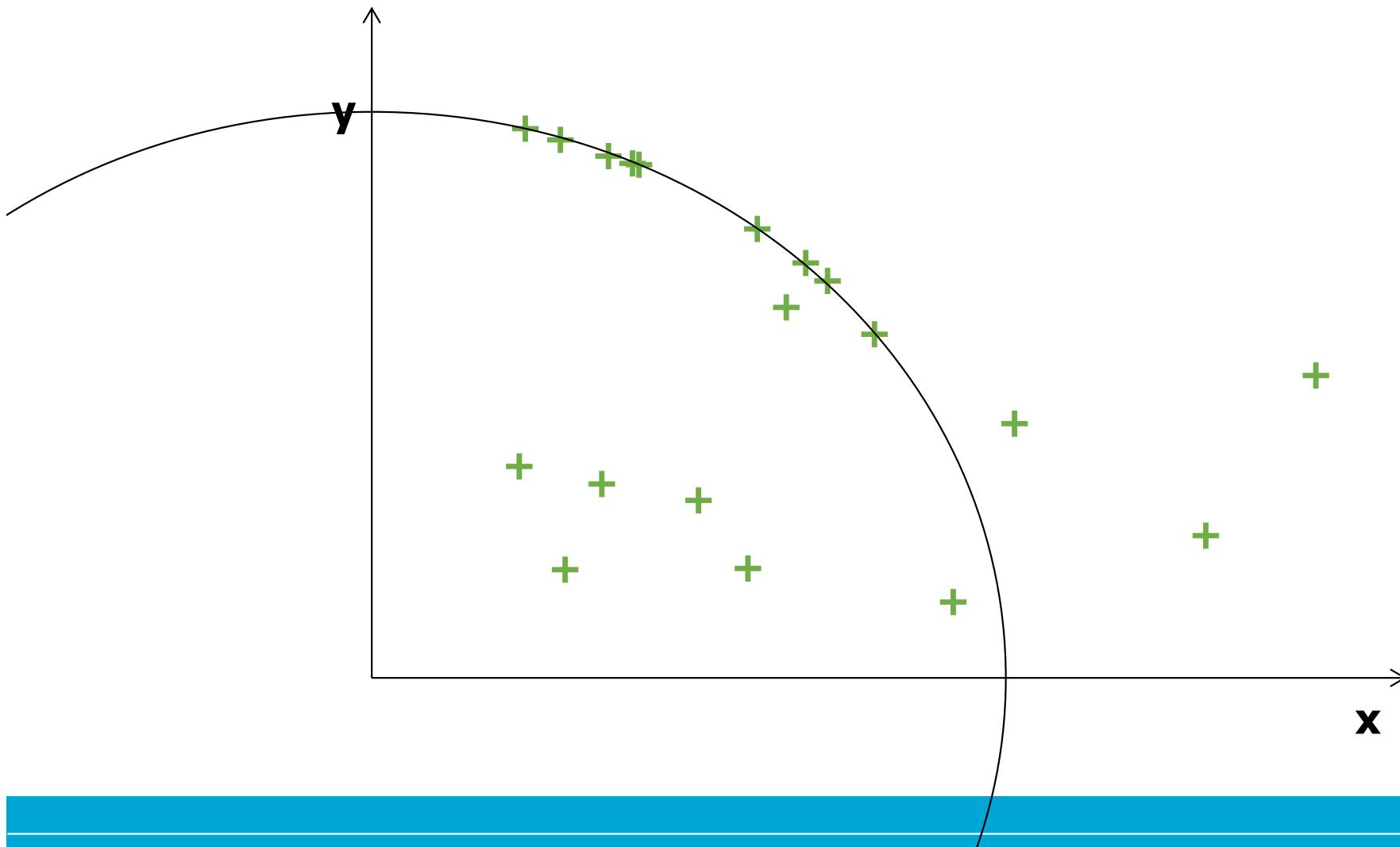
Row normalization



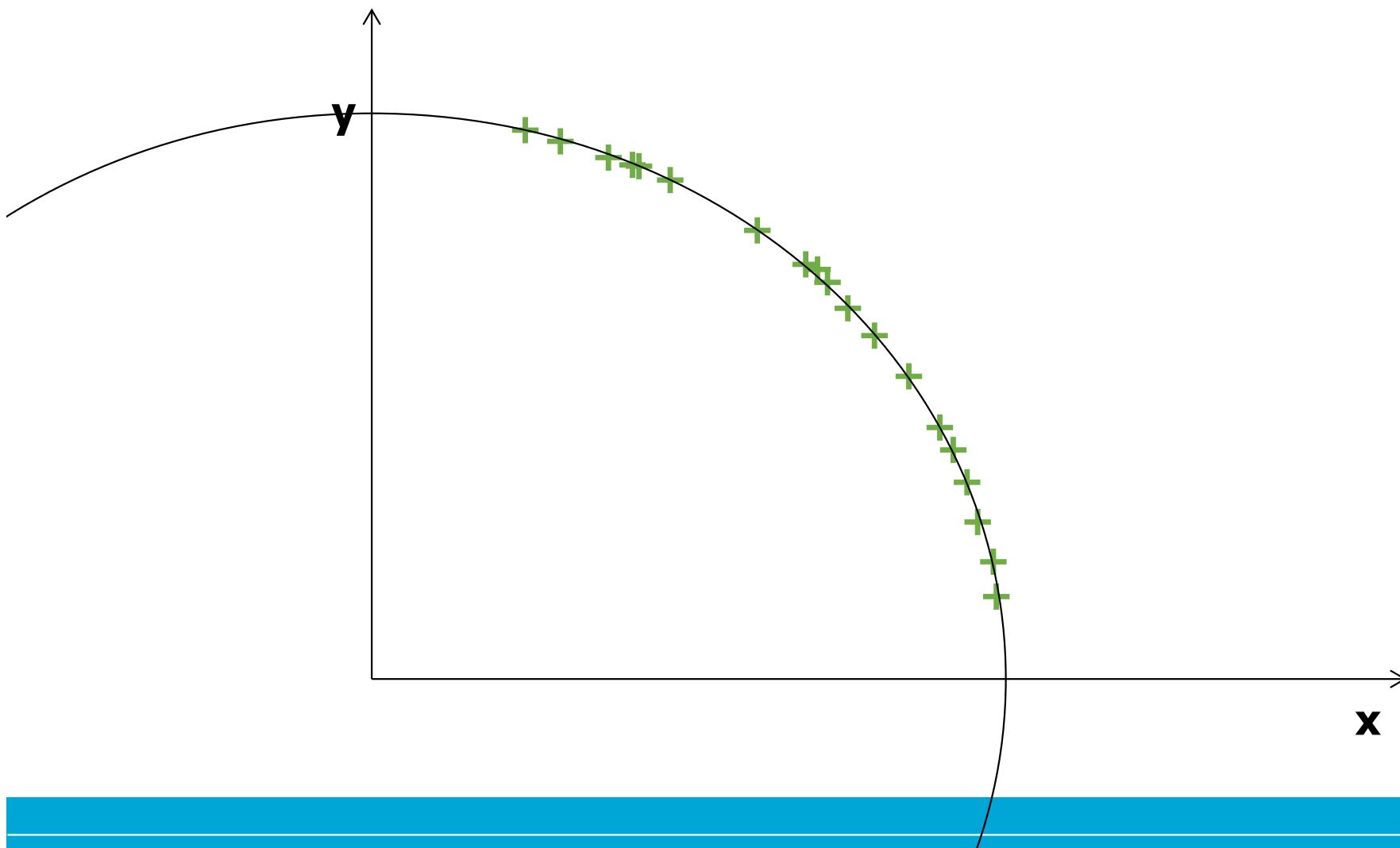
Row normalization



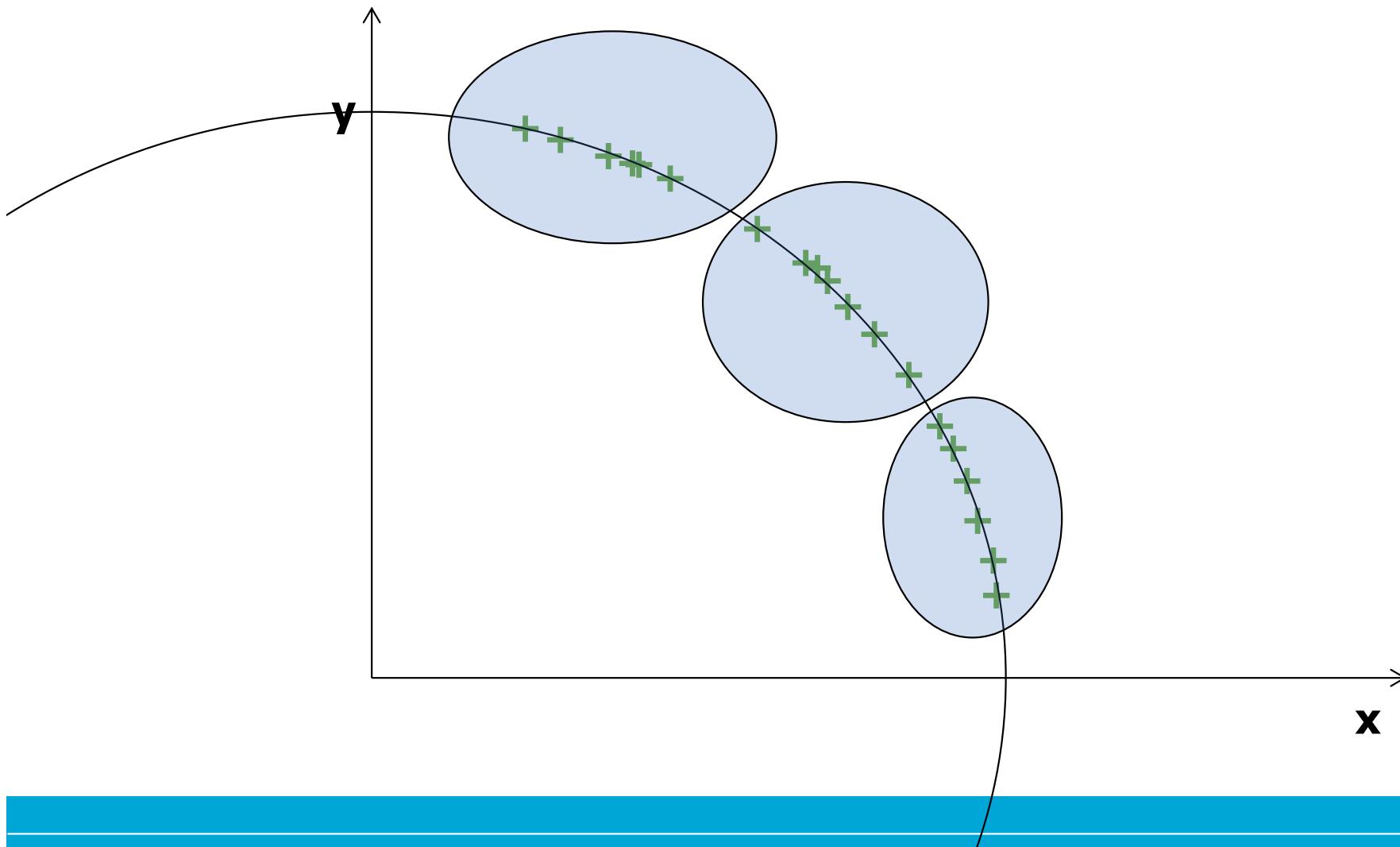
Row normalization



Row normalization



Row normalization – clustering



K-means for other distances?

- What about DTW?

Is DTW a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:
 - Non-negativity: $d(A, B) \geq 0$
 - Identity: $d(A, B) = 0$ iff $A = B$
 - Symmetry: $d(A, B) = d(B, A)$
 - Triangle inequality: $d(A, C) \leq d(A, B) + d(B, C)$

Is DTW a distance metric?

- A distance metric $d(\cdot, \cdot)$ should satisfy the metric properties:
 - Non-negativity: $d(A, B) \geq 0$
 - Identity: $d(A, B) = 0$ iff $A = B$
 - Symmetry: $d(A, B) = d(B, A)$
 - Triangle inequality: $d(A, C) \leq d(A, B) + d(B, C)$
- Does not satisfy Triangle inequality:
 - $x = [0, 1, 1, 2], y = [0, 1, 2], z = [0, 2, 2]$
 - Then $d(x, y) = 0, d(x, z) = 2$, and $d(y, z) = 1$
 - Counterexample: $d(x, z) > d(x, y) + d(y, z)$

K-means for other distances?

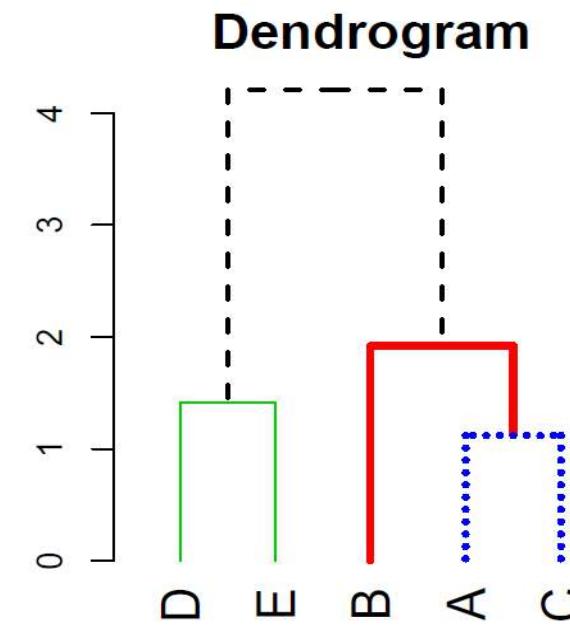
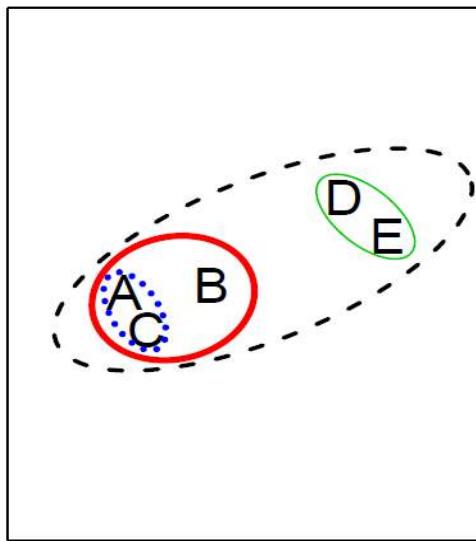
- What about DTW?
 - No, it is not Euclidean! What is the medoid?
 - It also does not satisfy the triangle inequality...

Another typical approach: bottom-up/top-down learning

- We do not need to commit to a particular choice of K.
- Bottom-up or agglomerative clustering
 - A dendrogram is built starting from the leaves and combining clusters up to the trunk
- Top-down or agglomerative clustering
 - A dendrogram is built starting from the root and splitting clusters to the leaves

Hierarchical Clustering Algorithm

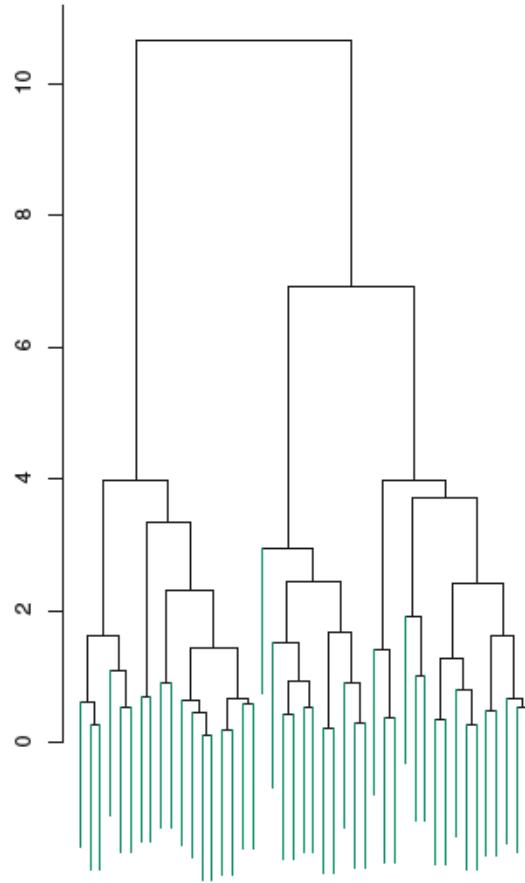
- Begin with n observations. Treat each observation as its own cluster
- Identify the closest two clusters and merge them
- Repeat
- Ends when all points are in a single cluster



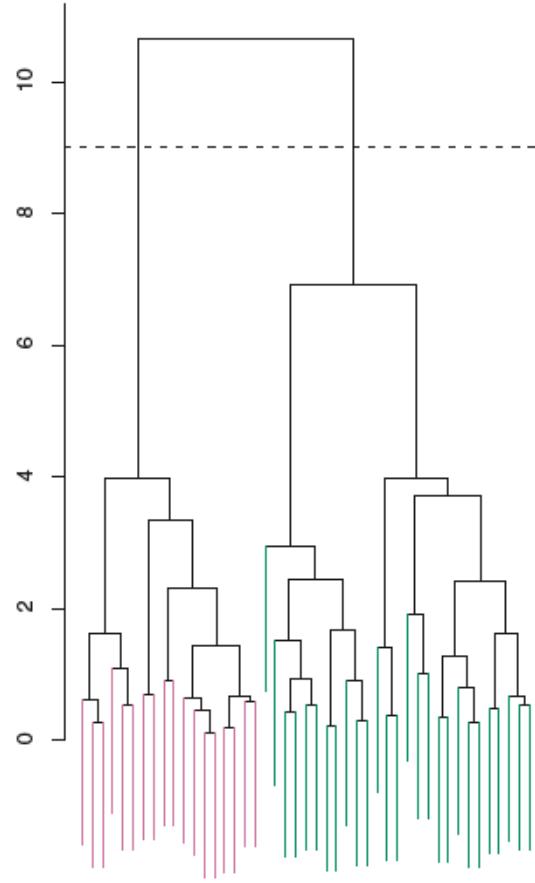
Cluster distance

- The distance between the clusters is called linkage
- Different specifications for linkage
 - Single linkage: Distance between clusters is the distance of the closest points (minimum spanning tree)
 - Complete linkage: Distance between clusters is the distance of the farthest points
 - Average linkage: mean distance between all the points in the two clusters
 - Ward distance: difference between the total within cluster sum of squares for the two clusters separately, and the within cluster sum of squares resulting from merging the two clusters in one cluster

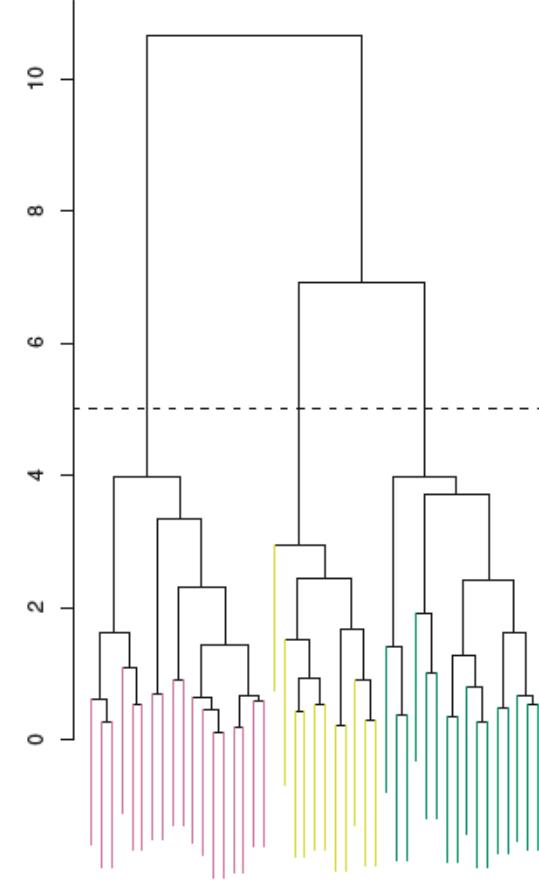
When to stop?



Dendrogram obtained



Cut the dendrogram at height 9, resulting in two distinct clusters



Cut the dendrogram at height 5, resulting in three distinct clusters

Hierarchical clustering

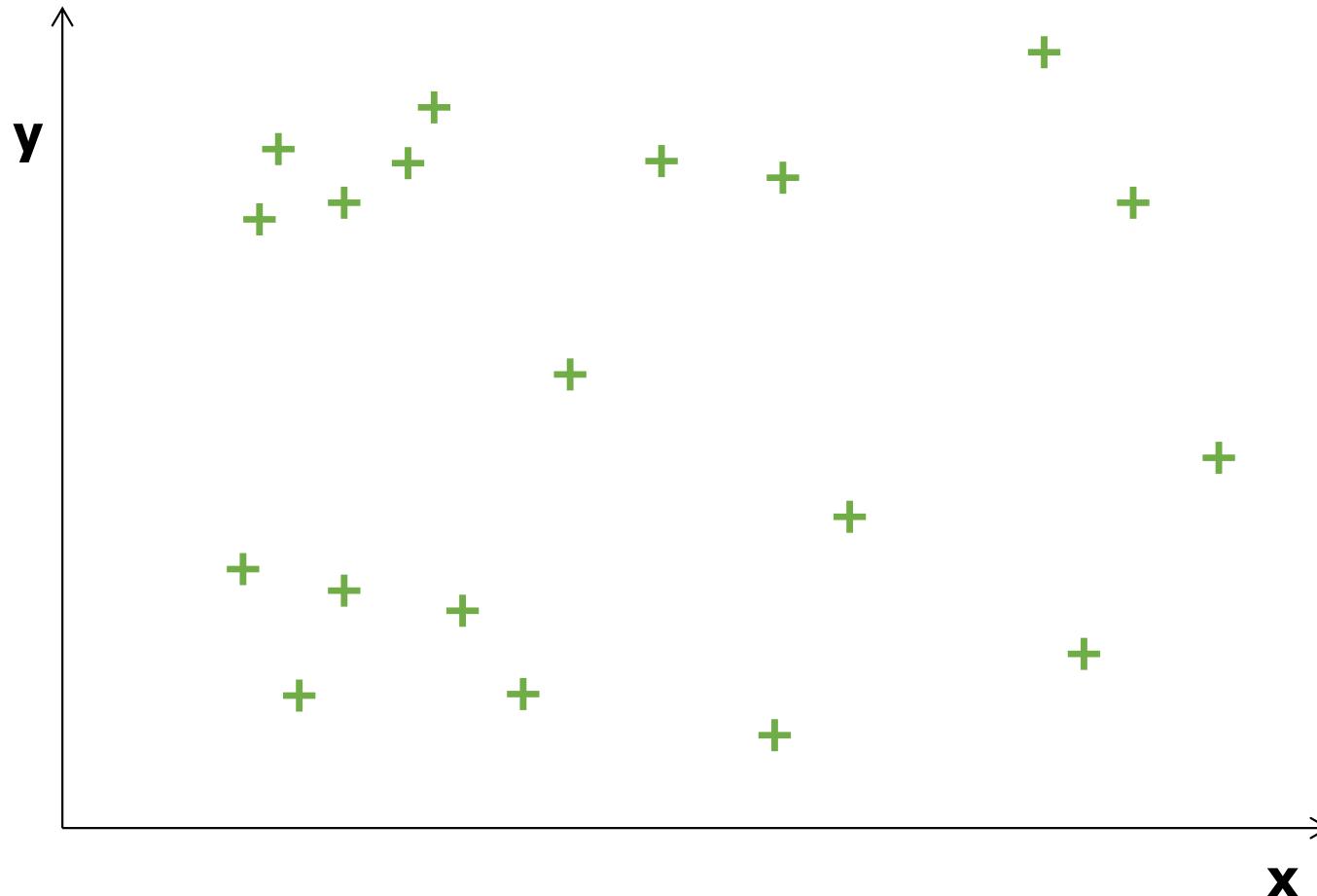
- Slow, but can handle non-centroid clusters
- Does the distance need to be a metric?

Hierarchical clustering

- Slow, but can handle non-centroid clusters
- Does the distance need to be a metric?
 - *Depends on applied linking method:*
 - Single, complete, and average only requires symmetry and non-negative
 - Ward, or centroid-based require Euclidean distance since they minimize squared error

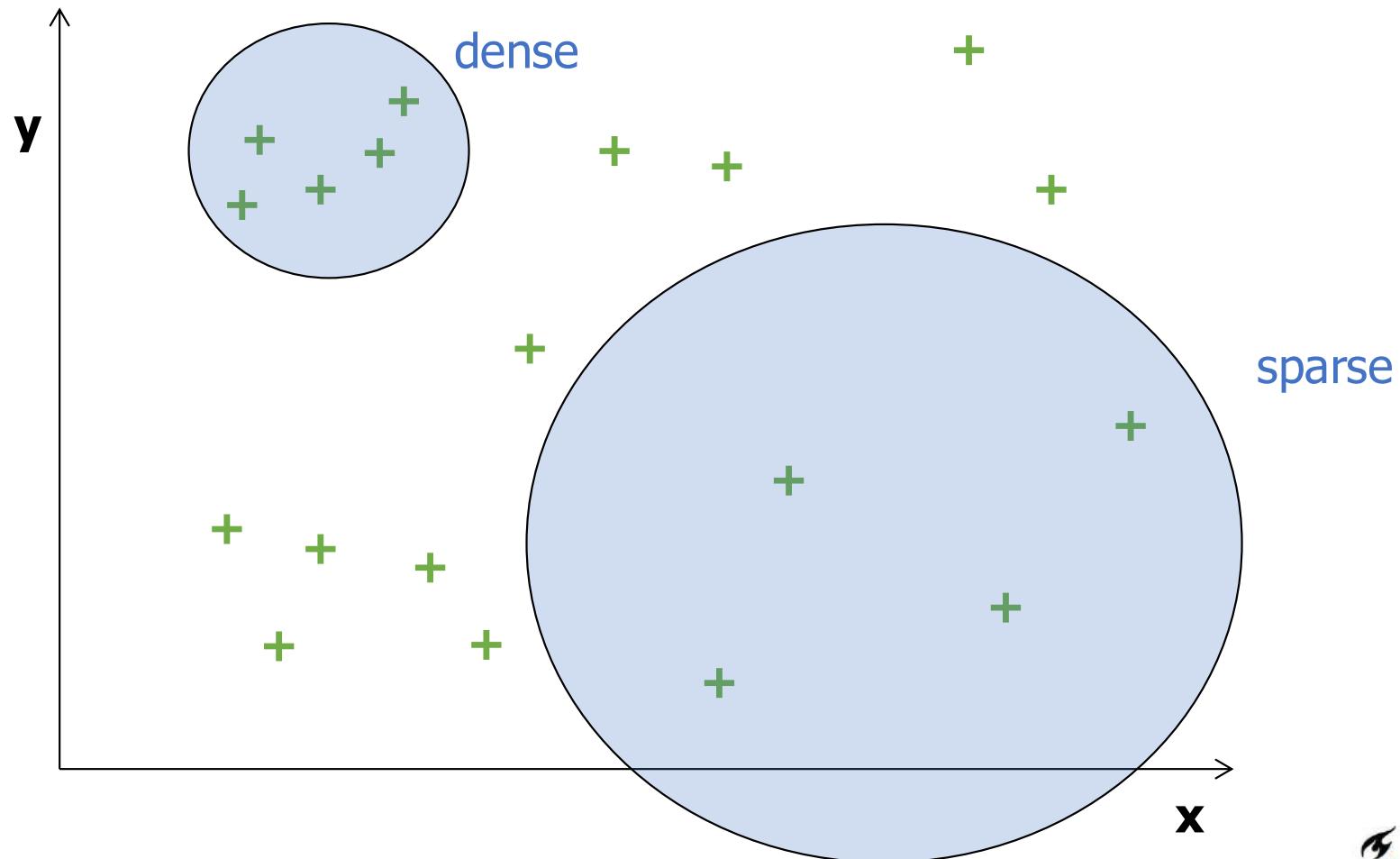
DBSCAN

- Clustering with density, what is density?



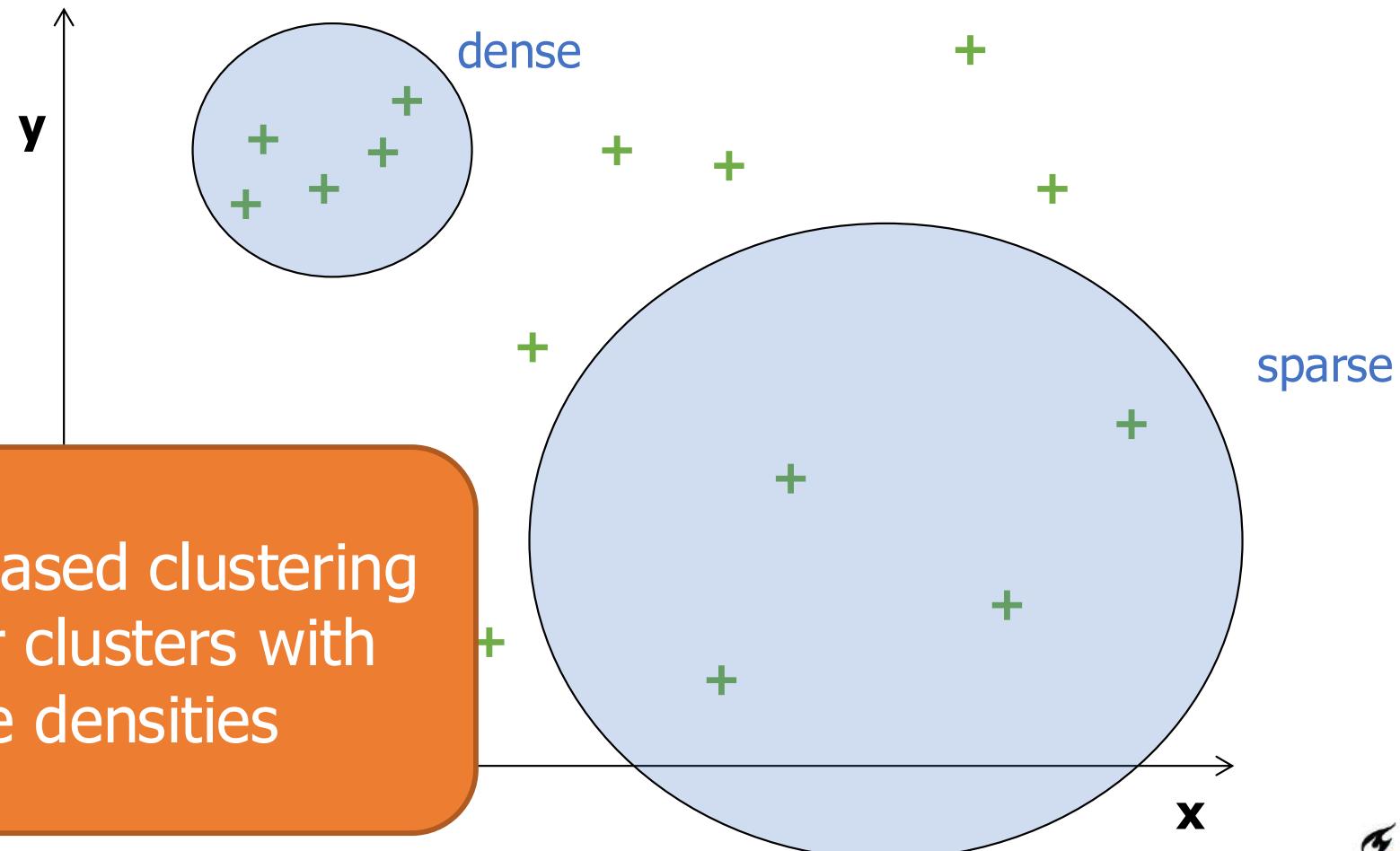
DBSCAN

- Clustering with density, what is density?



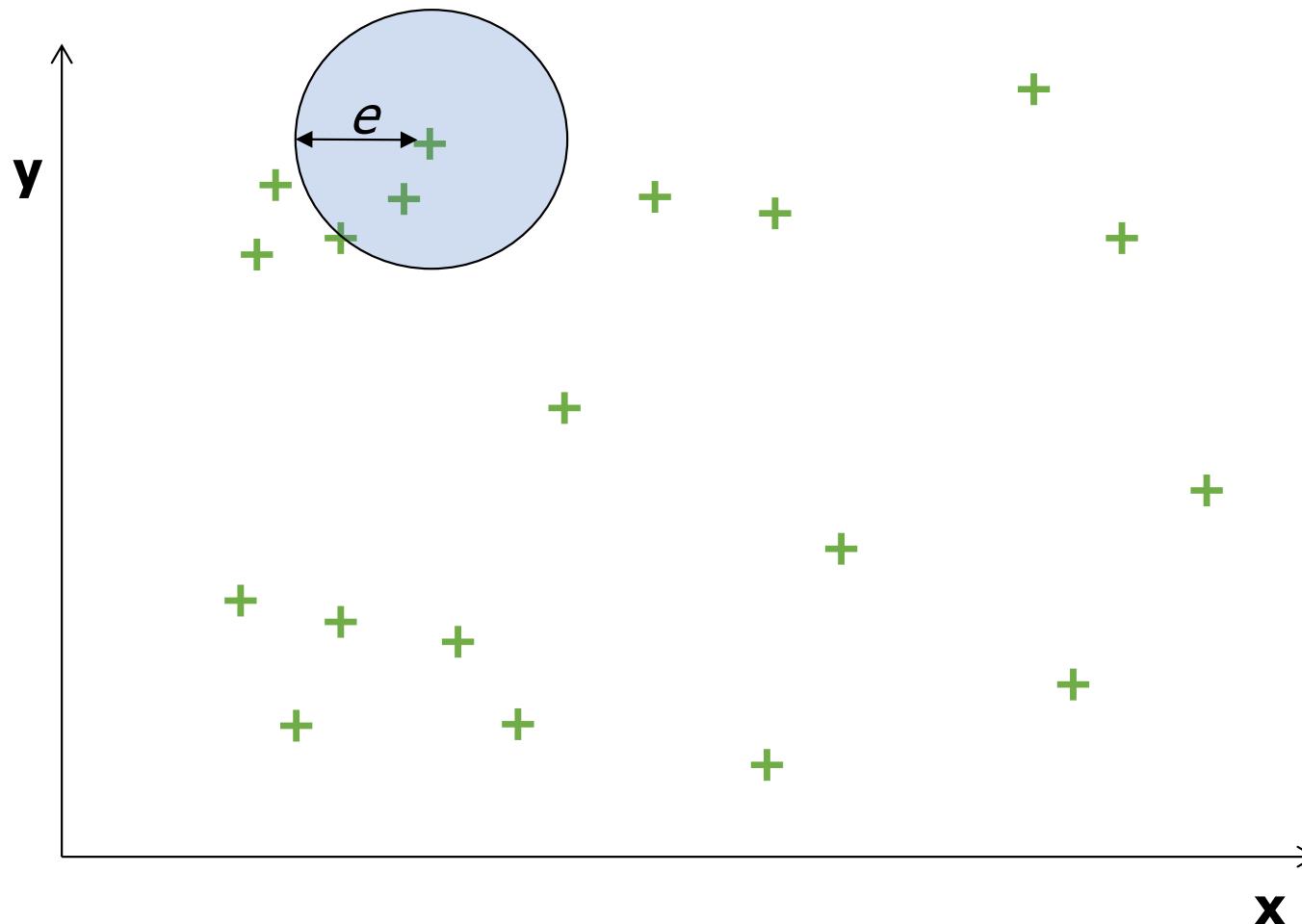
DBSCAN

- Clustering with density, what is density?



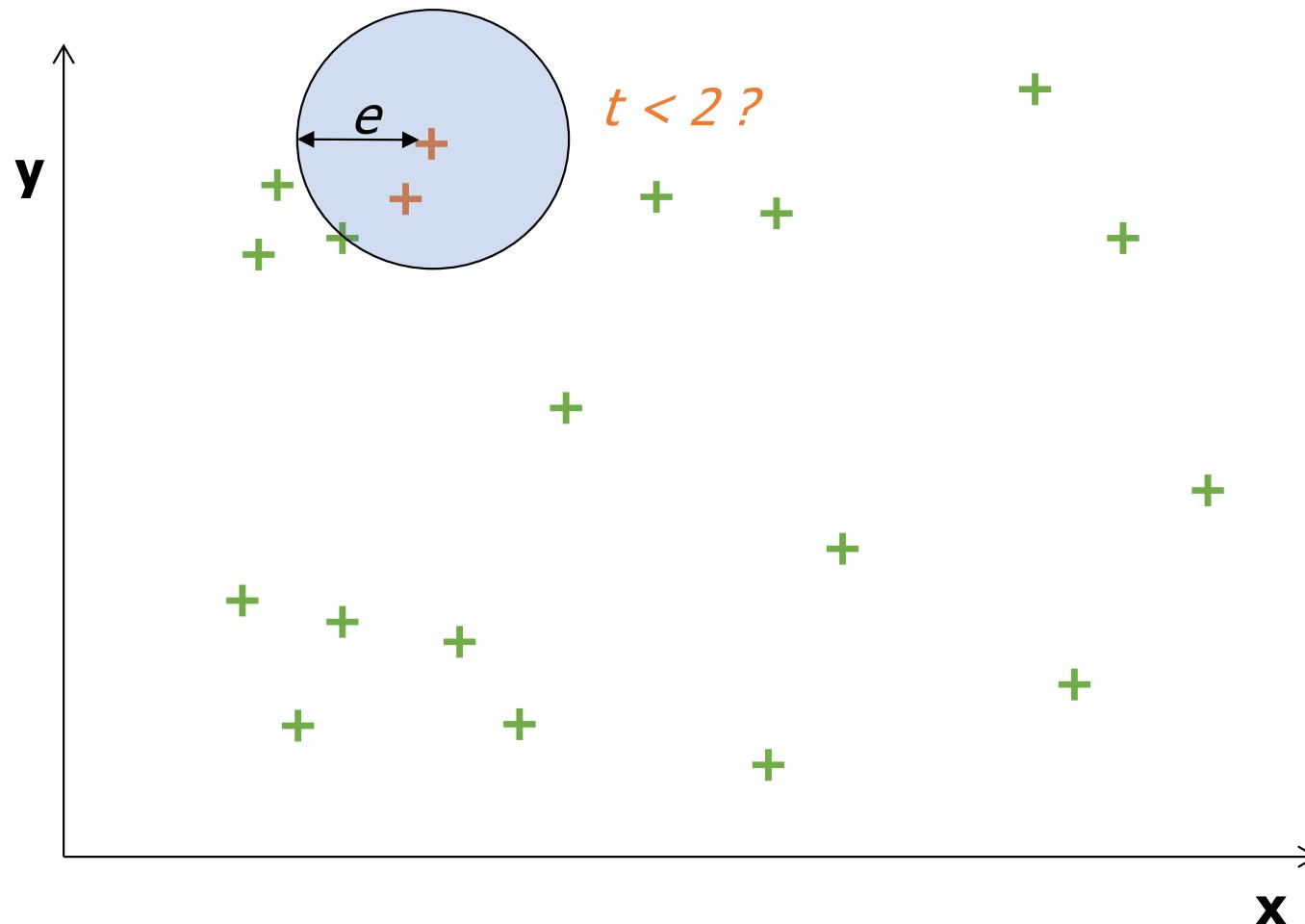
DBSCAN – Core, Border, Noise

- Given t and e



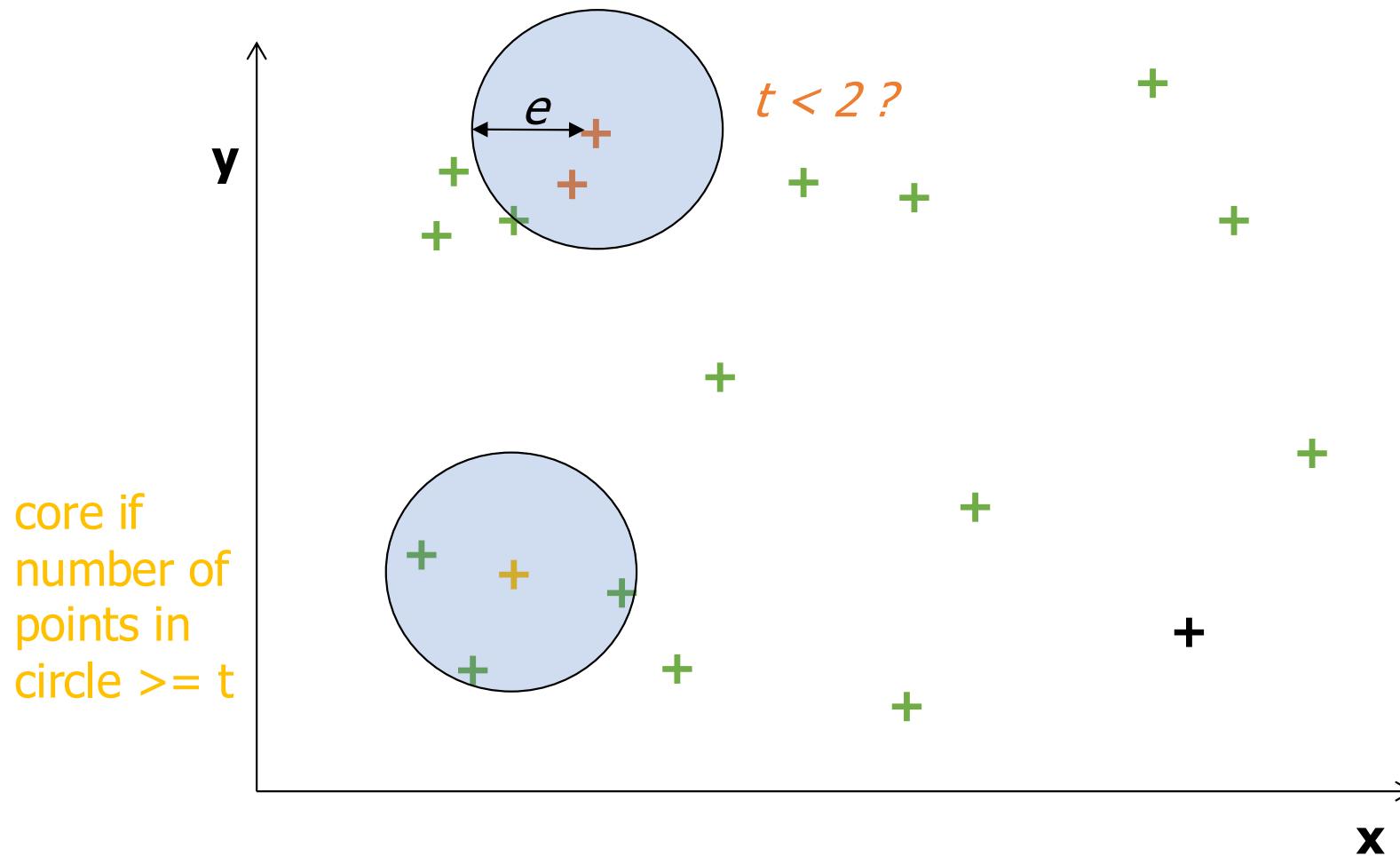
DBSCAN – Core, Border, Noise

- Given threshold t and radius e



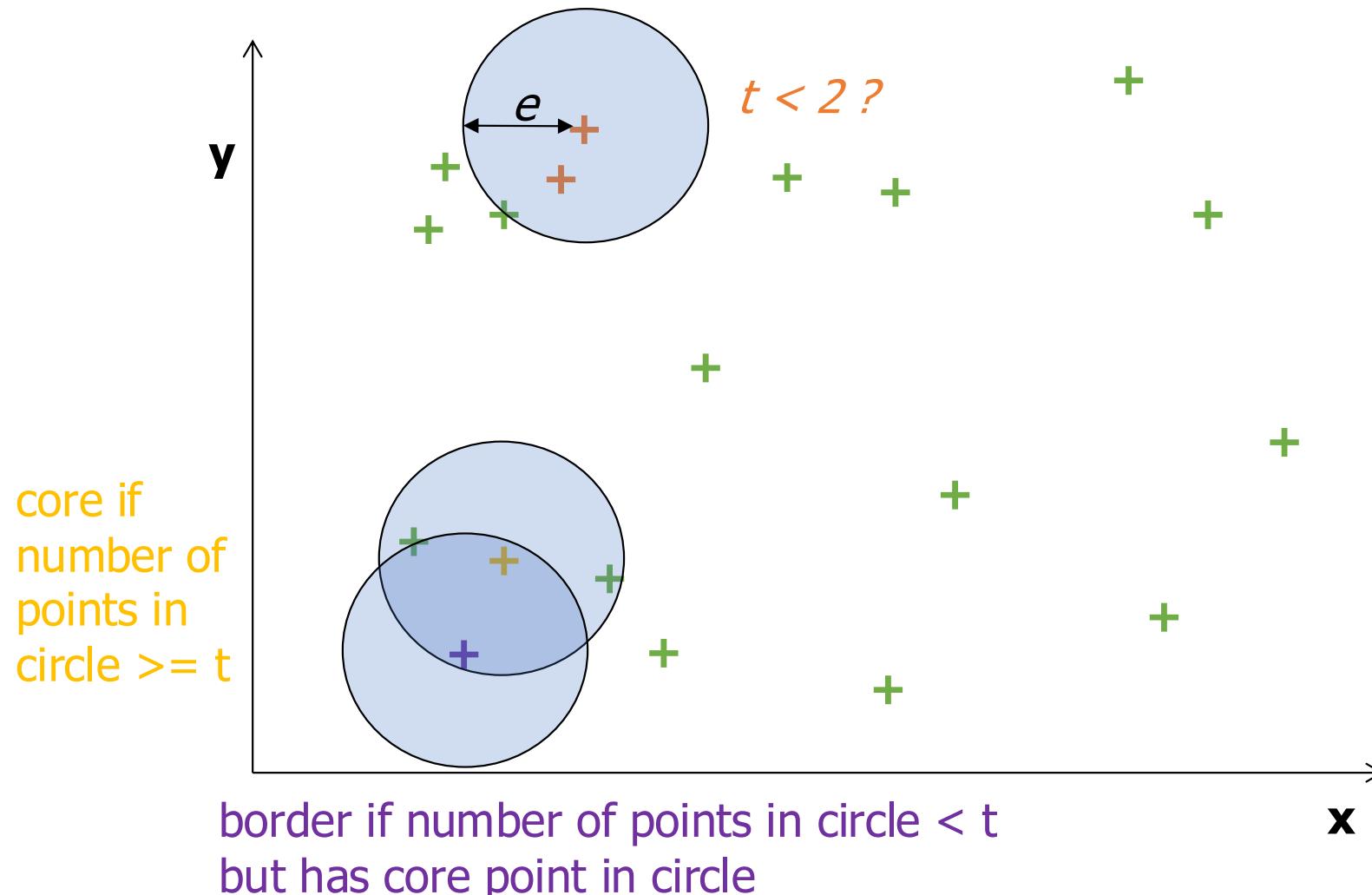
DBSCAN – Core, Border, Noise

- Given threshold t and radius e



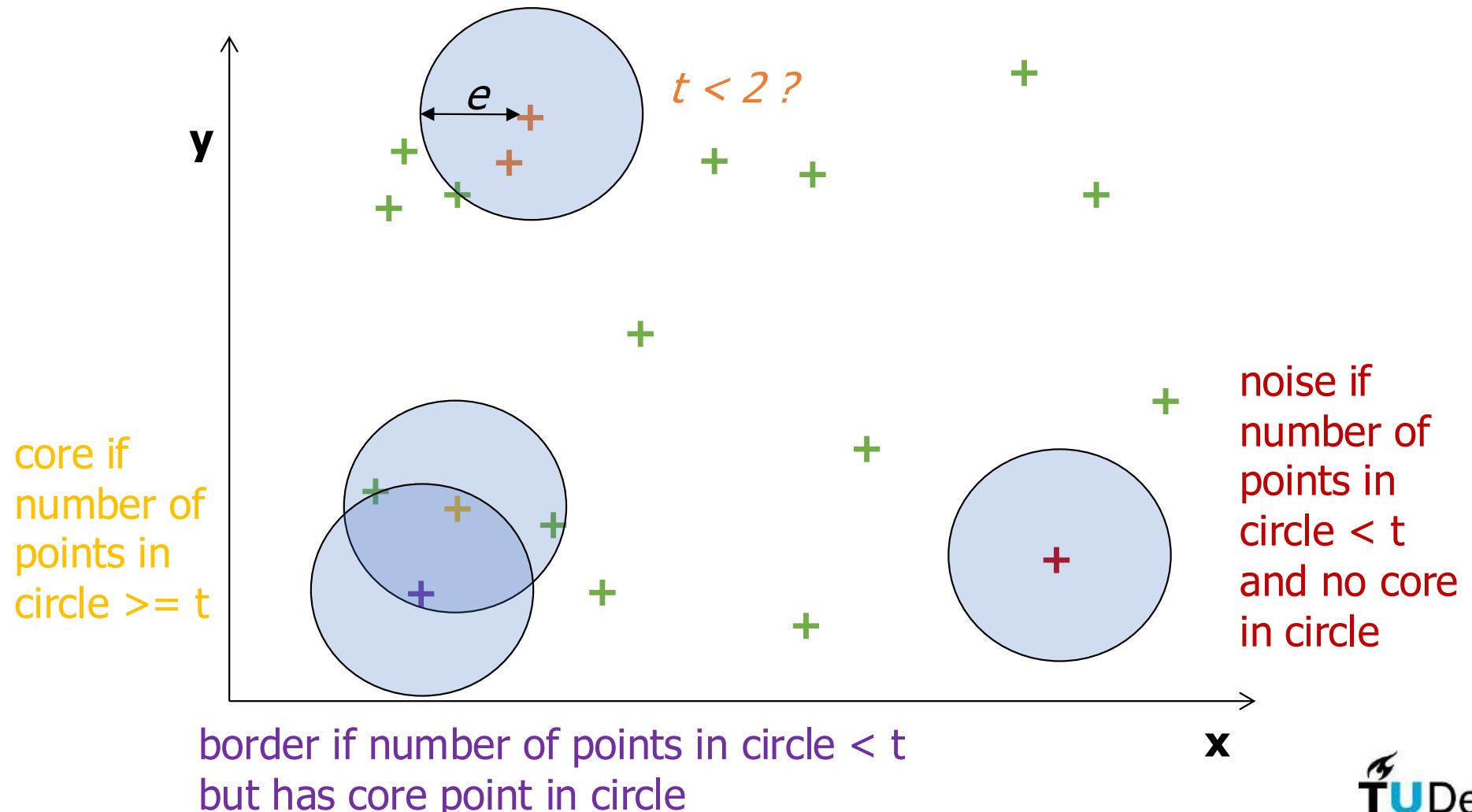
DBSCAN – Core, Border, Noise

- Given threshold t and radius e



DBSCAN – Core, Border, Noise

- Given threshold t and radius e

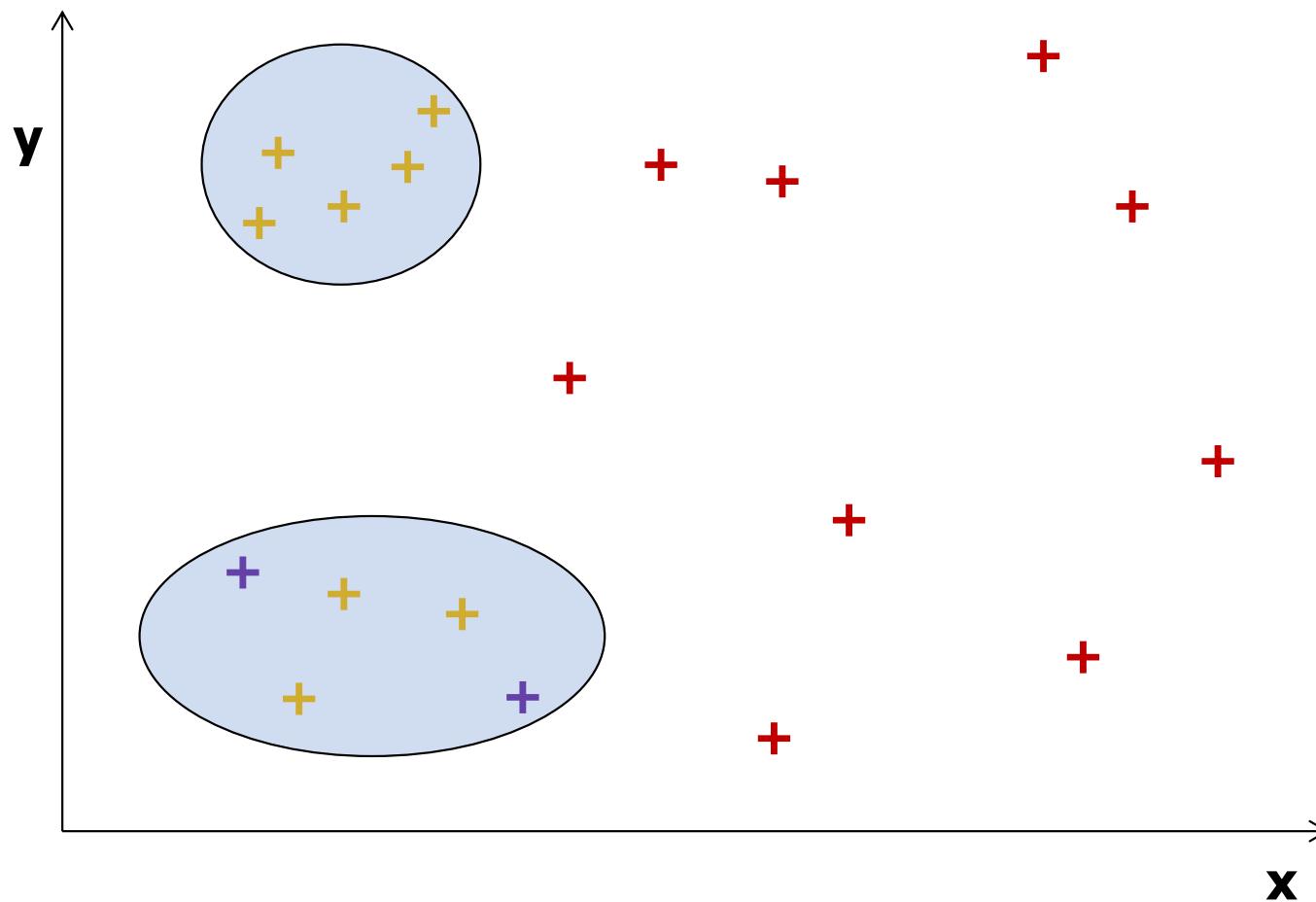


DBSCAN

1. Determine core, border, and noise points
2. Connect core points within distance ϵ
3. Find all connected components
4. Assign border points to closest connected component
5. Return all components as clusters

DBSCAN – Core, Border, Noise

- Could result two main clusters, and a noise cluster:



DBSCAN

- Requires Euclidean?

DBSCAN

- Requires Euclidean?
 - NO, does not use mean, or minimize squared error
- Requires a metric?

DBSCAN

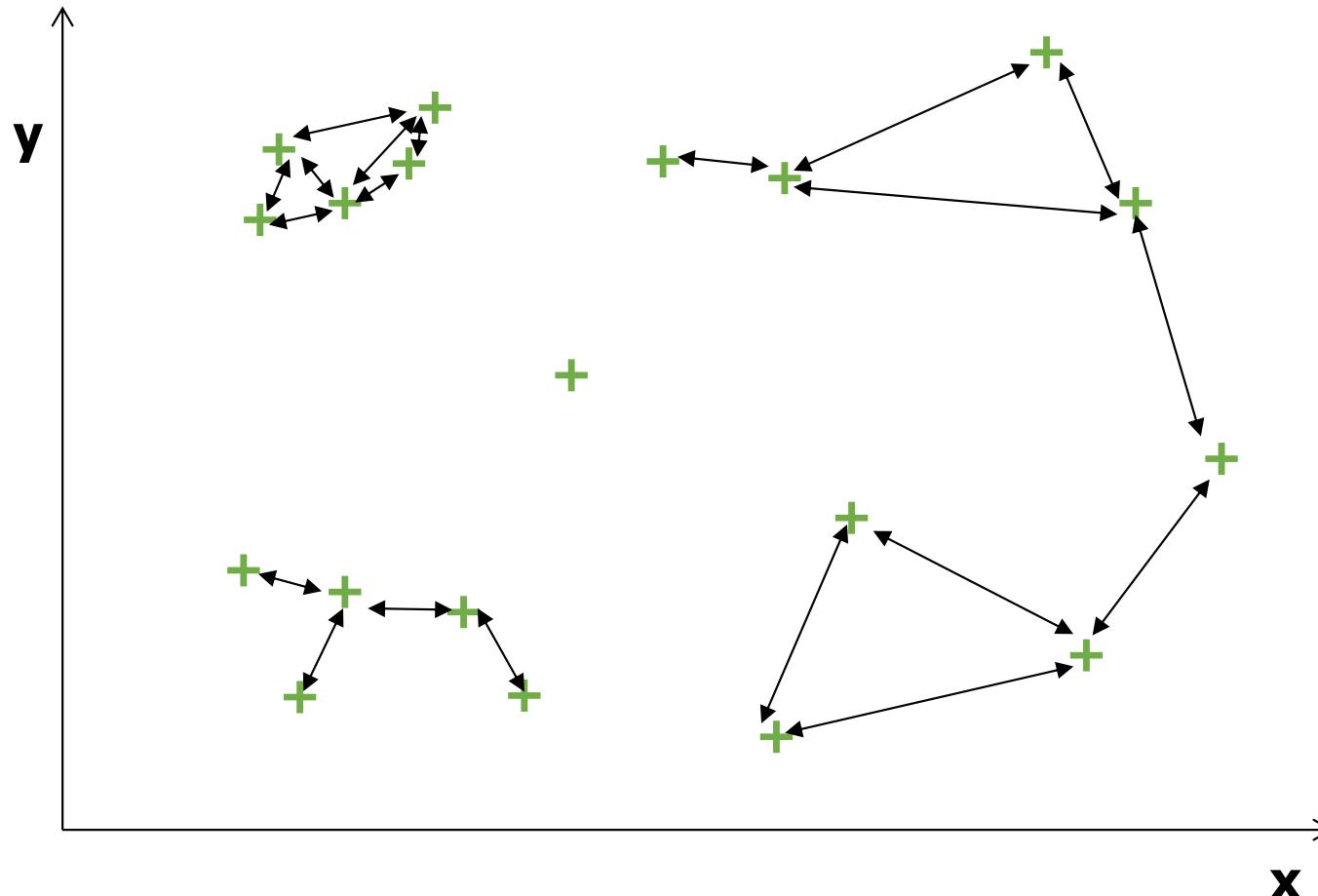
- Requires Euclidean?
 - NO, does not use mean, or minimize squared error
- Requires a metric?
 - Only computes paired distances between neighbors, so symmetry and non-negative are good to have

Graph-based (spectral) clustering

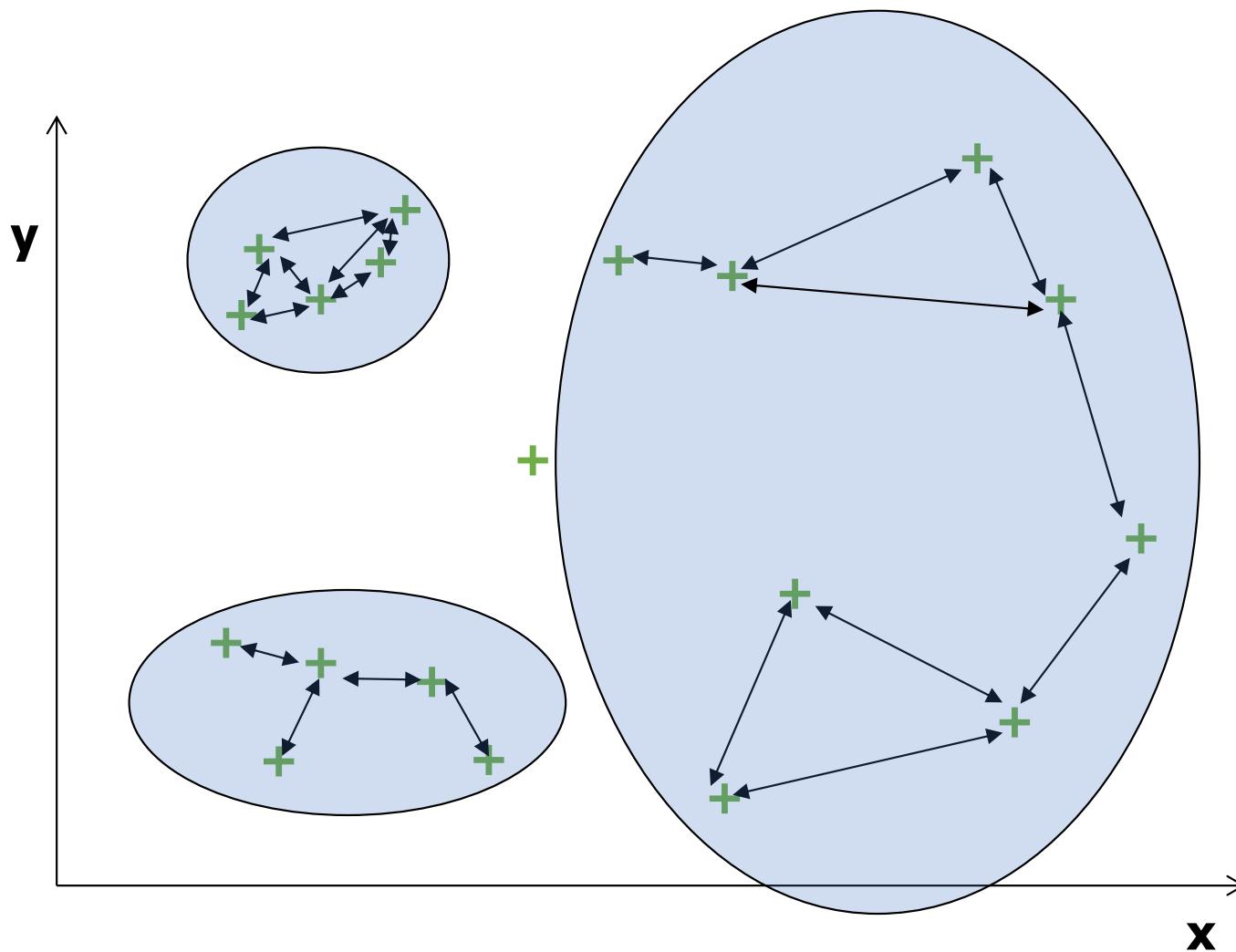
1. Construct neighborhood graph G , e.g.:
 - Connect points if their distance is below a threshold
 - Connect points if they are nearest neighbors of each other
 - ...
2. Set the weight on edges based on the point-wise distance
3. Find communities on G using graph mining (later lecture)
4. Return the found communities as clusters

Graph-based (spectral) clustering

- Showing pair-wise 3NN



Graph-based (spectral) clustering



Clustering: practical issues

- Should the observations or features first be standardized/normalized in some way?
This influences the distance!
- Cluster methods use the definition of a distance between observations
Which distance to use?
- How many clusters to choose?
 - Difficult problem. No agreed-upon method.
 - Can be determined by [visualizing](#) the cluster solution

Clustering: practical issues

- Should the observations or features first be standardized/normalized in some way?
This influences the distance!
- Cluster methods use the definition of a distance between observations
Which distance to use?
- How many clusters to choose?
 - Difficult to answer
 - Can be done by hand

What about Kruskal's clustering from
Algorithm Design?

Today

- Clustering
 - Recap on k-Means & hierarchical – *what distance is appropriate!*
 - Density – DBScan – *core, border, and noise points*
- Evaluation of unsupervised learning...
- Clustering large datasets
 - Batches
 - Prototyping and mini-clusters
 - Sufficient statistics

Evaluation of unsupervised ML

Evaluating anomaly detection is hard

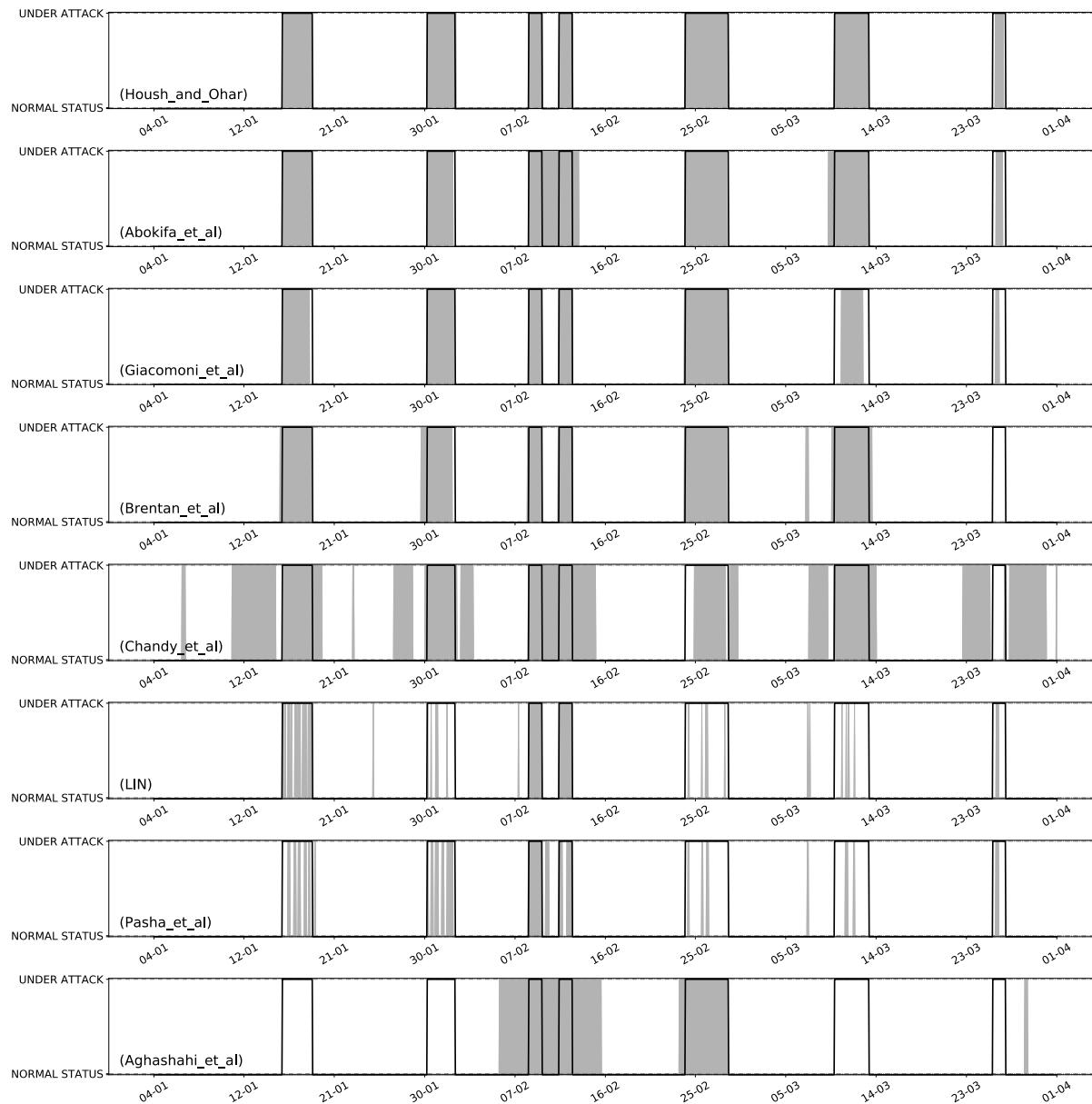
- *Often little/no information on positives*
 - Rely on quality of clustering, no clear quality measure exists
 - Unclear which distance to use
- *Anomalies are usually time periods instead of points*
 - An attack starts and stops
 - Is every detection within that period a true positive?
- *Unclear how to count positives*
 - Many alarms are raised in a few seconds, is this a single positive?
 - Should we group them over time?
 - ..
- For the challenge, we simply use point-based F1...

Evaluating anomaly detection is hard

- *Often little/no information on positives*
 - Rely on quality of clustering, no clear quality measure exists
 - Unclear which distance to use
- *Anomalies are usually time periods instead of points*
 - An attack starts and stops
 - Is every detection within that period a true positive?
- *Unclear how to count positives*
 - Many alarms
 - Should we count detections or clusters?
 - ..
 - For the cluster or for each point?

The only solution imho is to visualize the outcome and reason about its correctness/value

Measuring overlap in BATADAL



Clustering: Evaluation

- *is hard since we have no ground truth!*

Clustering: Evaluation

- *is hard since we have no ground truth!*
- for K-means, we can compute WCV $\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$
- for other methods? *there often is not even an objective!*

Clustering: Evaluation

- *is hard since we have no ground truth!*
- for K-means, we can compute WCV $\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$
- for other methods? *there often is not even an objective!*
 - *This is truly a data mining problem, we do not know what to optimize or how to optimize it, just that we want "insightful" or "sensible" clusters as end result*

Clustering: Evaluation

- *is hard since we have no ground truth!*
- for K-means, we can compute WCV $\underset{C_1, \dots, C_K}{\text{minimize}} \left\{ \sum_{k=1}^K \frac{1}{|C_k|} \sum_{i, i' \in C_k} \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right\}$
- for other methods? *there often is not even an objective!*
 - *This is truly a data mining problem, we do not know what to optimize or how to optimize it, just that we want "insightful" or "sensible" clusters as end result*
 - Still two main methods/heuristics for evaluation exist:
 1. Compare distances inside and outside clusters
 2. External validation

Intracluster and Intercluster (unsupervised)

- Let P be the set of point pairs from the same cluster
- Let Q be the set of point pairs from different clusters
- Then the average intra- and intercluster distances are:
 - Intra = $\text{sum}_{i,j \in P} \text{dist}(i,j) / |P|$
 - Inter = $\text{sum}_{i,j \in Q} \text{dist}(i,j) / |Q|$
- When ratio Intra/Inter is small, the clustering is considered good

Silhouette coefficient (unsupervised)

- Let C be the set of all clusters
- Let $i \in C$ be the cluster the point i is part of
- Then
 - $Davg_i^c = \text{avg } \{\text{dist}(i,j) : j \text{ is in cluster } c\}$
 - $Davg_i^{in} = Davg_i^c$ with i is in cluster in
 - $Dmin_i^{out} = \min_{c \in C, c \neq in} Davg_i^c$
- The silhouette coefficient S_i for point i is:

$$S_i = \frac{Dmin_i^{out} - Davg_i^{in}}{\max\{Dmin_i^{out}, Davg_i^{in}\}}$$

- This gives a value in $(-1,1)$ with large positive values indicating a good clustering

Silhouette coefficient (unsupervised)

- Let C be the set of all clusters
- Let $i \in C$ be the cluster the point i is part of
- Then
 - $Davg_i^c = \text{avg} \{ \text{dist}(i,j) : j \text{ is in cluster } c \}$
 - $Davg_i^{in} = Davg_i^c$ with i is in cluster in //average distance to i in in
 - $Dmin_i^{out} = \min_{c \in C, c \neq in} Davg_i^c$ //smallest average distance to i not in in
- The silhouette coefficient S_i for point i is:

$$S_i = \frac{Dmin_i^{out} - Davg_i^{in}}{\max\{Dmin_i^{out}, Davg_i^{in}\}}$$

- This gives a value in $(-1,1)$ with large positive values indicating a good clustering

Silhouette coefficient (unsupervised)

- Let C be the set of all clusters
- Let $i \in C$ be the cluster the point i is part of
- Then
 - $D_{avg_i^c} = \text{avg } \{\text{dist}(i,j) : j \text{ is in cluster } c\}$
 - $D_{avg_i^{in}} = D_{avg_i^c}$ with i is in cluster i
 - $D_{min_i^{out}} = \min_{c \in C, c \neq in} D_{avg_i^c}$
- The silhouette coefficient S_i for point i is:

$$S_i = \frac{D_{min_i^{out}} - D_{avg_i^{in}}}{D_{min_i^{out}} + D_{avg_i^{in}}}$$

These metrics kind of work for circular clusters, how to measure non-circular cluster quality is an open problem...

- This gives positive and negative

An option: external validation

- When some class labels are available, or ground truths, we could construct a confusion matrix:

Cluster\Class	A	B	C
A	5	50	10
B	70	10	20
C	4	8	22

An option: external validation

- When some class labels are available, or ground truths, we could construct a confusion matrix:

Cluster\Class	A	B	C
A	5	50	10
B	70	10	20
C	4	8	22

- We then do not really care whether A is assigned label A, but whether points from class A are all assigned to the same cluster
- In this case, 70 out of 79 are.

An option: external validation

- When some class labels are available, or ground truths, we could construct a confusion matrix:

Cluster\Class	A	B	C
A	5	50	10
B	70	10	20
C	4	8	22

- We then whether
- In this case

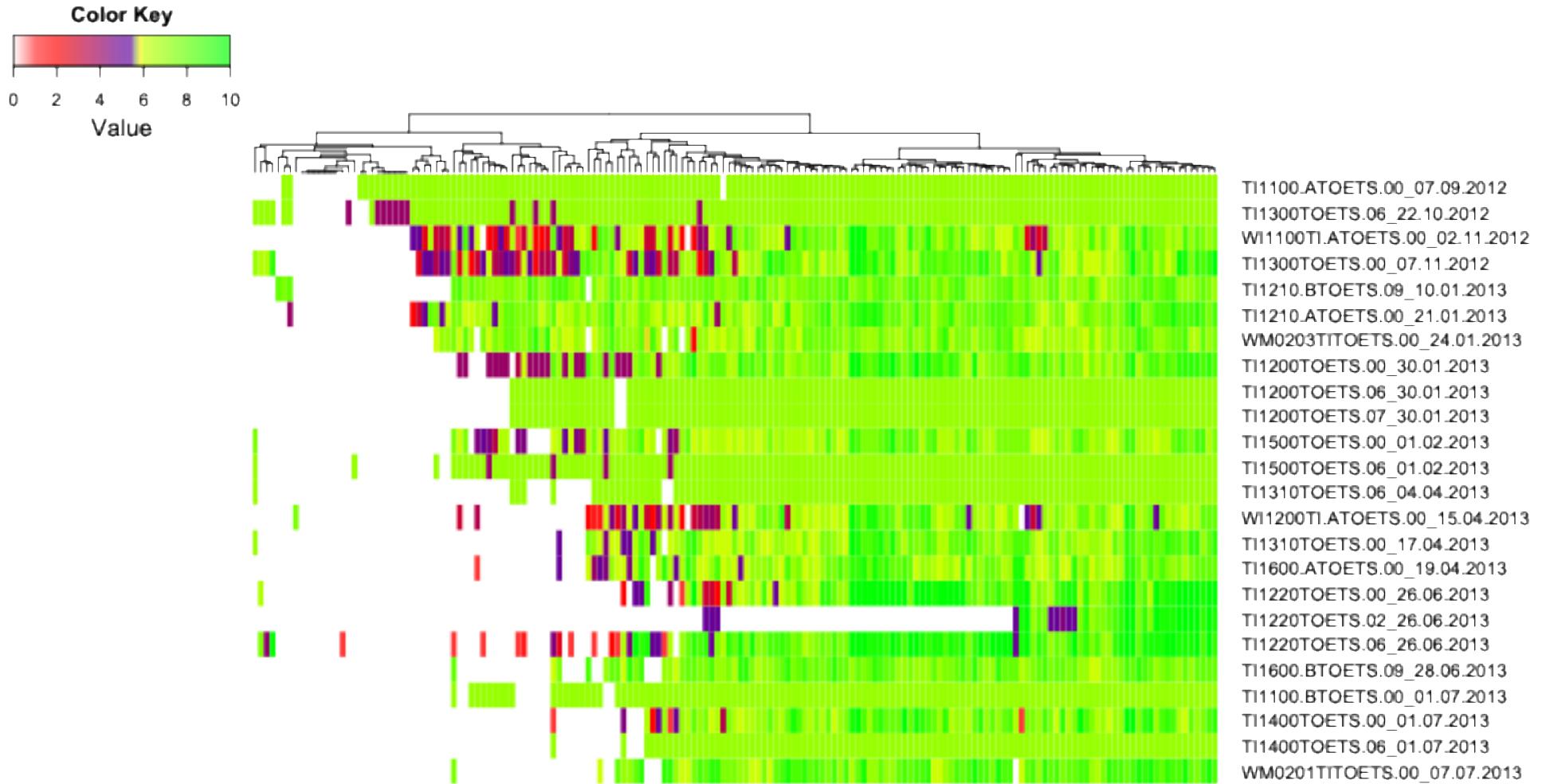
Versions of F1 score, purity, entropy also exist, requiring that points from the same class belong to the same cluster

My method: visualize!

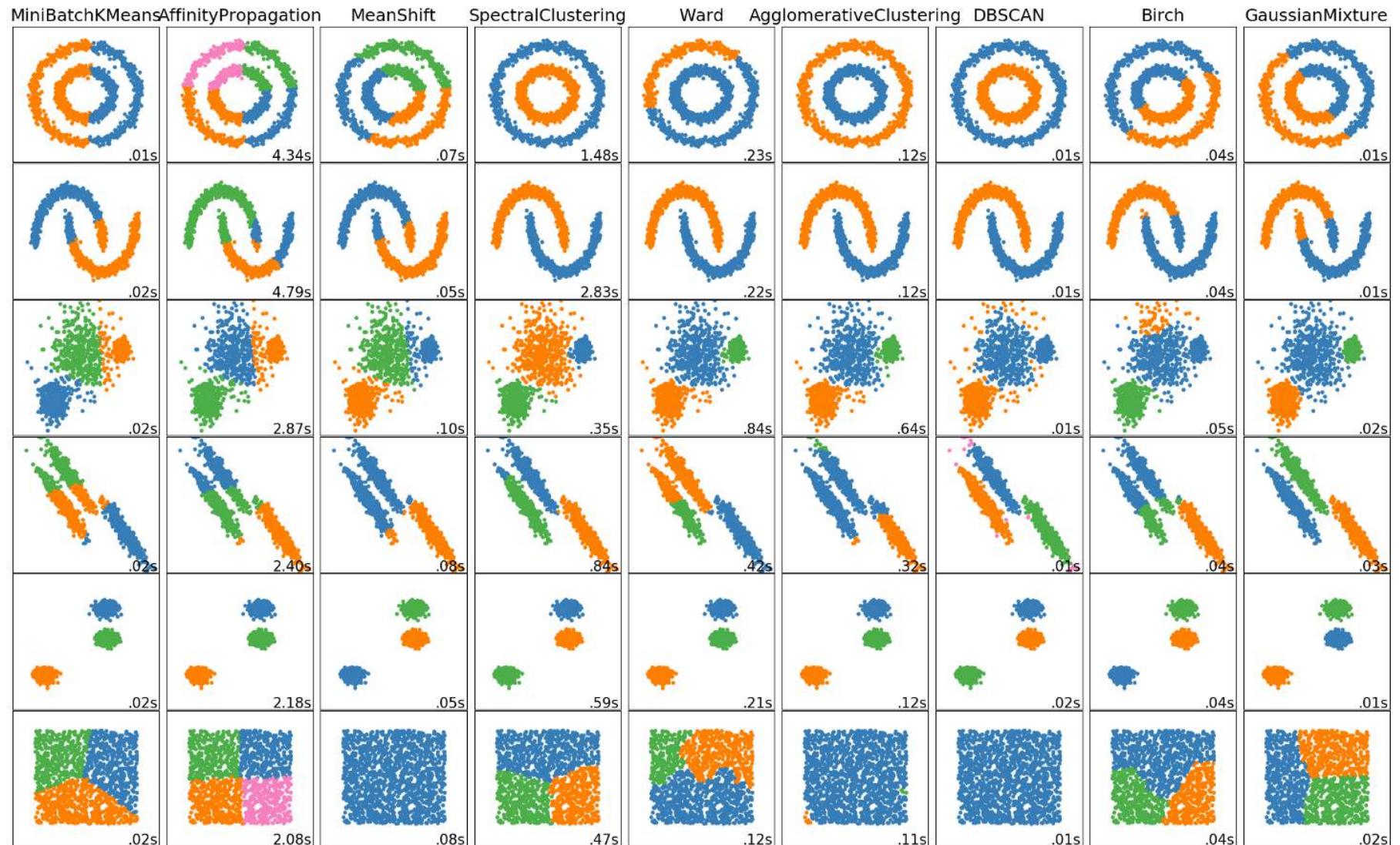


Row = connection
Column = time
Cell = Bytes transferred

My method: visualize!

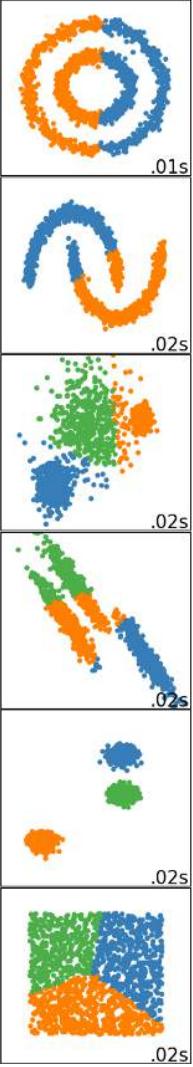


Many clustering methods...

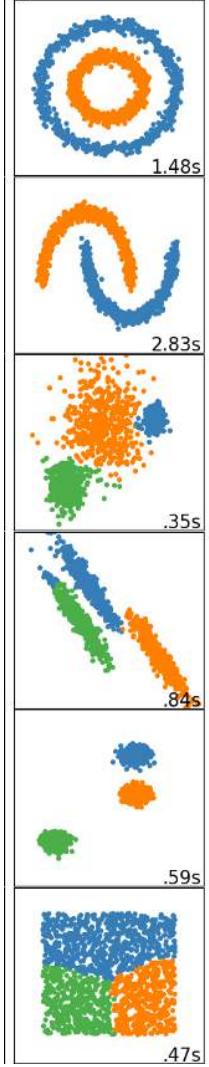


Many clustering methods...

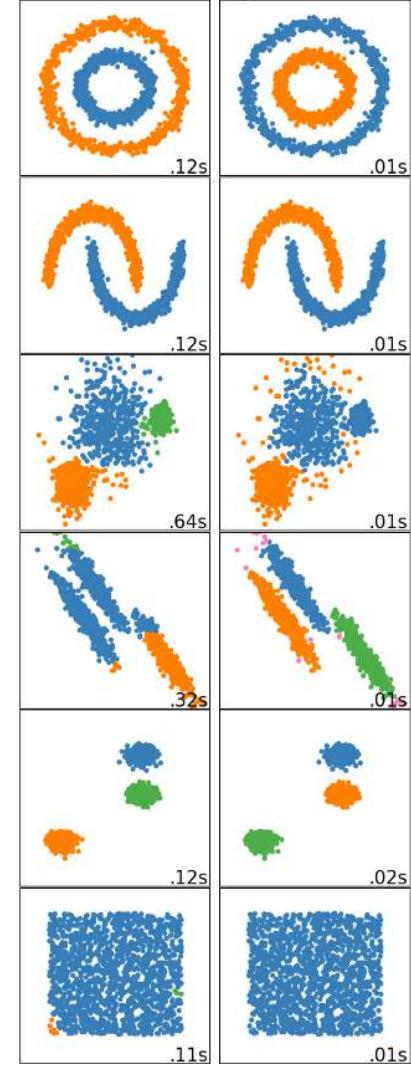
MiniBatchKMeans



SpectralClustering

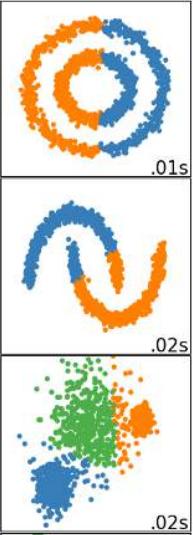


AgglomerativeClustering DBSCAN

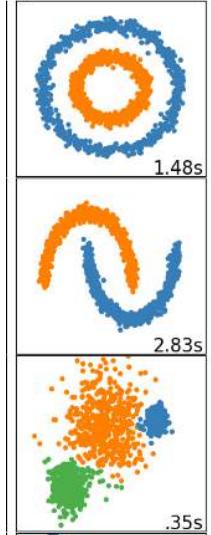


Many clustering methods...

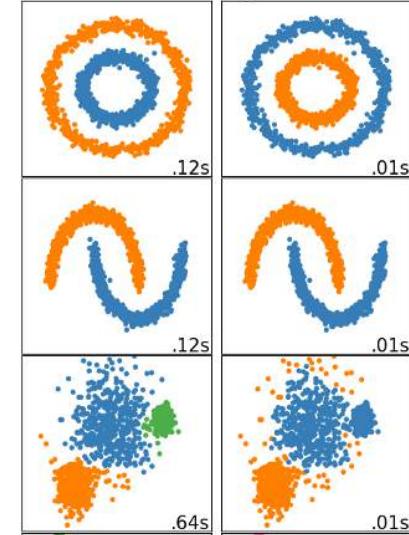
MiniBatchKMeans



SpectralClustering



AgglomerativeClustering DBSCAN



You should know which method can represent what type(s) of data

Clustering

- Clustering is an **optimization problem**, but often without an objective...
- Typical algorithms:
 - Expectation Maximization (K-means)
 - Bottom-up/top-down learning (hierarchical)
 - Density-based, graph-based (DBScan, Spectral)
- Which **distance** you use matters
- *Make sure you understand the obtained clustering*
- *What is the goal of clustering? To get a small silhouette coefficient? Or to understand your data?*

Today

- Clustering
 - Recap on k-Means & hierarchical – *what distance is appropriate!*
 - Density – DBScan – *core, border, and noise points*
- Evaluation of unsupervised learning... *is hard!*
- Clustering large datasets
 - Batches
 - Prototyping and mini-clusters
 - Sufficient statistics

Clustering large datasets

Issues with clustering

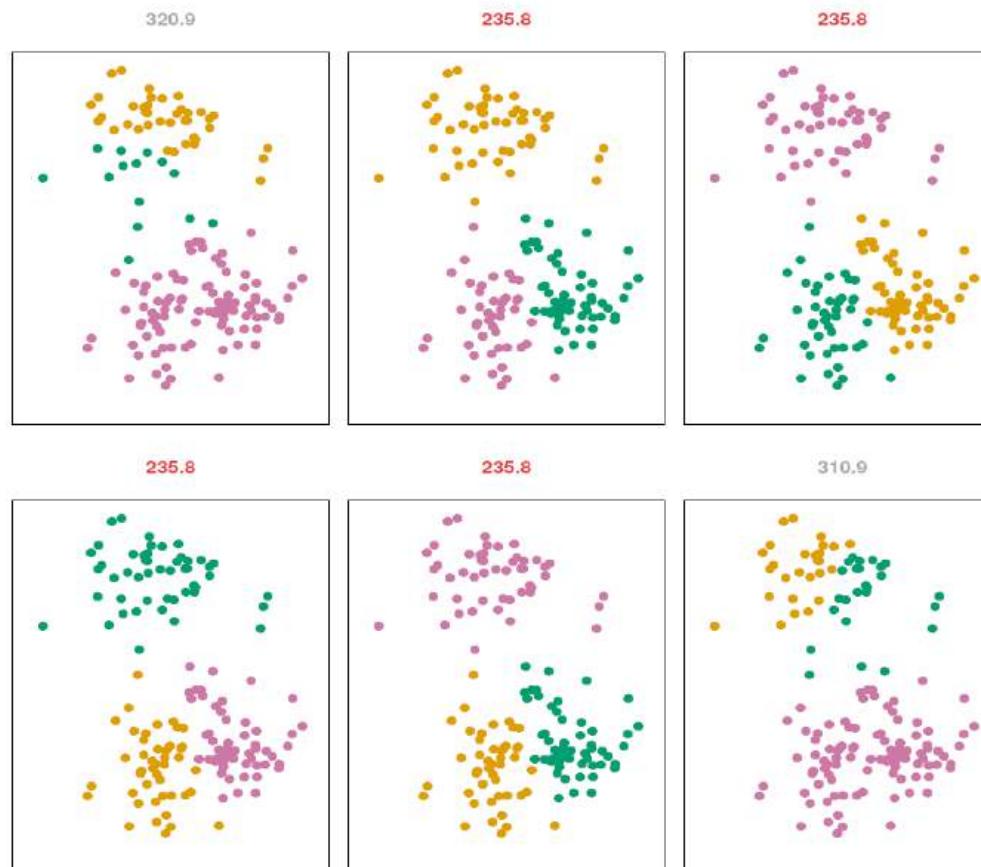
- Centroid-based
 - Gets stuck in local minima
 - Initialization can be problematic
 - How to handle streaming data?
 - Cannot handle non-spherical clusters
- Hierarchical and Density/Graph-based
 - Required computing distance matrix which is very expensive..
 - See lecture of locality sensitive hashing for solutions!
 - Streaming data?

Streaming data

- Continuous and rapid input of data
 - Network traffic
 - Audio/video
 - Sensor readings
 - ...
- Goal is to learn in real-time on-the-fly!
- Problems:
 - Limited memory to store the data (less than linear in the input size)
 - Limited time to process each element
 - Sequential access (no random access)
 - Algorithms have one ($p=1$) or very few passes ($p=\{2,3\}$) over the data

KMeans gets stuck in local minima

- Perform K-means clustering algorithm 6 times,
 - with different random starting assignments



How to fix KMeans?

k-means++

- Repeat and average or keep the best
- Select different starting points, how?

k-means++

1. Assign first centroid c to point uniformly at random
2. Add c to C
3. For each datapoint $x \in X$
 - Compute $d(x) = \min_{c \in C} \text{distance}(x, c)$
4. Assign next centroid c to a random point
 - with probability proportional to $d(x)^2$, e.g., $P(x) = d(x)^2 / \sum_y d(y)^2$
5. If less than k centroids are assigned, goto 3
6. Else, proceed with standard k-means

k-means++

1. Assign first centroid c to point uniformly at random
2. Add c to C
3. For each datapoint $x \in X$
 - Compute $d(x) = \min_{c \in C} \text{distance}(x, c)$
4. Assign next centroid c to a random point
 - with probability proportional to $d(x)^2$, e.g., $P(x) = d(x)^2 / \sum_y d(y)^2$

The book mentions to select the points with largest distance, a proportional probability is better because it avoids selecting outliers as centroids

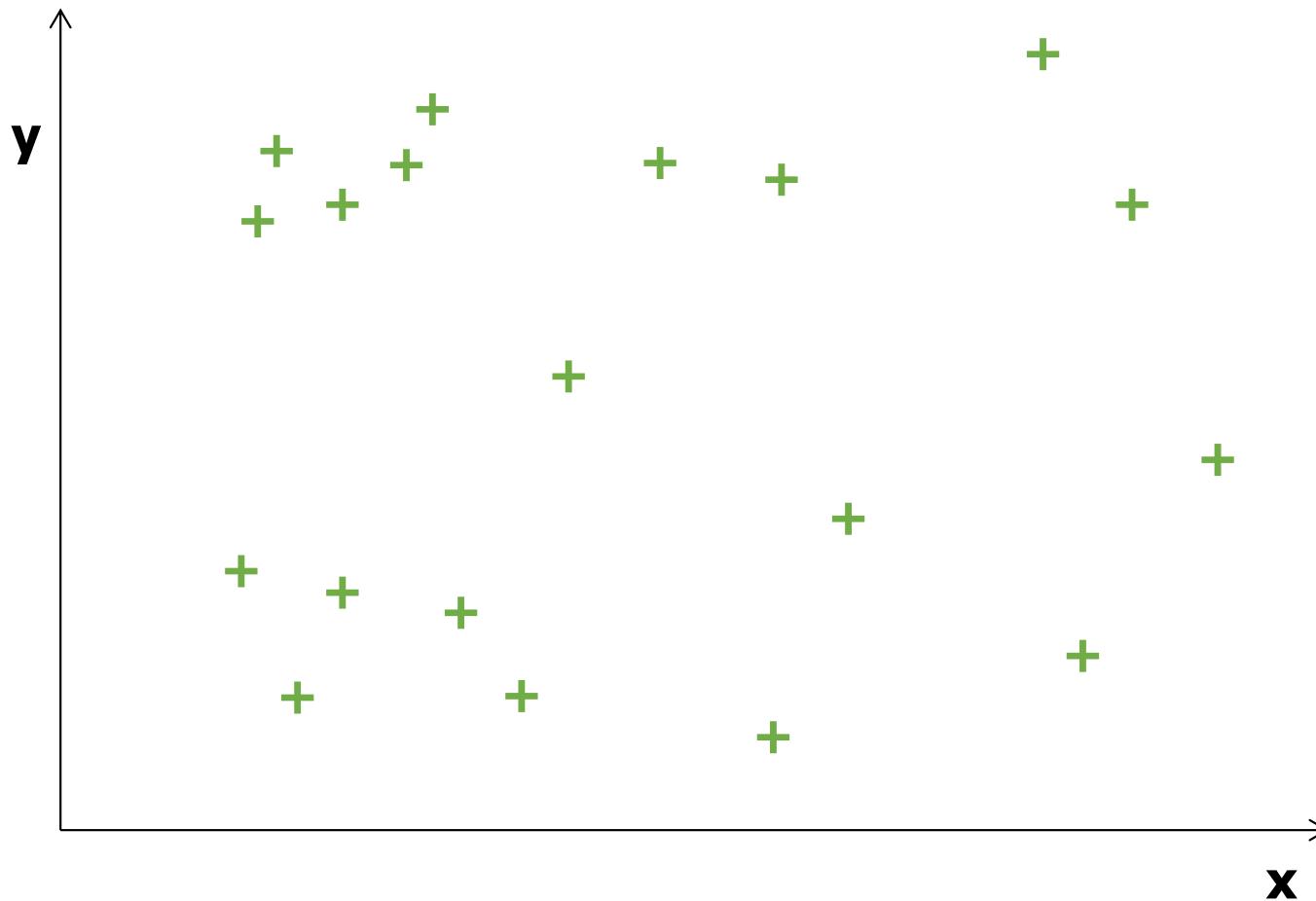
Batching: minibatch k-means

- Observation:
 - EM/gradient descent like methods can update using subsamples!
- Thus:
 1. Read n rows of data D (at random)
 2. Perform one update of centroids using D
 3. Goto 1, repeat until convergence

CURE – cluster using representatives (prototypes)

- Observation: non-centroid clustering is expensive
- Solution: run expensive operations on a small sample!
- Using representatives is more commonly known as **prototyping**:
 - Selecting a few datapoints to represent a cluster or group of points

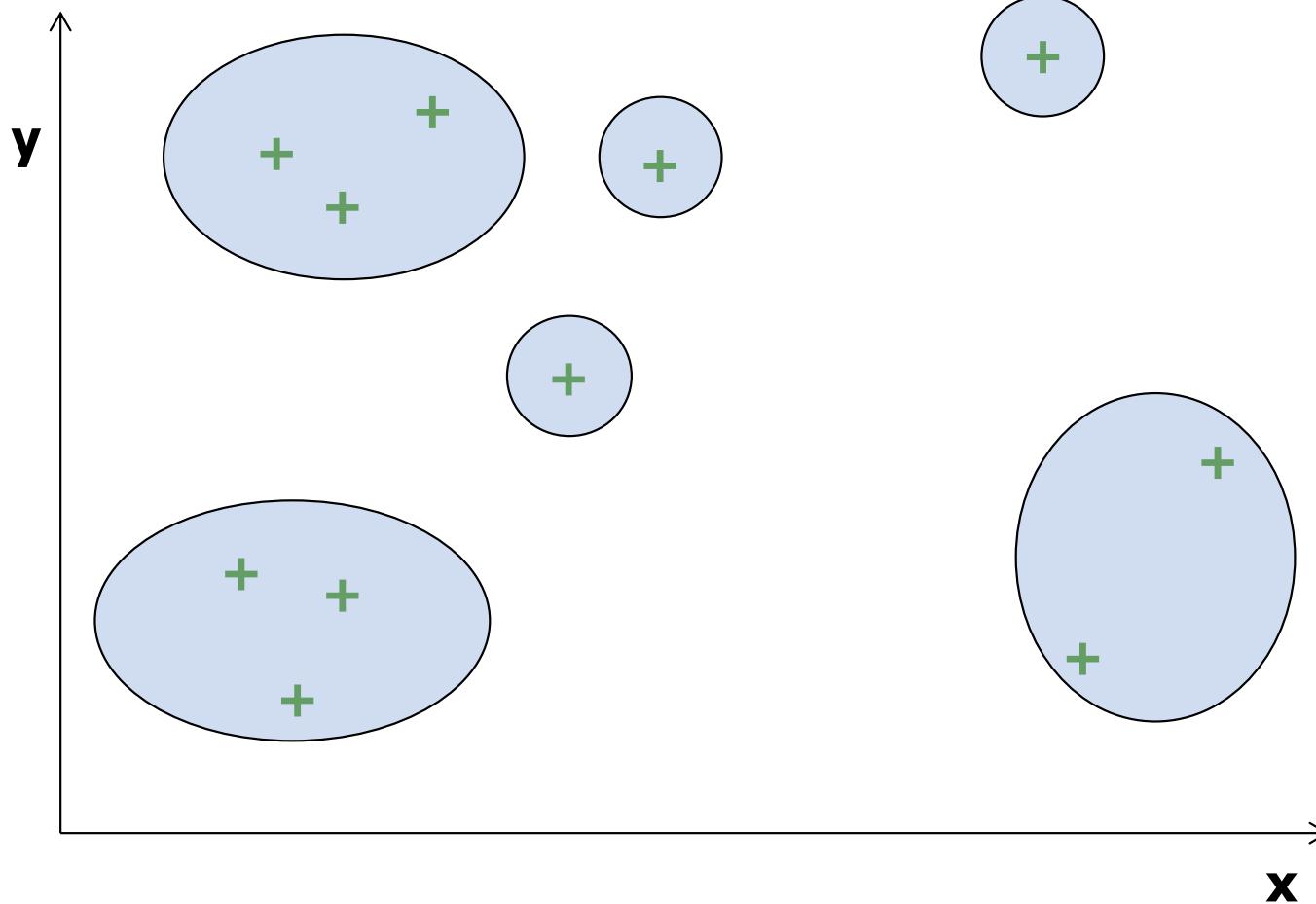
CURE



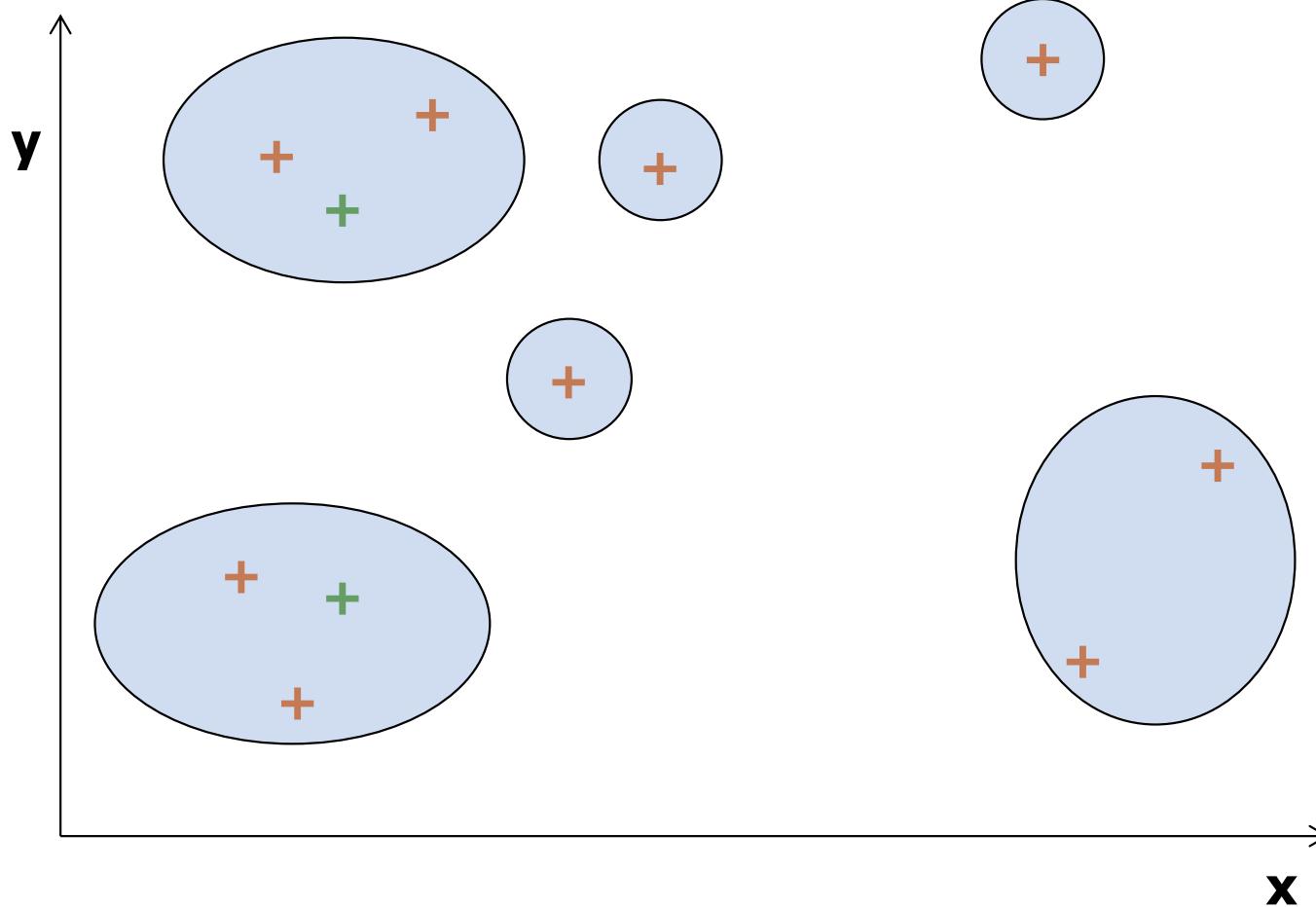
CURE – small batch – use DBSCAN



CURE – small batch – use DBSCAN



CURE – small batch – prototypes



CURE – cluster using representatives

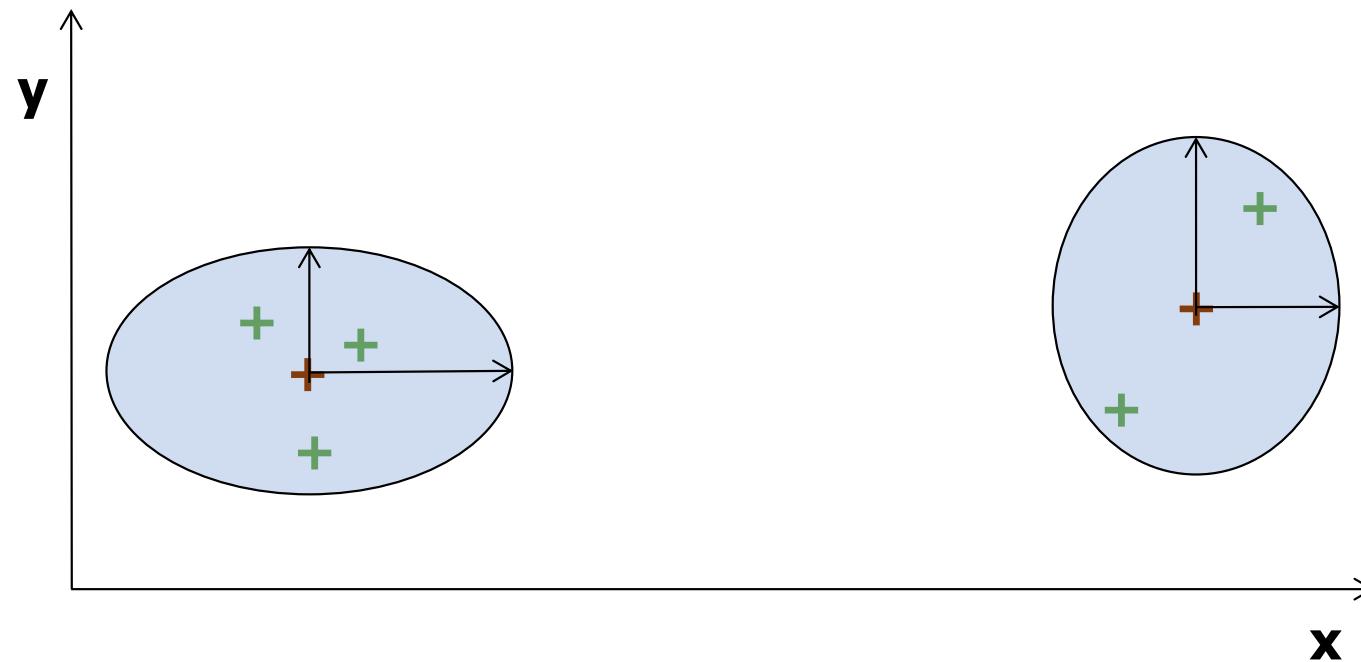
1. Learn from a sample
 - run any expensive clustering algorithm on this batch
2. Select a small set of representative points for each cluster
 - These points should be as far apart from each other, e.g. kmeans++
3. Maintain **mini-clusters** that will later be merged to form larger ones
 - merge clusters that are close to each other
 - use the representatives for speed!

BFR – (Bradley, Fayyad, Reina)

- Only store what is needed
- Points can be assigned to three sets:
 - Discard set – assigned to a pre-existing cluster
 - Compressed set – mini-clusters (a clustering of remaining points)
 - Retained set – outliers, points that do not belong to any cluster
- Only the retained set is stored in memory, for the other sets, we maintain *sufficient statistics*

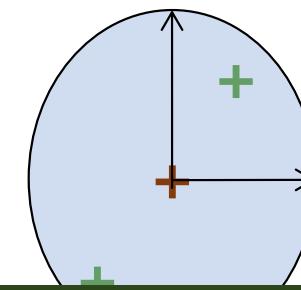
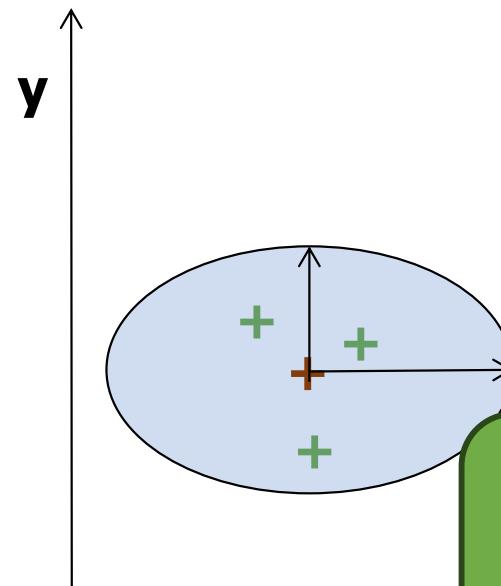
BFR - statistics

- A cluster is represented using a:
 - its mean and
 - a variance per feature (assumed to be independent)



BFR - statistics

- A cluster is represented using a:
 - its mean and
 - a variance per feature (assumed to be independent)



What do we need to count to maintain this information?

BFR - statistics

- For a cluster in BFR, we only require the following
 1. N – the number of data points in the cluster
 2. SUM – a vector containing with sums of all feature values
 3. SUMSQ – a vector with sums of squared feature values
- Using these we can
 1. Keep track of the centroid (mean) of the cluster
 2. Compute a threshold for cluster membership

BFR - statistics

How?

- For a set of data:

$$\text{mean} = \text{SUM} / N$$

1.

$$\text{variance} = \text{sum}(x - \text{mean})^2 / N$$

2.

$$= \text{sum}(x^2 - 2\text{mean} \cdot x + \text{mean}^2) / N$$

3.

$$= \text{sum}(X^2)/N - 2\text{mean} \cdot \text{sum}(x)/N + N \cdot \text{mean}^2/N$$

- Using the formula for variance:

$$= \text{sum}(X^2)/N - 2\text{mean}^2 + \text{mean}^2$$

1.

$$= \text{sum}(x^2)/N - \text{mean}^2$$

2.

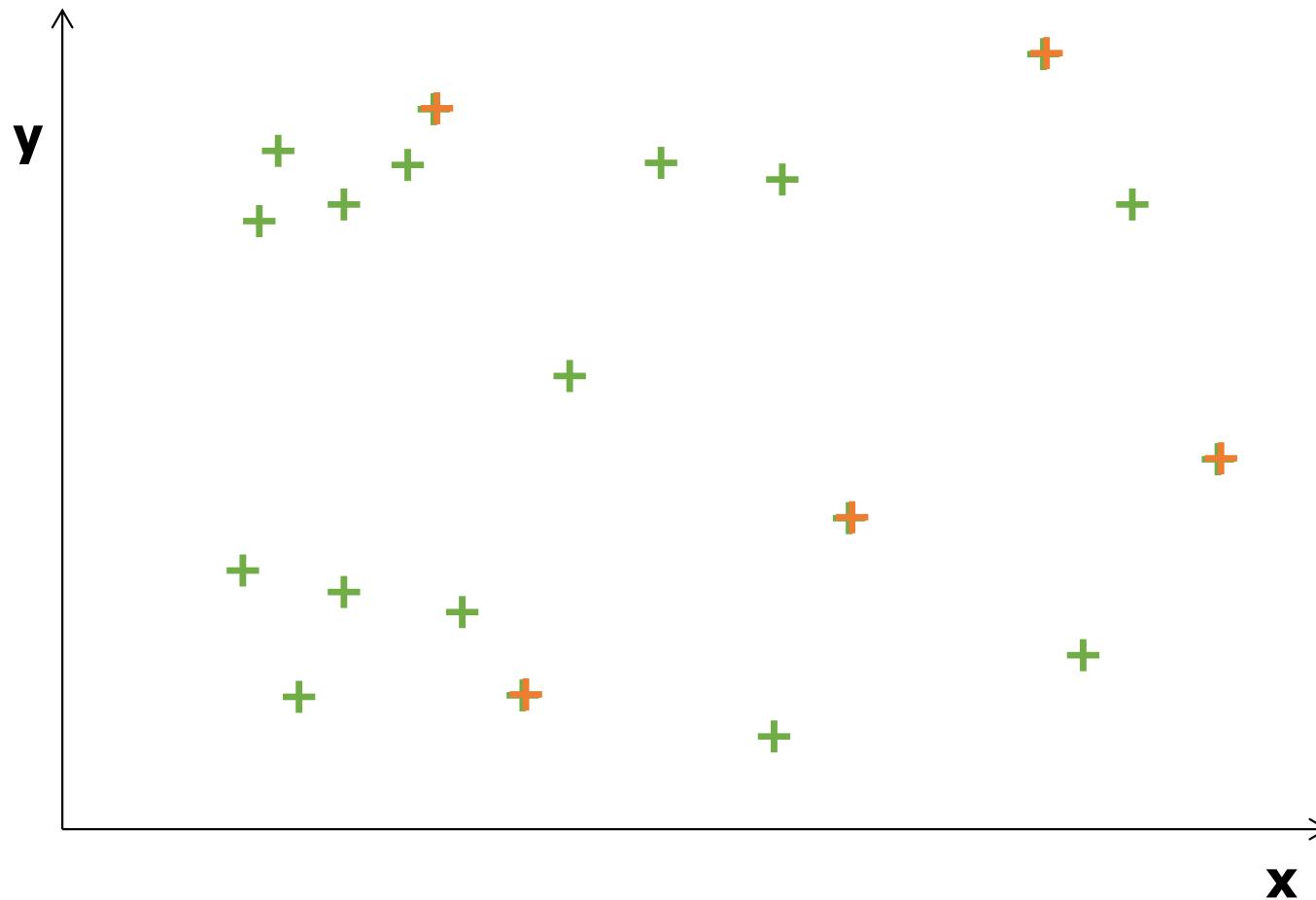
$$= \text{SUMSQ}/N - (\text{SUM}/N)^2$$

$$\text{std dev.} = \sqrt(\text{variance})$$

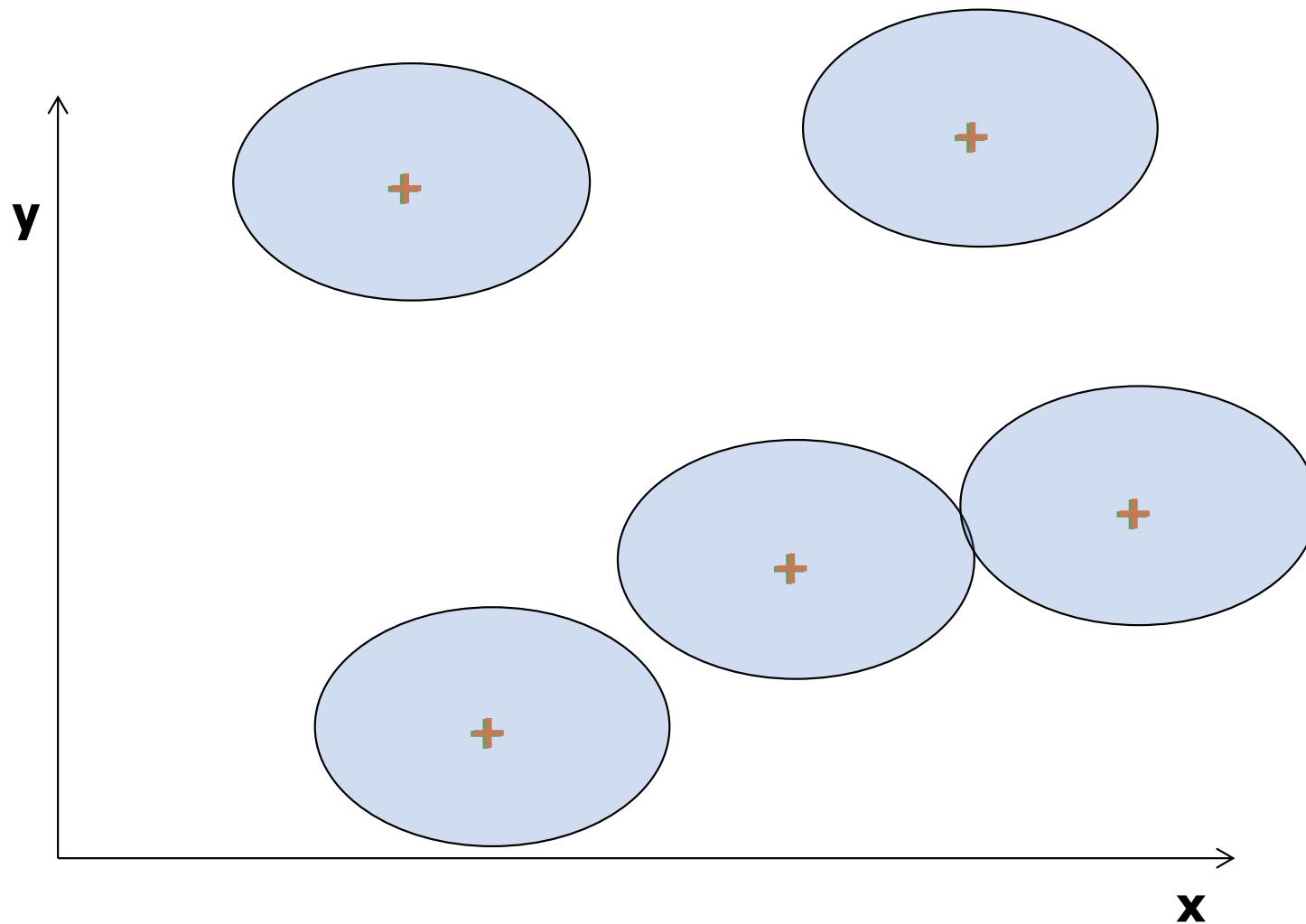
BFR - clustering

- For each batch of points:
 - If a point is in discard set, update statistics and discard
 - For remaining points
 - Use standard clustering to obtain **miniclusters**
 - Compute **statistics** for miniclusters, discard points
 - Keep unclustered points in retained set
 - Merge miniclusters and retained points with older miniclusters, update their statistics

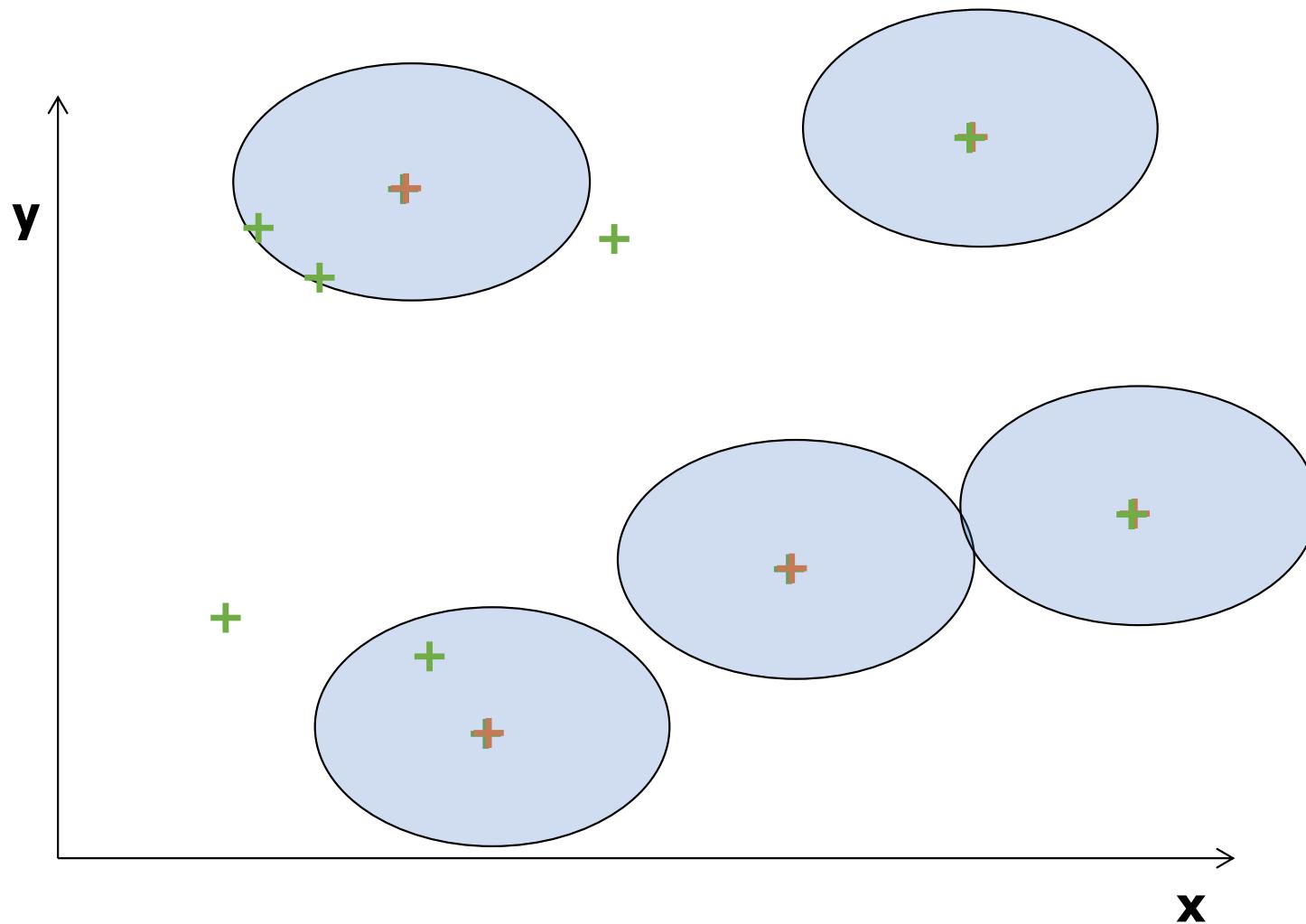
BFR – select k points



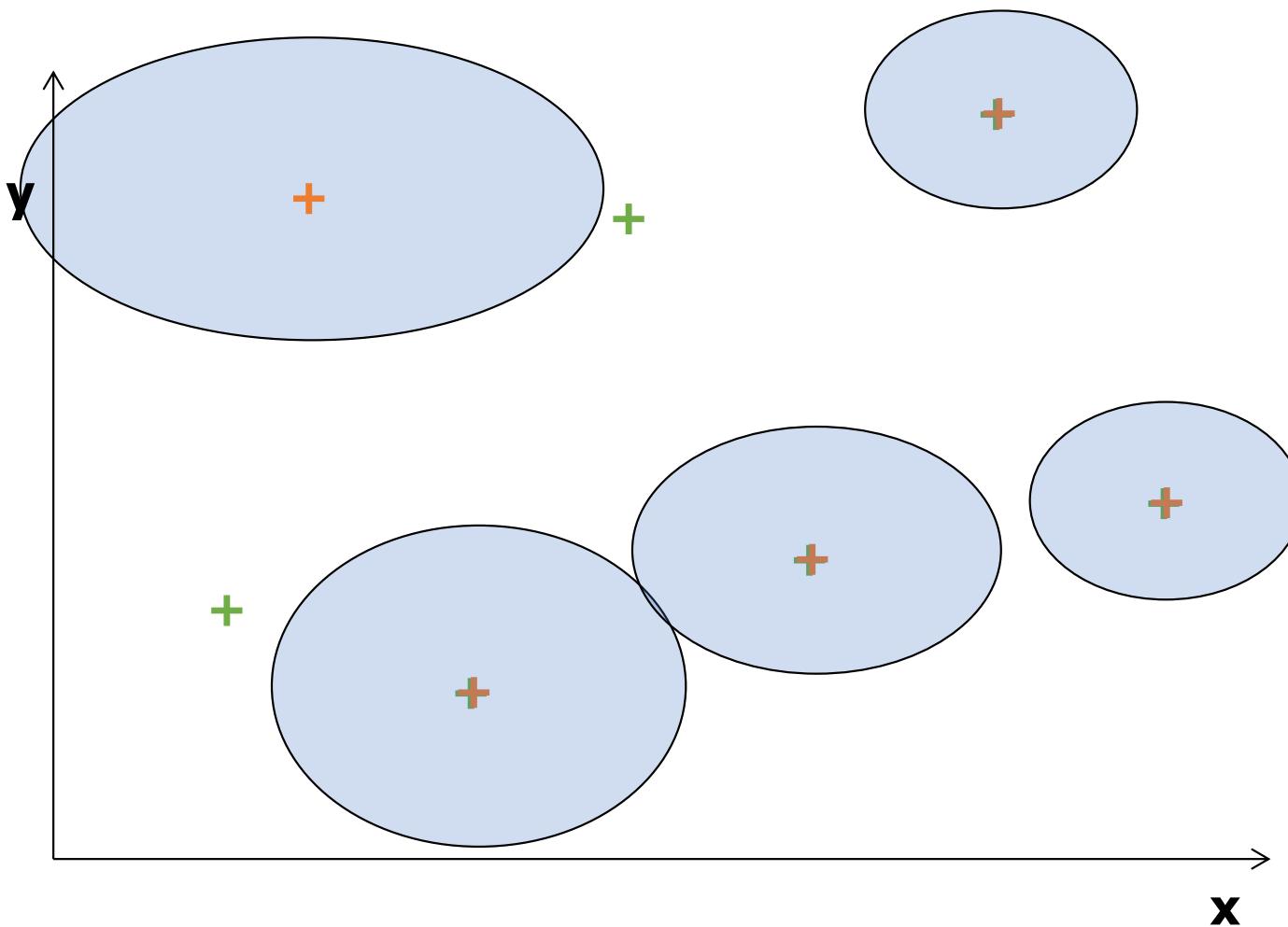
BFR – initialize clusters



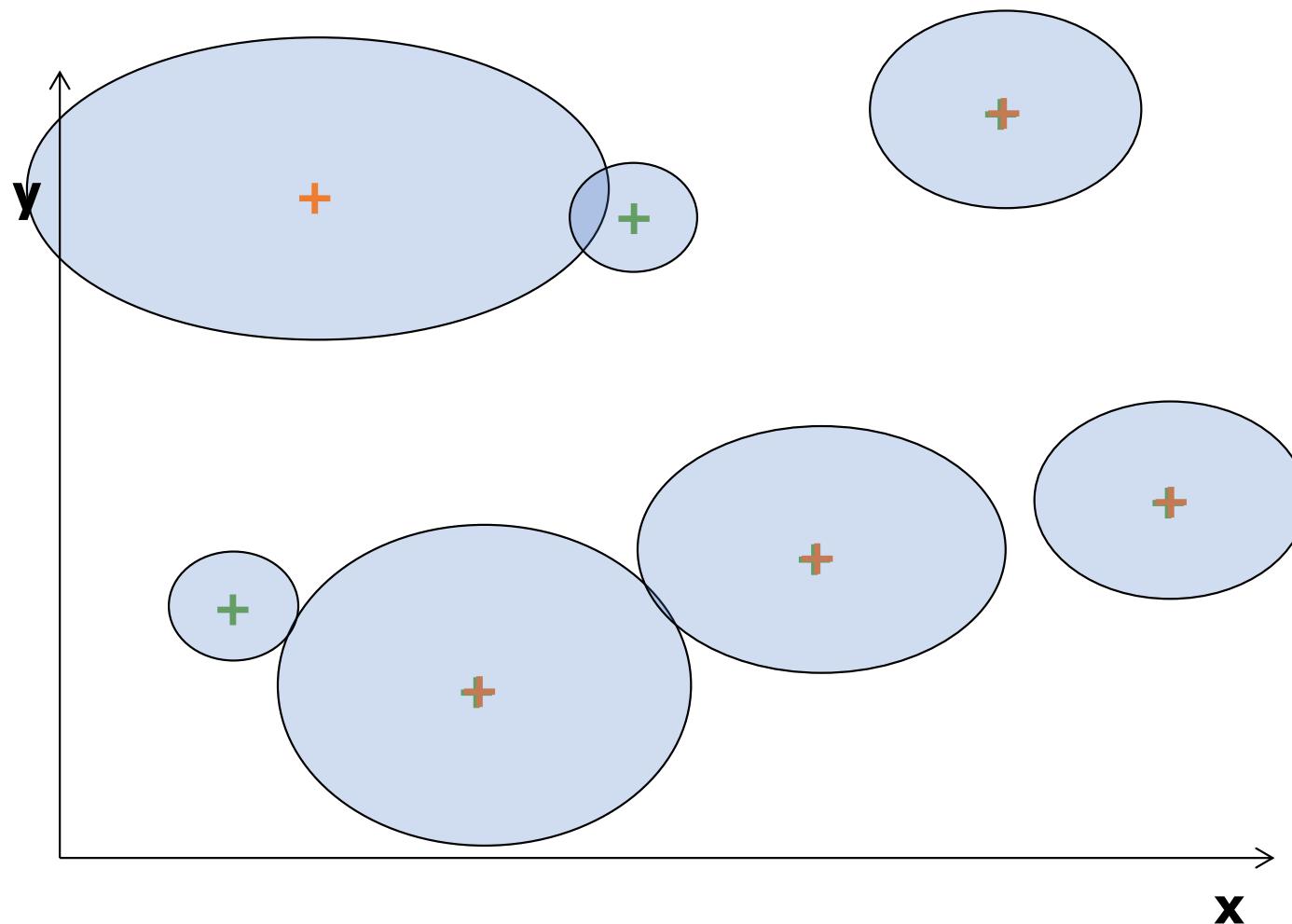
BFR – read batch



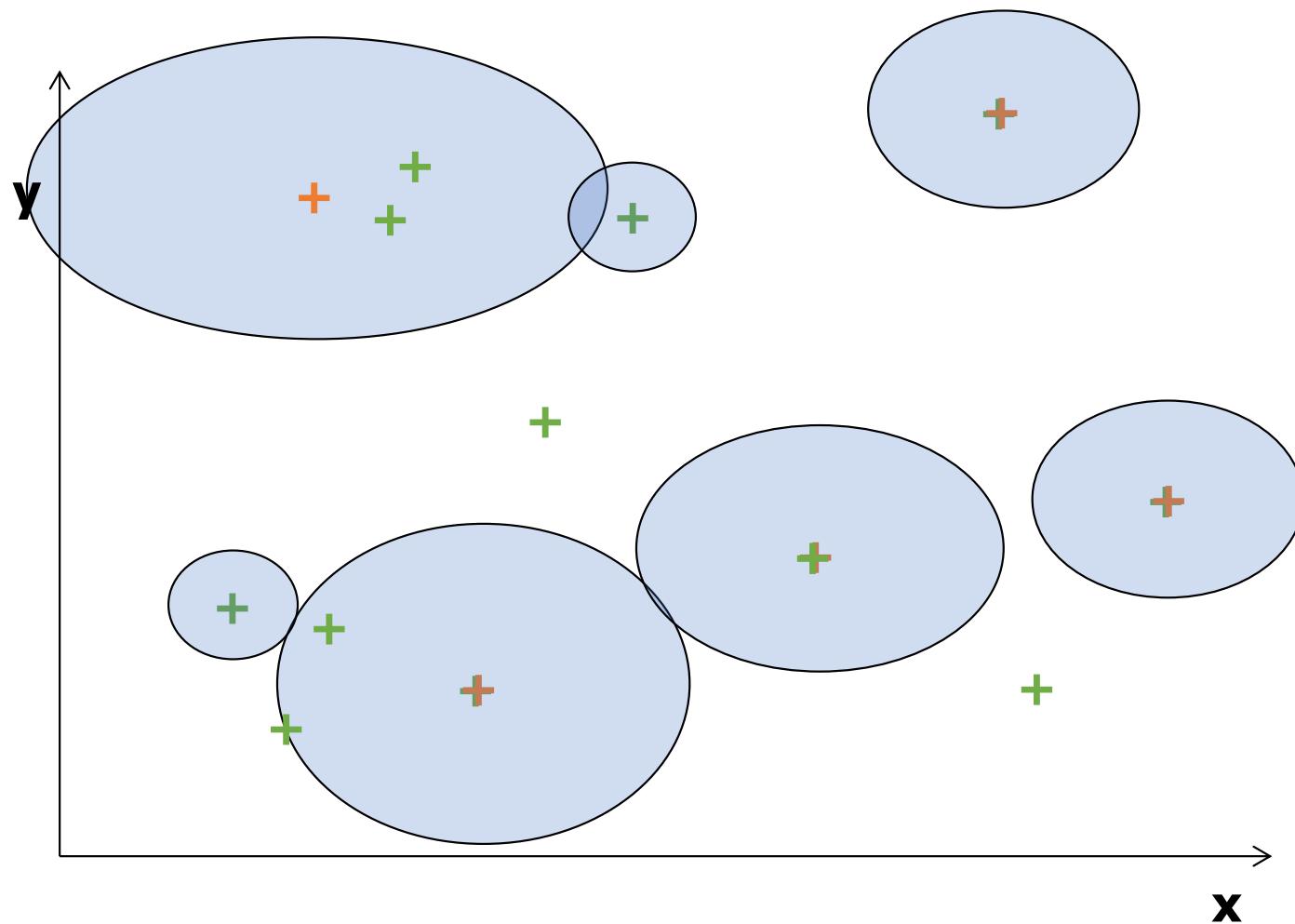
BFR – discard sufficiently close points, update mean & thresholds



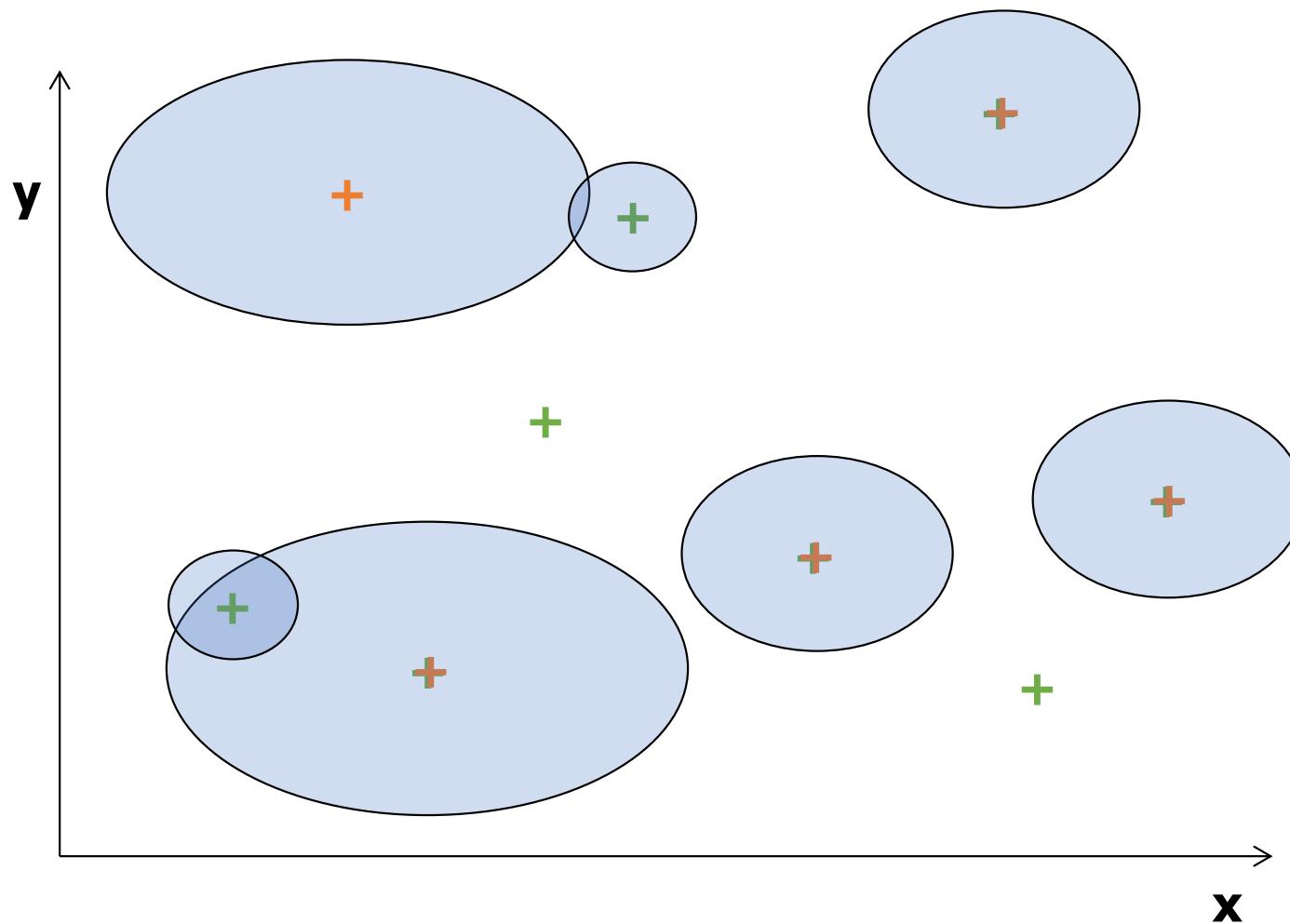
BFR – retain far away points, cluster them, using any method



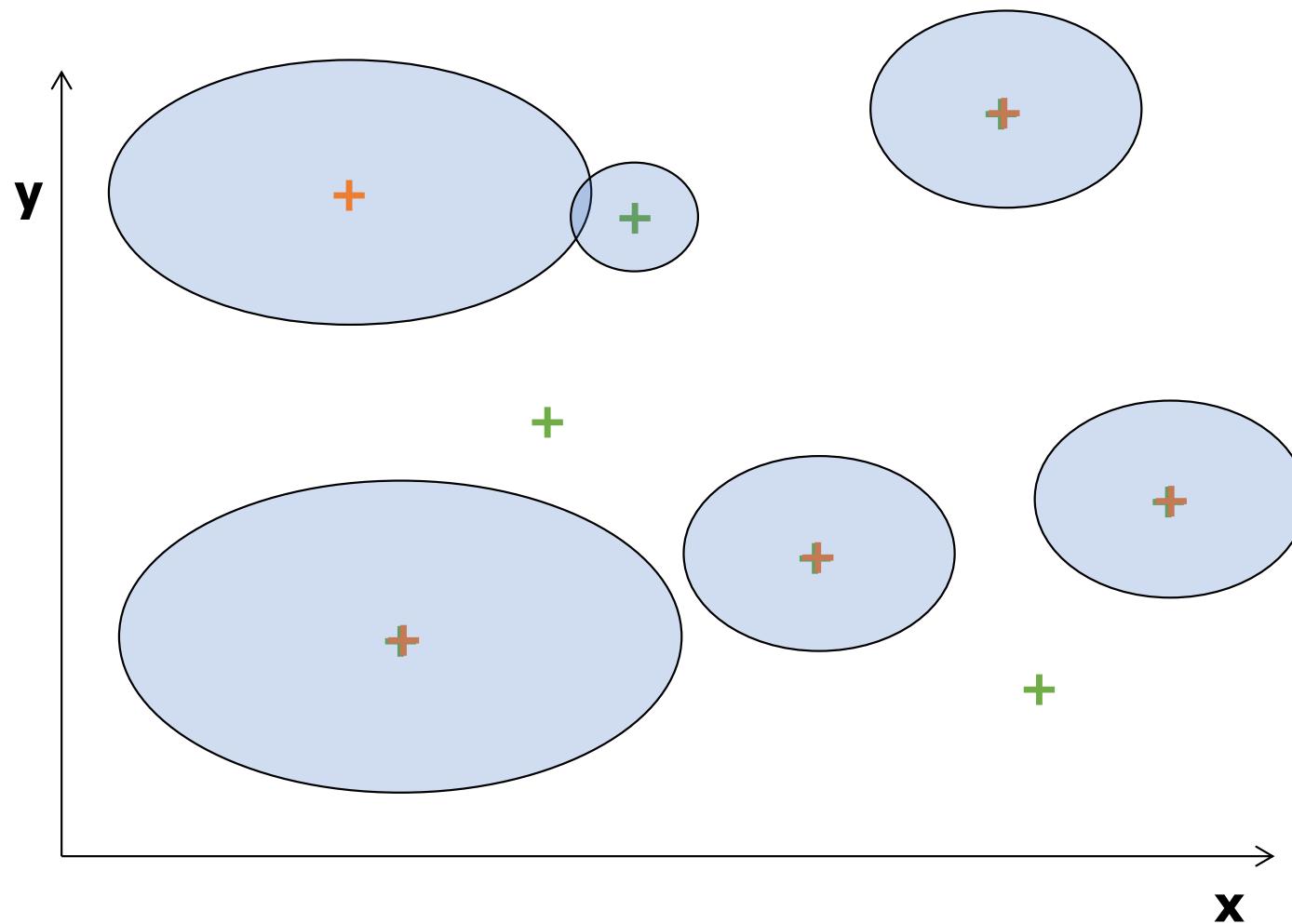
BFR – 2nd batch



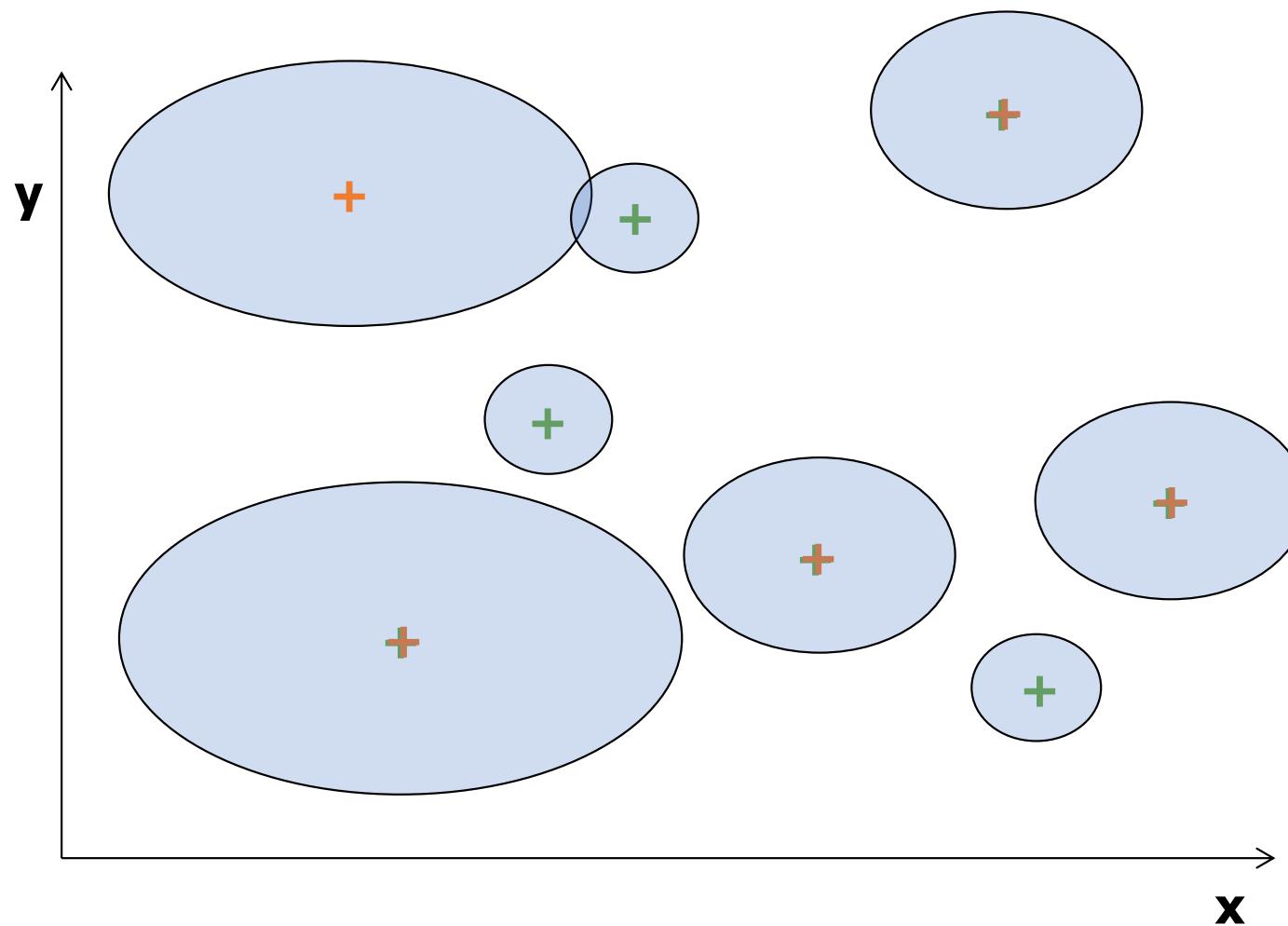
BFR – 2nd batch, discard and update



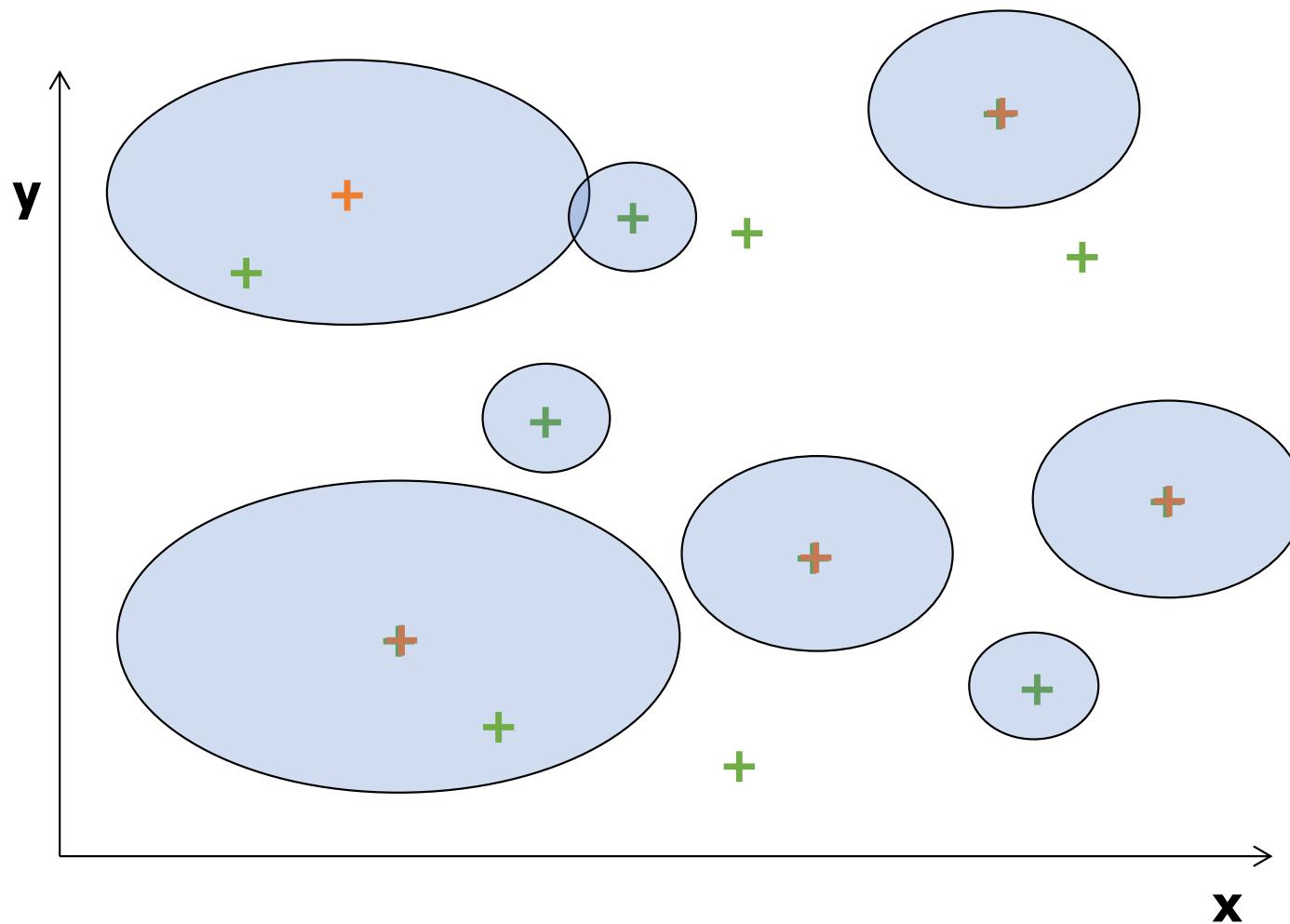
BFR – 2nd batch, discard and update – also miniclusters!



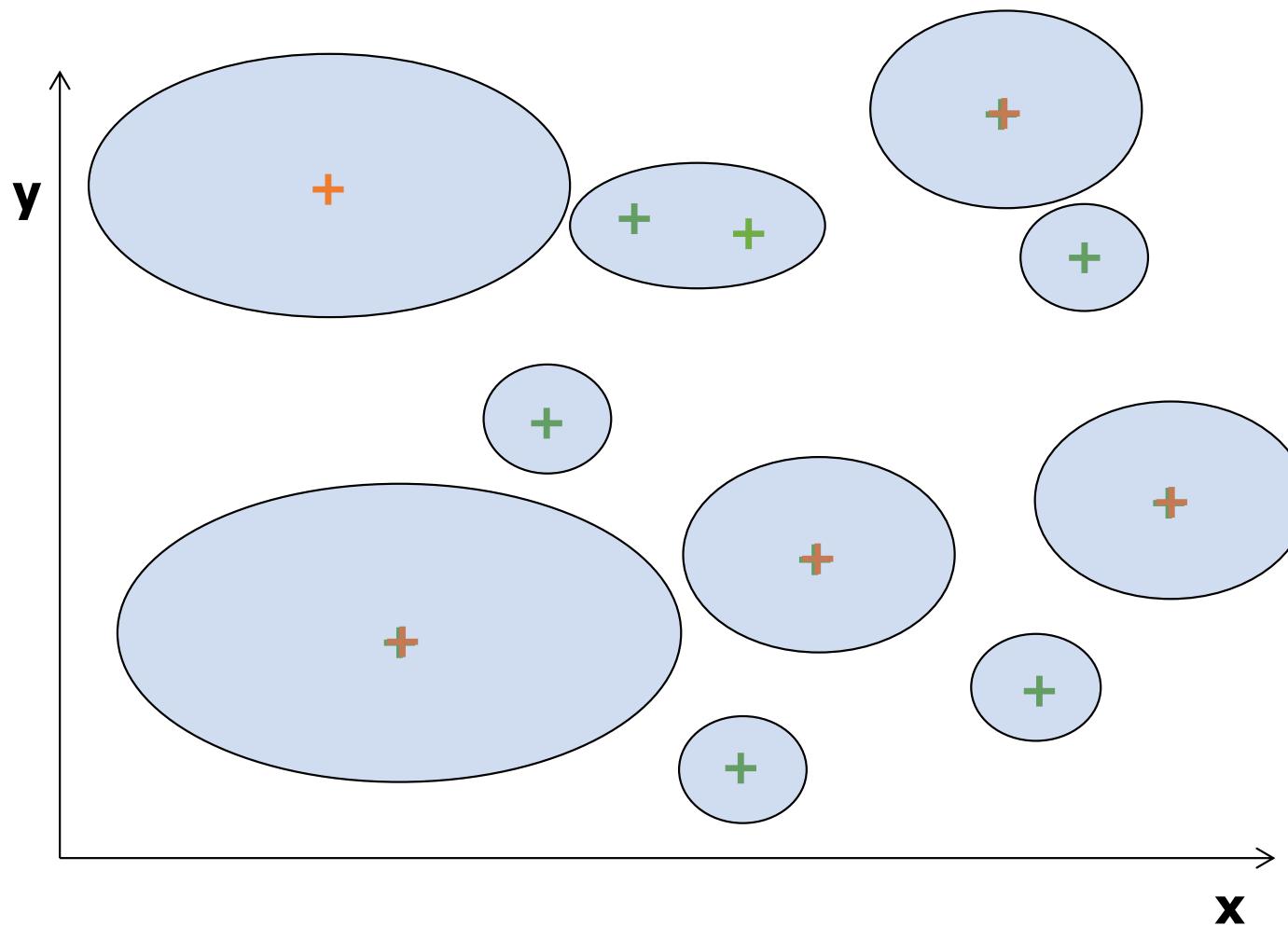
BFR – 2nd batch, create new miniclusters



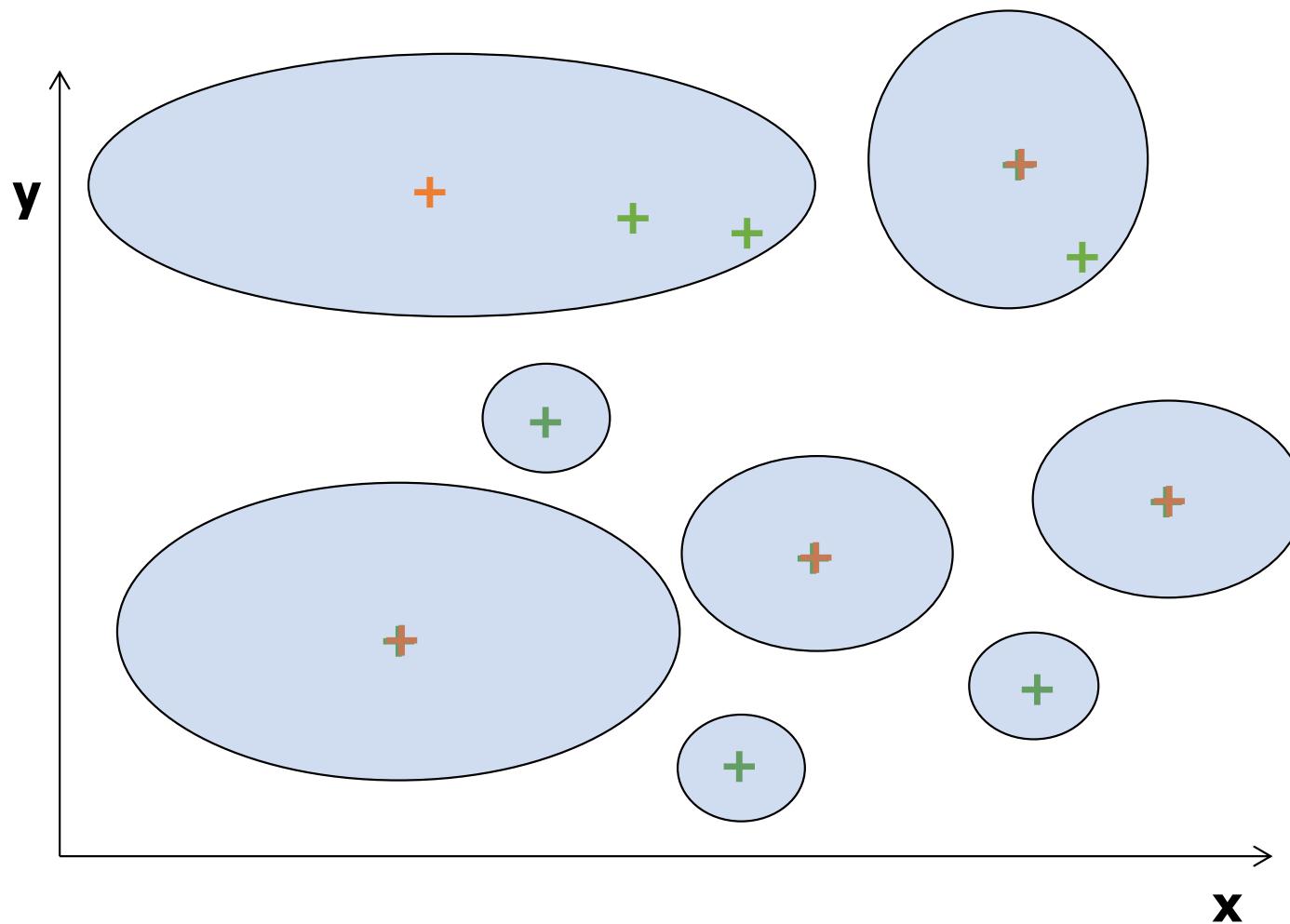
BFR – 3rd batch



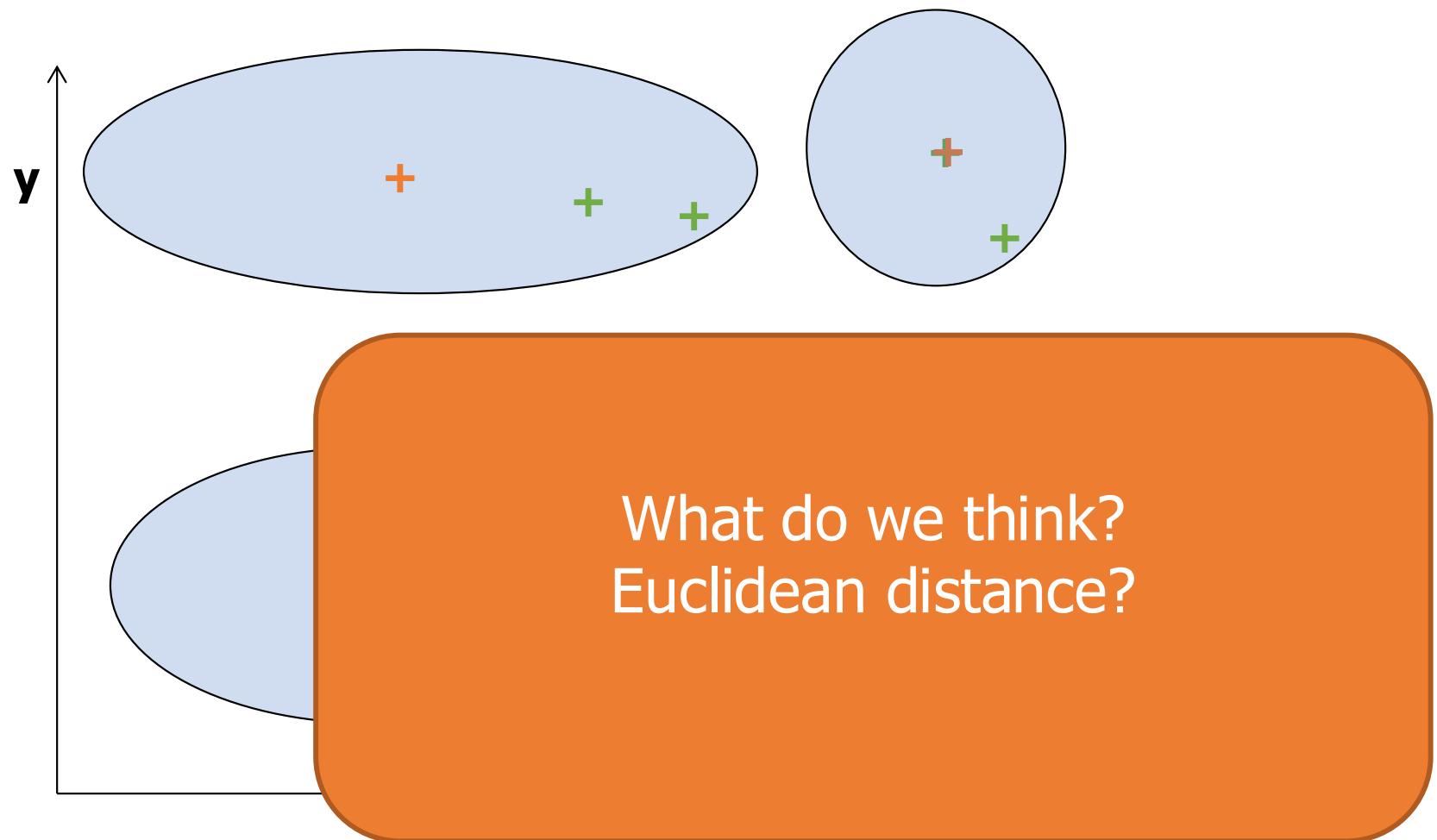
BFR – 3rd batch – discard, update, merge, also miniclusters!



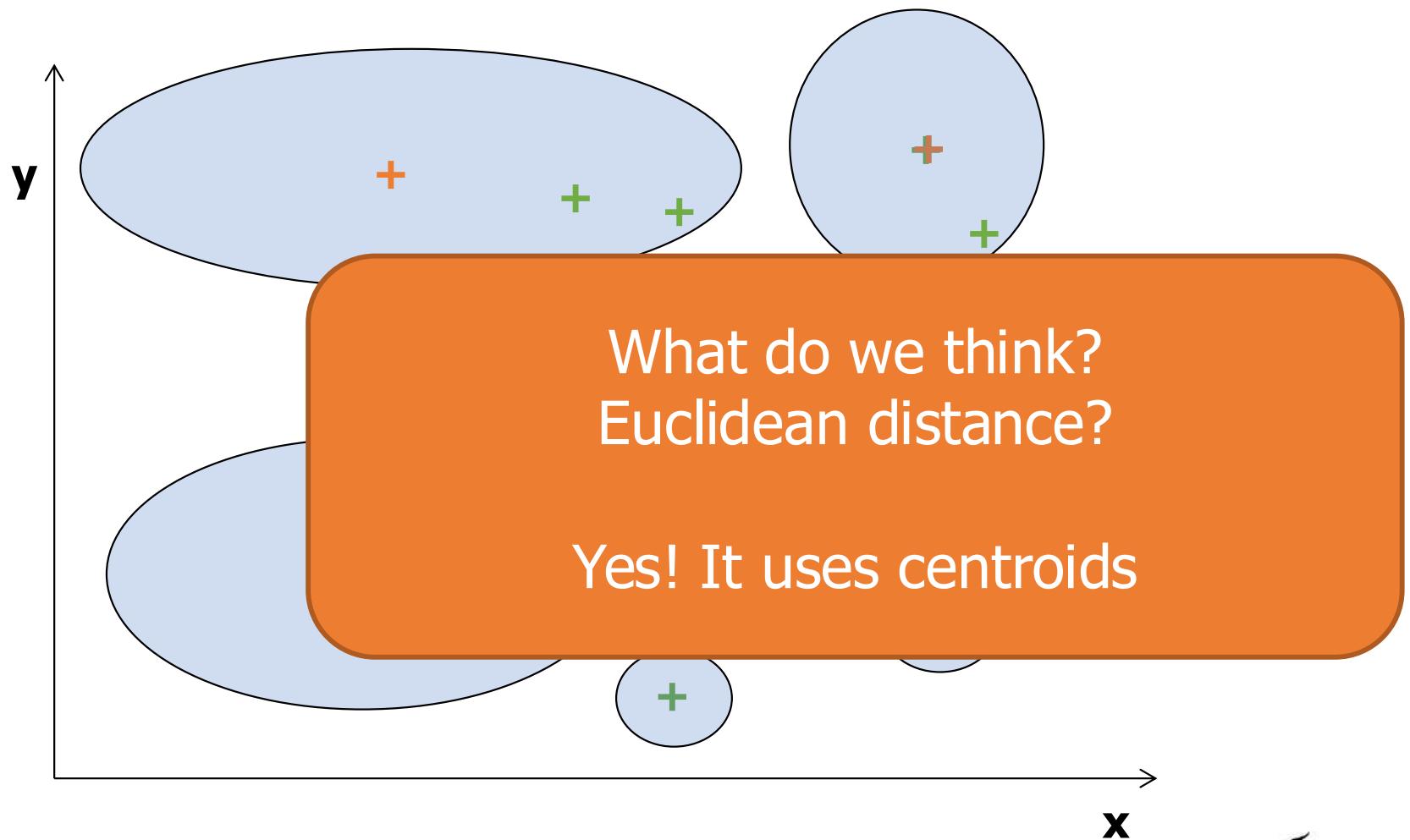
BFR – final step – decide what to do with miniclusters...



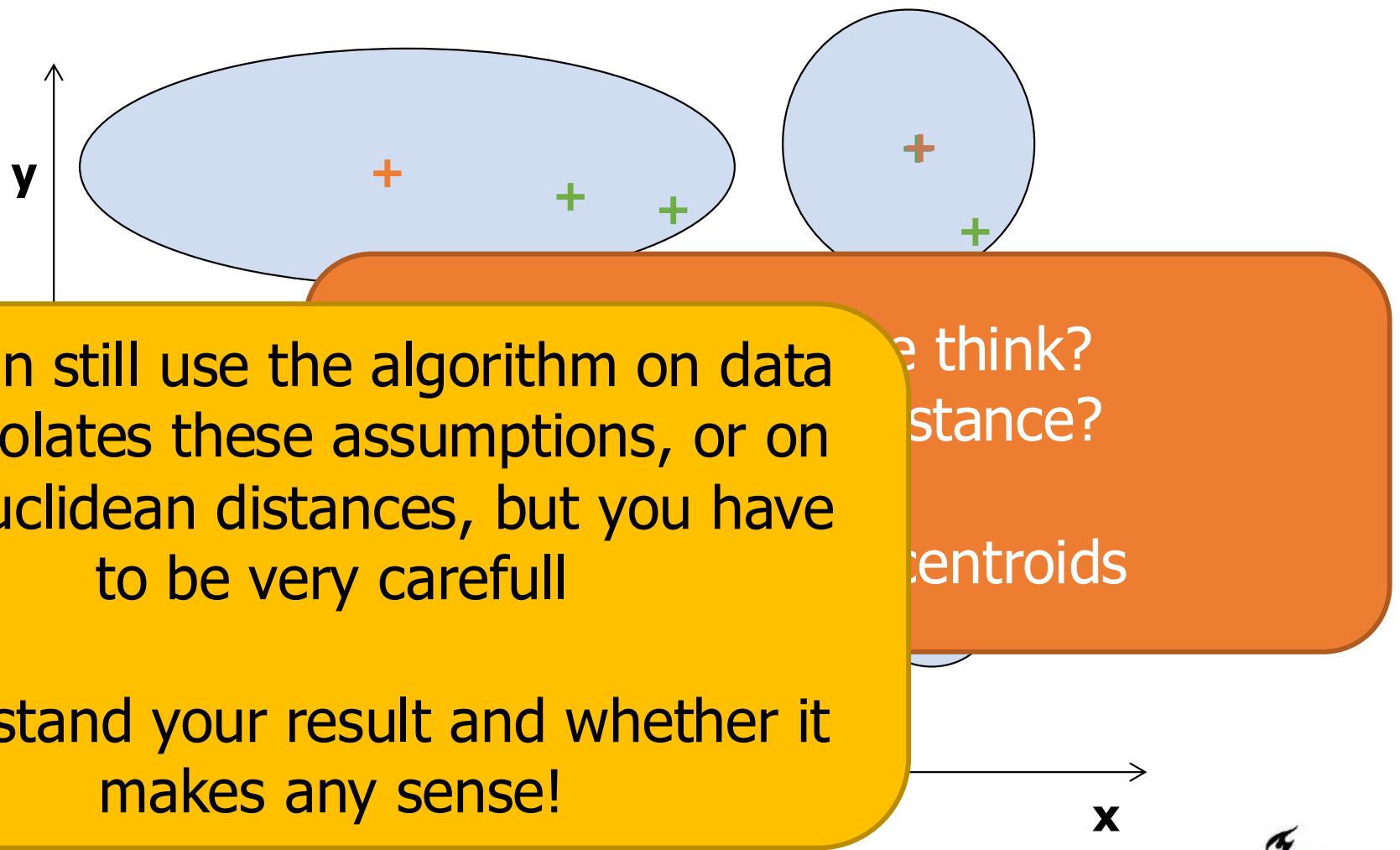
BFR – final step – decide what to do with miniclusters...



BFR – final step – decide what to do with miniclusters...



BFR – final step – decide what to do with miniclusters...



Clustering – take-away

- Is a typical data mining task
 - *Very dependent on distance*
 - Used algorithms also matters a lot
 - Methods get into local minima
 - Tricks to improve run-time
 - No clear objective function
- Avoid using default settings
- Visualize and understand the outcome
- Apply tricks for efficiency
 - Batches
 - Prototypes
 - Mini-clusters
 - Sufficient statistics

Exam material

- Concepts

- When is Euclidean distance needed?
- When is a metric needed?
- Which method can represent which type(s) of data?
- Understand the effects of normalization

- Skills

- Batching (Minibatch Kmeans)
- Prototyping (CURE)
- Initialization (Kmeans++)
- Sufficient statistics (BFR)
- Clustering and anomaly detection evaluation

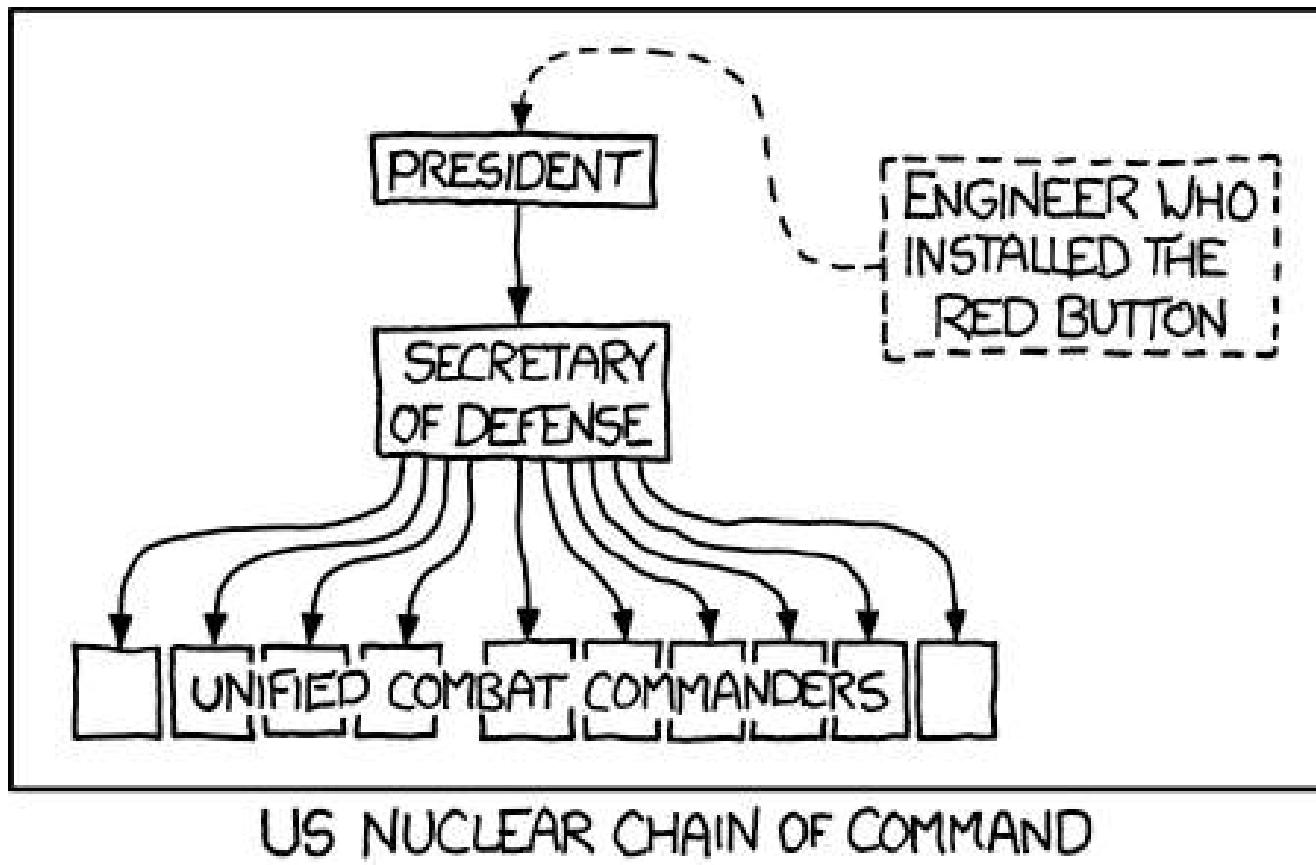
- Algorithms

- DBScan
- BFR

Ethics in mining data - discrimination

Undesired effects and methods to avoid them (partially)

Why ethics?



source: xkcd

Program

1. Statistics and misinterpretation
2. The dangers of opaque models
3. Discrimination in data mining

Sally Clark (1964-2007)



- Clark's first son died suddenly within a few weeks of his birth in September 1996, and in December 1998 her second died in a similar manner.
- A month later, she was arrested and subsequently tried for the murder of both children. The prosecution case relied on statistical evidence presented Professor Sir Roy Meadow, who testified that the chance of two children from a wealthy family suffering SIDS was 1 in 73 million.
- Meadow's law: "one sudden infant death in a family is a tragedy, two is suspicious, and three is murder unless proven otherwise"
- Clark was convicted in November 1999.

Prosecutor's fallacy

- What is the probability that a man is taller than 1m90, given that he is a professional basketball player?

Prosecutor's fallacy

- What is the probability that a man is taller than 1m90, given that he is a professional basketball player?
 - Reasonably large
- What is the probability that a man is a professional basketball player, given that he is taller than 1m90?

Prosecutor's fallacy

- What is the probability that a man is taller than 1m90, given that he is a professional basketball player?
 - Reasonably large
- What is the probability that a man is a professional basketball player, given that he is taller than 1m90?
 - Very small
- In math: $\Pr(A | B) \neq \Pr(B | A)$
- The **prosecutor's fallacy** is to think that the two are equal

Prosecutor's fallacy – Sally Clark

- What is the probability that both of Sally's children die within 3 months, given that Sally is not a murderer?
 - Very small, i.e., 1 in 73 million according to Meadow
- What is the probability that Sally is not a murderer, given that both of her children died within 3 months?
 - Unknown, but can be very large!

Prosecutor's fallacy – Sally Clark

- What is the probability that both of Sally's children die within 3 months, given that Sally is not a murderer?
 - Very small, i.e., 1 in 73 million according to Meadow
- What is the probability that both of her children die within 3 months
 - Unknown, but can be calculated

Two possible explanations:
 $P(I | E), P(-I | E)$

I = Innocent

$-I$ = not Innocent

E = Evidence

How to compute?

- $\Pr(I | E) / \Pr(\neg I | E) = ?$

more info (e.g., Lucia de Berk): Peter Grünwald

Bayes' theorem

- Posterior odds = likelihood ratio * prior odds
- $\Pr(I | E) / \Pr(-I | E) = \Pr(E | I) / \Pr(E | -I) * \Pr(I) / \Pr(-I)$

more info (e.g., Lucia de Berk): Peter Grünwald

Bayes' theorem

- Posterior odds = likelihood ratio * prior odds
- $\Pr(I | E) / \Pr(-I | E) = \Pr(E | I) / \Pr(E | -I) * \Pr(I) / \Pr(-I)$
- $\Pr(I | E) / \Pr(-I | E) = \text{very small} / \text{near 1} * \text{near 1} / \text{very small}$

more info (e.g., Lucia de Berk): Peter Grünwald

Bayes' theorem

- Posterior odds = likelihood ratio * prior odds
- $\Pr(I | E) / \Pr(-I | E) = \Pr(E | I) / \Pr(E | -I) * \Pr(I) / \Pr(-I)$
- $\Pr(I | E) / \Pr(-I | E) = \text{very small} / \text{near 1} * \text{near 1} / \text{very small}$
- $\Pr(I | E) / \Pr(-I | E) = \text{very small} * \text{very large} = ???$

more info (e.g., Lucia de Berk): Peter Grünwald

Does this apply to data mining?

Does this apply to data mining?

- YES! aka **Data Dredging**:

Performing many statistical tests and only reporting those with significant results

- or:

Claiming that a model obtained from data represents the truth

- or:

"If you torture the data long enough, it will confess to anything."

Learning from random data

```
from sklearn.tree import DecisionTreeClassifier  
from random import random  
x = [[random() for i in range(0,1000)] for j in range(0,1000)]  
y = [round(random()) for j in range(0,1000)]  
tree = DecisionTreeClassifier(max_depth=5).fit(x,y)  
print(sum(tree.predict(x) == y))
```

What accuracy do you get?

Learning from random data

```
from sklearn.tree import DecisionTreeClassifier  
from random import random  
x = [[random() for i in range(0,1000)] for j in range(0,1000)]  
y = [round(random()) for j in range(0,1000)]  
tree = DecisionTreeClassifier(max_depth=5).fit(x,y)  
print(sum(tree.predict(x) == y))
```

What accuracy do you get?

About 70-80%

Learning from random data

```
from sklearn.tree import DecisionTreeClassifier  
from random import random  
x = [[random() for _ in range(4)] for _ in range(100)]  
y = [round(random()) for _ in range(100)]  
tree = DecisionTreeClassifier().fit(x, y)  
print(sum(tree.feature_importances_))
```

Of course, this should be prevented using something like cross-validation

In data mining, however, we do not have labeled data, only patterns

And patterns exist even in completely random data...

Small probabilities



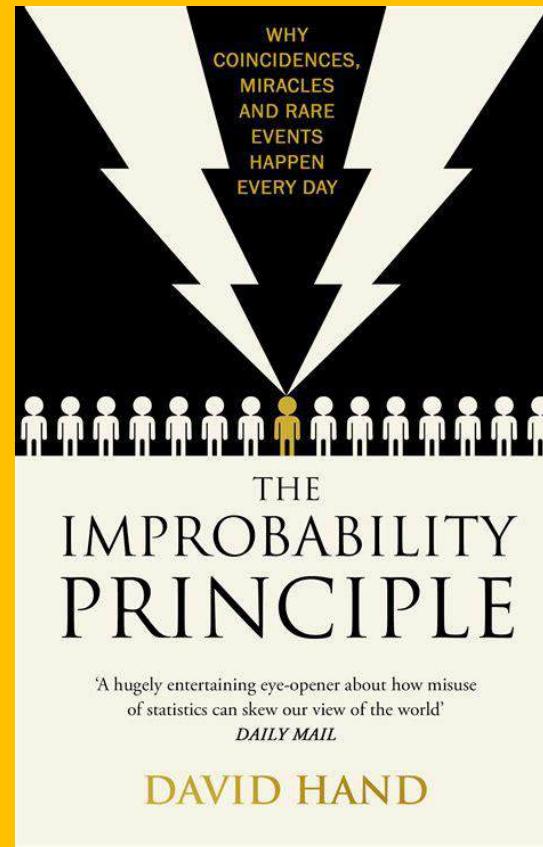
"You know, the most amazing thing happened to me tonight... I saw a car with the license plate ARW 357. Can you imagine? Of all the millions of license plates in the state, what was the chance that I would see that particular one tonight? Amazing!"

Richard Feynman

Small probabilities



Things with very small probability happen all the time!



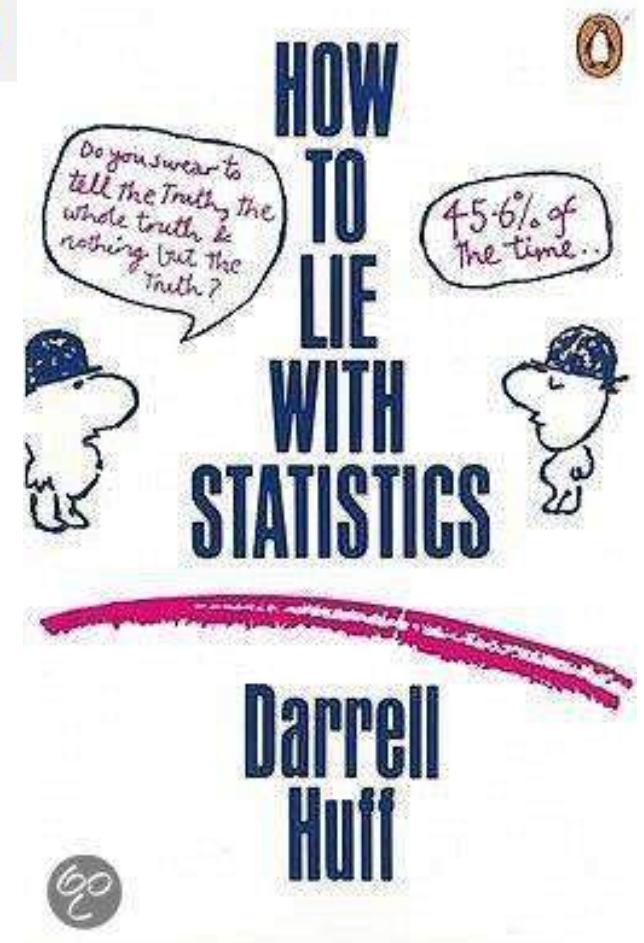
Lies, damned lies, and statistics



statistics are
statistics are
statistics are **bullshit**
statistics are **made up**
statistics are **not used by scientists to**

Press Enter to search.

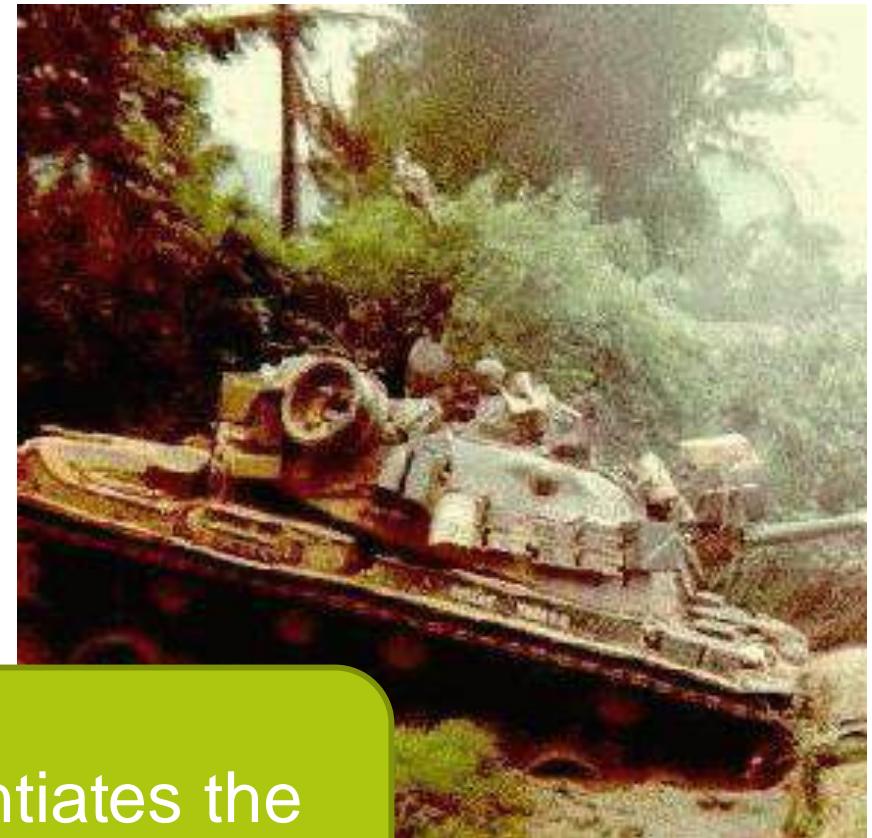
- Hard to understand
- Easy to misinterpret
- *The cause of many errors in data mining practice...*



Program

1. Statistics and misinterpretation
 - Prosecutor's fallacy - $Pr(A | B) \neq Pr(B | A)$
 - Data dredging – *using patterns in data as the truth*
2. The dangers of opaque models
3. Discrimination in data mining

Tank classification



What differentiates the
left image from the right?

Scary: predicting crime

- Criminaliteits Anticipatie Systeem (CAS), from NRC:

“... Om te voorspellen welke combinaties van kenmerken indicatief zijn voor criminaliteit in de nabije toekomst, wordt gebruik gemaakt van kunstmatige neurale netwerken ... De huidige CAS-model kan 36,3% van de woninginbraken en 57,7% van de straatrovers voorspellen.”



“De werking van CAS is niet wetenschappelijk bewezen en lastig te meten”... *aka, we do not really know how it works...*

Scary: predicting crime

- Criminaliteits Anticipatie Systeem (CAS), from NRC:

“... Om te voorspellen welke combinaties van kenmerken indicatief zijn voor criminelen in de nabije toekomst. CAS is een gebruik gemaakte kunstmatige neuraal netwerk. De huidige CAS bestaat uit een combinatie van de woningindicatoren en de straatrecords.”

“De werking van CAS is lastig te meten”

Actually, I recently learned that CAS is no longer used in practice...

Another danger of designing (complex) opaque-box systems: we do not use what we do not understand

Or is this a good property?



Beware of feedback loops



A simple loop example

- Say we use a rule to decide **who to investigate** for fraud, and we **use the results as new training data**, using $P(\text{House} \mid \text{Fraud})$

	Fraud	Non-fraud	Total
Hufflepuffs	51	49	100
Ravenclaws	49	51	100

Source: Ionica Smeets

A simple loop example

- Say we use a rule to decide **who to investigate** for fraud, and we **use the results as new training data**, using $P(\text{House} \mid \text{Fraud})$

	Fraud	Non-fraud	Total
Hufflepuffs	51	49	100
Ravenclaws	49	51	100

- We investigate 102 Hufflepuffs and 98 Ravenclaws

Source: Ionica Smeets

A simple loop example

- Say we use a rule to decide **who to investigate** for fraud, and we **use the results as new training data**, using $P(\text{House} \mid \text{Fraud})$

	Fraud	Non-fraud	Total
Hufflepuffs	51	49	100
	Fraud	Non-fraud	Total
Hufflepuffs	52	50	102
Ravenclaws	48	50	98

- We investigate 104 Hufflepuffs and 96 Ravenclaws

Source: Ionica Smeets

A simple loop example

- Say we use a rule to decide **who to investigate** for fraud, and we **use the results as new training data**, using $P(\text{House} \mid \text{Fraud})$

	Fraud	Non-fraud	Total
Hufflepuffs	51	49	100
	Fraud	Non-fraud	Total
Hufflepuffs	52	50	102
	Fraud	Non-fraud	Total
Hufflepuffs	53	51	104
Ravenclaws	47	49	96

Source: Ionica Smeets

A simple loop example

- Say we use a rule to decide **who to investigate** for fraud, and we **use the results as new training data**, using $P(\text{House} \mid \text{Fraud})$

	Fraud	Non-fraud	Total
Hufflepuffs	51	49	100
	Fraud	Non-fraud	Total
Hufflepuffs	52	50	102
	Fraud	Non-fraud	Total
Hufflepuffs	53	51	104
	Fraud	Non-fraud	Total
Hufflepuffs	54	52	106
Ravenclaws	46	48	94

Source: Ionica Smeets

A simple loop example

- Say we use a rule to decide **who to investigate** for fraud, and we **use the results as new training data**, using **P(Att | Fraud)**

This problem can occur anytime output data is used as new input, i.e.,
when we control what data we see

It is a very big problem in recommender systems

and in law enforcement/fraud detection

Never blindly maximize accuracy!

- Q: you have to investigate 200 people for fraud, who do you investigate to catch the largest number of fraudsters in expectation?

	Fraud	Non-fraud	Total
Hufflepuffs	51	49	100
Ravenclaws	49	51	100

Program

1. Statistics and misinterpretation
 - Prosecutor's fallacy - $Pr(A|B) \neq Pr(B|A)$
 - Data dredging – *using patterns in data as the truth*
2. The dangers of opaque models
 - Trusting algorithms over common sense - *don't!*
 - Beware of feedback loops - *avoid!*
3. Discrimination in data mining

Discrimination



Discrimination



Side-effect of inventory management

Black Barbie sells less often than White Barbie

Recent examples

Amazon ditched AI recruiting tool that favored men for technical jobs

Machine Bias

There's software used across the country to predict future criminals. And it's biased against blacks.

by Julia Angwin, Jeff Larson, Surya Mattu and Lauren Kirchner, ProPublica

May 23, 2016

COMPUTING

Racial Bias Found in a Major Health Care Risk Algorithm

Black patients lose out on critical care when systems equate health needs with costs

By Starre Vartan on October 24, 2019

Solutions?

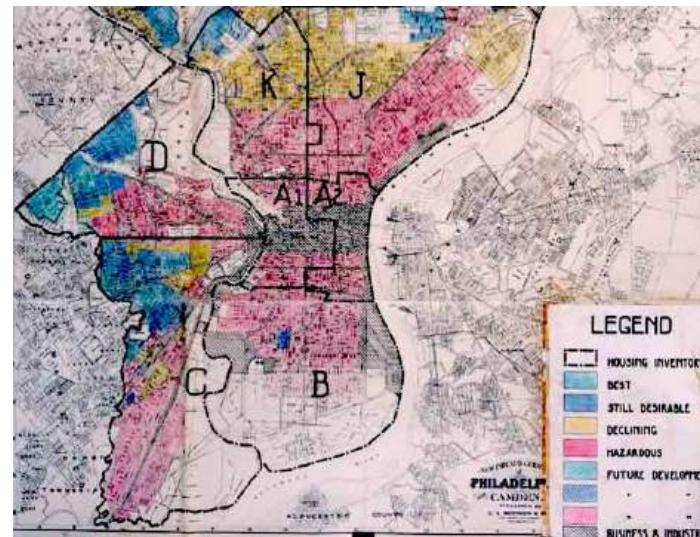
• ...

So, Let's Omit Sensitive Attributes

...

- *Just removing the sensitive attributes does not help!*

- Other attributes may be highly correlated with the sensitive attribute:
 - gender ↔ profession
 - race ↔ postal code
 - ...



The HOLC maps are part of the records of the FHLBB (RG195) at the [National Archives II](#)

- **We call this the *redlining effect***

Example: census dataset

% high income difference
males/females

Original data

	male	female
loan	3256	590
no loan	7604	4831

19%

Predictions

	male	female
loan	4559	422
no loan	6301	4999

31%

Predictions not based on gender

	male	female
loan	4134	567
no loan	6726	4854

28%

Example: census dataset

% high income difference

Note: unfortunately this study and data contain no information on non-binary genders

How to fix classification not to discriminate against a group of people not present in the data is another important open problem, but not part of today's lecture

Predictions

	male	female
4559		422
5301		4999

31%

Based on gender

	male	female
4134		567
3726		4854

28%

Solution

- Basic idea:
 - Instead of only maximizing accuracy/likelihood
 - Learn a model that maximizes accuracy, while minimizing discrimination
- First: what is discrimination?

Measuring discrimination

- Berkeley discrimination case:

	Applicants	Admitted
Men	8442	44%
Women	4321	35%

- The probability of acceptance:
 - $\Pr(\text{admitted} \mid \text{men}) = 0.44$
 - $\Pr(\text{admitted} \mid \text{women}) = 0.35$
- Standard discrimination measure = difference in acceptance probability
 - $\Pr(\text{admitted} \mid \text{men}) - \Pr(\text{admitted} \mid \text{women}) = 0.09$

First solution: preferential sampling

- Randomly draw examples from every sensitive group
- From a **discriminated** group:
 - Draw more examples with positive labels
 - Draw less examples with negative labels
- From a **favored** group:
 - Draw less examples with positive labels
 - Draw more examples with negative labels
- Until a data set is created with 0 discrimination
 - *Maintain the same fraction of positive/negative labels!*
- Learn a model from this data

Second solution: different thresholds

- Learn a probabilistic model
- Such a model gives a probability $P(i)$ that example i belongs to the positive class
 - Naïve Bayes is such a classifier
- Use different decision thresholds t for the different sensitive groups, such that
 - If $P(i) > t$, then i is labeled as positive
 - The fraction of examples labeled as positive is roughly equal for every sensitive group

Second solution: different thresholds

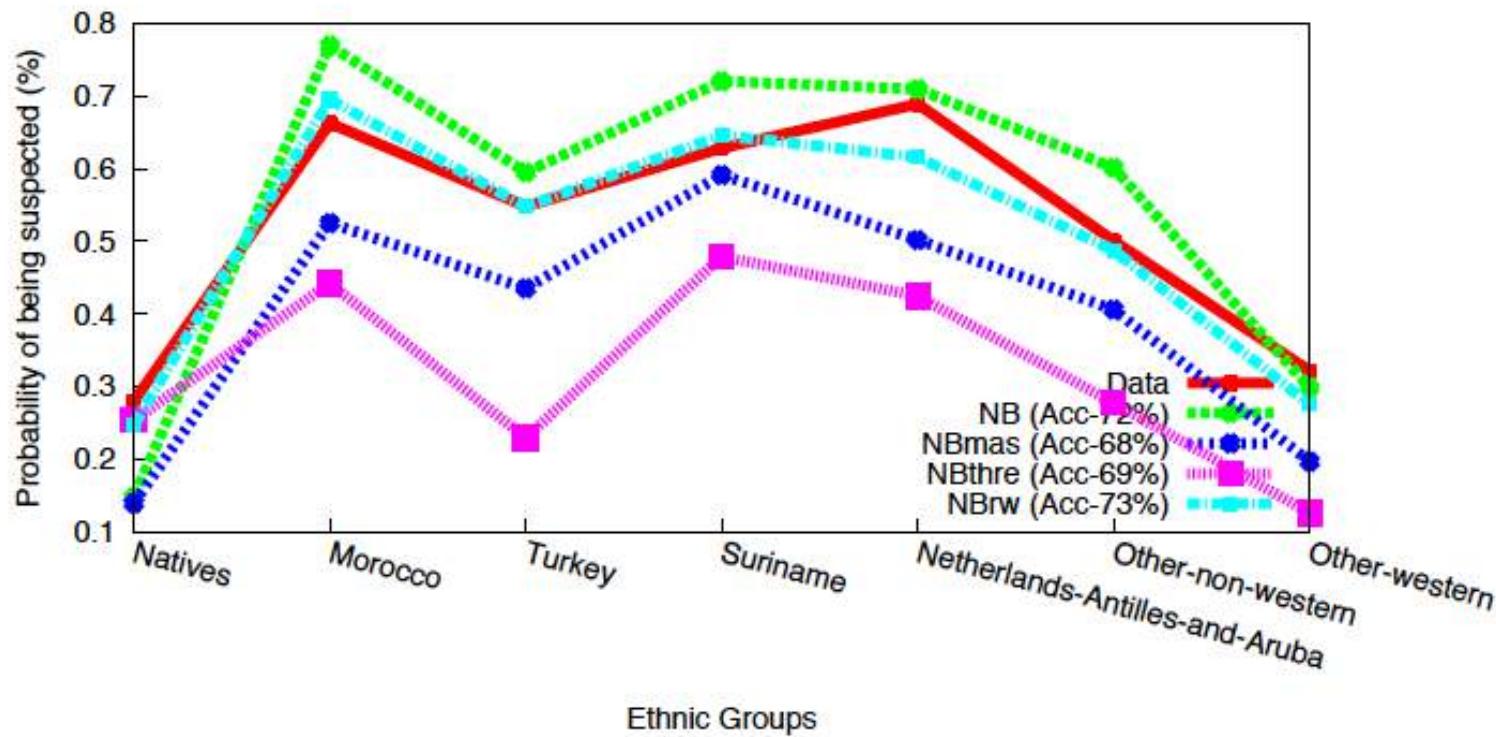
- Learn a probabilistic model
 - Such a model gives a probability $P(i)$ that example i belongs to the positive class
 - Naïve Bayes is such a classifier
 - Use different decision thresholds t for the different sensitive groups, such that
 - If $P(i) > t$, then i is in the positive group
 - The fraction of false positives is the same for every sensitive group
- But this does not avoid red-lining...

Third solution: different models

- Learn a probabilistic model **for every sensitive group**
- Such a model gives a probability $P(i)$ that example i belongs to the positive class
 - Naïve Bayes is such a classifier
- Use different decision thresholds t for the different groups, such that
 - If $P(i) > t$, then i is labeled as positive
 - The fraction of examples labeled as positive is roughly equal for each group

An ethnic profiling study

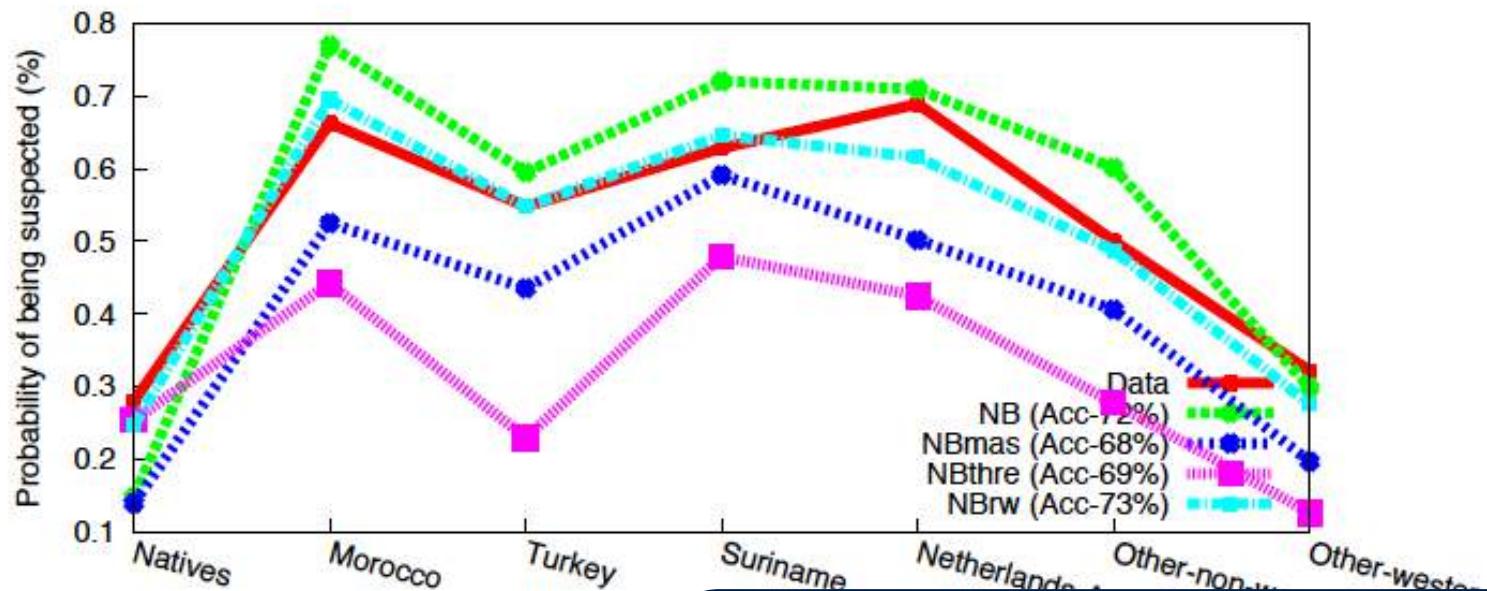
Kamiran, F., Karim, A., Verwer, S. and Goudriaan, H., Classifying socially sensitive data without discrimination: an analysis of a crime suspect dataset. In *2012 IEEE 12th ICDM Workshops 2012*.



- NB = learning naïve Bayes classifier
- NBmas, NBrw = forms of preferential sampling
- NBthre = use different models and thresholds

An ethnic profiling study

Kamiran, F., Karim, A., Verwer, S. and Goudriaan, H., Classifying socially sensitive data without discrimination: an analysis of a crime suspect dataset. In *2012 IEEE 12th ICDM Workshops 2012*.



- NB = learning naïve Bayes
- NBmas, NBrw = forms of priors
- NBthre = use different models

We can reduce discrimination
with little accuracy loss

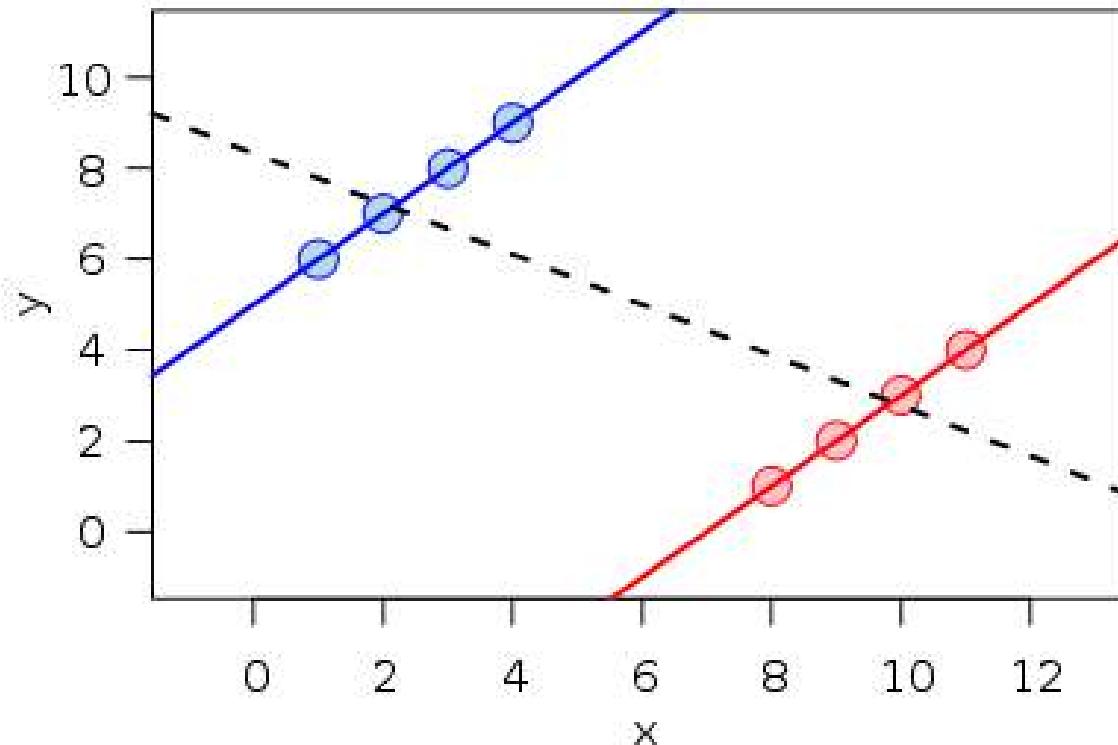
Its not so simple: Simpsons paradox

- Berkeley discrimination case, per department:

Department	Men		Women	
	Applicants	Admitted	Applicants	Admitted
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	6%	341	7%

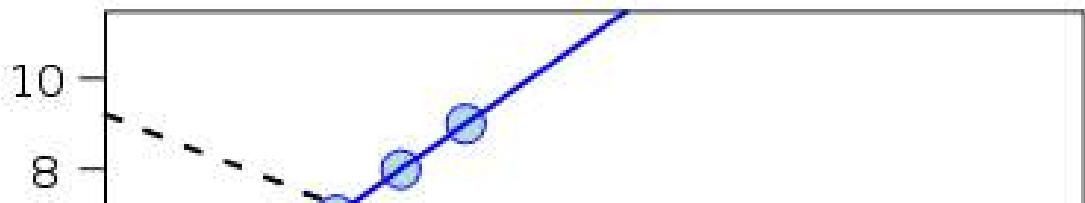
- Seems to favor woman!

Simpsons paradox



- A global trend can be completely reversed when data is split into groups!
- There was no discrimination at Berkeley

Simpsons paradox



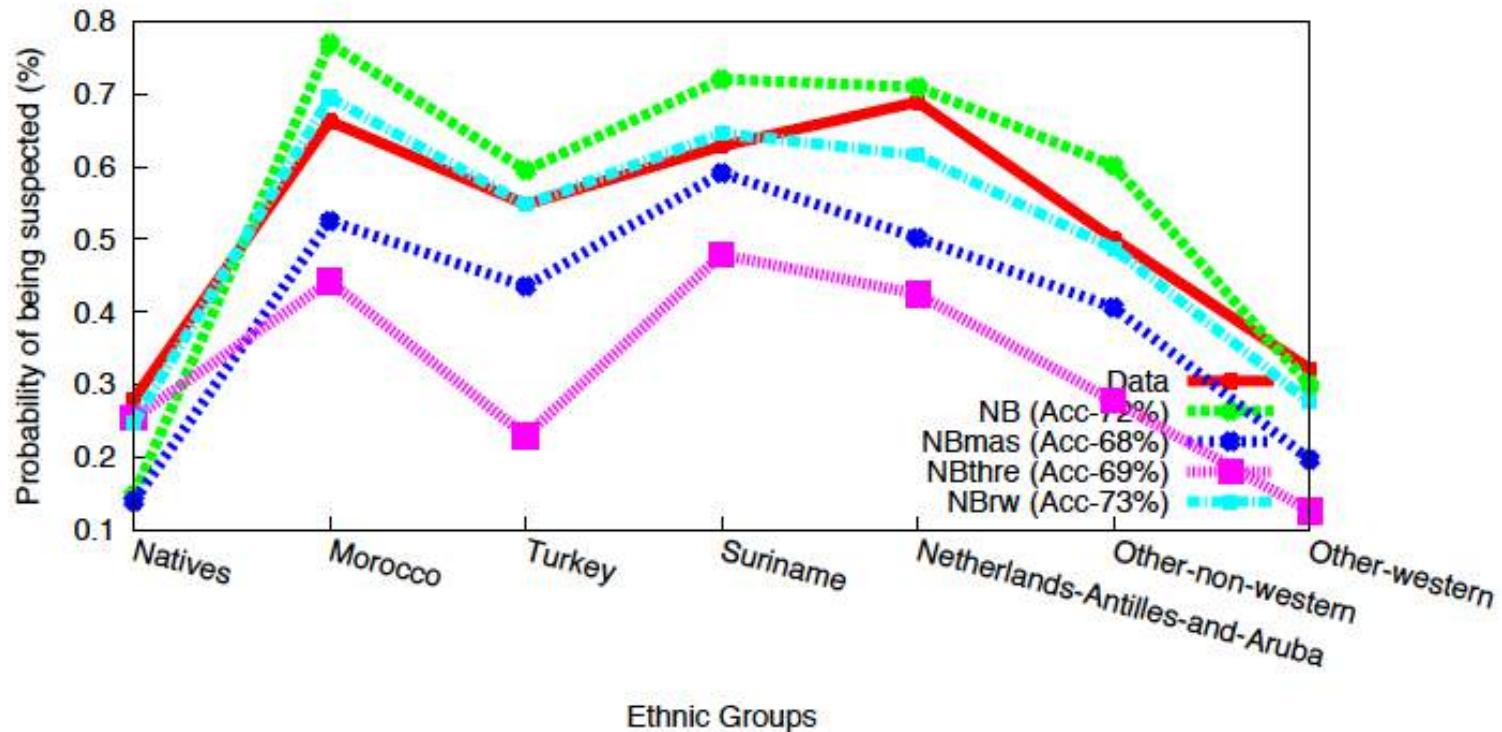
Some discrimination is OK

When it can be explained using other attributes
e.g., education, experience, salary, etc. ...

-

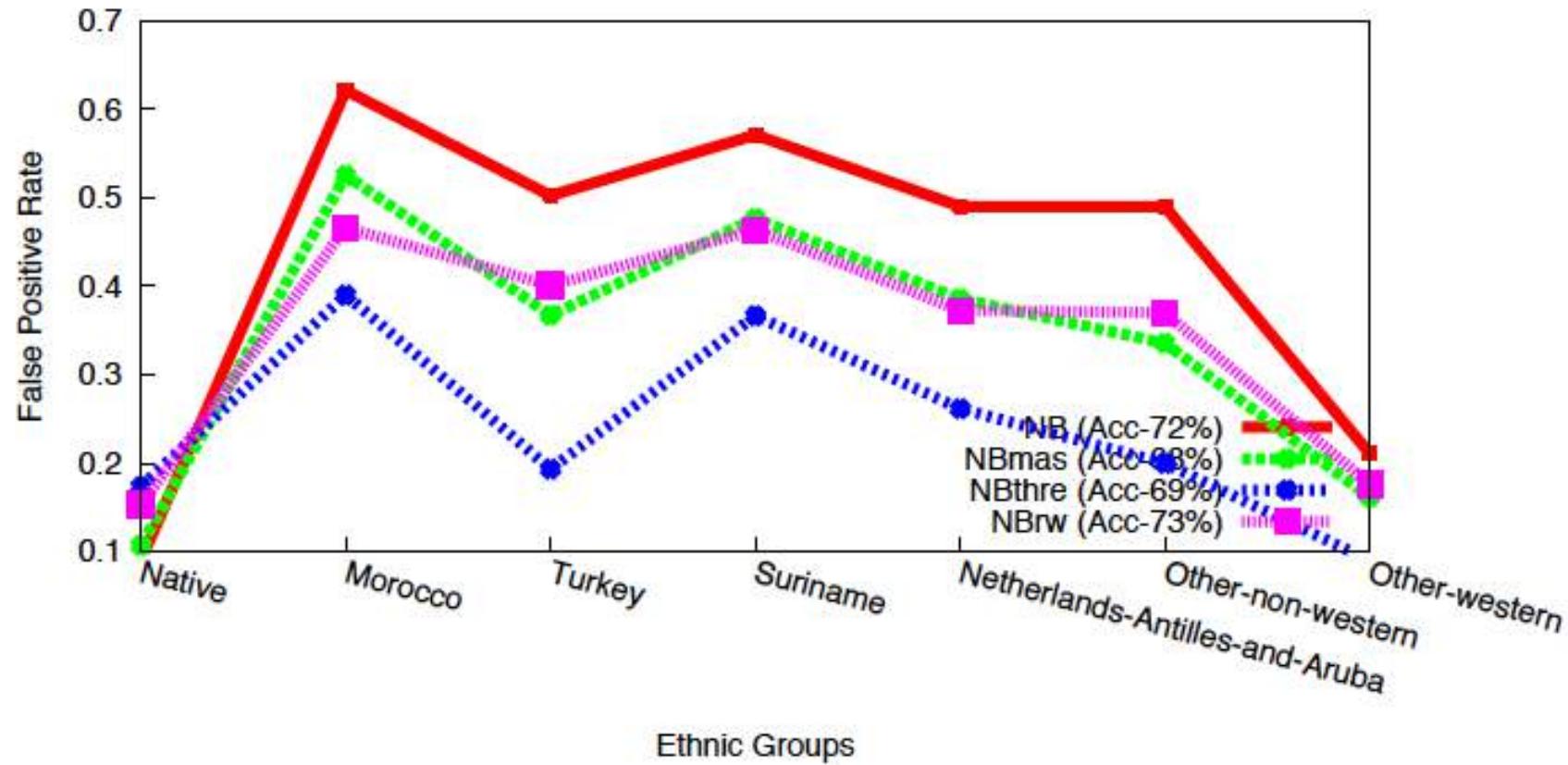
-

An ethnic profiling study



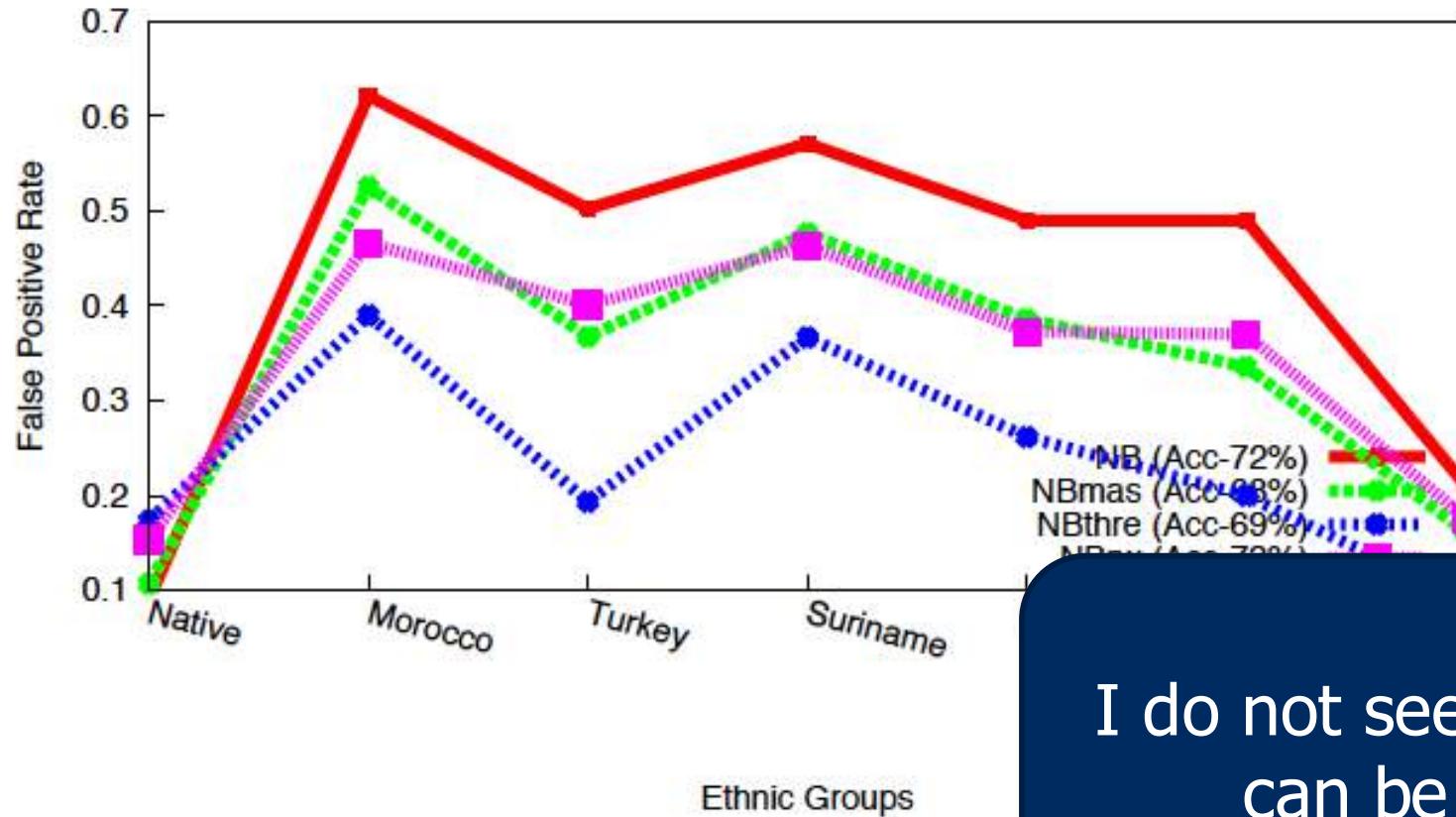
Is the discrimination OK?

Other measure: false positive rate



Native Dutch have 9% probability of being falsely accused by the model, Moroccans 63%...

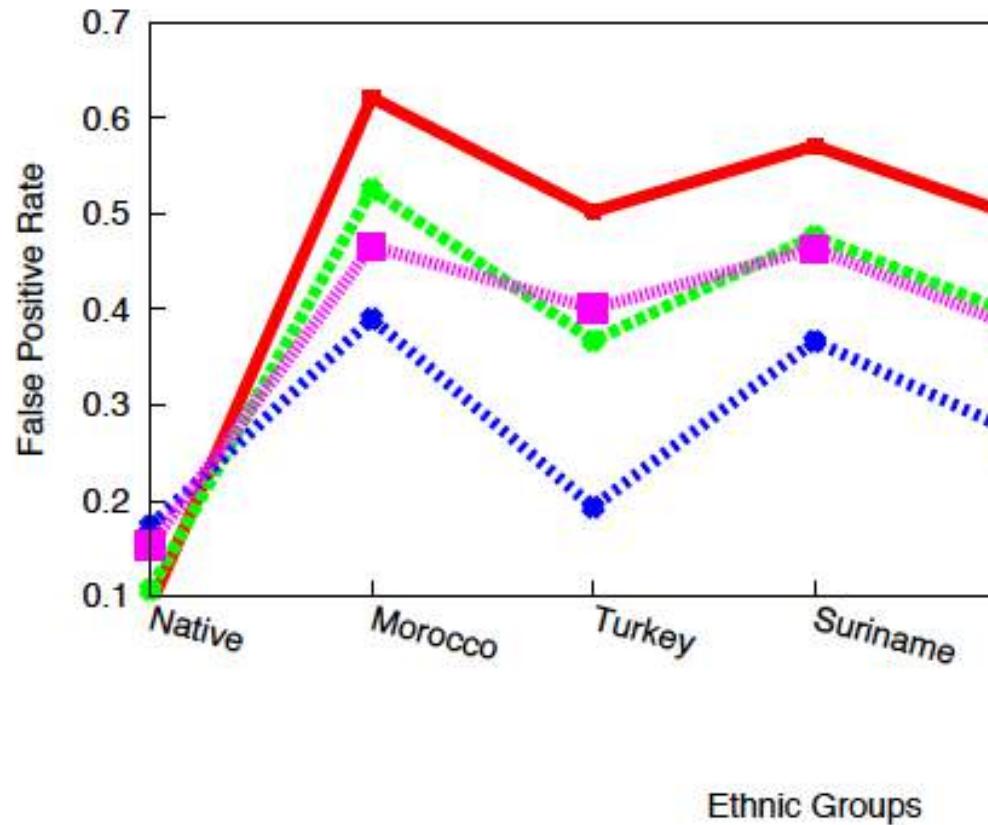
Other measure: false positive rate



Native Dutch have 9% probability of being flagged by the model, Moroccans 63%...

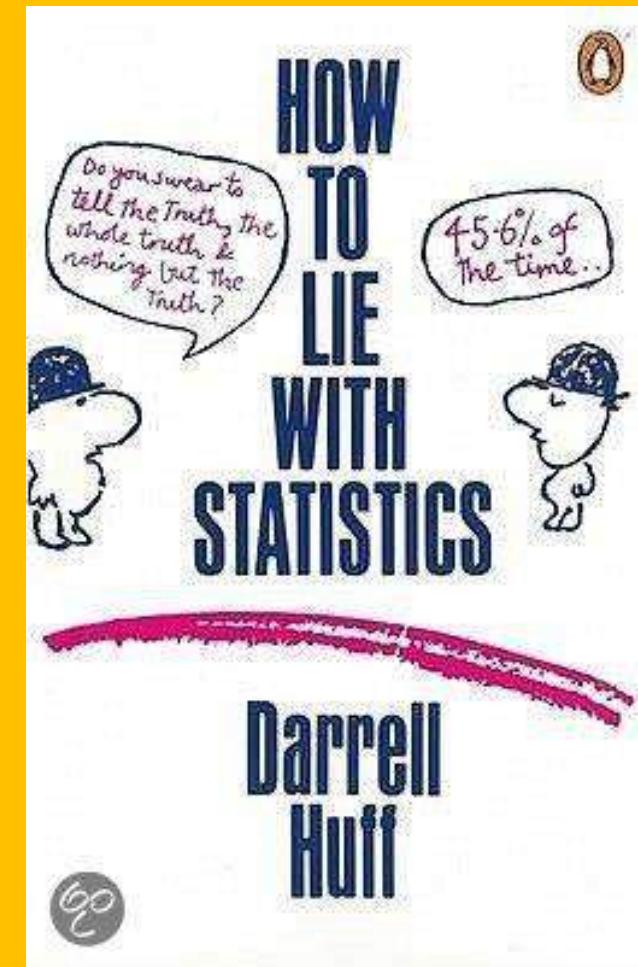
I do not see how this can be OK..

Other measure: false positive rate

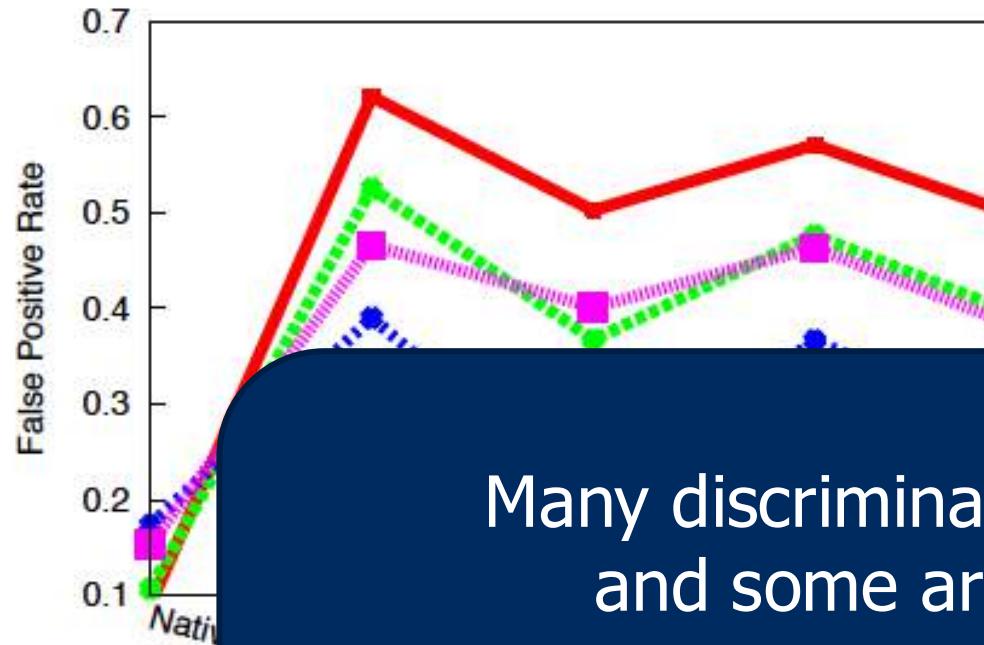


Native Dutch have 9% probability
by the model, Moroccans 63%...

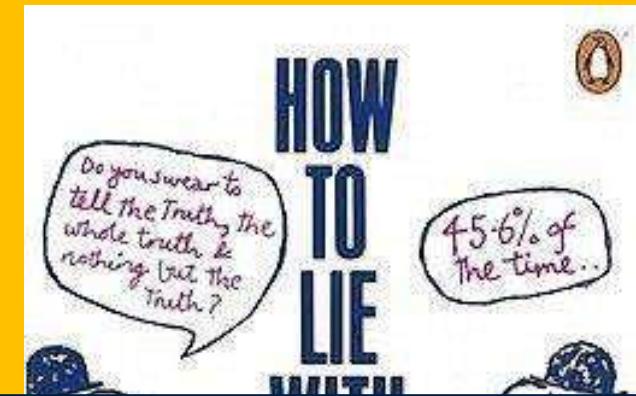
Remember:



Other measure: false positive rate



Remember:

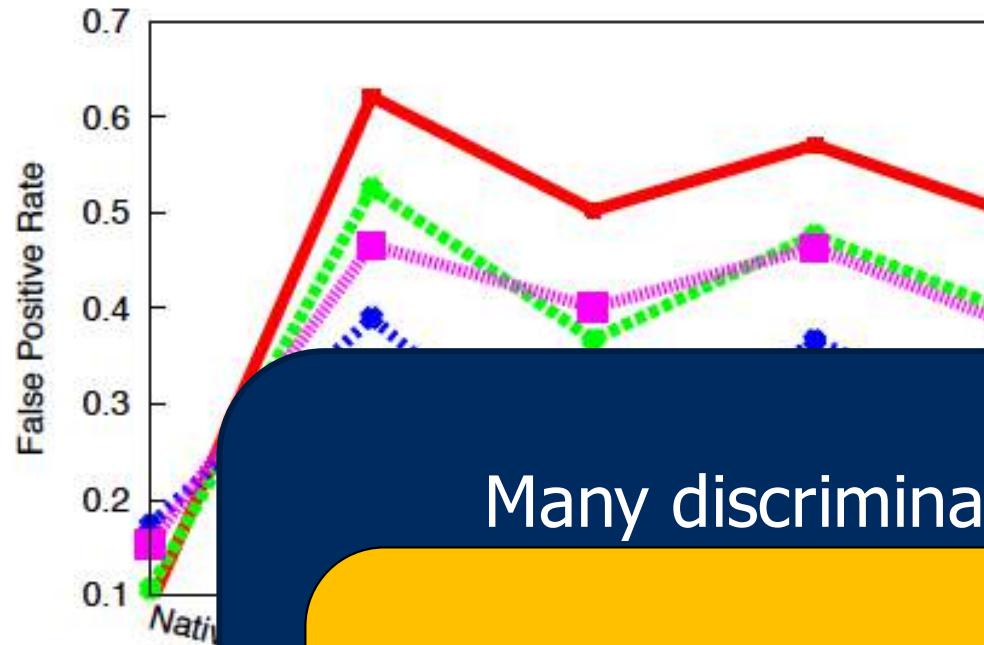


Many discrimination measures exist,
and some are contradictory...

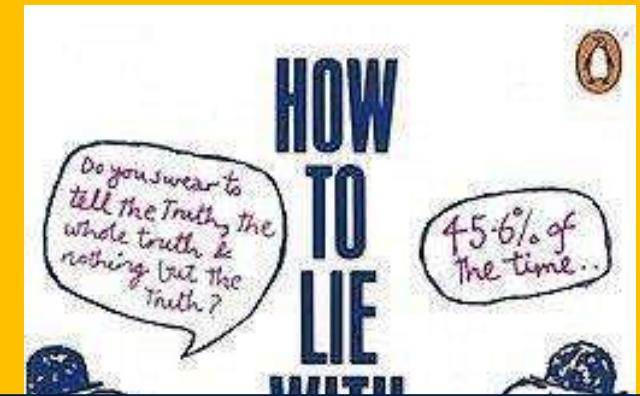
How to mine data is unclear in practice...
But that is also what makes it fun!

Native
by the

Other measure: false positive rate



Remember:



Many discrimination measures exist,

If anything, consider whether high accuracy is truly your objective...

Predictive policing in the Netherlands?

<https://nos.nl/artikel/2250767-politie-wil-zakkenrollers-en-plofkrakers-vangen-met-data.html>

Is this sufficient reason to use the system?

- The probability of criminals getting flagged correctly is greater than the probability of citizens getting flagged incorrectly.

Is this sufficient reason to use the system?

- The probability of criminals getting flagged correctly is greater than the probability of citizens getting flagged incorrectly.

Is the statement correct?

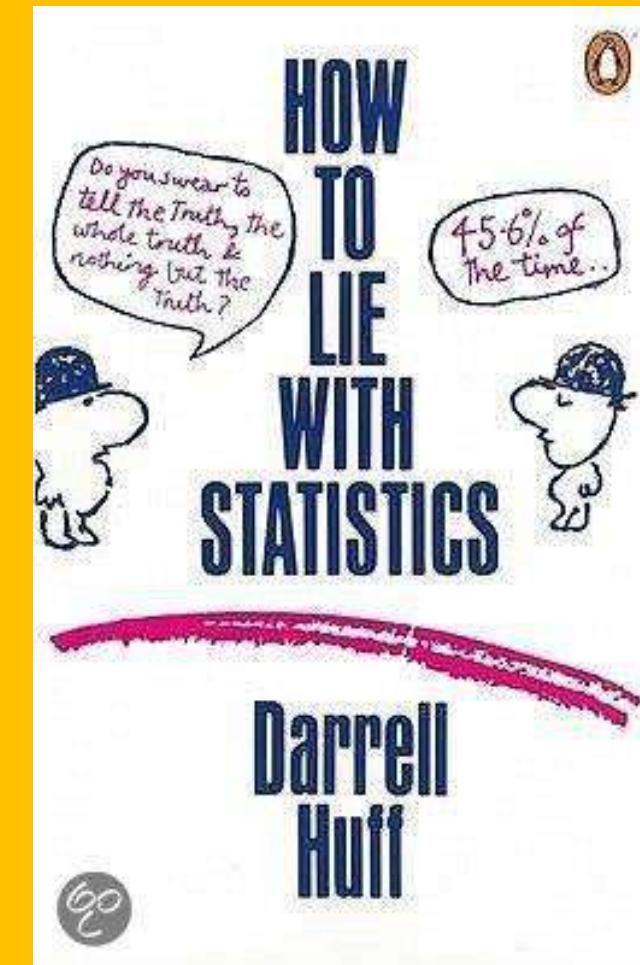
Is this sufficient reason to use the system?

- The probability of criminals getting flagged correctly is greater than the probability of citizens getting flagged incorrectly.
- $P(\text{Flagged} \mid \text{Criminal}) > P(\text{Flagged} \mid \text{Citizen})$
- $\#\text{Crim. Flagged} / \#\text{Crim.} > \#\text{Cit. Flagged} / \#\text{Cit.}$
- Say:
 - $2 / 10 > 10 / 10.000$
 - Of course, this is true!

Is this sufficient reason to use the system?

- The probability of criminals getting flagged is greater than the probability of citizens being flagged incorrectly.
- $P(\text{Flagged} \mid \text{Criminal}) > P(\text{Flagged} \mid \text{Citizen})$
- $\#\text{Crim. Flagged} / \#\text{Crim.} > \#\text{Cit. Flagged} / \#\text{Cit.}$
- Say:
 - $2 / 10 > 10 / 10.000$
 - Of course, this is true!

Remember:



Program (also exam content)

1. Statistics and misinterpretation

- Prosecutor's fallacy - $Pr(A | B) \neq Pr(B | A)$
- Data dredging – *using patterns in data as the truth*

2. The dangers of opaque models

- Trusting algorithms over common sense - *don't!*
- Beware of feedback loops - *avoid!*

3. Discrimination in data mining

- Redlining – *using attributes correlated with sensitive to predict*
- Simpson's paradox – *correlation can reverse in subgroups*
- Affirmative action – *assigning more positive labels to discriminated groups*

Interested? Further developments

- Research continued in many directions:
- Fair machine learning algorithms:
 - Bias-aware classification
 - Fair representations
 - Fair regression, ranking, clustering, ...
- Measures for discrimination/bias
 - Incompatibility results
- Causal models for fairness

Further developments : *optimal/declarative learning*

	f1	f2	f3	f4	t
1	5.1	3.5	1.4	0.2	s
2	4.9	3.0	1.4	0.2	s
3	4.7	3.2	1.3	0.3	v

translate

add
objective

$$l_{5,1} + l_{6,1} + 2 \cdot t_{n,1} \leq 2$$

$$l_{3,2} + l_{4,2} + l_{5,2} - 3 \cdot t_{n,1} \leq 0$$

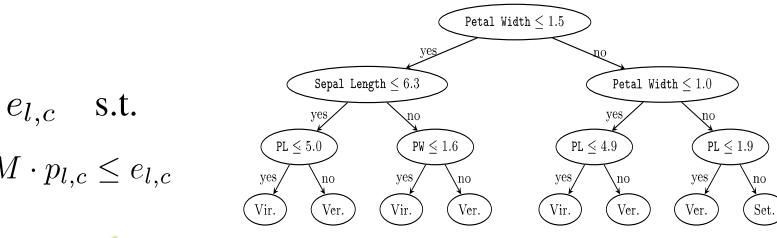
$$l_{3,1} + l_{4,1} + 2 \cdot t_{n,1} + 2 \cdot t_{n,2} \leq 4$$

$$l_{3,2} + l_{4,2} - 2 \cdot t_{n,2} \leq 0$$

$$l_{6,1} + 1 \cdot t_{n,2} \leq 1$$

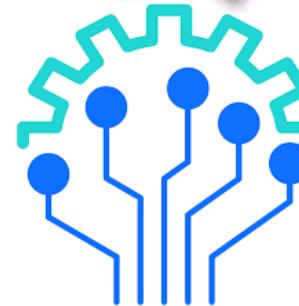
$$l_{6,2} - 1 \cdot t_{n,1} - 1 \cdot t_{n,2} \leq 0$$

$$\begin{aligned} \min \sum_{l,c} e_{l,c} \quad & \text{s.t.} \\ \sum_{r:C_r=c} l_{r,l} - M \cdot p_{l,c} \leq e_{l,c} \end{aligned}$$



solve

translate

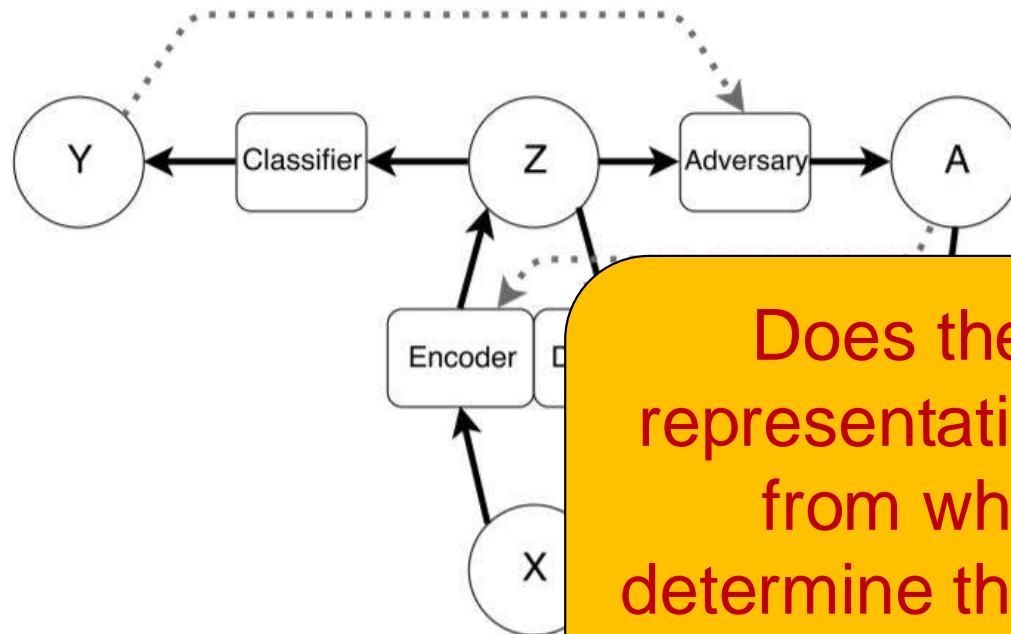


add bias constraints

Verwer and Zhang. "Learning decision trees with flexible constraints and objectives using integer optimization" CPAIOR, 2017

Further developments : *adversarial learning*

- Learn intermediate representation that allows to predict target but disallows inferring the sensitive attribute



Does there exist a representation (think PCA) from which we can determine the class, but not a person's ethnicity?

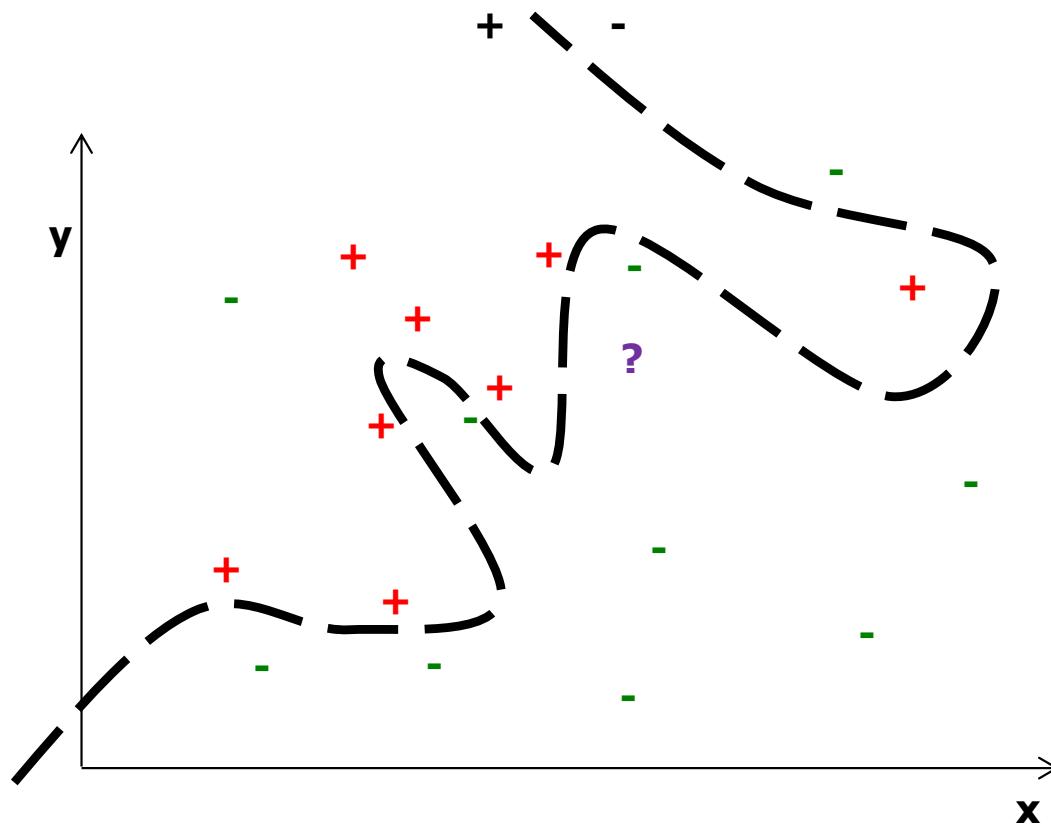
Madras, Creager Pitassi, Zemel. Learning Adversarially Fair Representations.
ICML 2018

Bonus: XAI – understanding
opaque boxes
(not part of exam content)

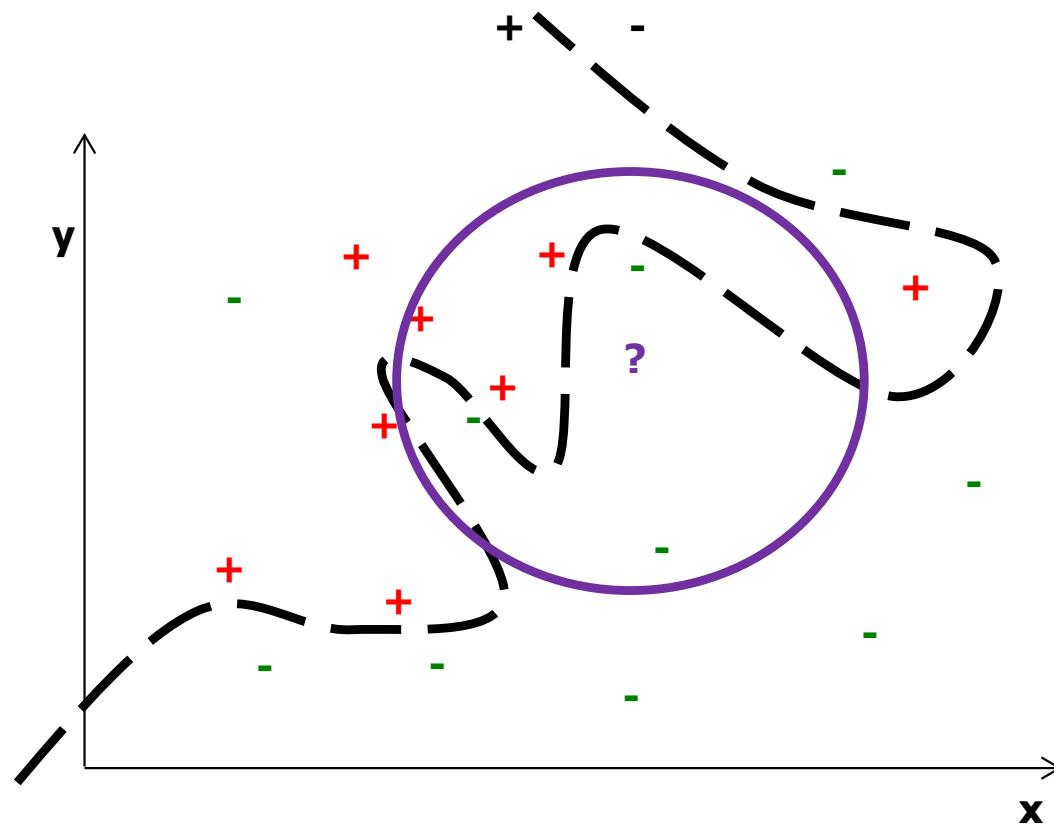
Opaque vs Transparant models

- An opaque model aims for optimal performance and does not care (so much) about the function it represents
- A transparant model aims to learn a function that can be understood by human experts

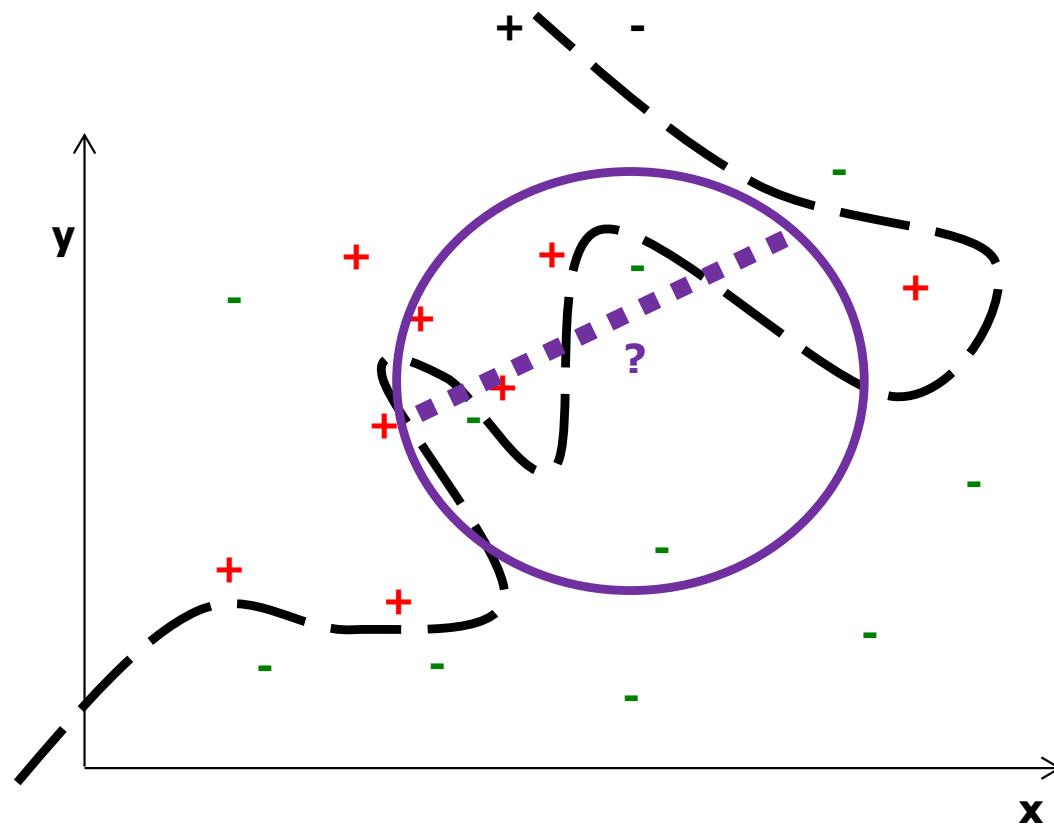
Explaining an Opaque model



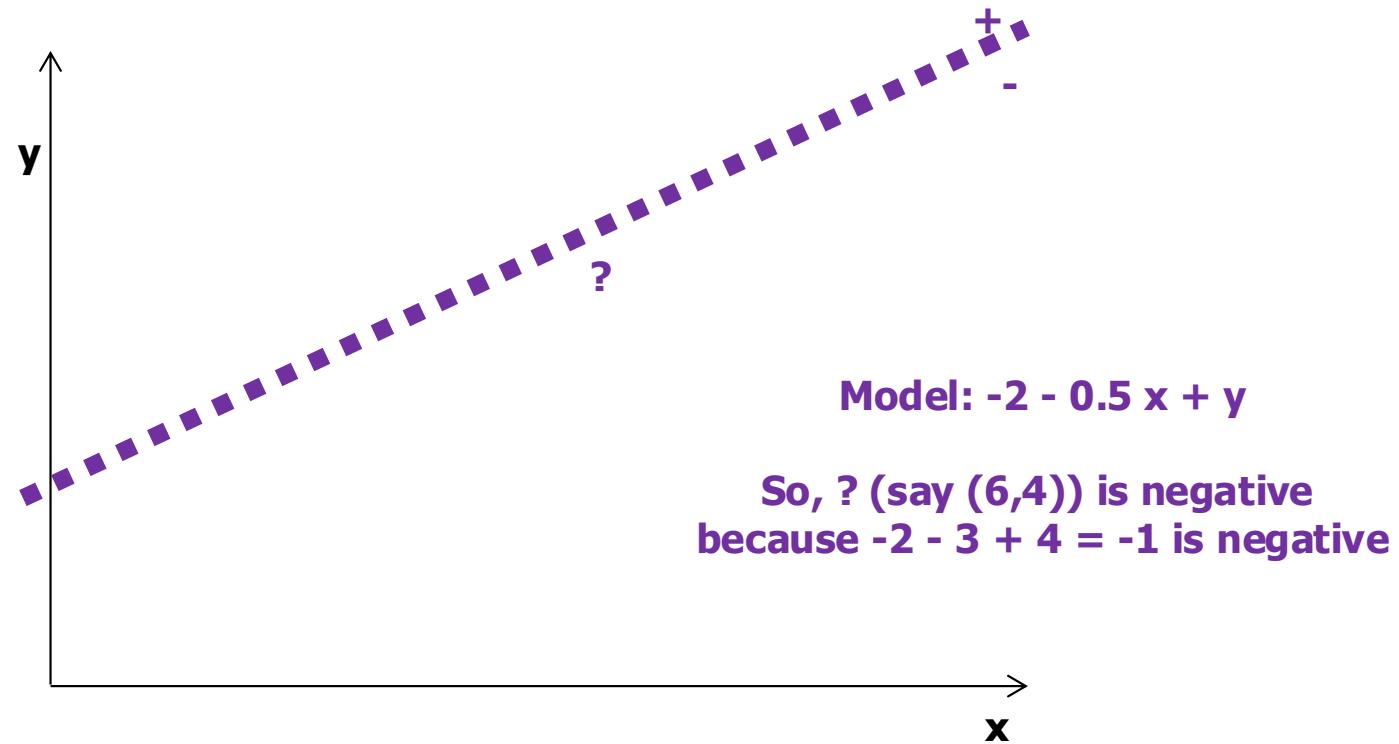
Explaining an Opaque model - LIME



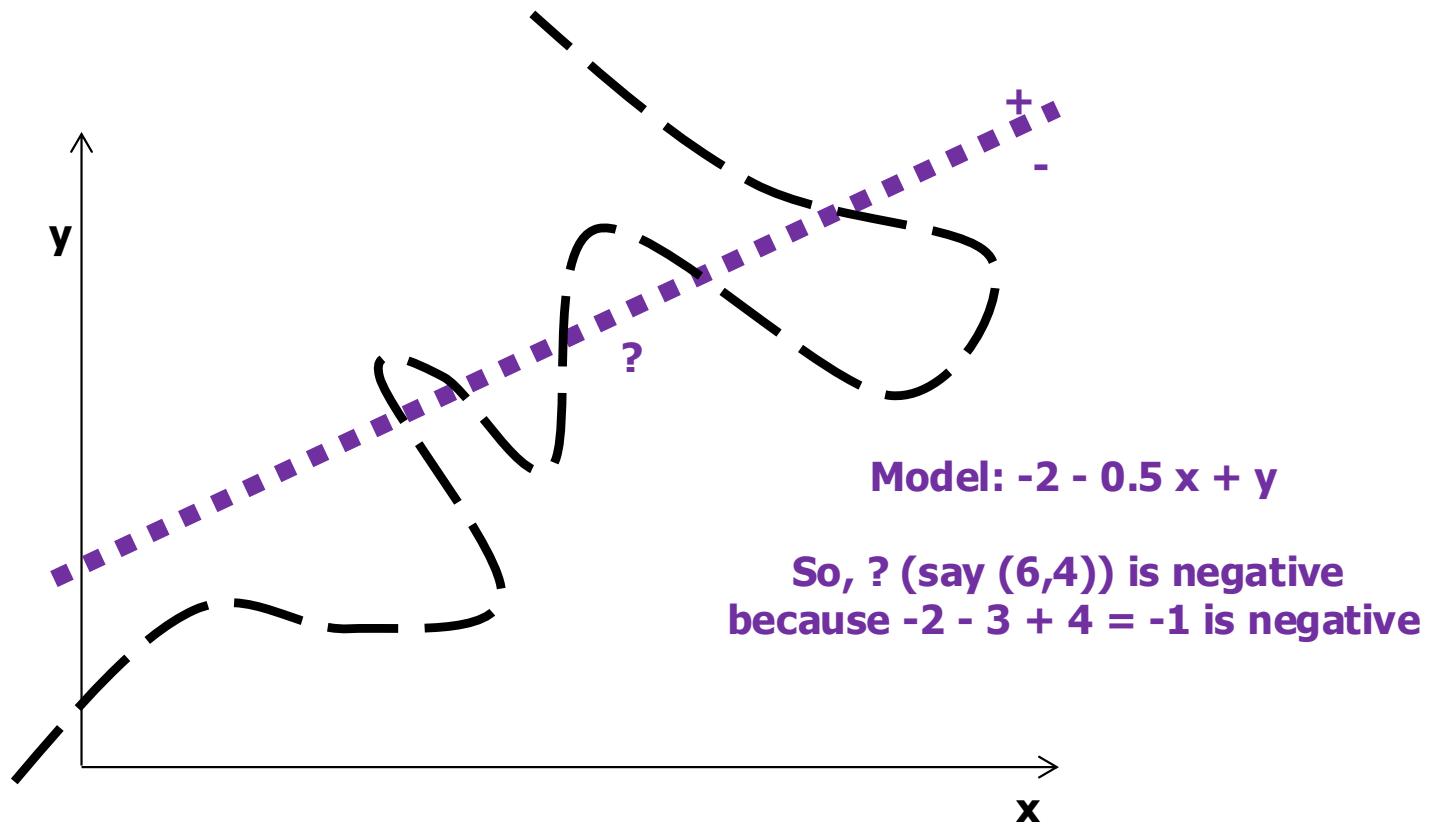
Explaining an Opaque model - LIME



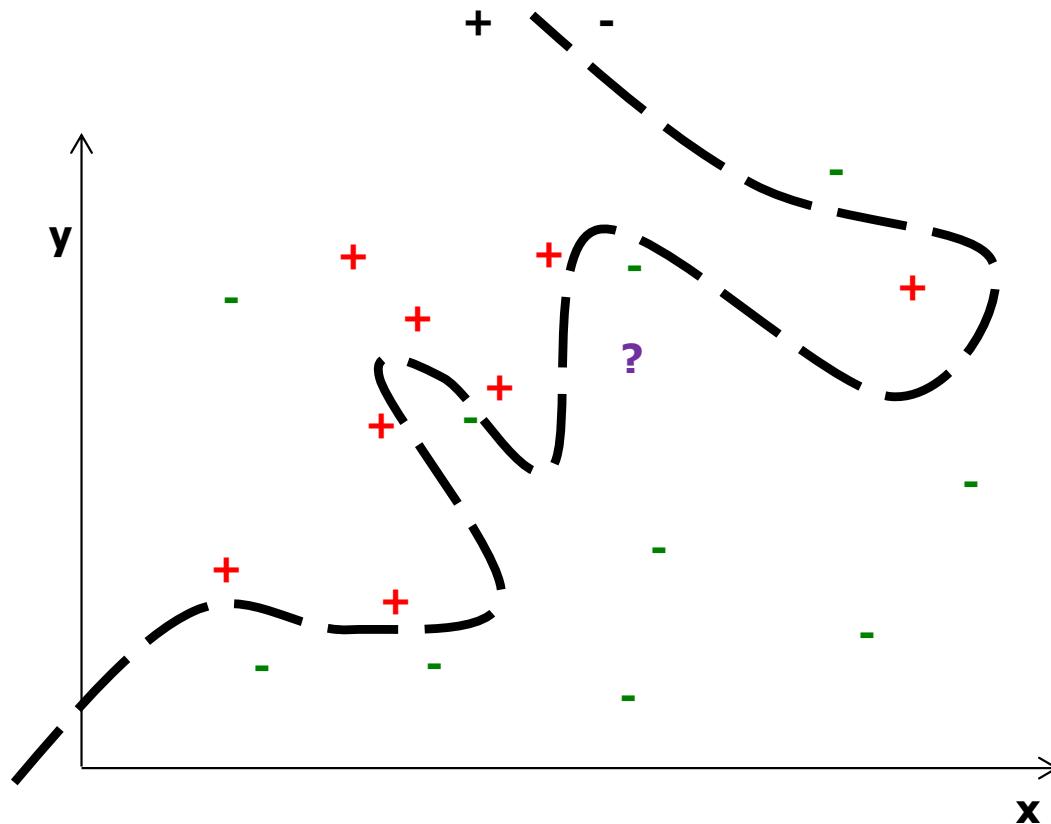
Explaining an Opaque model - LIME



Explaining an Opaque model - LIME

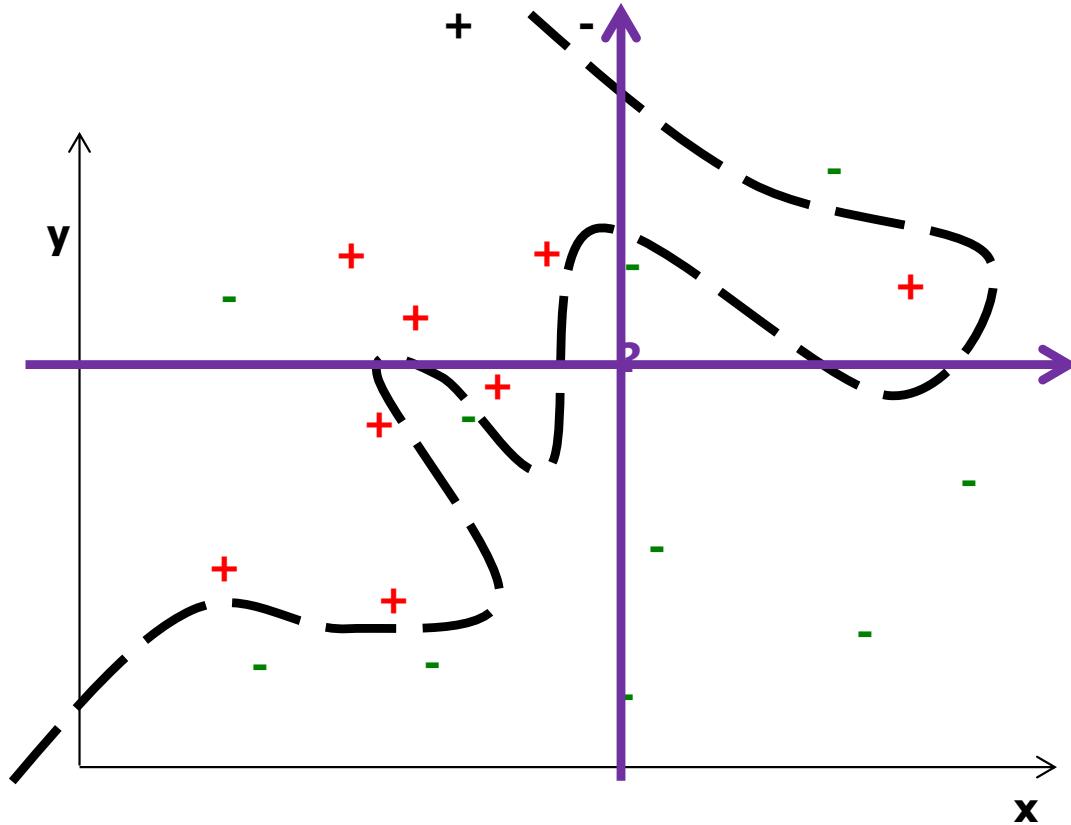


Explaining an Opaque model - SHAP



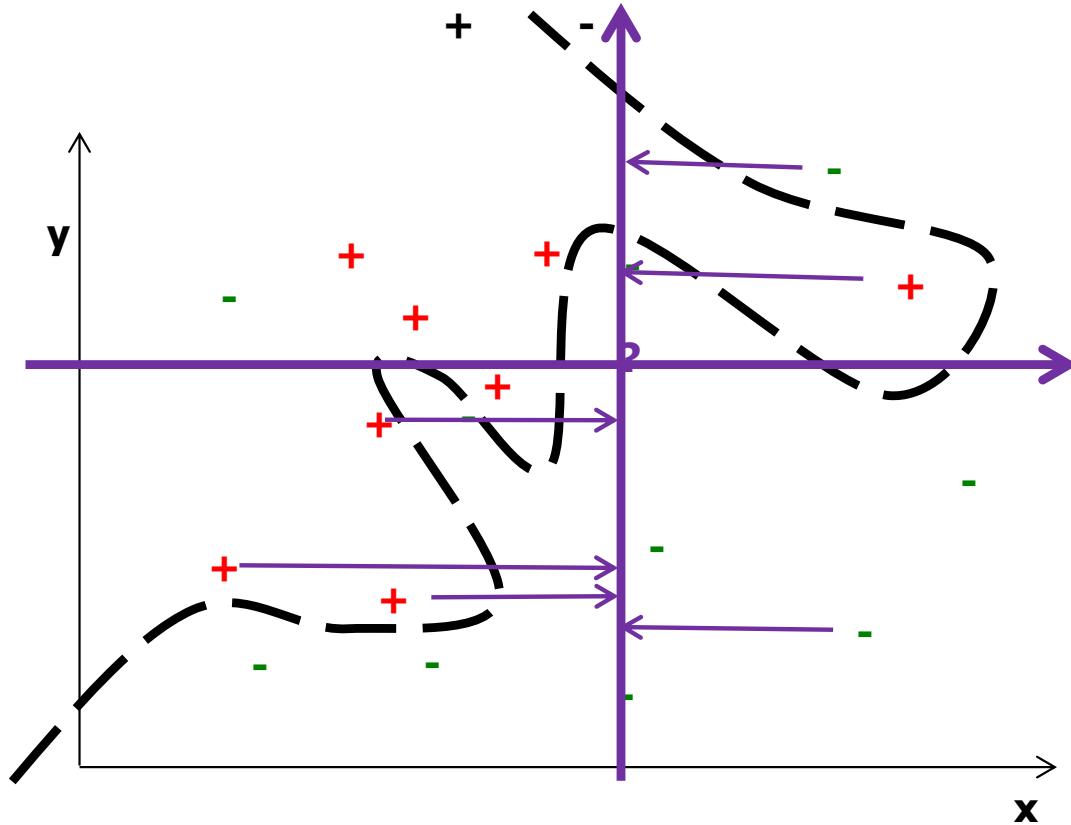
How much does each feature contribute?

Explaining an Opaque model - SHAP



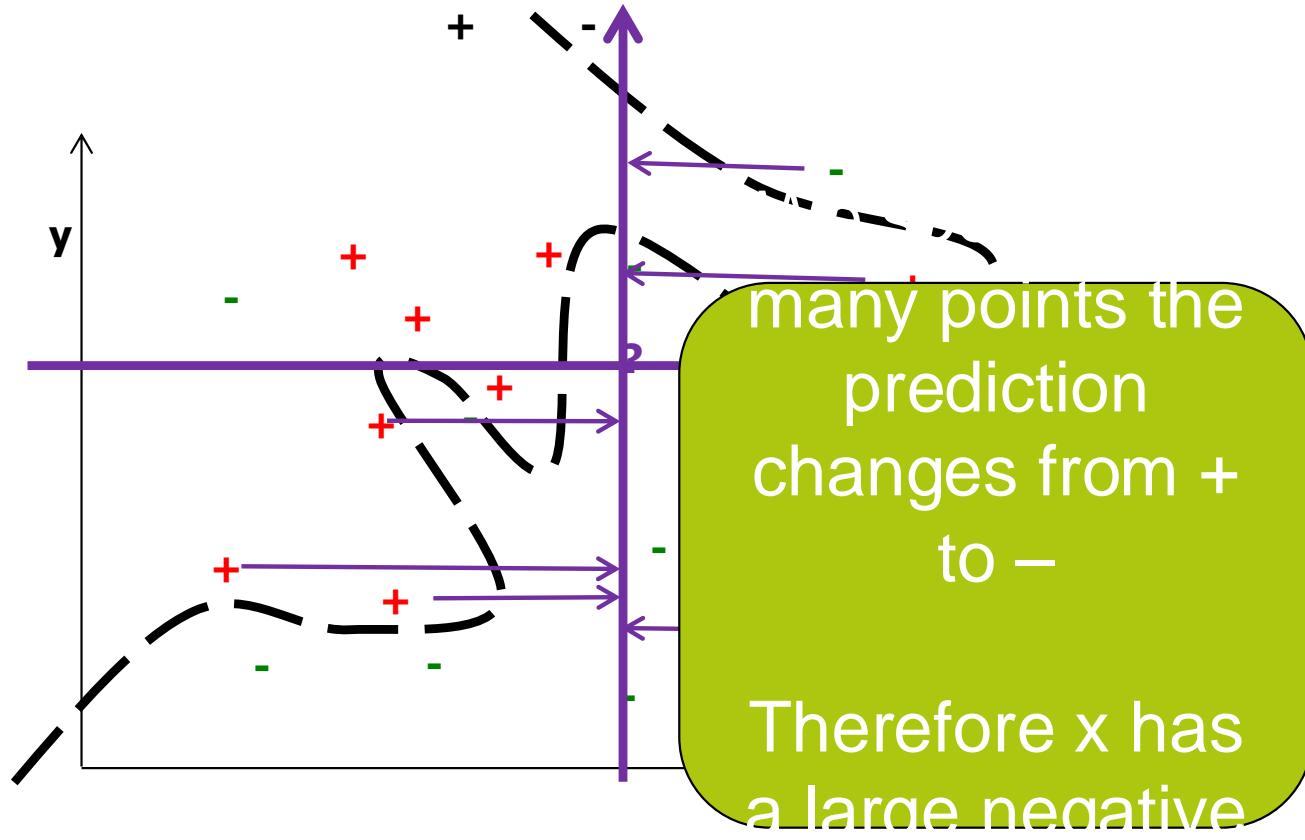
How much does each feature contribute?

Explaining an Opaque model - SHAP

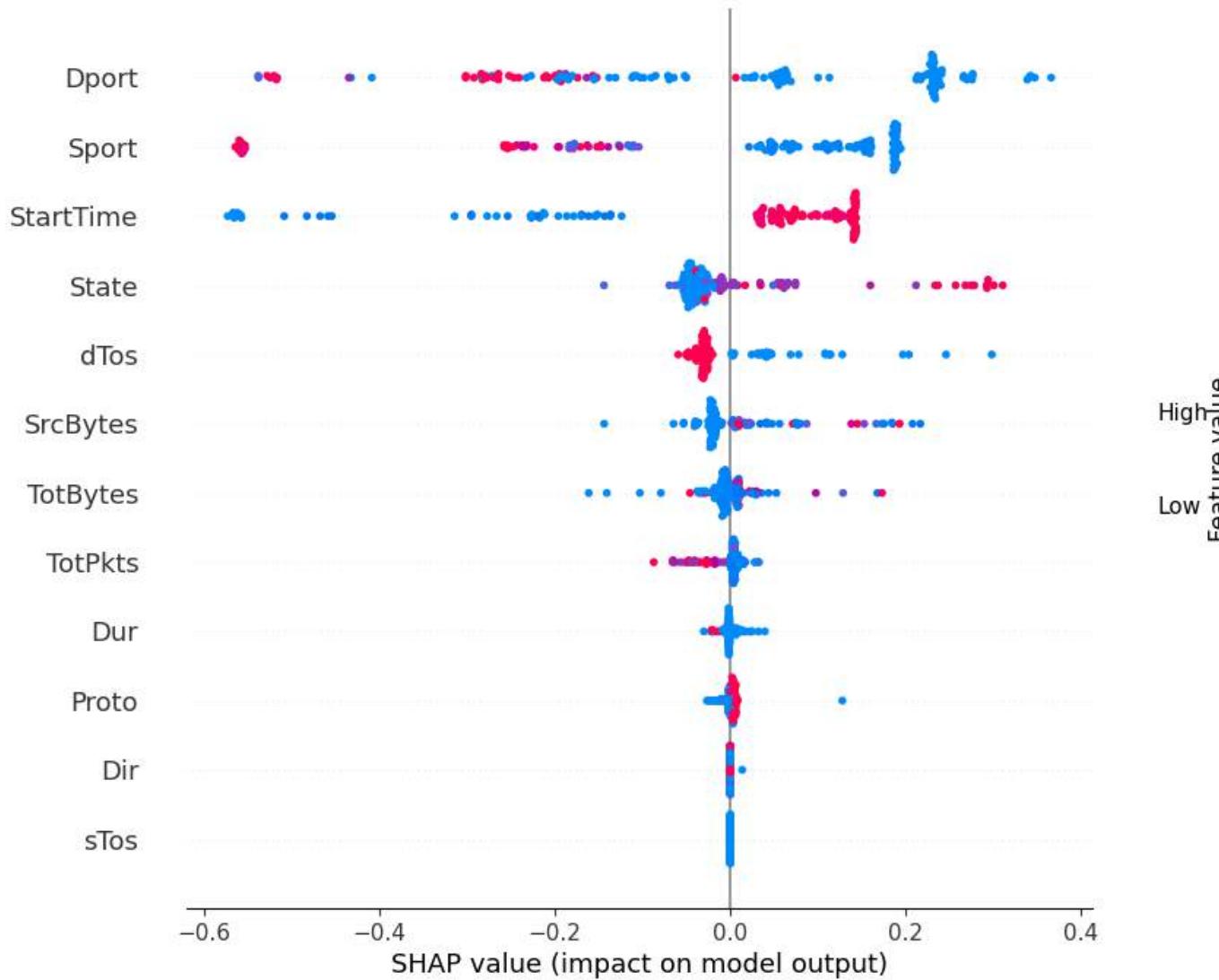


How much does each feature contribute?

Explaining an Opaque model - SHAP



SHAP



LIME

Feature	Value	LIME Rule	Weight
Dport	3389	Dport = 3389	0.18
StartTime	1313571534	1313537772.00 < Start...	0.13
Sport	4505	Sport=4505	0.09
TotPkts	10	TotPkts > 4.00	0.07
State	16	State=16	0.04
Proto	0	Proto=0	0.03
SrcBytes	437	SrcBytes > 186.0	0.03
TotBytes	1076	TotBytes > 494.25	0.02
Dir	2	Dir = 2	0.01
Dur	60.95	Duration > 9.01	0.01

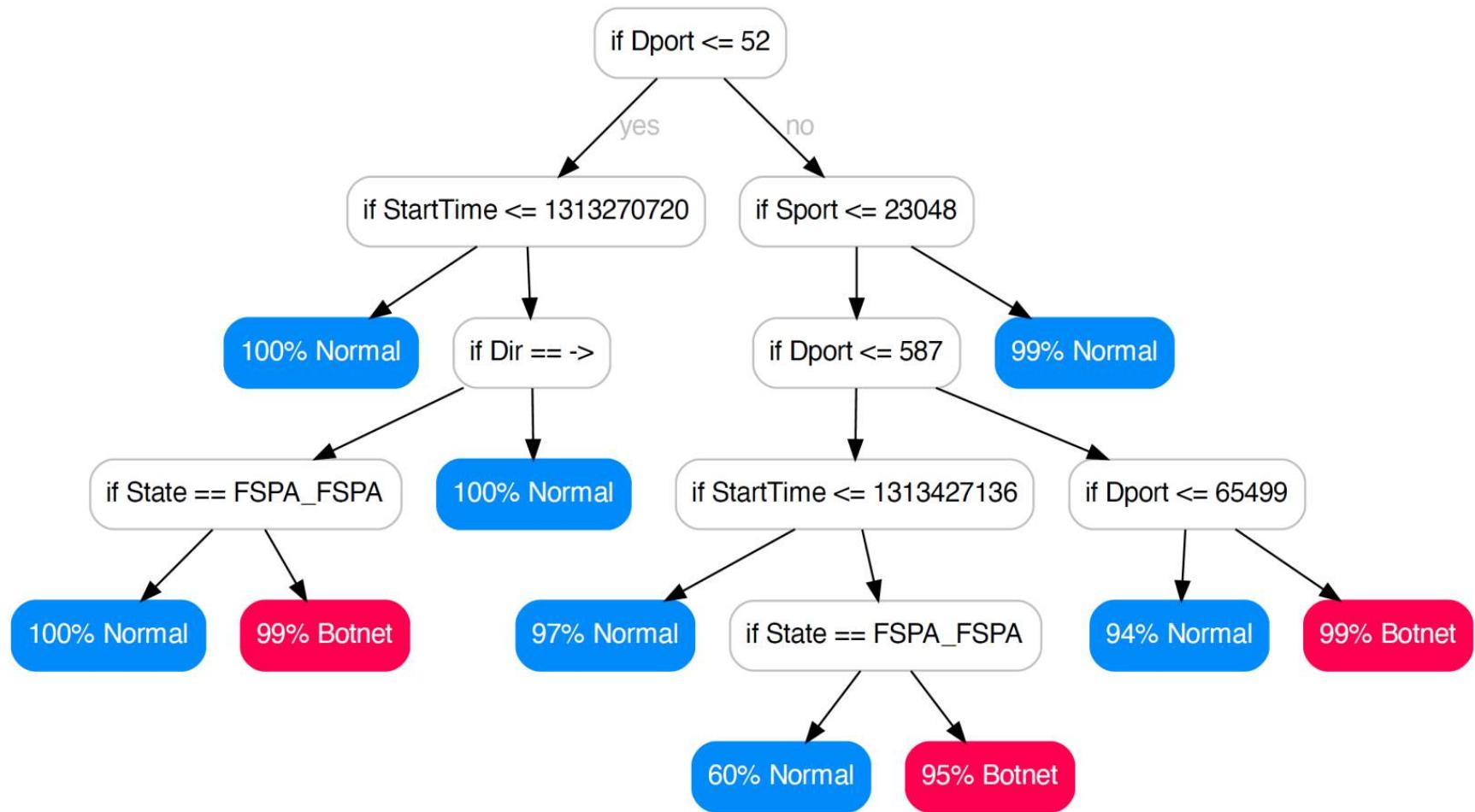
Explanations often disagree

Feature	SHAP Value
State = 54	0.2339
SrcBytes = 186	-0.1756
StartTime = 1313593252	-0.1210
Dport = 80	0.1121
Sport = 1703	-0.1113
dTos = 0	0.0465
TotPkts = 8	0.0408
TotBytes = 492	-0.0295
Proto = 0	0.0266
Dur = 8.96	-0.0247
Dir = 2	0.0
sTos = 0	0.0

Feature	Value	LIME Rule	Weight
Sport	1703	Sport=1703	0.17
StartTime	1313593252	1313537772.00 < Start...	0.14
Dport	80	Dport = 80	0.12
TotPkts	8	TotPkts > 4.00	0.05
Proto	0	Proto=0	0.03
TotBytes	492	271.50 < TotBytes <= 4...	0.02
State	54	State=54	0.01
Dir	2	Dir = 2	0.01
Dur	8.96	0.13 < Dur <= 9.01	0.01
SrcBytes	186	83.50 < SrcBytes <= 1...	0.0

LIME and SHAP explain the same prediction...

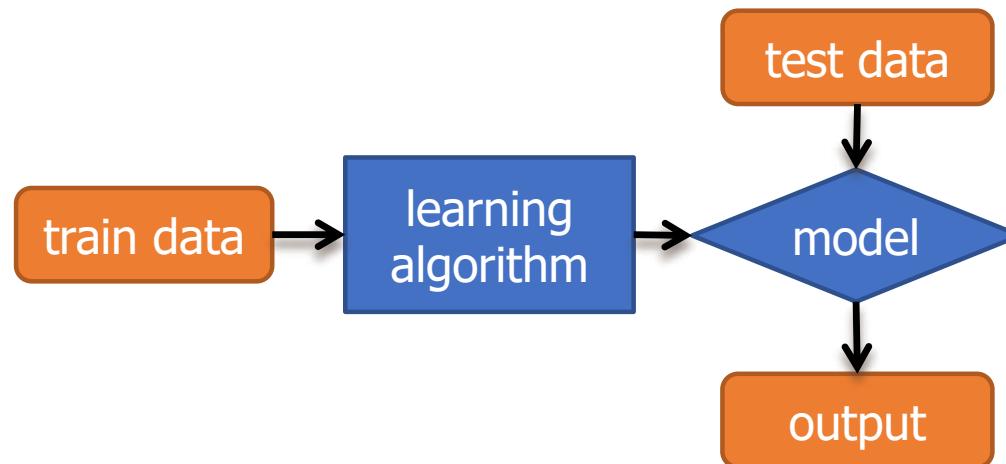
A Decision Tree (Transparent)



CSE2525 Embeddings

Data Mining vs. Machine learning

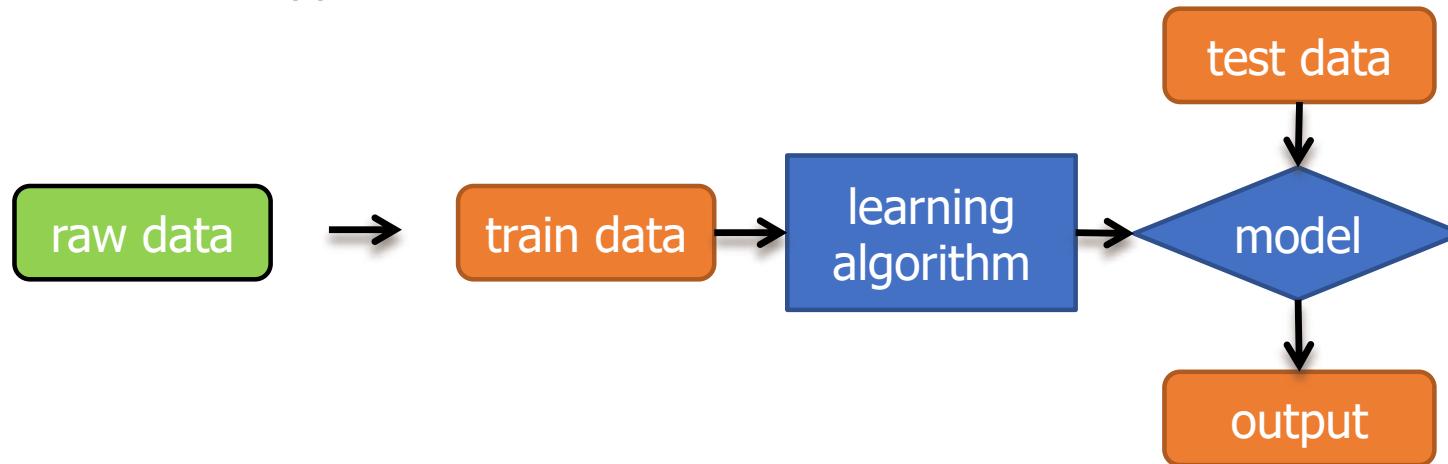
- Classic ML Approach:



- take a huge data set
- compute features
- train a classifier
- deploy the classifier on test

Data Mining vs. Machine learning

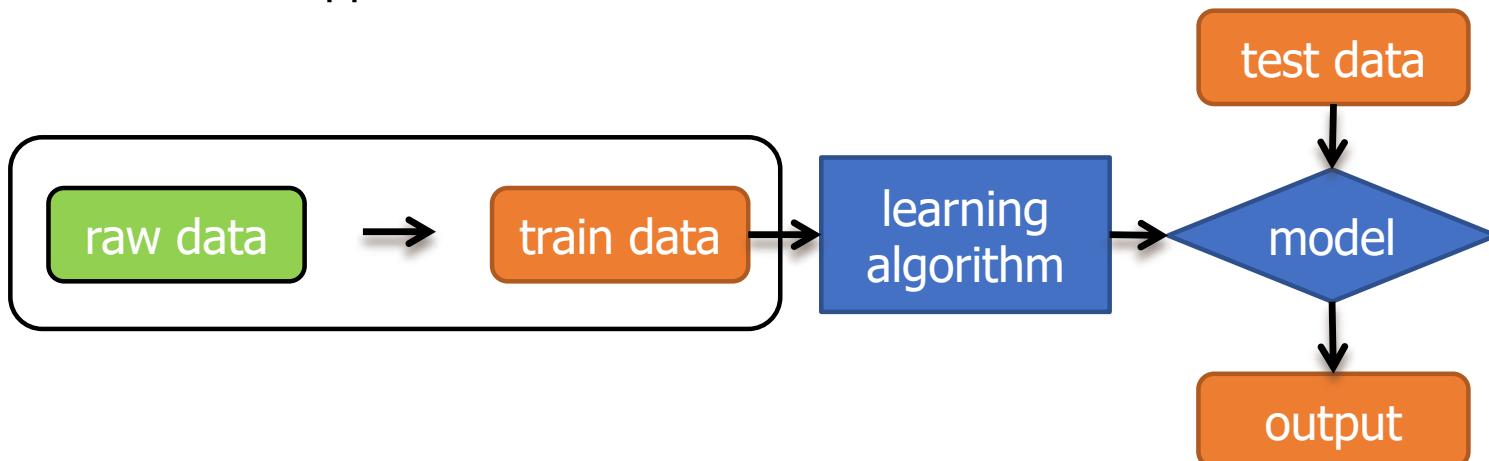
- Classic ML Approach:



- take a huge data set
- **compute features**
- train a classifier
- deploy the classifier on test

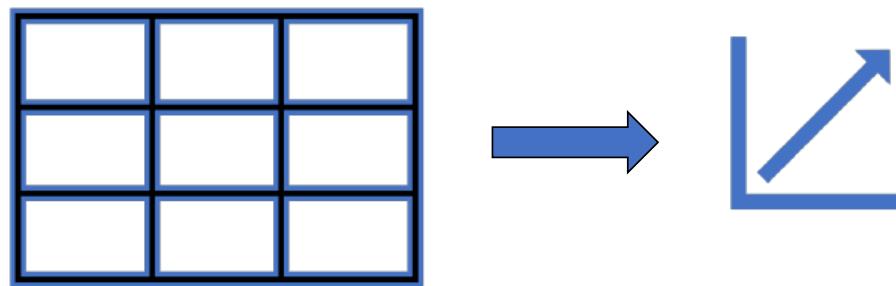
How do we vectorize our input data ?

- Classic ML Approach:

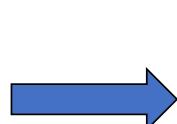
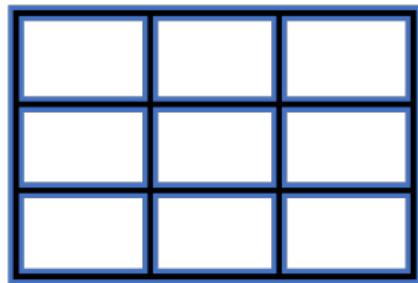


- **Raw datatypes:** tabular, graph, text,

Tabular Data – What is the input instance ?



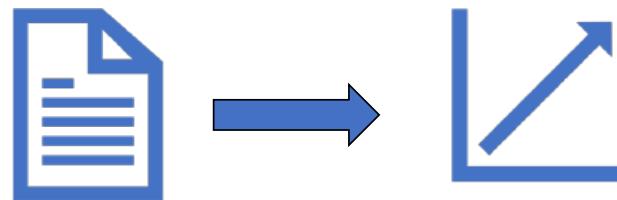
Tabular Data



Each column is a dimension

Each row is an instance vector

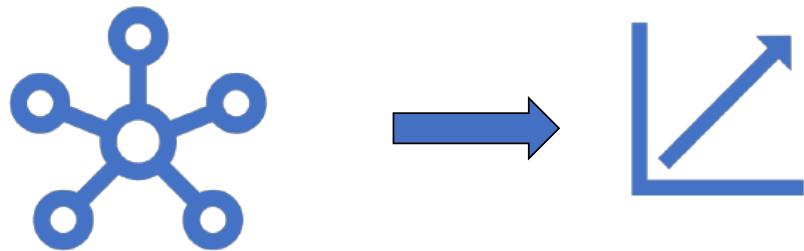
Text Data



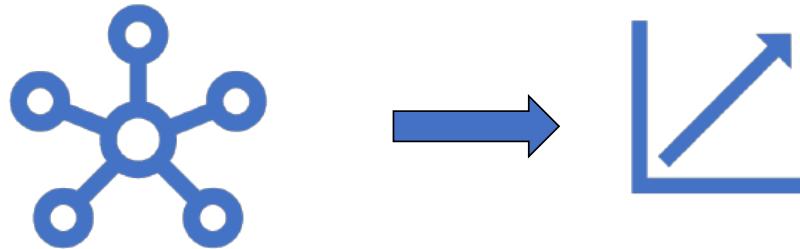
Each word is a dimension

Bag of words: word frequency is value of each dim.

Graph Data



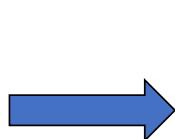
Graph Data



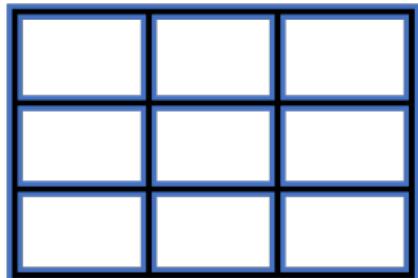
Each vertex is a dimension

a dimension is set or 1 if you have a edge with the vertex

Graph Data



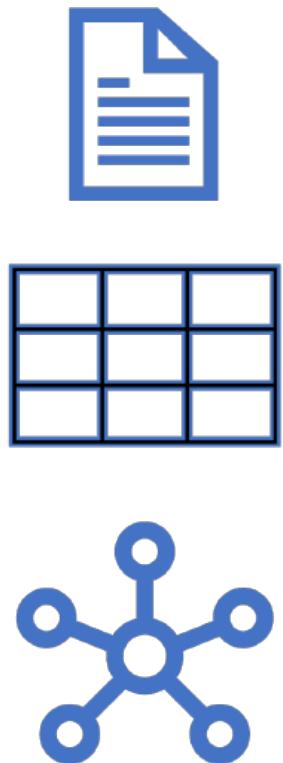
Each vertex is a dimension



a dimension is set or 1 if you have a edge with the vertex

One vertex is a row in a adjacency matrix

Typical feature space



Sparse,
Large,
Non-semantic,
Hand crafted

But what is the problem ?

CSE2525 Embeddings

[Word Embeddings](#)

Issues with sparse representations

- Distances in vector spaces are misleading – high dimensionality
- Vocabulary mismatch
- Large space requirements
- Large time requirements (most of the times)

banana 

mango 

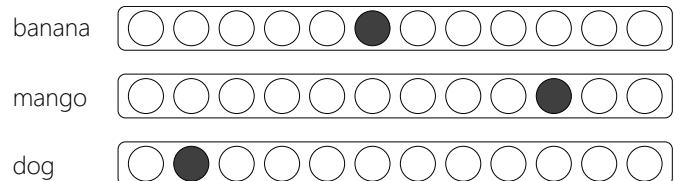
dog 

Local representations (1-hot)

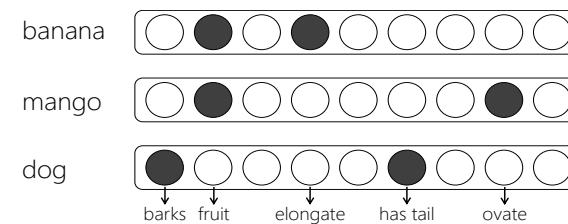
$$(w^{hotel})^T w^{motel} = (w^{hotel})^T w^{cat} = 0$$

Issues with sparse representations

- Distances in vector spaces are misleading – high dimensionality
- Vocabulary mismatch
- Large space requirements
- Large time requirements (most of the times)



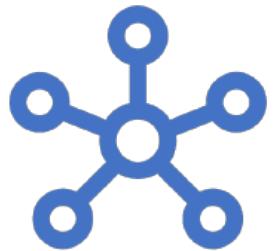
Local representations (1-hot)



Distributed representations

$$(w^{hotel})^T w^{motel} = (w^{hotel})^T w^{cat} = 0$$

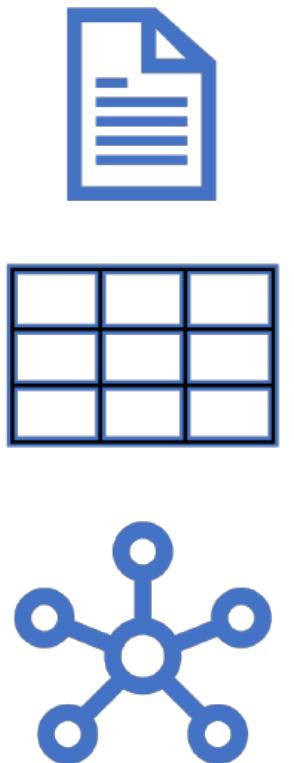
Sparsity in adjacency matrices



Sparse,
Large,
Non-semantic,
Hand crafted

Number of edges <<< vertex

Automatic Feature Extraction

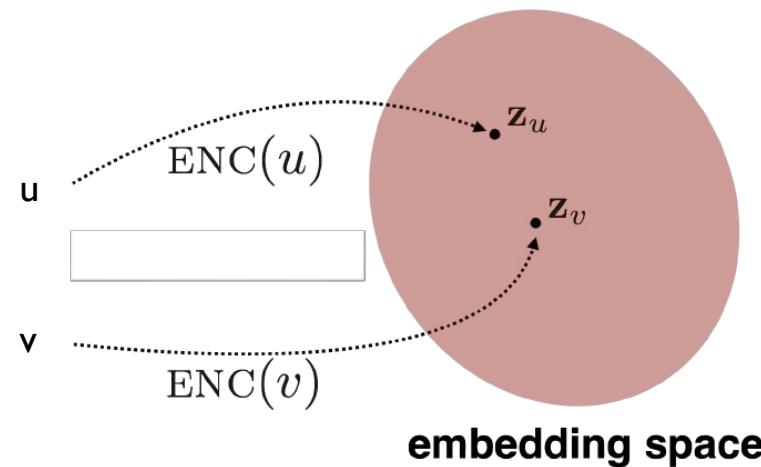
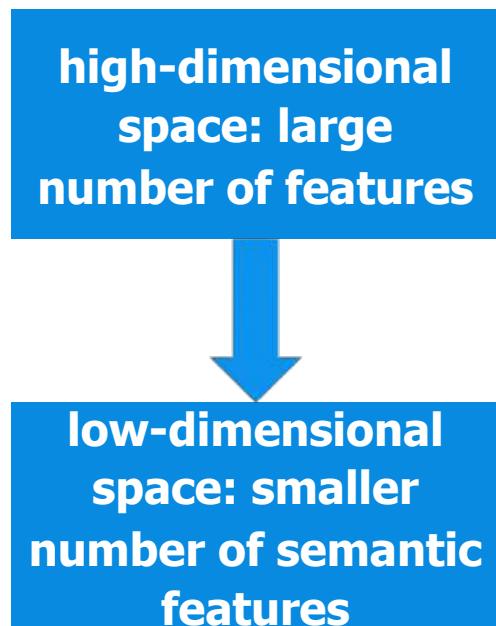


Sparse,
Large,
Non-semantic,
Hand crafted

Dense,
smaller,
Semantic,
Automatic extraction

What we want ?

Learning representations for words is to encode nodes so that **similarity in the embedding space (e.g., dot product)** approximates **similarity in the textual context**



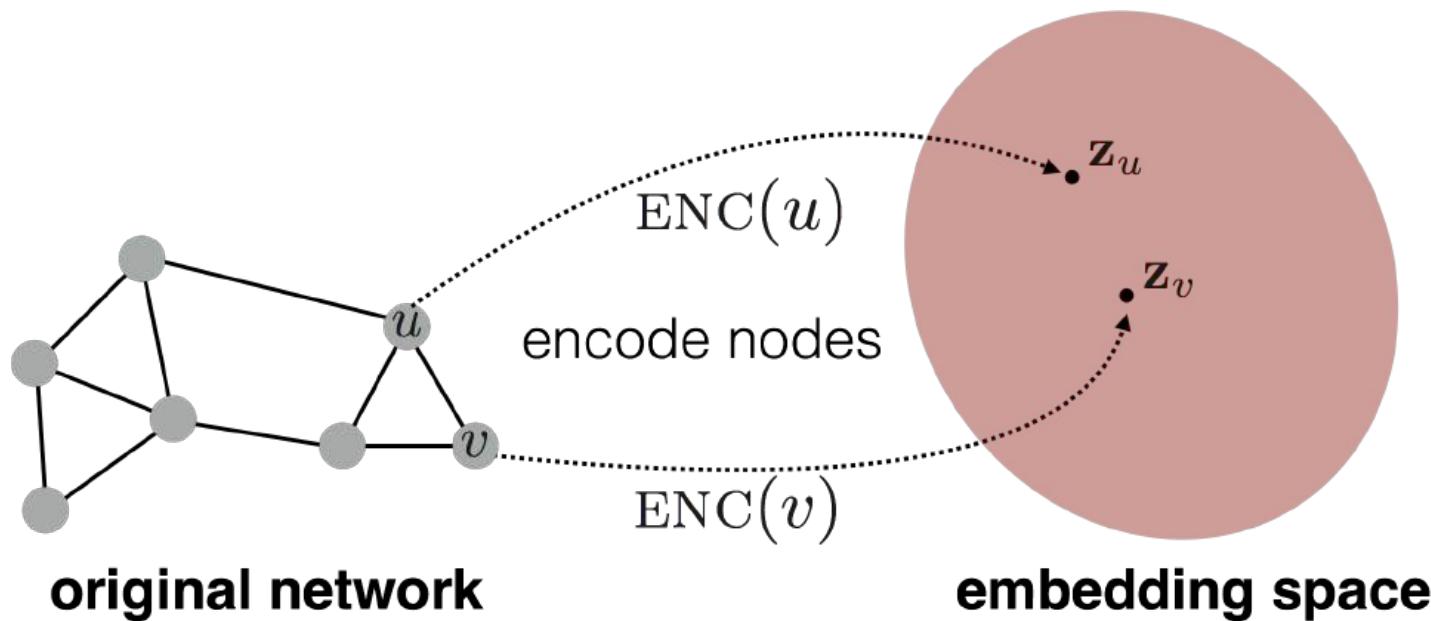
What is context ?

Words that occur in similar contexts have similar meanings

- We took two hours to reach the **top** of the **hill**.
- The **top** of the **hill** had the best view

What we want ?

Learning representations for nodes is to encode nodes so that **similarity in the embedding space (e.g., dot product)** approximates **similarity in the original network**.



Distributional similarity

- “*You shall know a word by the company it keeps.*” – Firth 1957
 - This purple **top** will go well with my blue jeans.
 - We took two hours to reach the **top** of the **hill**.
 - The **top** of the **hill** had the best view
- Words that occur in similar contexts have similar meanings

Word Embeddings: Word2Vec

- Key idea: *context provides meaning*

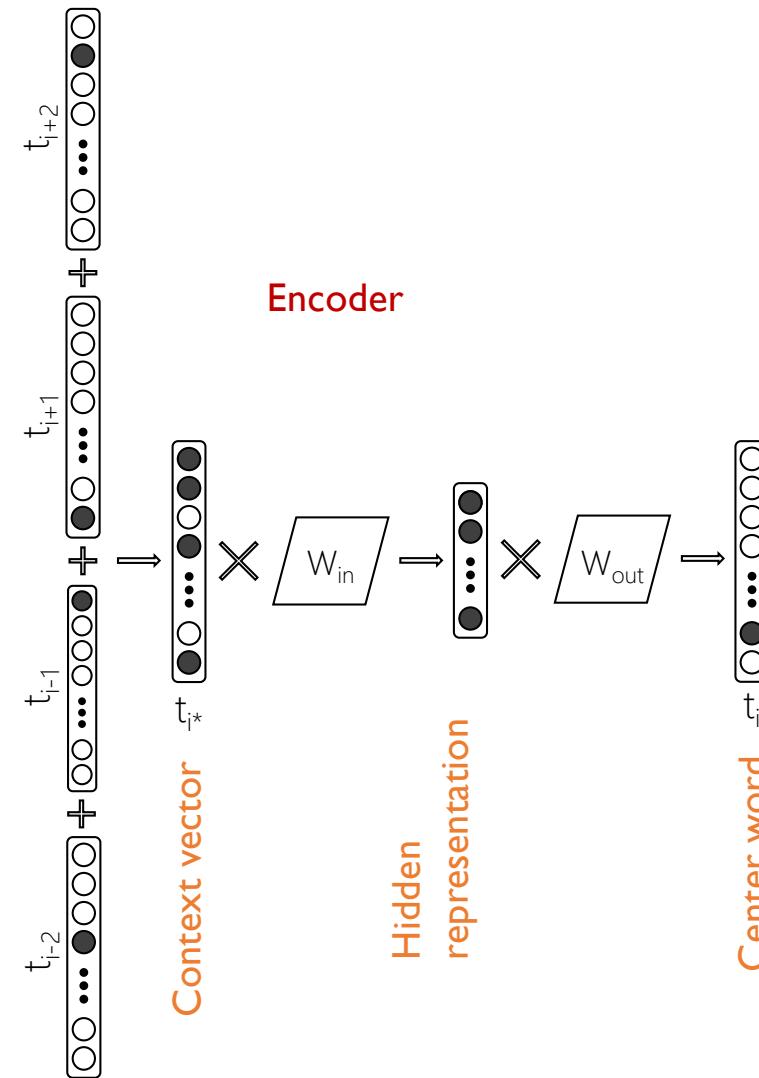
If it walks like a duck and quacks like a duck, it must be a duck

- Contexts:

1. like a __ and quacks
2. like a __ it must
3. be a __

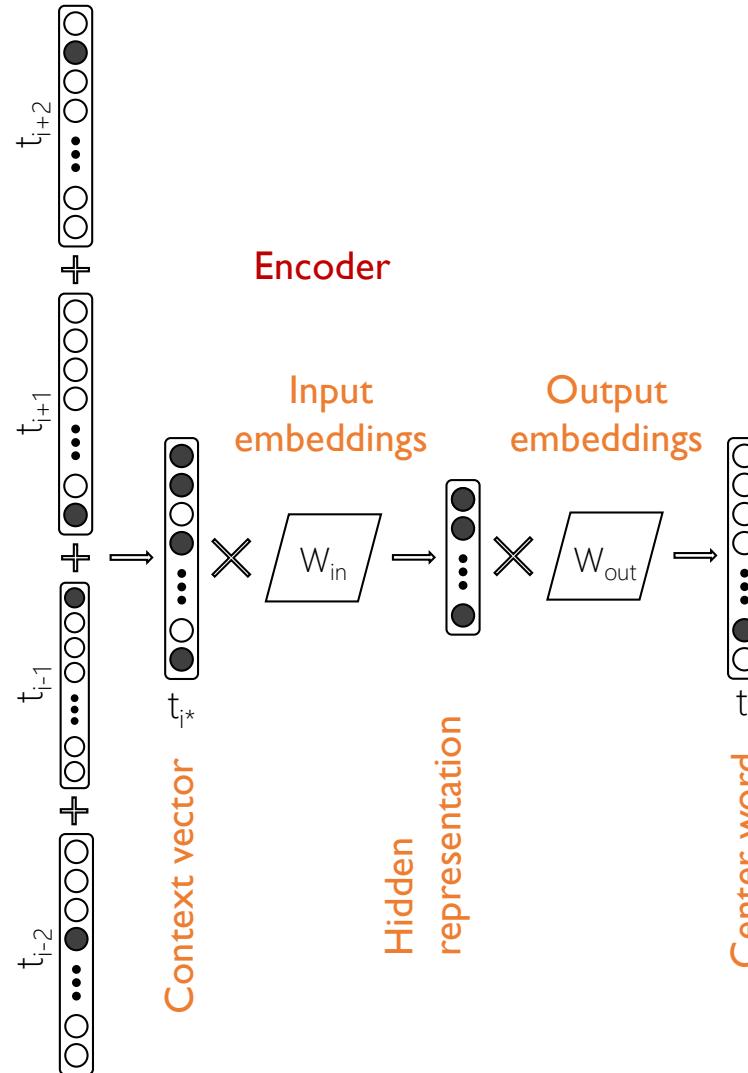
- Q: How to use these contexts?

Word2vec Model

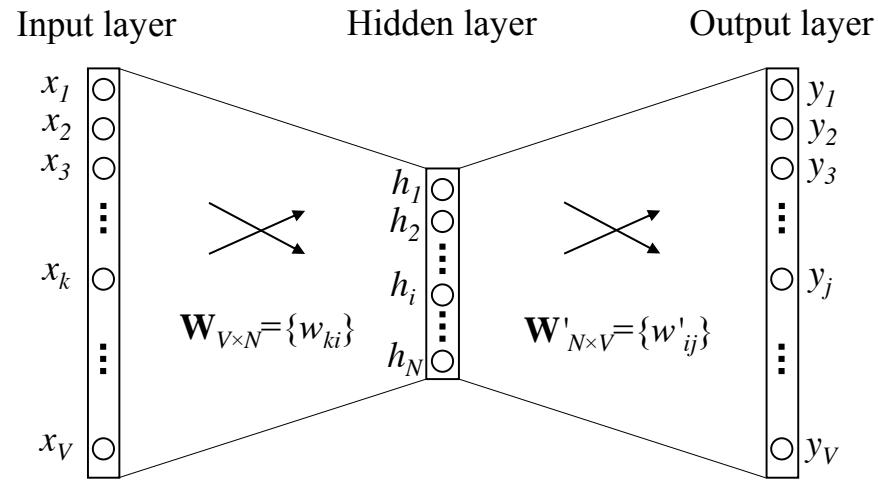


Word2vec Model

- Hidden Layers size is much smaller than the original dimensionality
 - Embeddings – 300
 - Vocab – Millions
- Input and Output dimensions are same size as the vocabulary

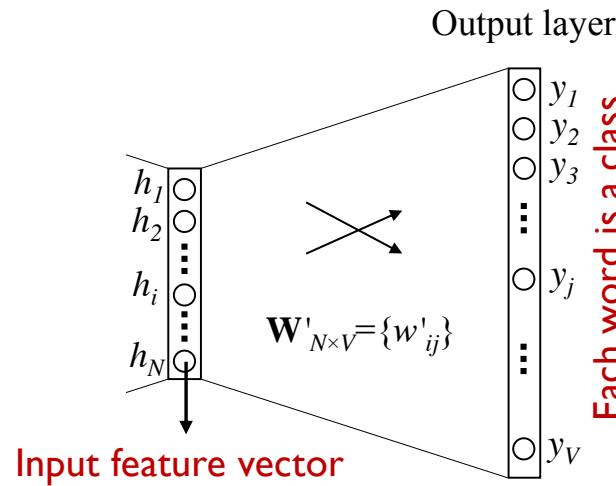


Multi-class classification



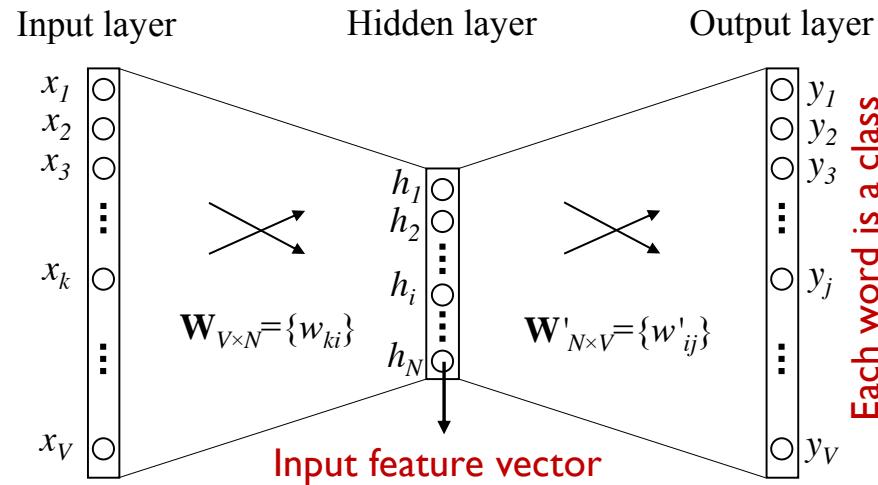
- Consider **<Word, Context>** pairs, and we need to predict the context given the word

Multi-class classification



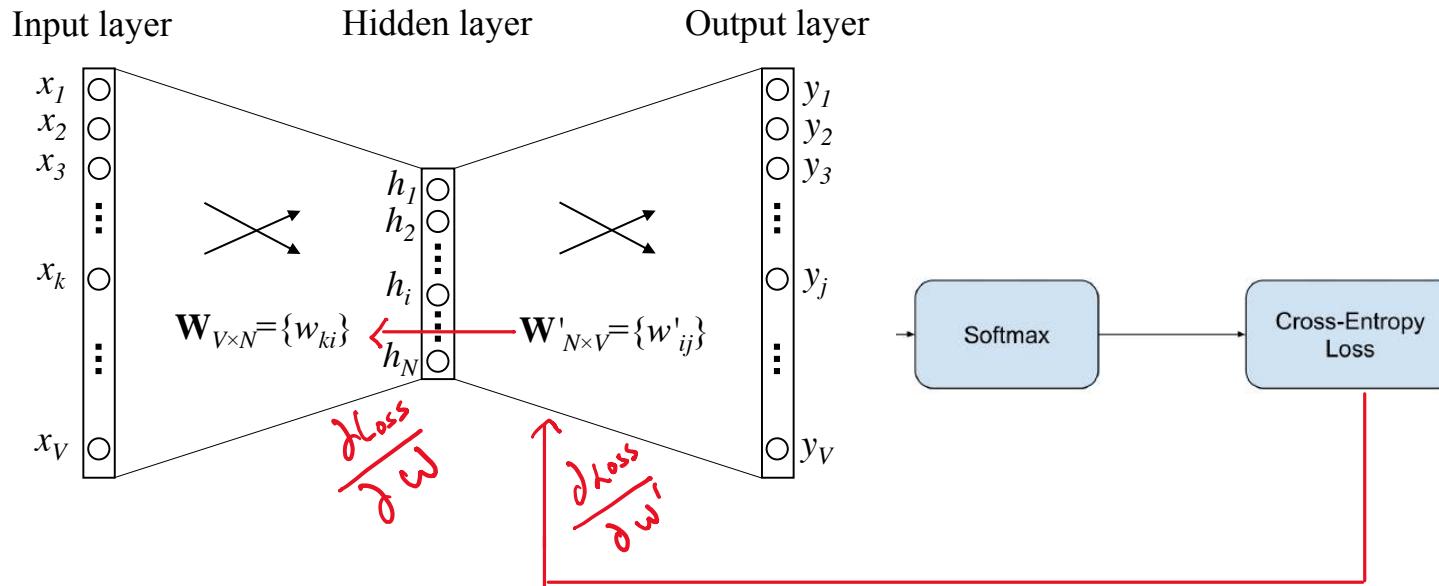
- Consider **<Word, Context>** pairs, and we need to predict the context given the word

How Do we learn the params ?

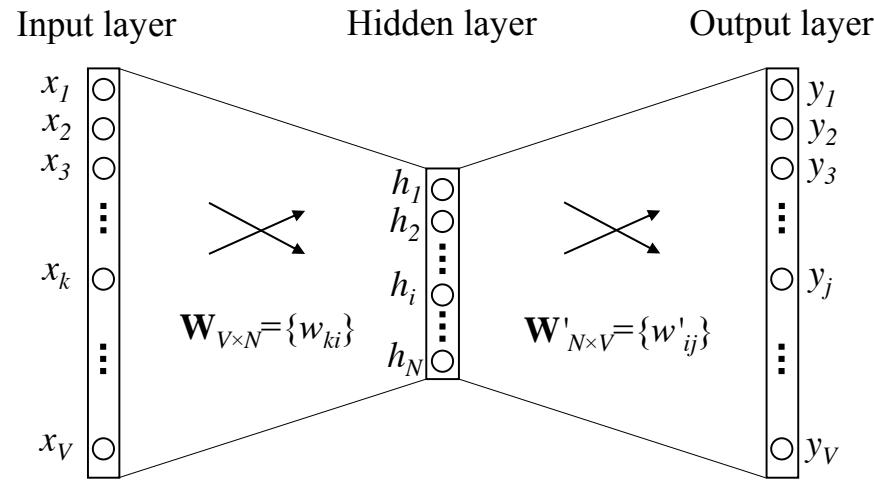


- Consider **<Word, Context>** pairs, and we need to predict the context given the word

Training using gradient descent

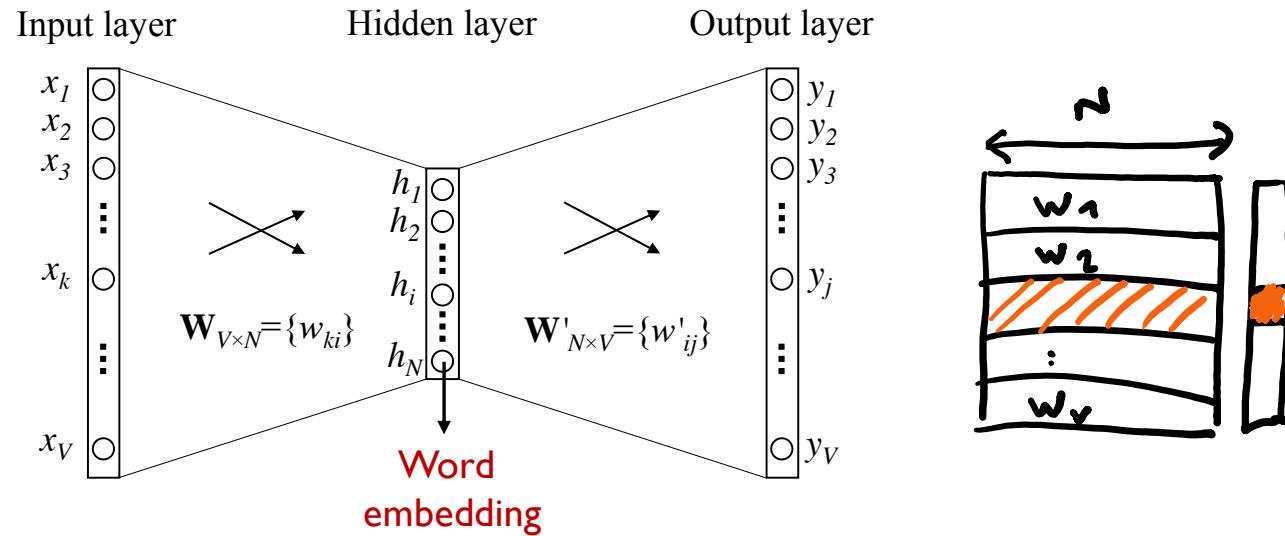


Where are the embeddings ?



- Consider **<Word, Context>** pairs, and we need to predict the context given the word
- Where are the embeddings ?

Parameters as Embeddings



- Consider **<Word, Context>** pairs, and we need to predict the context given the word
- Where are the embeddings in this model ?

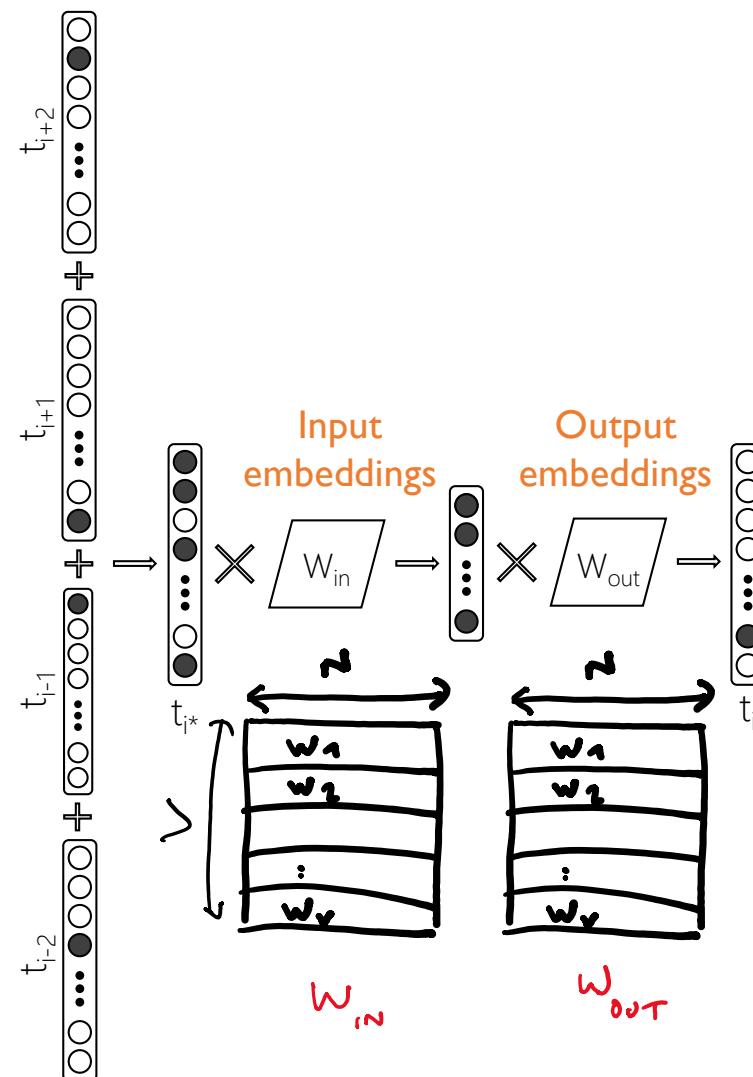
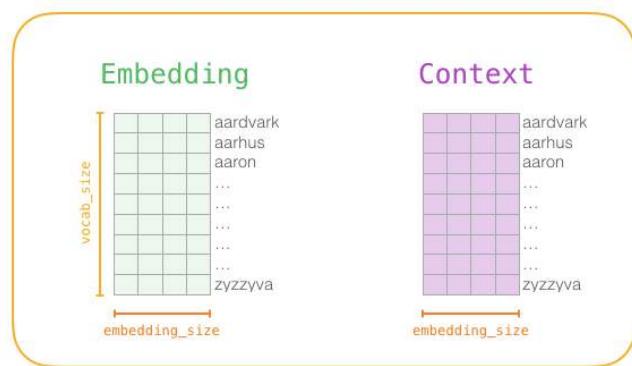
Negative Sampling (not exam material)

- Computing the denominator of softmax is expensive (V is typically large)
- Sample a limited set of negative words for updating
- Update equation only words in the context and the negative samples

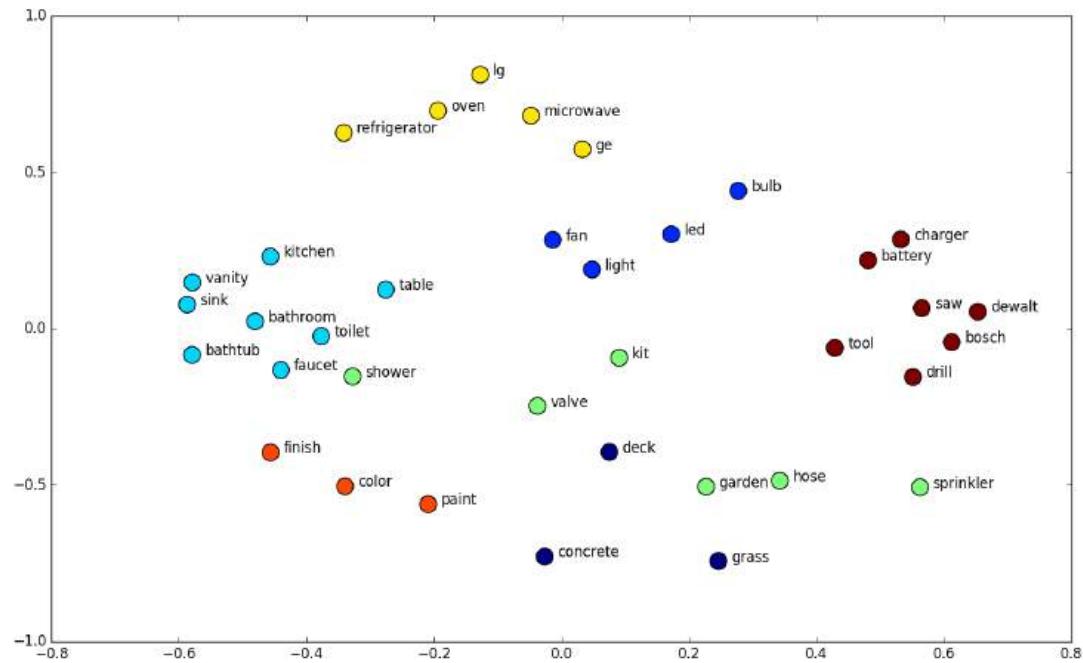
$$\hat{y}_{ij} = \underset{j}{\text{softmax}}(\vec{w}_\ell \cdot \vec{x}_i) = \frac{e^{\vec{w}_j \cdot \vec{x}_i}}{\sum_{\ell=1}^C e^{\vec{w}_\ell \cdot \vec{x}_i}}$$

Instead of all N -words replace
by C random words

Word2vec Embeddings



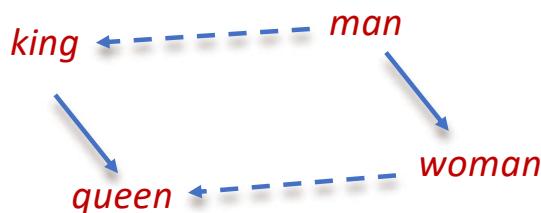
Examples



Libraries exist: <https://github.com/piskvorky/gensim>

Geometry of embeddings

- $\text{vector}(\text{'king'}) - \text{vector}(\text{'man'}) + \text{vector}(\text{'woman'}) = \text{vector}(\text{'queen'})$

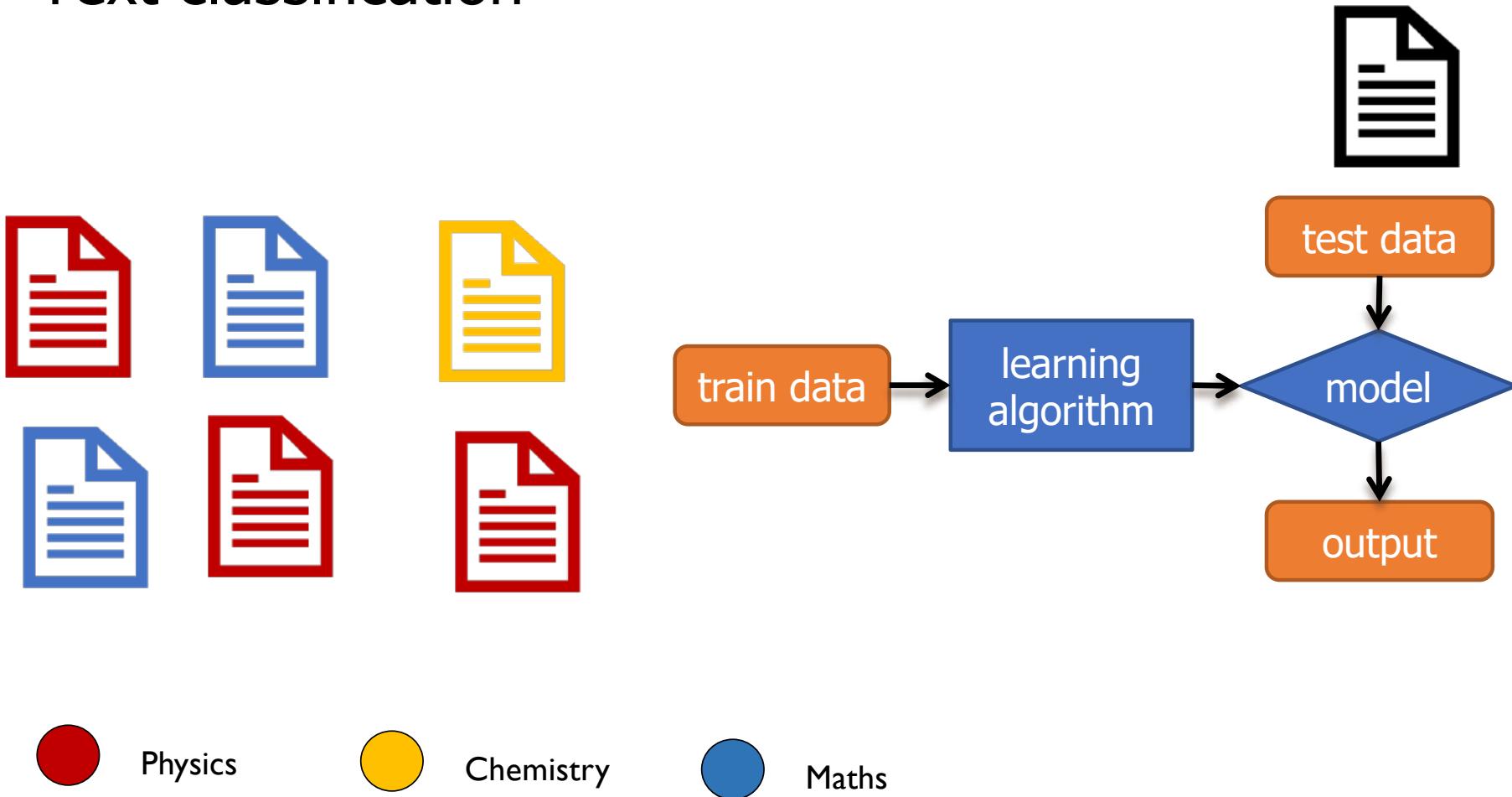


- Often pre-computed and are easily available

CSE2525 Applying Embeddings for Supervised Tasks

Word Embeddings

Text classification



From word to representing documents

Each word is a vector



Bag of Words



Document represented as the **average** or **sum** of all the words contained

Many implementations exist

Library: Gensim -- <https://github.com/piskvorky/gensim>

```
from gensim.test.utils import datapath
from gensim import utils

class MyCorpus:
    """An iterator that yields sentences (lists of str)."""

    def __iter__(self):
        corpus_path = datapath('lee_background.cor')
        for line in open(corpus_path):
            # assume there's one document per line, tokens separated by whitespace
            yield utils.simple_preprocess(line)
```

```
import gensim.models

sentences = MyCorpus()
model = gensim.models.Word2Vec(sentences=sentences)
```

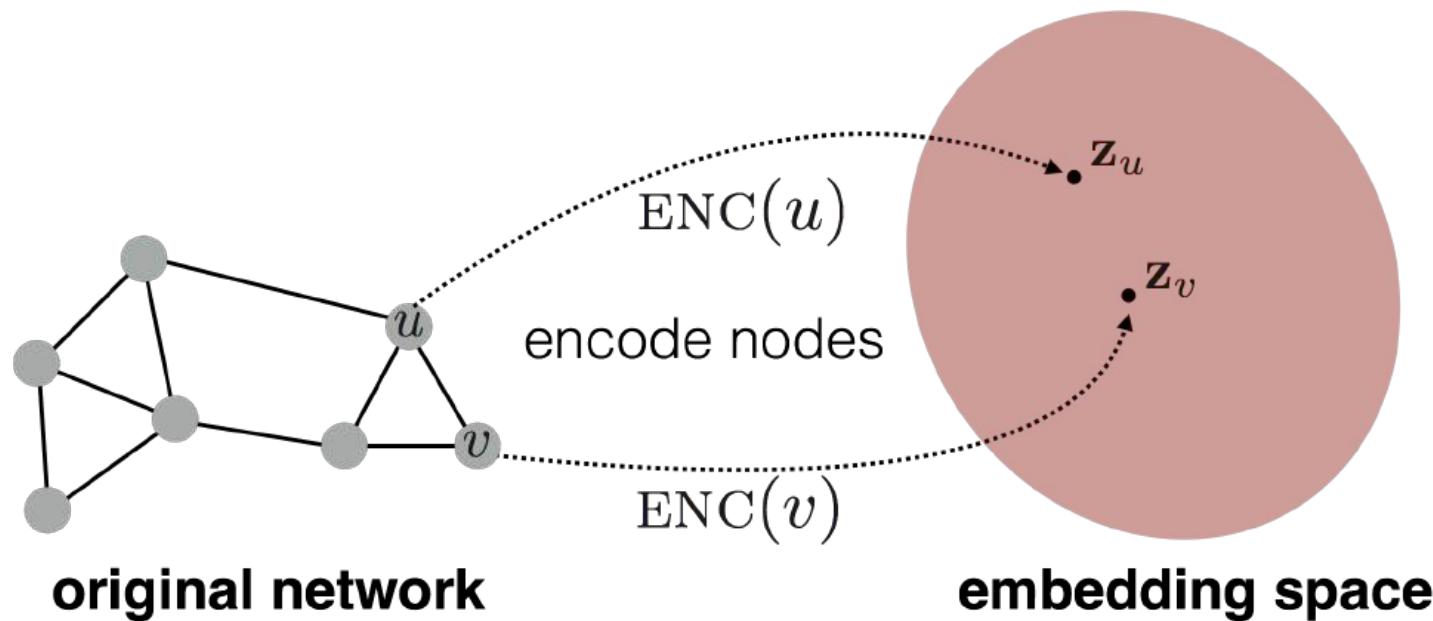
```
vec_king = model.wv['king']
```

CSE2525 Embeddings

Graph Embeddings

What we want ?

Learning representations for nodes is to encode nodes so that **similarity in the embedding space (e.g., dot product)** approximates **similarity in the original network**.



Node Embeddings: Motivation from Word2vec

frequent words often provide little information

- What if we define a context of similarity like in language?
- Use **Random Walks**

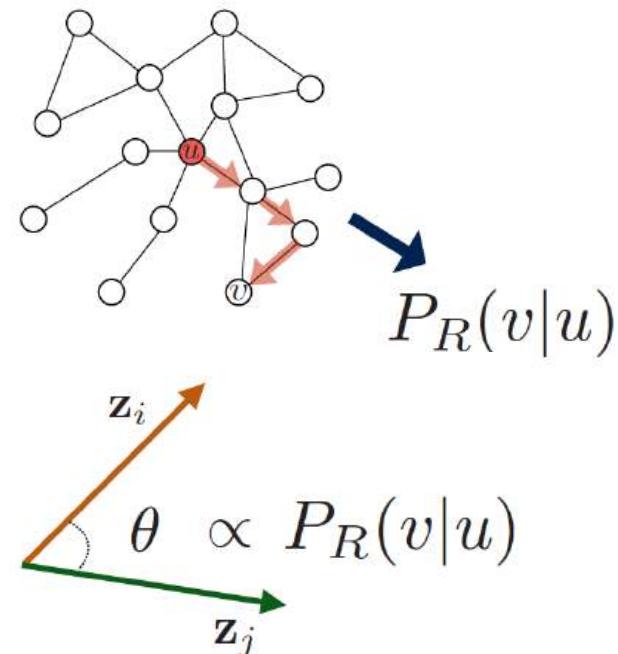
Random-walk Embeddings

$$\mathbf{z}_u^\top \mathbf{z}_v \approx$$

probability that u and
 v co-occur on a
random walk over the
network

Random-walk Embeddings

1. Estimate probability of visiting node v on a random walk starting from node u using some random walk strategy R .
2. Optimize embeddings to encode these random walk statistics.



Putting Things Together (Lab 3 will be based on this)

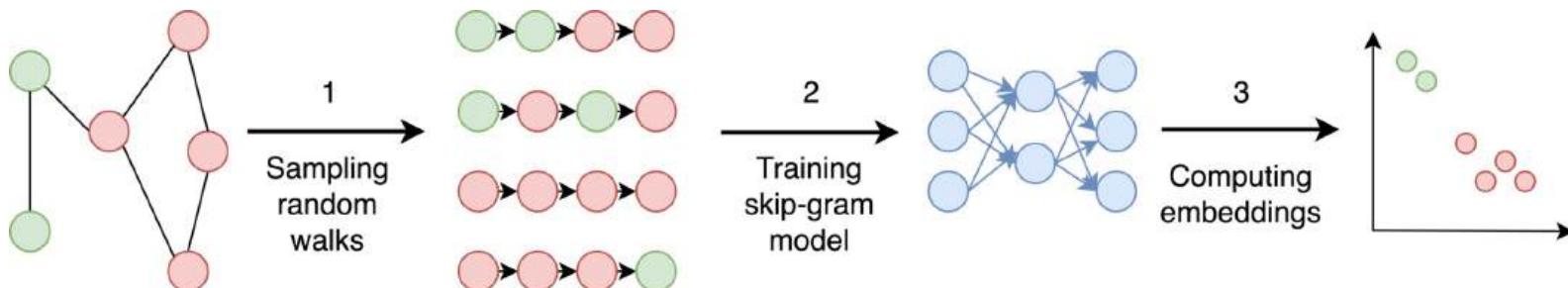
Two basic stages:

1) Create training data

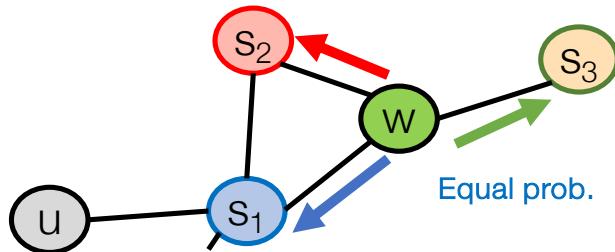
- Use random walks – **fixed window, biased**

2) Training or Optimization

- Given an input node train to predict RW nodes – **Skip Gram with Negative sampling**



Fixed Window: Deepwalk

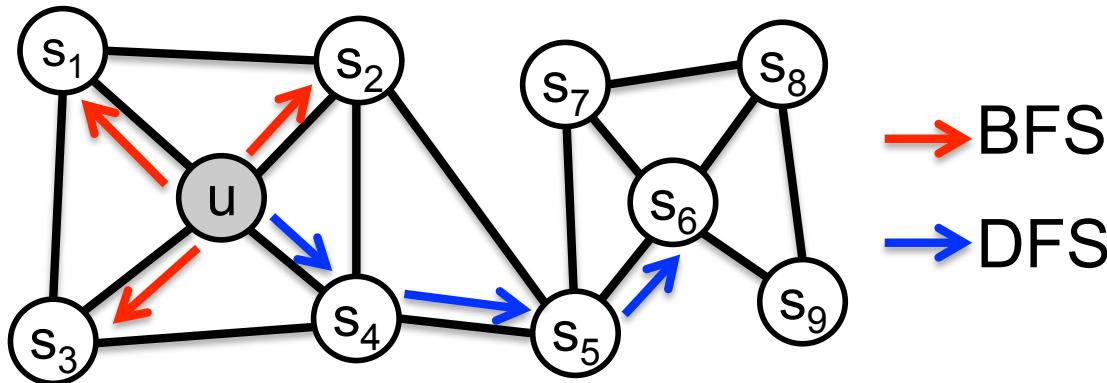


1. Start from a random node
2. Equal probability of going to each of its neighbours
3. Stop after a fixed length of RW

Note: Same node can be sampled multiple times in the same RW

Biased Walks: node2vec

Two classic strategies to define a neighborhood $N_R(u)$ of a given node u :



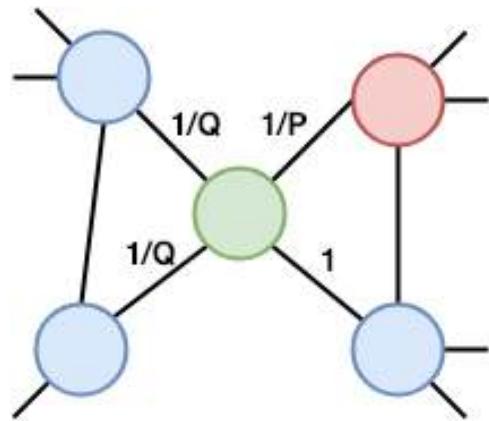
$$N_{BFS}(u) = \{ s_1, s_2, s_3 \}$$

Local microscopic view

$$N_{DFS}(u) = \{ s_4, s_5, s_6 \}$$

Global macroscopic view

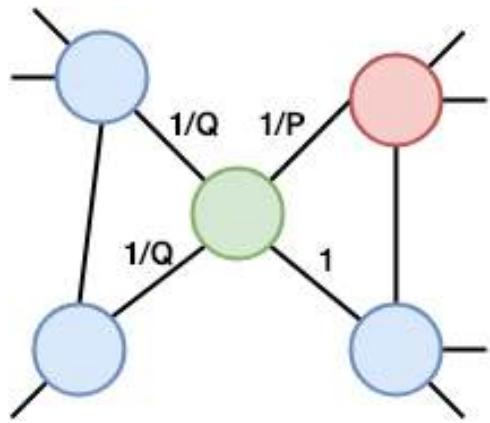
Biased Walks in Node2Vec



Return **parameter p** :
Return back to the previous node

In-out **parameter q** :
Moving outwards (DFS) vs.
inwards (BFS)

Biased Walks in Node2Vec



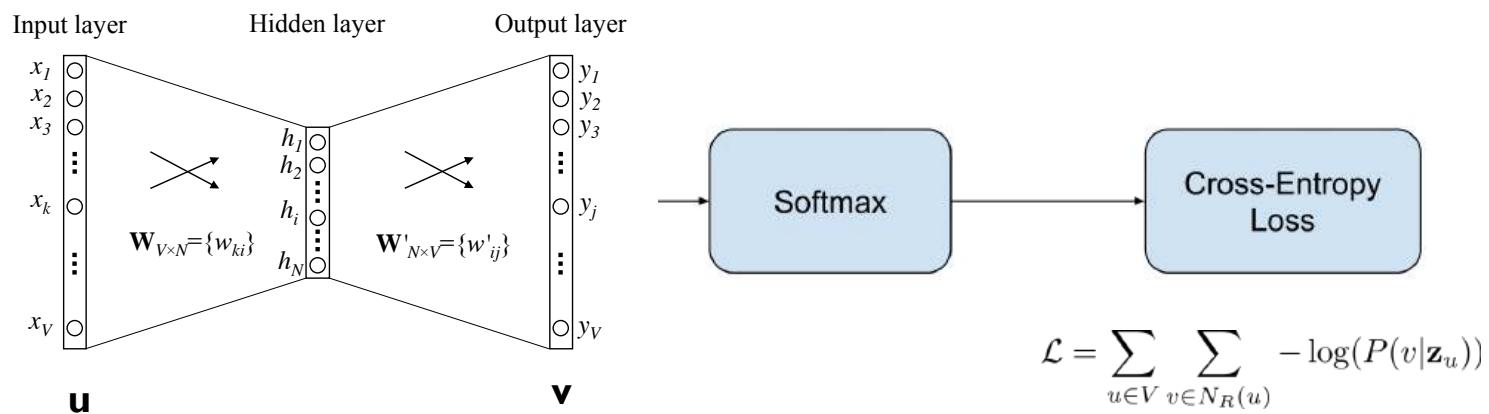
Return **parameter p** :
Return back to the previous node

In-out **parameter q** :
Moving outwards (DFS) vs.
inwards (BFS)

Let 'red node' be the starting node and the walker is currently at 'green node'

1. Go back to the red node with transition probability weighted by **$1/p$**
2. Go to the blue node nearer to the red node with transition probability(t.p.) weighted by **1**
3. Go to any other node with t.p. weighted by **$1/q$**

Step 2: Training Flashback



Random Walk Optimization

Putting things together:

$$\mathcal{L} = \sum_{u \in V} \left(\sum_{v \in N_R(u)} - \log \left(\frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)} \right) \right)$$

sum over all nodes u

sum over nodes v seen on random walks starting from u

predicted probability of u and v co-occurring on random walk

Optimizing random walk embeddings =

Finding embeddings \mathbf{z}_u that minimize L

Random Walk Optimization

But doing this naively is too expensive!!

$$\mathcal{L} = \sum_{u \in V} \sum_{v \in N_R(u)} -\log \left(\frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)} \right)$$

Nested sum over nodes
gives $O(|V|^2)$ complexity!!

Negative Sampling (not exam material)

Solution: Negative sampling

$$\log \left(\frac{\exp(\mathbf{z}_u^\top \mathbf{z}_v)}{\sum_{n \in V} \exp(\mathbf{z}_u^\top \mathbf{z}_n)} \right)$$
$$\approx \log(\sigma(\mathbf{z}_u^\top \mathbf{z}_v)) - \sum_{i=1}^k \log(\sigma(\mathbf{z}_u^\top \mathbf{z}_{n_i})), n_i \sim P_V$$

↑
sigmoid function

random distribution over all
nodes

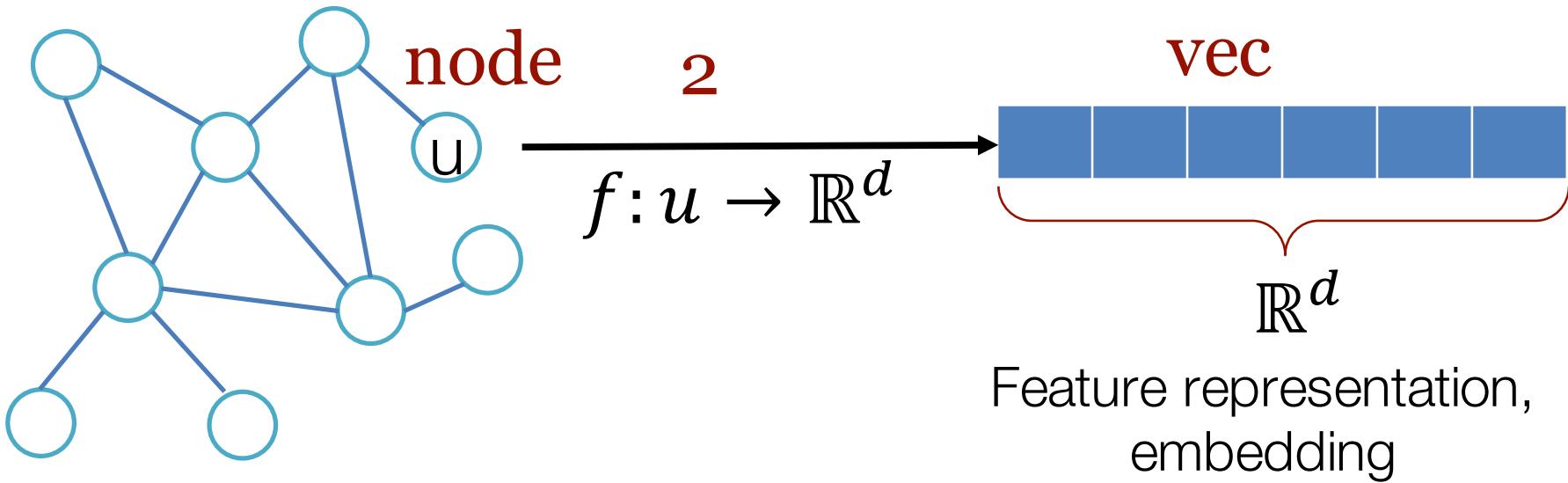
i.e., instead of normalizing w.r.t. all nodes, normalize
against k random “negative samples”



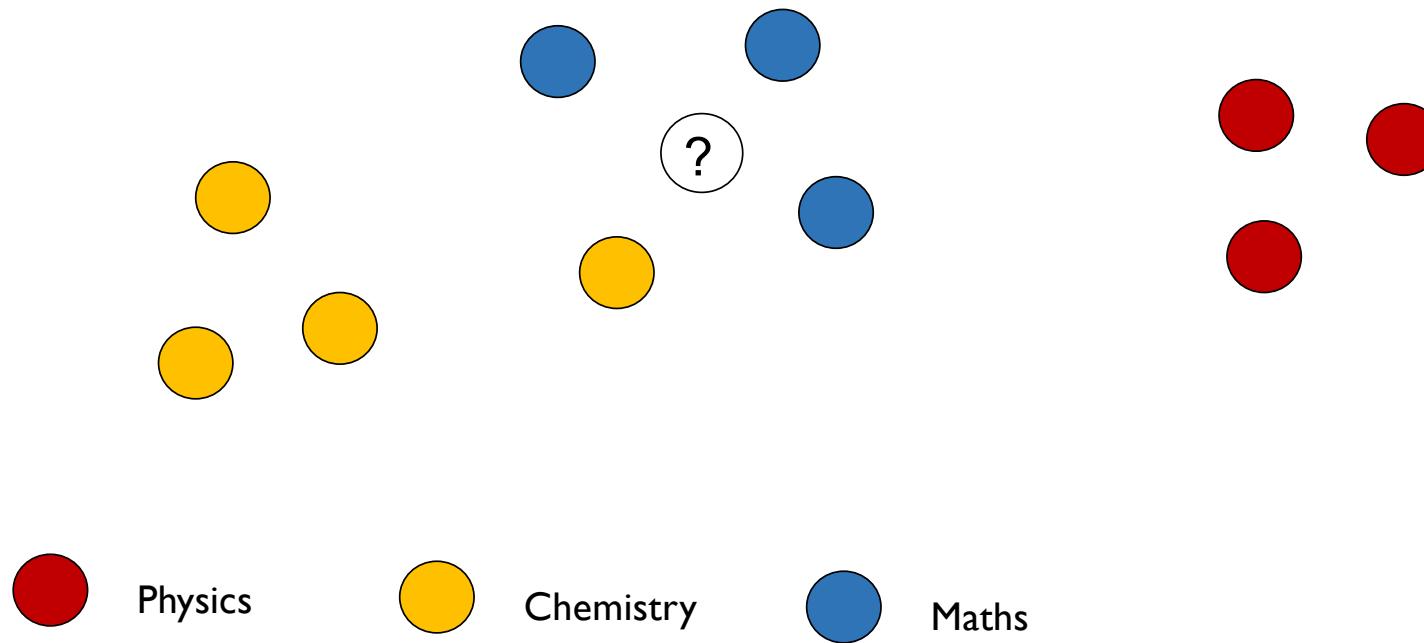
CSE2525 Applying Embeddings for Supervised Tasks

Graph Embeddings

Compute embeddings – No labels



Node classification -- Majority Voting



References

- Distributed Representations of Words and Phrases and their Compositionality Mikolov et al. Neurips 2013.
- Github: <https://github.com/RaRe-Technologies/gensim>
- **Blog:** An illustrated word2vec: <https://jalammar.github.io/illustrated-word2vec/>
- Book chapter: https://d2l.ai/chapter_natural-language-processing-pretraining/word2vec.html

References

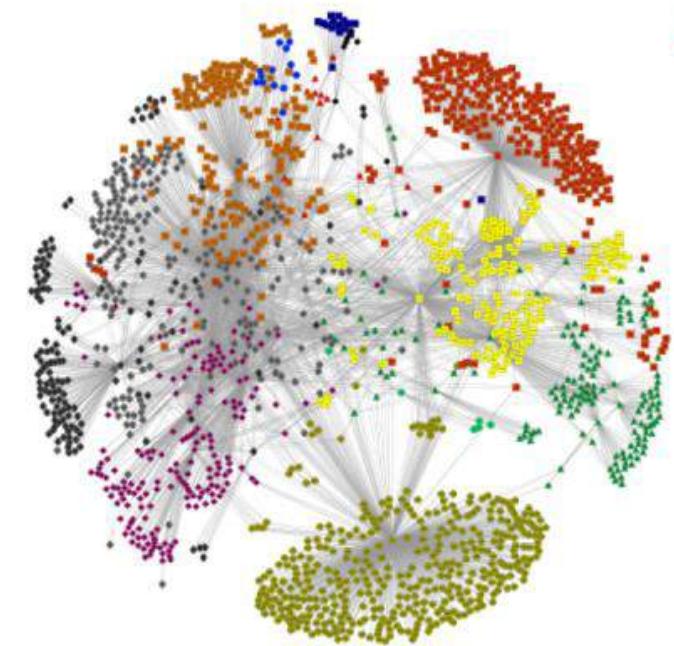
- Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "*Deepwalk: Online learning of social representations.*" Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014.
- Grover, A., & Leskovec, J. (2016, August). "*node2vec: Scalable feature learning for networks.*" In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 855-864).
- William Hamilton. "Graph Representation Learning" --
https://www.cs.mcgill.ca/~wlh/grl_book/files/GRL_Book.pdf

CSE2525 Graph Mining

Graph Properties, Centrality, Clustering

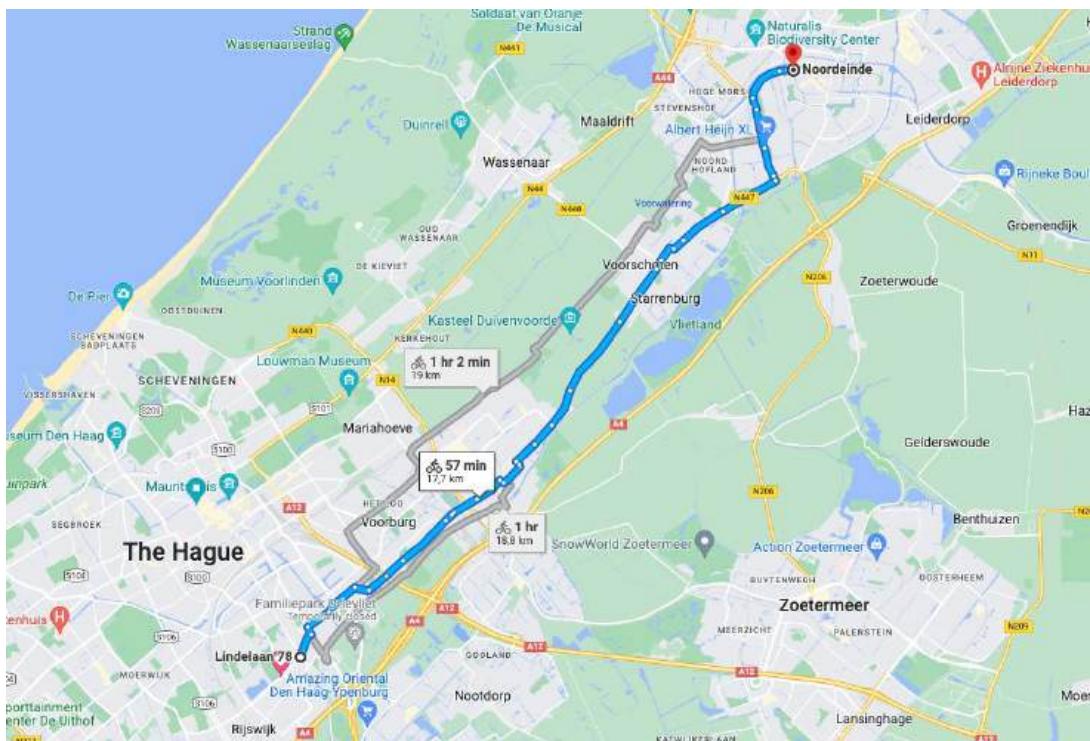
Why Graphs ?

- Many Real world datasets are graphs
 - Social Networks
 - Web graphs
 - XML parse trees
 - Protein-protein interactions
 - Road Networks
 -
- **Many of these Graphs are massive and need automatic methods to understand them**



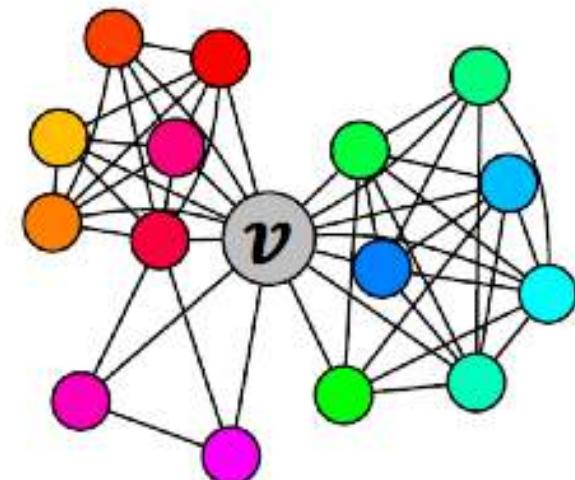
Whats possible with Graphs ?

- Finding shortest paths
 - Road networks, Social Networks
 - Internet routing



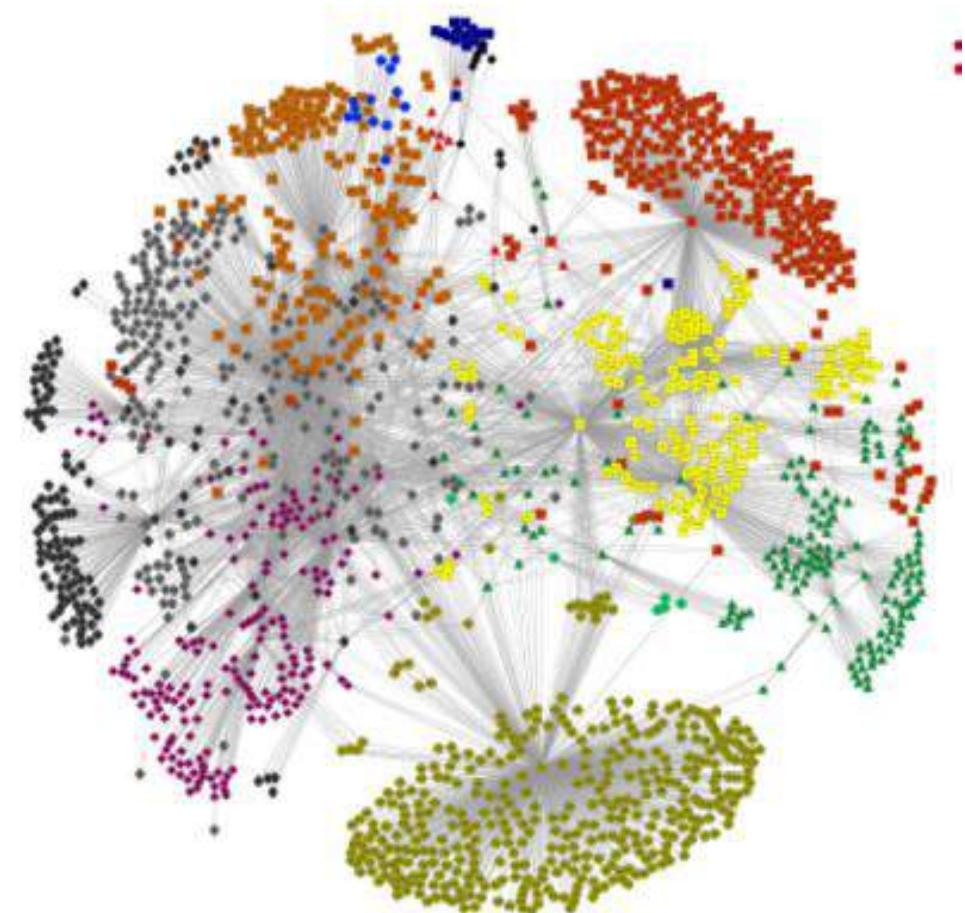
Whats possible with Graphs ?

- Finding important nodes
 - **People** – social media influencers, early adopters, ..
 - **Websites** – news, authoritative sources,..
 - Page rank as an influential algorithms
 - **Molecules**
 - Protein-Protein interaction graphs,..
 - **Products**



Whats possible with Graphs ?

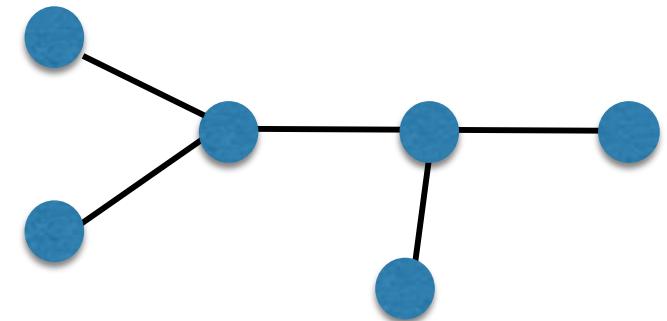
- Discovering **communities**
 - People, developers, customers
 - Papers, articles
 - Citation networks



What are Graphs ?

$$(V, E \subseteq V^2)$$

- A graph is composed of vertices and edges
 - Elements in V are **vertices** or **nodes**
 - Pairs (v,u) in E are **edges** of the graph
 - Pairs can be ordered or unordered
 - Directed and undirected graphs



$$|V| = n$$

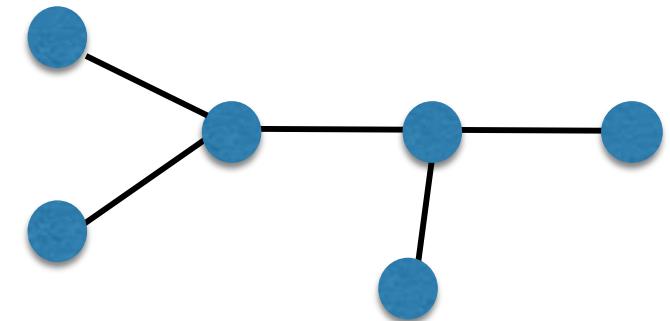
$$|E| = m$$

$$(v_i, v_j) \in E$$

What are Graphs ?

$$(V, E \subseteq V^2)$$

- Graphs can be labelled
 - Vertices can have a **labeling $L(v)$**
 - Edges can have a **labeling $L(u,v)$**
- A tree is a rooted, connected and acyclic graph
- Graphs can be represented using Adjacency matrices
 - $|V| \times |V|$ matrix **A** with $A(i,j) = 1$ if $(v_i, v_j) \in E$



$$|V| = n$$

$$|E| = m$$

How to store graphs ?

Graphs can be represented using **Adjacency matrices**

- $|V| \times |V|$ matrix **A** with $A(i,j) = 1$ if there is an edge
- Fast processing, high space complexity

```
[[0, 1, 1, 0],  
 [1, 0, 1, 0],  
 [1, 1, 0, 1],  
 [0, 0, 1, 0]]
```

Graphs can be represented using **Adjacency Lists**

- A linked list for each vertex
- Slow processing, low space complexity

```
adj_list = {  
 0: [1, 2],  
 1: [0, 2],  
 2: [0, 1, 3],  
 3: [2]  
}
```

CSE2525 Graph Mining

Graph Properties

Graph Properties

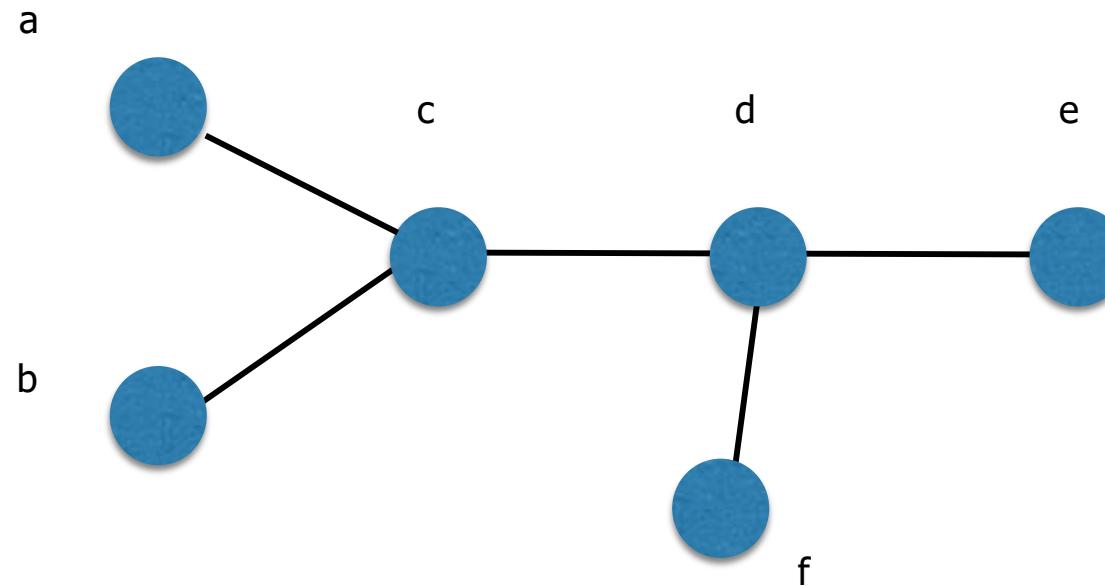
- Several real-world graphs exhibit certain characteristics and properties
 - Studying what these are and explaining why they appear is an important area of network research
 - As data miners, we need to understand the consequences of these characteristics.
 - Finding a result explained merely by one of these characteristics is not interesting

Eccentricity, Radius & Diameter

- The **distance** $d(u,v)$ between two vertices is the (weighted) length of the shortest path between them
- The **eccentricity** of a vertex v , $e(v)$, is its maximum distance to any other vertex
- The **radius** of a connected graph, $r(G)$, is the minimum eccentricity of any vertex
- The **diameter** of a connected graph, $d(G)$, is the maximum eccentricity of any vertex
- The **effective diameter** of a graph is smallest number that is larger than the eccentricity of a large fraction of the vertices in the graph

Eccentricity, Radius & Diameter

- What is the sum of outdegrees of all nodes in a graph $G(n,m)$?



- Find the eccentricity (of nodes a,c), radius and diameter of graph

CSE2525 Graph Mining

Computing Betweenness

Small World Property

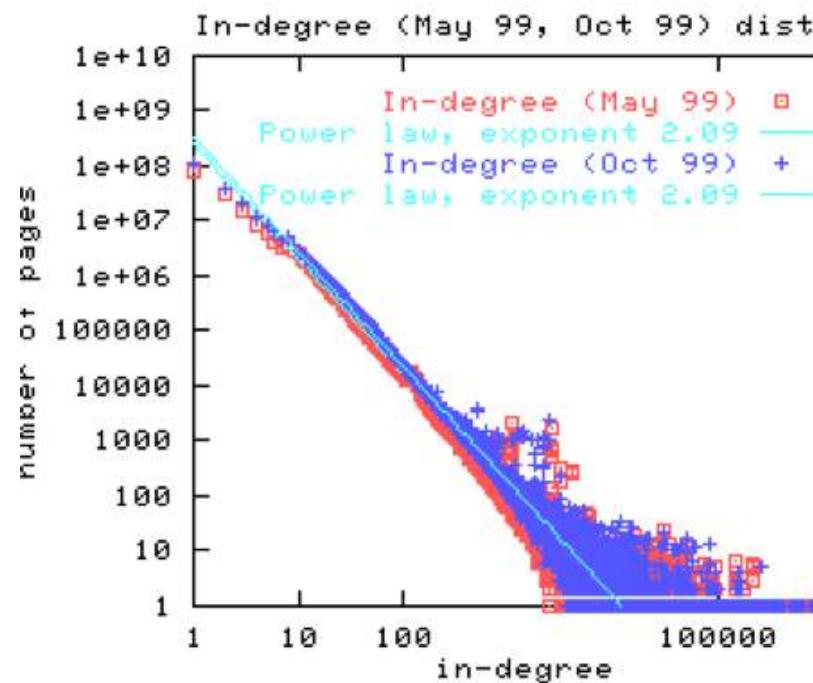
- A graph G is said to exhibit a **small-world property** if its average path length scales logarithmically
 - The six degrees of Kevin Bacon is based on this property
 - How far a mathematician is from Hungarian combinatorist Paul Erdős ?
 - A radius of a large, connected mathematical co-authorship network (268K authors) is 12 and diameter 23

Scale Free Property

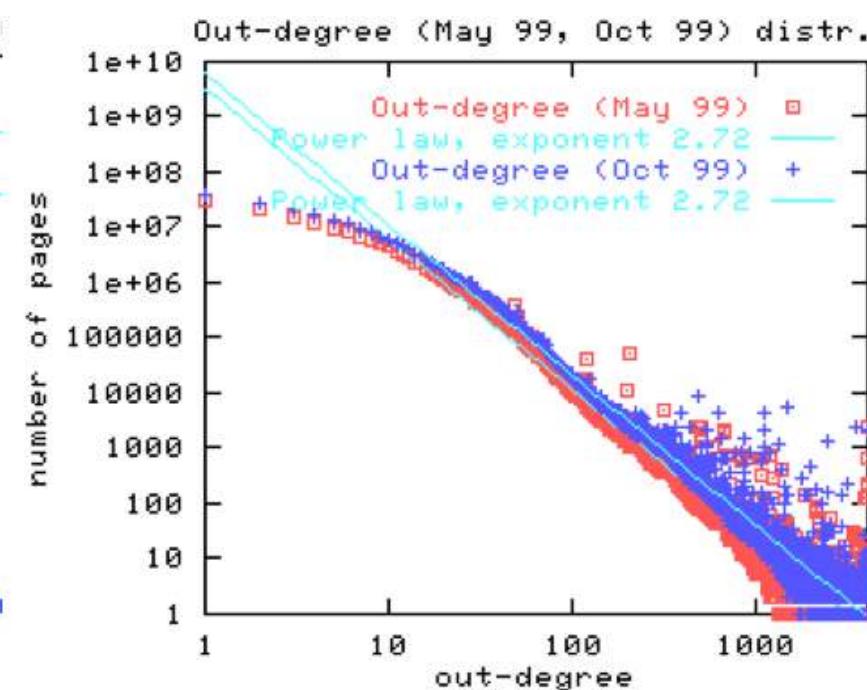
- The **degree distribution** of a graph is the distribution of its vertex degrees
 - How many vertices with degree 1, how many with degree 2, etc.
 - $f(k)$ is the number of vertices with degree k
- A graph is said to exhibit **scale-free property** if $f(k) \propto k^{-\gamma}$
 - So-called power-law distribution
 - Majority of vertices have small degrees, few have very high degrees
- Scale-free: $f(ck) = a(ck)^{-\gamma} = (ac^{-\gamma})k^{-\gamma} \propto k^{-\gamma}$

Example- WWW

In-degree



Out-degree



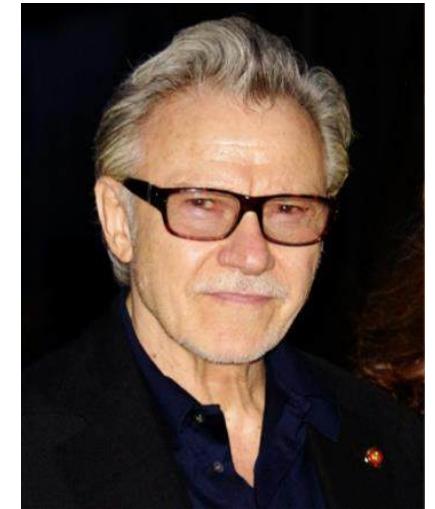
Broder et al. *Graph structure in the web*. WWW'00

$$s = 2.09$$

$$s = 2.72$$

Centrality

- The more central a node the faster it can reach other nodes
 - “Every actor is related to Kevin Bacon by no more than 6 hops”
- Measures for centrality
 - Degree and Eccentricity centrality
 - Closeness centrality
 - Betweenness centrality



Degree and Eccentricity centrality

Centrality is a function $c: V \rightarrow \mathbb{R}$ that induces a total order in V

- The higher the centrality of a vertex, the more important it is
- In **degree centrality** $c(v_i) = d(v_i)$, the degree of the vertex
- In **eccentricity centrality**, the least eccentric vertex is the most central one, $c(v_i) = 1/e(v_i)$
 - The least eccentric vertex is *central*
 - The most eccentric vertex is *peripheral*

Closeness centrality

- In **closeness centrality** the vertex with least distance to *all other* vertices is the center

closeness centrality $c(x)$ of a vertex x

$$c(x) = \frac{1}{\sum_{y \neq x \in V} d(y, x)}.$$

- In eccentricity centrality we aim to minimize the maximum distance
- In closeness centrality we aim to minimize the average distance
 - This is the distance used to measure the centre of Hollywood

Betweenness centrality

- The **node betweenness centrality or betweenness** measures the number of shortest paths that travel through a node/vertex x
- Can also be defined for edges — **edge betweenness**
- Edges with high edge betweenness are called **weak ties**

betweenness centrality $b(x)$ of a vertex x :

$$b(x) = \sum_{\substack{s \neq x \neq t \in V \\ s \neq t}} \frac{\sigma_{st}(x)}{\sigma_{st}}$$

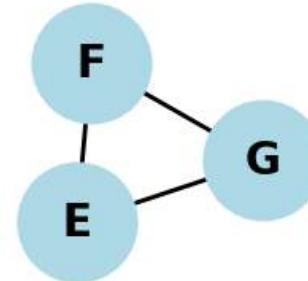
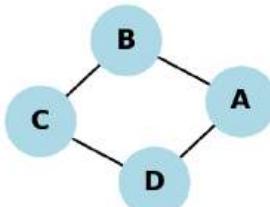
σ_{st} : number of SPs from s to t

$\sigma_{st}(x)$: how many of them pass through x

CSE2525 Graph Mining

Spectral Clustering

Connected Components



- Why do we need connected components ?
 - Most real-world graphs are sparse
 - Parallelize computation
 - Optimize storage

How to find connected components ?

```
adj_list = {  
    'A': ['B', 'D'],  
    'B': ['A', 'C'],  
    'C': ['B', 'D'],  
    'D': ['A', 'C'],  
    'E': ['F', 'G'],  
    'F': ['E', 'G'],  
    'G': ['E', 'F'],  
    'H': []  
}
```

	A	B	C	D	E	F	G	H
A	0	1	0	1	0	0	0	0
B	1	0	1	0	0	0	0	0
C	0	1	0	1	0	0	0	0
D	1	0	1	0	0	0	0	0
E	0	0	0	0	0	1	1	0
F	0	0	0	0	1	0	1	0
G	0	0	0	0	1	1	0	0
H	0	0	0	0	0	0	0	0

How to find connected components ?

```
adj_list = {  
    'A': ['B', 'D'],  
    'B': ['A', 'C'],  
    'C': ['B', 'D'],  
    'D': ['A', 'C'],  
    'E': ['F', 'G'],  
    'F': ['E', 'G'],  
    'G': ['E', 'F'],  
    'H': []  
}
```

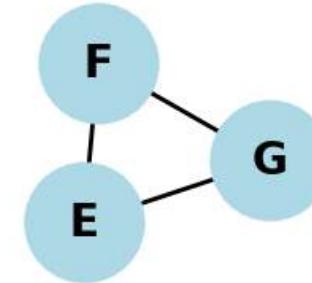
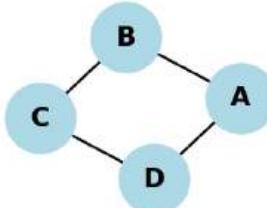
	A	B	C	D	E	F	G	H
A	0	1	0	1	0	0	0	0
B	1	0	1	0	0	0	0	0
C	0	1	0	1	0	0	0	0
D	1	0	1	0	0	0	0	0
E	0	0	0	0	0	1	1	0
F	0	0	0	0	1	0	1	0
G	0	0	0	0	1	1	0	0
H	0	0	0	0	0	0	0	0

Component 1: [(A, B), (B, C), (C, D), (D, A)]

Component 2: [(E, F), (F, G), (G, E)]

Component 3: []

Algorithm for connected components



FindConnectedComponents(Graph G):

 Initialize a set *visited* $\leftarrow \emptyset$ # Tracks visited nodes

 Initialize a list *components* $\leftarrow []$ # Stores connected components

FOR each node u IN G: # Iterate through all nodes

IF u NOT IN *visited*:

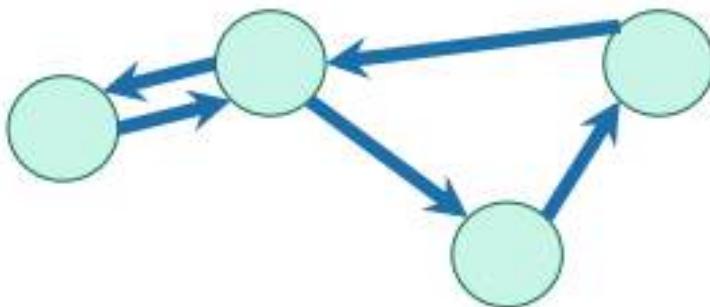
 Initialize a list *component* $\leftarrow []$ # New connected component

component \leftarrow CALL **DFS**(G, u, *visited*, *component*) # Perform DFS

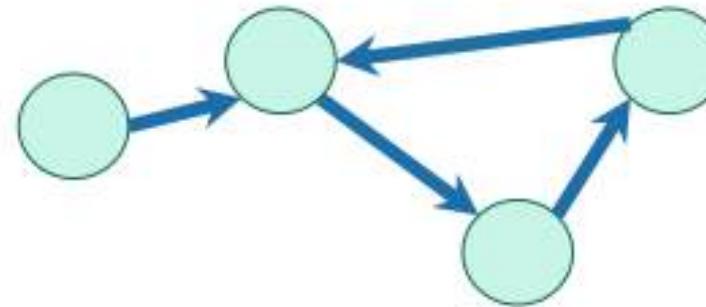
 ADD *component* TO *components* # Save the connected component

RETURN *components*

Strongly Connected Components



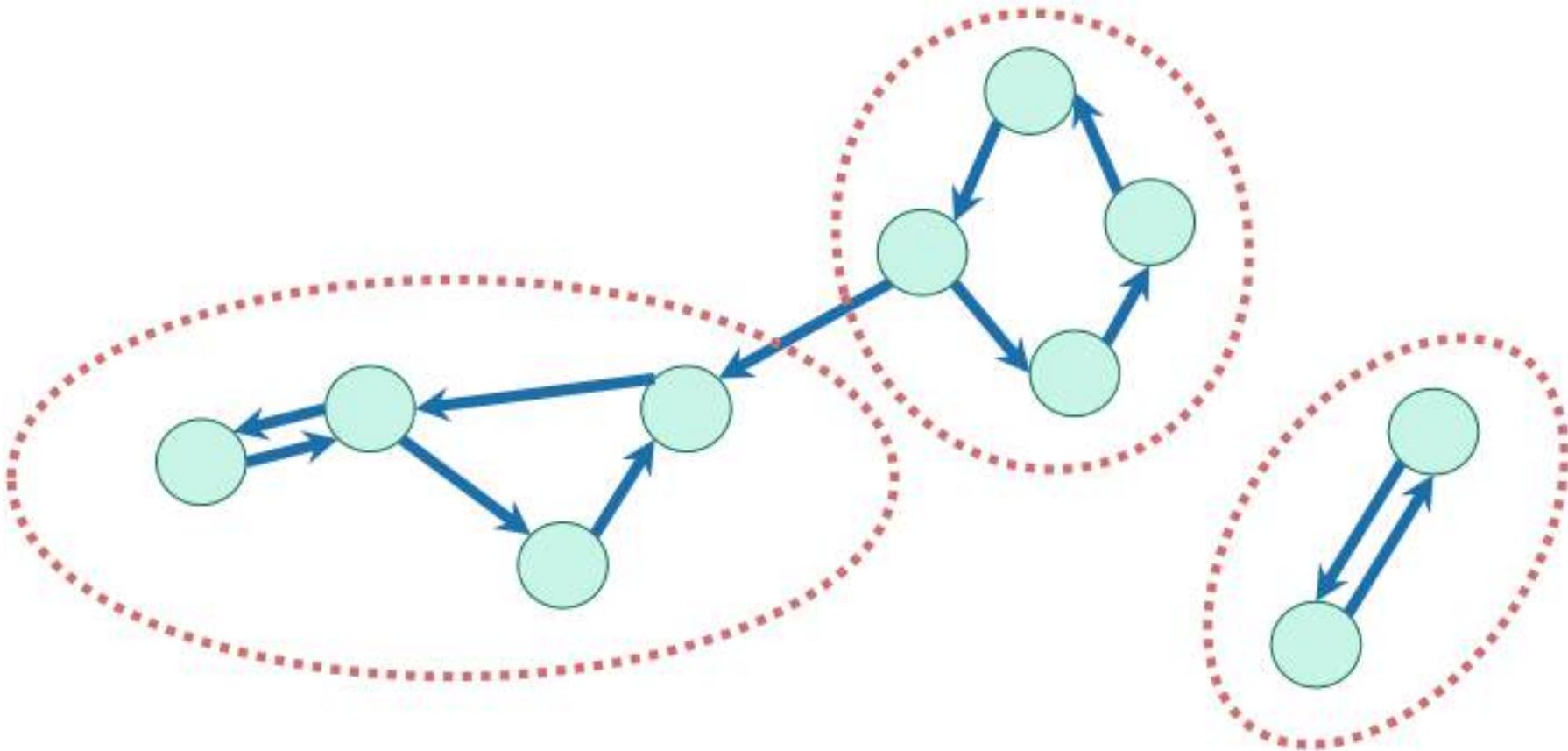
strongly connected



not strongly connected

- A directed graph $G = (V, E)$ is **strongly connected** if:
- for all v, w in V :
 - there is a path from v to w and
 - There is a path from w to v .

A connected component can be decomposed into multiple SCCs

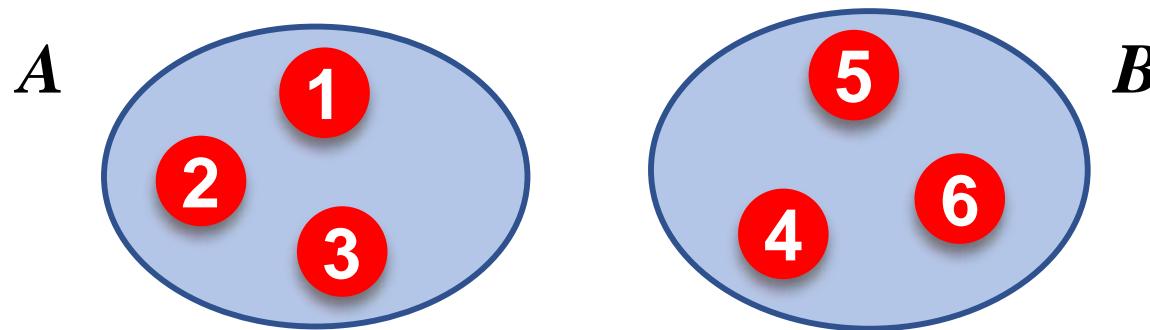
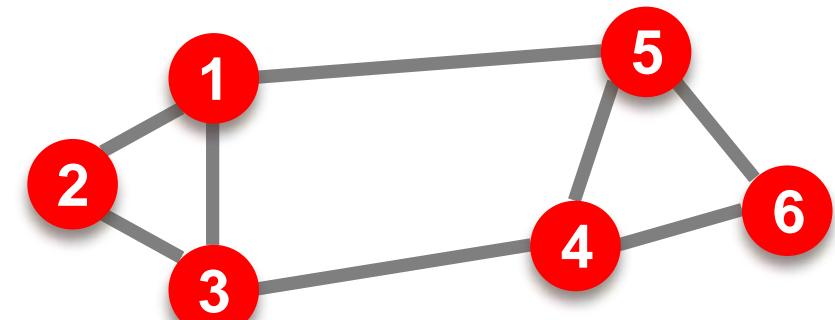


CSE2525 Graph Mining

Connected Components

Graph Partitioning

- **Undirected graph $G(V, E)$:**
- **Bi-partitioning task:**
 - Divide vertices into two disjoint groups A, B

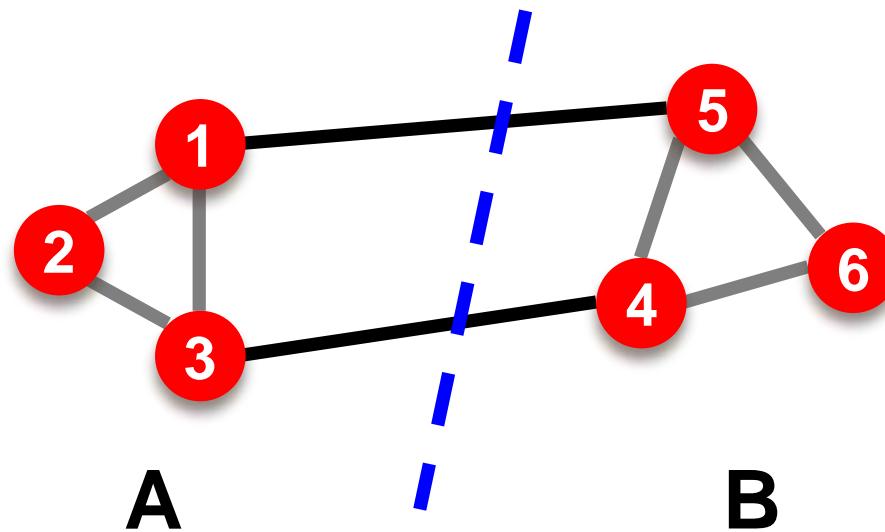


- **Questions:**
 - How can we define a “good” partition of G ?
 - How can we efficiently identify such a partition?

Graph Partitioning

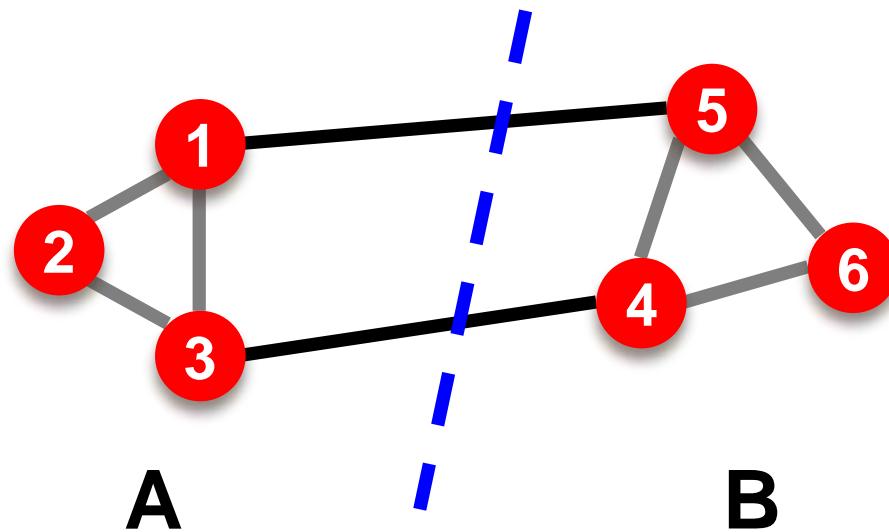
- **What makes a good partition?**

- Maximize the number of within-group connections
- Minimize the number of between-group connections



Spectral Clustering

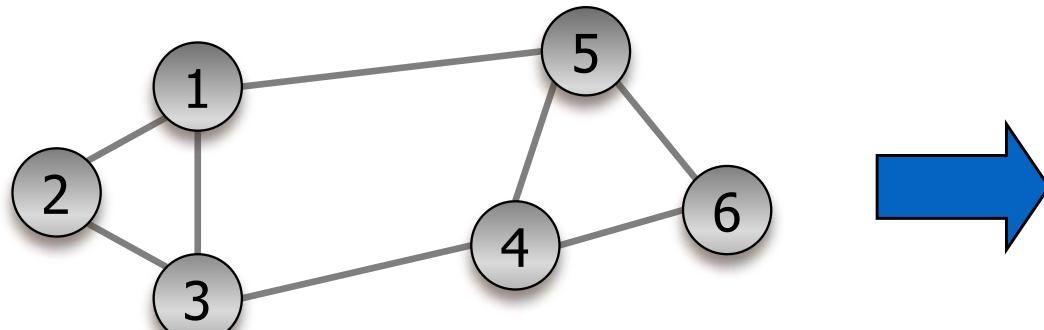
- Will use properties of the Graph Matrix to find communities



Matrix Representations

- **Adjacency matrix (A):**

- $n \times n$ matrix $A = [a_{ij}]$, $a_{ij} = 1$ if edge between node i and j

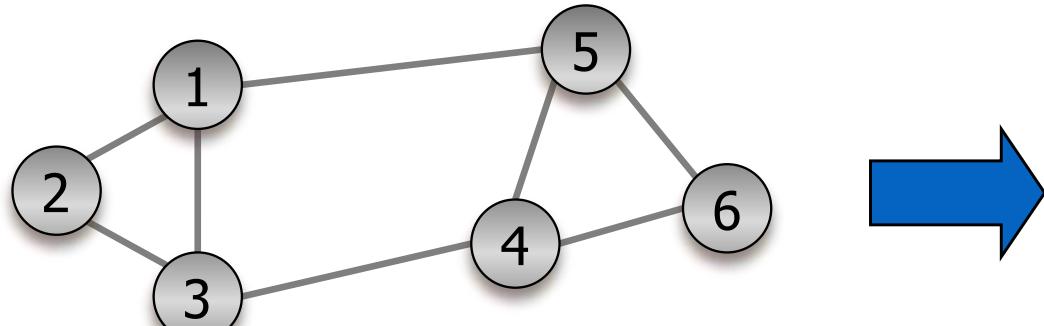


	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	0	0	1	0	1
6	0	0	0	1	1	0

Matrix Representations

- **Degree matrix (D):**

- $n \times n$ diagonal matrix
- $D = [d_{ii}]$, d_{ii} = degree of node i

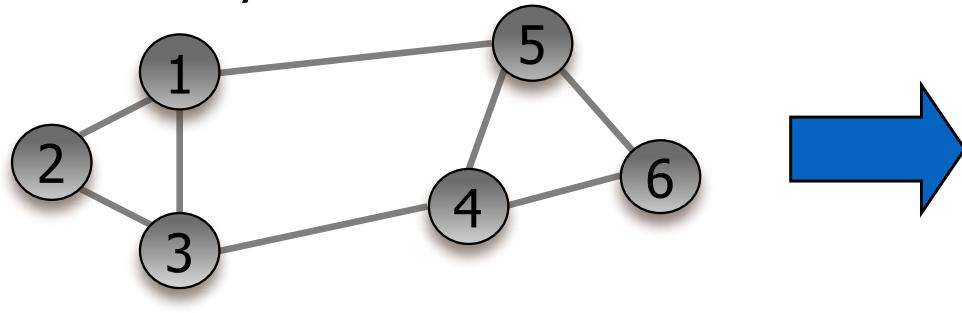


	1	2	3	4	5	6
1	3	0	0	0	0	0
2	0	2	0	0	0	0
3	0	0	3	0	0	0
4	0	0	0	3	0	0
5	0	0	0	0	3	0
6	0	0	0	0	0	2

Matrix Representations

- **Laplacian matrix (L):**

- $n \times n$ symmetric matrix



	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

$$L = D - A$$

Graph Properties

x is a vector in \Re^n with components (x_1, \dots, x_n)

Think of it as a label/value of each node of G

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

A: adjacency matrix of undirected G

$A_{ij} = 1$ if (i, j) is an edge, else 0

$$y_i = \sum_{j=1}^n A_{ij} x_j = \sum_{(i,j) \in E} x_j$$

What is the meaning of y ?

$$y_i = \sum_{j=1}^n A_{ij} x_j = \sum_{(i,j) \in E} x_j$$

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

Entry y_i is a sum of labels x_j of neighbors of i

What is the meaning of Ax ?

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$\color{blue}{A \cdot x = \lambda \cdot x}$

What are the scalars and vector here ?

Spectral Graph Theory

- **Spectral Graph Theory:**

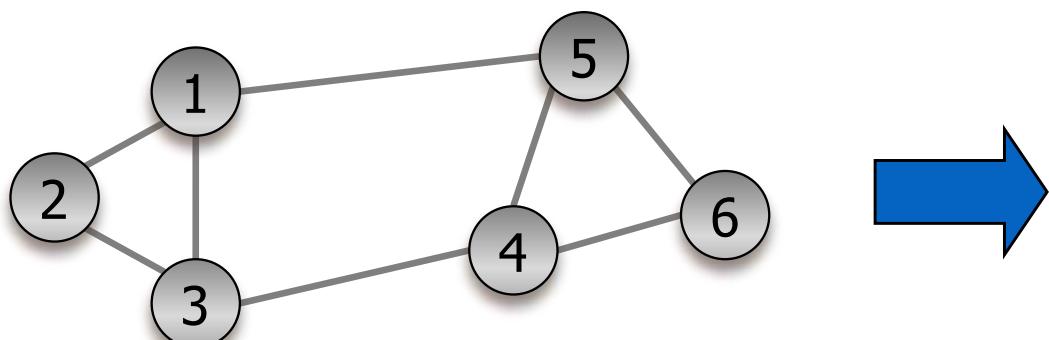
- Analyze the “spectrum” of matrix representing G
- **Spectrum:** Eigenvectors x_i of a graph, ordered by the magnitude (strength) of their corresponding eigenvalues λ_i :

$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad \Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_n\}$$
$$\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$$

Matrix Representations

- **Adjacency matrix (A):**

- $n \times n$ matrix $A = [a_{ij}]$, $a_{ij} = 1$ if edge between node i and j

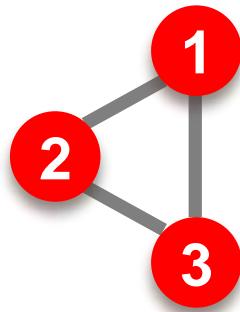


- **Important properties:**

- Symmetric matrix
- Eigenvectors are real and orthogonal

	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	0	0	1	0	1
6	0	0	0	1	1	0

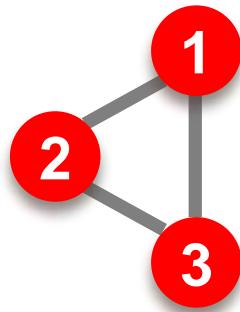
Matrix Representations



$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

What is a valid non-zero (eigenvector, eigenvalue) of the Adjacency matrix of the following fully connected graph ?

Matrix Representations



$$\begin{bmatrix} a_{11} & \dots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \lambda \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

What is a valid non-zero (eigenvector, eigenvalue) of the Adjacency matrix of the following fully connected graph ?

$$x = (1, \dots, 1)$$
 then $L \cdot x = 2x$

$$y_j = \sum_{i=1}^n A_{ij} x_i = \sum_{(j,i) \in E} x_i$$

Example: d-regular graph

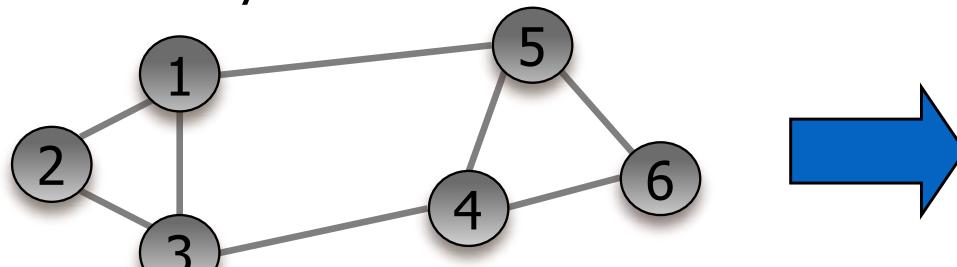
- Suppose all nodes in G have degree d and G is connected
- **What are some eigenvalues/vectors of G ? $A \cdot x = \lambda \cdot x$**
What is λ ? What x ?
 - $x = (1, 1, \dots, 1)$
 - $A \cdot x = (d, d, \dots, d) = \lambda \cdot x$. **So:** $\lambda = d$
 - $x = (1, 1, \dots, 1)$, $\lambda = d$

But how do you use it for clustering ?

The Laplacian Matrix

- **Laplacian matrix (L):**

- $n \times n$ symmetric matrix



	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

$$L = D - A$$

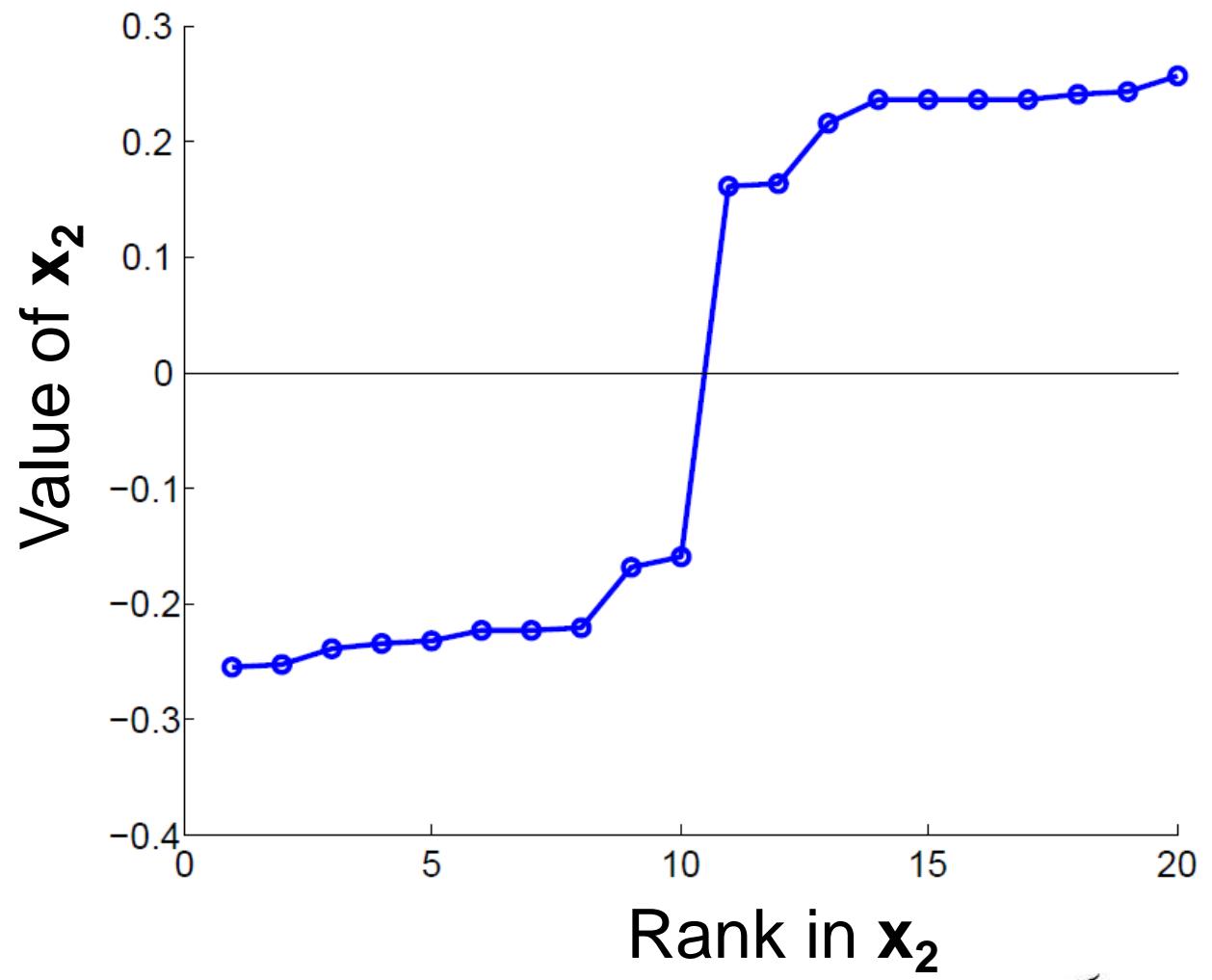
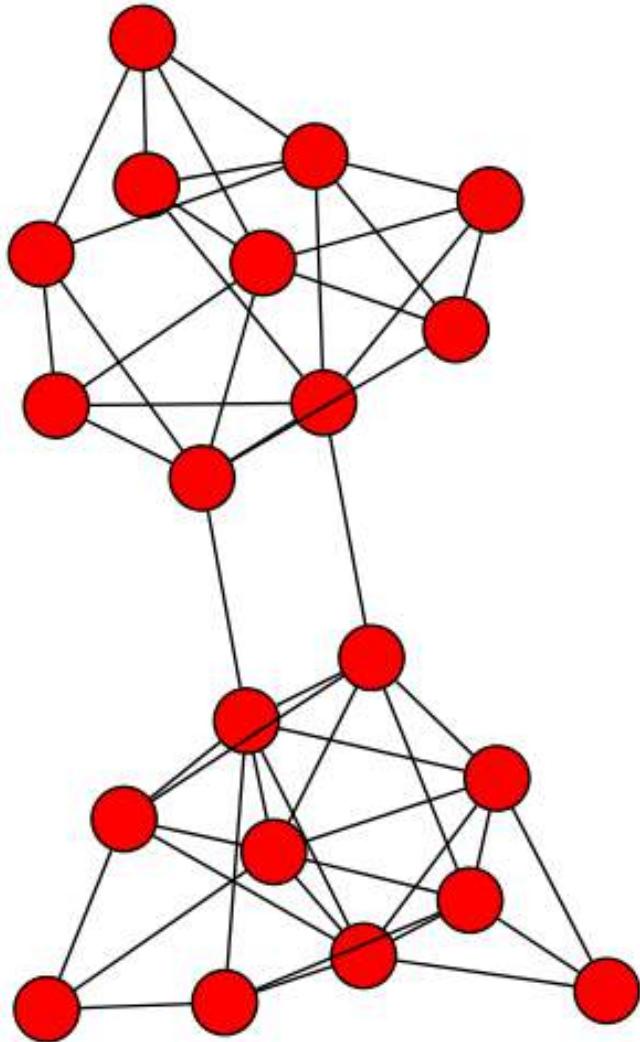
- **What is trivial eigenpair?**

- $x = (1, \dots, 1)$ then $L \cdot x = 0$ and so $\lambda = \lambda_1 = 0$

- **Important properties:**

- **Eigenvalues** are non-negative real numbers
- **Eigenvectors** are real and orthogonal

The second eigenvector



Spectral Clustering Algorithms

- **Three basic stages:**

- 1) Pre-processing**

- Construct a matrix representation of the graph

- 2) Decomposition**

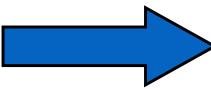
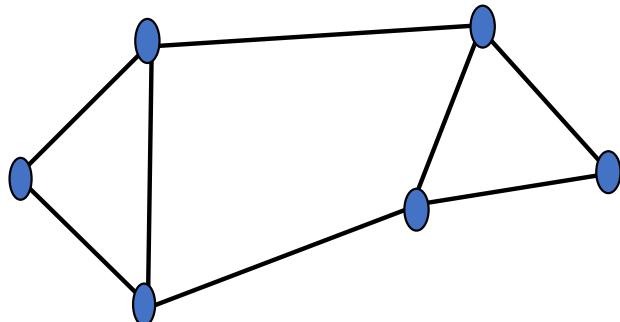
- Compute eigenvalues and eigenvectors of the matrix
- Map each point to a lower-dimensional representation based on one or more eigenvectors

- 3) Grouping**

- Assign points to two or more clusters, based on the new representation

Step 1 - Pre-Processing

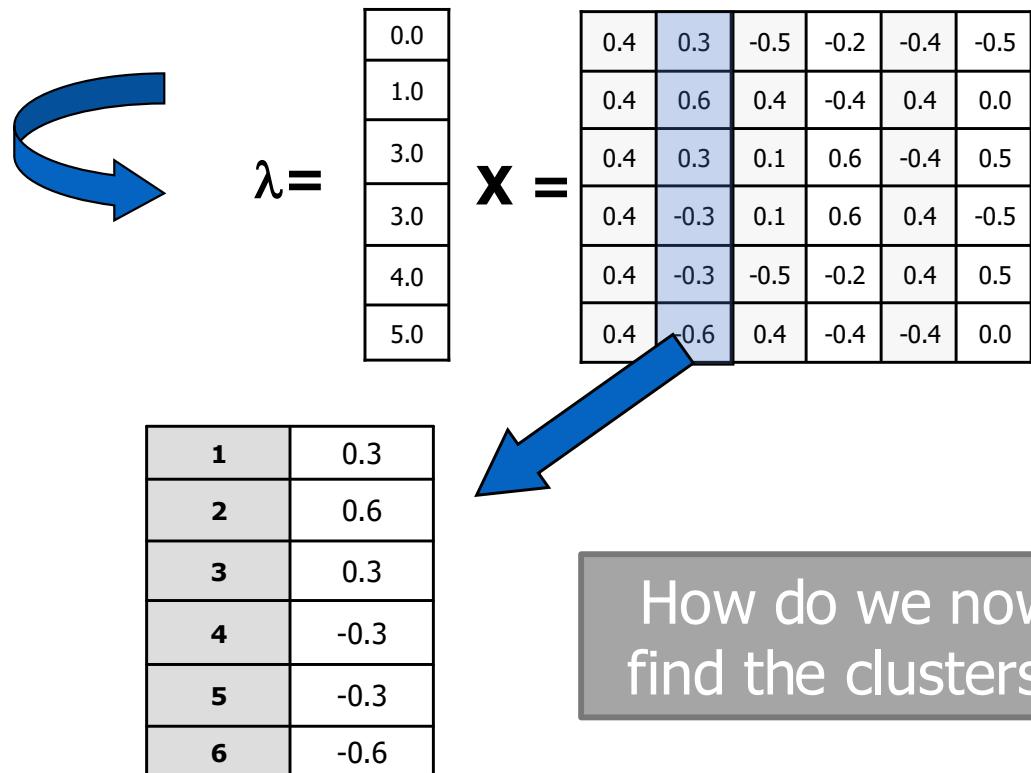
- Build Laplacian matrix L of the graph



	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

Step 2 - Decomposition

- **Find** eigenvalues λ and eigenvectors x of the matrix L
- **Map** vertices to corresponding components of λ_2



Step 3 - Grouping

- Sort components of reduced 1-dimensional vector
- Identify clusters by splitting the sorted vector in two

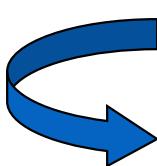
• How to choose a splitting point?

- Naïve approaches:
 - Split at **0** or median value
- More expensive approaches:
 - Attempt to minimize normalized cut in 1-dimension (sweep over ordering of nodes induced by the eigenvector)

Split at 0:

Cluster A: Positive points

Cluster B: Negative points

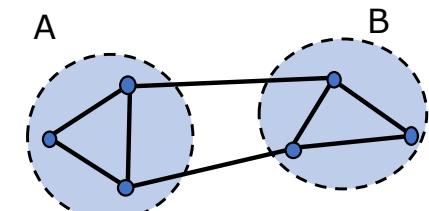


1	0.3
2	0.6
3	0.3
4	-0.3
5	-0.3
6	-0.6

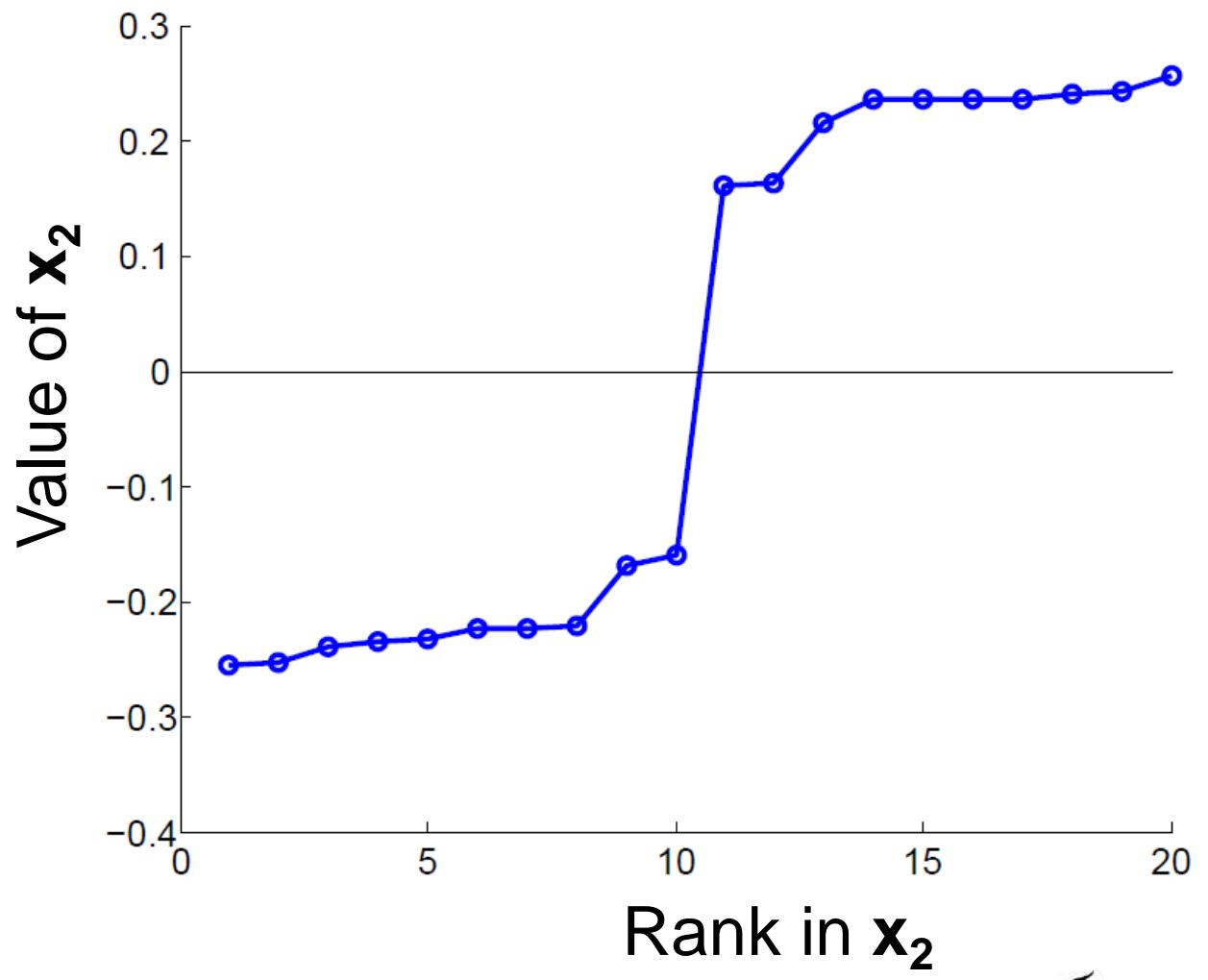
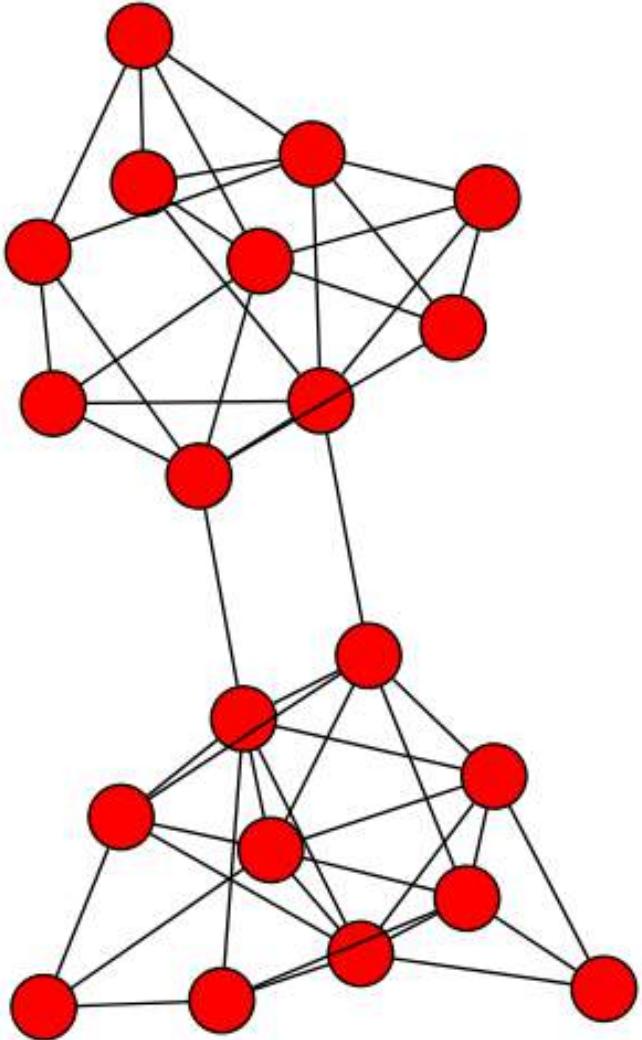


1	0.3
2	0.6
3	0.3

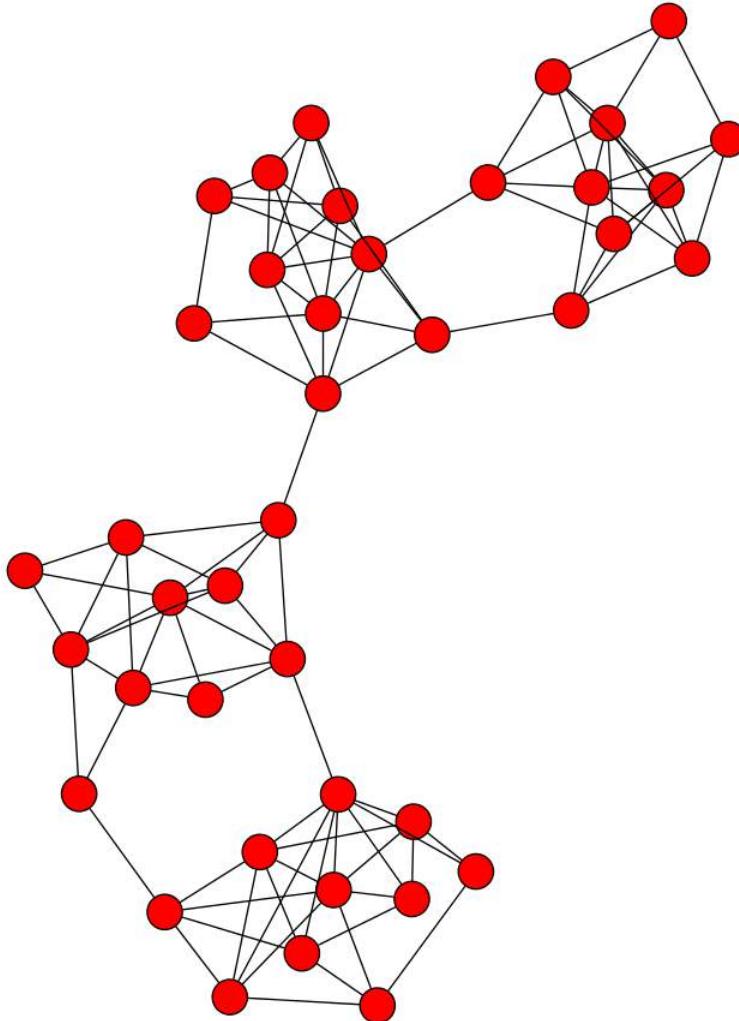
4	-0.3
5	-0.3
6	-0.6



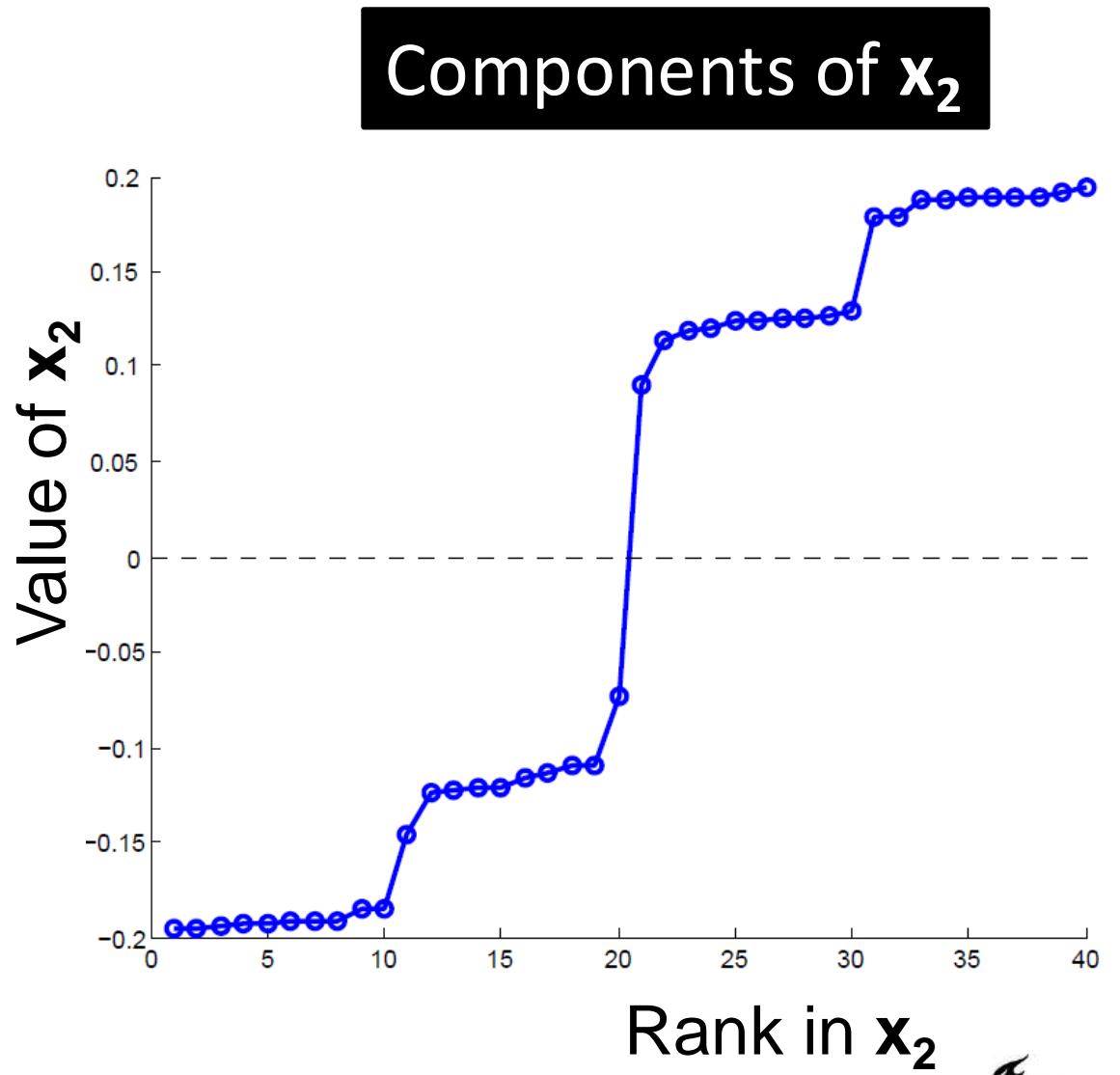
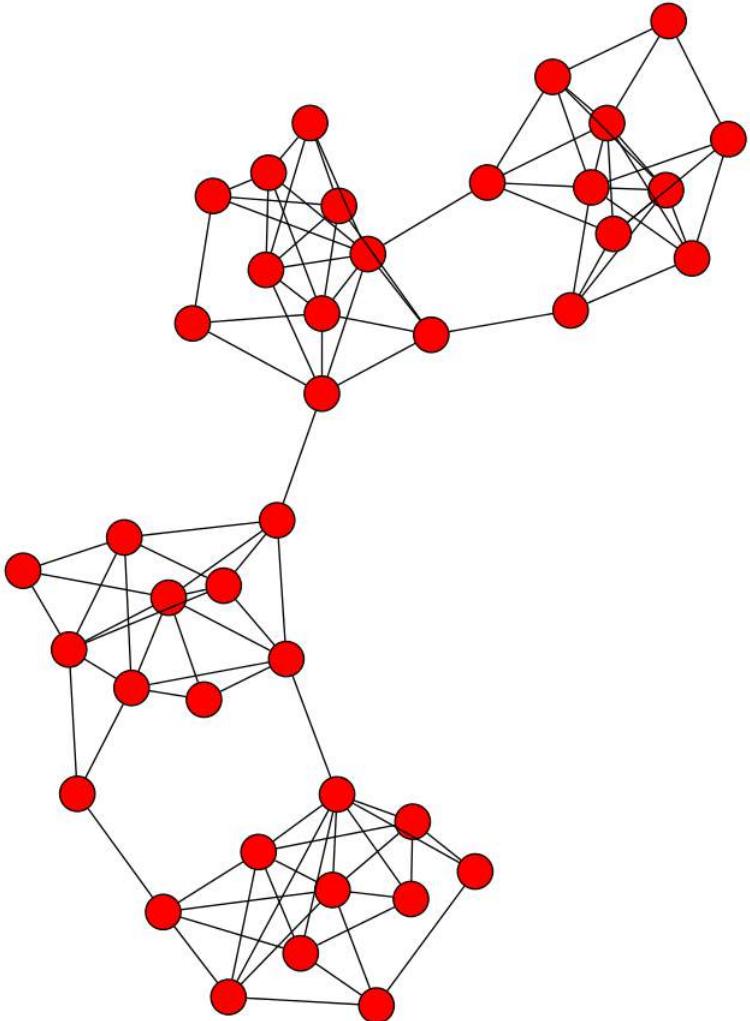
Example: Spectral Partitioning



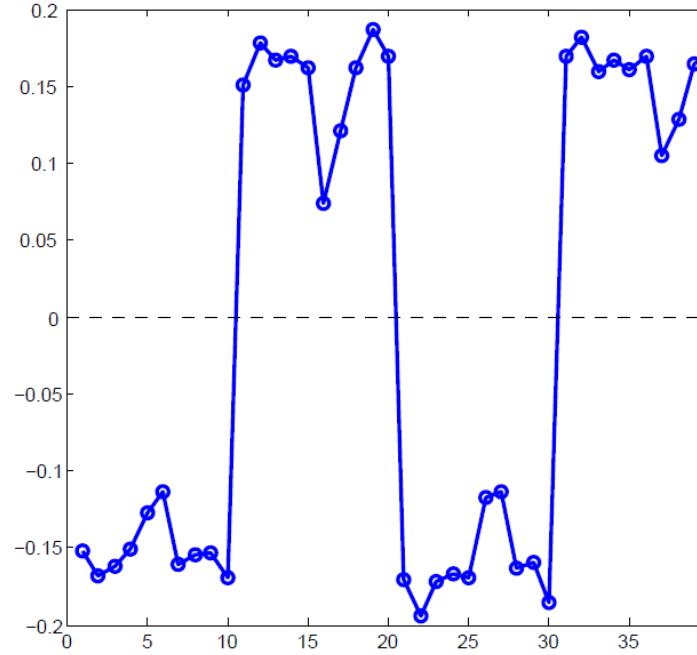
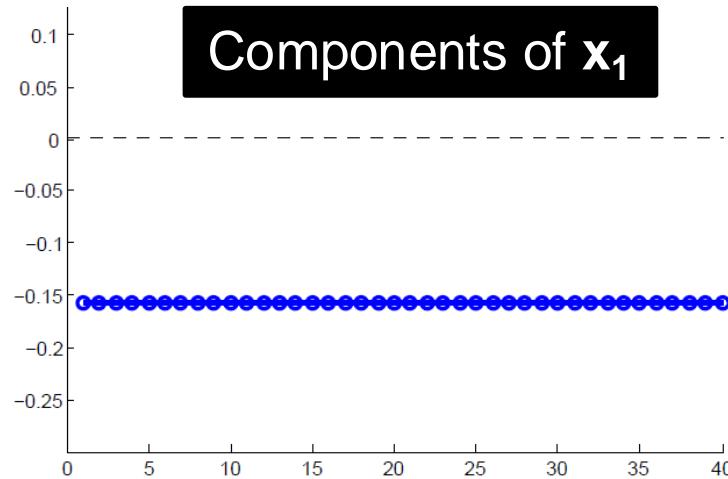
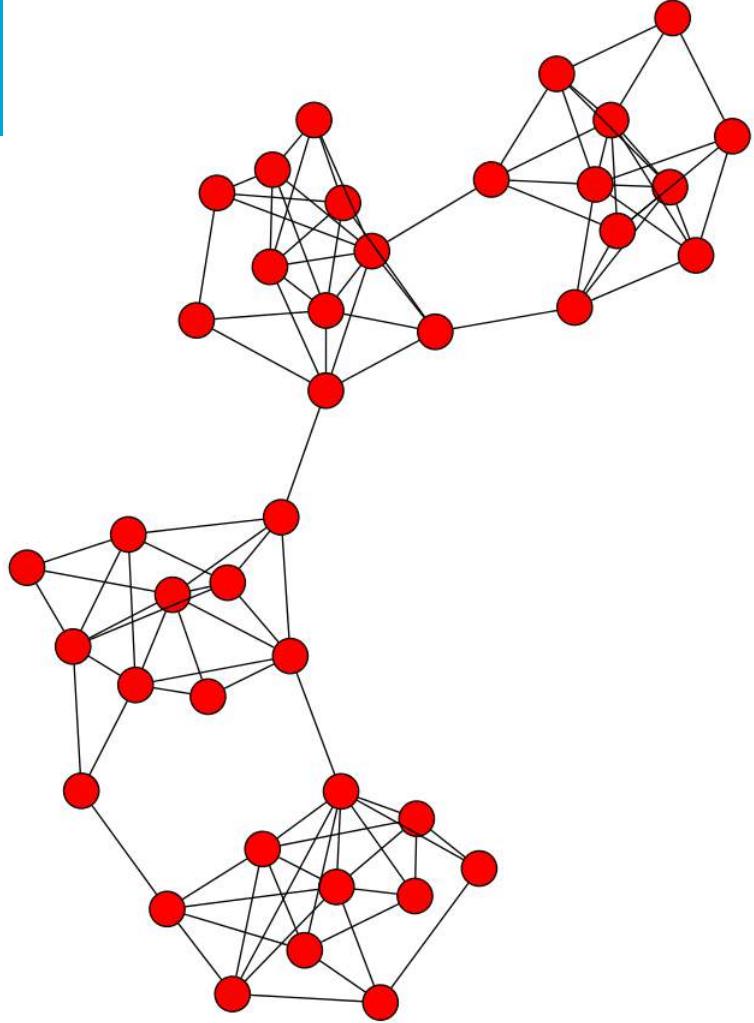
What about multiple clusters ?



Example: Spectral Partitioning



Example: Spectral partitioning



Components of \mathbf{x}_3

k-Way Spectral Clustering

How do we partition a graph into k clusters?

- **Two basic approaches:**
 - **Recursive bi-partitioning** [Hagen et al., '92]
 - Recursively apply bi-partitioning algorithm in a hierarchical divisive manner
 - Disadvantages: Inefficient, unstable
 - **Cluster multiple eigenvectors** [Shi-Malik, '00]
 - Build a reduced space from multiple eigenvectors
 - Commonly used in recent papers
 - A preferable approach...

Eigenvectors as embeddings

$\mathbf{X} =$

0.4	0.3	-0.5	-0.2	-0.4	-0.5
0.4	0.6	0.4	-0.4	0.4	0.0
0.4	0.3	0.1	0.6	-0.4	0.5
0.4	-0.3	0.1	0.6	0.4	-0.5
0.4	-0.3	-0.5	-0.2	0.4	0.5
0.4	-0.6	0.4	-0.4	-0.4	0.0

Summary

- Graphs are everywhere, and large
- Common properties used to study and model real-world graphs
- Spectral clustering over graphs

CSE2525 Graph Mining

Lab Assignment

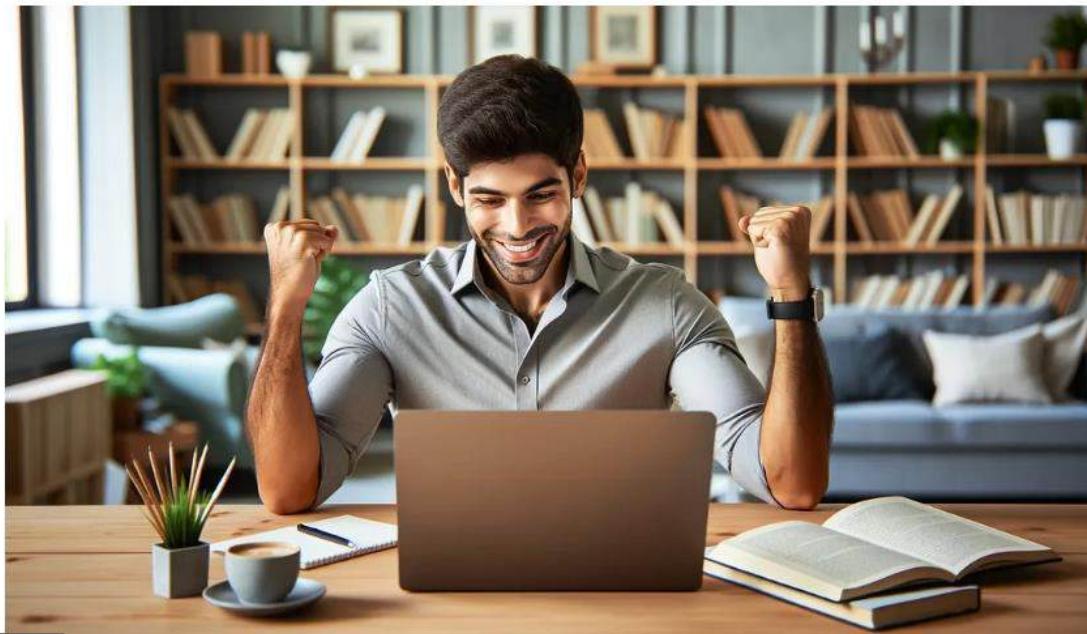
Member-only story

Embarking on the Writer's Journey: Your Gateway to Article Writing Success



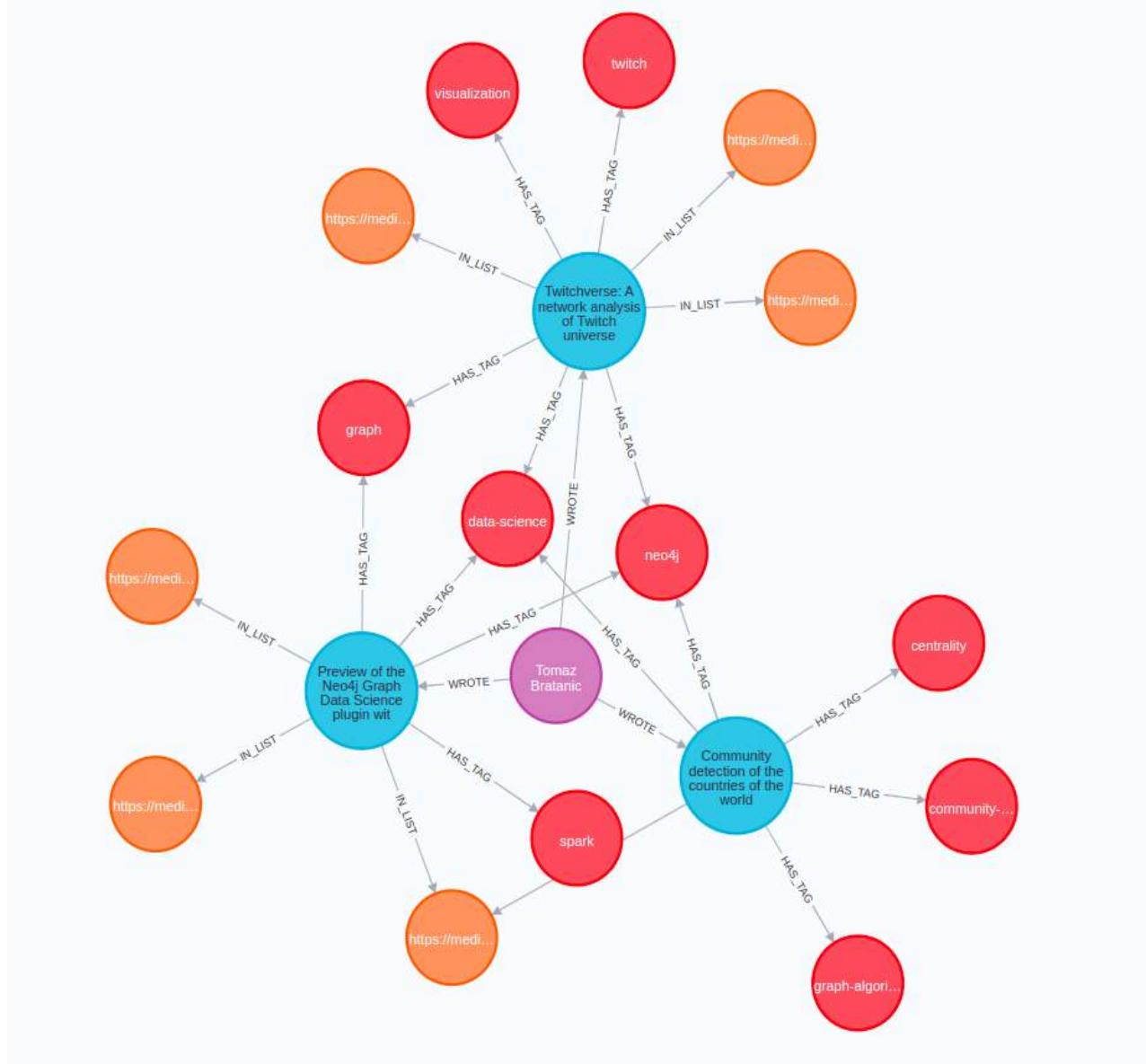
The Kayro · [Follow](#)
3 min read · 23 hours ago

19

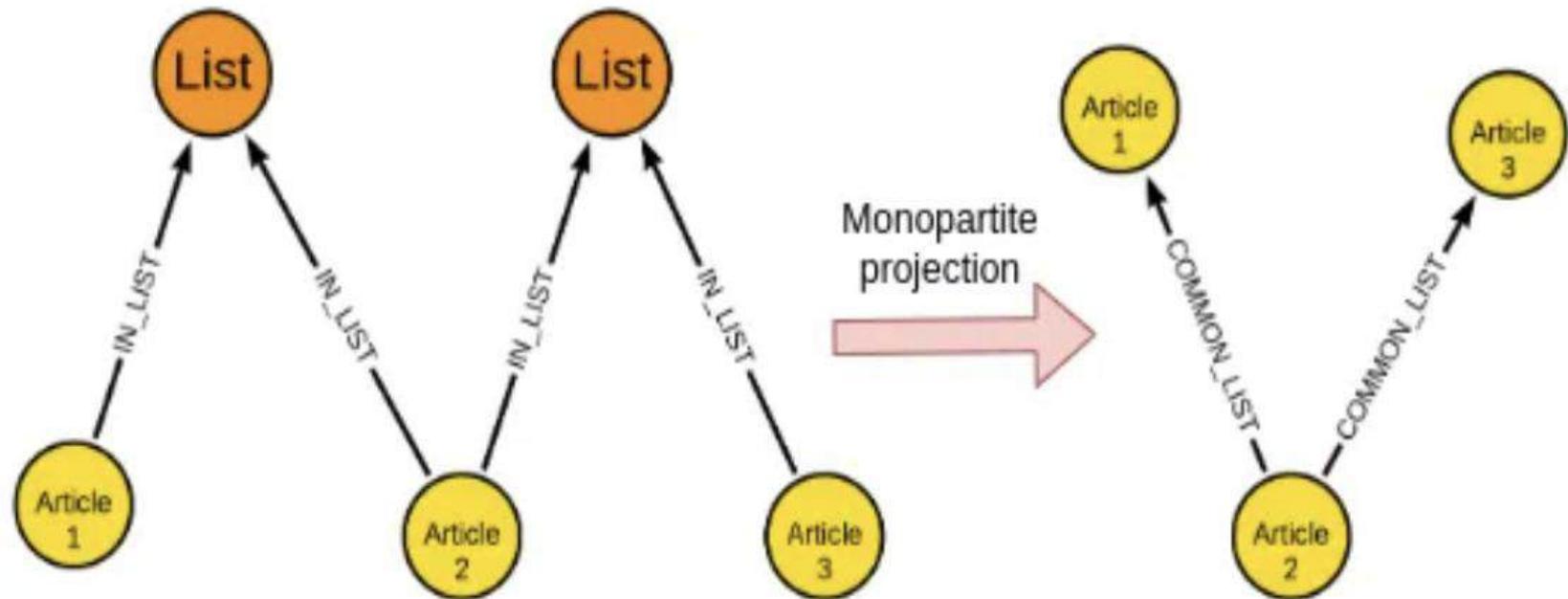


[Display a menu](#)

Real World Graphs: Medium Articles



Assignment

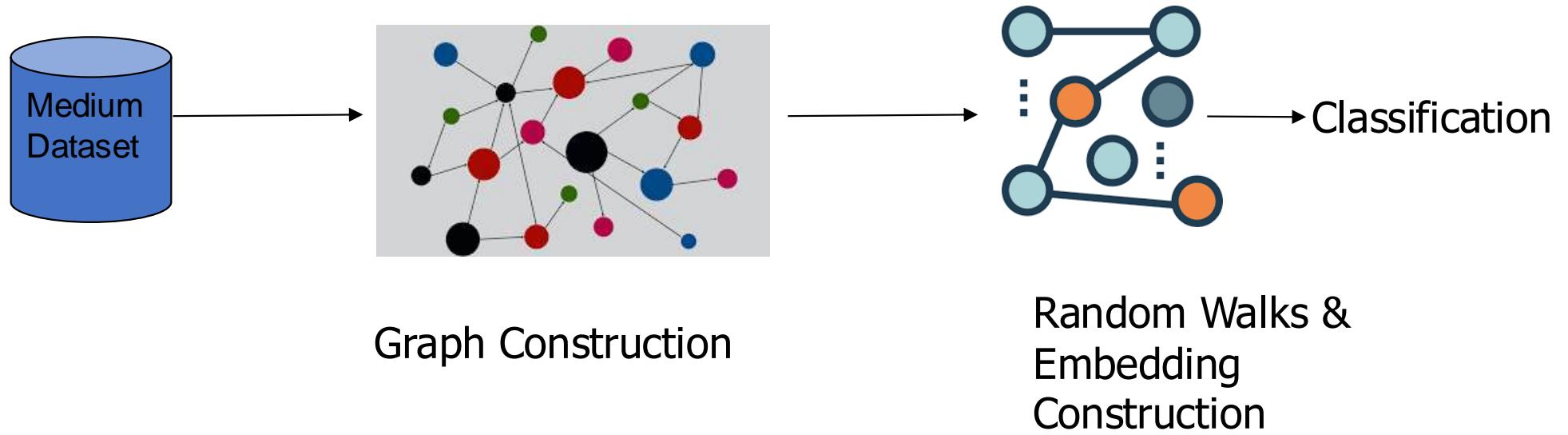


Medium articles along with subscription lists

Labels: Topic tags

Task: 3-way classification to the topics software-dev, Ai and UX

Assignment Pipeline



Assignment: Spectral embeddings

**How to represent nodes as low-dimensional vectors
using eigenvalues and eigenvectors?**

1. Construct the (unnormalized) Laplacian of the graph
2. Compute its eigenvalues and eigenvectors
3. Use them to obtain node representations

Assignment: Random walks

Random walks are used by popular node embedding approaches (Node2Vec, DeepWalk).

1. Start from any node in the graph, and randomly traverse the graph along its edges
2. Do this for all nodes (fixed length walks)

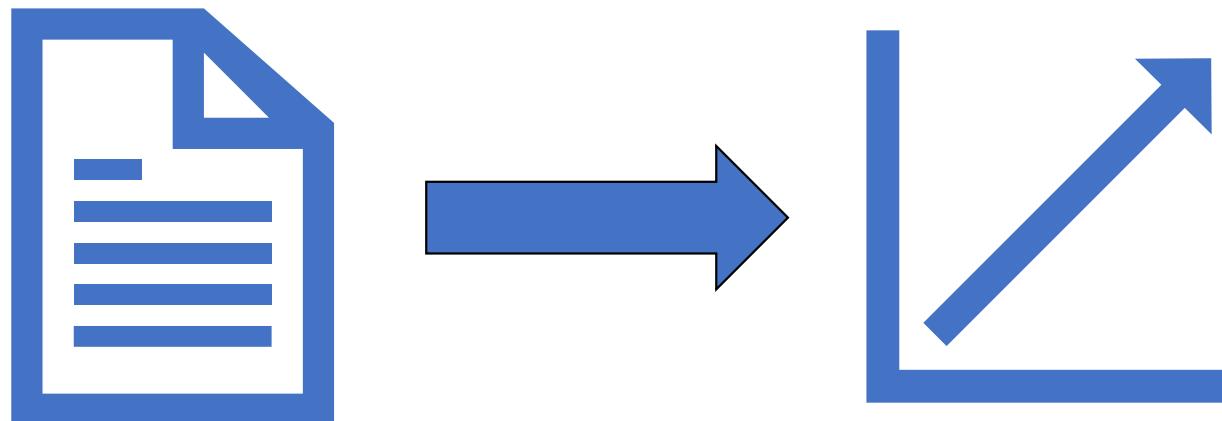
CS2525 – More Embeddings & Locality Sensitive Hashing

Machine Learning from text

- Given a collection of text documents, we want to find similarities, and apply machine learning algorithms
- Q: How?

Machine Learning from text

- Given a collection of text documents, we want to find similarities, and apply machine learning algorithms
- Q: How?



Bag of Words

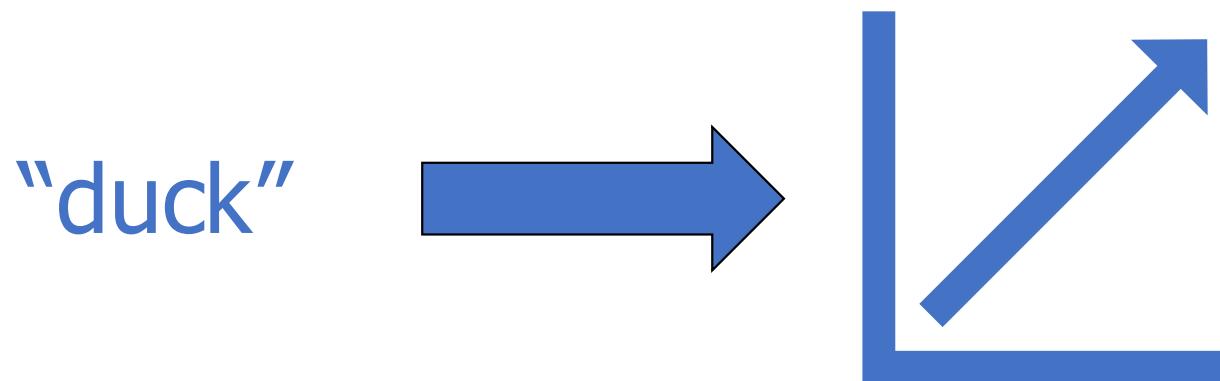
- For a given text, count the frequency of each word
- This gives a feature vector:

john	likes	to	watch	movies	mary	too
1	2	1	1	2	1	1

- for "*John likes to watch movies. Mary likes movies too.*"
- This process is called **feature extraction**
- Afterwards, you can run any classifier on the obtained data

Word Embeddings: Word2Vec

- Sometimes, we want to find similar/predict words



Word Embeddings: Word2Vec

- Key idea: *context provides meaning*

If it walks like a **duck** and quacks like a **duck**, it must be a **duck**

Word Embeddings: Word2Vec

- Key idea: *context provides meaning*

If it walks like a duck and quacks like a duck, it must be a duck

Word Embeddings: Word2Vec

- Key idea: *context provides meaning*

If it walks like a duck and quacks like a duck, it must be a duck

Word Embeddings: Word2Vec

- Key idea: *context provides meaning*

If it walks like a duck and quacks like a duck, it must be a duck

- Contexts:

1. like a __ and quacks
2. like a __ it must
3. be a __

- Q: How to use these contexts?

Word Embeddings: Word2Vec

- Key idea: *context provides meaning*

If it walks like a duck and quacks like a duck, it must be a duck

- Contexts:

1. like a __ and quacks
2. like a __ it must
3. be a __

Build a Bag of Words!

- Q: How to use these contexts?

Word Embeddings: Word2Vec

like	a	and	quacks	it	must	be
1	1	1	1	0	0	0
1	1	0	0	1	1	0
0	1	0	0	0	0	1

1. like a __ and quacks
2. like a __ it must
3. be a __

Word Embeddings: Word2Vec

like	a	and	quacks	it	must	be
1	1	1	1	0	0	0
1	1	0	0	1	1	0
0	1					

A simple version of Word2Vec

--

1. like a __ and quack
2. like a __ it must
3. be a __

You can
Differentiate past and future
Assign different weights to words
Use a NN to produce the mapping

...

Word Embeddings: Word2Vec

like	a	and	quacks	it	must	be
1	1	1	1	0	0	0
1	1	0	0	1	1	0
0	1					

A simple version of Word2Vec

1. like a __ and quack
2. like a __ it must
3. be a __

Key idea of an embedding:

Convert objects to vectors, based
on some notion of similarity

Large tables

like	a	and	quacks	it	must
1	1	1	1	0	0
1	1	0	0	1	1
0	0	0	0	1	1

- These tables can become really **REALLY LARGE**
- Q: Why is this a problem?
 - run-time, memory
 - curse of dimensionality
- Q: What to do about this?

Large tables

like	a	and	quacks	it	must
1	1	1	1	0	0
1	1	0	0	1	1
0	0	0	0	1	1

- These tables can become really **REALLY LARGE**
- Q: Why is this a problem?
 - run-time, memory
 - curse of dimensionality
- Q: What to do about this?
 - ***use less words!***
 - ***use sparse representations!***
 - ***approximate!***

Which words are important?

John	likes	to	watch	movies	Mary	too
1	2	1	1	2	1	1

Which words are important?

John	likes	to	watch	movies	Mary	too
1	2	1	1	2	1	1

- Frequent words/terms

Which words are important?

John	likes	to	watch	movies	Mary	too
1	2	1	1	2	1	1

- Frequent words/terms
 - Not: and, not, to, too, for, ...
 - Not: names?
 - Not: adjectives?

Which words are important for row 1?

John	likes	to	watch	movies	Mary	too	dislikes	TV
1	2	1	1	2	1	1	0	0
1	0	1	1	0	1	1	2	2
1	0	1	1	0	1	1	2	2

Which words are important for row 1?

John	likes	to	watch	movies	Mary	too	dislikes	TV
1	2	1	1	2	1	1	0	0
1	0	1	1	0	1	1	2	2
1	0	1	1	0	1	1	2	2

- Features that occur in row 1, but not in other rows
- Features that do not occur in row 1, but occur in other rows

Which words are important?

- Combine:
 - TF - term frequency – *importance of word in document*
 - IDF - inverse document frequency – *how well word distinguishes documents*

$$TF.IDF_{ij} = TF_{ij} \times IDF_i = \frac{f_{ij}}{\max_k f_{kj}} \times \log_2 \left(\frac{N}{n_i} \right)$$

Which words are important?

$$TF.IDF_{ij} = TF_{ij} \times IDF_i = \frac{f_{ij}}{\max_k f_{kj}} \times \log_2 \left(\frac{N}{n_i} \right)$$

Term Frequency 

Inverse document frequency 

frequency of word i in document j 

maximum frequency of any word in document j 

total number of documents 

number of documents in which word i appears 

Which words are important?

$$TF.IDF_{ij} = TF_{ij} \times IDF_i = \frac{f_{ij}}{\max_k f_{kj}} \times \log_2 \left(\frac{N}{n_i} \right)$$

normalized term frequency

inverse document frequency

how important word i is in document j

how rare it is for a document to contain word i

f_{ij}

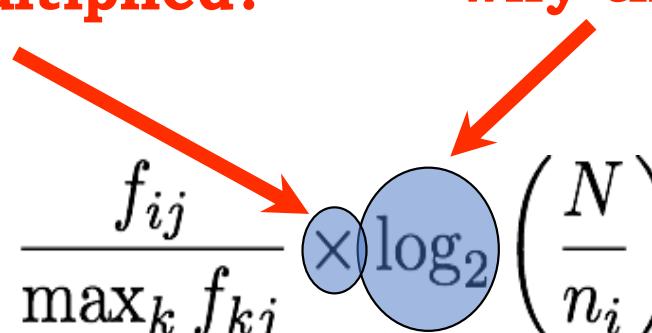
$\max_k f_{kj}$

$\log_2 \left(\frac{N}{n_i} \right)$

Which words are important?

$$TF.IDF_{ij} = TF_{ij} \times IDF_i = \frac{f_{ij}}{\max_k f_{kj}} \times \log_2 \left(\frac{N}{n_i} \right)$$

why multiplied? why this log?



Which words are important?

$$TF.IDF_{ij} = TF_{ij} \times IDF_i = \frac{f_{ij}}{\max_k f_{kj}} \times \log_2 \left(\frac{N}{n_i} \right)$$

why multiplied? **why this log?**



1. Words need large *TF* AND *IDF*
2. Decreases the effect of large *N*, *TF* and *IDF* should have similar ranges

Which words are important?

Important theme in data mining:

is log?

Find features and metrics that give desired properties in practice!

Why it works in theory is often worked out later

2.

Side story - BM 25 (not exam content)

- The most frequently used TF*IDF like document ranker

$$\text{score}(D, Q) = \sum_{i=1}^n \text{IDF}(q_i) \cdot \frac{f(q_i, D) \cdot (k_1 + 1)}{f(q_i, D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)}$$

- f is frequency, avgdl is average document length
- A larger score means that document D is more relevant to search query Q
- This has been used by search engines for many years, now finally machine learned functions start to outperform this ranking

Side story - BM 25 (not exam content)

- The most frequently used TF*IDF like document ranker

scor

Interesting is where the name BM25 comes from

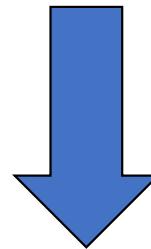
- f
- A
- 25 stands for the number of the experiment that they ran when trying to find a good ranker
- M

Shingling (aka Ngrams)

John	likes	to	watch	movies	Mary	too
1	2	1	1	2	1	1

Shingling (aka Ngrams)

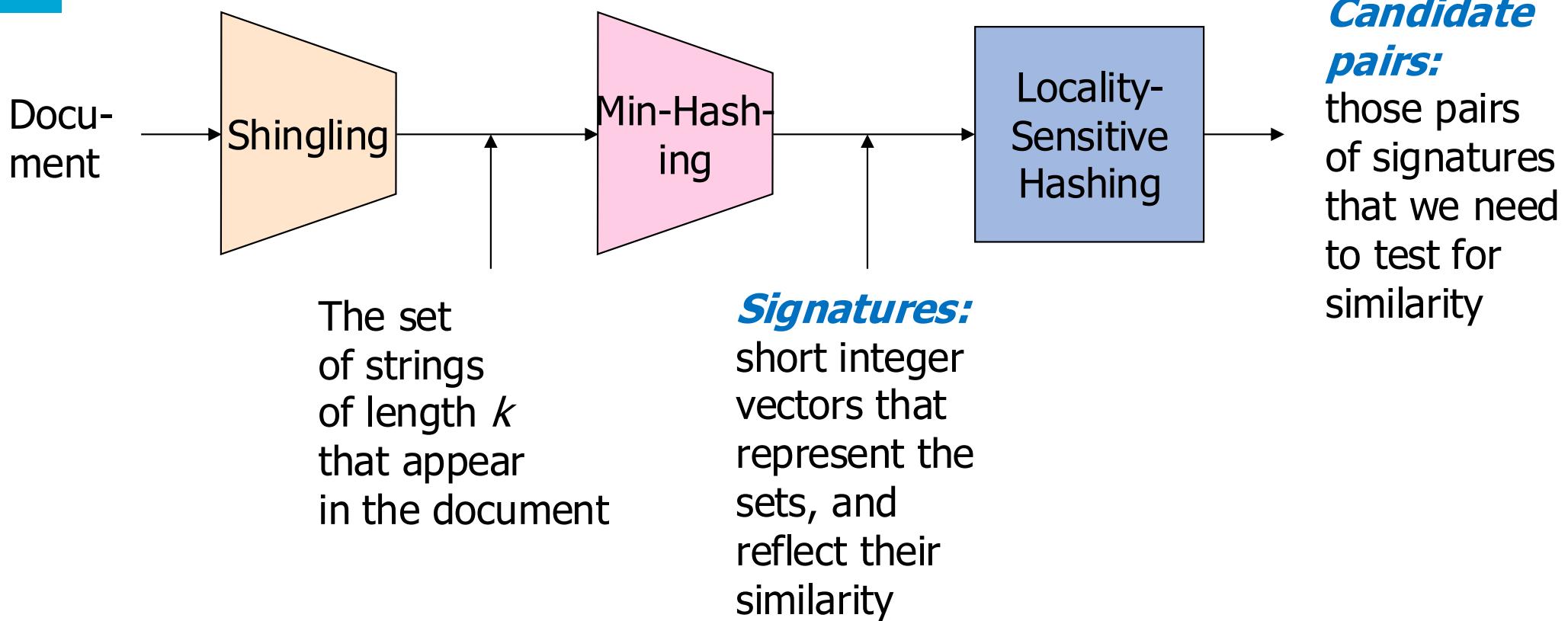
John	likes	to	watch	movies	Mary	too
1	2	1	1	2	1	1



Jo	oh	hn	n_	_l	li	ik	ke	es	s_	_t	to	o_
1	1	1	1	1	1	1	1	2	2	2	2	2

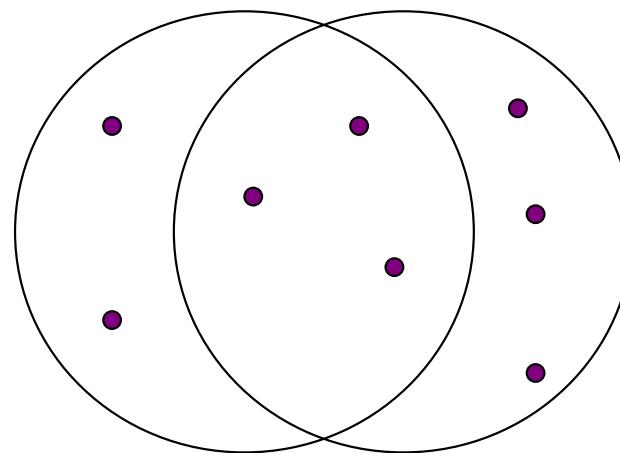
- is simply to make counts for all substrings of length k

Finding similar items



Jaccard Similarity of Sets

- The Jaccard similarity of two sets is the size of their intersection divided by the size of their union.
 - $\text{Sim } (C_1, C_2) = |C_1 \cap C_2| / |C_1 \cup C_2|.$



3 in intersection.
8 in union.
Jaccard similarity
 $= 3/8$

Four Types of Rows

- Given columns C1 and C2 (documents or contexts), rows may be classified as:

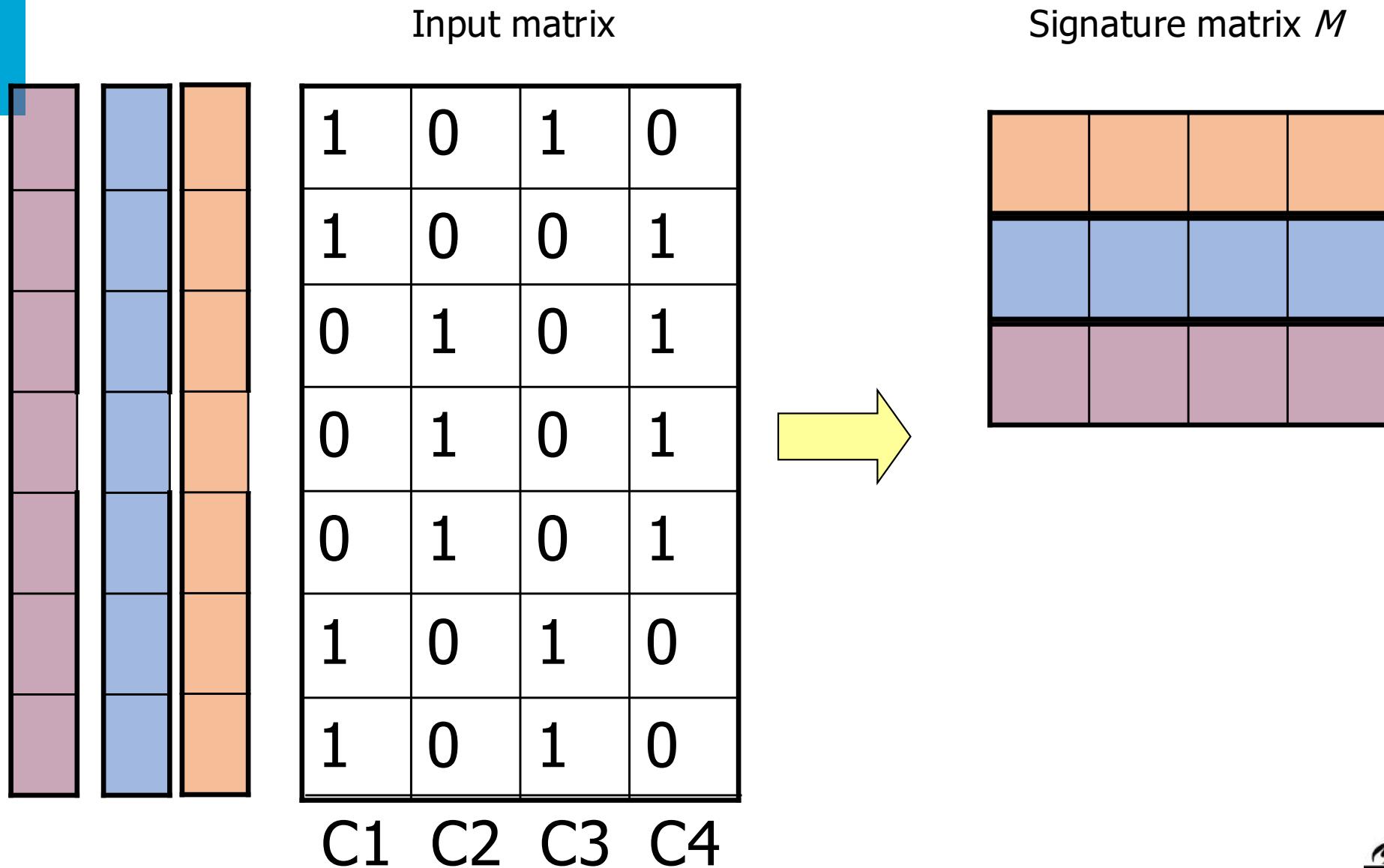
	C1	C2
a	1	1
b	1	0
c	0	1
d	0	0

- Also, $a = \# \text{ rows of type } a$, etc.
- Note $\text{Sim}(C1, C2) = a / (a + b + c)$.

Minhashing

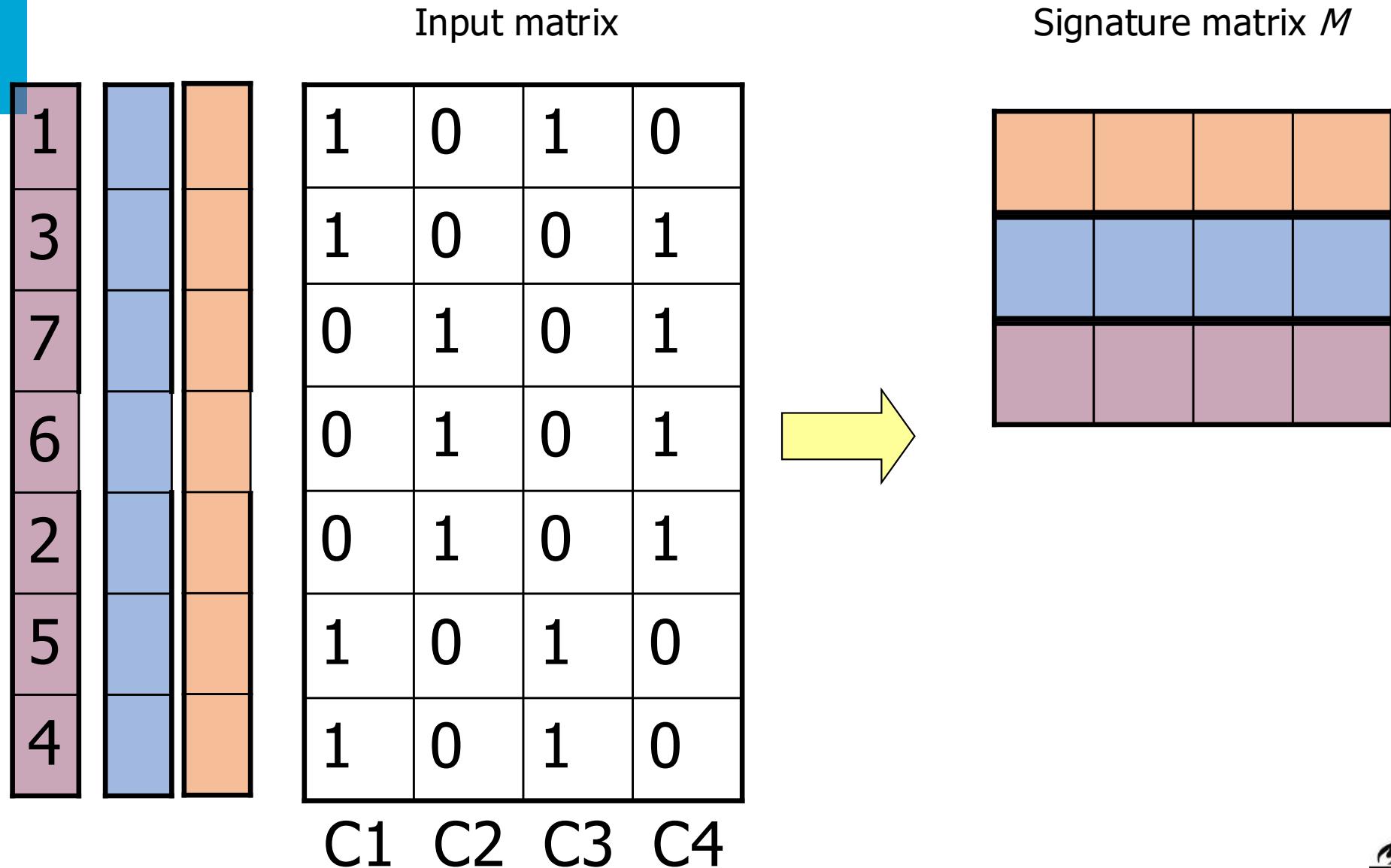
- Imagine the rows permuted randomly.
- Define “hash” function $h(C)$
 - the row number of the first (in the permuted order) row in which column C has 1.
- Use several (e.g., 100) independent hash functions to create a signature.

Minhashing Example

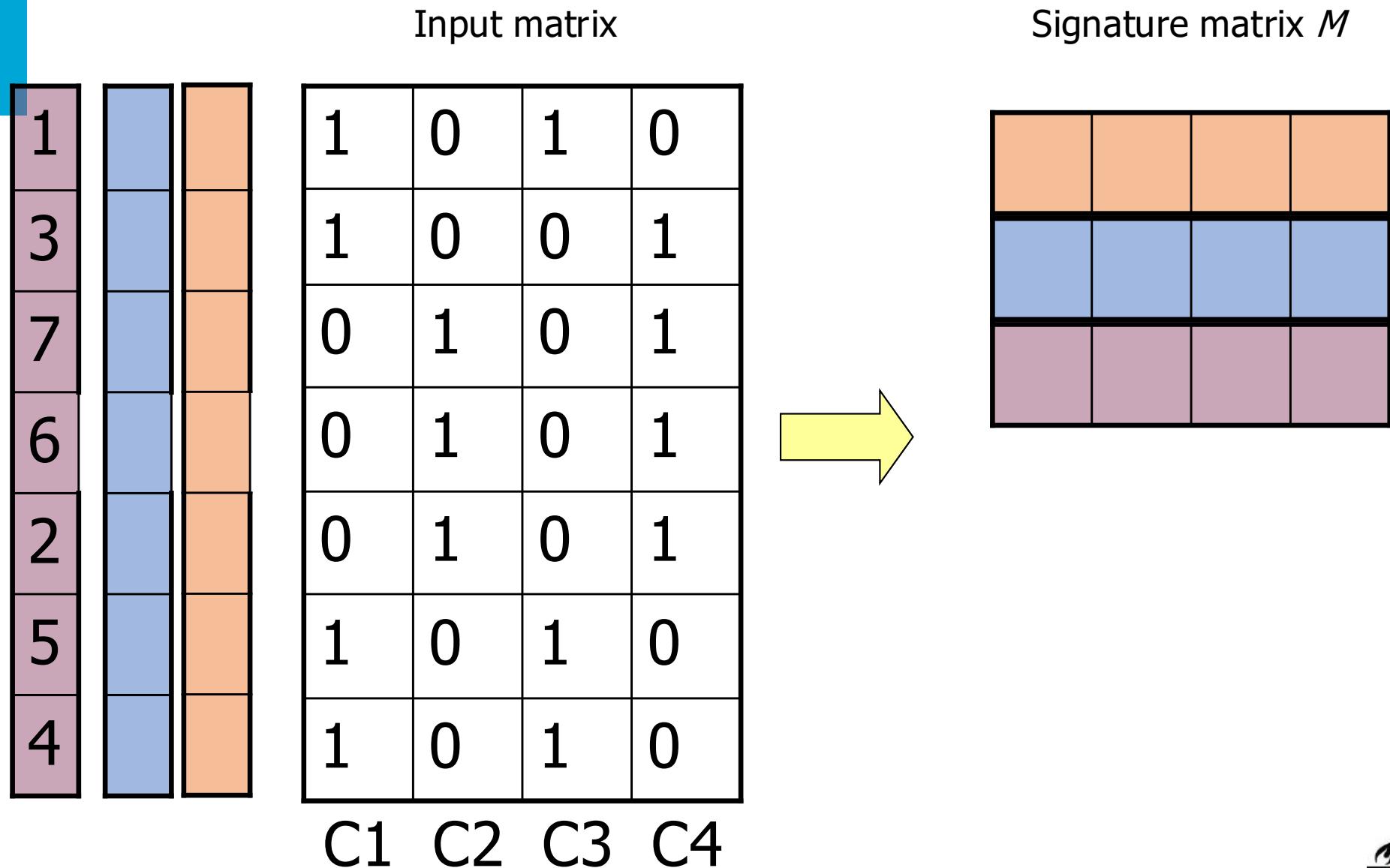


We start with a new random order

Example



Look in each input column for the first 1
in the new order



Look in each input column for the new order

Input matrix

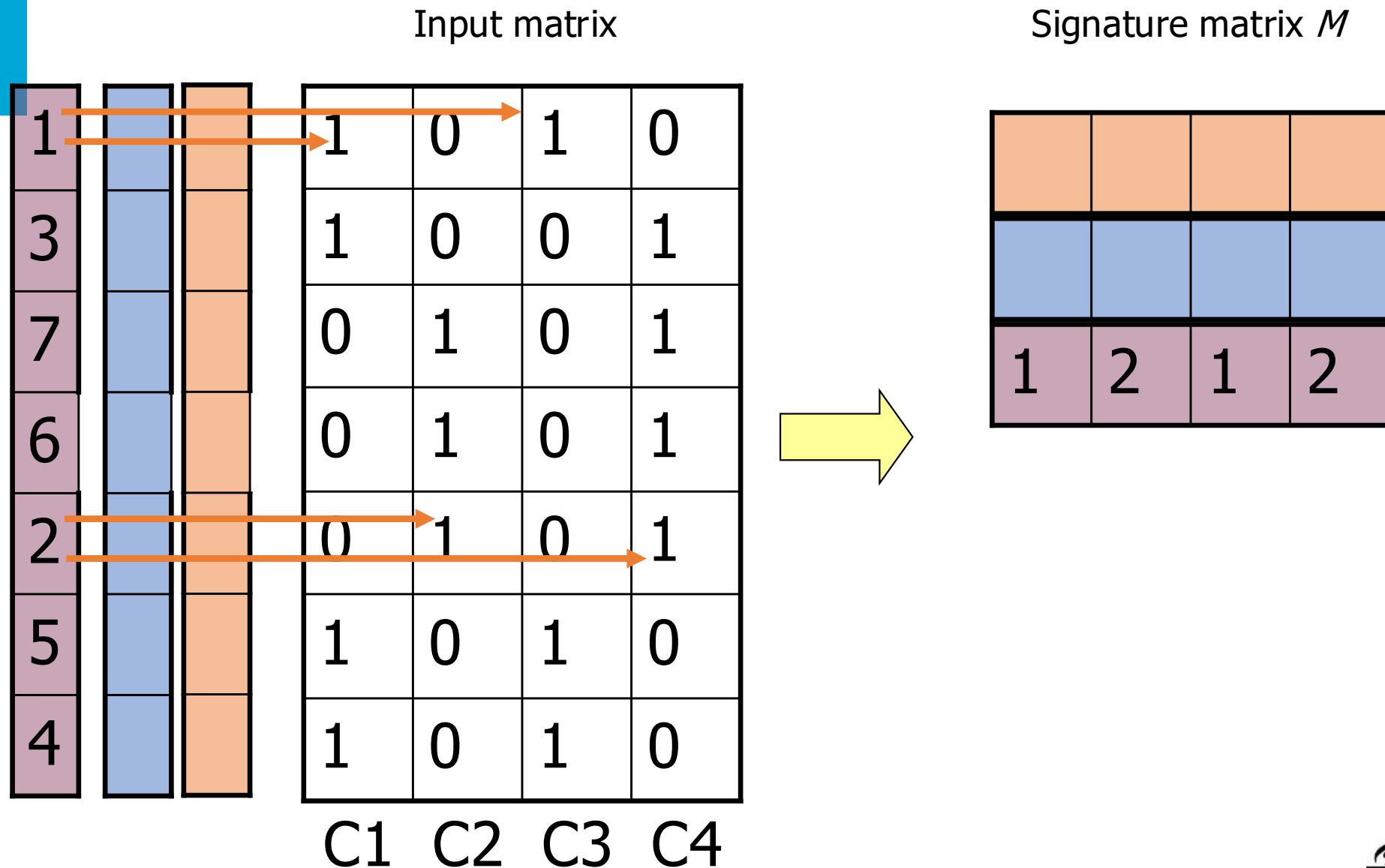
The figure consists of three vertical bars. The leftmost bar is purple and has a black border. It contains seven horizontal segments, each with a black outline. The segments are labeled with the numbers 1, 3, 7, 6, 2, 5, and 4 from top to bottom. The middle bar is blue and the rightmost bar is orange, both with black outlines. Each of these two bars is divided into four equal segments by black horizontal lines.

1	0	1	0
1	0	0	1
0	1	0	1
0	1	0	1
0	1	0	1
1	0	1	0
1	0	1	0

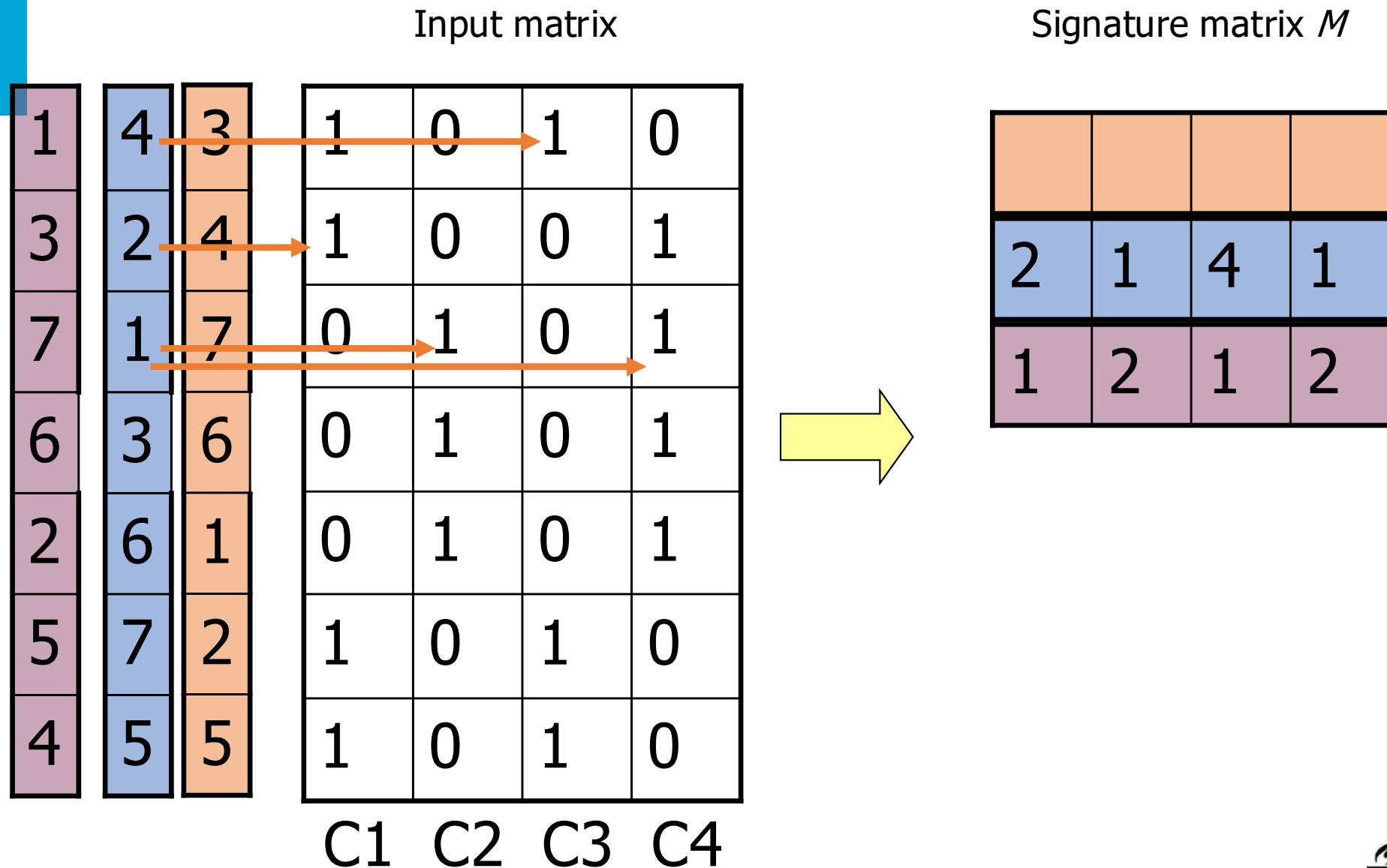
Resorted the table becomes:

1	0	1	0
0	1	0	1
1	0	0	1
1	0	1	0
1	0	1	0
0	1	0	1
0	1	0	1

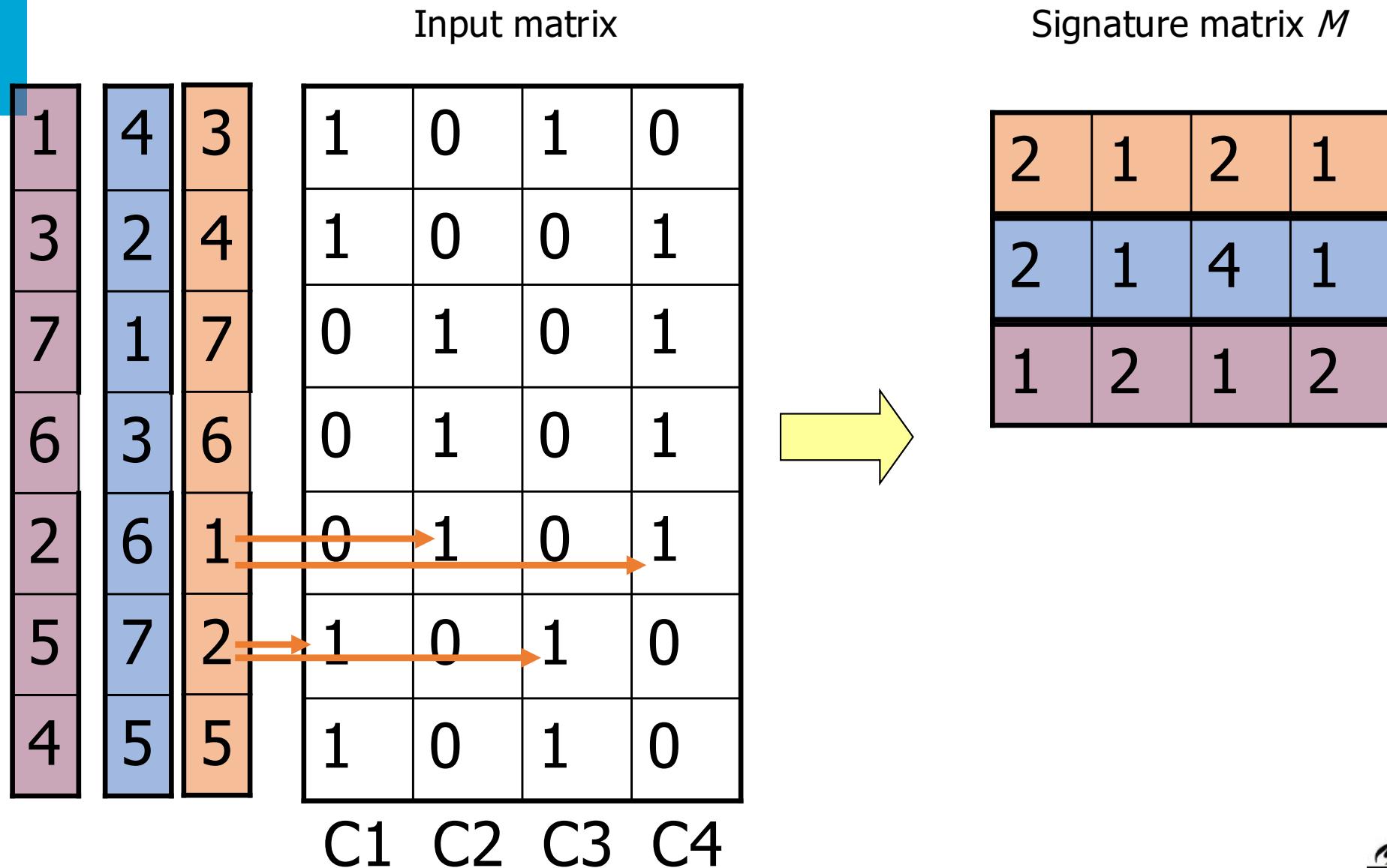
Fill in the smallest number from the order
with a 1 value in the signature



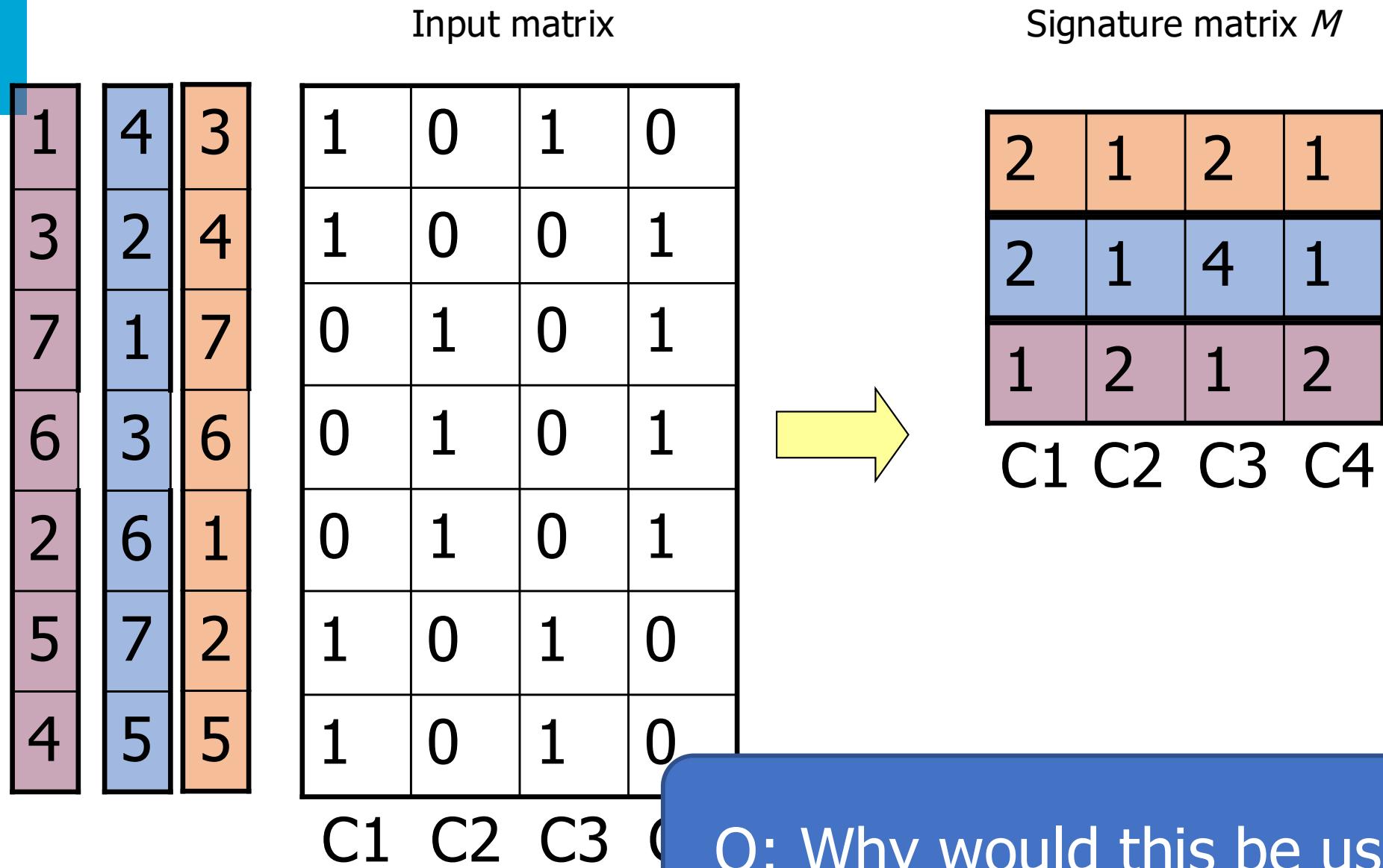
Minhashing Example



Minhashing Example



Minhashing Example



Surprising Property

- The probability (over all permutations of the rows) that $h(C_1) = h(C_2)$ is the same as $\text{Sim}(C_1, C_2)$.

	C1	C2
a	1	1
b	1	0
c	0	1
d	0	0

- Both are $a / (a + b + c)$!
- (a = number of a-type rows, etc.)
- **Why?**

Surprising Property

- The probability (over all permutations of the rows) that $h(C_1) = h(C_2)$ is the same as $\text{Sim}(C_1, C_2)$.

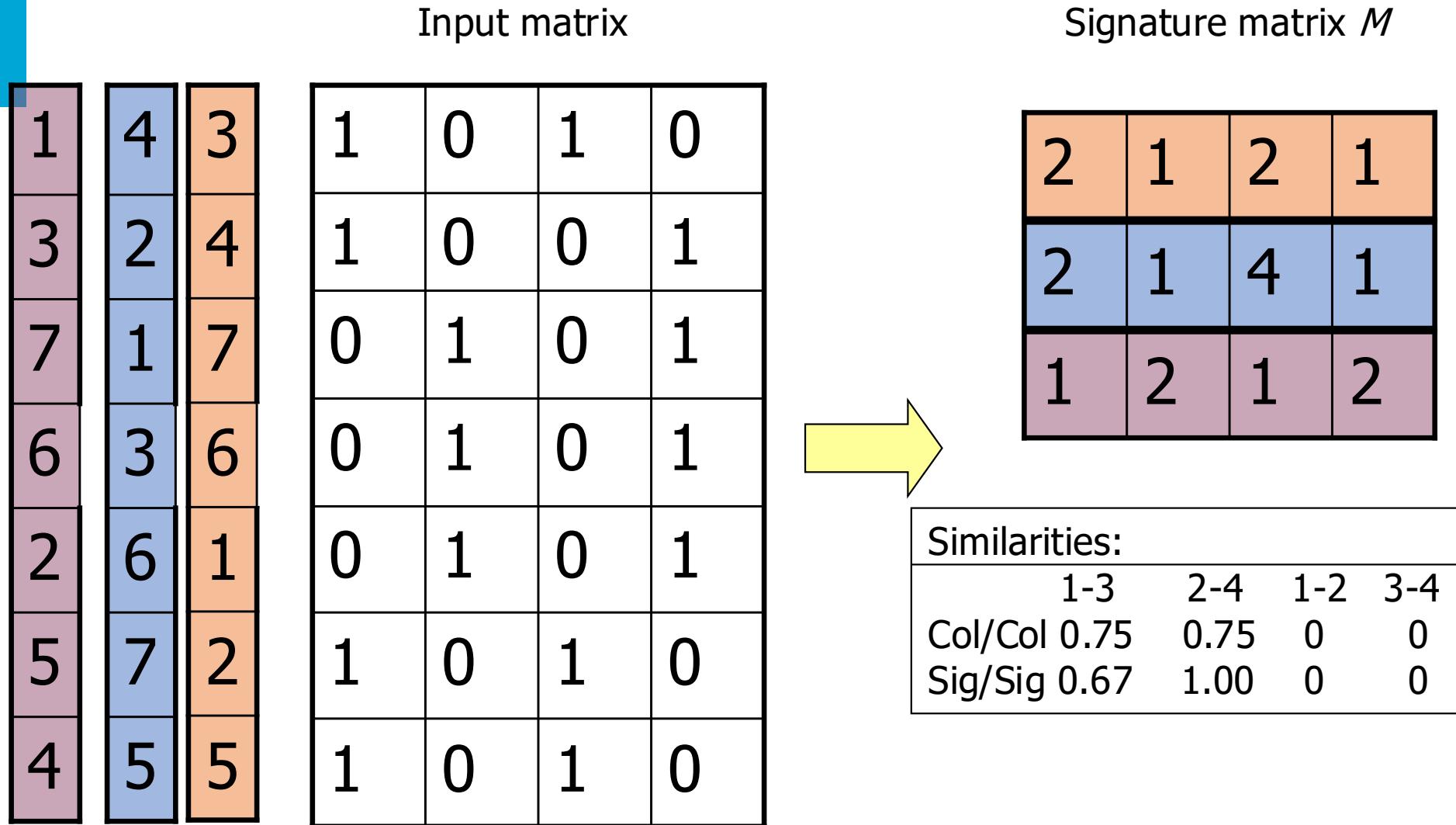
- Both are $a / (a + b + c)!$
 - (a = number of a-type rows, etc.)
- Why?
 - Look down the permuted columns C_1 and C_2 until we see a 1.
 - Every row has equal probability to be the first with a 1.
 - If it is a type-a row, then $h(C_1) = h(C_2)$.
 - If it is a type-b or type-c row, then not.
 - Type d rows have no 1s
 - $P(h(C_1) = h(C_2)) = \#success / \#options = a / (a + b + c)$

	C1	C2
a	1	1
b	1	0
c	0	1
d	0	0

Hash using Signatures

- Key idea: “hash” each column C to a small signature $\text{Sig}(C)$, such that:
 1. $\text{Sig}(C)$ is small enough that we can fit a signature in main memory for each column.
 2. $\text{Sim}(C_1, C_2)$ is the same as “similarity” of $\text{Sig}(C_1)$ and $\text{Sig}(C_2)$.
 - i.e., *Sig() is locality sensitive!*

Min Hashing – Example



Implementation: Hashing!

- Suppose 1 billion rows.
 - Hard to pick a random permutation from 1...billion.
- A good approximation to permuting rows:
 - pick 100 (?) hash functions.
- For each column c and each hash function h_i , keep a "slot" $M(i, c)$.
- Intent: $M(i, c)$ will become the smallest value of $h_i(r)$ for which column c has 1 in row r .

Computing minhashes efficiently

- Pseudo-code implementation for computing a minhash efficiently:

```
for each row  $r$  do begin
    for each hash function  $h_i$  do
        compute  $h_i(r)$ ;
    for each column  $c$ 
        if  $c$  has 1 in row  $r$ 
            for each hash function  $h_i$  do
                if  $h_i(r)$  is smaller than  $M(i, c)$  then
                     $M(i, c) := h_i(r)$ ;
end;
```

Example

Row	C1	C2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

	Sig1	Sig2
$h(1) = 1$	1	-
$g(1) = 3$	3	-

$$h(x) = x \bmod 5$$
$$g(x) = 2x+1 \bmod 5$$

Example

Row	C1	C2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

	Sig1	Sig2
$h(1) = 1$	1	-
$g(1) = 3$	3	-
$h(2) = 2$	1	2
$g(2) = 0$	3	0

$$h(x) = x \bmod 5$$
$$g(x) = 2x+1 \bmod 5$$

Example

Row	C1	C2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

	Sig1	Sig2
$h(1) = 1$	1	-
$g(1) = 3$	3	-
$h(2) = 2$	1	2
$g(2) = 0$	3	0
$h(3) = 3$	1	2
$g(3) = 2$	2	0

$$h(x) = x \bmod 5$$
$$g(x) = 2x+1 \bmod 5$$

Example

Row	C1	C2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

	Sig1	Sig2
$h(1) = 1$	1	-
$g(1) = 3$	3	-
$h(2) = 2$	1	2
$g(2) = 0$	3	0
$h(3) = 3$	1	2
$g(3) = 2$	2	0
$h(4) = 4$	1	2
$g(4) = 4$	2	0

$$h(x) = x \bmod 5$$
$$g(x) = 2x+1 \bmod 5$$

Example

Row	C1	C2
1	1	0
2	0	1
3	1	1
4	1	0
5	0	1

$$h(x) = x \bmod 5$$
$$g(x) = 2x+1 \bmod 5$$

	Sig1	Sig2
$h(1) = 1$	1	-
$g(1) = 3$	3	-
$h(2) = 2$	1	2
$g(2) = 0$	3	0
$h(3) = 3$	1	2
$g(3) = 2$	2	0
$h(4) = 4$	1	2
$g(4) = 4$	2	0
$h(5) = 0$	1	0
$g(5) = 1$	2	0

Creating hash functions

- Universal hashing:

- $h(x, a, b) = ((ax+b) \bmod p) \bmod m$
- a is a random number between 1 to p-1 inclusive.
- b is a random number between 0 to p-1 inclusive.
- p is a prime number that is much greater than m
- m is max possible value for hash code + 1
- Changing a and b will yield independent hash functions

Nearest neighbors

Finding nearest neighbors

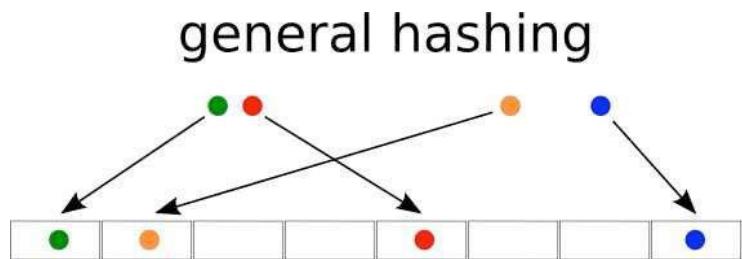
- In many situations, we want to find the most similar object given a query:
 - Find the document in large corpus that is most similar to a query document
 - Find the website on the internet that is most relevant given a query
 - Find the song that is most similar to the one currently on the radio
- This most similar object is referred to as the nearest neighbor

Finding nearest neighbors

- Suppose we want to find the webpage that is most similar to document q
- Google indexes about 60 billion webpages
 - Bing ~ 12 billion
- How many Jaccard distance computations need to be done?
- If each Jaccard distance computation takes 1000 CPU instructions, how long would the search take?
 - $1000 \times 60B \text{ webpages} / 400\ 000 \text{ MIPS} = \sim 125 \text{ seconds}$
 - Probably not what Google is doing...

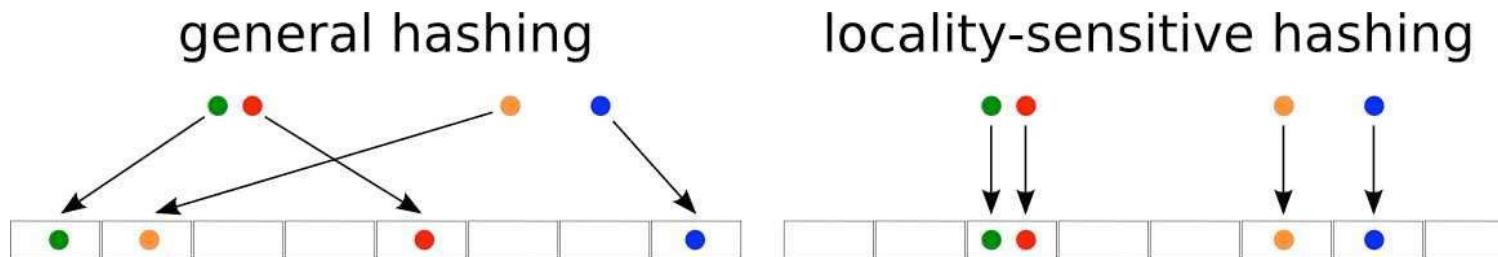
Locality-sensitive hashing

- LSH uses hashing functions that take “location” of object in consideration:



Locality-sensitive hashing

- LSH uses hashing functions that take “location” of object in consideration:

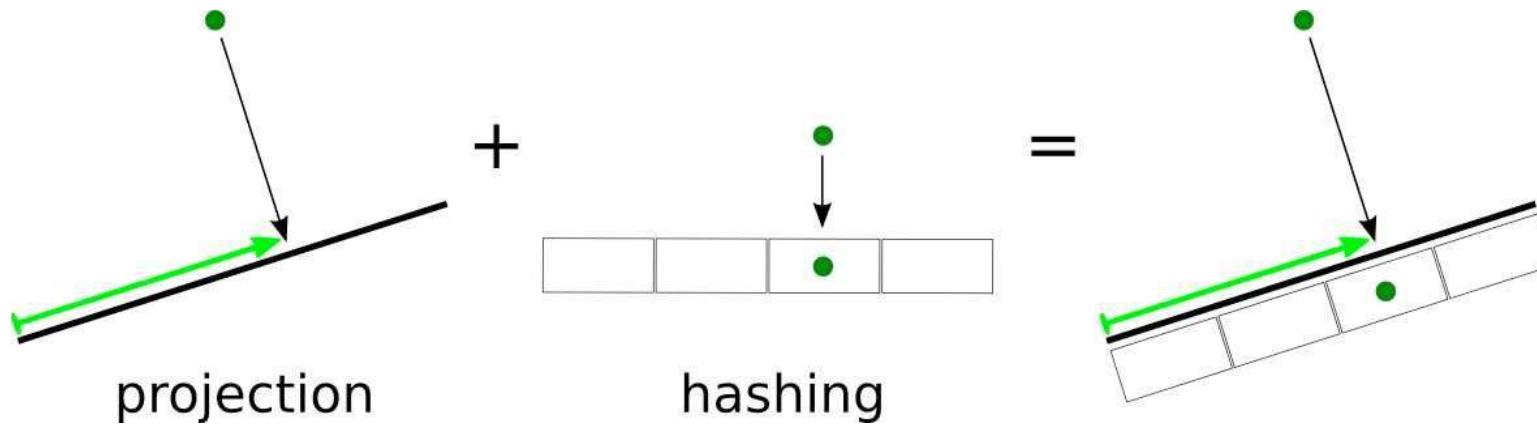


Normal vs. LSH

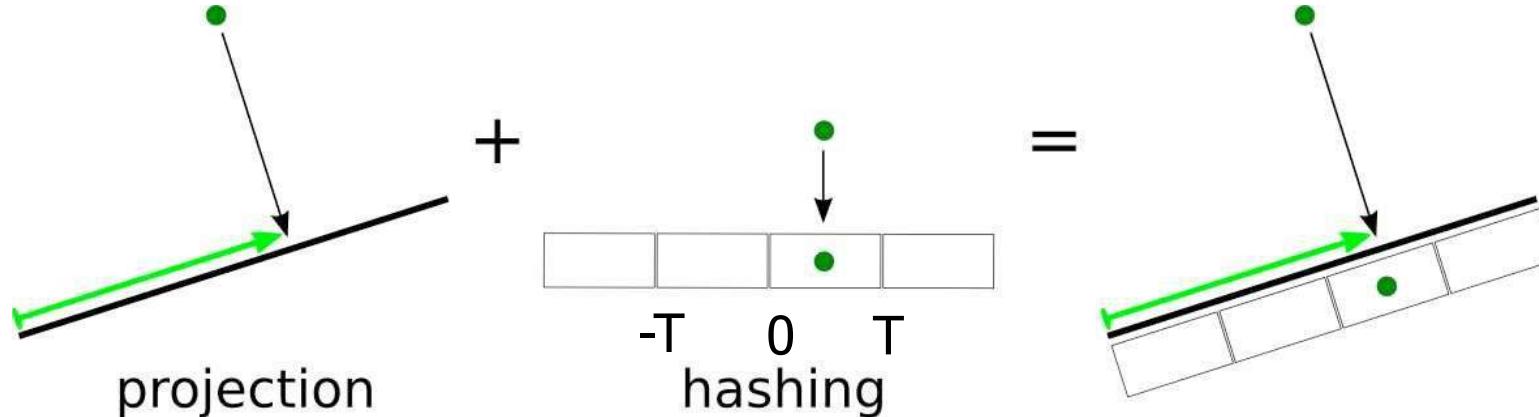
- Normal hash functions try to minimize the probability of collision
- LSH hash functions try to **maximize** probability of **similar** items colliding.

Hashing points in space

- Example of a locality-sensitive hashing function for points in a space:
 - Project the point onto a random subspace; divide result into 4 buckets (2 bits)



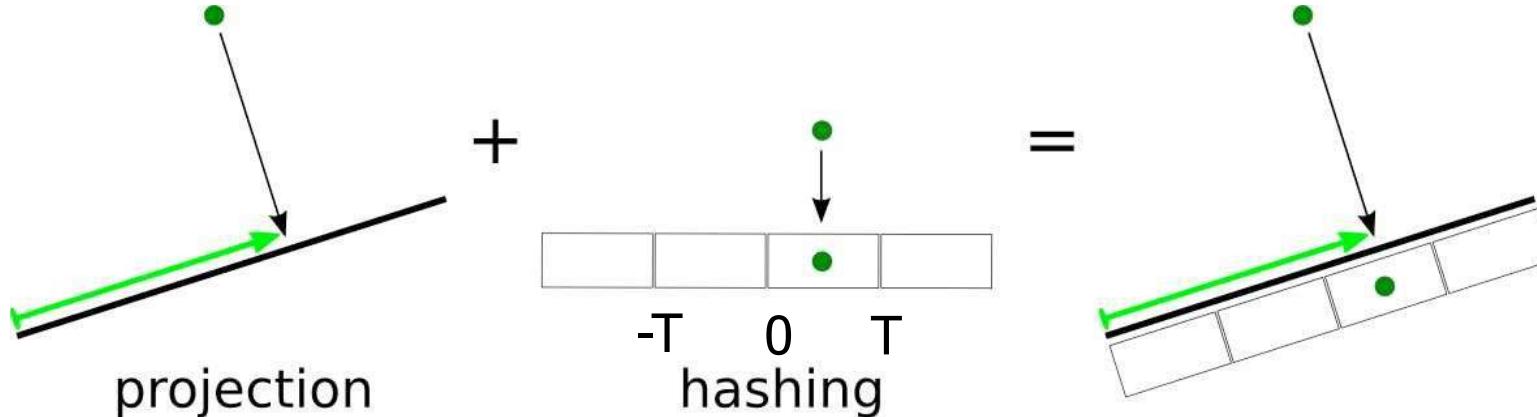
Locality-sensitive hashing



- Mathematically, we could express this locality sensitive hash function as:

$$h(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{w}^\top \mathbf{x} \leq -\tau \\ 1 & \text{if } -\tau < \mathbf{w}^\top \mathbf{x} \leq 0 \\ 2 & \text{if } 0 < \mathbf{w}^\top \mathbf{x} \leq \tau \\ 3 & \text{if } \mathbf{w}^\top \mathbf{x} > \tau \end{cases}$$

Locality-sensitive hashing



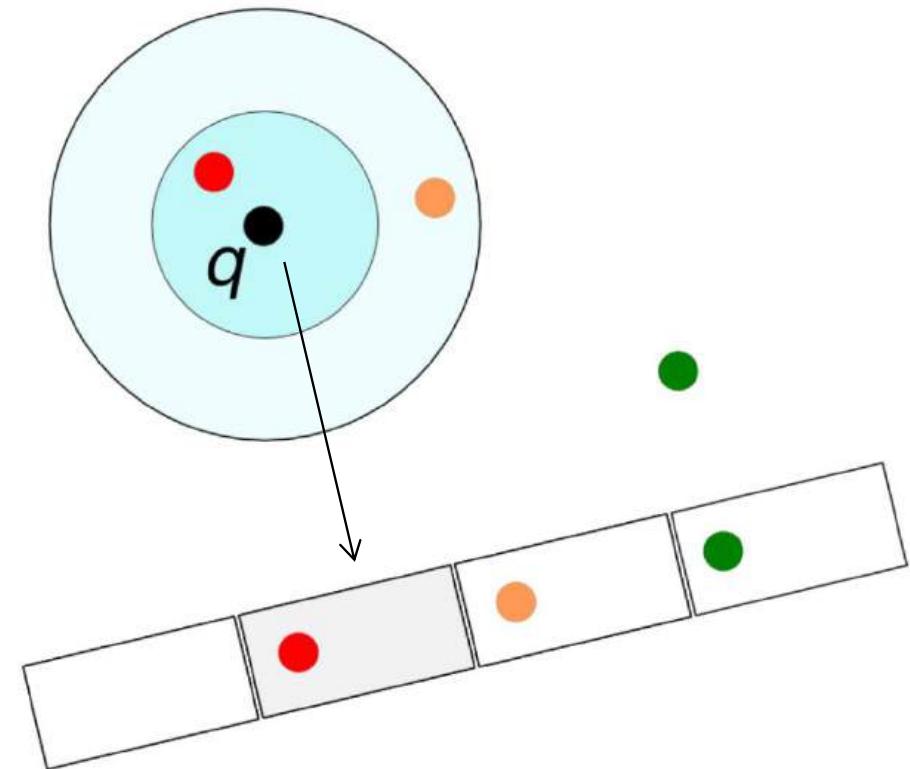
- Mathematically, we could express this locality sensitive hash function as:

$$h(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{w}^\top \mathbf{x} \leq -\tau \\ 1 & \text{if } -\tau < \mathbf{w}^\top \mathbf{x} \leq 0 \\ 2 & \text{if } 0 < \mathbf{w}^\top \mathbf{x} \leq \tau \\ 3 & \text{if } \mathbf{w}^\top \mathbf{x} > \tau \end{cases}$$

random projection **threshold parameter**

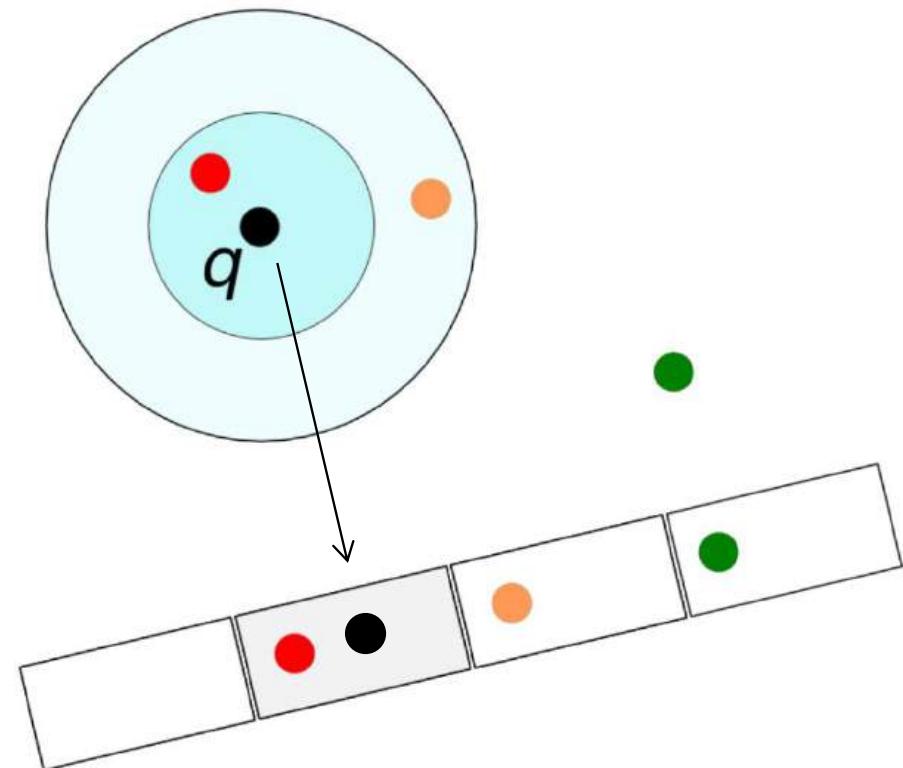
Locality-sensitive hashing

- Retrieval of nearest neighbors of a query point q using LSH works as follows:
 - Hash all data points using locality-sensitive hash
 - Compute locality-sensitive hash of query point



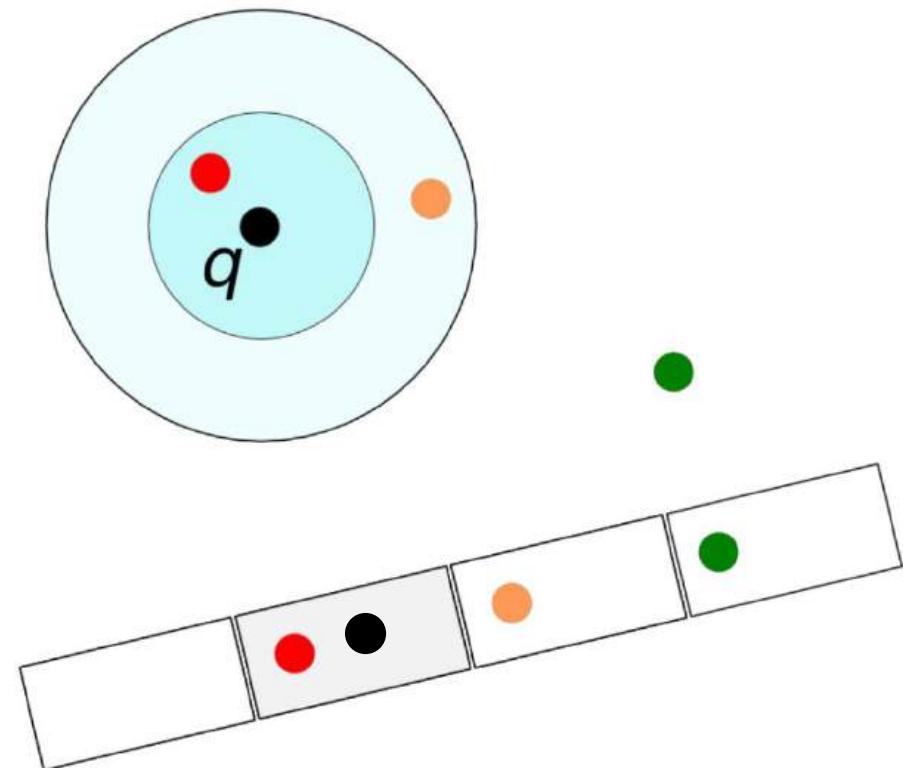
Locality-sensitive hashing

- Retrieval of nearest neighbors of a query point q using LSH works as follows:
 - Hash all data points using locality-sensitive hash
 - Compute locality-sensitive hash of query point



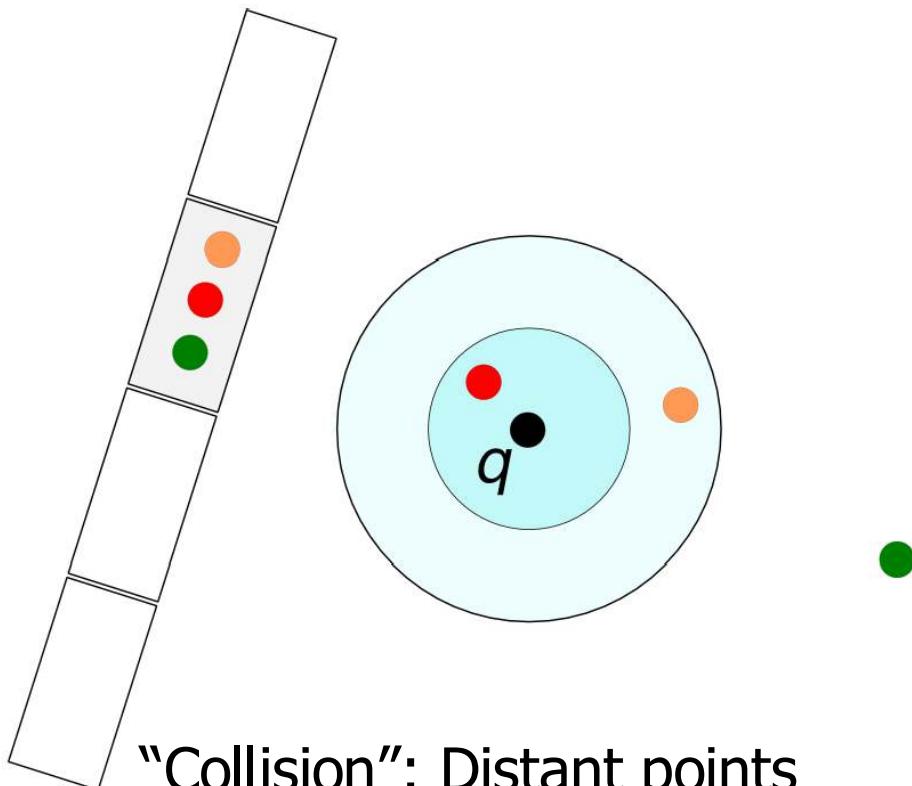
Locality-sensitive hashing

- Retrieval of nearest neighbors of a query point q using LSH works as follows:
 - All data points in the bucket are **candidate** near neighbors
 - Compute distances only to candidate points to find true nearest neighbors



Locality-sensitive hashing

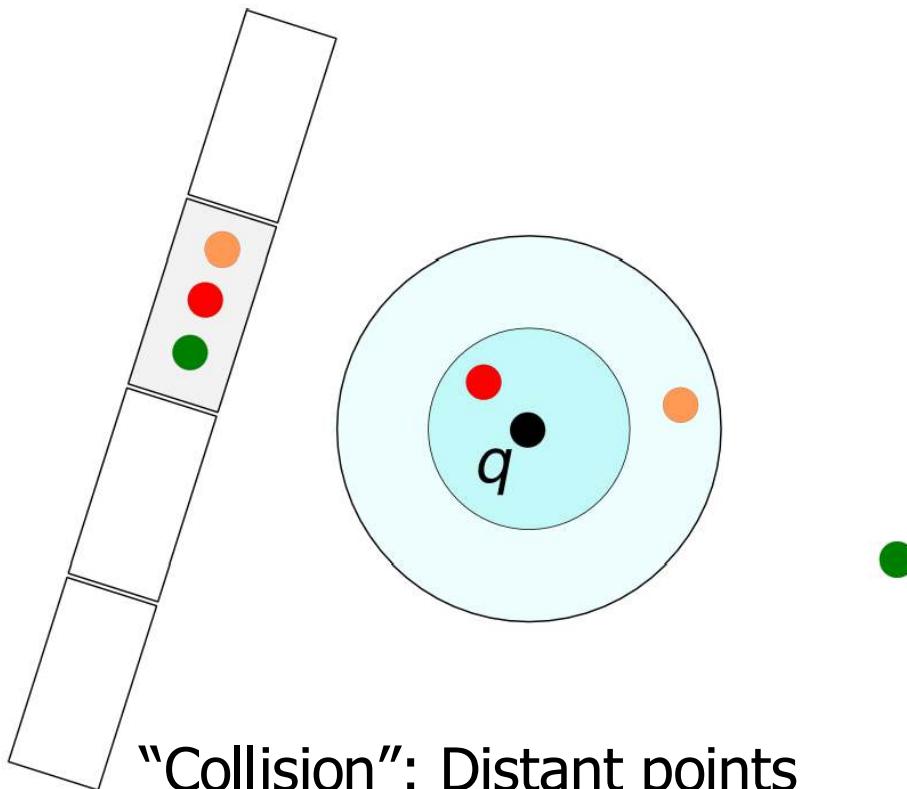
- LSH projections may be “unlucky” in two main ways:



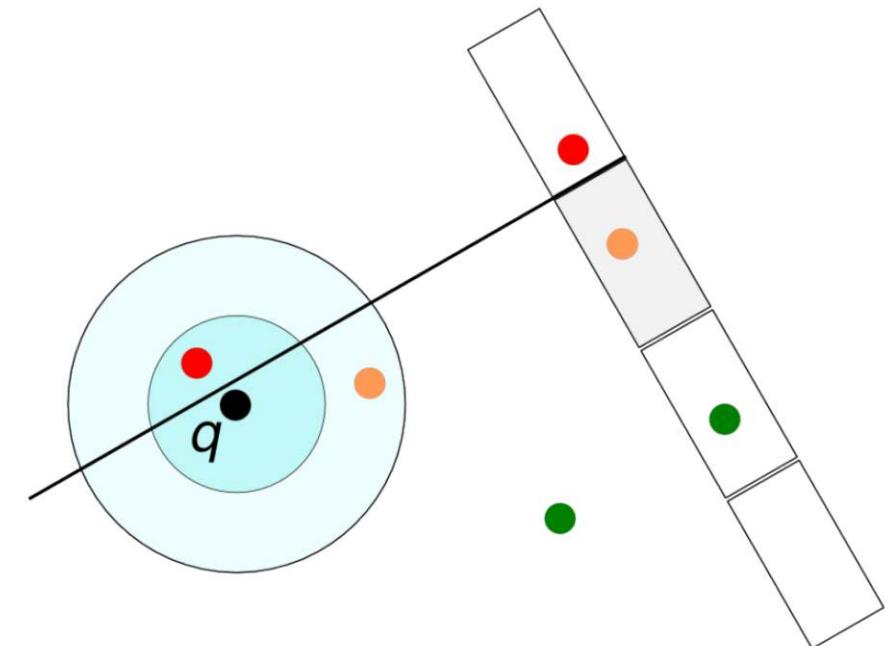
“Collision”: Distant points
hashed in the same bucket

Locality-sensitive hashing

- LSH projections may be “unlucky” in two main ways:



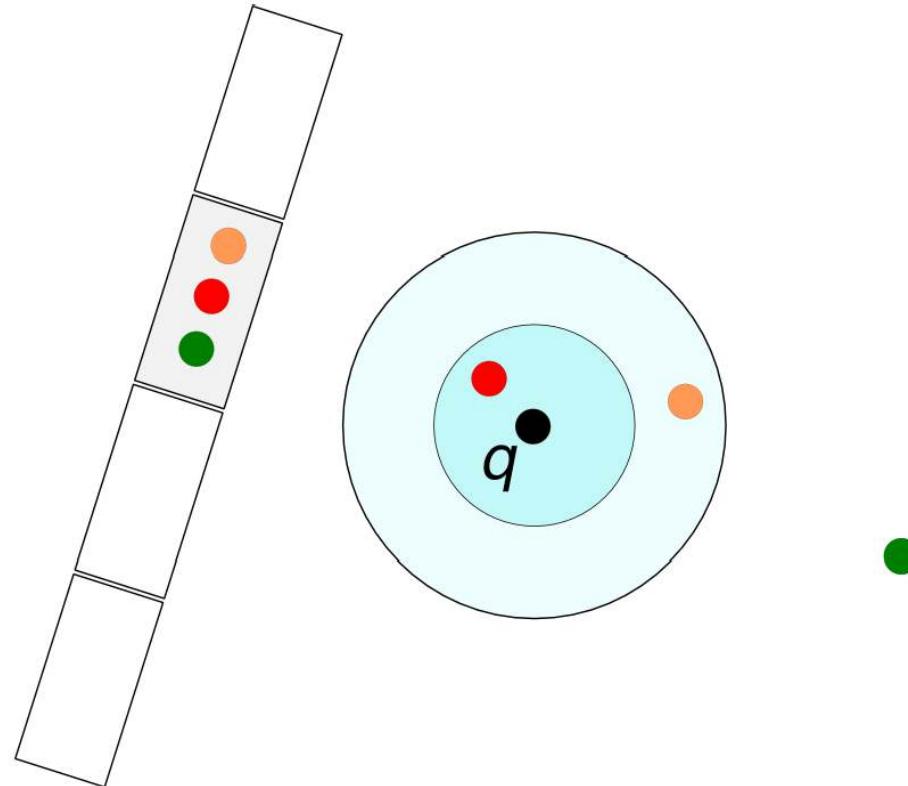
“Collision”: Distant points
hashed in the same bucket



“Split”: Nearby points
hashed in different buckets

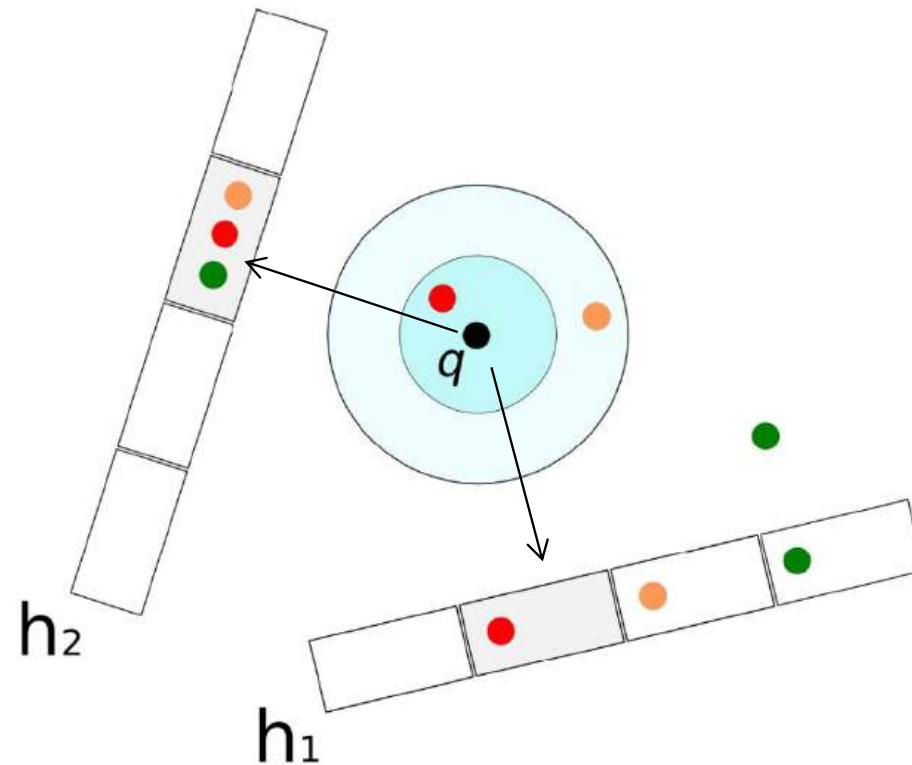
Locality-sensitive hashing: collision

- “Collision”: Distant points hashed in the same bucket



LSH: resolving collisions

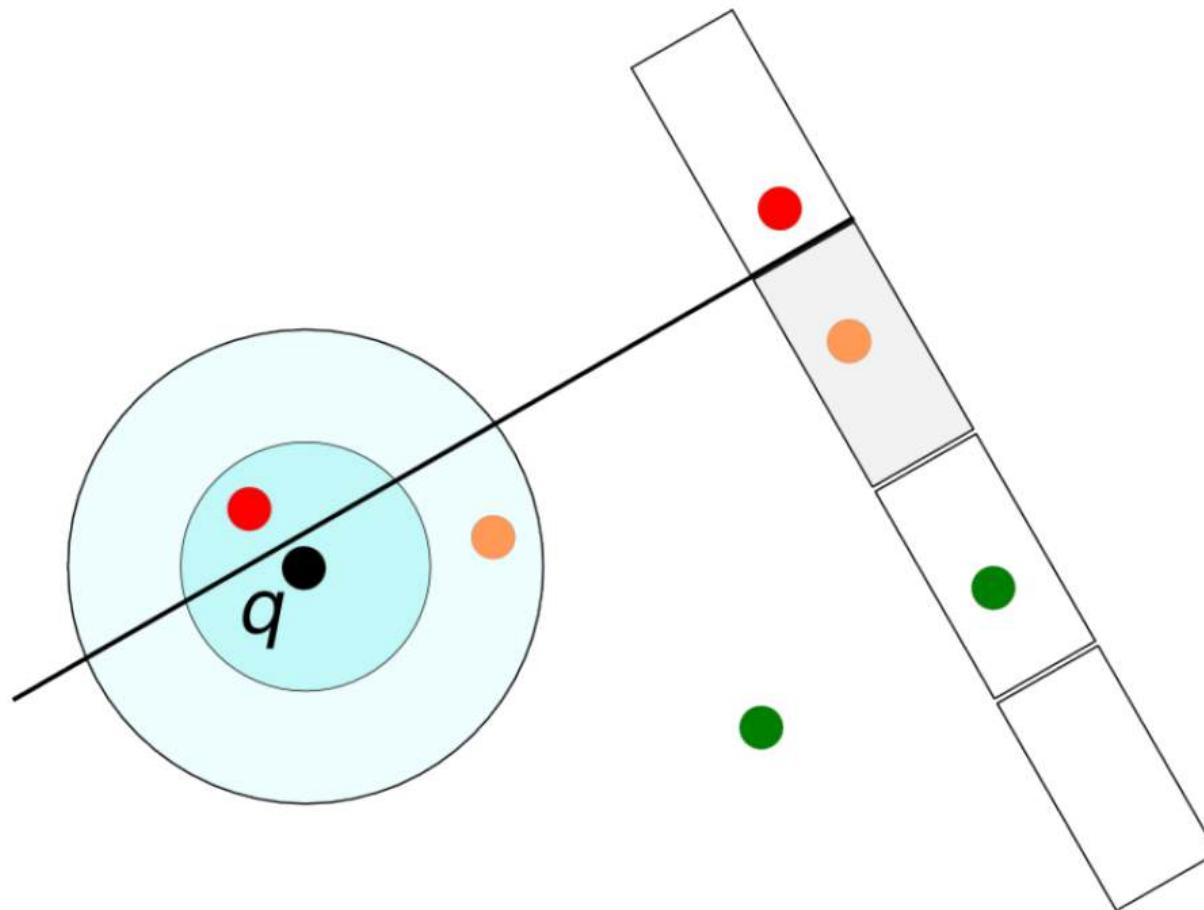
- Using multiple projections in an LSH resolves “collisions”:



- Points are candidates if they occur in all query bins
- **AND-construction**

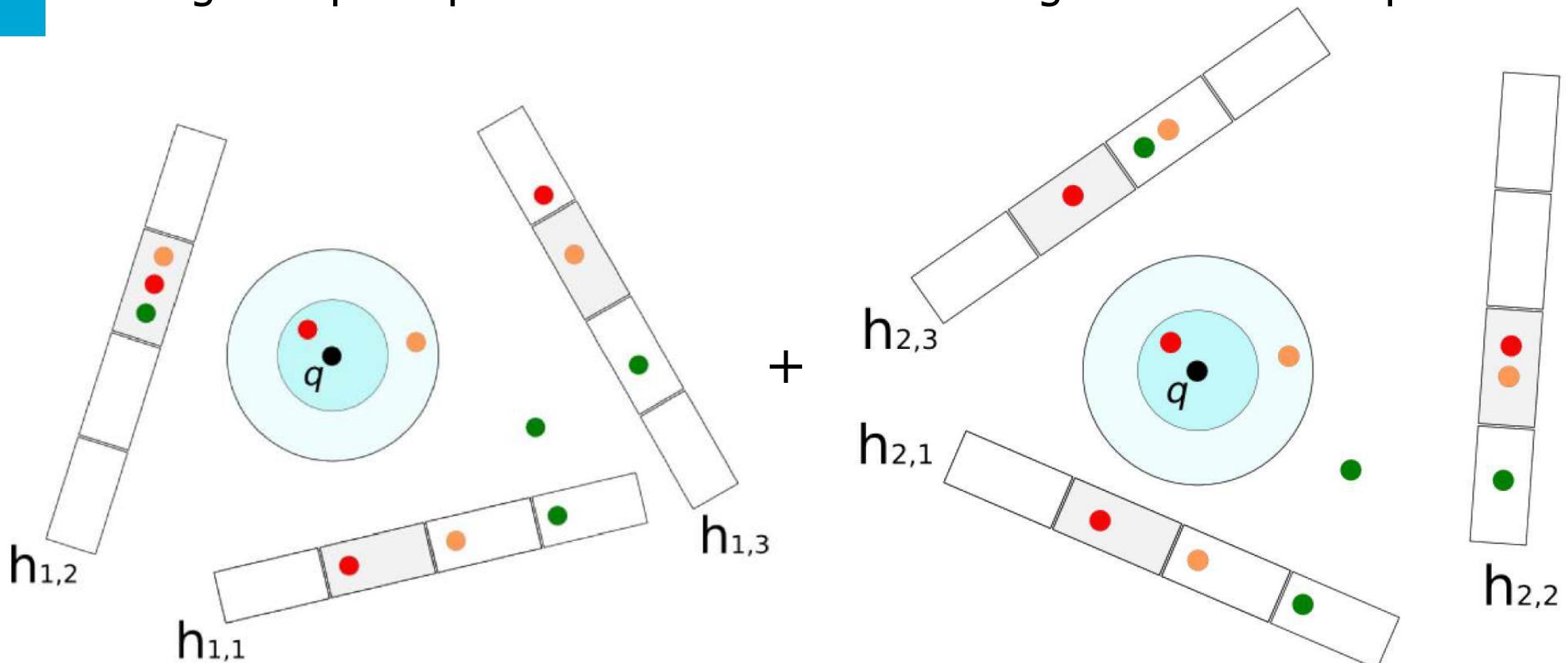
Locality-sensitive hashing: split

- “Split”: Nearby points hashed in different buckets



LSH: resolving splits

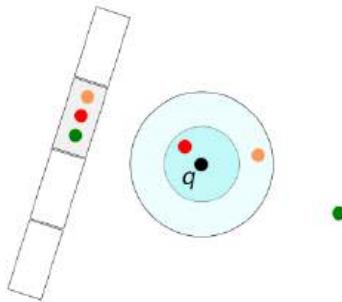
- Using multiple separate hash tables when doing LSH resolves “splits”:



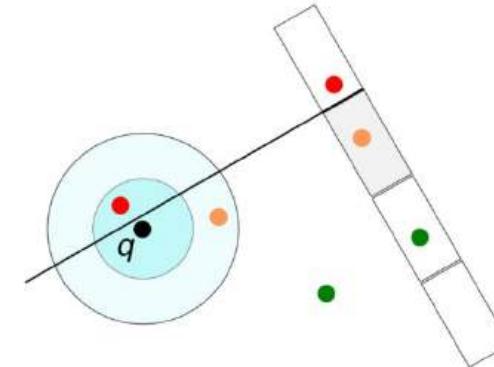
- Points are candidate neighbors if candidate in any of the hash tables
- OR-constructions**

LSH: collisions and splits

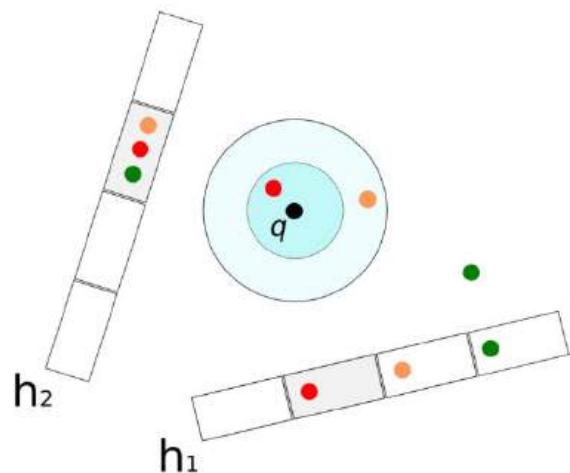
Collision



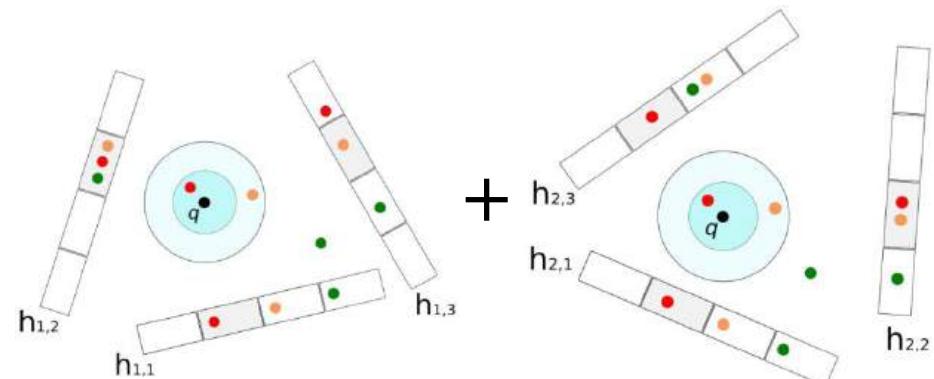
Split



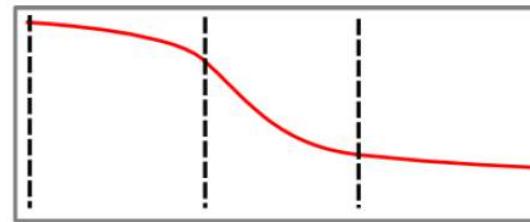
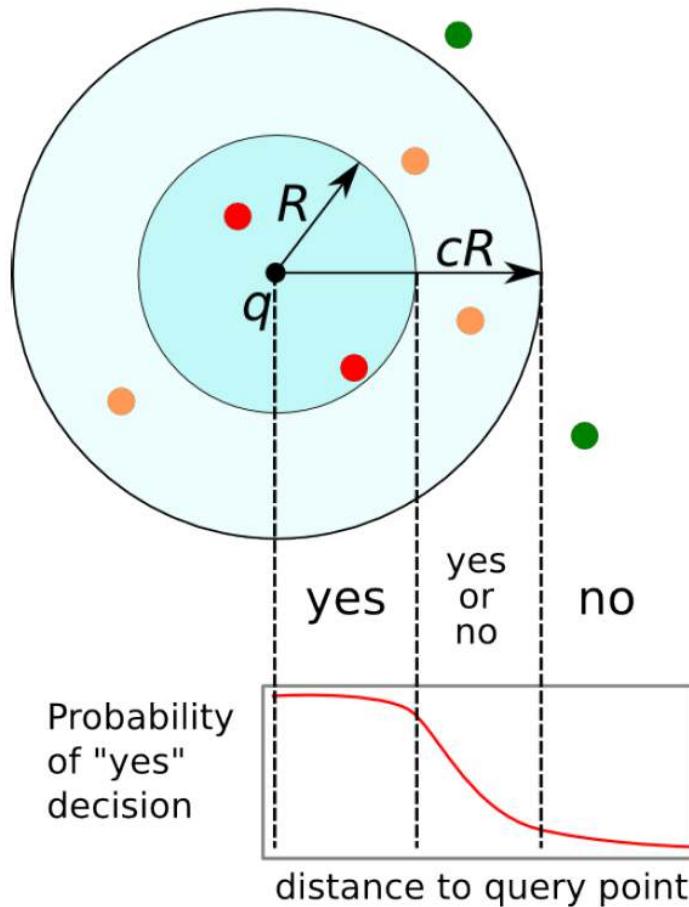
- Solution:



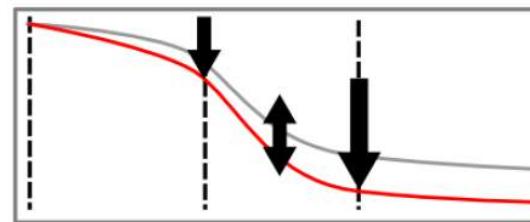
- Solution:



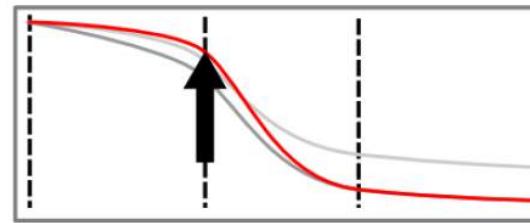
LSH: Error analysis



one Ish function



set of Ish functions
AND



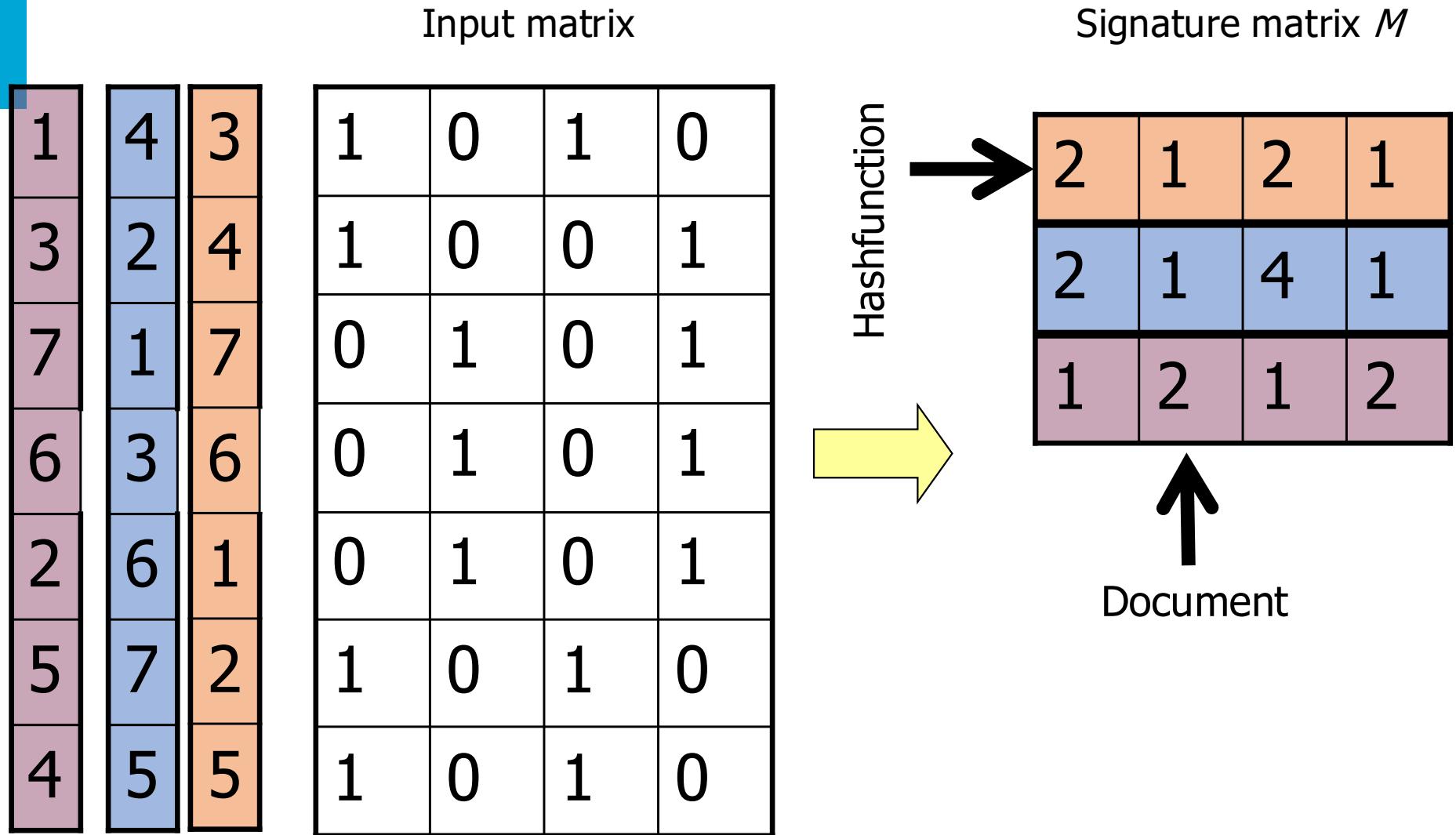
several sets
of Ish functions
OR

LSH in space summary

- Hashes are done using random projections
- Combining projections using AND reduces FP and slightly increases FN
- Combining sets of projections using OR reduces FN and slightly increases FP
- Cascading AND/OR constructions for optimal performance

Locality-sensitive hashing with minhashes

Building signatures



Locality-sensitive hashing

- Locality-sensitive hashing can be used with minhash signatures:
 - Divide signature matrix into b **bands** consisting of r rows each
 - Hash each **sub-signature** of length r into a hash table per band
 - Two sets with **at least one identical sub-signature** will hash in the same bucket (at least once)
→ Candidate column pairs for similarity

Identical sub-signatures:
2 bands - 2 rows

Signature matrix M

	C1	C2	C3	C4
B1	1	2	1	2
B2	2	1	2	1
	2	1	4	1

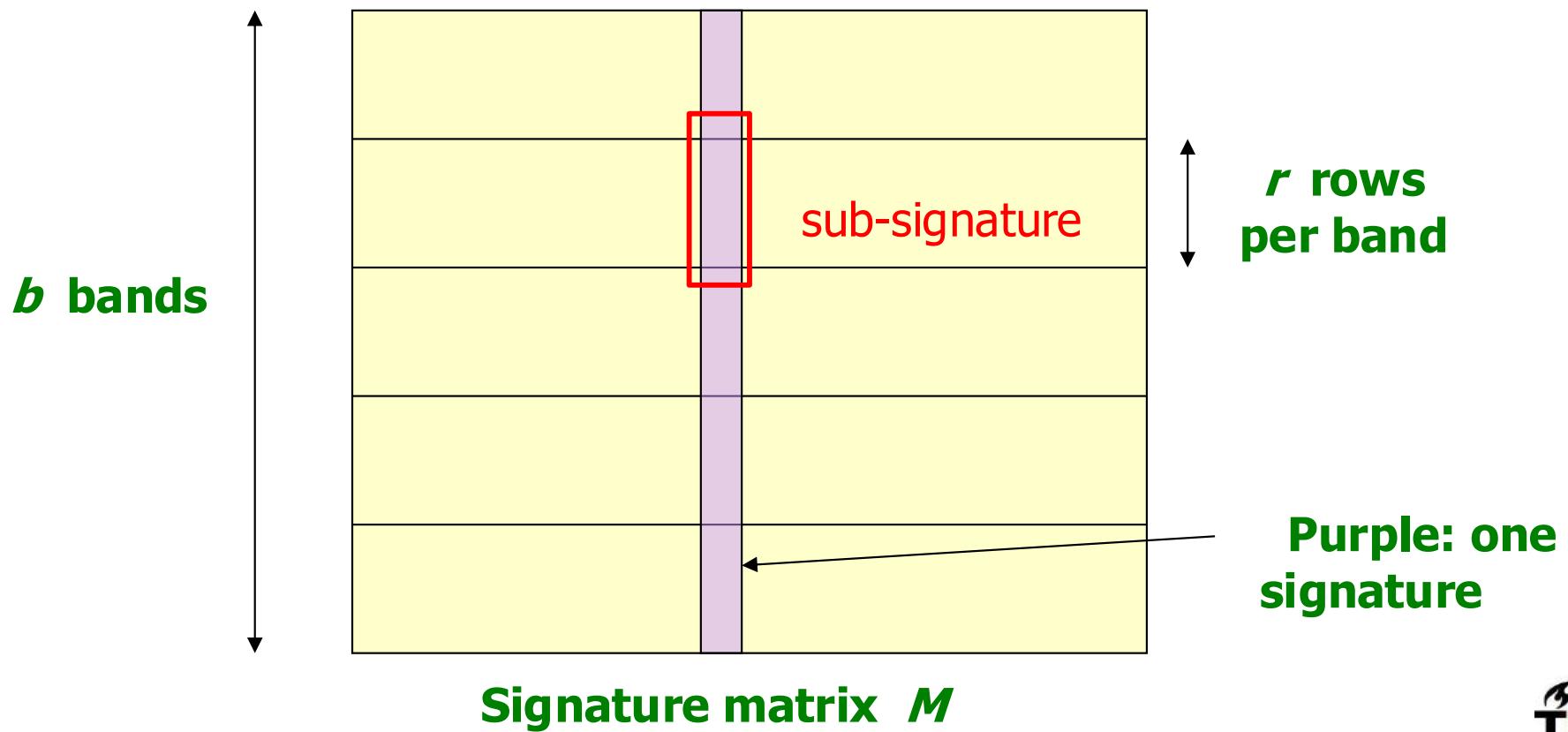
Example: 2 bands - 2 rows

Signature matrix M

	C1	C2	C3	C4	
OR	1	2	1	2	AND
	1	3	1	2	
OR	2	1	2	1	AND
	2	1	4	1	

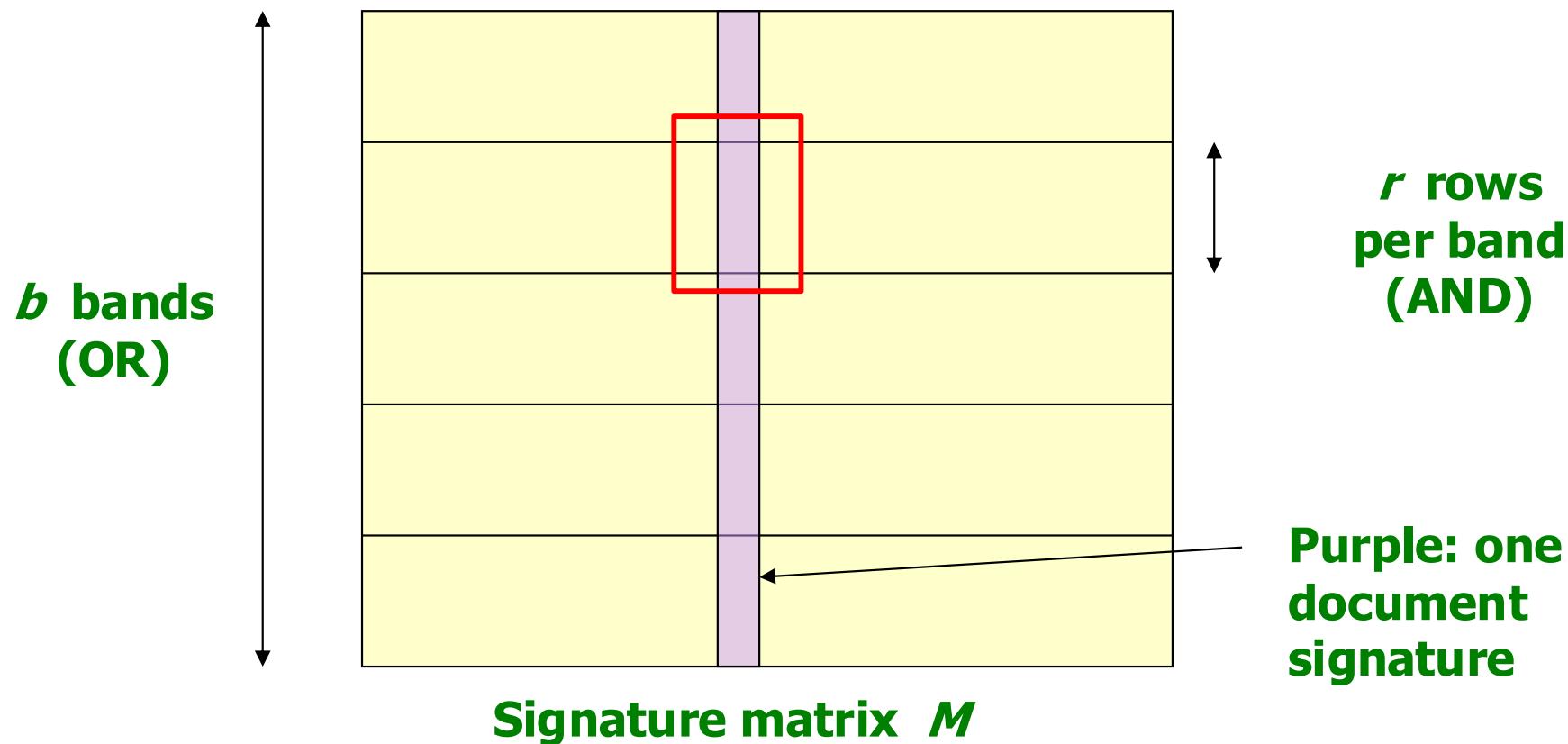
Locality-sensitive hashing

- Two sets with at least one identical sub-signature will hash in the same bucket
 - → Candidate column pairs for similarity



How to choose R and B?

- Locality-sensitive hashing can be used with minhash signatures: Divide signature matrix into b bands consisting of r rows each



Example

- Suppose 100,000 columns.
- Signatures of 100 integers.
- We want all pairs of 80% similar documents.
 - *5,000,000,000 pairs of signatures can take a while to compare...*
- Choose 20 bands of 5 values/band
 - $b=20$; $r=5$.

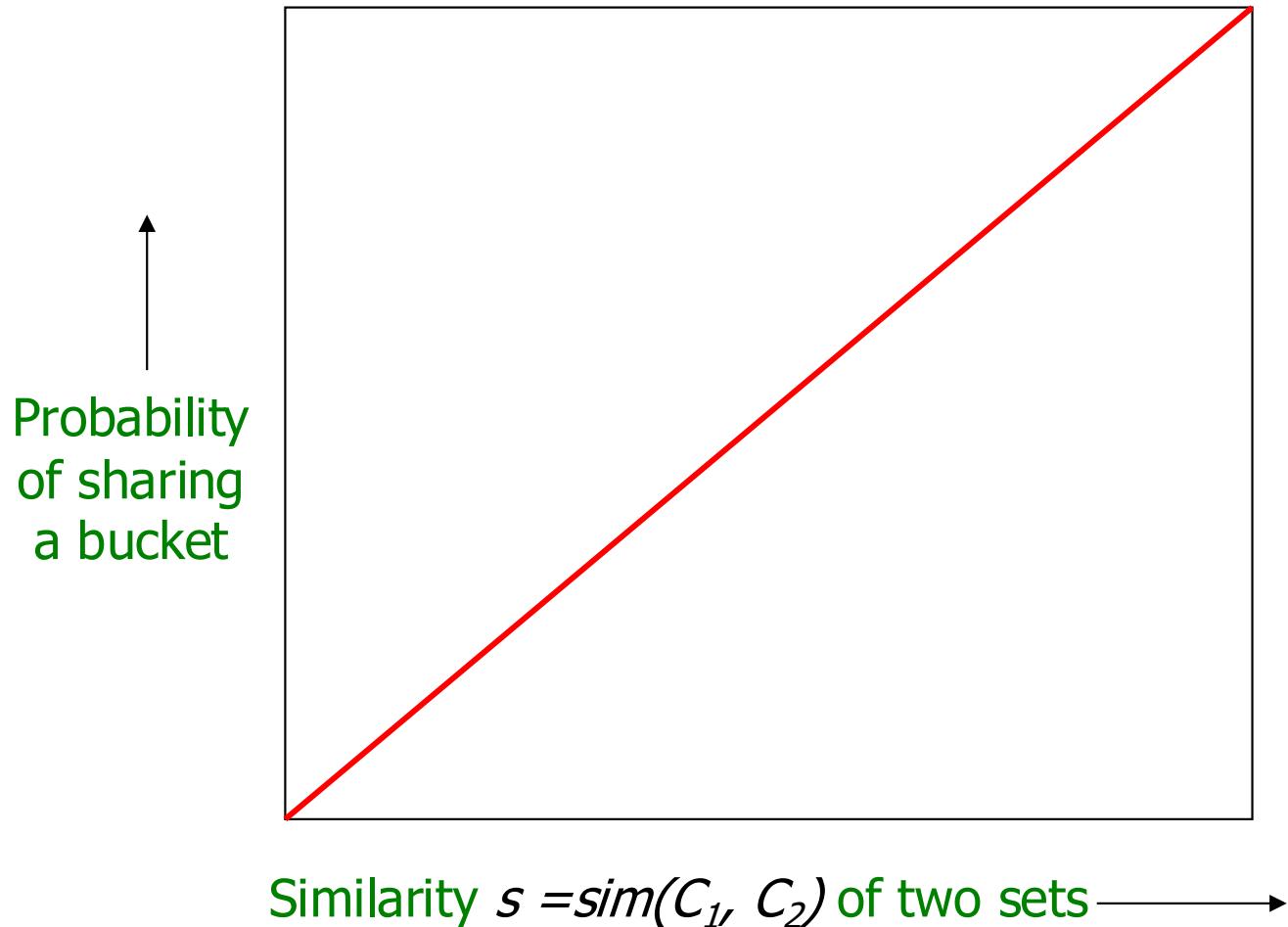
Example: 80% pair

- Suppose C1 and C2 are 80% similar
- Probability C1, C2 identical in one particular band:
 - $(0.8)^5 = 0.328.$
- Probability C1, C2 are *not* similar in any of the 20 bands:
 - $(1-0.328)^{20} = .00035$

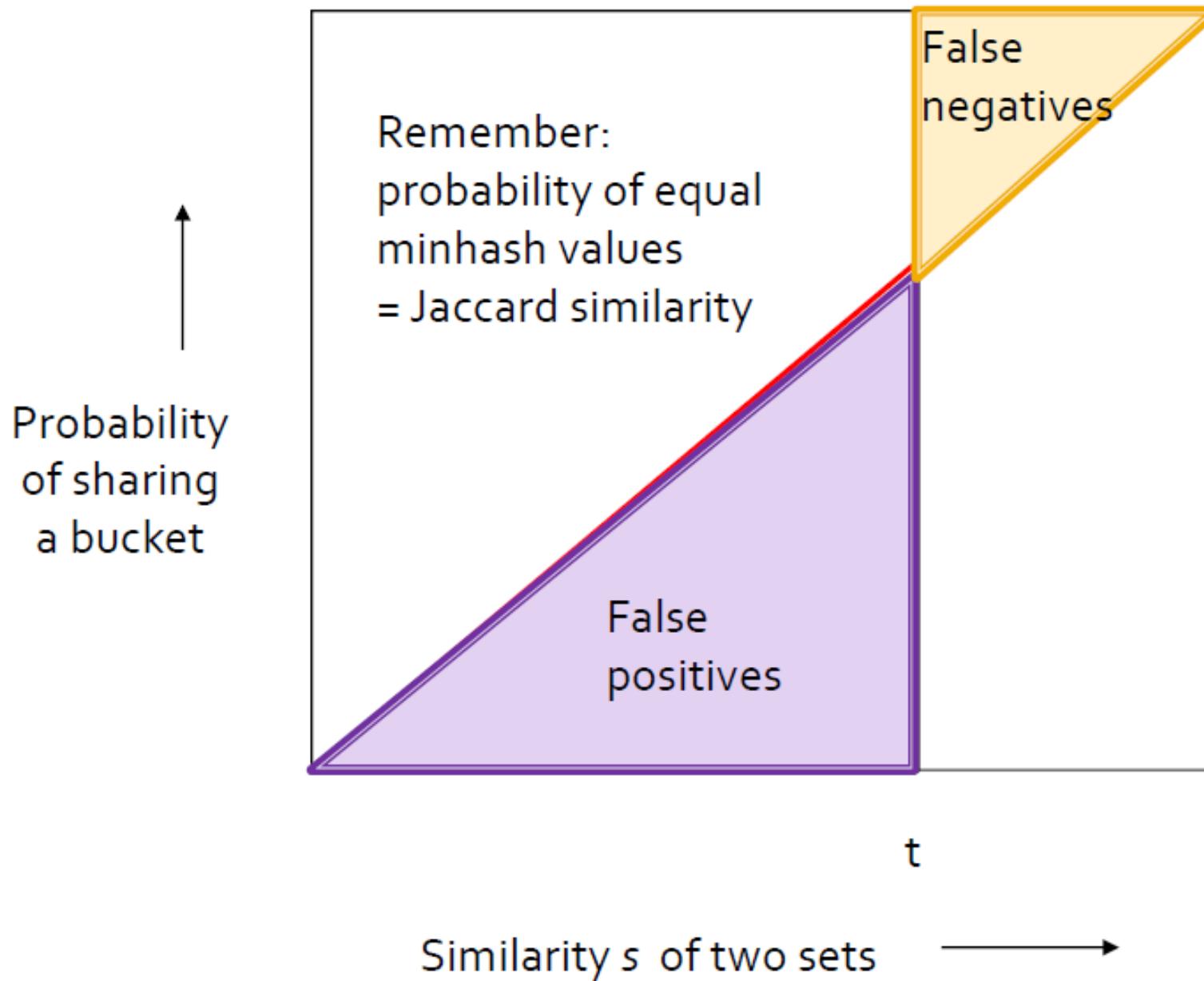
Example: 40% pair

- Suppose C3 and C4 are 40% similar
- Probability C3, C4 identical in one particular band:
 - $(0.4)^5 = 0.01.$
- Probability C3, C4 are *not* identical in any of the 20 bands:
 - $(1-0.01)^{20} = .82.$
- 18% of docs with 40% similarity will become candidates -> false positive

1 row of 1 band



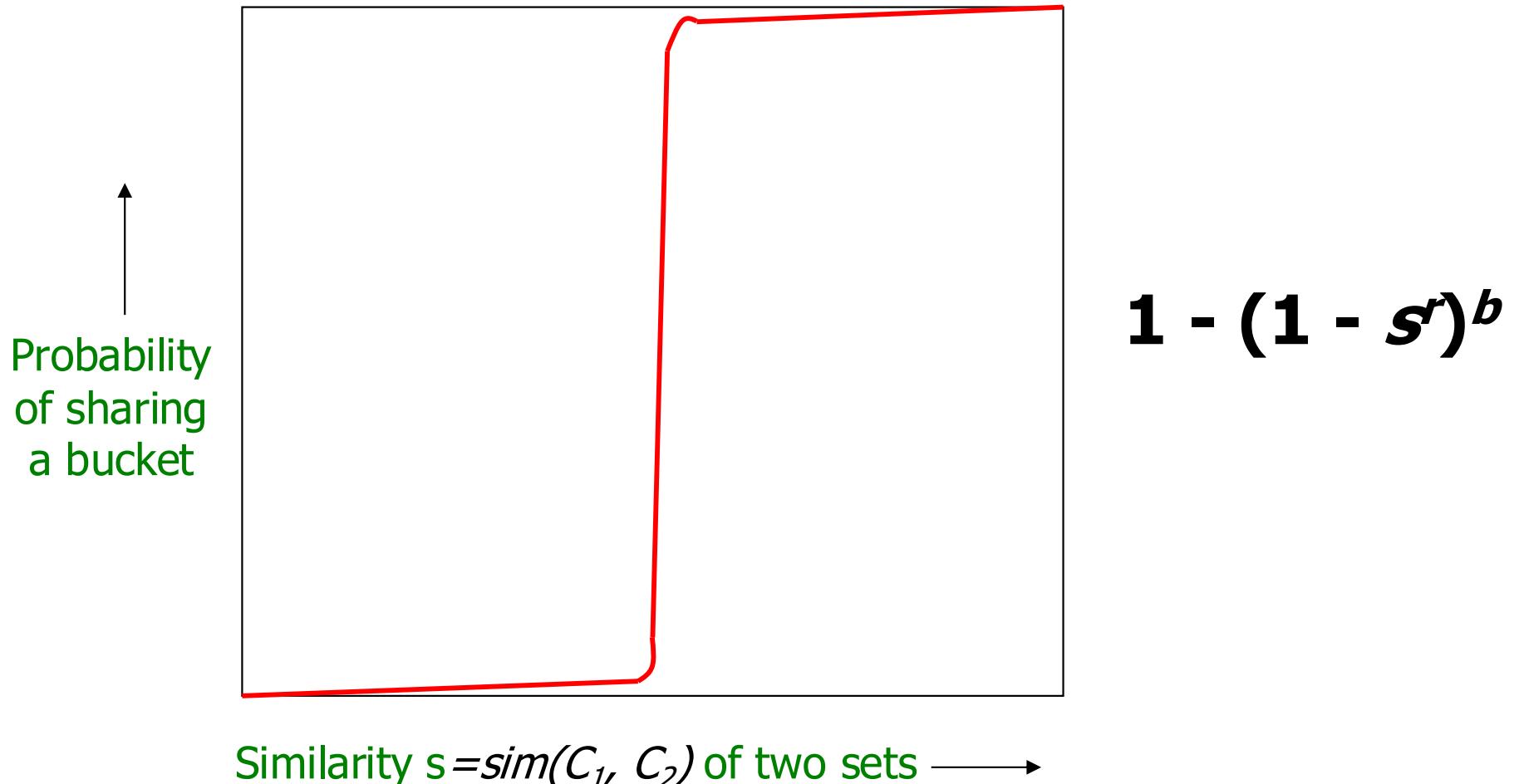
1 row of 1 band



B bands with R rows/band

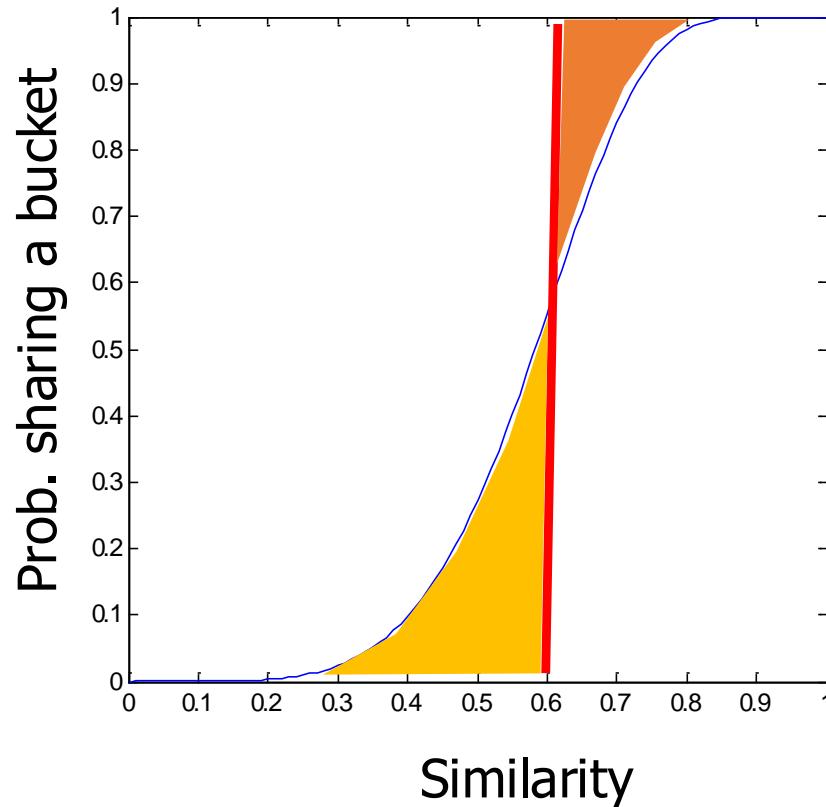
- Columns C_1 and C_2 have similarity s
- Pick any band (r rows)
 - Prob. that all rows in band equal = s^r
 - Prob. that some row in band unequal = $1 - s^r$
- Prob. that no band identical = $(1 - s^r)^b$
- Prob. that at least 1 band identical = $1 - (1 - s^r)^b$

What b bands of r rows Gives You



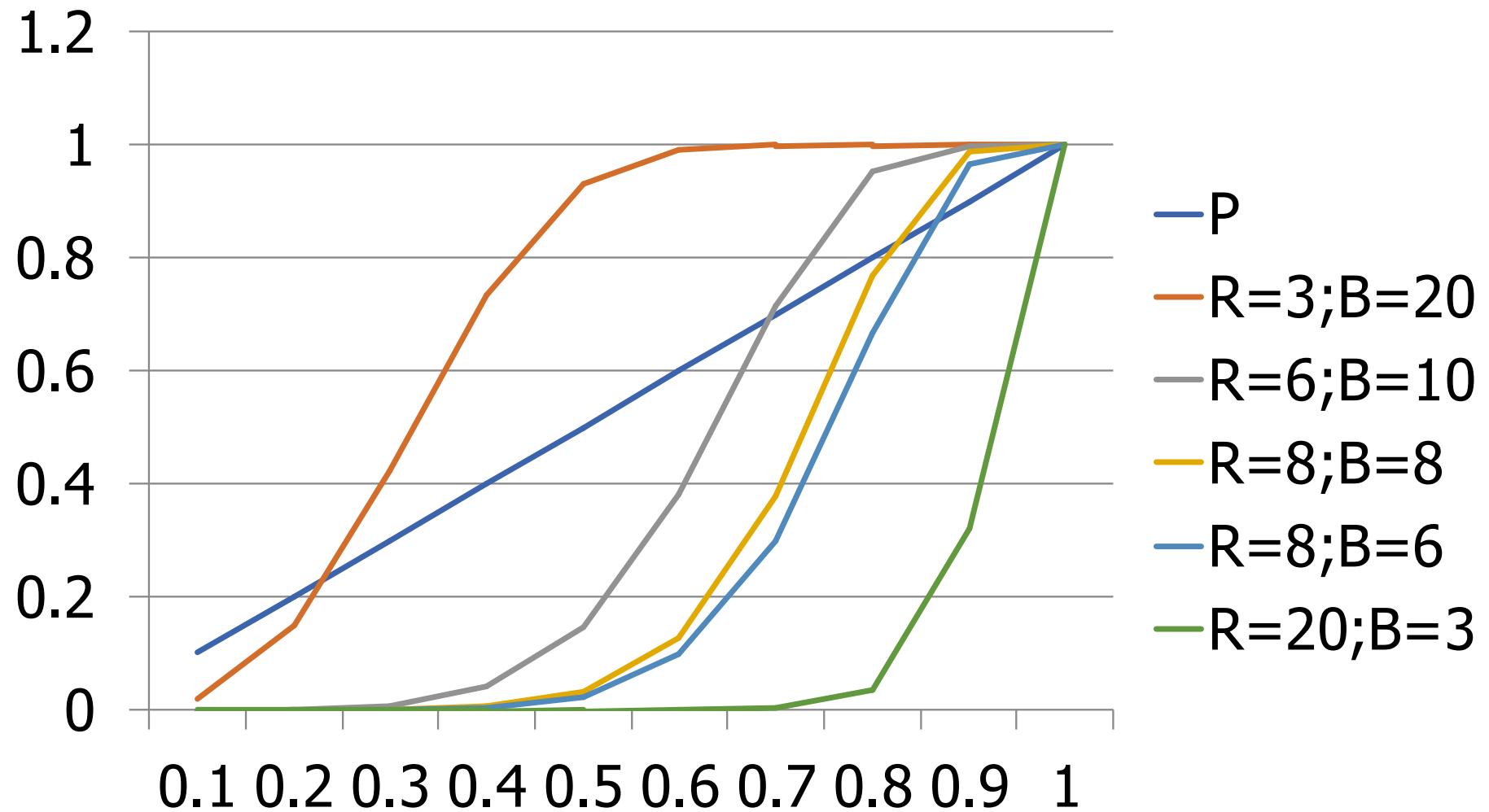
Optimize r and b for best S-curve

- Example: 50 hash-functions ($r=5$, $b=10$)



Red area: False Negative rate
Yellow area: False Positive rate

Tuning R and B for specific similarity



Locality-sensitive hashing

- In practice, the choice of LSH function depends on the **application**:
 - Random projections are frequently used with real-valued points
 - The mining massive datasets book describes a number of other LSH functions for various distances
- Google News uses LSH (with minhash Jaccard distances) for personalization!

Putting it all together

- **Shingling (Ngrams):** Convert documents to sets
 - We used hashing to assign each shingle an ID
- **Min-Hashing:** Convert large sets to short signatures, while preserving similarity
 - We used **similarity preserving hashing** to generate signatures with property $\Pr[h_\pi(C_1) = h_\pi(C_2)] = \text{sim}(C_1, C_2)$
 - We used hashing to get around generating random permutations
 - The result is essentially an embedding that preserves distance
- **Locality-Sensitive Hashing:** Focus on pairs of signatures likely to be from similar documents
 - We used hashing to find **candidate pairs** of similarity $\geq t$
 - Optimize r and b to have steep S-function at similarity

CSE2525 Indexing Text

How to Index and retrieve text ?

Text mining tasks

Text Categorization

- Categorizing Web documents, tweets, news articles, ..

Finding Relevant Information

- Information retrieval tasks, plagiarism detection, finding patent violations, ..

Clustering and Topic detection

- Latent topic finding, document clustering, ...

Indexing text

Why

- Why do we index large collections ?

What

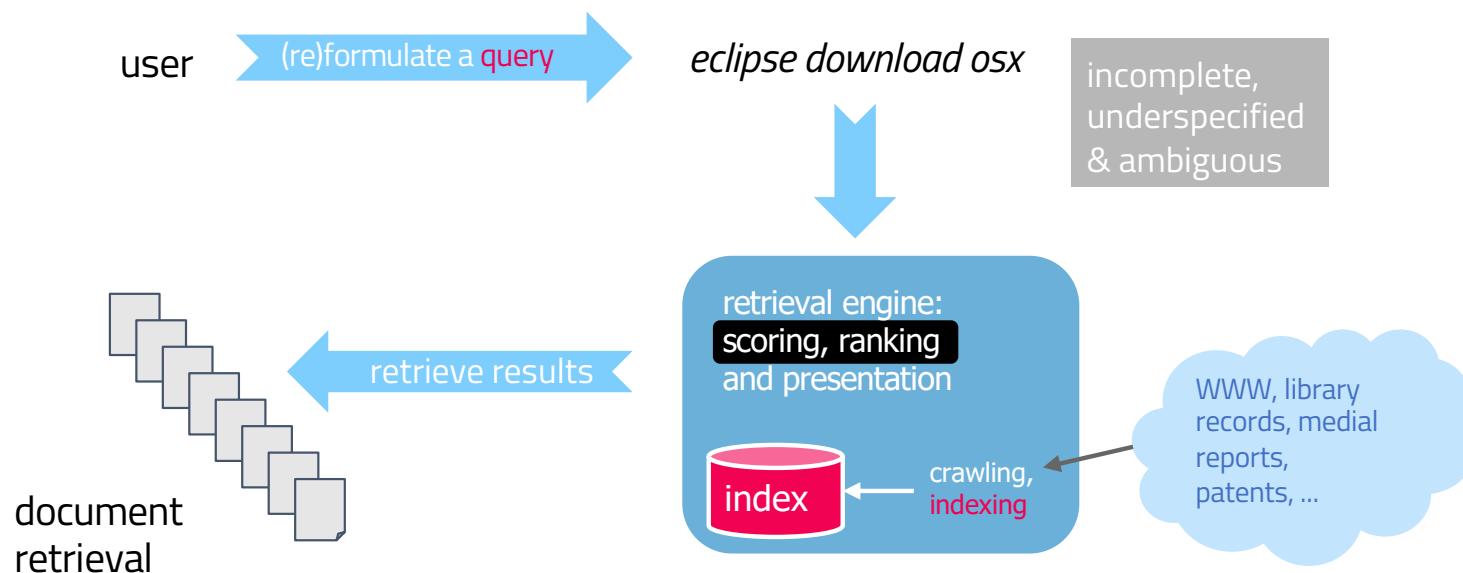
- What data structures are used ?

How

- How do we index and process queries ?

Application: Web Search

Information need: Looks like I need Eclipse for this job. Where can I download the latest beta version for macOS Sierra?



< 100 – 200 ms

Slides credit: C. Hauff

Information need
Topic the user wants to know more about
Query
Translation of need into an input for the search engine
Relevance
A document is relevant if it (partially) provides answers to the information need

Application: Plagiarism Detection

Within academia, plagiarism by students, professors, or researchers is considered academic dishonesty or academic fraud, and offenders are subject to academic censure, up to and including expulsion. Some institutions use plagiarism detection software to uncover potential plagiarism and to deter students from plagiarizing.

Some universities address the issue of academic integrity by providing students with thorough orientations, required writing courses, and clearly articulated honor codes. Indeed, there is a virtually uniform understanding among college students that plagiarism is wrong. Nevertheless, each year students are brought before their institutions' disciplinary boards on charges that they have misused sources in their schoolwork.

<https://en.wikipedia.org/Plagiarism>

Some institutions use plagiarism detection software to uncover potential plagiarism and to deter students from plagiarizing.

Application: Vector Search



Indexing text

Why

- Why do we index large collections ? **Fast similarity computation**

What

- What data structures are used ? **Inverted Index, Forward indexes**

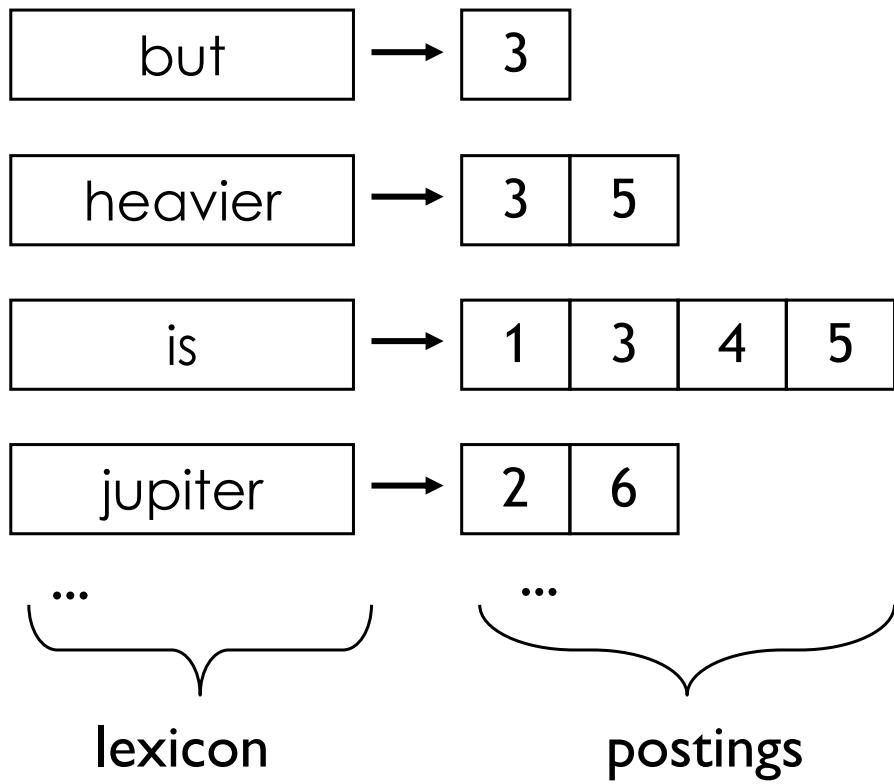
How

- How do we index and process queries ? **Inversion and QP algos**

CSE2525 Indexing Text

Inverted Index

Posting lists

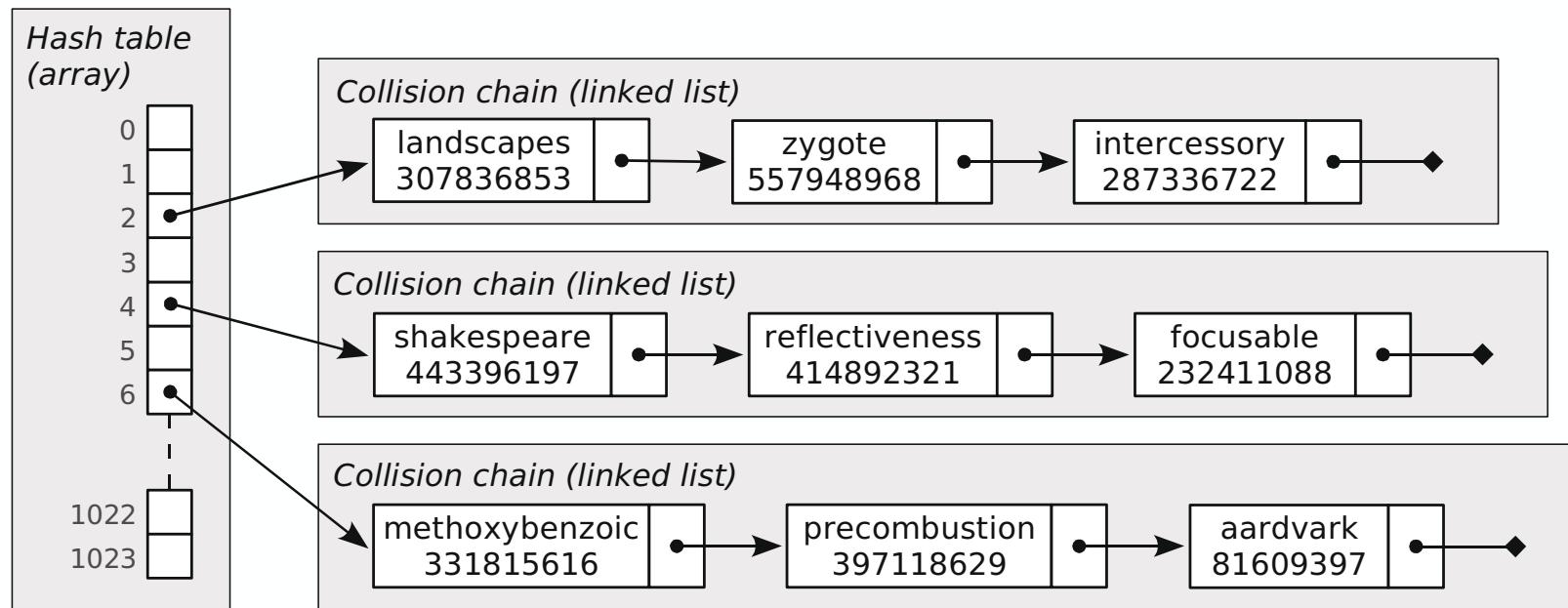


- From term-document matrices to **postings lists**
 - For each **term t**, we must store a list of all documents that contain **t**
 - Identify each doc by a **docID**, a document serial number
 - **Posting:** docID, doc score, positions, ...

Dictionary or Lexicon

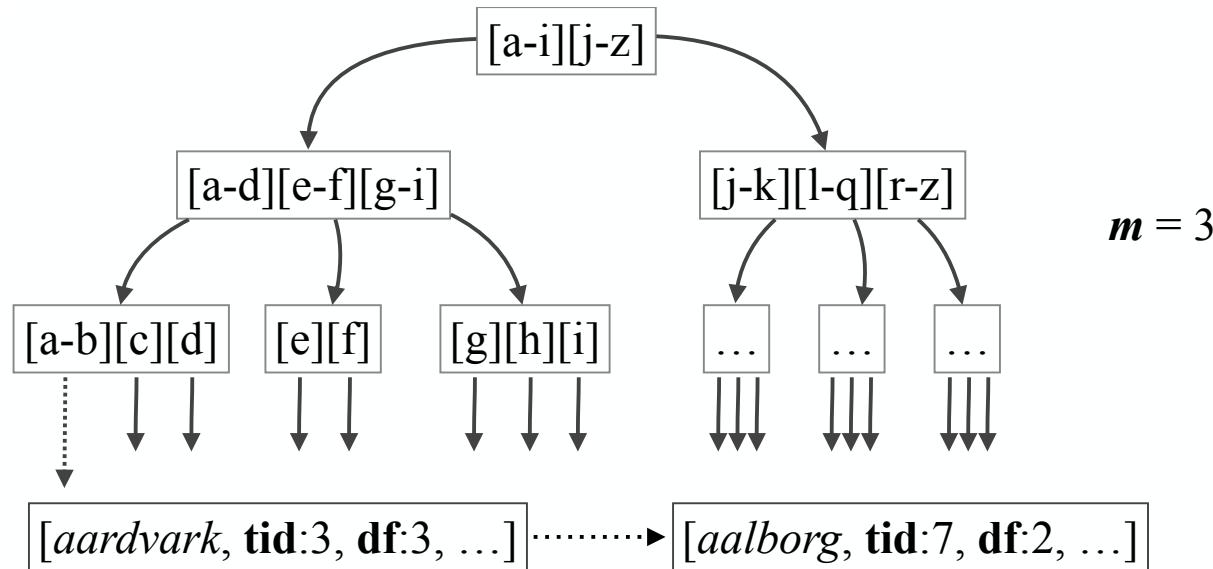
- Maintains **statistics** and **information** about the indexed unit (word, n-gram etc)
`< delft ; location: 82271; tid:12 ; df:23, ... >`
- Value:
 - **Posting list location** - for posting list retrieval
 - **Term identifier** - for term lookups, matching and range queries
 - *document frequency* and associated statistics - for ranking
- Data Structures for Lexicon
 - *Hash-based Lexicon*
 - *B+-Tree based Lexicon*

Hash-Based Lexicon



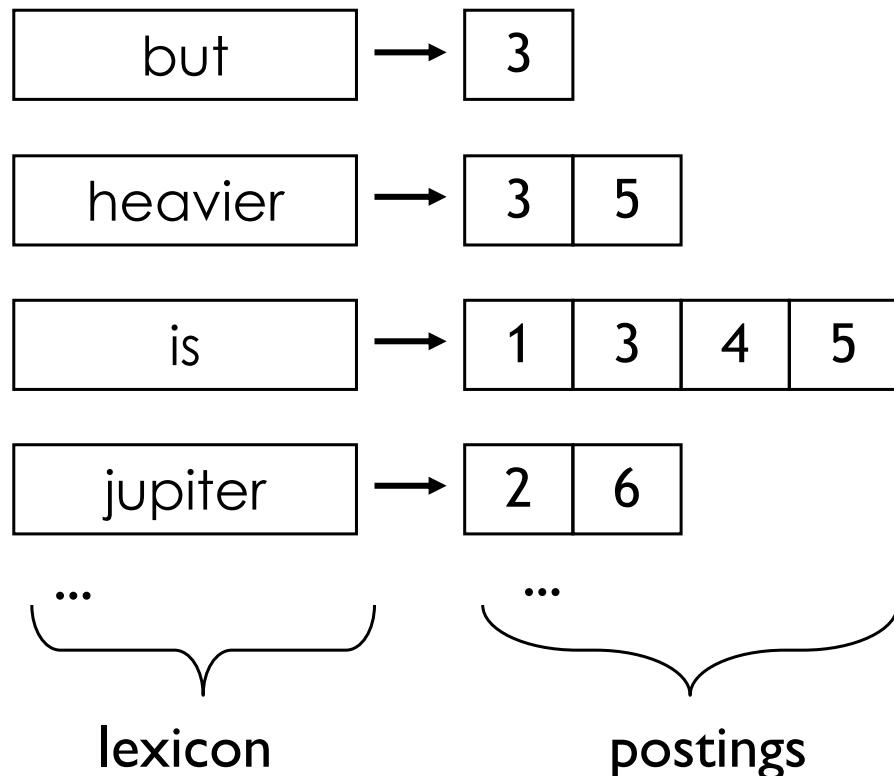
- Constant lookups based on a Hash table
- Entire Lexicon loaded to the memory
- Updates difficult
- Range Searches, Matching, Substring queries not supported

B⁺ Trees



- **B+-Tree:** Leaf nodes additionally linked for efficient range search
- Supports lookups in $O(\log n)$ and range searches in $O(\log n + k)$
- Vocabulary dynamics (i.e., new or removed terms) no problem
- Works on **secondary storage**

Query processing

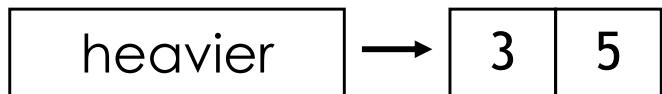
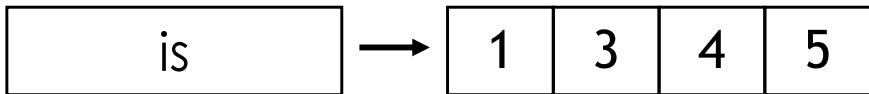


- How do we process queries ?
 - Query: **is heavier**
 - Conjunctive queries:
 - “is” AND “heavier”
 - Disjunctive queries
 - “is” OR “heavier”

Note: Posting lists sorted by doc. id

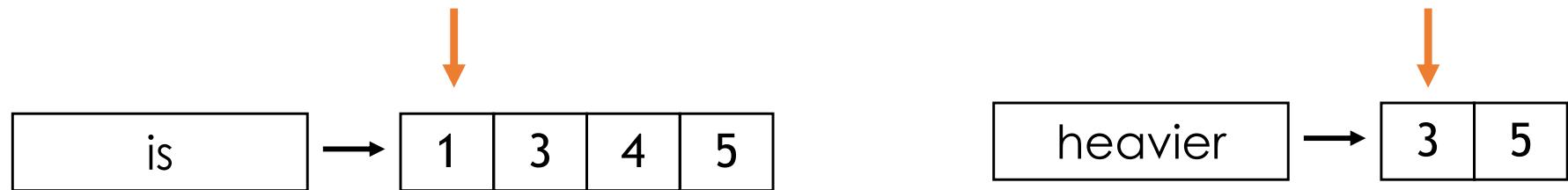
Query Processing: Term-at-a-time

• Query: **is heavier**



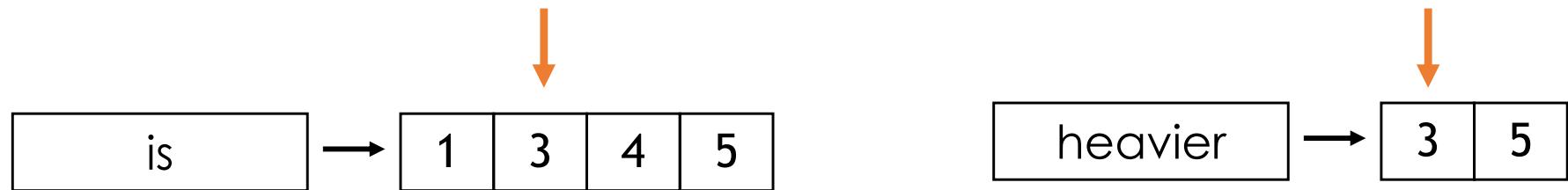
- Process one term list at a time
- Maintain **accumulators** in memory for partial results and update them
- Best for disjunctive queries

Query Processing: Doc-at-a-time



- Open pointers to all lists at the same time

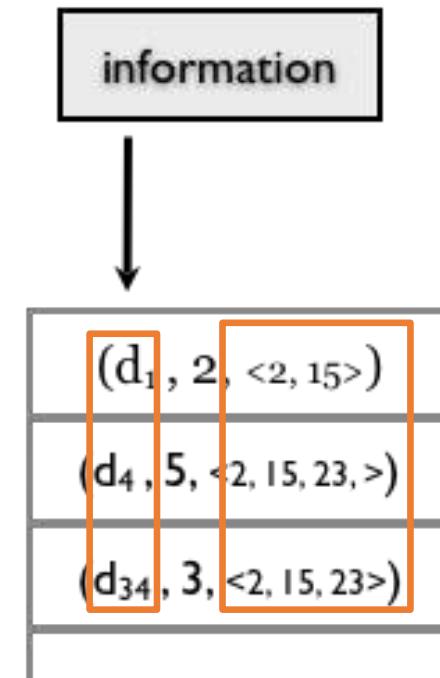
Query Processing: Doc-at-a-time



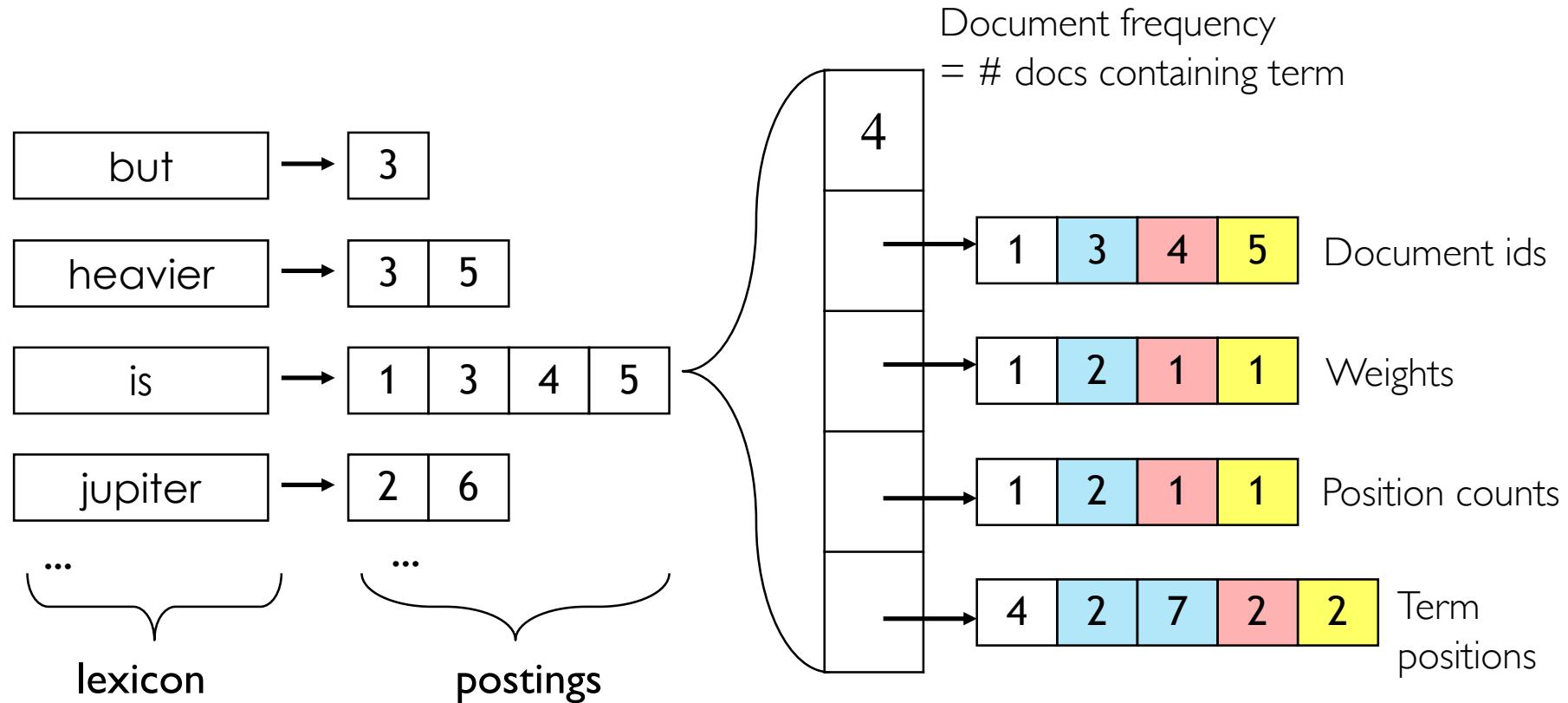
- Open pointers to all lists at the same time
- Systematically move all pointers to find the intersection
 - If all pointers point to the same doc-id → report as a result and move all pointers by one step
 - Find the max doc-id
 - Move all the pointers until max doc id

Logical Layout

- Posting lists contain
 - Doc ids
 - Tf-idf or scores
 - Positions
 - ...
- Logical layout as a sequence of postings

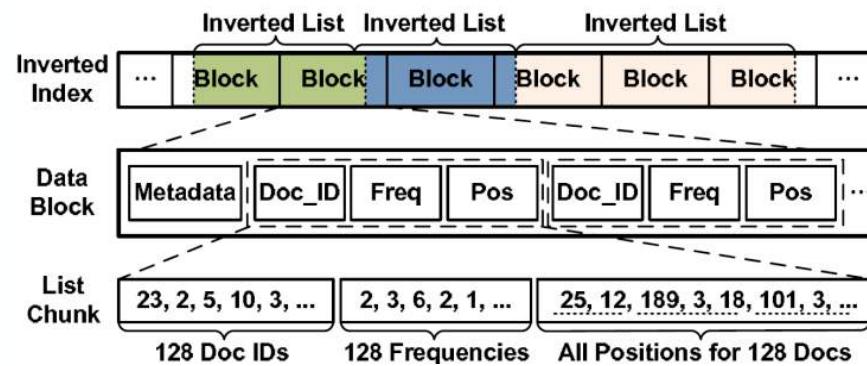


Physical layout



Data Organization

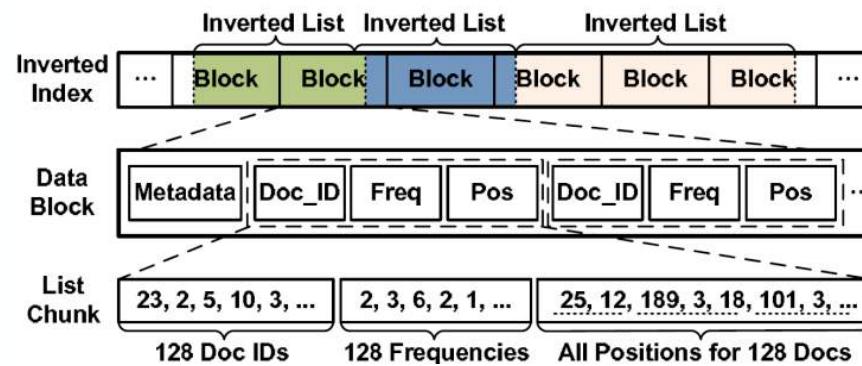
Data blocks, say
of size 64KB, as
basic unit for list
caching



- Posting list storage is implemented as
 - list of document identifiers (did)
 - list of scores, positions

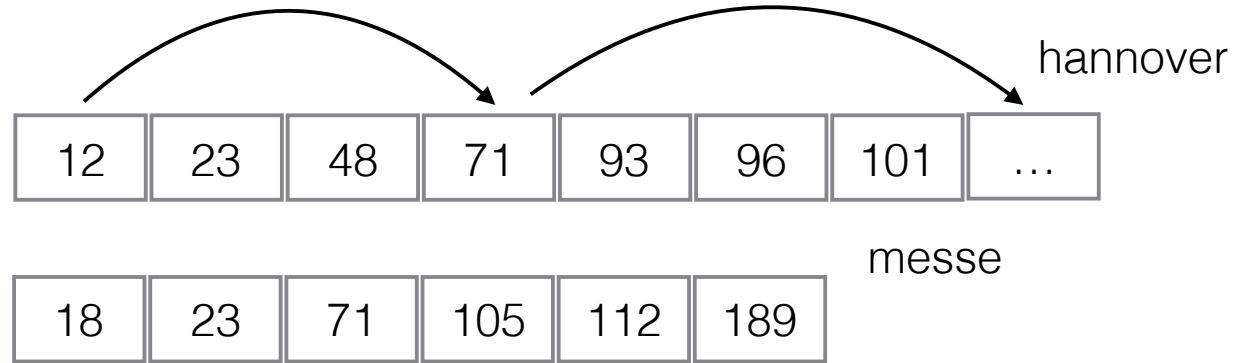
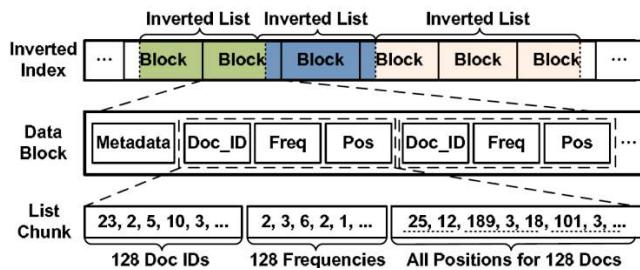
Data Organization

Data blocks, say
of size 64KB, as
basic unit for list
caching



- Posting list storage is implemented as
 - list of document identifiers (did)
 - list of scores, positions
- Many chunks are skipped over, but very few blocks are
- Also, may prefetch the next, say 2MB of index data from disk

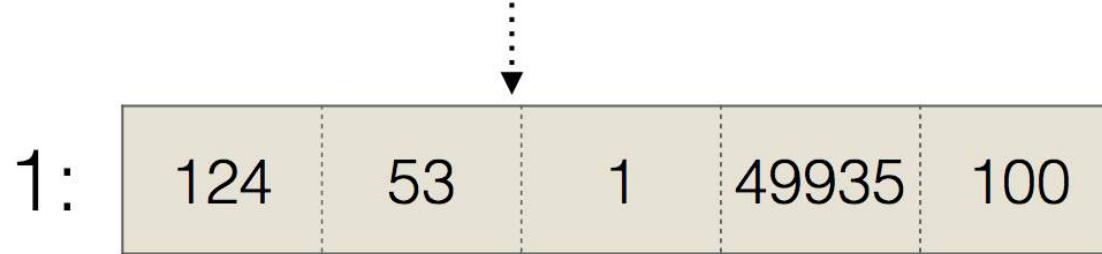
Skip Lists



- Skip list allow fast intersections acting as a secondary index over posting lists
- Typically skip over fixed number of postings and are square root of the length of the postings list

Forward Index

1: “*what does the fox say ?*”



- Mapping of doc-ids to term-ids
- Maintain same order
- Efficient retrieval of terms from (already parsed) text
- **snippet generation**
- **proximity features** for proximity-aware ranking
- per-doc term distribution for query expansions

Inverted Index Construction

- We are given a set of documents D, where each document d is considered as a bag of terms
- Inverted Lists are created by a process termed as **Inversion**
- *Memory-based Inversion*
 - Takes place entirely in-memory
 - For small collections, where the index + lexicon fits in memory
- *Disk-based Inversion*
 - Sort-based inversion vs Merge-based inversion

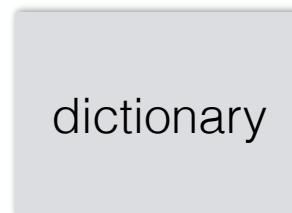
Memory-based Inversion

1: “what does the fox say ?” [term, positions] [the, <3>] [fox, <4>]

2: “the fox jumped over the fence” [the, <1,5>] [fox, <2>]

doc: [term, positions]

1:[the, <1,5>]



[term, posting list]

[1, <3>]

[2, <1,5>]

- Dictionary required for efficient single-term lookup and insertion
- An extensible (i.e., dynamic) list data structure needed to store the postings

Sort-based Inversion

- Input Collection D $>>$ memory size M
- Inversion can be seen as a sort operation on the term identifiers
- This method is based on *external sort* over data which does not fit into the memory
 - Read data of size M into memory, sort them and write back to disk
 - Multiway merge of D/M sorted lists to create index
- **Shortcomings**
 - Dictionary might not fit in-memory
 - Large memory requirements due to intermediate data

Exercise

- **Simple Computational Model**
 - Total number of postings = N
 - Number of postings which fit in memory = M
 - Cost of disk read/write of a posting = c
- What is the estimated cost of sort-based Inversion in terms of N, M and c ?
- How does the cost compare with in-memory sort-based inversion (assuming we had enough memory or $N > M$) ?

Merge-based Inversion



- Reads input collection to create an in-memory index of size **M** and write it to disk to create partial indexes with local lexicons
- Compression in posting lists in partial indexes
- Multi-way Merge of corresponding lists from the partial indexes to create one consolidated index
- Useful for index updates – when there is new data

Map-Reduce

- Programming paradigm for distributed data processing
- Improves overall throughput by parallelizing loading of data
- Data is partitioned into the nodes which process the data in the following phases
 - *Map*: Generates (key, value) pairs
 - *Shuffle*: Shuffles the pairs over the network to the reducers
 - *Reduce*: Operates on all values for the same key
- How would you build the inverted index using Map-reduce ?
 - What are the key-value pairs as defined by the Mapper ?
 - What does the reducer do with the values of the same key ?

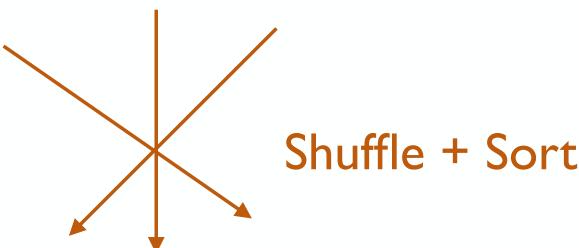
Map-Reduce

1: “what does the fox say ?”

Mapper - 1

what : 1
does : 1
the : 1
fox : 1
say : 1

mappers emit <word, freq>



2: “the fox jumped over the fence”

Mapper - 2

jumped : 1
over : 1
the : 2
fox : 1
fence : 1

Reducer - 1

reducers aggr. freq.

does : 1	fence : 1	jumped : 1	fox : 1	+	over : 1	the : 1	say : 1	what : 1
				+			+	
			fox : 1			the : 2		

Reducer - 2

References

- Query Processing and Boolean retrieval --
<https://nlp.stanford.edu/IR-book/pdf/01bool.pdf>
- Vocabulary and Postings list -- <https://nlp.stanford.edu/IR-book/pdf/02voc.pdf>
- Index Construction -- <https://nlp.stanford.edu/IR-book/pdf/04const.pdf>

Questions

Example

Row	C1	C2
1	0	0
2	1	1
3	1	1
4	1	0
5	0	1
6	0	1

Sig1 Sig2

$$h(1) =$$
$$g(1) =$$

$$h(x) = 3x \bmod 5$$
$$g(x) = 4x+1 \bmod 5$$

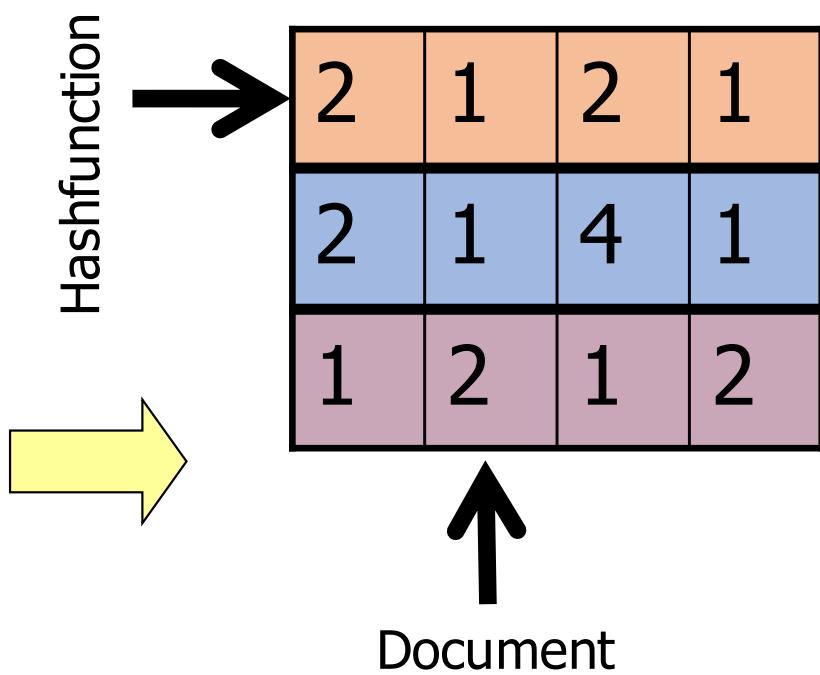
Locality-sensitive hashing with minhashes

Building signatures

Input matrix

1	4	3
3	2	4
7	1	7
6	3	6
2	6	1
5	7	2
4	5	5

Signature matrix M



Locality-sensitive hashing

- Locality-sensitive hashing can be used with minhash signatures:
 - Divide signature matrix into b **bands** consisting of r rows each
 - Hash each **sub-signature** of length r into a hash table per band
 - Two sets with **at least one identical sub-signature** will hash in the same bucket (at least once)
→ Candidate column pairs for similarity

Identical sub-signatures:
2 bands - 2 rows

Signature matrix M

	C1	C2	C3	C4
B1	1	2	1	2
B2	2	1	2	1
	2	1	4	1

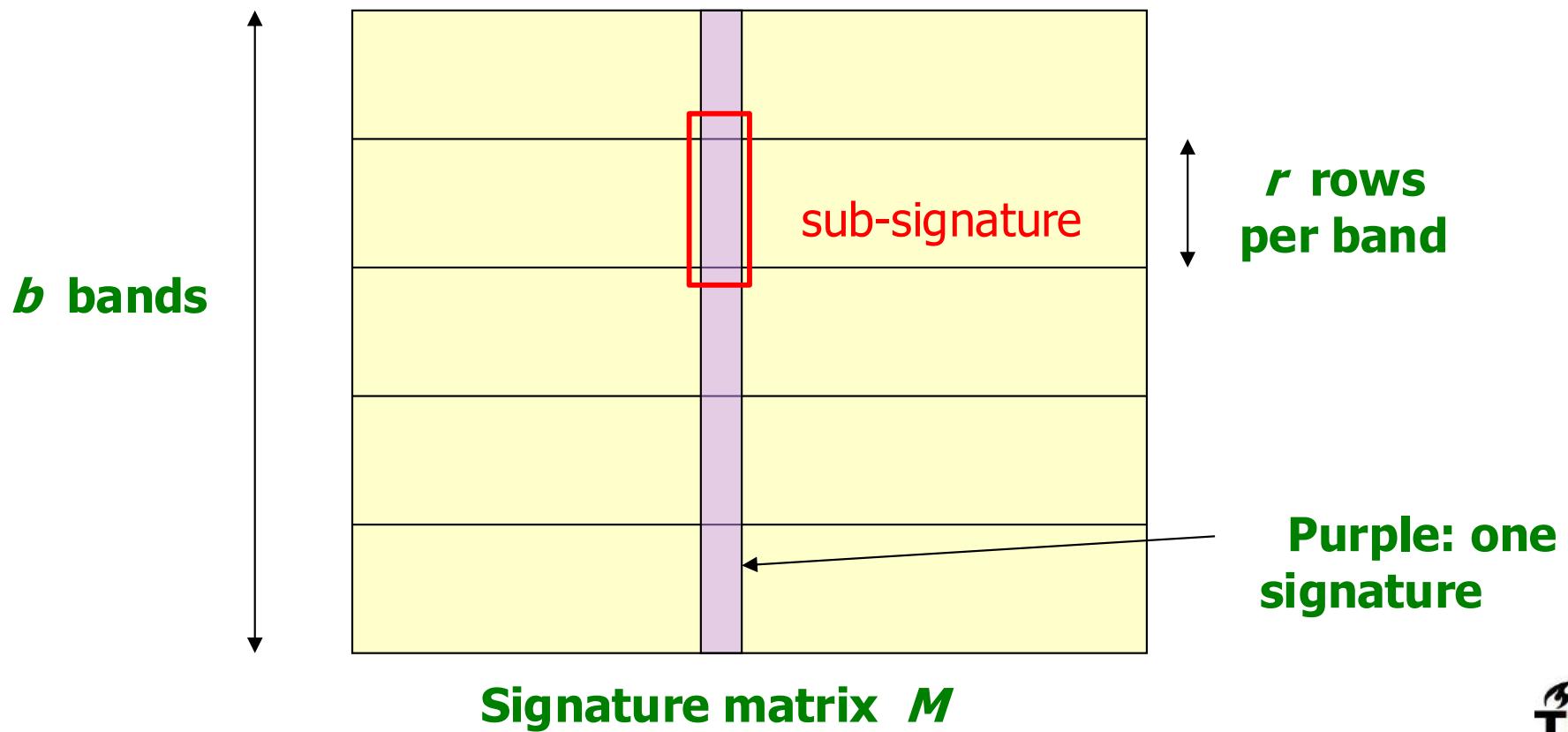
Example: 2 bands - 2 rows

Signature matrix M

	C1	C2	C3	C4	
OR	1	2	1	2	AND
	1	3	1	2	
OR	2	1	2	1	AND
	2	1	4	1	

Locality-sensitive hashing

- Two sets with at least one identical sub-signature will hash in the same bucket
 - → Candidate column pairs for similarity



B bands with R rows/band

- Columns C_1 and C_2 have similarity s
- Pick any band (r rows)
 - Prob. that all rows in band equal = s^r
 - Prob. that some row in band unequal = $1 - s^r$
- Prob. that no band identical = $(1 - s^r)^b$
- Prob. that at least 1 band identical = $1 - (1 - s^r)^b$

Example

- Suppose 100,000 columns.
- Signatures of 100 integers.
- We want all pairs of 80% similar documents.
 - *5,000,000,000 pairs of signatures can take a while to compare...*
- Choose 20 bands of 5 values/band
 - $b=20$; $r=5$.

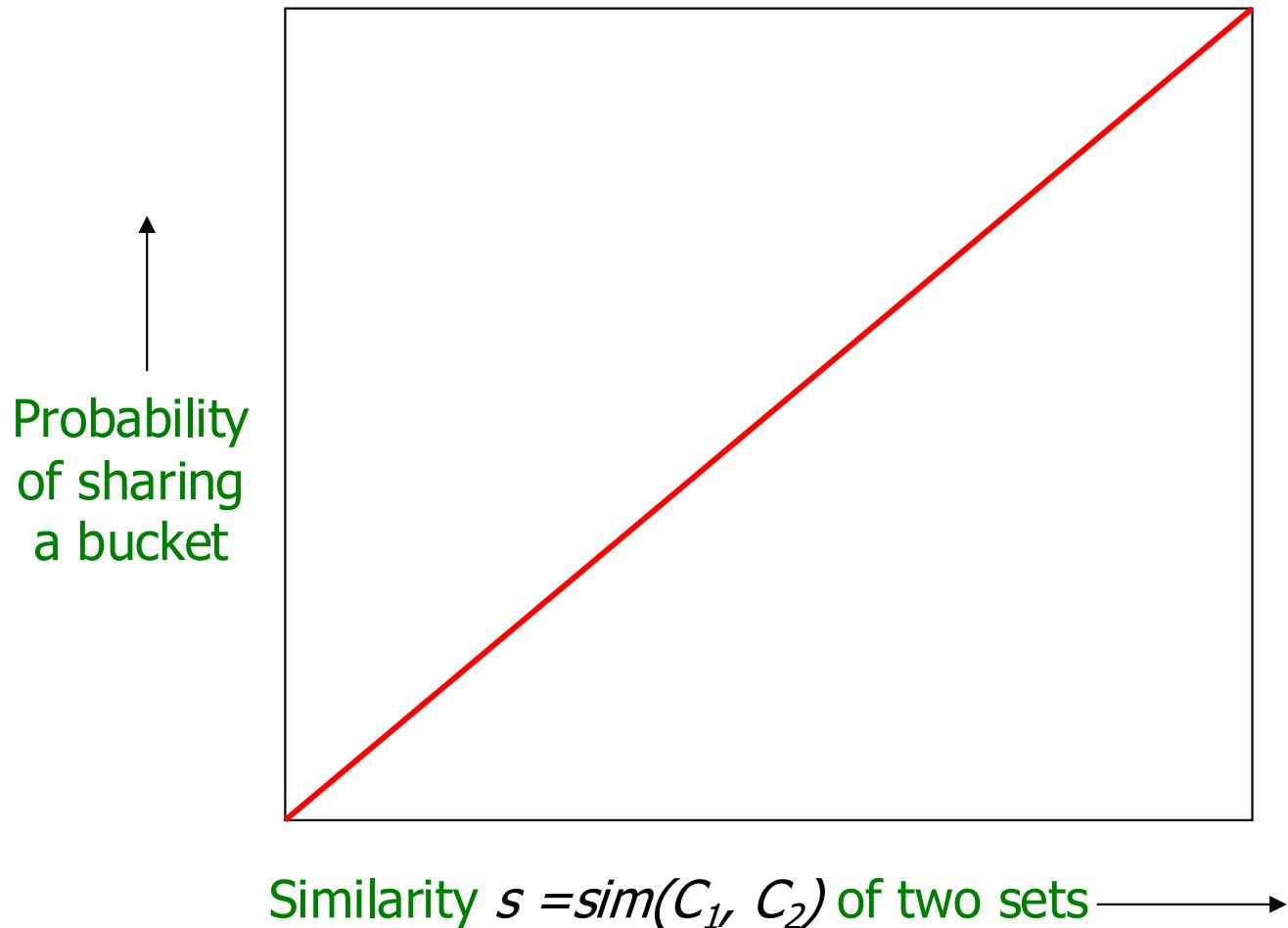
Example: 80% pair

- Suppose C1 and C2 are 80% similar
- Probability C1, C2 identical in one particular band:
 - $(0.8)^5 = 0.328.$
- Probability C1, C2 are *not* similar in any of the 20 bands:
 - $(1-0.328)^{20} = .00035$

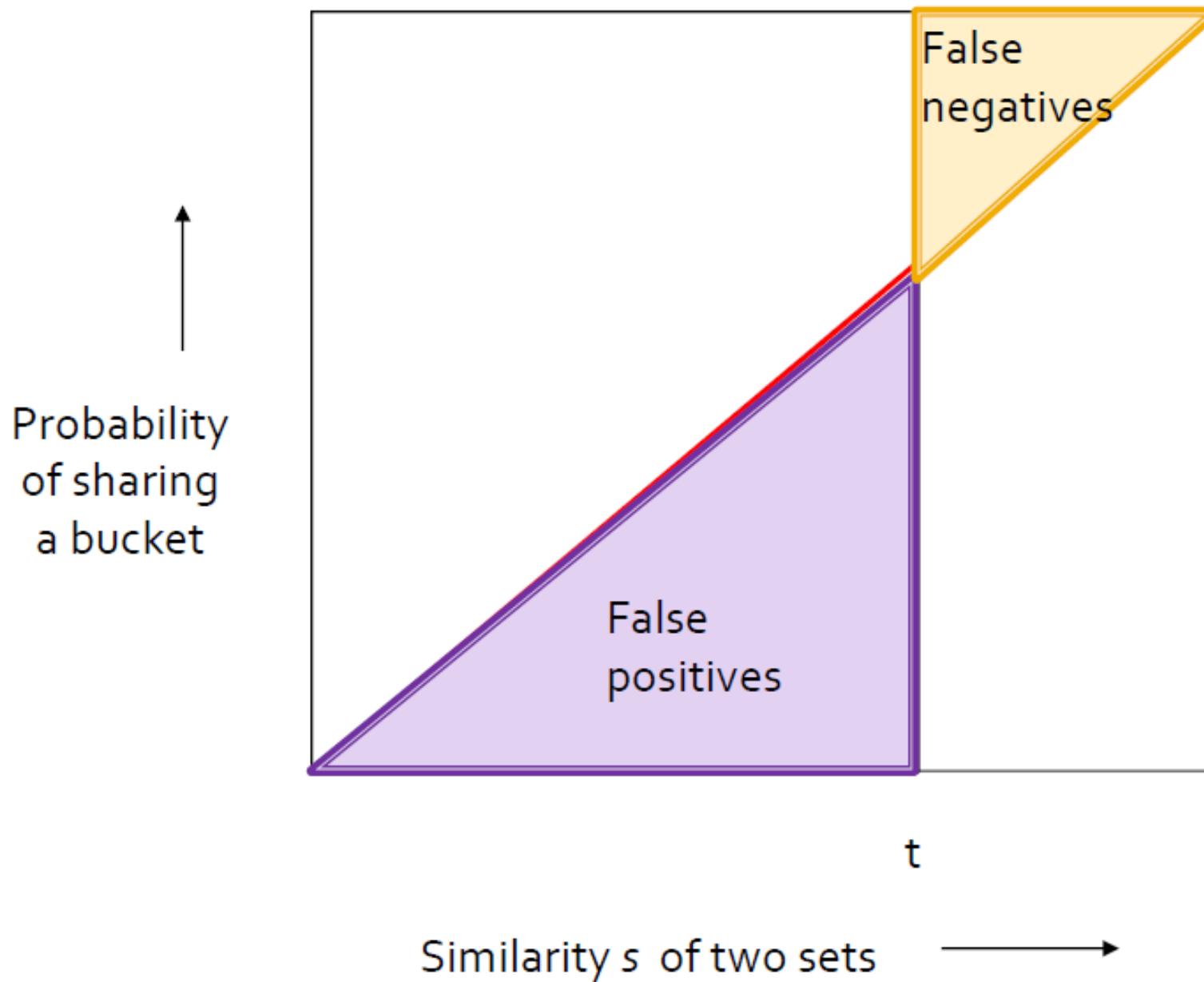
Example: 40% pair

- Suppose C3 and C4 are 40% similar
- Probability C3, C4 identical in one particular band:
 - $(0.4)^5 = 0.01.$
- Probability C3, C4 are *not* identical in any of the 20 bands:
 - $(1-0.01)^{20} = .82.$
- 18% of docs with 40% similarity will become candidates -> false positive

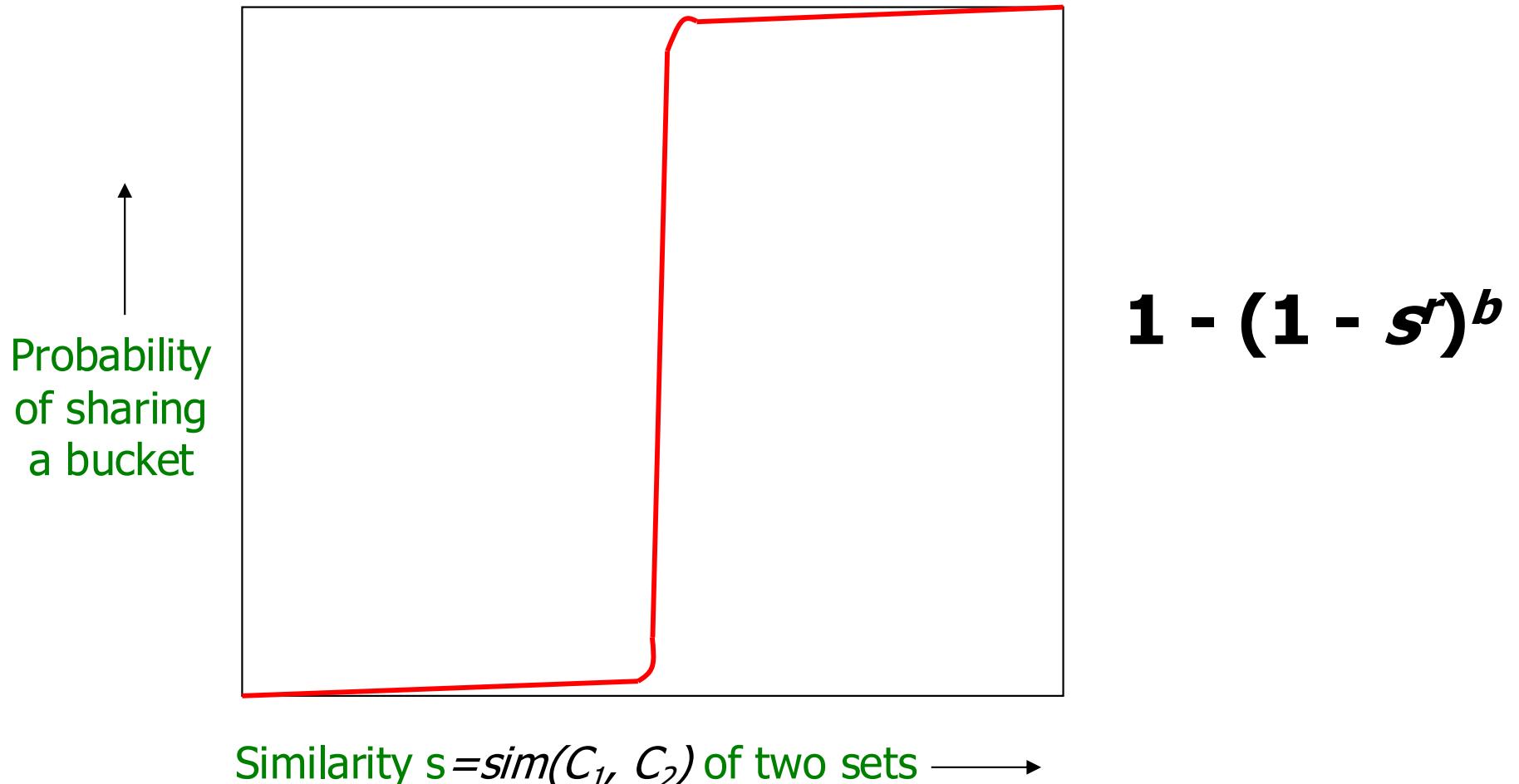
1 row of 1 band



1 row of 1 band

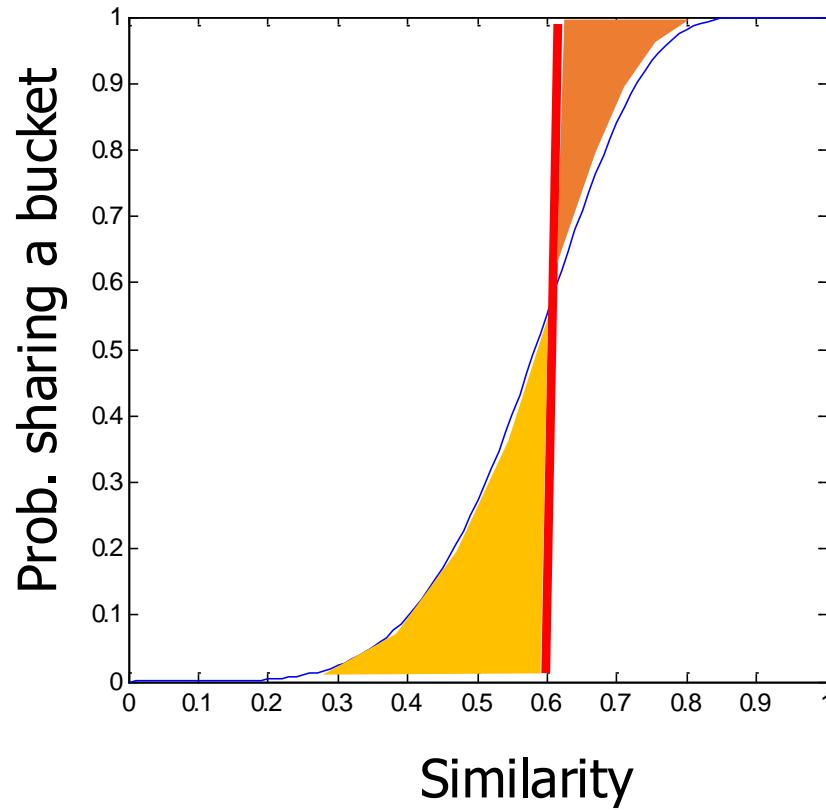


What b bands of r rows Gives You



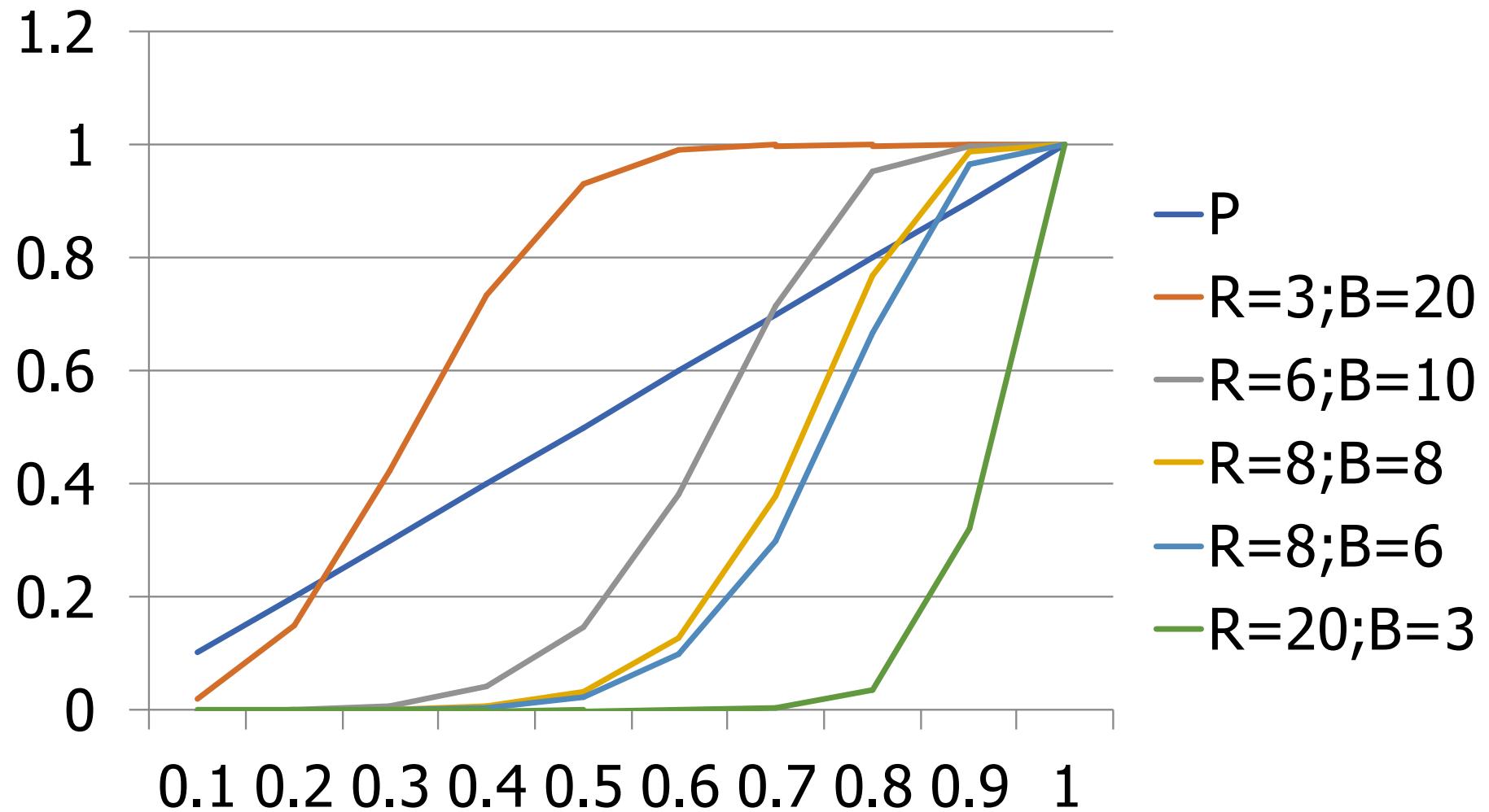
Optimize r and b for best S-curve

- Example: 50 hash-functions ($r=5$, $b=10$)



Red area: False Negative rate
Yellow area: False Positive rate

Tuning R and B for specific similarity



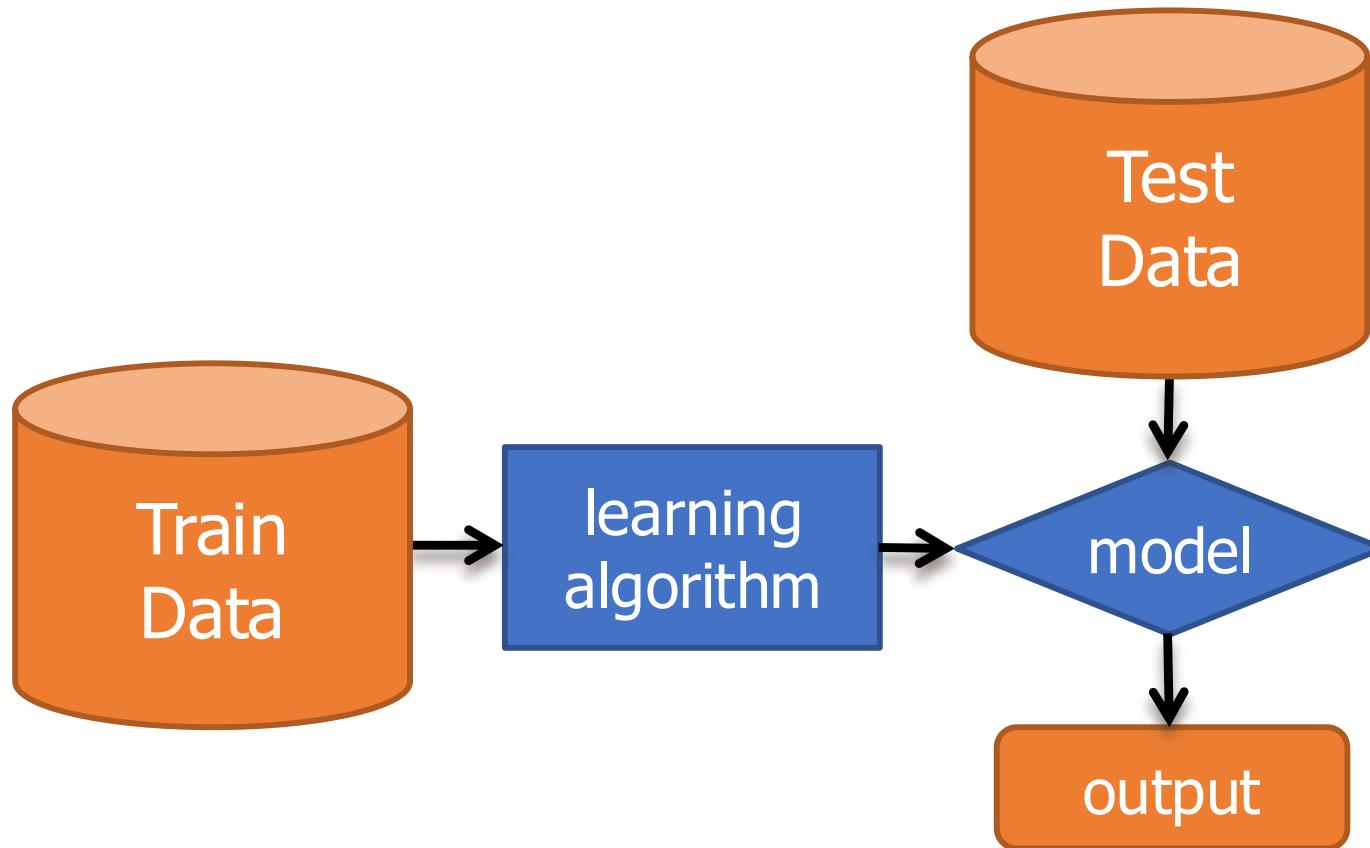
Locality-sensitive hashing

- In practice, the choice of LSH function depends on the **application**:
 - Random projections are frequently used with real-valued points
 - The mining massive datasets book describes a number of other LSH functions for various distances
- Google News uses LSH (with minhash Jaccard distances) for personalization!

Putting it all together

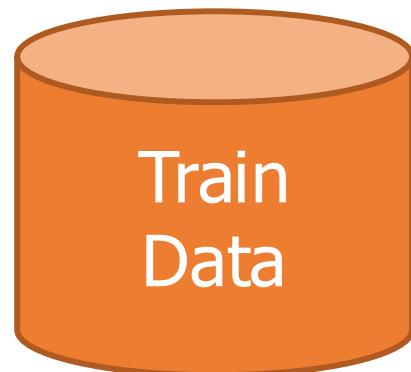
- **Shingling (Ngrams):** Convert documents to sets
 - We used hashing to assign each shingle an ID
- **Min-Hashing:** Convert large sets to short signatures, while preserving similarity
 - We used **similarity preserving hashing** to generate signatures with property $\Pr[h_\pi(C_1) = h_\pi(C_2)] = \text{sim}(C_1, C_2)$
 - We used hashing to get around generating random permutations
 - The result is essentially an embedding that preserves distance
- **Locality-Sensitive Hashing:** Focus on pairs of signatures likely to be from similar documents
 - We used hashing to find **candidate pairs** of similarity $\geq t$
 - Optimize r and b to have steep S-function at similarity

CS2525 – Counting

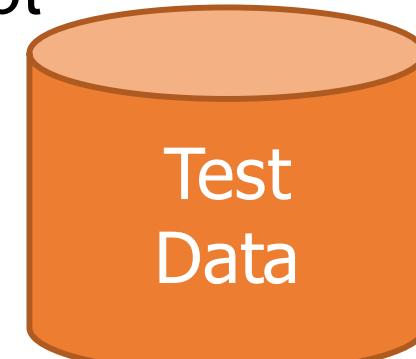


What can we modify?

We can adapt
in any way we
want



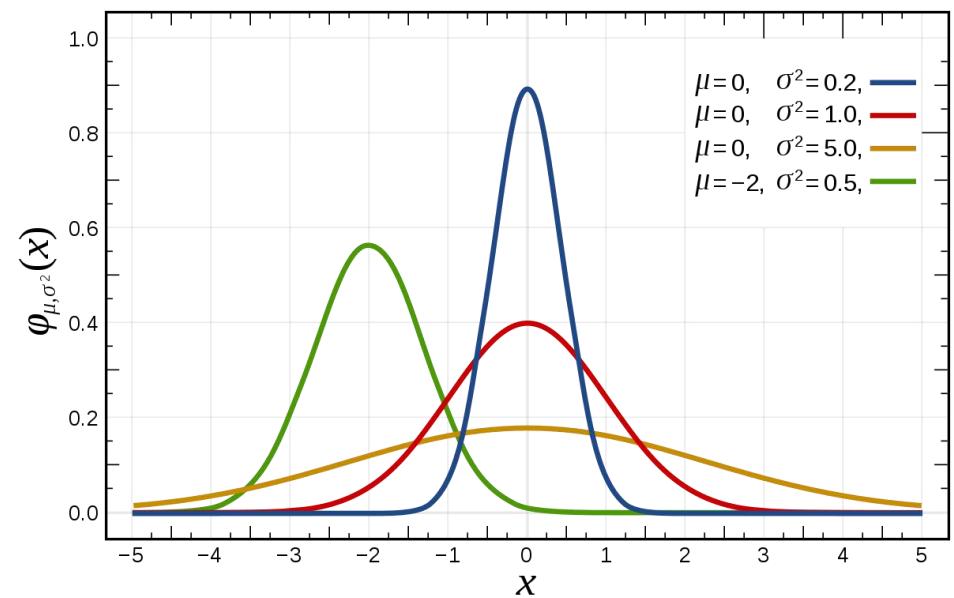
We can adapt
but never
ever use the
class label!



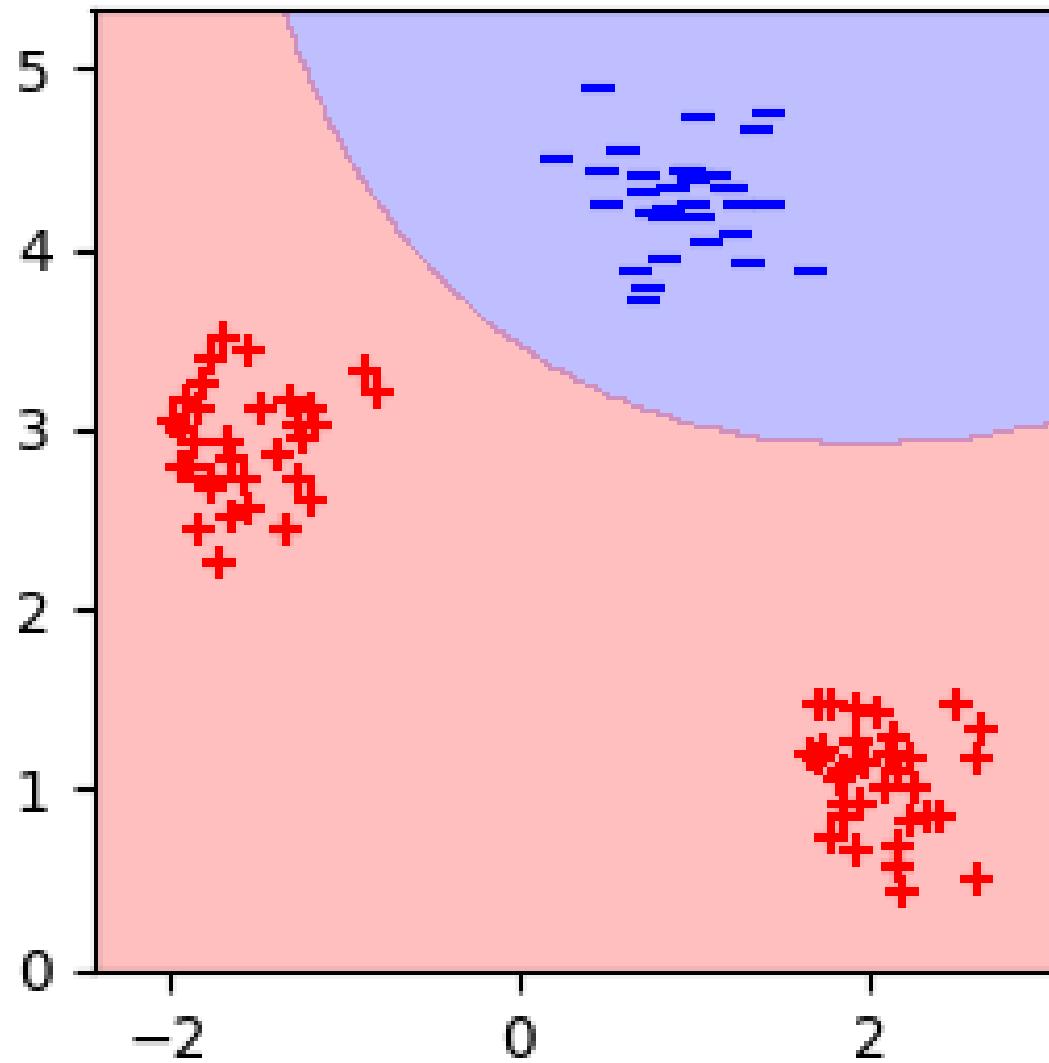
What can we modify?

Gaussian distributed

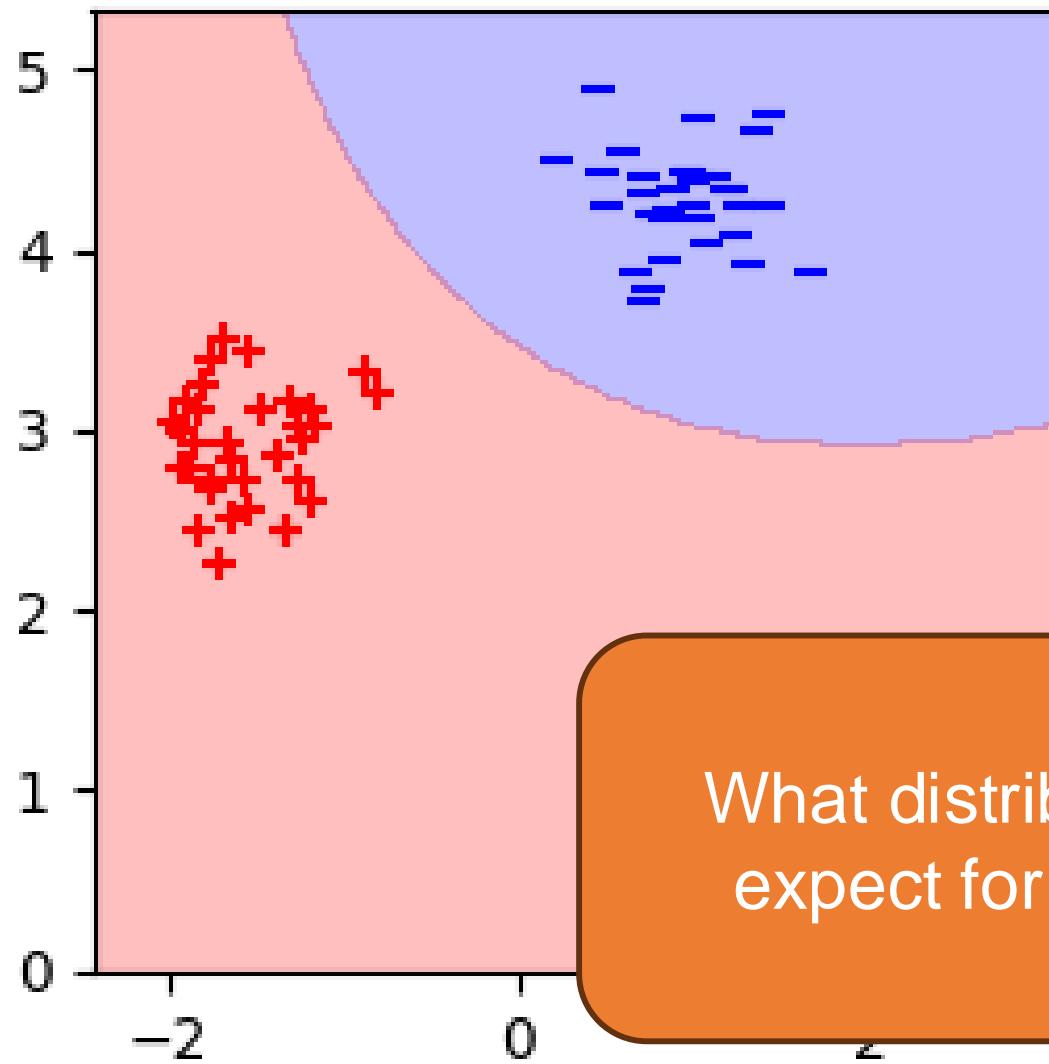
- Much data is Gaussian distributed
- CLT: the normalized sum of any independent random data tends towards a Gaussian distribution
- Several ML model assume Gaussian distributed data (LDA, QDA, ...), so can we run them on non-normal data?



Wrong models can still be useful!

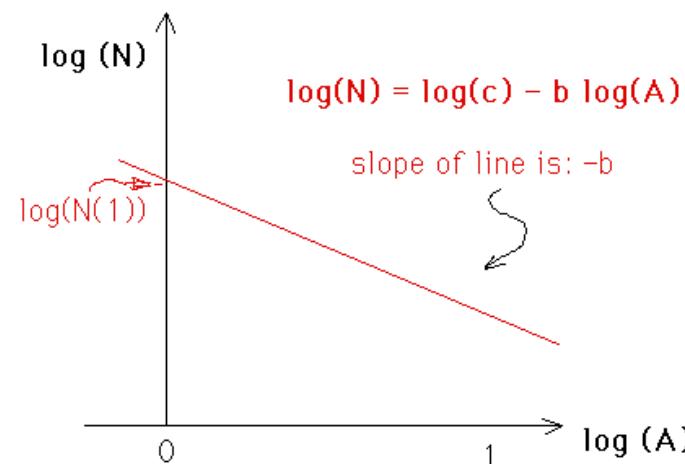
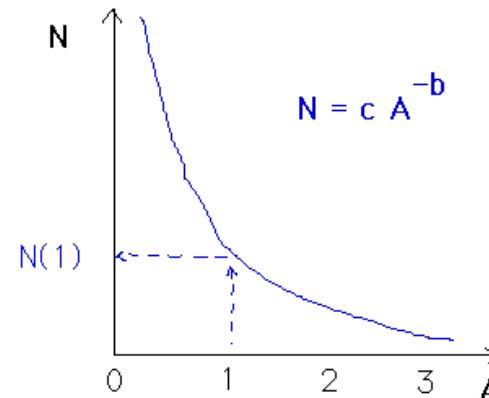


Wrong models can still be useful!

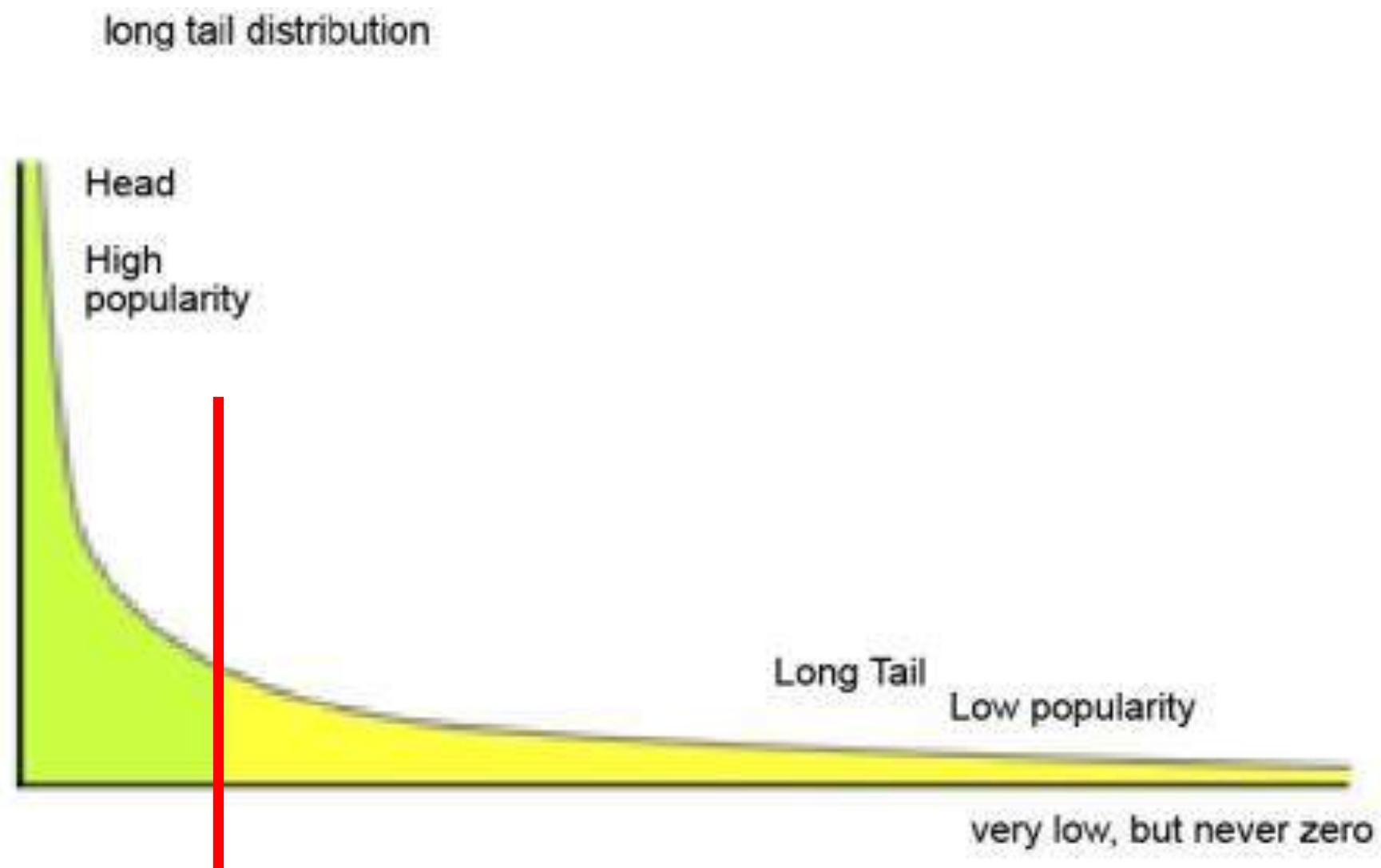


Power laws

- Count data often follows power laws
- A power law is a linear relationship between the logarithms of two variables

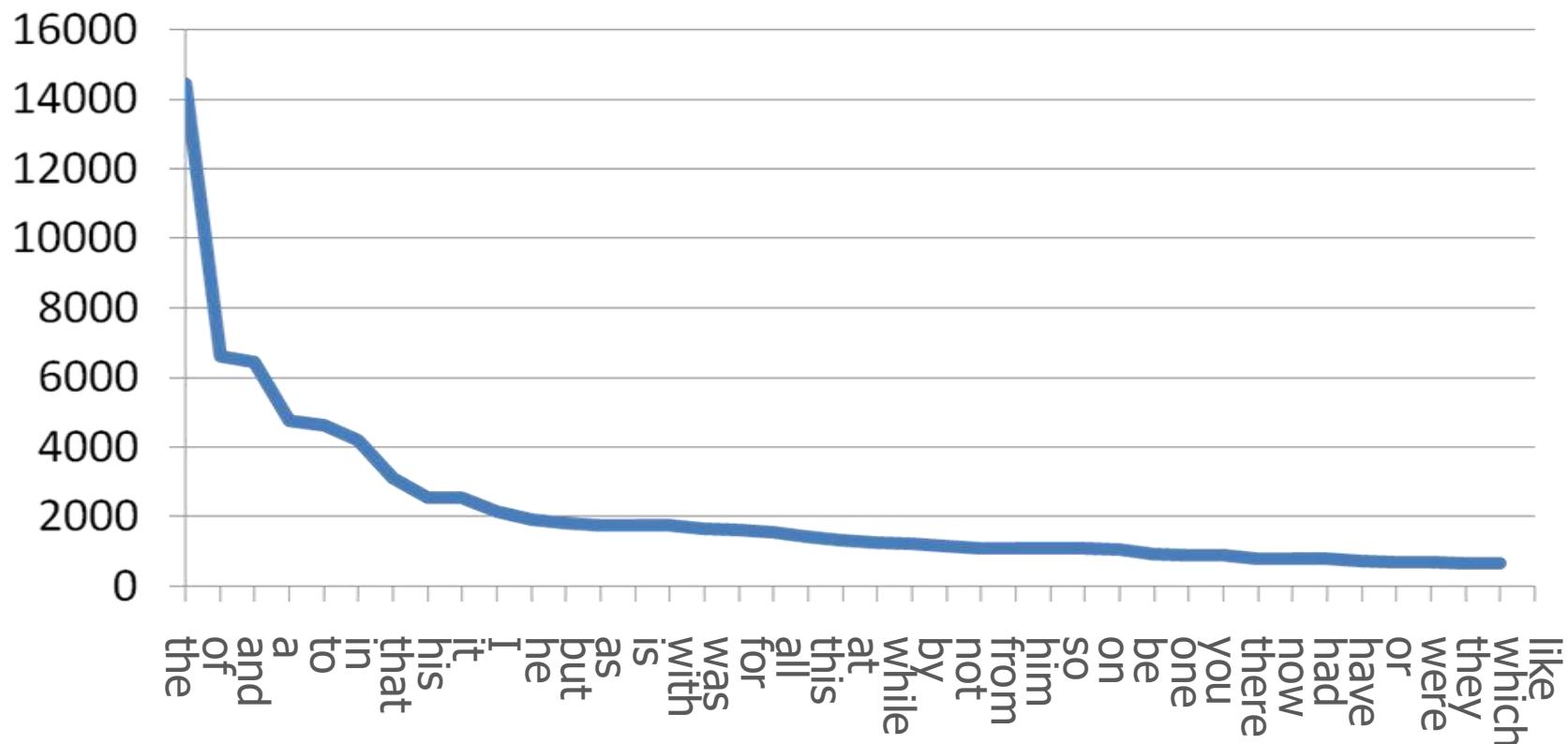


Power laws: typical long tail



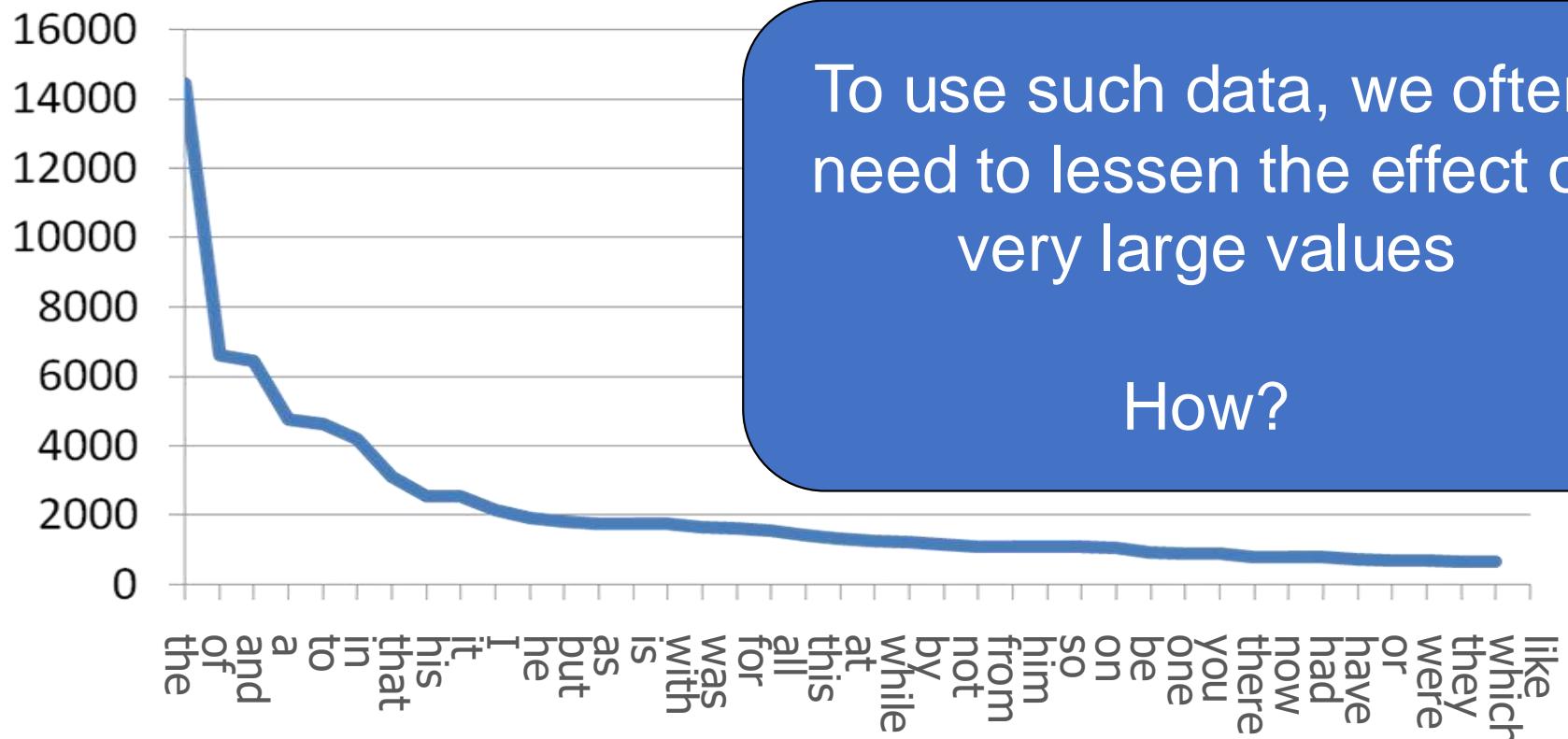
Power laws: Example

- Order words by frequency in a large number of documents



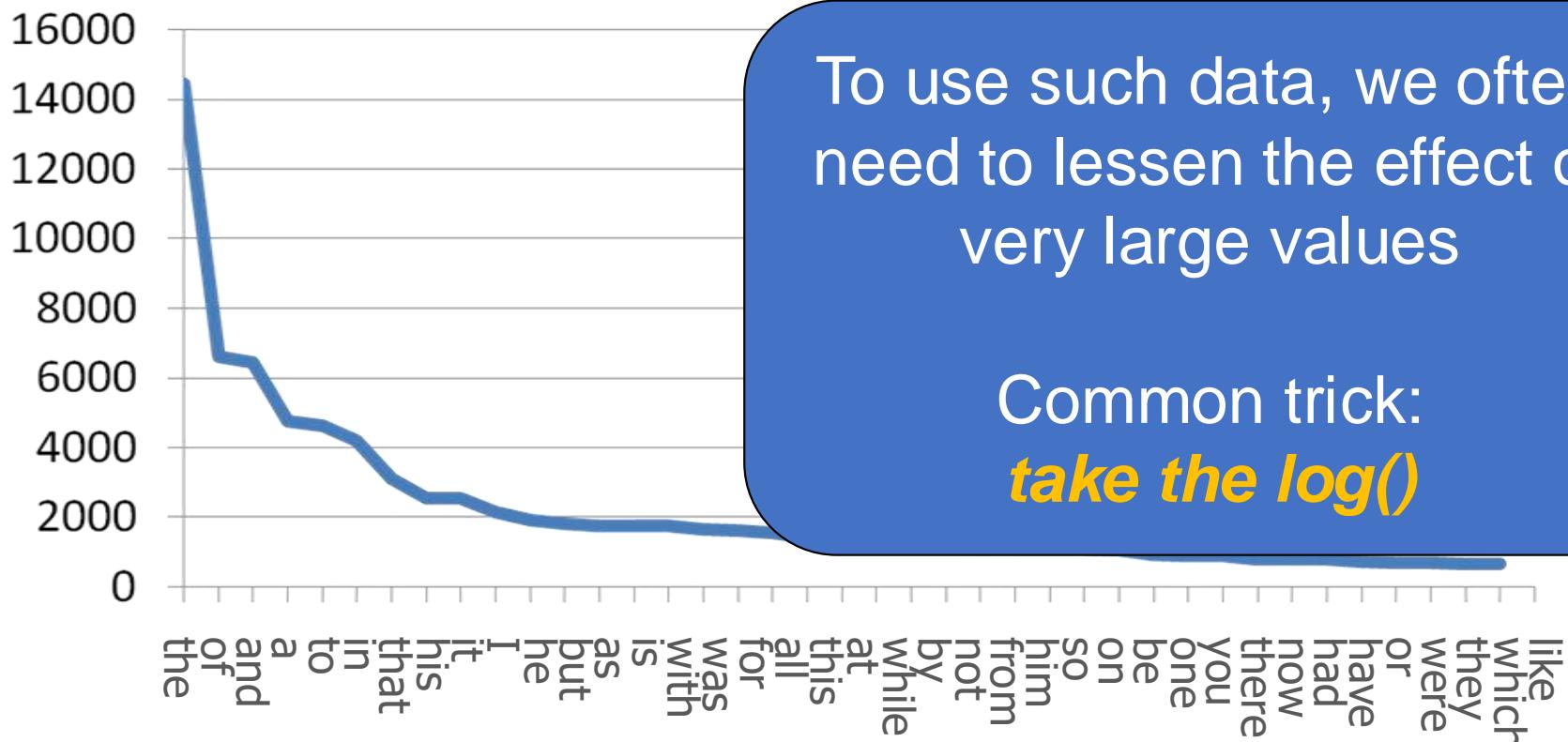
Power laws: Example

- Order words by frequency in a large number of documents



Power laws: Example

- Order words by frequency in a large number of documents



Power laws

- Why do we encounter power laws in many real-world data?

Power laws

- Why do we encounter power laws in many real-world data?
 - If many people read a book, more people will buy it on Amazon
 - If a website is popular, it will be linked more often on the Web
 - Popular cities will attract more people
 - Popular words will be used more often
 - Things are popular or not
- *the rich get richer*

Box-Cox transform

- There exist transformation between several distributions, e.g.
 - normal \leftrightarrow lognormal
 - Pareto \leftrightarrow exponential
- We can also try to make any data normally distributed via optimization:

- Find a λ for $y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}$

that maximizes the **likelihood** = $P(\text{data} \mid \text{model})$, typically using a normal distribution as model

Box-Cox transform

- There exist transformation between:
 - normal \leftrightarrow lognormal
 - Pareto \leftrightarrow exponential
- We can also try to make any data more normally distributed via optimization:

This is called a power transform, a family of monotonic transformations using power functions

- Find a λ for $y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}$

that maximizes the likelihood = $P(\text{data} \mid \text{model})$, typically using a normal distribution as model

Box-Cox can model several transforms:

- $\lambda = 1.00$: original data
- $\lambda = 0.50$: square root transformation
- $\lambda = 0.33$: cube root transformation
- $\lambda = 0.00$: log transformation
- $\lambda = -0.50$: inverse square root transformation
- $\lambda = -1.00$: inverse transformation

...

called a power transform, a family of bijective transformations power functions

optimization:

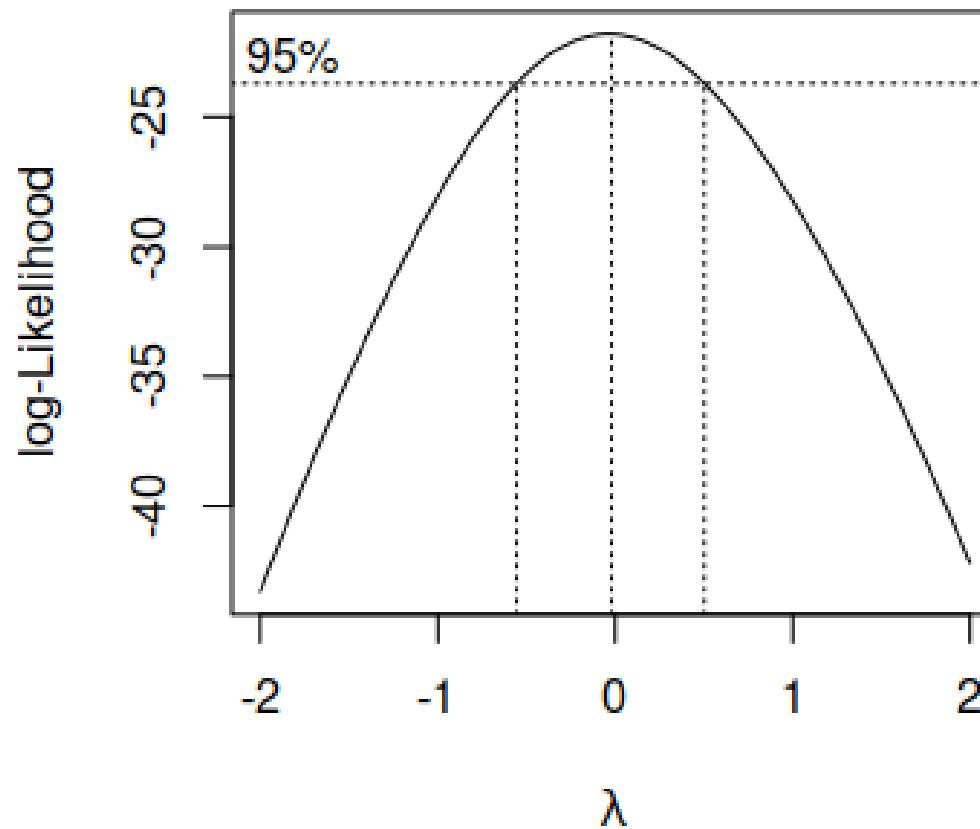
- Find a λ for

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln(y_i) & \text{if } \lambda = 0, \end{cases}$$

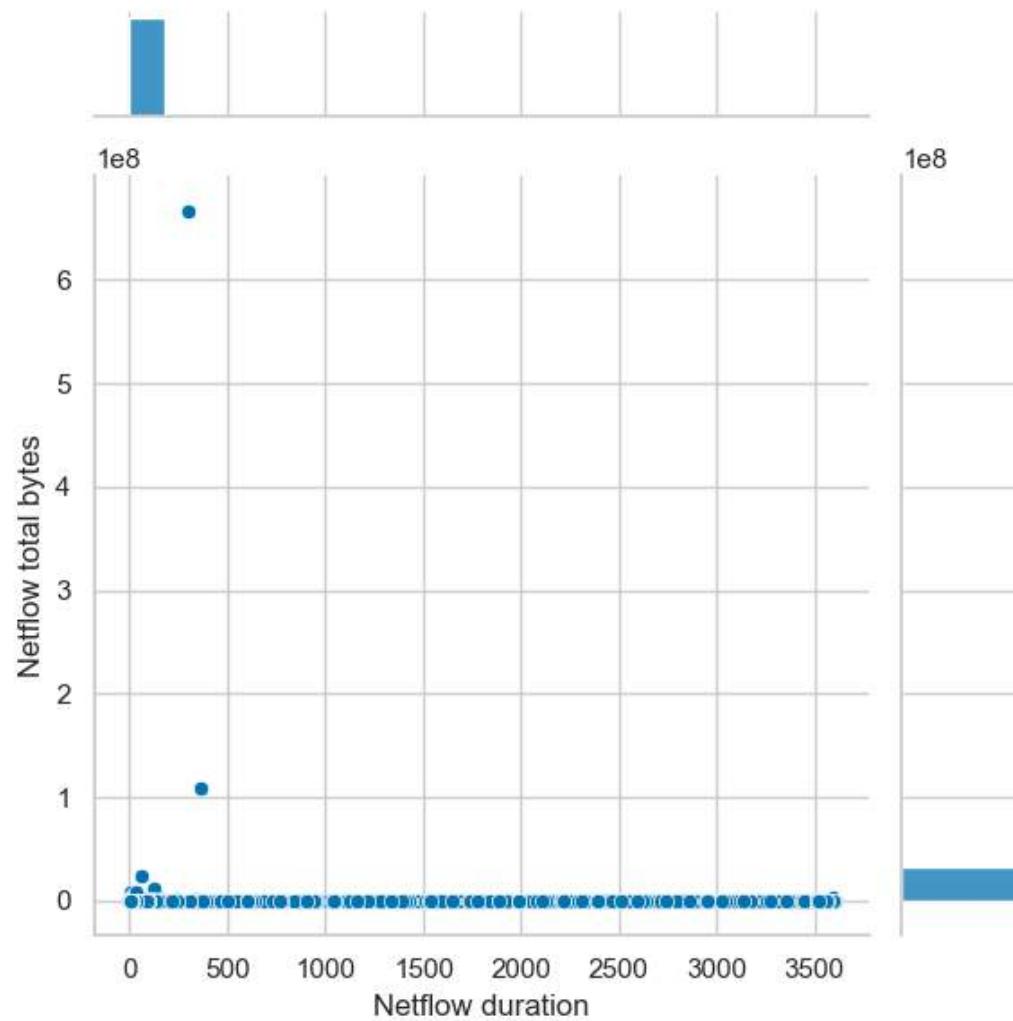
that maximizes the likelihood = $P(\text{data} \mid \text{model})$, typically using a normal distribution as model

Box-Cox transform

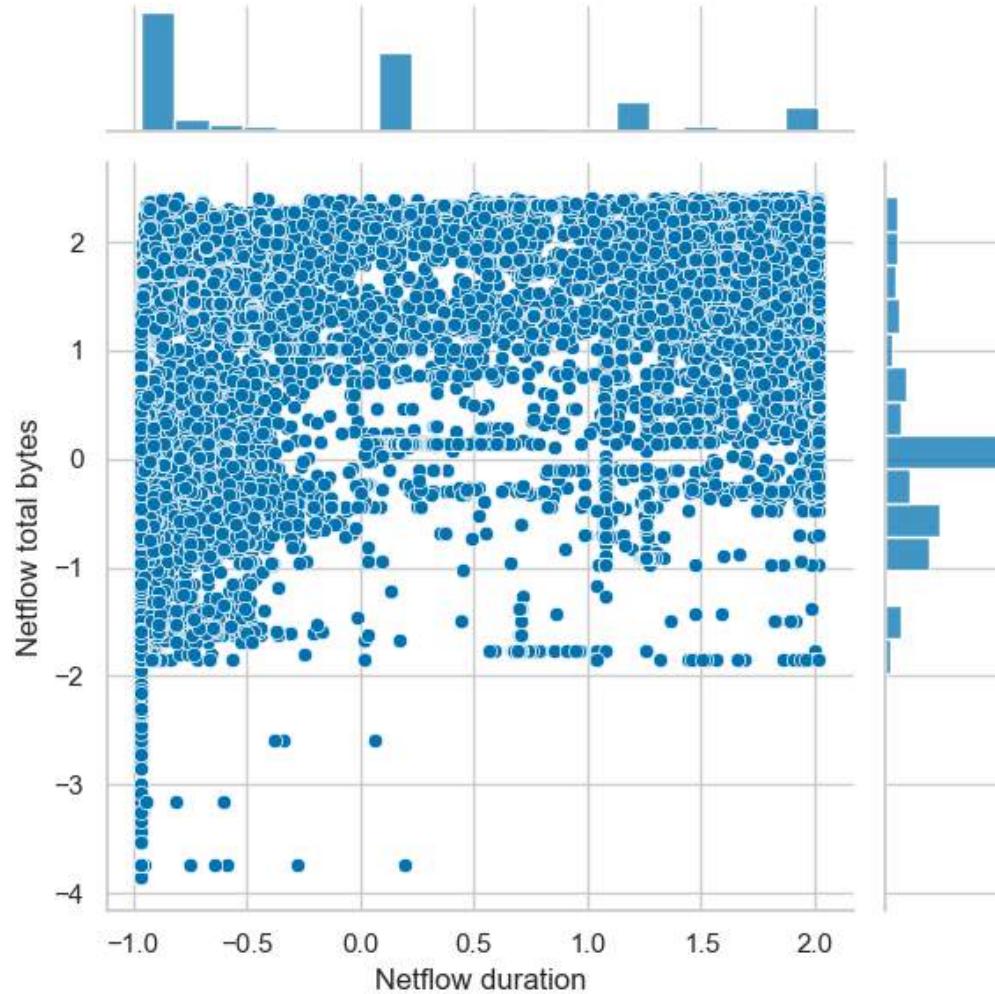
- How to optimize λ ? Try all, pick the maximum



Examples – NetFlow data



Examples – NetFlow data – power



Counting



- Suppose we have stream of items, and we want to count how many items are in the stream
- First method:
 - Set counter to 0, whenever we see an item, increase the counter
 - If asked for count, return the counter

Counting



- Suppose we have stream of items, and we want to count how many items are in the stream
- First method:
 - Set counter to 0, whenever we see an item, increase the counter
 - If asked for count, return the counter
- Second method:
 - Set counter to 0, whenever we see an item, flip a coin:
 - If heads: increase the counter
 - If tails: do nothing
 - If asked for count, return the counter times two

Counting



- Suppose we have stream of items, and we want to count how many items are in the stream
- First method:
 - Set counter to 0, whenever we see an item, increase the counter
 - If asked for count, return the counter
- Second method:
 - Set counter to 0, whenever we see an item, flip a coin:
 - If heads: increase the counter
 - If tails: do nothing
 - If asked for count, return the counter times two

Q: Why would this be useful?

Counting



- Suppose we have stream of items, and we want to count how many items are in the stream
- First method:
 - Set counter to 0, whenever we see an item, increase the counter
 - If asked for count, return the counter
- Second method:
 - Set counter to 0, whenever we see an item, flip a coin:
 - If heads: increase the counter
 - If tails: do nothing
 - If asked for count, return the counter times two

Uses $\log(n)$ bits

Uses $\log(n)-1$ bits

Q: Why would this be useful?

Morris counting

- On data streams, we have to reduce memory usage
- Approximate counting uses several such counters, with different head/tail probabilities:
 - Set counter to 0
 - Update:
 - Draw random number x from $[0, 1]$
 - If $(x \leq 2^{-c})$ $c = c + 1$
 - Query: return $2^c - 2$
- And it runs in $\log(\log(n))$ memory ($E[c] = \log(n)$ )

Data streaming

- Continuous and rapid input of data
- Limited memory to store the data (less than linear in the input size)
- Limited time to process each element
- Sequential access (no random access)
- Algorithms have one ($p=1$) or very few passes ($p=\{2,3\}$) over the data

Data stream models

- Massively long input stream
- Basic model: $\sigma = \langle a_1, a_2, a_3, \dots, a_m \rangle$
with elements drawn from $[n] := 1, 2, \dots, n$
- Space complexity goal: s bits of random-access memory with
$$s = o(\min\{m, n\})$$

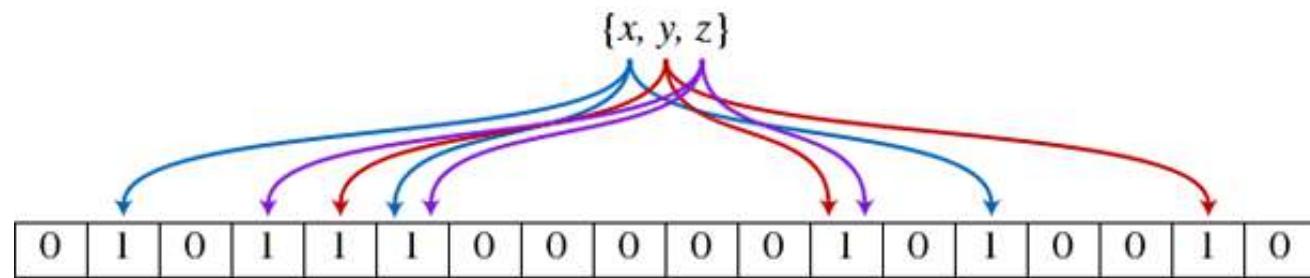
$$s = O(\log m + \log n)$$

“holy grail”

$$s = \text{poly log}(\min(m, n))$$

“reality”

Sketching

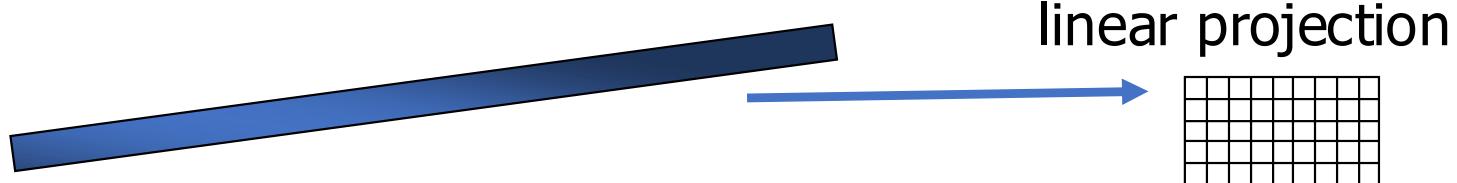


Bloom filter

- <https://www.jasondavies.com/bloomfilter/>
- Why is this useful?

Sketches

- Not every problem can be solved with sampling
 - Example: counting how many distinct items in the stream
 - If a large fraction of items are not sampled, don't know if they are all same or all different
- Other techniques take advantage that the algorithm can “see” all the data even if it can’t “remember” it all
- “*Sketch*”: essentially, a linear transform of the input
 - Model stream as defining a vector, sketch is result of multiplying stream vector by an (implicit) matrix



Bloom filter

- **Given**

- A set of hash functions $\{h_1, h_2, \dots, h_k\}$, $h_i : W \rightarrow [1, n]$
- A bit vector of size n (initialized to **0**)

- To **add** an element to W :

- Compute $h_1(e), h_2(e), \dots, h_k(e)$
- Set the corresponding bits in the bit vector to 1

Usually done once in bulk with few updates.

- To **test** whether an element is in W :

- Compute $h_1(e), h_2(e), \dots, h_k(e)$
- Sum up the returned bits
- Return TRUE if sum=k, FALSE otherwise

Operation on the data stream.

Bloom filter: element testing

- **Case 1:** the element is in W
 - $h_1(e), h_2(e), \dots, h_k(e)$ are all set to 1
 - TRUE is returned with probability 1
- **Case 2:** the element is not in W
 - TRUE is returned if due to some other element all hash values are set

$$P(BV_j \text{ set after } m \text{ inserts}) = 1 - P(BV_j \text{ not set after } m \text{ inserts})$$

Bloom filter: element testing

- **Case 1:** the element is in W
 - $h_1(e), h_2(e), \dots, h_k(e)$ are all set to 1
 - TRUE is returned with probability 1
- **Case 2:** the element is not in W
 - TRUE is returned if due to some other element all hash values are set

$$\begin{aligned} P(BV_j \text{ set after } m \text{ inserts}) &= 1 - P(BV_j \text{ not set after } m \text{ inserts}) \\ &= 1 - P(BV_j \text{ not set after } k \times m \text{ hashes}) \end{aligned}$$

Bloom filter: element testing

- **Case 1:** the element is in W
 - $h_1(e), h_2(e), \dots, h_k(e)$ are all set to 1
 - TRUE is returned with probability 1
- **Case 2:** the element is not in W
 - TRUE is returned if due to some other element all hash values are set

$$\begin{aligned} P(BV_j \text{ set after } m \text{ inserts}) &= 1 - P(BV_j \text{ not set after } m \text{ inserts}) \\ &= 1 - P(BV_j \text{ not set after } k \times m \text{ hashes}) \\ &= 1 - \left(1 - \frac{1}{n}\right)^{k \times m} \end{aligned}$$

Bloom filter: element testing

- **Case 1:** the element is in W
 - $h_1(e), h_2(e), \dots, h_k(e)$ are all set to 1
 - TRUE is returned with probability 1
- **Case 2:** the element is not in W
 - TRUE is returned if due to some other element all hash values are set

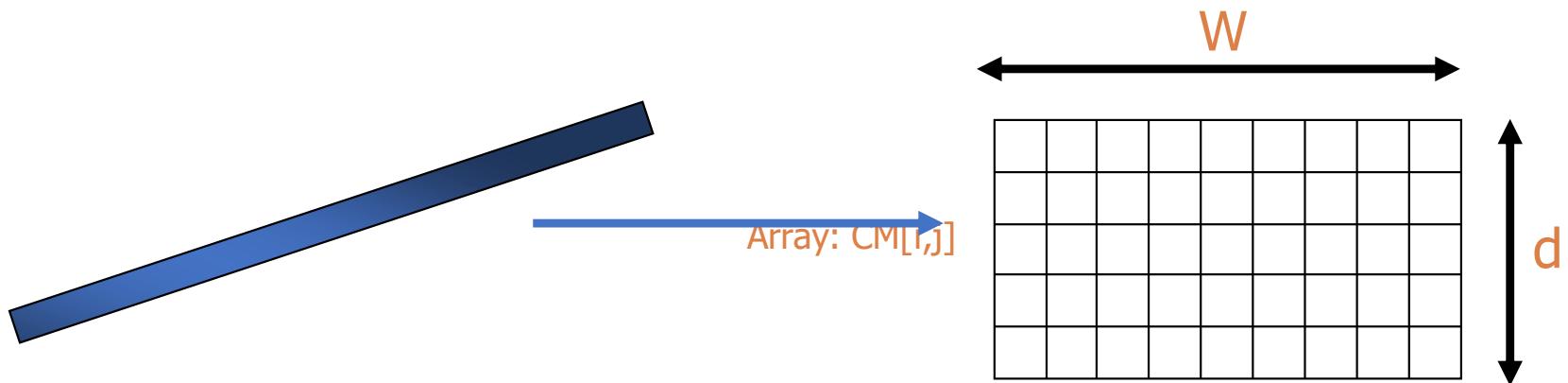
$$\begin{aligned} P(BV_j \text{ set after } m \text{ inserts}) &= 1 - P(BV_j \text{ not set after } m \text{ inserts}) \\ &= 1 - P(BV_j \text{ not set after } k \times m \text{ hashes}) \\ &= 1 - \left(1 - \frac{1}{n}\right)^{k \times m} \\ P(\text{false positive}) &= \left(1 - \left(1 - \frac{1}{n}\right)^{km}\right)^k \end{aligned}$$

Q: A BLOOM filter TRUE/FALSE?

- When a BLOOM filter returns true, the queried element was in the data stream
- When a BLOOM filter returns false, the queried element was not in the data stream

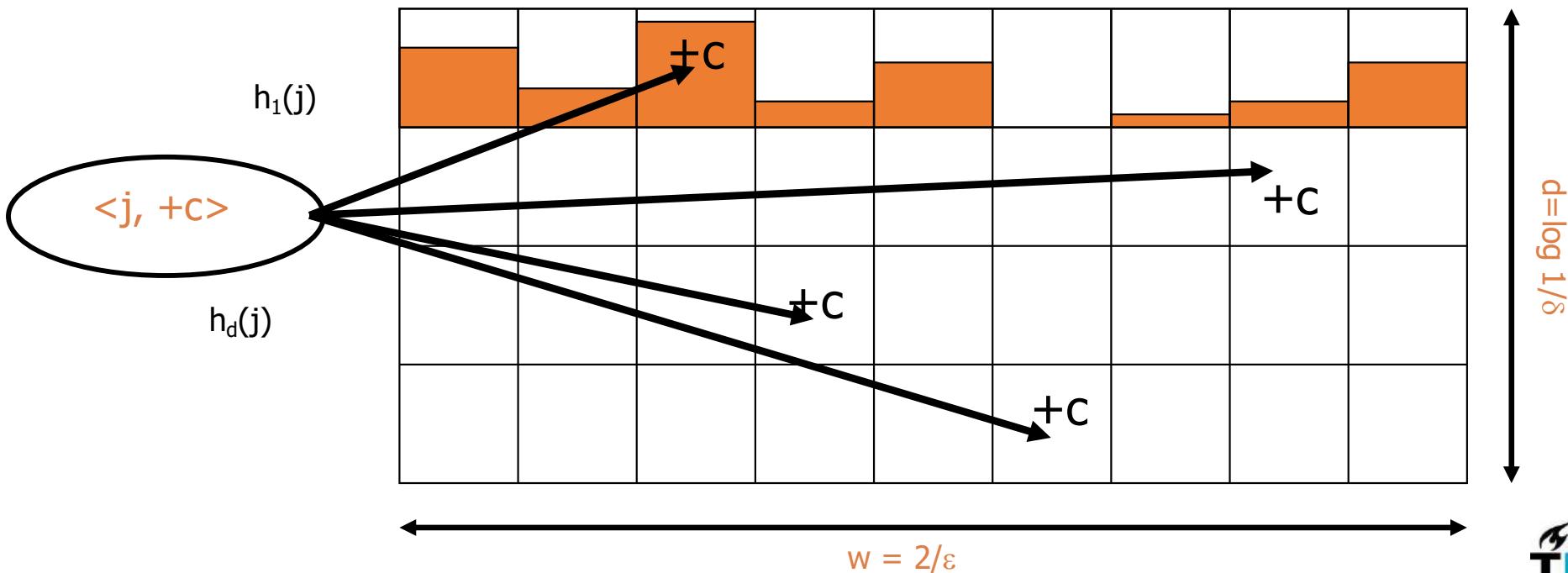
Count-Min Sketch

- Simple sketch idea, can be used for as the basis of many different stream mining tasks
 - Join aggregates, range queries, moments, ...
- Creates a small summary as an array of $w \times d$ in size
- Use d hash functions to map vector entries to $[1..w]$



CM Sketch Structure

- Assume a stream of length m
- Modeled using a vector $A[]$ of counts
- Entries from $A[]$ are mapped to one bucket per row
- Counts merged in buckets are summed up
- Estimate $A[j]$ by taking $\min \{ CM[k, h_k(j)] \}$



CM Sketch Guarantees

- CM sketch guarantees approximation error on point queries less than ϵm in space $O(1/\epsilon \log 1/\delta)$
 - Probability of more error is less than $1-\delta$
- Counts are *biased (overestimates)* due to collisions

CM Sketch Analysis (not exam)

Estimate $A' [j] = \min_k \{ CM[k, h_k(j)] \}$

CM Sketch Analysis (not exam)

Estimate $A'[j] = \min_k \{ CM[k, h_k(j)] \}$

- Analysis: In k 'th row, $CM[k, h_k(j)] = A[j] + X_{k,j}$

CM Sketch Analysis (not exam)

Estimate $A'[j] = \min_k \{ CM[k, h_k(j)] \}$

- Analysis: In k 'th row, $CM[k, h_k(j)] = A[j] + X_{k,j}$
 - $X_{k,j} = \sum A[i \neq j] \mid h_k(i) = h_k(j)$

CM Sketch Analysis (not exam)

Estimate $A'[j] = \min_k \{ CM[k, h_k(j)] \}$

- Analysis: In k 'th row, $CM[k, h_k(j)] = A[j] + X_{k,j}$
 - $X_{k,j} = \sum A[i \neq j] \mid h_k(i) = h_k(j)$
 - $E[X_{k,j}] = \sum A[i \neq j] * \Pr[h_k(i) = h_k(j)] \quad (\text{recall } 2/\varepsilon \text{ columns})$ $\leq (\varepsilon/2) * \sum A[i \neq j] \leq \varepsilon m/2 \quad (\text{recall } m = \text{stream length})$

CM Sketch Analysis (not exam)

Estimate $A'[j] = \min_k \{ CM[k, h_k(j)] \}$

- Analysis: In k 'th row, $CM[k, h_k(j)] = A[j] + X_{k,j}$
 - $X_{k,j} = \sum A[i \neq j] \mid h_k(i) = h_k(j)$
 - $E[X_{k,j}] = \sum A[i \neq j] * \Pr[h_k(i) = h_k(j)] \quad (\text{recall } 2/\varepsilon \text{ columns})$
$$\leq (\varepsilon/2) * \sum A[i \neq j] \leq \varepsilon m/2 \quad (\text{recall } m = \text{stream length})$$
 - $\Pr[X_{k,j} \geq \varepsilon m] = \Pr[X_{k,j} \geq 2E[X_{k,j}]] \leq 1/2 \quad \text{by Markov inequality}$
 - $P(X \geq a) \leq E(x) / a$

CM Sketch Analysis (not exam)

Estimate $A'[j] = \min_k \{ CM[k, h_k(j)] \}$

- Analysis: In k 'th row, $CM[k, h_k(j)] = A[j] + X_{k,j}$
 - $X_{k,j} = \sum A[i \neq j] \mid h_k(i) = h_k(j)$
 - $E[X_{k,j}] = \sum A[i \neq j] * \Pr[h_k(i) = h_k(j)] \quad (\text{recall } 2/\varepsilon \text{ columns})$
$$\leq (\varepsilon/2) * \sum A[i \neq j] \leq \varepsilon m/2 \quad (\text{recall } m = \text{stream length})$$
 - $\Pr[X_{k,j} \geq \varepsilon m] = \Pr[X_{k,j} \geq 2E[X_{k,j}]] \leq 1/2 \quad \text{by Markov inequality}$
 - $\Pr(X \geq a) \leq E(x) / a$
 - $\Pr[A'[j] \geq A[j] + \varepsilon m] = \Pr[\forall k. X_{k,j} > \varepsilon m] \leq 1/2^{\log 1/\delta} = \delta$

CM Sketch Analysis (not exam)

Estimate $A'[j] = \min_k \{ CM[k, h_k(j)] \}$

- Analysis: In k 'th row, $CM[k, h_k(j)] = A[j] + X_{k,j}$
 - $X_{k,j} = \sum A[i \neq j] \mid h_k(i) = h_k(j)$
 - $E[X_{k,j}] = \sum A[i \neq j] * Pr[h_k(i) = h_k(j)]$ (recall $2/\varepsilon$ columns)
 $\leq (\varepsilon/2) * \sum A[i \neq j] \leq \varepsilon m / 2$ (recall m = stream length)
 - $Pr[X_{k,j} \geq \varepsilon m] = Pr[X_{k,j} \geq 2E[X_{k,j}]] \leq 1/2$ by Markov inequality
 - $P(X \geq a) \leq E(x) / a$
 - $Pr[A'[j] \geq A[j] + \varepsilon m] = Pr[\forall k. X_{k,j} > \varepsilon m] \leq 1/2^{\log 1/\delta} = \delta$
 - Final result: with certainty $A[j] \leq A'[j]$ and with probability at least $1-\delta$: $A'[j] < A[j] + \varepsilon m$

FM-sketch (Flajolet-Martin) (not exam)

- Approach: hash data stream elements uniformly to N bit values, i.e.:
- Task: Given a data stream of, estimate the **number of distinct elements** occurring in it (recall Morris counting)
- Assumption: the larger the number of distinct elements in the stream, the more distinct the occurring hash values, and the more likely one with an **unusual property** appears

$$h : a_i \rightarrow \{0, 1\}^N$$

FM-sketch (not exam)

One possibility of interpreting unusual is the hash tail: the number of 0's a binary hash value ends in

100110101110

100110101100

100110000000

for all $a_i \in S$ (our stream):

$$h(a_i) \rightarrow \{0, 1\}^N$$

maximum hash tail seen so far

$$K = \max_{a_i \in S} \text{tail}_{a_i} h(a_i)$$

$$\text{return } |\hat{S}| = 2^K$$

N must be **long enough**;
there must be **more possible results** of the hash function **than elements** in the stream

FM-sketch (not exam)

One possibility of interpreting unusual is the hash tail: the number of 0's a binary hash value ends in

100110101110

100110101100

100110000000

for all $a_i \in S$ (our stream):

$$h(a_i) \rightarrow \{0, 1\}^N$$

maximum hash tail seen so far

$$K = \max_{a_i \in S} \text{tail}_{a_i} h(a_i)$$

$$\text{return } |\hat{S}| = 2^K$$

N must be **long enough**;

the sketch

Why?

FM-sketch (not exam)

- What is unusual?
- Intuition:
 - Say I roll 6 dice, then the probability that I get 1,2,3,4,5,6 as roll is very small
 - But If I repeat this a million times, then the probability that this occurs at some point is very high
 - If I tell you that I rolled the dice X times, and I observed 1,2,3,4,5,6 once
 - You can use these probabilities to estimate X!

FM-sketch (Flajolet-Martin) (not exam)

Intuitive justification

$$P(h(a) \text{ has tail length of at least } r) = \frac{1}{2 \times 2 \dots \times 2} = \frac{1}{2^r}$$

r 0's occur

When there are m distinct elements in the stream

$$P(\text{none has tail length } \geq r) = \left(1 - \frac{1}{2^r}\right)^m$$

if $m \gg 2^r$: the prob. of finding a tail $\geq r$ reaches 1

if $m \ll 2^r$: the prob. of finding a tail $\geq r$ reaches 0

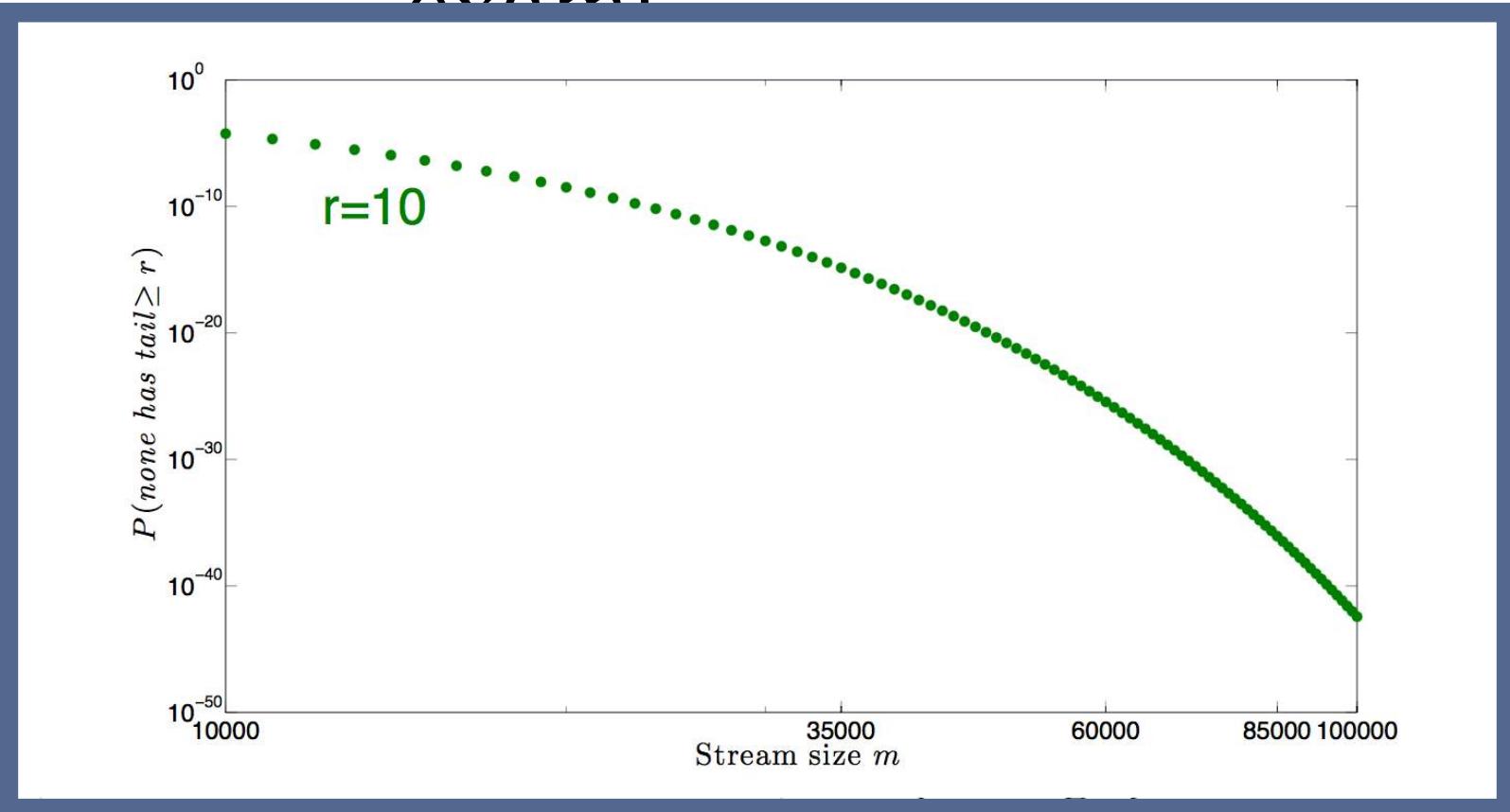
FM-sketch (Flajolet-Martin) (not exactly)

Intu

F

Wher

F



if $m \gg 2^r$: the prob. of finding a tail $\geq r$ reaches 1

if $m \ll 2^r$: the prob. of finding a tail $\geq r$ reaches 0

FM-sketch (Flajolet-Martin) (not exam)

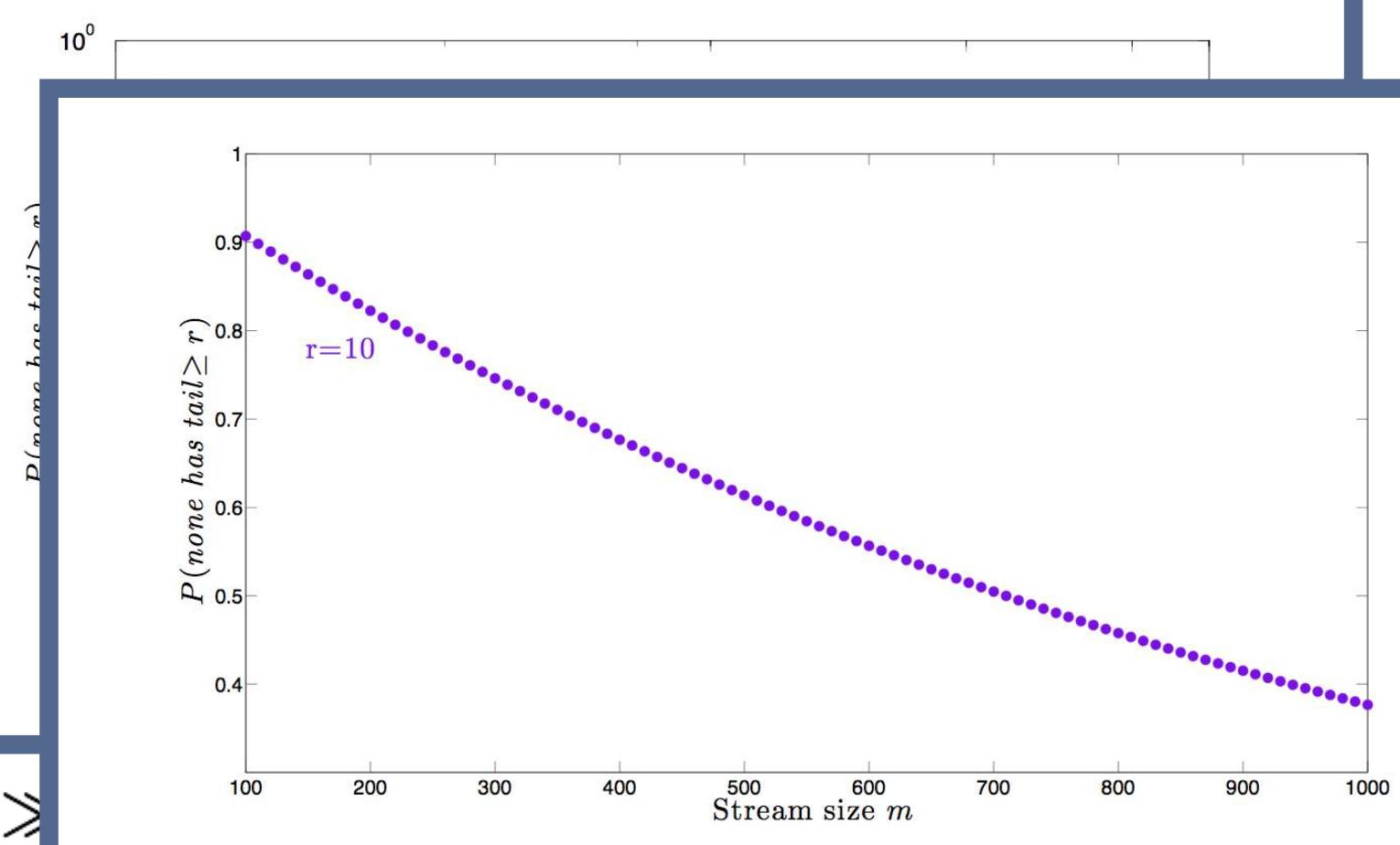
Intu

H

Wh

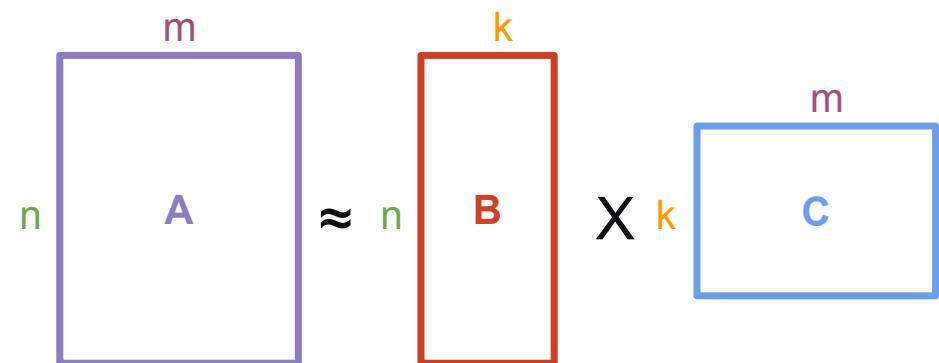
H

if $m \gg$



if $m \ll 2^r$: the prob. of finding a tail $\geq r$ reaches 0

Non-negative Matrix Factorization (NMF)

$$\begin{matrix} m \\ n \end{matrix} \text{A} \approx \begin{matrix} n \\ n \end{matrix} \text{B} \times \begin{matrix} k \\ m \end{matrix} \text{C}$$


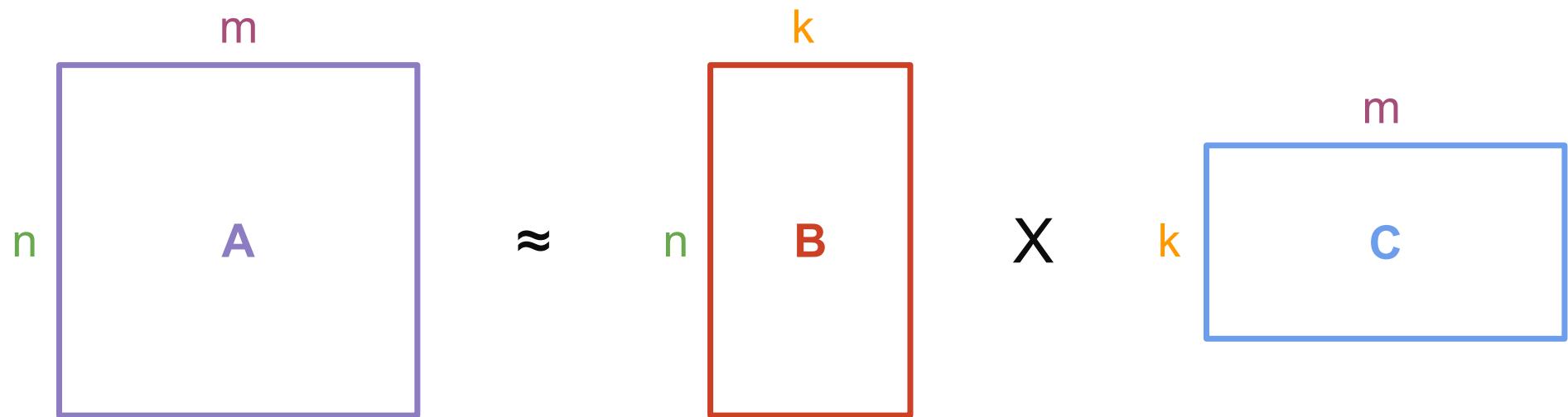
CSE2525, Data Mining

Nergis Tomen

06.01.2025

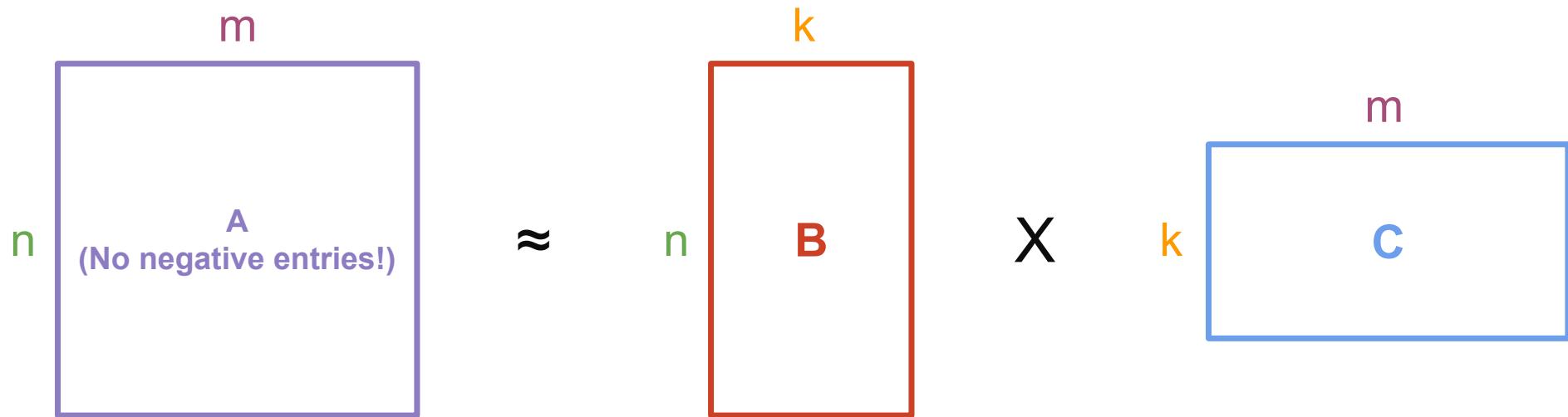
What is NMF?

We factorize a matrix $A_{n \times m}$ into a matrix multiplication of $B_{n \times k}$ and $C_{k \times m}$.



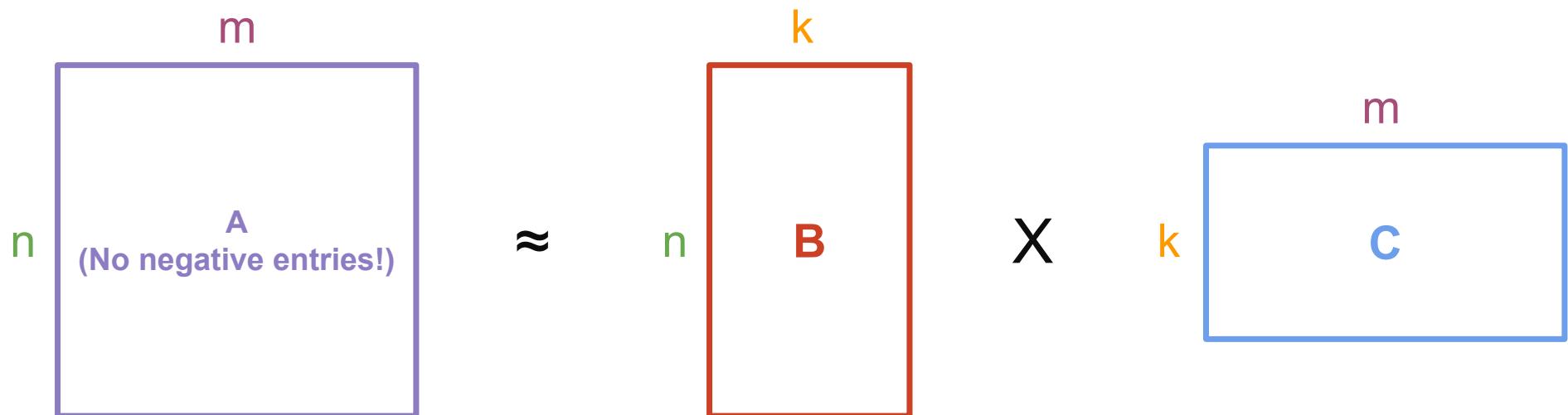
What is NMF? - Assumption

$A_{n \times m}$ is a non-negative matrix.



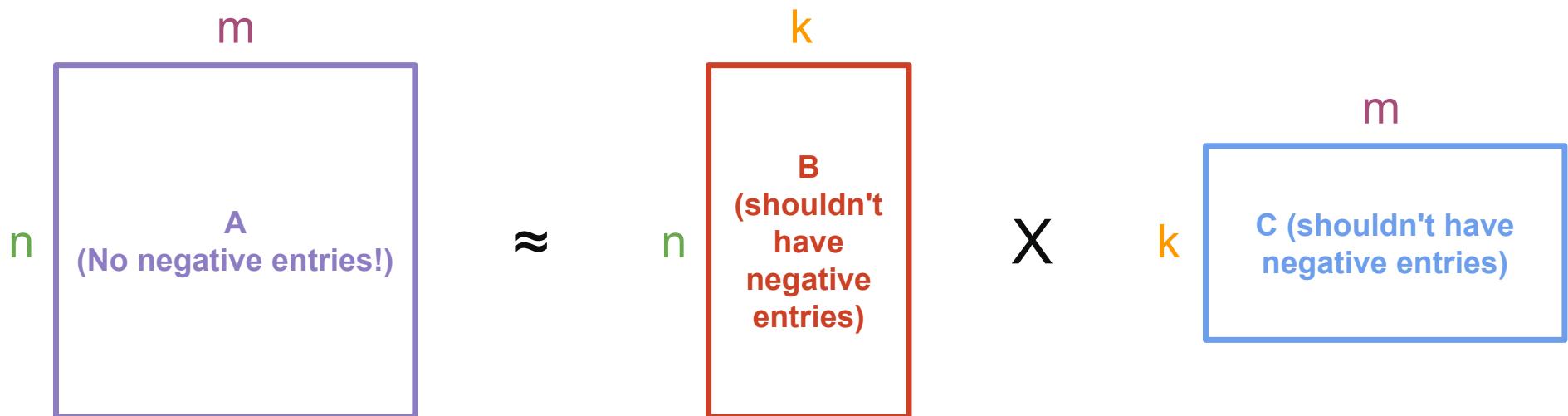
What is NMF? - Assumption

$A_{n \times m}$ is a non-negative matrix. What constraints would you put on this problem?



What is NMF? - Constraints

We want to find non-negative matrices B_{nxk} and C_{kxm} .



Example

n babies

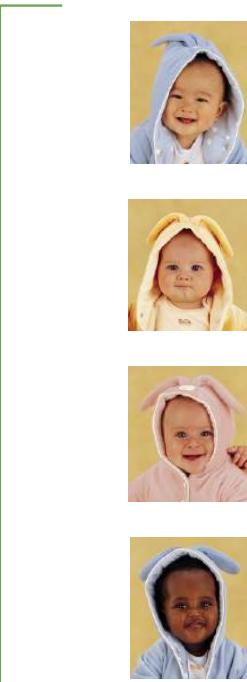


m toys



Example

n babies



m toys



3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

Example

n babies



m toys



3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

What is interesting about this matrix?

Example

n babies



m toys



3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

Ratings are strictly non-negative (1-5).

Example

n babies



m toys



3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

But this doesn't look like our typical "Data matrix" with n 'samples' and m 'features'...

Ratings are strictly non-negative (1-5).

Example

Conceptually: We are connecting sample to sample...

n babies



m toys

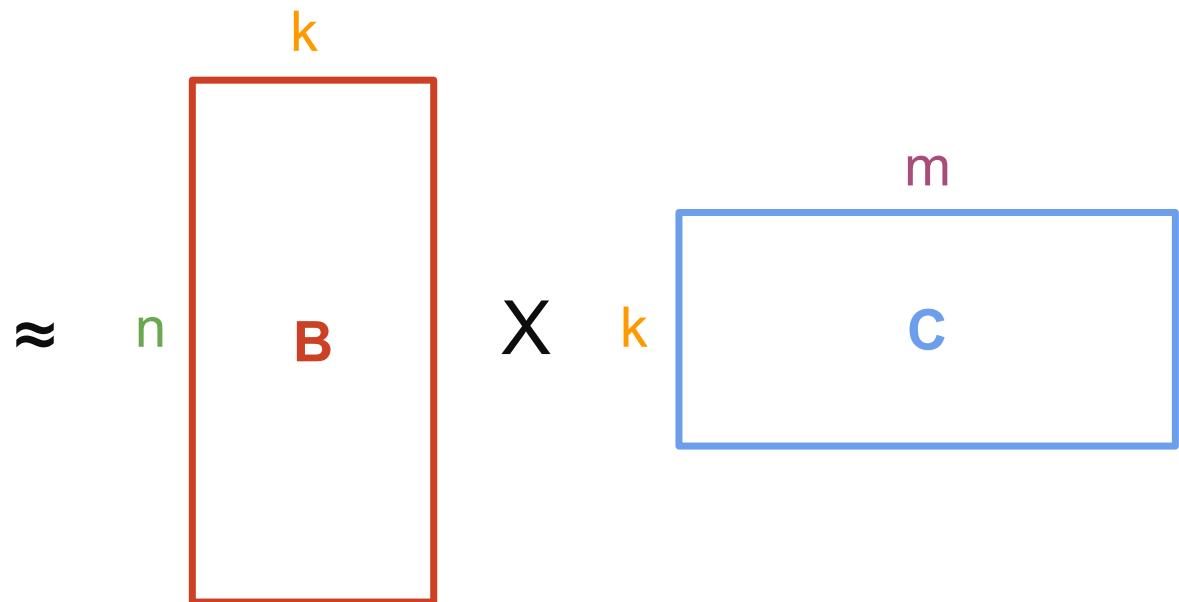
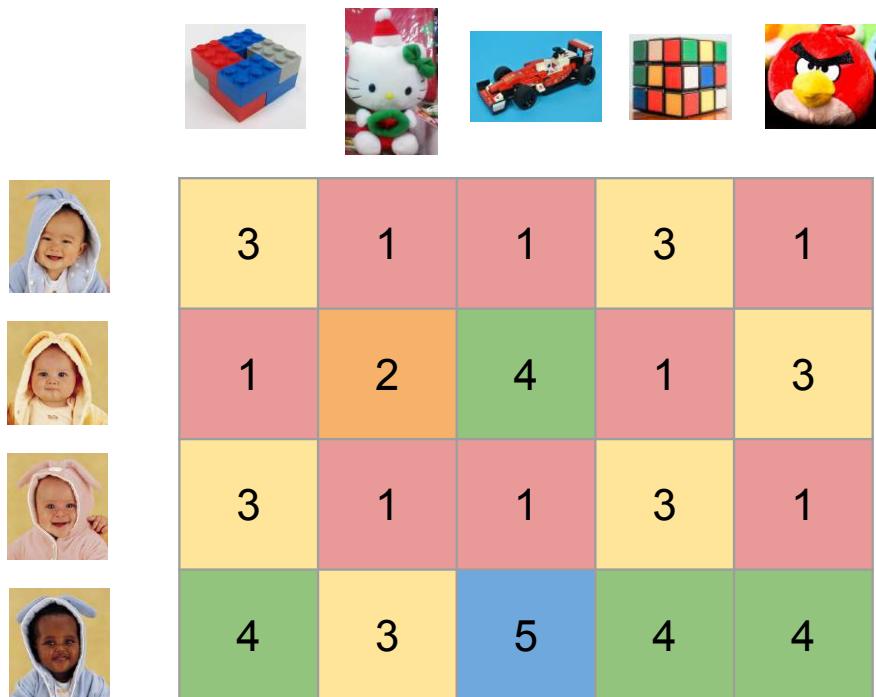


3	1	1	3	1
1	2	4	1	3
3	1	1	3	1
4	3	5	4	4

But this doesn't look like our typical "Data matrix" with n 'samples' and m 'features'...

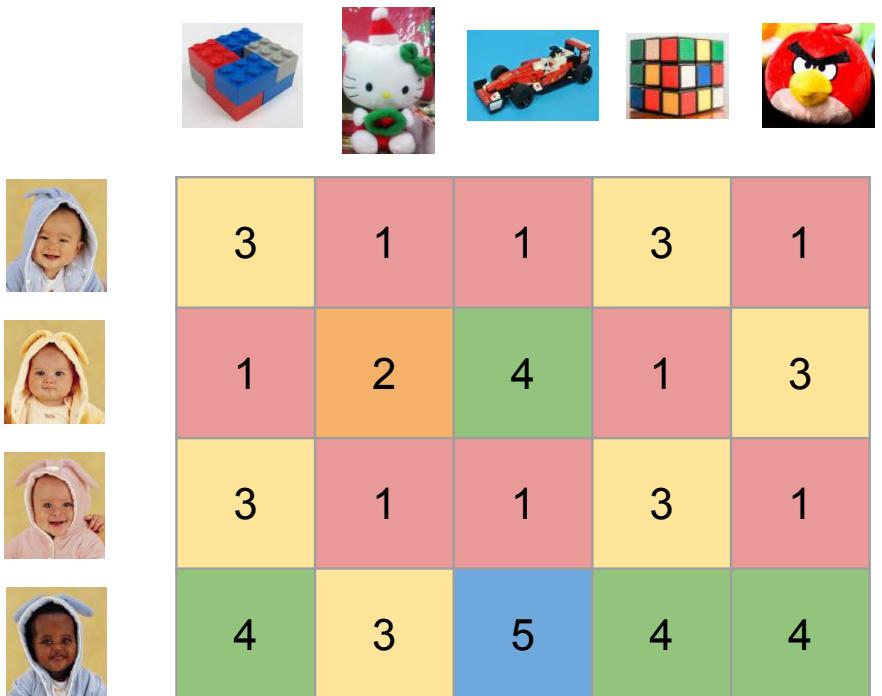
Ratings are strictly non-negative (1-5).

Example

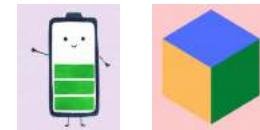


Assumption: Each baby's preference depends on some latent "features" of the toy. e.g. Toy 1 and Toy 4 are "blocky" and maybe Baby 1 prefers "blocky" toys to other shapes.

Example



≈



**B
(feature
prefer-e
ncests)**

n

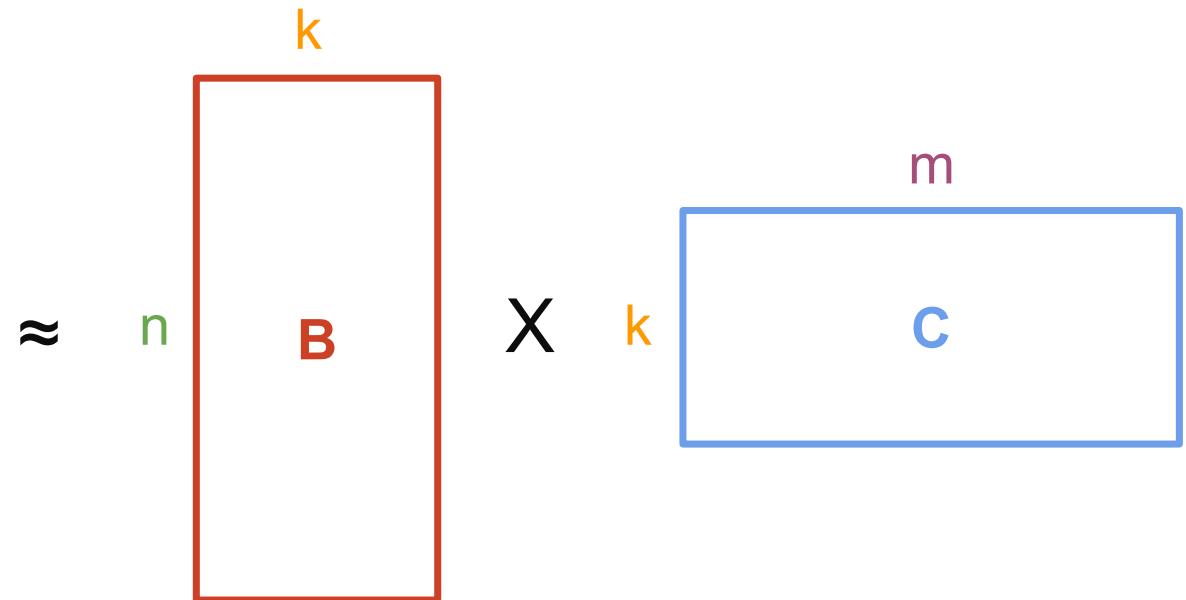


Assumption: Each baby's preference depends on some latent "features" of the toy. e.g. Toy 1 and Toy 4 are "blocky" and maybe Baby 1 prefers "blocky" toys to other shapes.

Example



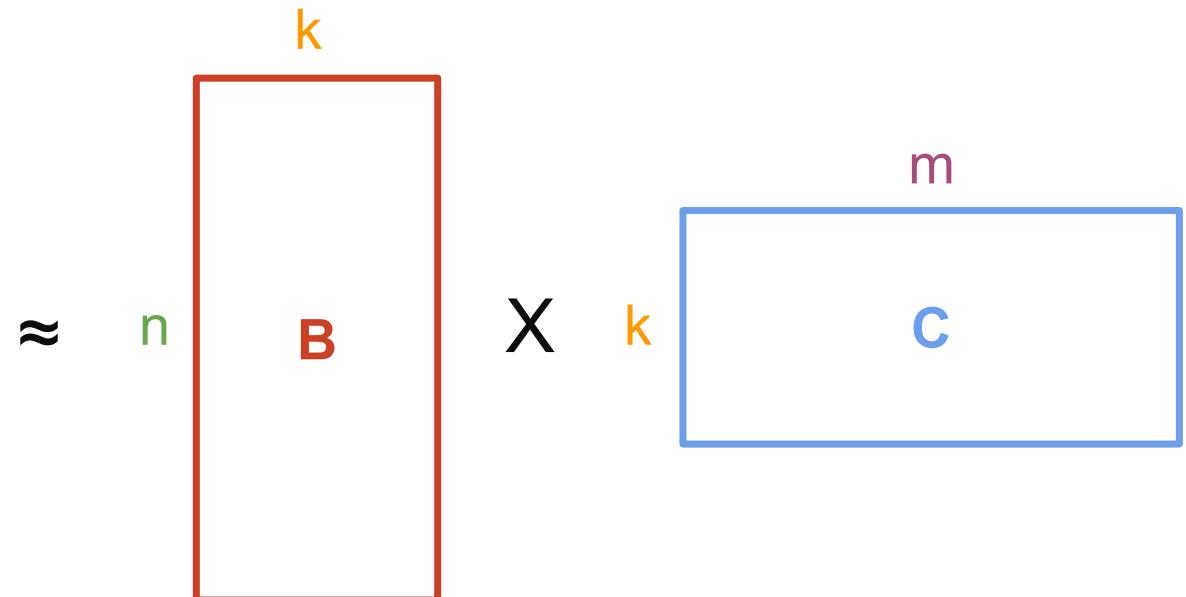
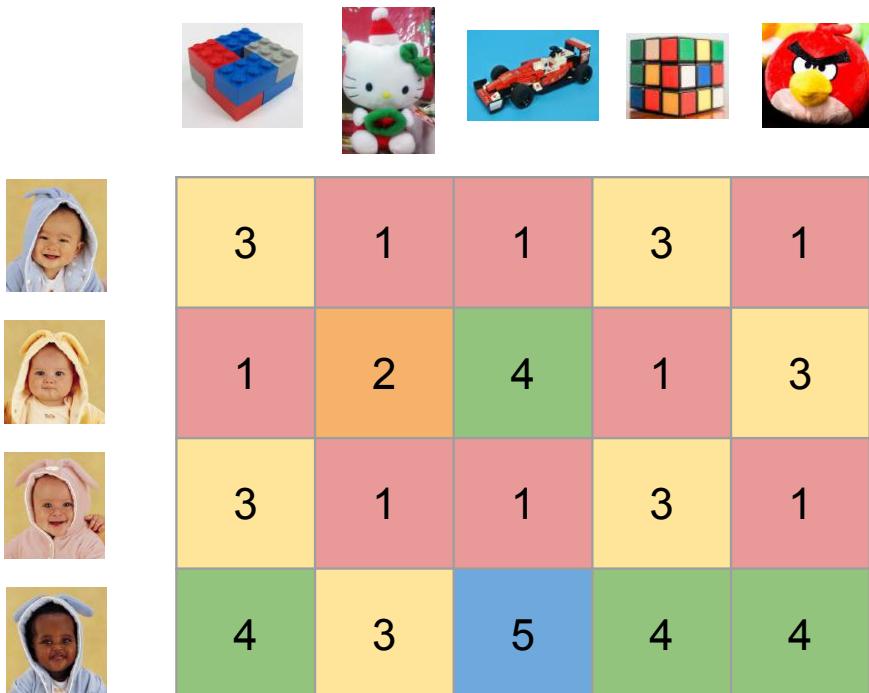
	Blocky Toy	Hello Kitty	Car Toy	Rubik's Cube	Angry Bird
Baby 1	3	1	1	3	1
Baby 2	1	2	4	1	3
Baby 3	3	1	1	3	1
Baby 4	4	3	5	4	4



Assumption: Each baby's preference depends on some latent "features" of the toy. e.g. Toy 1 and Toy 4 are "blocky" and maybe Baby 1 prefers "blocky" toys to other shapes.

How do we find the "latent" features? How do we rate each feature for each toy/each baby? Not clear *a priori*!

Example



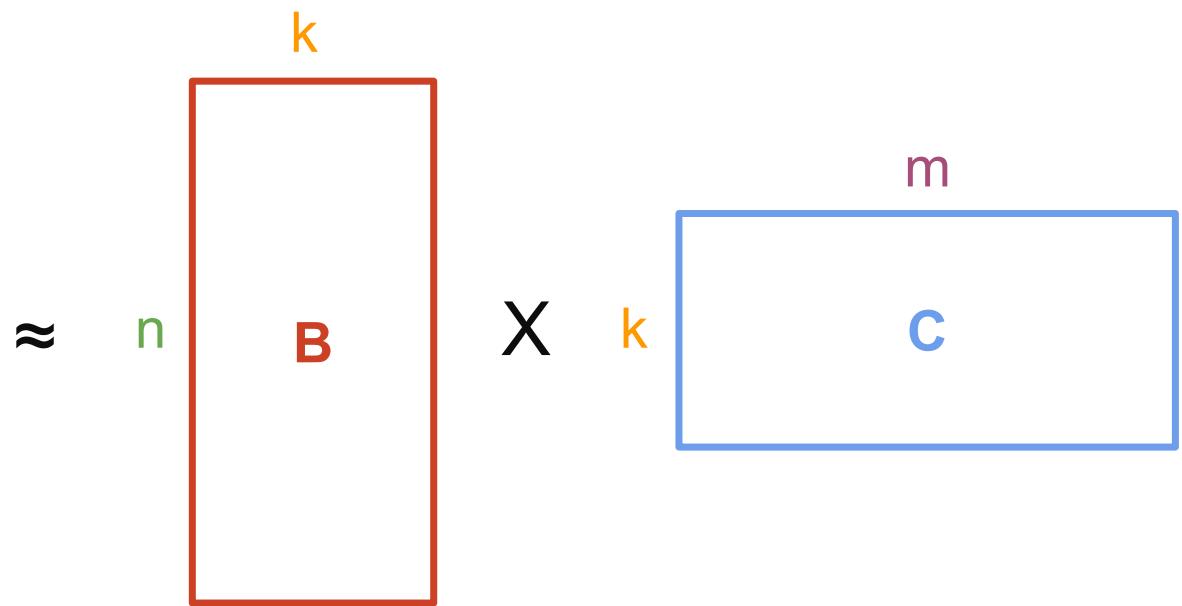
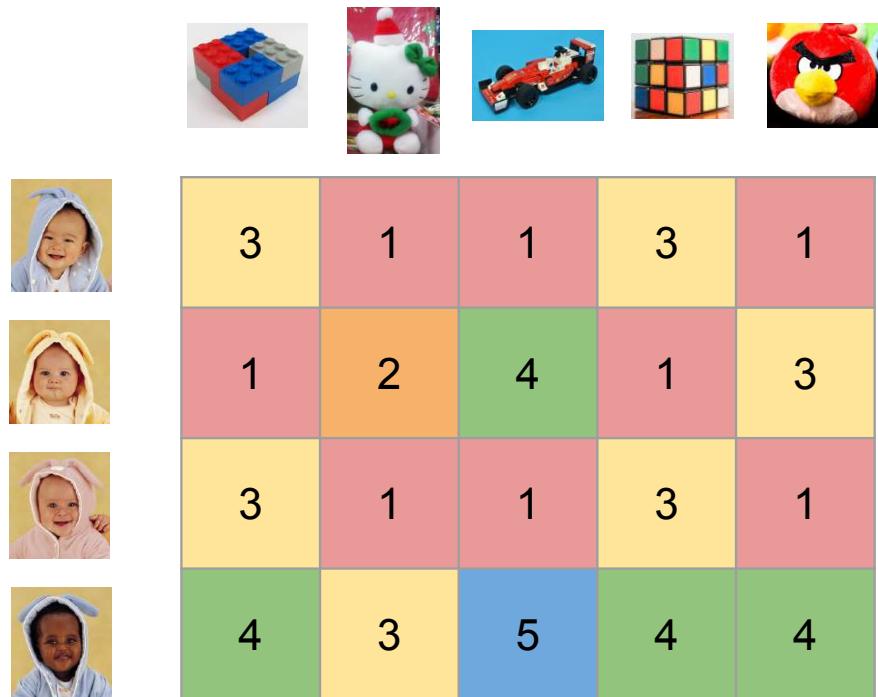
Assumption: Each baby's preference depends on some latent "features" of the toy. e.g. Toy 1 and Toy 4 are "blocky" and maybe Baby 1 prefers "blocky" toys to other shapes.

How do we find the "latent" features? How do we rate each feature for each toy/each baby?

Use NMF!

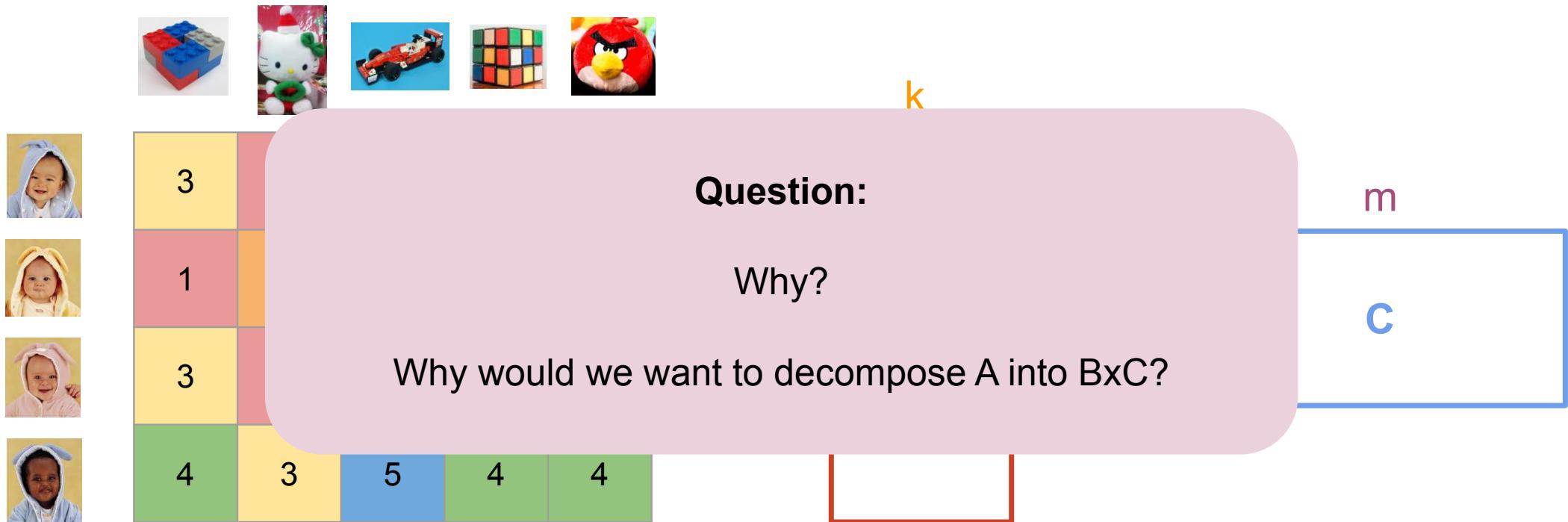
Questions?

Example



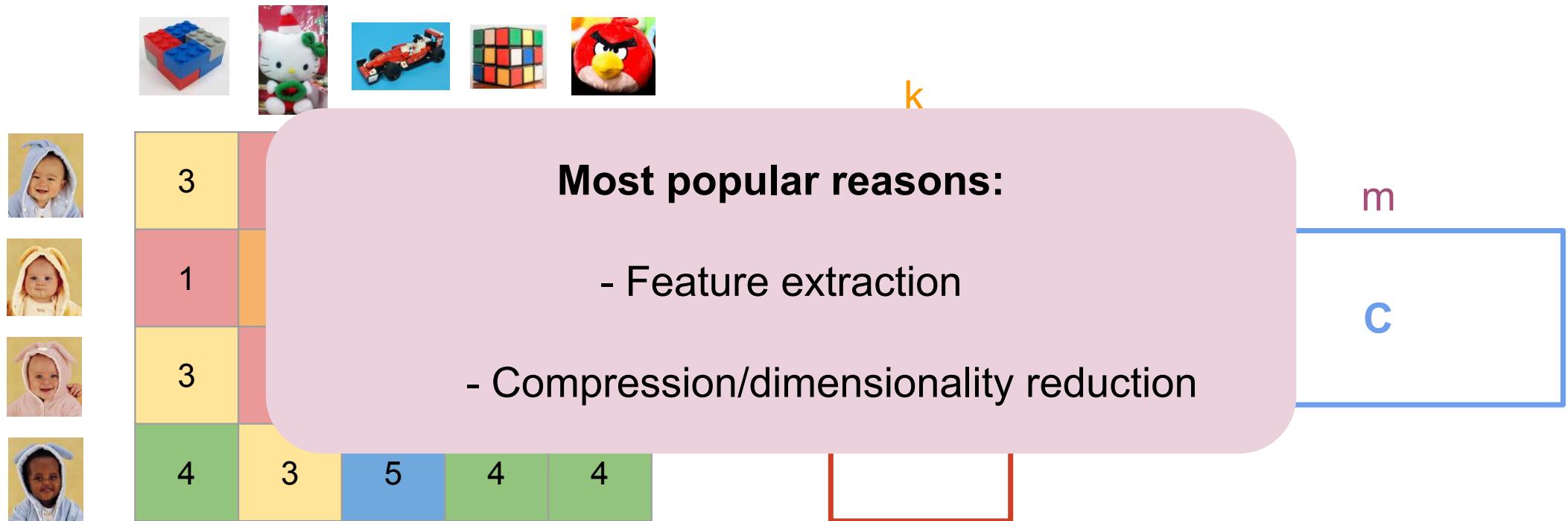
Assumption: Each baby's preference depends on some latent "features" of the toy. e.g. Toy 1 and Toy 4 are "blocky" and maybe Baby 1 prefers "blocky" toys to other shapes.

Example



Assumption: Each baby's preference depends on some latent "features" of the toy. e.g. Toy 1 and Toy 4 are "blocky" and maybe Baby 1 prefers "blocky" toys to other shapes.

Example



Assumption: Each baby's preference depends on some latent "features" of the toy. e.g. Toy 1 and Toy 4 are "blocky" and maybe Baby 1 prefers "blocky" toys to other shapes.

Why is it important?

Many applications where finding "latent features" is important:



- **Astronomy:** Sensor/image recordings are non-negative.
- **Text mining:** Automated feature/topic identification (e.g. A has n documents and m words).



Why is it important?

Many applications where finding "latent features" is important:



- **Astronomy:** Sensor/image recordings are non-negative.
- **Text mining:** Automated feature/topic identification (e.g. A has n documents and m words).



	Princess	Elf	Politician	US Congress	...
Document 1	5	7	0	0	...
Document 2	0	0	8	3	...
...

Why is it important?

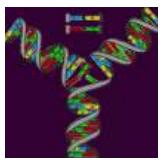
Many applications where finding "latent features" is important:



- **Astronomy:** Sensor/image recordings are non-negative.



- **Text mining:** Automated feature/topic identification (e.g. A has n documents and m words).



- **Genetics/bioinformatics:** e.g. detecting genetically similar clusters in a population.



- **Graph embeddings:** Finding clusters of people with similar "features", e.g. interests.

- **Data completion:** Recommender systems.

Example

n babies



m toys



new
toy

3	1	1	3	1	1
1	2	4	1	3	2
3	1	1	3	1	???
4	3	5	4	4	5

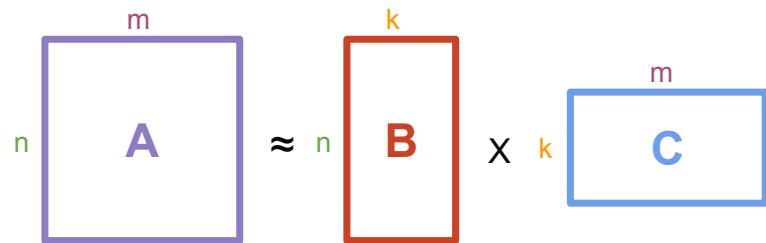
How is it computed?

$$\begin{matrix} & m \\ n & \boxed{A} \end{matrix} \approx \begin{matrix} & k \\ n & \boxed{B} \end{matrix} \times \begin{matrix} & m \\ k & \boxed{C} \end{matrix}$$

Minimization problem:

find matrices **B** and **C**, which minimizes $\|A-BC\|^2$, subject to $B \geq 0$ and $C \geq 0$.

How is it computed?



Minimization problem:

find matrices **B** and **C**, which minimizes $\|A - BC\|^2$, subject to $B \geq 0$ and $C \geq 0$.

$\|A - BC\|$ defines some distance, for NMF it's typically the Frobenius norm (Euclidean distance):

$$\|A\|^2 = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2} = \sqrt{\text{trace}(A^* A)}$$

How is it computed?

$$\begin{matrix} m \\ n \end{matrix} \boxed{A} \approx \begin{matrix} n \\ n \end{matrix} \boxed{B} \times \begin{matrix} k \\ m \end{matrix} \boxed{C}$$

NP-hard optimization problem! [6]

Minimization problem:

find matrices **B** and **C**, which minimizes $\|A-BC\|^2$, subject to $B \geq 0$ and $C \geq 0$.

$\|A-BC\|$ defines some distance, for NMF it's typically the Frobenius norm (Euclidean distance):

$$\|A\|^2 = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2} = \sqrt{\text{trace}(A^* A)}$$

How is it computed?

$$\begin{matrix} m \\ n \end{matrix} \boxed{A} \approx \begin{matrix} n \\ n \end{matrix} \boxed{B} \times \begin{matrix} k \\ m \end{matrix} \boxed{C}$$

NP-hard optimization problem! [6]

Convex in **B** only, or **C** only. Not convex in both variables together. [7]

Minimization problem:

find matrices **B** and **C**, which minimizes $\|A-BC\|^2$, subject to $B \geq 0$ and $C \geq 0$.

$\|A-BC\|$ defines some distance, for NMF it's typically the Frobenius norm (Euclidean distance):

$$\|A\|^2 = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2} = \sqrt{\text{trace}(A^* A)}$$

How is it computed?

$$\begin{matrix} m \\ n \end{matrix} \boxed{A} \approx \begin{matrix} n \\ n \end{matrix} \boxed{B} \times \begin{matrix} k \\ m \end{matrix} \boxed{C}$$

Needs to be solved numerically, but not convex.

Try to find a good local minimum!

Minimization problem:

find matrices **B** and **C**, which minimizes $\|A-BC\|^2$, subject to $B \geq 0$ and $C \geq 0$.

$\|A-BC\|$ defines some distance, for NMF it's typically the Frobenius norm (Euclidean distance):

$$\|A\|^2 = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2} = \sqrt{\text{trace}(A^* A)}$$

How is it computed?

$$\begin{matrix} m \\ n \end{matrix} \boxed{A} \approx \begin{matrix} n \\ n \end{matrix} \boxed{B} \times \begin{matrix} k \\ m \end{matrix} \boxed{C}$$

Question:

Try to find a good local minimum:
How would you compute it?

Minimization problem:

find matrices **B** and **C**, which minimizes $\|A-BC\|^2$, subject to $B \geq 0$ and $C \geq 0$.

$\|A-BC\|$ defines some distance, for NMF it's typically the Frobenius norm (Euclidean distance):

$$\|A\|^2 = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2} = \sqrt{\text{trace}(A^* A)}$$

How is it computed?

$$\begin{matrix} m \\ n \end{matrix} \boxed{A} \approx \begin{matrix} n \\ n \end{matrix} \boxed{B} \times \begin{matrix} k \\ m \end{matrix} \boxed{C}$$

Gradient descent is typical, but

- Convergence is slow in high-dimensions
- Sensitive to choice of step size

Minimization problem:

find matrices **B** and **C**, which minimizes $\|A-BC\|^2$, subject to $B \geq 0$ and $C \geq 0$.

$\|A-BC\|$ defines some distance, for NMF it's typically the Frobenius norm (Euclidean distance):

$$\|A\|^2 = \sqrt{\sum_i^m \sum_j^n |a_{ij}|^2} = \sqrt{\text{trace}(A^* A)}$$

How is it computed?

Algorithm: Multiplicative update [7] to find $\mathbf{W}\mathbf{H} \approx \mathbf{V}$.

$$\underset{n}{\textcolor{purple}{\boxed{V}}} \underset{m}{\text{ }} \approx \underset{n}{\textcolor{green}{\boxed{W}}} \underset{k}{\text{ }} \times \underset{k}{\textcolor{blue}{\boxed{H}}}$$

Notation side note: In literature **W** and **H** are often used to refer to “Weights” and “Hidden units” respectively. **V** is used for “Visible units”.

How is it computed?

Algorithm: Multiplicative update [7] to find $\mathbf{WH} \approx \mathbf{V}$. First initialize \mathbf{W} and \mathbf{H} randomly.

Minimize $\|\mathbf{V} - \mathbf{WH}\|^2$ by alternately updating \mathbf{W} and \mathbf{H} as

$$W_{ij} \leftarrow W_{ij} (VH^T)_{ij} / (WHH^T)_{ij}$$

$$H_{ij} \leftarrow H_{ij} (W^T V)_{ij} / (W^T WH)_{ij}$$

How is it computed?

Algorithm: Multiplicative update [7] to find $\mathbf{WH} \approx \mathbf{V}$. First initialize \mathbf{W} and \mathbf{H} randomly.

Minimize $\|\mathbf{V} - \mathbf{WH}\|^2$ by alternately updating \mathbf{W} and \mathbf{H} as

$$W_{ij} \leftarrow W_{ij} (VH^T)_{ij} / (WHH^T)_{ij}$$

$$H_{ij} \leftarrow H_{ij} (W^T V)_{ij} / (W^T WH)_{ij}$$

Advantages?

How is it computed?

Algorithm: Multiplicative update [7] to find $\mathbf{WH} \approx \mathbf{V}$. First initialize \mathbf{W} and \mathbf{H} randomly.

Minimize $\|\mathbf{V} - \mathbf{WH}\|^2$ by alternately updating \mathbf{W} and \mathbf{H} as

$$W_{ij} \leftarrow W_{ij} (\mathbf{V}\mathbf{H}^T)_{ij} / (\mathbf{W}\mathbf{H}\mathbf{H}^T)_{ij}$$

$$H_{ij} \leftarrow H_{ij} (\mathbf{W}^T\mathbf{V})_{ij} / (\mathbf{W}^T\mathbf{W}\mathbf{H})_{ij}$$

Advantages?

- Just multiplication, very **easy to implement**, ensures non-negativity
- **Faster** than gradient descent based methods, no need to finetune step size
- Euclidean distance $\|\mathbf{V} - \mathbf{WH}\|^2$ is guaranteed to be **non-increasing**
- **Guaranteed to converge** (to some local minimum) in finite time

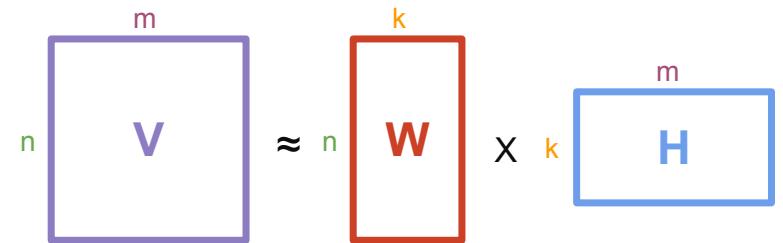
How is it computed?

$$\underset{n}{\text{V}} \overset{m}{=} \underset{n}{\text{W}} \times \underset{k}{\text{H}}$$

Algorithm: Multiplicative update in steps (lab assignment).

0) Normalize? (Not always necessary! Check the range of your data! It might be a good idea to divide V by a scalar if the range \gg your initializations.)

How is it computed?



Algorithm: Multiplicative update in steps (lab assignment).

0) Normalize? (Not always necessary! Check the range of your data! It might be a good idea to divide V by a scalar if the range $>>$ your initializations.)

1) Initialize W and H randomly (all W_{ij} and H_{ij} drawn from a uniform distribution in the interval $(0,1)$).

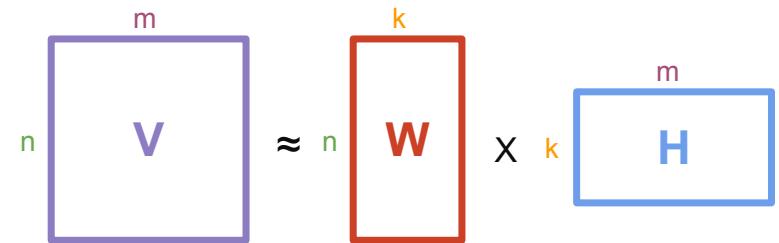
Compute the reconstruction error $E = \|V - WH\|^2$.

2) Individually update W and H to minimize $\|V - WH\|^2$ using

$$W_{ij} \leftarrow W_{ij} (VH^T)_{ij} / ((WHH^T)_{ij} + \epsilon)$$

$$H_{ij} \leftarrow H_{ij} (W^T V)_{ij} / ((W^T WH)_{ij} + \epsilon)$$

How is it computed?



Algorithm: Multiplicative update in steps (lab assignment).

0) Normalize? (Not always necessary! Check the range of your data! It might be a good idea to divide V by a scalar if the range \gg your initializations.)

1) Initialize W and H randomly (all W_{ij} and H_{ij} drawn from a uniform distribution in the interval $(0,1)$).

Compute the reconstruction error $E = \|V - WH\|^2$.

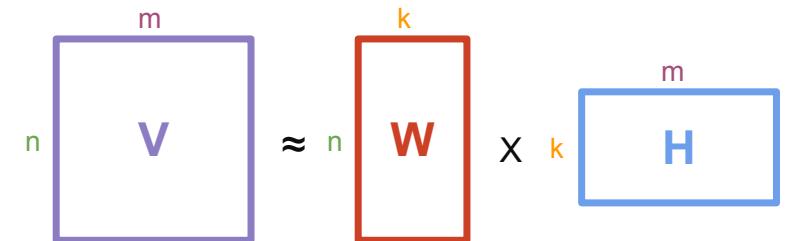
2) Individually update W and H to minimize $\|V - WH\|^2$ using

$$W_{ij} \leftarrow W_{ij} (VH^T)_{ij} / ((WHH^T)_{ij} + \epsilon)$$

$$H_{ij} \leftarrow H_{ij} (W^T V)_{ij} / ((W^T WH)_{ij} + \epsilon)$$

Open or closed
interval?

How is it computed?



Algorithm: Multiplicative update in steps (lab assignment).

0) Normalize? (Not always necessary! Check the range of your data! It might be a good idea to divide V by a scalar if the range \gg your initializations.)

1) Initialize W and H randomly (all W_{ij} and H_{ij} drawn from a uniform distribution in the interval $(0,1)$).

Compute the reconstruction error $E = \|V - WH\|^2$.

2) Individually update W and H to minimize $\|V - WH\|^2$ using

$$W_{ij} \leftarrow W_{ij} (VH^T)_{ij} / ((WHH^T)_{ij} + \epsilon)$$

$$H_{ij} \leftarrow H_{ij} (W^T V)_{ij} / ((W^T WH)_{ij} + \epsilon)$$

Make sure there is no division by 0 (can use an ϵ term).

3) Compute the new reconstruction error $E_{\text{new}} = \|V - WH\|^2$

4) Stop updating and end optimization if $E - E_{\text{new}} <$ a predefined error tolerance.

5) While $(E - E_{\text{new}})$ isn't small enough, repeat steps 2-4 for a predefined number of maximum iterations.

How is it computed?

$$\begin{matrix} & m \\ V & \end{matrix} \approx \begin{matrix} & n \\ W & \end{matrix} \times \begin{matrix} & k \\ H & \end{matrix}$$

Algorithm: Multiplicative update in steps (lab assignment).

0) Normalize? (Not always necessary! Check the range of your data! It might be a good idea to divide V by a scalar if the range \gg your initializations.)

1) Initialize W and H randomly from the uniform distribution in the interval $(0,1)$.

Compute the reconstruction WH .

Potential design choices:

- Normalization (yes/no?)
- Number of features k (for W, H initialization)
 - Error tolerance
 - Maximum iterations

Make sure there is no division by zero.

3) Compute the new reconstruction error $E_{\text{new}} = \|V - WH\|^2$

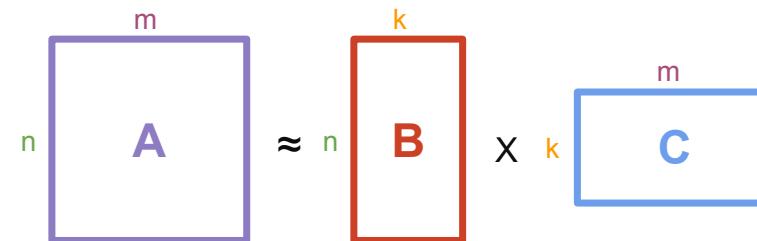
4) Stop updating and end optimization if $E - E_{\text{new}} <$ a predefined error tolerance.

5) While $(E - E_{\text{new}})$ isn't small enough, repeat steps 2-4 for a predefined number of maximum iterations.

Questions?

How do we choose k?

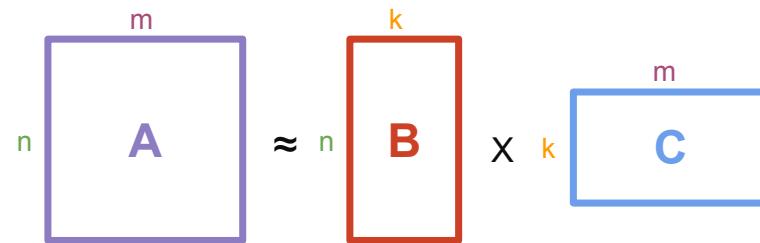
The dimensions of the factors **B** and **C** can be significantly smaller than dimensions of **A**.



What happens if $k \ll n$ and $k \ll m$?...

How do we choose k?

The dimensions of the factors **B** and **C** can be significantly smaller than dimensions of **A**.



What happens if $k \ll n$ and $k \ll m$?

Compression!... but compression works best when it's eliminating statistical redundancy.

Linear dependence

Linear dependence is a source of "statistical redundancy" in a data matrix. We need linear dependence in the data to have meaningful and/or lossless compression by NMF.

Linear dependence

Linear dependence is a source of "statistical redundancy" in a data matrix. We need linear dependence in the data to have meaningful and/or lossless compression by NMF.

Definition: A sequence of vectors $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_k$ are linearly dependent, if there exist any scalars $a_1, a_2, \dots, a_k > 0$ such that

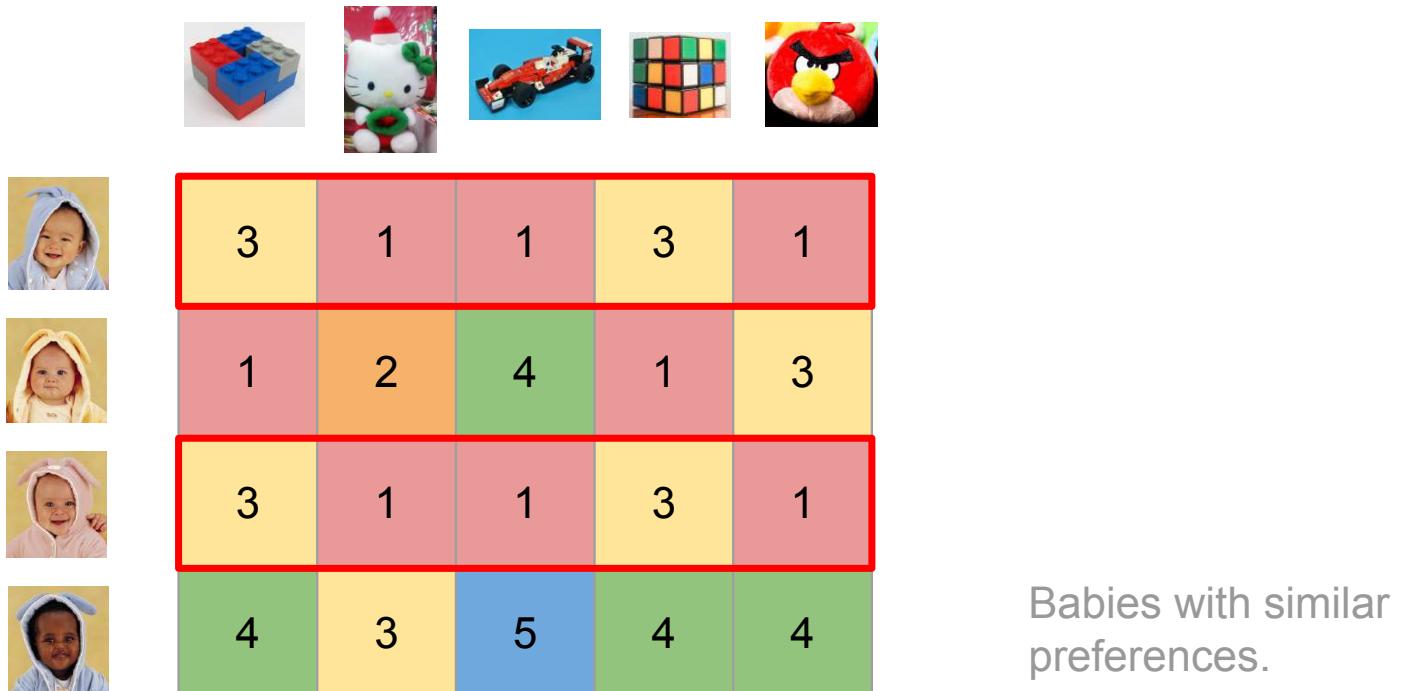
$$a_1\mathbf{v}_1 + a_2\mathbf{v}_2 + \dots + a_k\mathbf{v}_k = \mathbf{0}.$$

This is true if, for example,

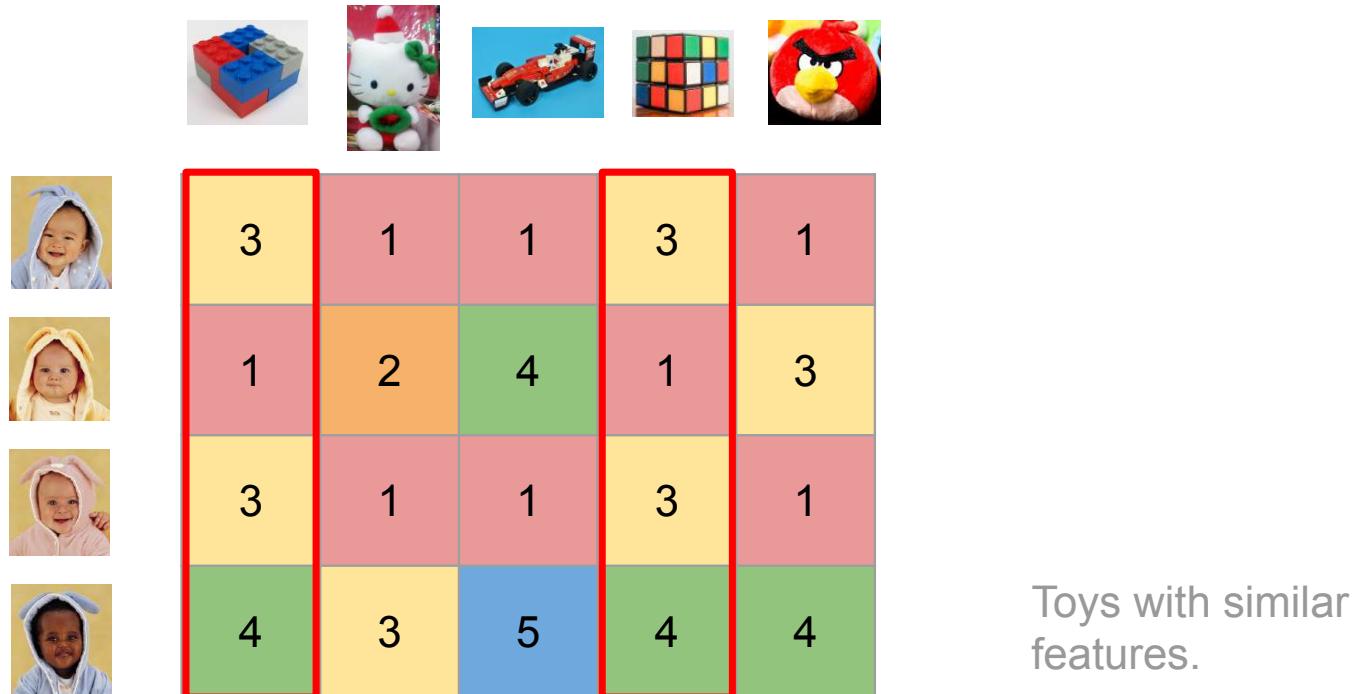
- one column of matrix \mathbf{A} is a scalar multiple of another column.
- one row of matrix \mathbf{A} is the sum of two other rows...

Redundancies in real data

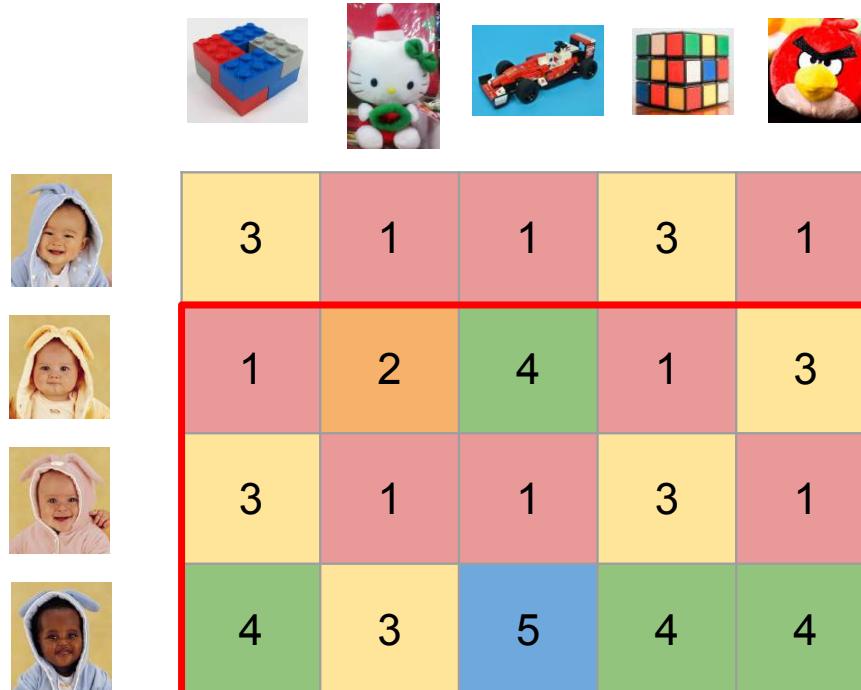
Luckily, there are many sources of redundancy in real data.



Redundancies in real data

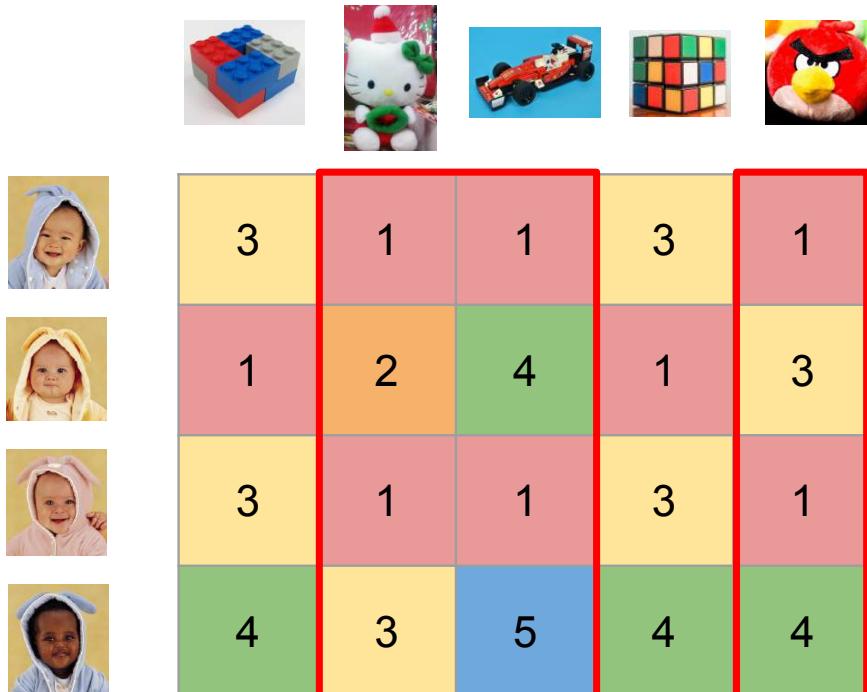


Redundancies in real data



Babies with "additive" preferences, e.g. one baby likes multiple features of a toy.

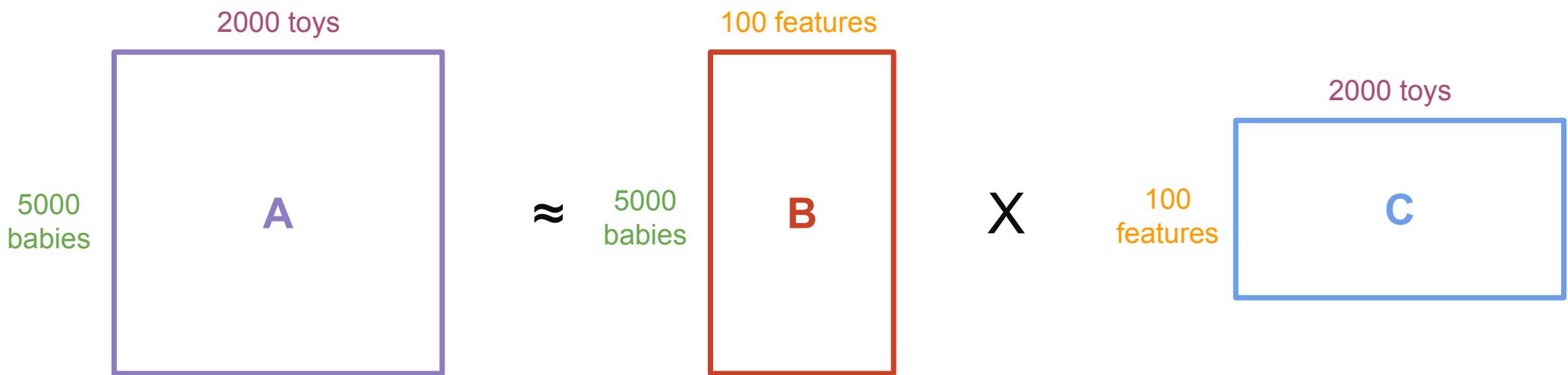
Redundancies in real data



Toys with "additive" features, e.g. one toy has different combinations of features from other toys.

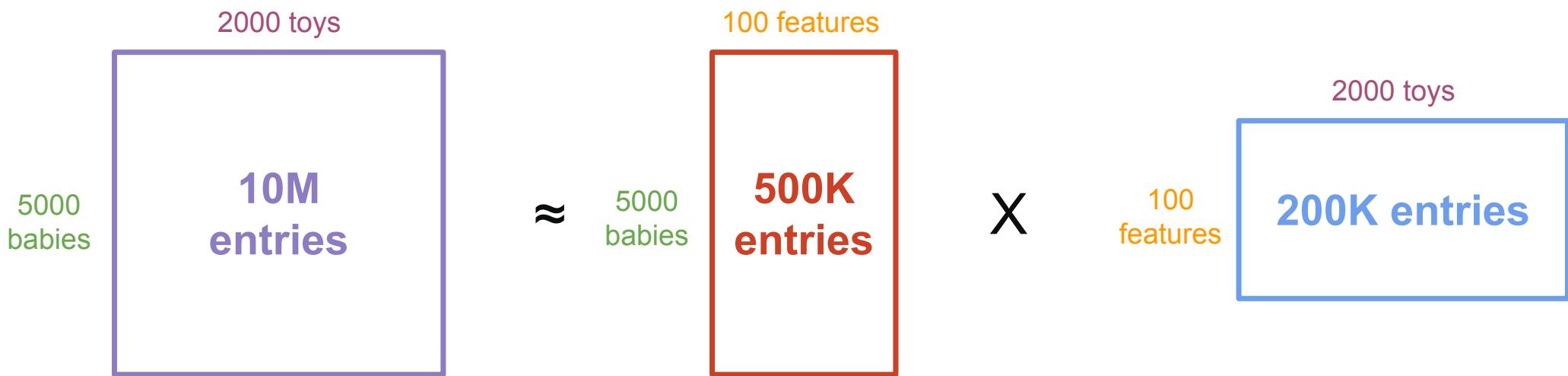
Compression via NMF: parameter reduction

Matrix visualization



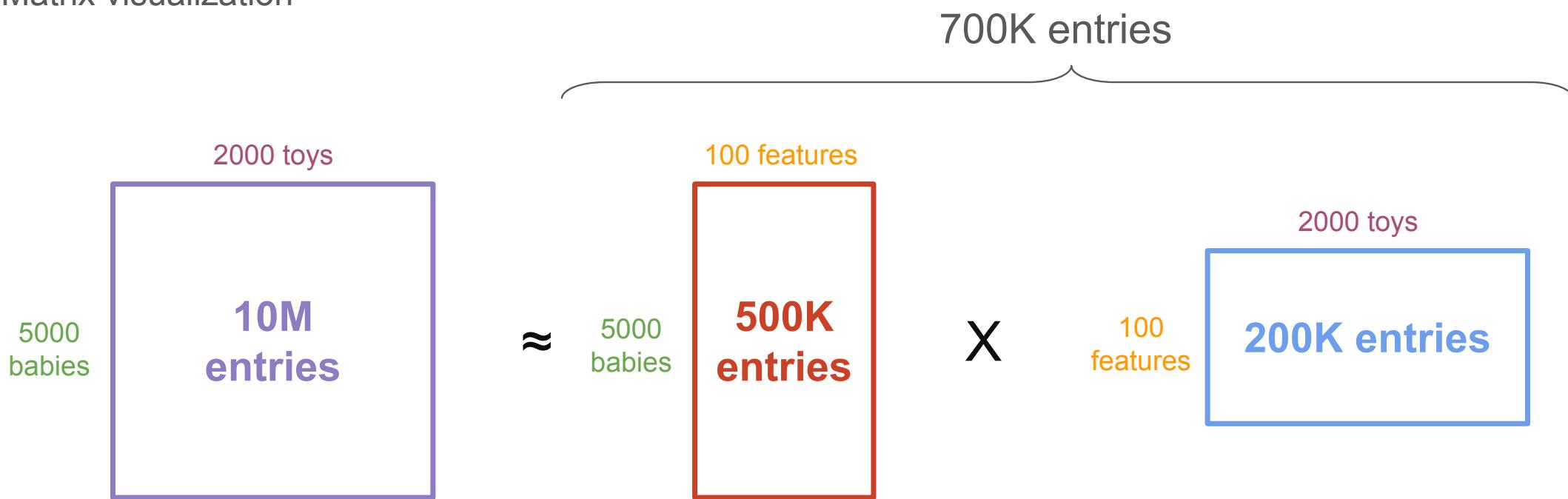
Compression via NMF: parameter reduction

Matrix visualization



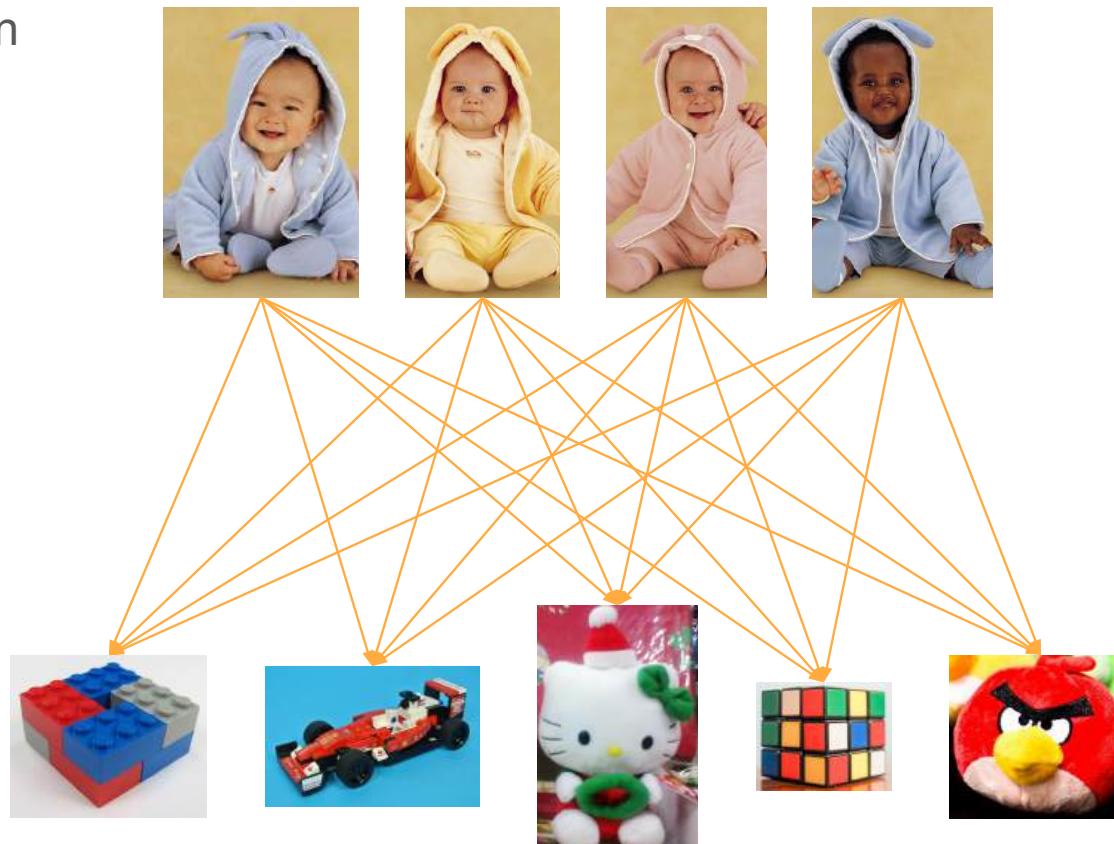
Compression via NMF: parameter reduction

Matrix visualization



Compression via NMF: parameter reduction

Graph visualization



Same math:

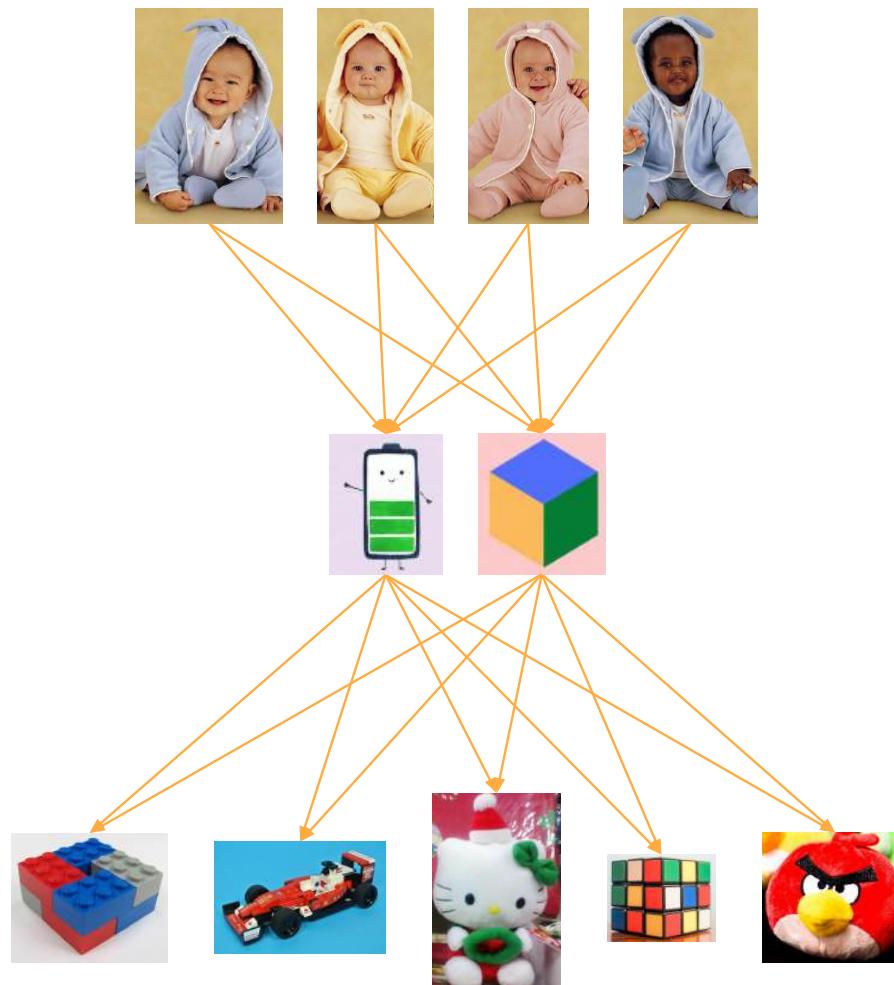
4 babies x 5 toys = 20 entries

Image source:

https://flickr.com/photos/yuchao_li/5936429322/, <https://flickr.com/photos/141802211@N05/33408806770>, <https://flickr.com/photos/56558327@N08/7423883676>,
<https://flickr.com/photos/oskay/2157692222>, <https://flickr.com/photos/aukirk/26858881315>, <https://flickr.com/photos/13698839@N00/6384317609>, <https://flickr.com/photos/omarriba/6702856409>

Compression via NMF: parameter reduction

Graph visualization



With 2 features:

4 babies x 2 features = 8 entries

2 features x 5 toys = 10 entries

Total 18 entries

Questions?

Example 1

Document-term
matrix

	LION	TIGER	CHEETAH	JAGUAR	PORSCHE	FERRARI
\bar{X}_1	2	2	1	2	0	0
\bar{X}_2	2	3	3	3	0	0
\bar{X}_3	1	1	1	1	0	0
\bar{X}_4	2	2	2	3	1	1
\bar{X}_5	0	0	0	1	1	1
\bar{X}_6	0	0	0	2	1	2

D

Figure 6.22: An example of non-negative matrix factorization

Example 1

Document-term matrix

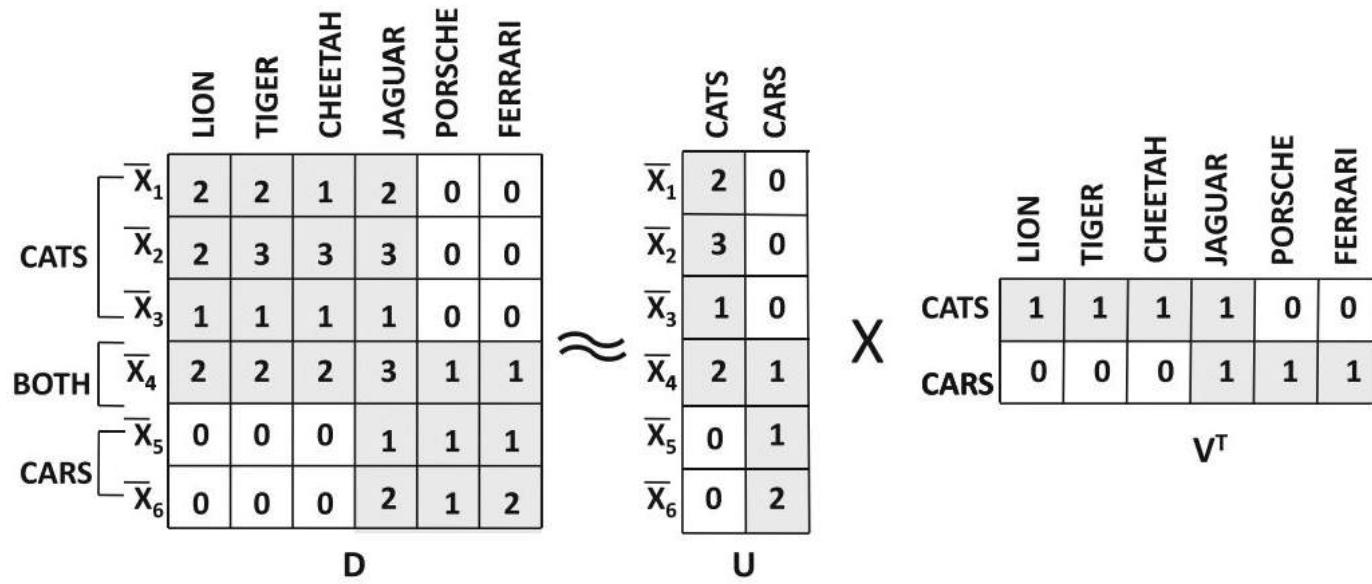


Figure 6.22: An example of non-negative matrix factorization

Example 1

Useful for:

- Interpretability
 - Detecting clusters

CATS		LION	TIGER	CHEETAH	JAGUAR	PORSCHE	FERRARI	
X ₁	2							
X ₂	3							
X ₃	1							
X ₄	2							
X ₅	0							
X ₆	0							

X CATS

=

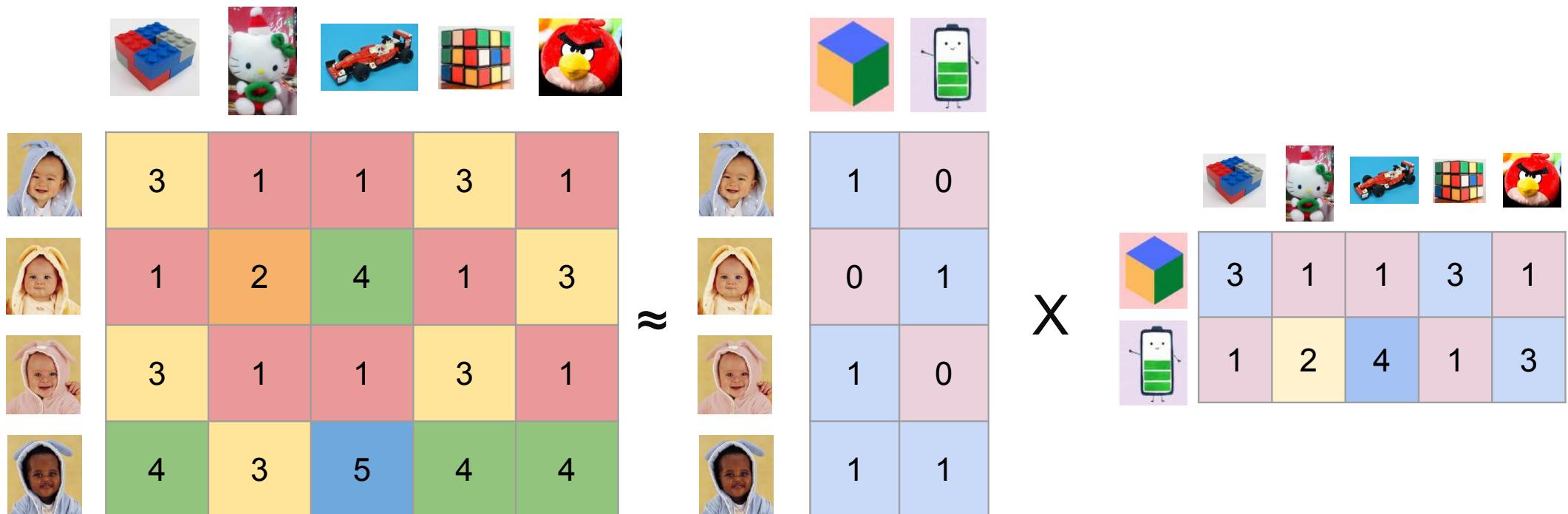
	LION	TIGER	CHEETAH	JAGUAR	PORSCHE	FERRARI	
X ₁	2	2	2	2	0	0	
X ₂	3	3	3	3	0	0	
X ₃	1	1	1	1	0	0	
X ₄	2	2	2	2	0	0	
X ₅	0	0	0	0	0	0	
X ₆	0	0	0	0	0	0	

LATENT COMPONENT (CATS)

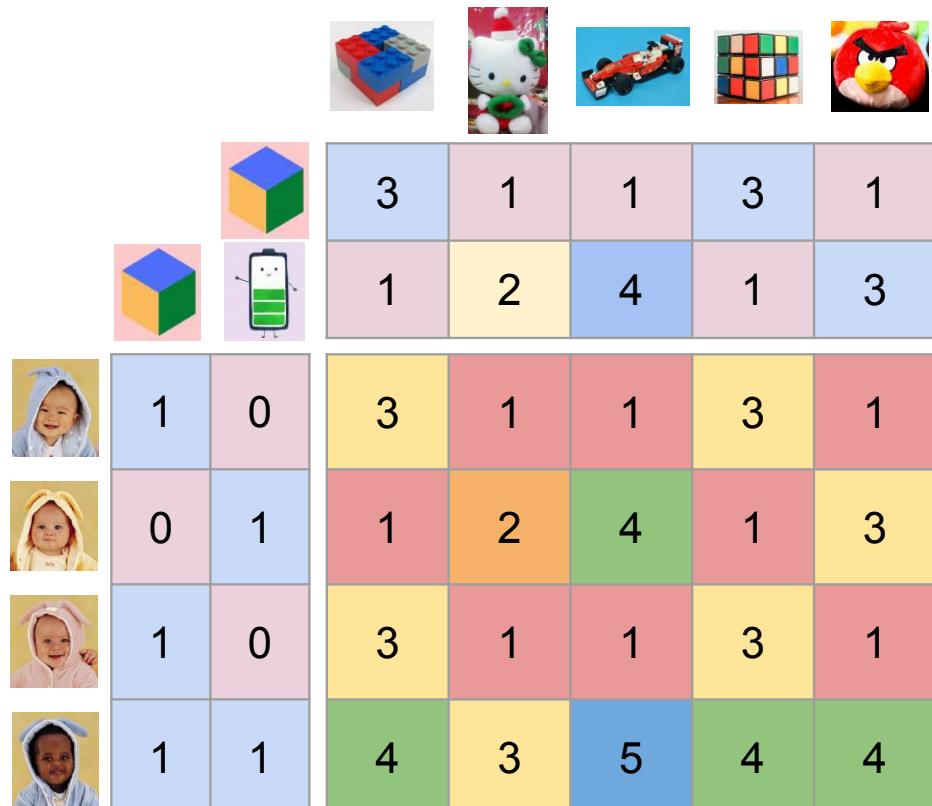
$$\begin{array}{c}
 \text{CARS} \\
 | \\
 X_1 \quad 0 \\
 | \\
 X_2 \quad 0 \\
 | \\
 X_3 \quad 0 \\
 | \\
 X_4 \quad 1 \\
 | \\
 X_5 \quad 1 \\
 | \\
 X_6 \quad 2
 \end{array}
 \times
 \begin{array}{c}
 \text{CARS} \\
 | \\
 \text{LION} \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1 \\
 | \\
 \text{TIGER} \\
 | \\
 \text{CHEETAH} \\
 | \\
 \text{JAGUAR} \\
 | \\
 \text{PORSCHE} \\
 | \\
 \text{FERRARI}
 \end{array}
 =
 \begin{array}{c}
 \text{LION} \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
 | \\
 \text{TIGER} \\
 | \\
 \text{CHEETAH} \\
 | \\
 \text{JAGUAR} \\
 | \\
 \text{PORSCHE} \\
 | \\
 \text{FERRARI}
 \end{array}$$

Questions?

Example 2



Example 2



NMF using gradient descent based algorithm,
random initialization in (0,1), no normalization:

Total reconstruction error: 1.47e-07

prediction C

0.69	1.38	2.76	0.69	2.07
1.93	0.35	0	1.93	0.18

prediction B

0.36	1.43	3.00003	0.99997	1.00014	3.00003	0.99995
1.45	0	1.00012	2.00006	3.99982	1.00012	3.00007
0.36	1.43	3.00003	0.99997	1.00014	3.00003	0.99995
1.81	1.42	3.99993	3.00005	5.00008	3.99993	4.00009

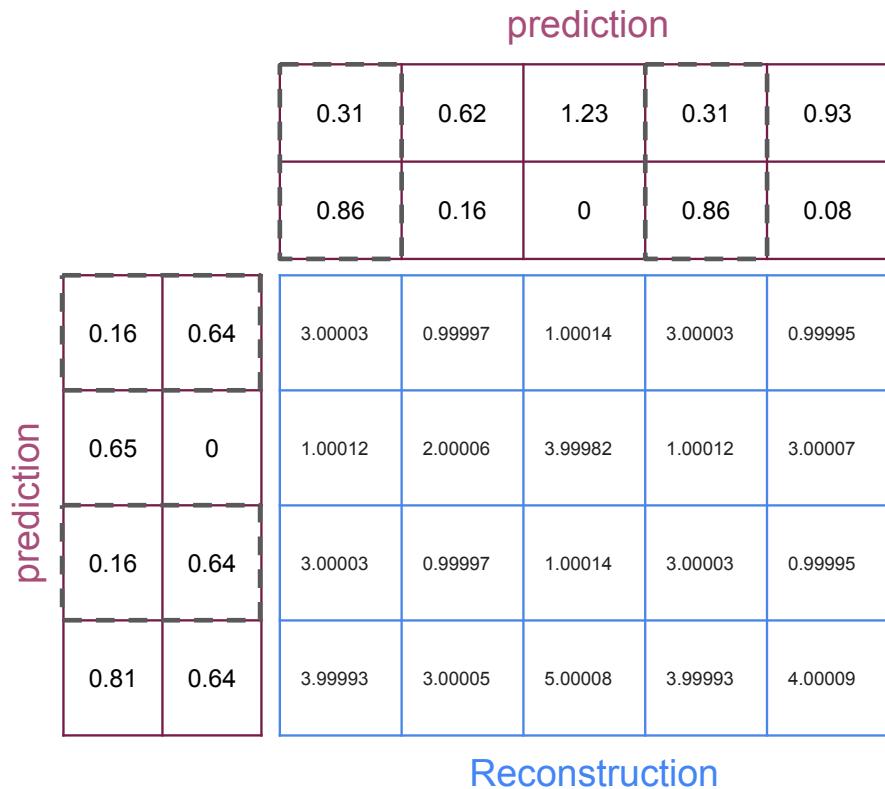
Reconstruction A

Example 2



NMF using gradient descent based algorithm, random initialization in $(0,1)$, same rng seed as previous example, **normalization**: divide by 5

Total reconstruction error: $1.47e-07$



Example 2



NMF using **multiplicative update** algorithm,
random initialization in (0,1), no normalization:

Total reconstruction error: 5.33e-10

prediction

3.02	0.86	0.68	3.02	0.77
0.63	1.73	3.56	0.63	2.65
0.97	0.09	3.00000	1.00000	1.00001
0.10	1.10	1.00001	2.00000	3.99999
0.97	0.09	3.00000	1.00000	1.00001
1.08	1.20	3.00000	3.00000	5.00000

Reconstruction

prediction

3.00000	1.00000	1.00001	3.00000	0.99999
1.00001	2.00000	3.99999	1.00001	3.00001
3.00000	1.00000	1.00001	3.00000	0.99999
4.00000	3.00000	5.00000	4.00000	4.00000

Example 2



NMF using **multiplicative update** algorithm,
random initialization in (0,1), no normalization.

Effect of using different number of components
(features "k"):

	Total reconstruction error
1 component	15.24
2 components	5.33e-10
3 components	2.76e-08
4 components	7.56e-05

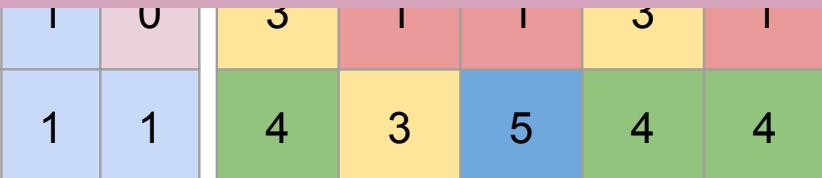
Perform cross-validation!

Example 2



Potential design choices:

- Normalization (yes/no?)
- Number of features k (for \mathbf{W} , \mathbf{H} initialization)
 - Error tolerance
 - Maximum iterations



NMF using **multiplicative update** algorithm,
random initialization in (0,1), no normalization.

Effect of using different number of components
(features "k"):

	Total reconstruction error
1 component	15.24
2 components	5.33e-10
3 components	2.76e-08
4 components	7.56e-05

Perform cross-validation!

Questions?

Extra information

Clustering:

- NMF is sometimes used for clustering.

"The technique of non-negative matrix factorization (NMF) [9, 7] has been used to cluster the document/word matrix into a non-negative set of basis vectors. Documents are then clustered by commonality with respect to the basis vectors used in the reconstruction."

Zass R, Shashua A. A unifying approach to hard and probabilistic clustering.
ICCV 2005.

- Certain types of NMF are equivalent to a relaxed form of K-means clustering.
- C. Ding, X. He, H.D. Simon. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. SIAM Int'l Conf. Data Mining 2005.

	LION	TIGER	CHEETAH	JAGUAR	PORSCHE	FERRARI
\bar{X}_1	2	2	2	2	0	0
\bar{X}_2	3	3	3	3	0	0
\bar{X}_3	1	1	1	1	0	0
\bar{X}_4	2	2	2	2	0	0
\bar{X}_5	0	0	0	0	0	0
\bar{X}_6	0	0	0	0	0	0

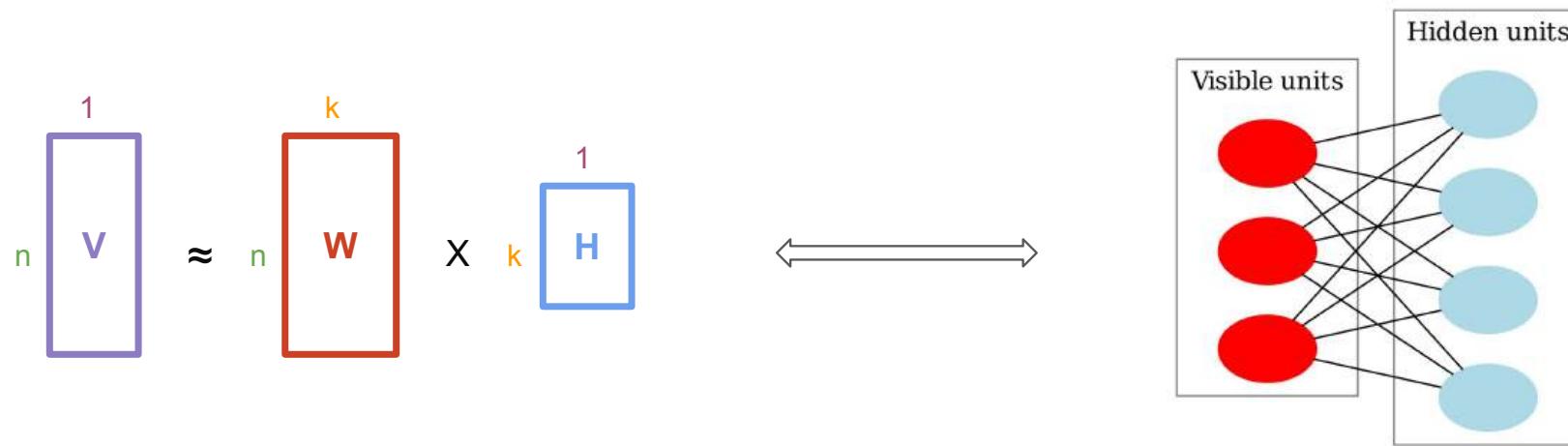
LATENT COMPONENT (CATS)

	LION	TIGER	CHEETAH	JAGUAR	PORSCHE	FERRARI
\bar{X}_1	0	0	0	0	0	0
\bar{X}_2	0	0	0	0	0	0
\bar{X}_3	0	0	0	0	0	0
\bar{X}_4	0	0	0	1	1	1
\bar{X}_5	0	0	0	1	1	1
\bar{X}_6	0	0	0	2	2	2

LATENT COMPONENT (CARS)

Extra information

Consider an $n \times 1$ matrix **V** to be factorized using NMF. The factors are $n \times k$ matrix **W** and $k \times 1$ matrix **H**.



This can be interpreted as a directed, probabilistic graph model with one layer of observed random variables (**V**) and one layer of hidden random variables (**H**) with weights **W**. (Naming convention.)

Data imputation

Data imputation

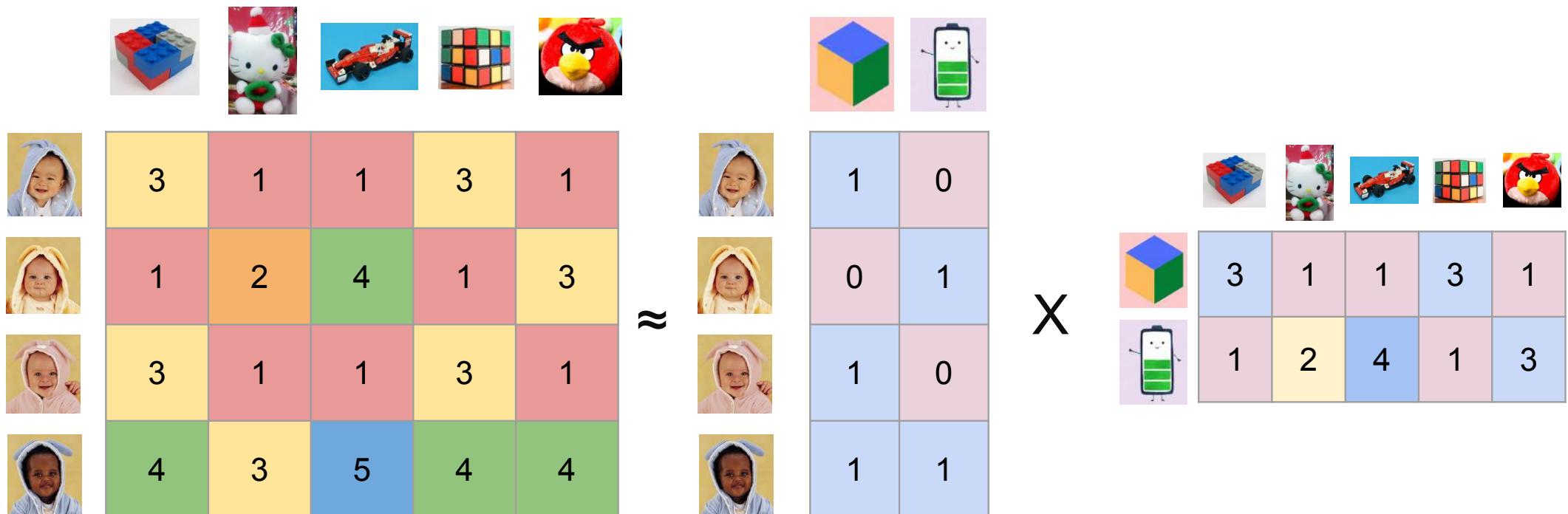
Question: How to find the missing values? (a.k.a. the 'data imputation' problem.)

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1	?		5		2
U_2		5			4	
U_3	5	3		1		
U_4		?	3		?	4
U_5	?			3	5	
U_6	5		4			?

(a) Ratings-based utility

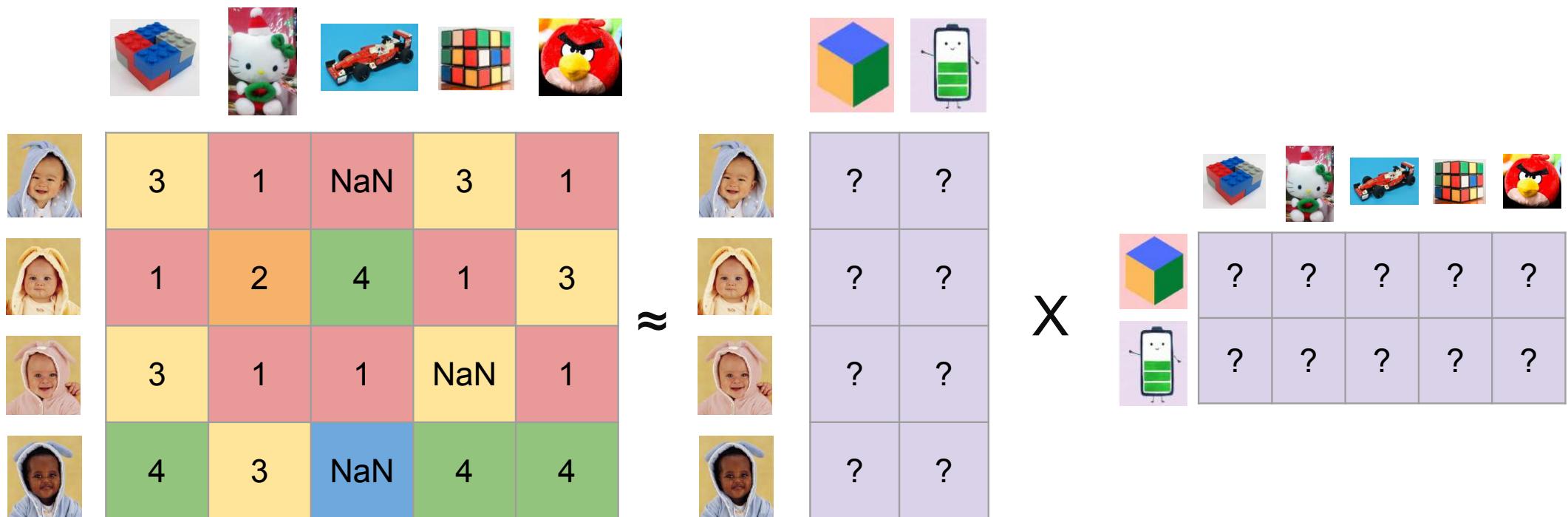
So far:

NMF with dense matrices...



Data imputation

Missing values?



Lab assignment!

Note:

For **part 1** of the lab assignment (i.e. Non-negative Matrix Factorization (NMF) implementation on WebLab): the multiplicative update algorithm given before is enough.

Lab assignment!

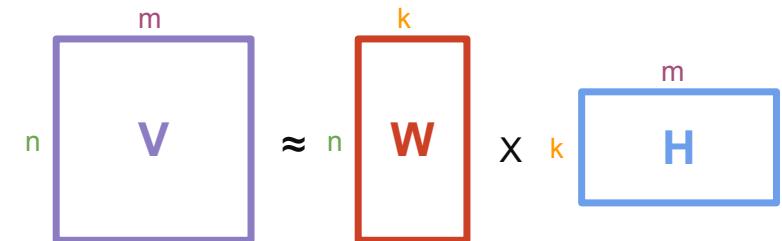
Note:

For **part 1** of the lab assignment (i.e. Non-negative Matrix Factorization (NMF) implementation on WebLab): the multiplicative update algorithm given before is enough.

For **part 4** of the lab assignment (NMF-based recommender system), performed on real-world data: Step 1 is to make sure your algorithm can handle NaNs, as described next!

How is it computed?

Algorithm: Multiplicative update in steps (lab assignment).



- 1) Initialize W and H randomly (all W_{ij} and H_{ij} drawn from a uniform distribution in the interval $(0,1)$).

Compute the reconstruction error $E = \|V - WH\|^2$. (treat missing values as "unknown" → no loss contribution!)

- 2) Individually update W and H to minimize $\|V - WH\|^2$ using (treat missing values like 0s → no contribution to matmul!)

$$W_{ij} \leftarrow W_{ij} (VH^T)_{ij} / ((WHH^T)_{ij} + \epsilon)$$

$$H_{ij} \leftarrow H_{ij} (W^T V)_{ij} / ((W^T WH)_{ij} + \epsilon)$$

Make sure there is no division by 0 (can use an ϵ term).

- 3) Compute the new reconstruction error $E_{\text{new}} = \|V - WH\|^2$ (treat missing values as "unknown" → no loss contribution!)

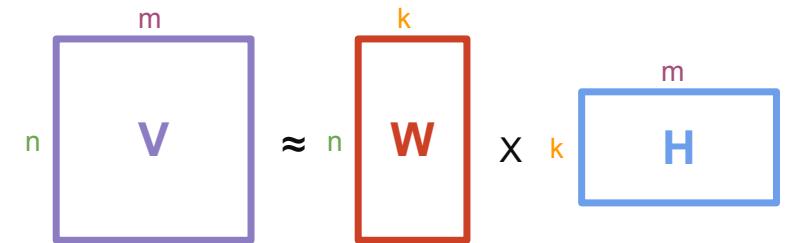
- 4) Stop updating and end optimization if $E - E_{\text{new}} <$ a predefined error tolerance.

- 5) While $(E - E_{\text{new}})$ isn't small enough, repeat steps 2-4 for a predefined number of maximum iterations.

How is it computed?

Changes to "standard" NMF:

- 1) Compute the reconstruction error $E = \|\mathbf{V} - \mathbf{WH}\|^2$. (treat missing values as "unknown" → no loss contribution!)



How is it computed?

Changes to "standard" NMF:

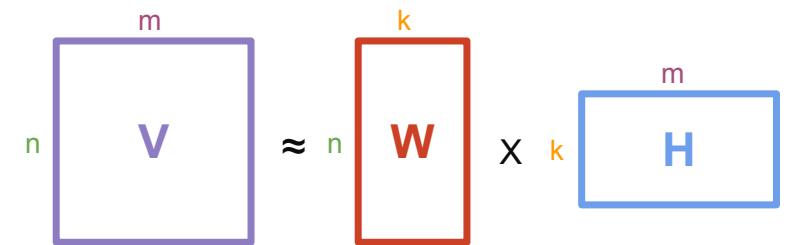
1) Compute the reconstruction error $E = \|\mathbf{V} - \mathbf{WH}\|^2$. (treat missing values as "unknown" → no loss contribution!)

- For example create a binary mask: `E = np.sum((V - np.matmul(W, H))**2 * mask_train)`

with

`mask_train =`

1	1	0	1	1
1	1	1	1	1
1	1	1	0	1
1	1	0	1	1



How is it computed?

Changes to "standard" NMF:

1) Compute the reconstruction error $E = \|\mathbf{V} - \mathbf{WH}\|^2$. (treat missing values as "unknown" → no loss contribution!)

- For example create a binary mask: `E = np.sum((V - np.matmul(W, H))**2 * mask_train)`

with

`mask_train =`

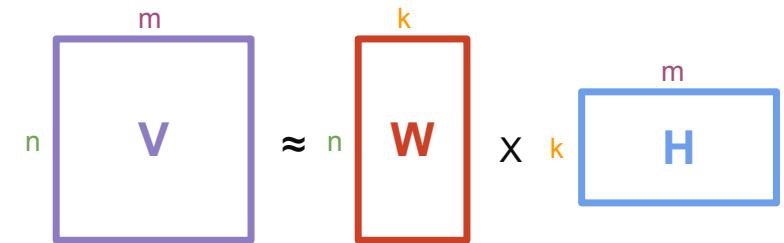
1	1	0	1	1
1	1	1	1	1
1	1	1	0	1
1	1	0	1	1

- Or directly use the utility matrix with NaNs: `E = np.nansum((V - np.matmul(W, H))**2)`

with

`V =`

3	1	np.nan	3	1
1	2	4	1	3
3	1	1	np.nan	1
4	3	np.nan	4	4



How is it computed?

Changes to "standard" NMF:

- 2) Individually update **W** and **H** (treat missing values like 0s→no contribution to matrix multiplication!)

$$\underset{n}{\underset{m}{\text{V}}} \approx \underset{n}{\underset{k}{\text{W}}} \times \underset{k}{\underset{m}{\text{H}}}$$

How is it computed?

Changes to "standard" NMF:

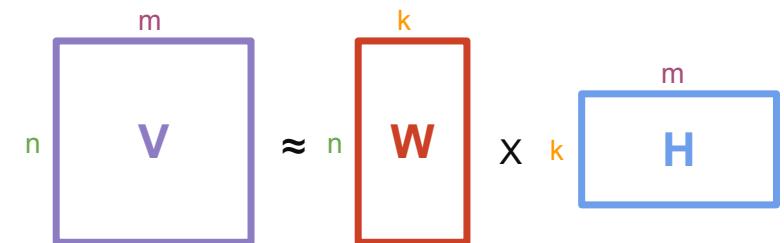
2) Individually update **W** and **H** (treat missing values like 0s→no contribution to matrix multiplication!)

- Recommended solution (also for computing the loss):

Use the numpy masked array module (`np.ma`):

```
>>> a = np.ma.array([[1, 2, 3], [4, 5, 6]], mask=[[1, 0, 0], [0, 0, 0]])
>>> b = np.ma.array([[1, 2], [3, 4], [5, 6]], mask=[[1, 0], [0, 0], [0, 0]])
>>> np.ma.dot(a, b)

masked_array(
  data=[[21, 26],
        [45, 64]])
```



Next time

- Recommender systems:
 - Collaborative filtering
 - Filling in "missing values" using NMF
 - Cross-validation



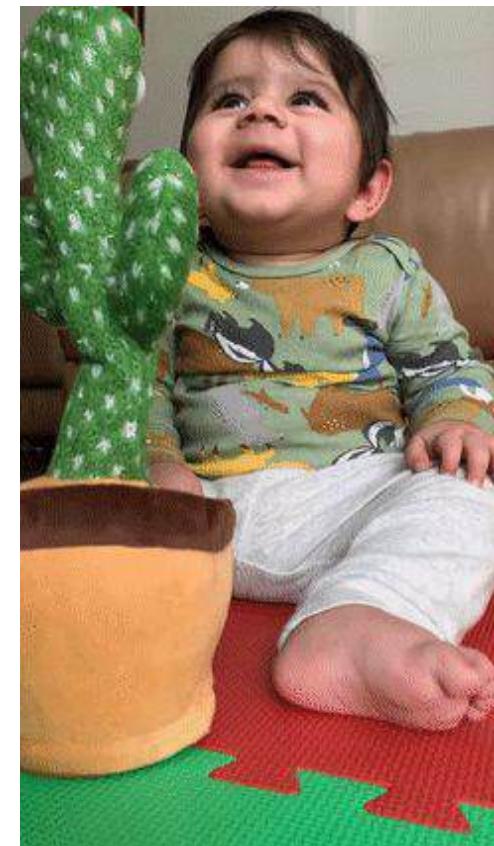
Next time

- Recommender systems:
 - Collaborative filtering
 - Filling in "missing values" using NMF
 - Cross-validation

	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
1	✓		✓	✓	
2		✓			✓
3	✓	✓	✓		
4				✓	✓

Icons representing users:

- User 1: Girl with blue hair and red hat
- User 2: Girl with dark skin and black hair
- User 3: Girl with orange hair and glasses
- User 4: Girl with blonde hair and grey hat



References / Recommended reading

- [1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [2] L. van der Maaten, and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research* 9, no. 11 (2008).
- [3] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint*, arXiv:1802.03426 (2018).
- [4] G. Hinton, and S. Roweis, "Stochastic neighbor embedding," NeurIPS 2002.
- [5] <https://www.youtube.com/watch?v=6BPI81wGGP8>
- [6] Vavasis SA. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*. 2010;20(3):1364-77.
<https://arxiv.org/abs/0708.4149>
- [7] Lee D, Seung HS. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*. 2000;13.

Recommender Systems



Data Mining CSE2525

Nergis Tomen

09.01.2025

Overview

- Recommender systems
 - Collaborative filtering & the utility matrix
 - NMF for data imputation
 - Cross-validation
 - Performance metrics
 - Privacy
 - Alternatives to NMF

Recommender systems

Consider a streaming service:

Simple picture → 4 **subscribers**, 5 **movies**... technically can recommend all movies to all subscribers.

	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
Subscribers					
Subscribing	✓		✓	✓	
Subscribing		✓			✓
Subscribing	✓	✓	✓		
Subscribing				✓	✓

Recommender systems

Consider a streaming service:

Real picture → 200 million **subscribers**, 10,000 movies.

	Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento
1	✓		✓	✓	
2		✓			✓
3	✓	✓	✓		
4				✓	✓
⋮					

Recommender systems: Data imputation

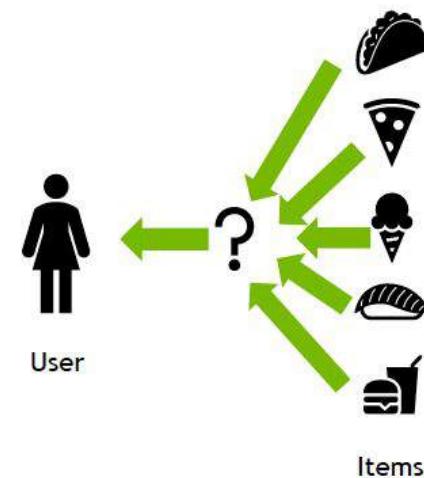
How to find the missing values? (a.k.a. the 'data imputation' problem.)

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1	?		5		2
U_2		5			4	
U_3	5	3		1		
U_4		?	3		?	4
U_5	?			3	5	
U_6	5		4			?

(a) Ratings-based utility

Recommender systems

The aim of a recommender system is to **provide suggestions** for **items** that are most likely to be interesting to a **user**.



Recommender systems

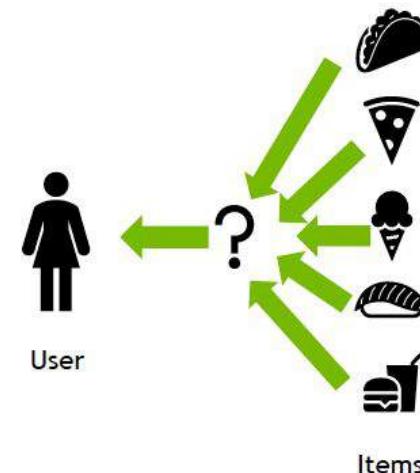
The aim of a recommender system is to **provide suggestions** for **items** that are most likely to be interesting to a **user**.

Providers:

- have thousands/millions of items on offer

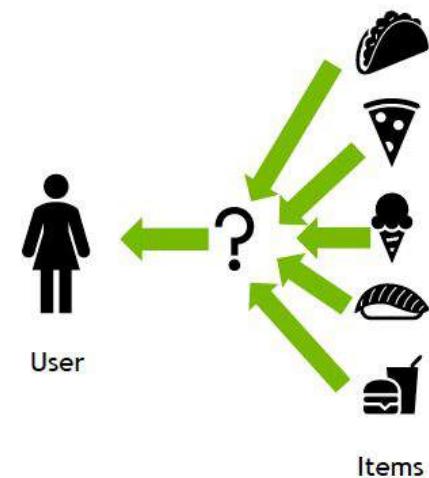
Users:

- have limited time
- have limited budget
- cannot watch, buy, eat everything
- may want to discover something new



Recommender systems

What are some examples of recommender systems you know?



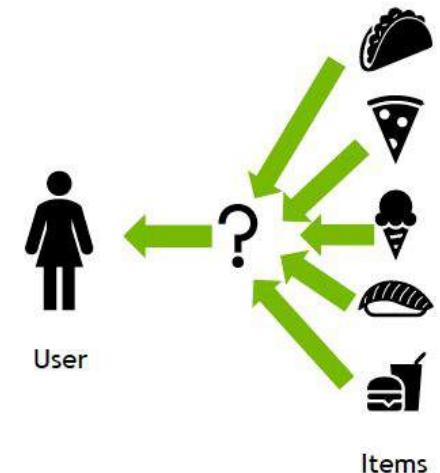
Recommender systems

Used for recommending, for example:

- **Movies, shows, music** in streaming services
- **Products** to purchase in e-commerce



Shrek



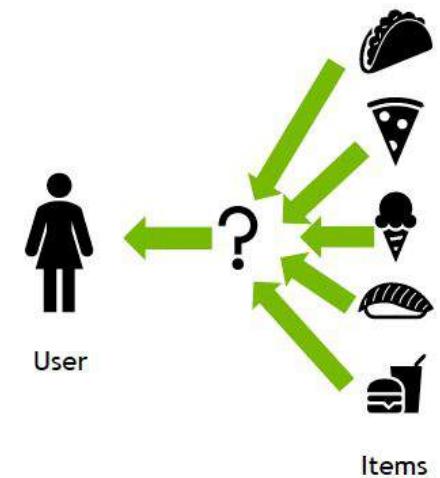
Recommender systems

Used for recommending, for example:

- **Movies, shows, music** in streaming services
- **Products** to purchase in e-commerce
- Articles to read in online news sites
- **Content** in social media
- **Books** to buy or audiobooks to listen to



Shrek



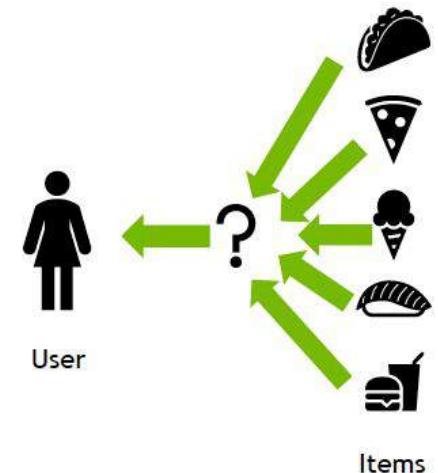
Recommender systems

Used for recommending, for example:

- **Movies, shows, music** in streaming services
- **Products** to purchase in e-commerce
- Articles to read in online news sites
- **Content** in social media
- **Books** to buy or audiobooks to listen to
- Restaurants in a user's vicinity
- Potential **dates** in online dating
- etc...



Shrek



Recommender systems

Used for recommending, for example:

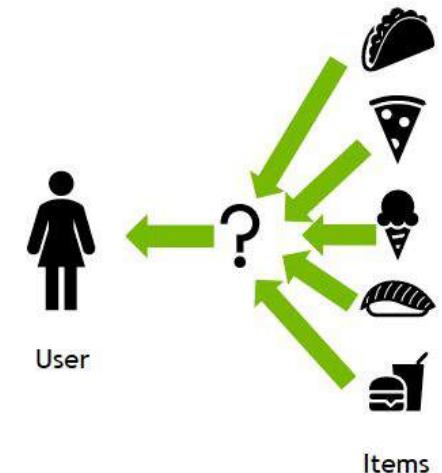
- **Movies, shows, music** in streaming services
- **Products** to purchase in e-commerce
- Articles to read in online news
- **Content** in social media
- **Books** to buy or audiobooks
- Restaurants in a user's vicinity
- Potential **dates** in online dating
- etc...

2 popular approaches:

- Collaborative filtering
- Content-based filtering

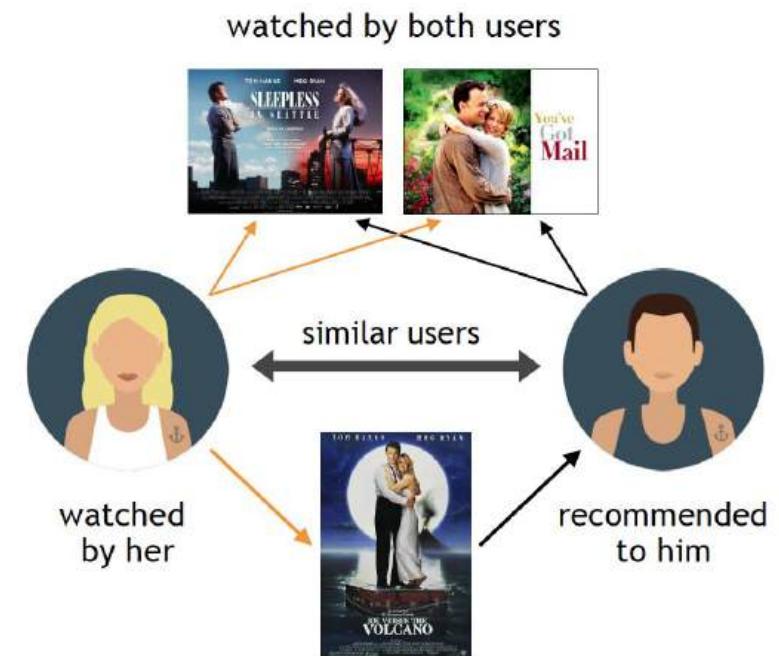


Shrek



Collaborative filtering

"Collaborative filtering, is the leveraging of **user preferences** in the form of ratings or buying behavior in a "collaborative" way, ... to determine either **relevant users** for specific items, or **relevant items** for specific users in the recommendation process."



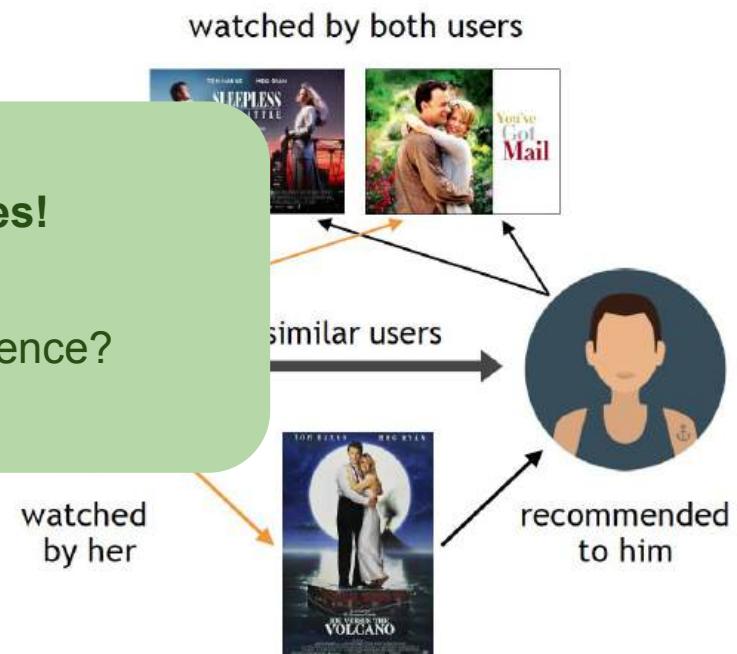
Data Mining the Textbook, Chapter 18.5

Collaborative filtering

"Collaborative filtering, is the leveraging of **user preferences** in the form of ratings or buying behavior in a "collaborative" environment to determine either **relevant items**, or **relevant items** for the recommendation.

This is what NMF does!

Remember linear dependence?



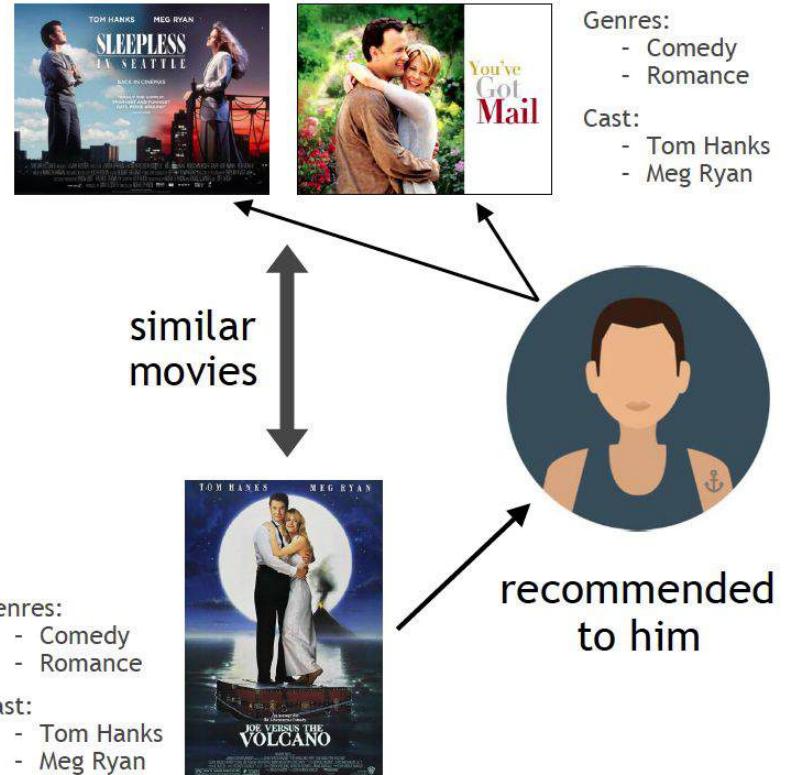
Data Mining the Textbook, Chapter 18.5

Content-based filtering

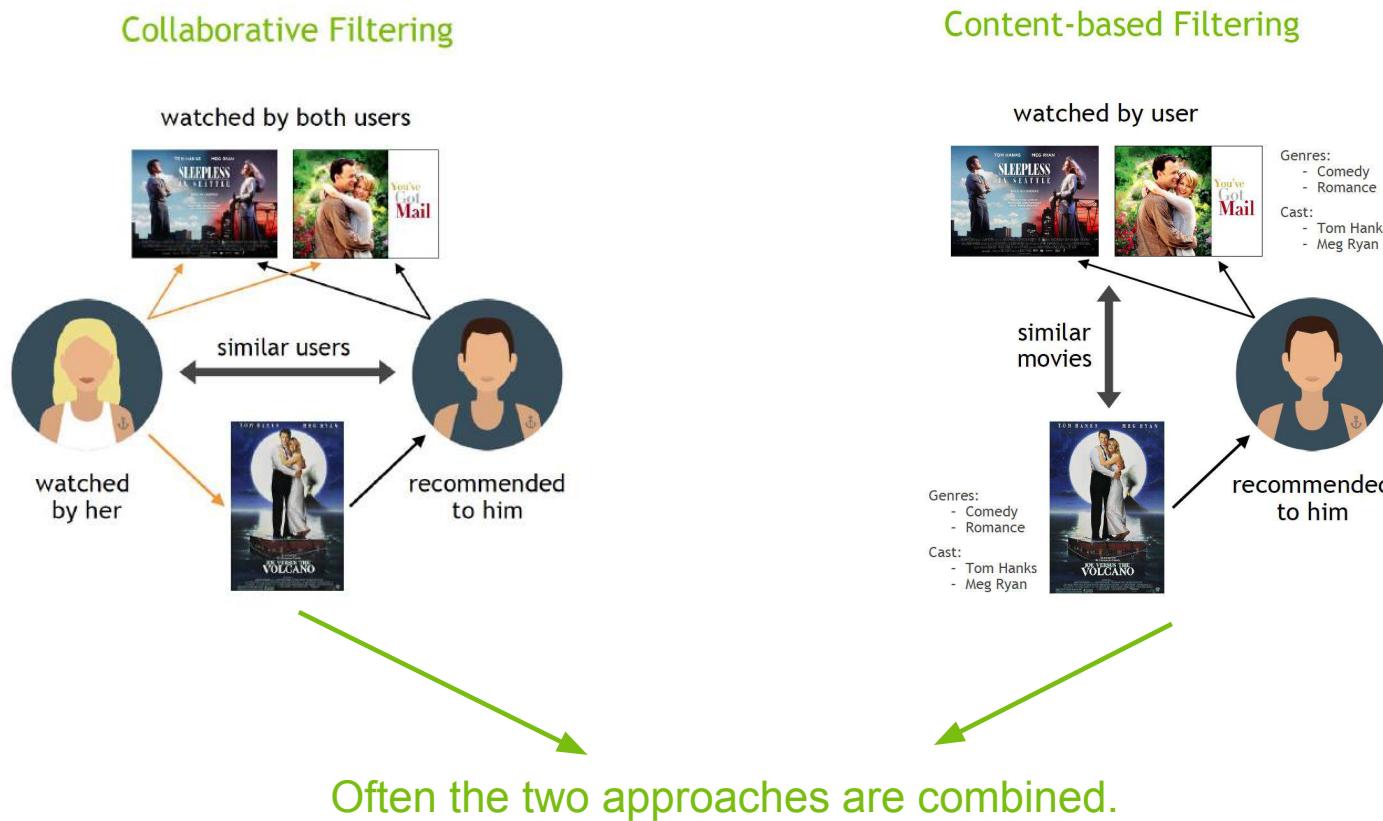
"**Users** and **items** are both associated with **feature-based descriptions**. For example, ... the text of the item description (meta-data). A user might also have explicitly specified their interests in their profile (knowledge-based)... or can be **inferred from** their buying or browsing **behavior**."

Data Mining the Textbook, Chapter 18.5

watched by user



Recommender systems



Questions?

Long-tail phenomenon of online recommender systems

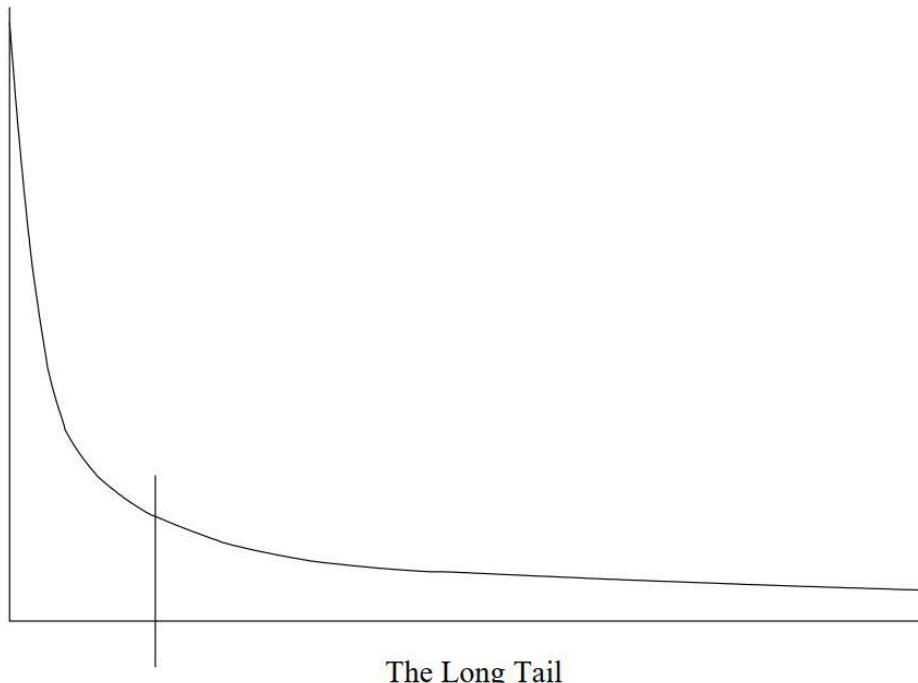


Figure 9.2: The long tail: physical institutions can only provide what is popular, while on-line institutions can make everything available

Long-tail phenomenon of online recommender systems

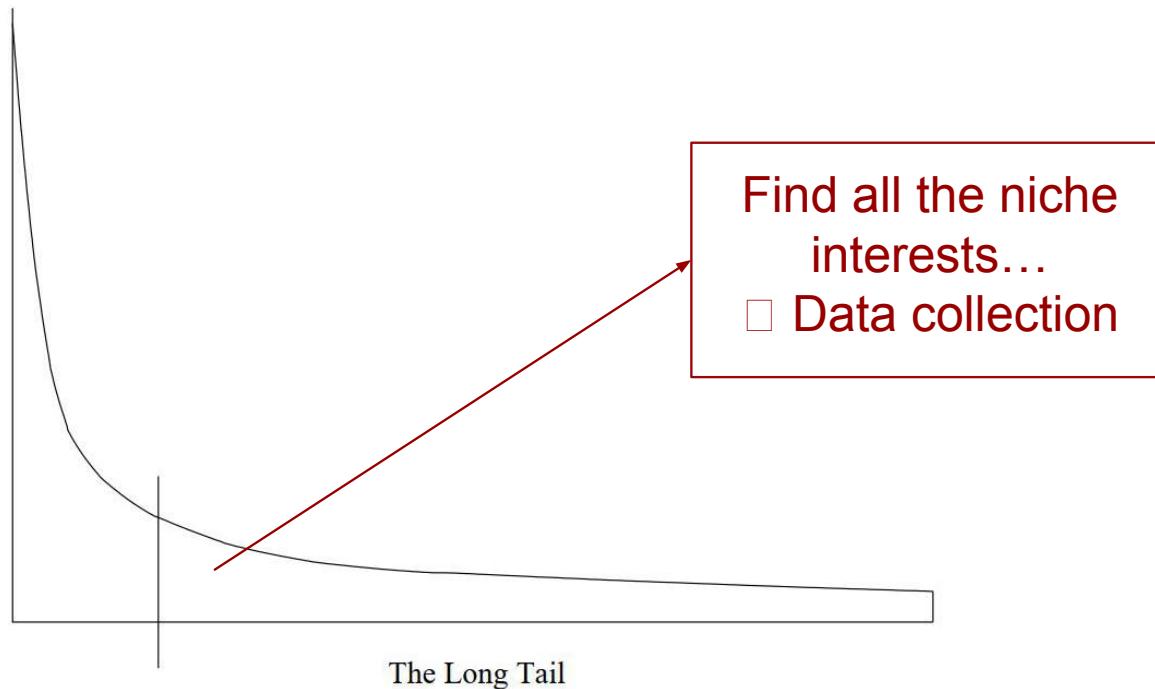


Figure 9.2: The long tail: physical institutions can only provide what is popular, while on-line institutions can make everything available

Utility matrix

Online services collect an increasingly **large amount of information** about user behaviour!

There are 2 main types of utility matrices.

Utility matrix

There are 2 main types of utility matrices.

		Items						
		GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS	
		U ₁	1			5		2
		U ₂		5			4	
		U ₃	5	3		1		
		U ₄			3			4
		U ₅				3	5	
		U ₆	5		4			

(a) Ratings-based utility

		Items						
		GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS	
		U ₁	1			1		1
		U ₂		1			1	
		U ₃	1	1		1		
		U ₄			1			1
		U ₅				1	1	
		U ₆	1		1			

(b) Positive-preference utility

Utility matrix

There are 2 main types of utility matrices.

Users gave ratings (explicitly collected)



		Items					
		GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
Users	U_1	1			5		2
	U_2		5			4	
U_3	5	3			1		
U_4				3			4
U_5					3	5	
U_6	5		4				

(a) Ratings-based utility

		Items					
		GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
Users	U_1	1			1		1
	U_2		1			1	
U_3	1	1			1		
U_4				1			1
U_5					1	1	
U_6	1		1				

(b) Positive-preference utility

Users watched a movie (implicitly collected)



Utility matrix

There are 2 main types of utility matrices.

Which one do you think is better?

Users gave ratings (explicitly collected)

Users

		Items					
		GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
Users	U ₁	1			5		2
	U ₂		5			4	
U ₃	5	3			1		
U ₄				3			4
U ₅					3	5	
U ₆	5		4				

(a) Ratings-based utility

		GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
Users	U ₁	1			1		1
U ₂		1				1	
U ₃	1	1			1		
U ₄				1			1
U ₅					1	1	
U ₆	1		1				

(b) Positive-preference utility

Users watched a movie (implicitly collected)

Utility matrix

There are 2 main types of utility matrices.

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

(a) Ratings-based utility

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1				1	1
U_2		1			1	
U_3	1	1			1	
U_4			1			1
U_5				1	1	
U_6	1		1			

(b) Positive-preference utility

Utility matrix

There are 2 main types of utility matrices.

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			5		2
U_2		5			4	
U_3	5	3		1		
U_4			3			4
U_5				3	5	
U_6	5		4			

Might be harder to collect

(a) Ratings-based utility

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1			1		1
U_2		1			1	
U_3	1	1		1		
U_4			1			1
U_5				1	1	
U_6	1		1			

Might be harder to work with

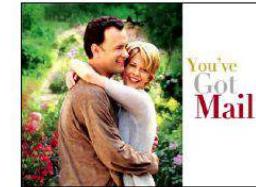
(b) Positive-preference utility

Collaborative filtering

Advantages:

- Doesn't rely on hand-crafted or machine-extracted (e.g. from text) 'content' (both prone to errors)

watched by user



Genres:

- Comedy
- Romance

Cast:

- Tom Hanks
- Meg Ryan

Collaborative filtering

Advantages:

- Doesn't rely on hand-crafted or machine-extracted (e.g. from text) '**content**' (both prone to errors)
- Can work with **implicitly** collected data (e.g. user watched a movie)
- Can work with **explicitly** collected data (e.g. user rated a movie)

Collaborative filtering

Advantages:

- Doesn't rely on hand-crafted or machine-extracted (e.g. from text) '**content**' (both prone to errors)
- Can work with **implicitly** collected data (e.g. user watched a movie)
- Can work with **explicitly** collected data (e.g. user rated a movie)

Disadvantages:

- Needs many non-unique (e.g. similar, linearly dependent) users which interact with many different items.
- **Cold start:** Hard to find recommendations for 'new' items or users with little past information or activity.
- **Scalability:** Big data needs big processing power.
- **Sparsity:** Thousands of items, each user interacts with only a few.

Questions?

Collaborative filtering using NMF

Question: How to find the missing values? (a.k.a. the 'data imputation' problem.)

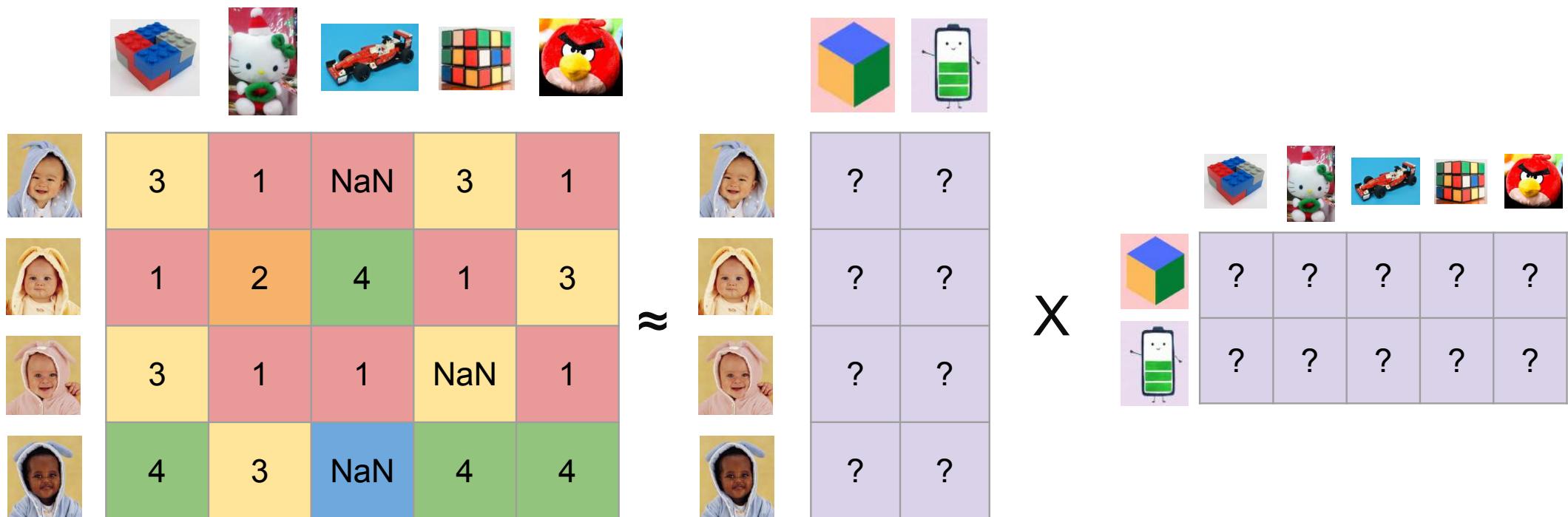
Use NMF!

	GLADIATOR	GODFATHER	BEN-HUR	GOODFELLAS	SCARFACE	SPARTACUS
U_1	1	?		5		2
U_2		5			4	
U_3	5	3		1		
U_4		?	3		?	4
U_5	?			3	5	
U_6	5		4			?

(a) Ratings-based utility

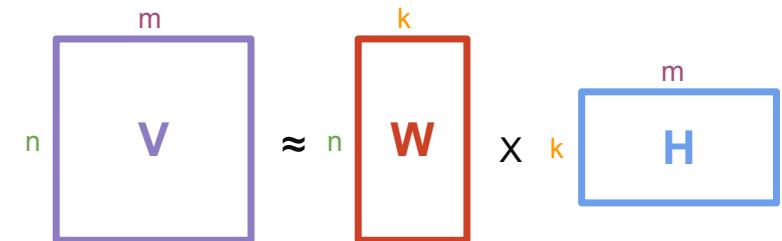
Utility matrix

Missing values?



How is it computed?

Algorithm: Multiplicative update in steps (lab assignment).



- 1) Initialize W and H randomly (all W_{ij} and H_{ij} drawn from a uniform distribution in the interval $(0,1)$).

Compute the reconstruction error $E = \|V - WH\|^2$. (treat missing values as "unknown" → no loss contribution!)

- 2) Individually update W and H to minimize $\|V - WH\|^2$ using (treat missing values like 0s → no contribution to matmul!)

$$W_{ij} \leftarrow W_{ij} (VH^T)_{ij} / ((WHH^T)_{ij} + \epsilon)$$

$$H_{ij} \leftarrow H_{ij} (W^T V)_{ij} / ((W^T WH)_{ij} + \epsilon)$$

Make sure there is no division by 0 (can use an ϵ term).

- 3) Compute the new reconstruction error $E_{\text{new}} = \|V - WH\|^2$ (treat missing values as "unknown" → no loss contribution!)

- 4) Stop updating and end optimization if $E - E_{\text{new}} <$ a predefined error tolerance.

- 5) While $(E - E_{\text{new}})$ isn't small enough, repeat steps 2-4 for a predefined number of maximum iterations.

How is it computed?

Changes to "standard" NMF:

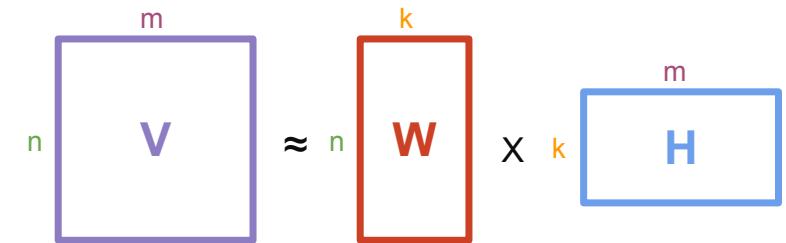
- 1) Compute the reconstruction error $E = \|\mathbf{V} - \mathbf{WH}\|^2$. (treat missing values as "unknown"→no loss contribution!)
- 2) Individually update \mathbf{W} and \mathbf{H} (treat missing values as unknown/0s→no contribution to matrix multiplication!)

- Recommended solution (also for computing the loss):

Use the numpy masked array module (`np.ma`):

```
>>> a = np.ma.array([[1, 2, 3], [4, 5, 6]], mask=[[1, 0, 0], [0, 0, 0]])
>>> b = np.ma.array([[1, 2], [3, 4], [5, 6]], mask=[[1, 0], [0, 0], [0, 0]])
>>> np.ma.dot(a, b)

masked_array(
  data=[[21, 26],
        [45, 64]])
```



NMF example from last time



NMF using multiplicative update algorithm,
random initialization in (0,1), no normalization:

Total reconstruction error: 5.33e-10

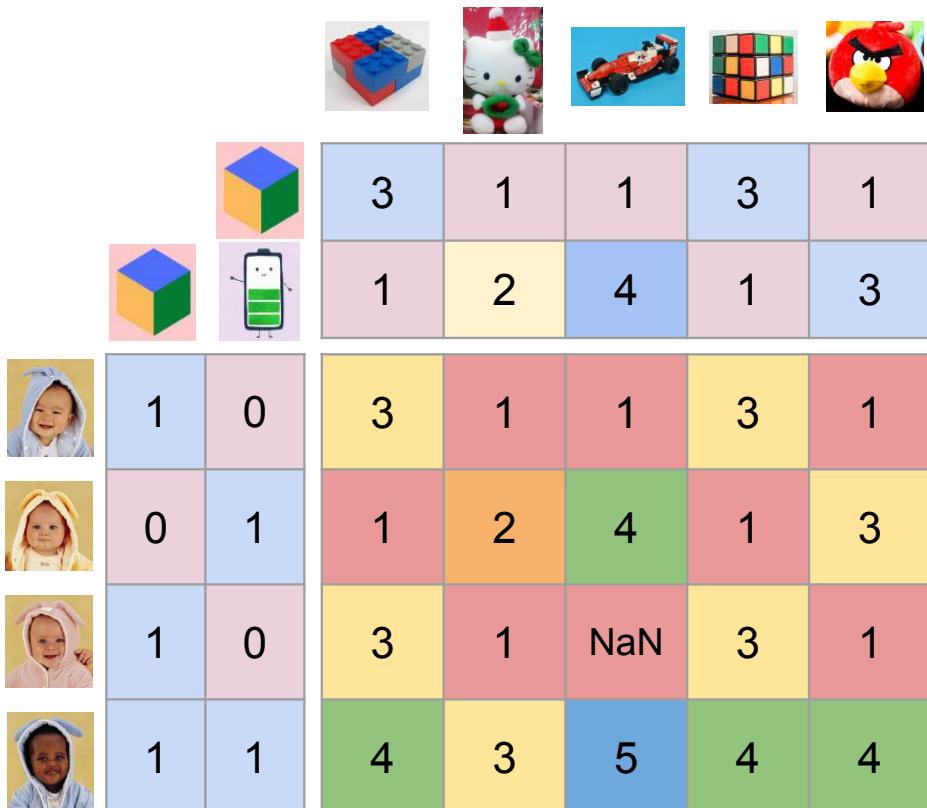
prediction

3.02	0.86	0.68	3.02	0.77
0.63	1.73	3.56	0.63	2.65
0.97	0.09			
0.10	1.10			
0.97	0.09			
1.08	1.20			
3.00000	1.00000	1.00001	3.00000	0.99999
1.00001	2.00000	3.99999	1.00001	3.00001
3.00000	1.00000	1.00001	3.00000	0.99999
4.00000	3.00000	5.00000	4.00000	4.00000

Reconstruction

prediction

Example 1, 1 missing value



NMF using **multiplicative update** algorithm,
random initialization in (0,1), no normalization:

Reconstruction error known data: 6.29e-07
Reconstruction error missing data: 1.67e-06

prediction

3.48	0.86	0.50	3.48	0.68
0.61	1.34	2.70	0.61	2.02
0.82	0.219			
0.03	1.48			
0.82	0.22			
0.85	1.70			
3.00004	0.99979	1.00035	3.00005	0.99960
0.99997	2.00012	3.99976	0.99997	3.00024
2.99993	1.00020	1.00129	2.99993	1.00028
4.00003	2.99992	5.00012	4.00003	3.99984

Reconstruction

prediction

Example 2, 3 missing values



NMF using multiplicative update algorithm,
random initialization in (0,1), no normalization:

Reconstruction error known data: 2.76e-08
Reconstruction error missing data: **12.15**

prediction

3.15	0.58	2.32	3.15	0.30
0.72	1.70	3.34	0.72	2.57
0.89	0.29			
0.09	1.16			
0.89	0.29			
0.94	1.45			

Reconstruction

3.00000	0.99994	3.01226	3.00000	1.00006
1.00006	2.00005	4.00000	1.00005	2.99993
3.00000	0.99994	3.01226	3.00000	1.00006
3.99999	3.00001	7.01226	3.99998	4.00002

Example 2, 3 missing values



Sparsity problem...

NMF using multiplicative update algorithm,
random initialization in (0,1), no normalization:

Reconstruction error known data: 2.76e-08

Reconstruction error missing data: **12.15**

prediction

3.15	0.58	2.32	3.15	0.30
0.72	1.70	3.34	0.72	2.57

prediction

0.89	0.29
0.09	1.16
0.89	0.29
0.94	1.45

Reconstruction

3.00000	0.99994	3.01226	3.00000	1.00006
1.00006	2.00005	4.00000	1.00005	2.99993
3.00000	0.99994	3.01226	3.00000	1.00006
3.99999	3.00001	7.01226	3.99998	4.00002

Example 3, missing column

		Lego	Hello Kitty	Car	Rubik's Cube	Angry Bird
Block	3	1	1	3	1	
Battery	1	2	4	1	3	
Child 1	1	0	3	1	NaN	3
Child 2	0	1	1	2	NaN	1
Child 3	1	0	3	1	NaN	3
Child 4	1	1	4	3	NaN	4

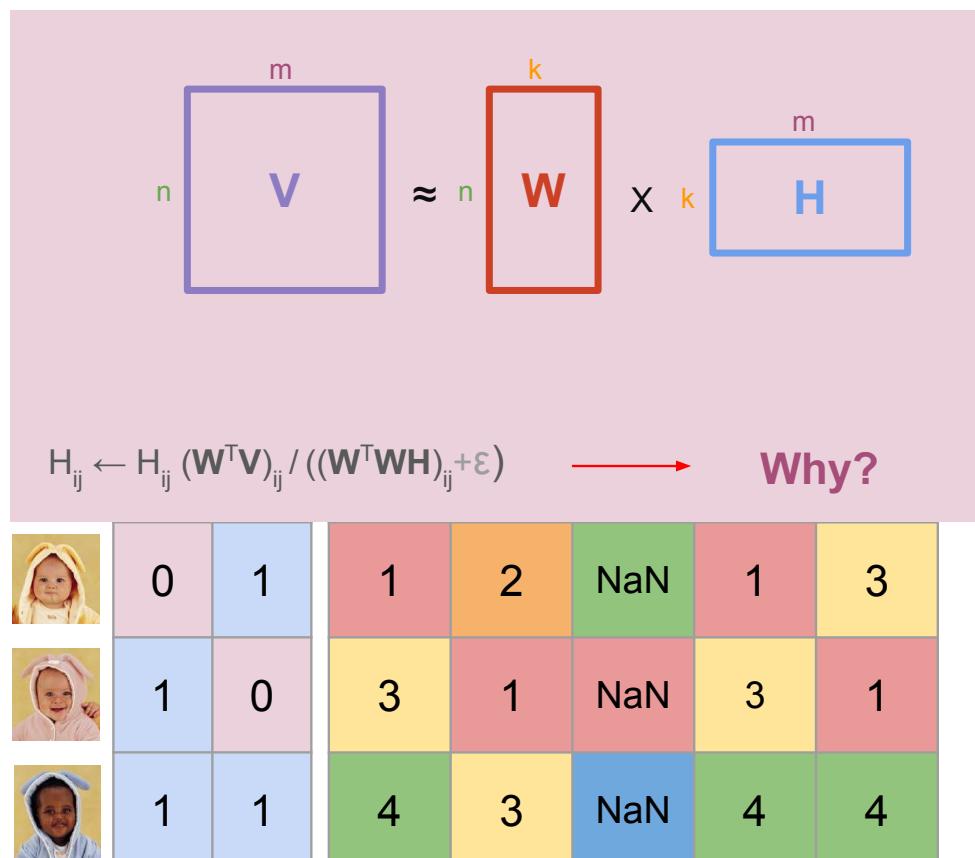
NMF using multiplicative update algorithm,
random initialization in (0,1), no normalization:

Reconstruction error known data: 3.23e-08
Reconstruction error missing data: **43.00**

		prediction				
		3.38	0.61	0.00	3.38	0.30
		0.75	1.55	0.00	0.75	2.33
		0.82	0.33			
		3.00000	0.99996	0.00	3.00000	1.00004
		1.00009	2.00002	0.00	1.00008	2.99992
		0.01	1.29			
		0.82	0.33			
		3.00000	0.99996	0.00	3.00000	1.00004
		3.99997	3.00001	0.00	3.99997	4.00004
		0.83	1.61			

Reconstruction

Example 3, missing column



NMF using multiplicative update algorithm,
random initialization in (0,1), no normalization:

Reconstruction error known data: 3.23e-08

Reconstruction error missing data: **43.00**

prediction

3.38	0.61	0.00	3.38	0.30
0.75	1.55	0.00	0.75	2.33

prediction

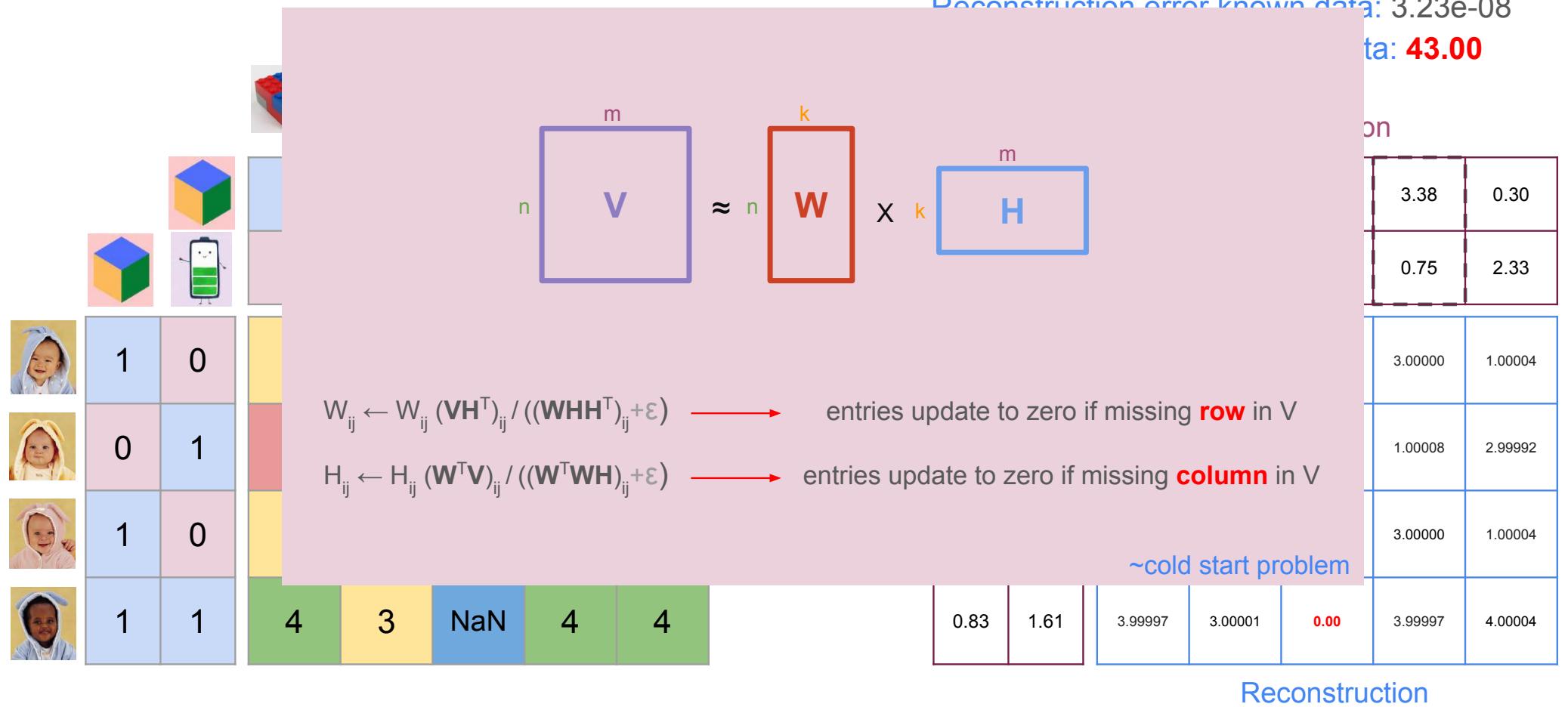
0.82	0.33			
0.01	1.29			
0.82	0.33			
0.83	1.61			

Reconstruction

3.00000	0.99996	0.00	3.00000	1.00004
1.00009	2.00002	0.00	1.00008	2.99992
3.00000	0.99996	0.00	3.00000	1.00004
3.99997	3.00001	0.00	3.99997	4.00004

Example 3, missing column

NMF using multiplicative update algorithm,
random initialization in (0,1), no normalization:



Example 4, sparse data

						
	3	1	1	3	1	
	1	2	4	1	3	
	1	0	Nan	1	1	Nan
	0	1	Nan	Nan	Nan	3
	1	0	3	1	Nan	3
	1	1	4	Nan	Nan	4

NMF using multiplicative update algorithm,
random initialization in (0,1), no normalization:

Reconstruction error known data: 2.08e-07
Reconstruction error missing data: **41.24**

		prediction				
		2.88	0.58	0.95	2.88	0.58
		0.87	1.52	1.36	0.87	1.51
		0.24	0.57	1.26	1.50	
		4.92	3.00	3.23	4.92	3.00
		0.95	0.29	3.00	1.00	1.30
		0.67	2.38	4.00	4.00	3.88
		Reconstruction				

Questions?

Recommendations

So far, we only considered reconstruction...

Question?

How do we build a recommender system?



NMF using **multiplicative update** algorithm,
random initialization in (0,1), no normalization:

Reconstruction error known data: 2.08e-07
Reconstruction error missing data: **41.24**

		prediction				
		2.88	0.58	0.95	2.88	0.58
		0.87	1.52	1.36	0.87	1.51
		0.24	0.57			
		1.26	1.50			
		0.95	0.29			
		0.67	2.38			
		Reconstruction				
		1.18	1.00	1.00	1.18	1.00
		4.92	3.00	3.23	4.92	3.00
		3.00	1.00	1.30	3.00	1.00
		4.00	4.00	3.88	4.00	4.00

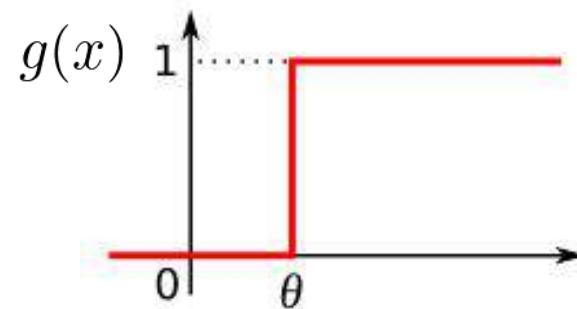
Recommendations

Threshold the elements of the reconstruction $\mathbf{V}' = \mathbf{W}\mathbf{H}$:

$$g(V'_{ij}) \in \{0, 1\}$$

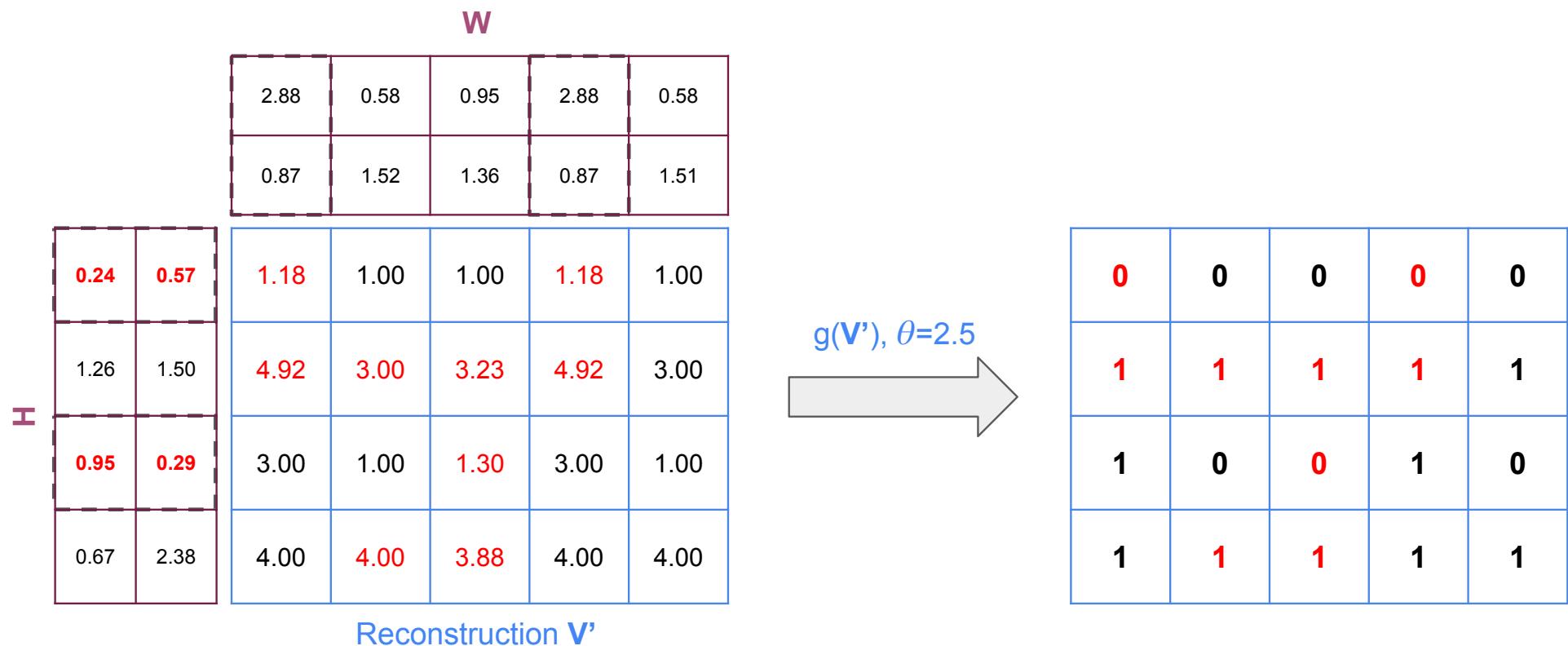
where

$$g(x) = \begin{cases} 0 & \text{if } x < \theta \\ 1 & \text{if } x \geq \theta \end{cases}$$



and 1 means recommend, 0 means don't recommend...

Recommendations



Recommendation accuracy

Accuracy is the percentage of correct recommendations.

$g(\mathbf{V}'), \theta=2.5$

0	0	0	0	0
1	1	1	1	1
1	0	0	1	0
1	1	1	1	1

Binarized reconstruction

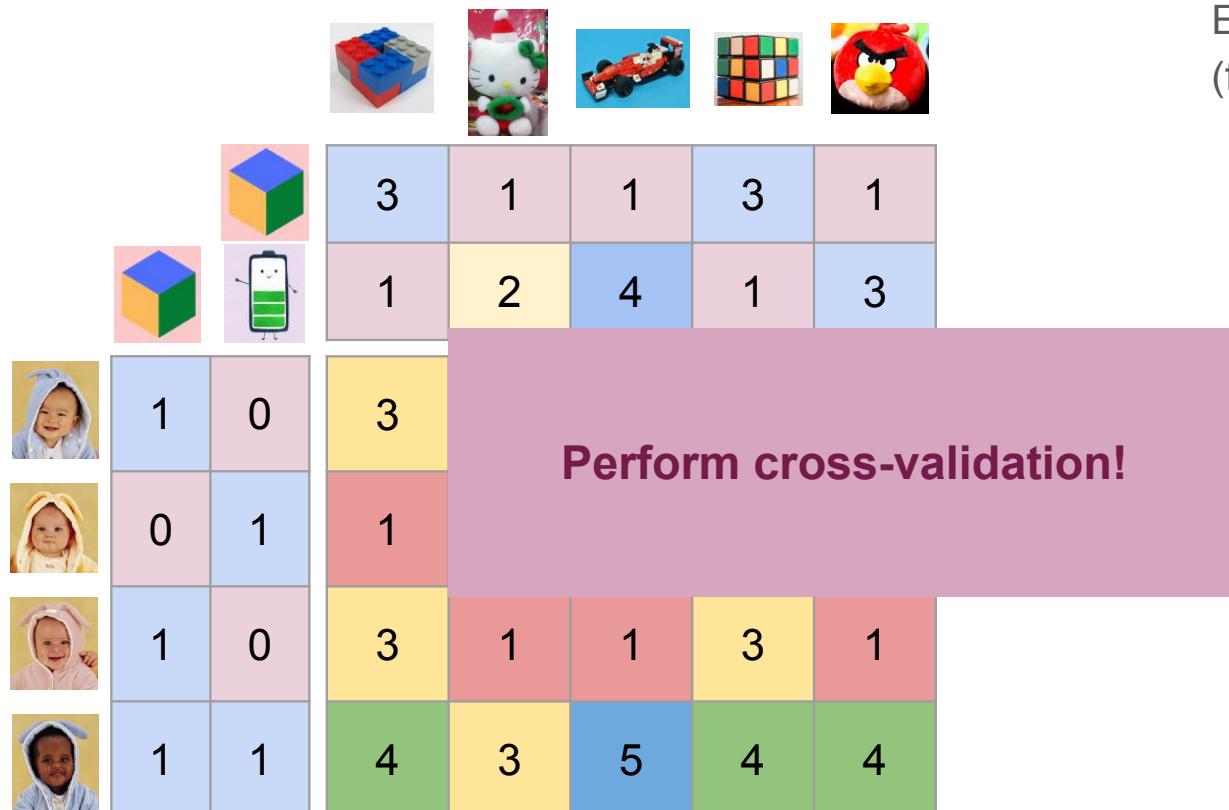
Ground truth

1	0	0	1	0
0	0	1	0	1
1	0	0	1	0
1	1	1	1	1

In this case $4/9 \approx 44.4\%$.

Questions?

From last time...



NMF using **multiplicative update** algorithm,
random initialization in (0,1), no normalization.

Effect of using different number of components
(features "k"):

	Total reconstruction error
1 component	15.24
2 components	5.33e-10
3 components	2.76e-08
4 components	7.56e-05

Perform cross-validation!

Cross-validation

Assume we evaluate NMF performance using the recommendation accuracy.

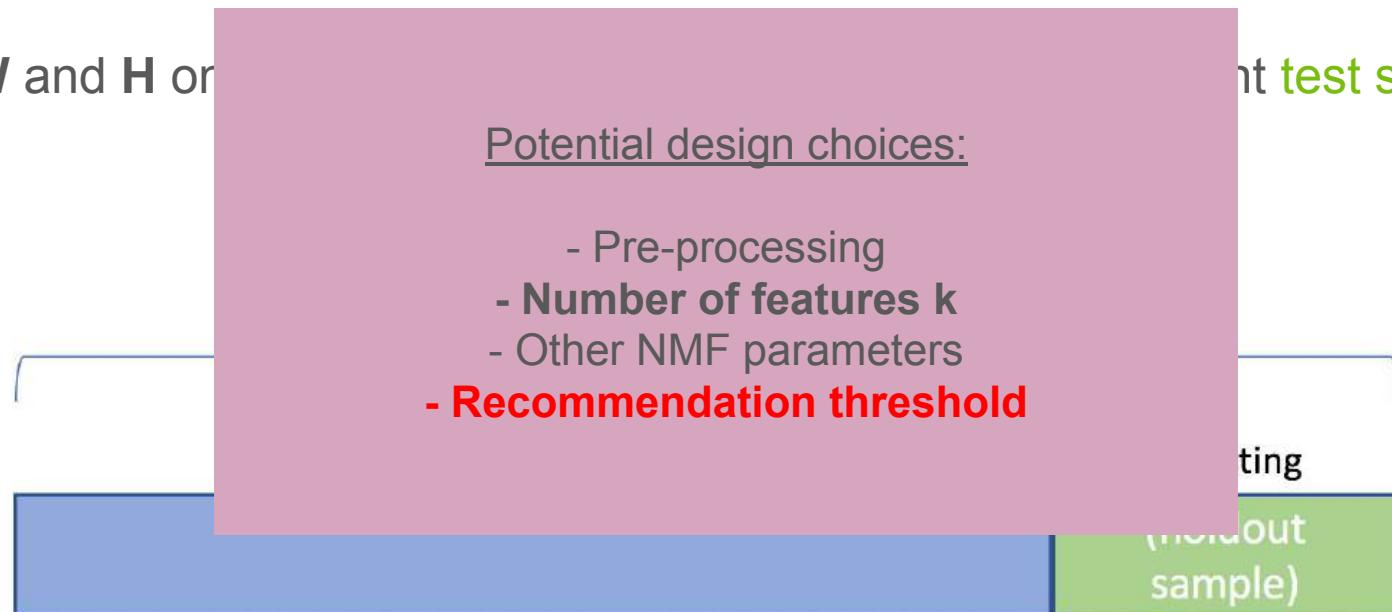
We optimize **W** and **H** on a **training dataset**, evaluate on an independent **test set**.



Cross-validation

Assume we evaluate NMF performance using the recommendation accuracy.

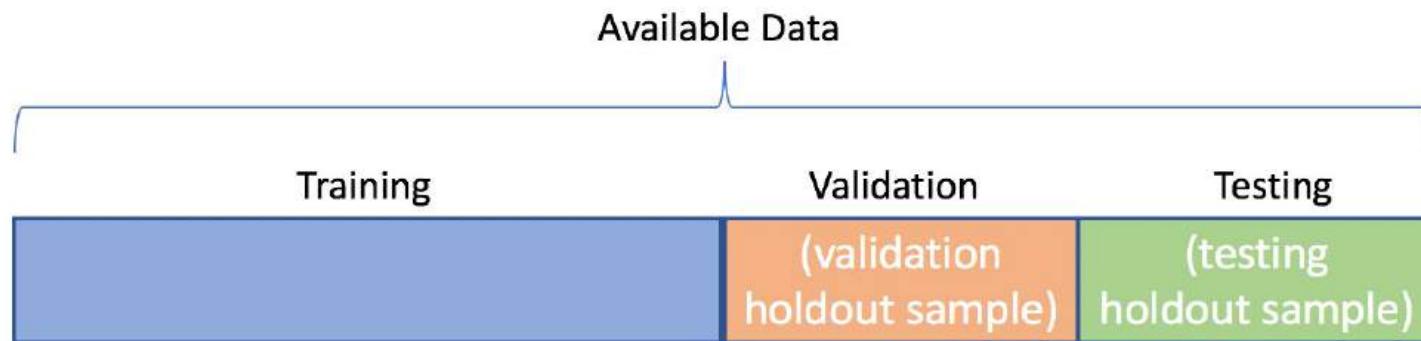
We optimize **W** and **H** or **W** only on the training set and evaluate on the test set.



Cross-validation

We don't want to overfit our model parameters/design choices to the test set.

Use an independent validation set!

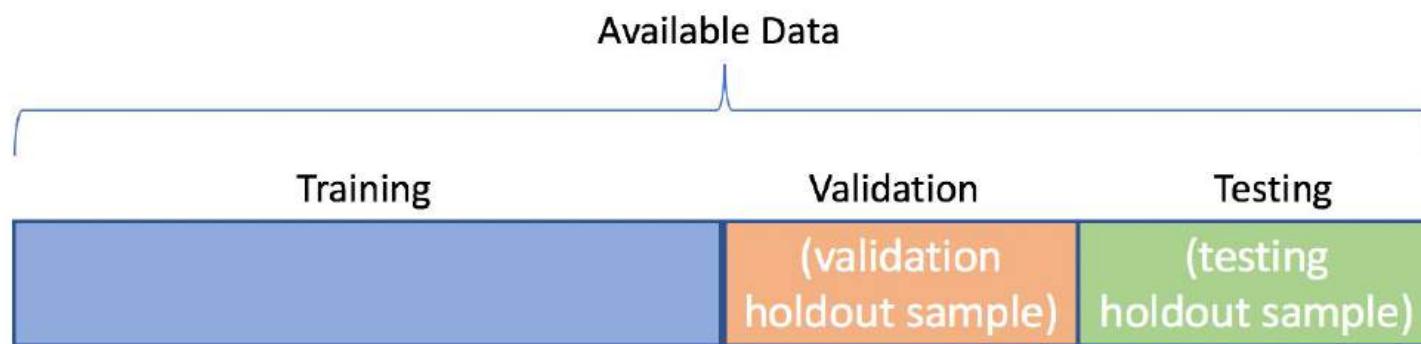


Cross-validation

We don't want to overfit our model. We want to evaluate its performance on an independent test set.

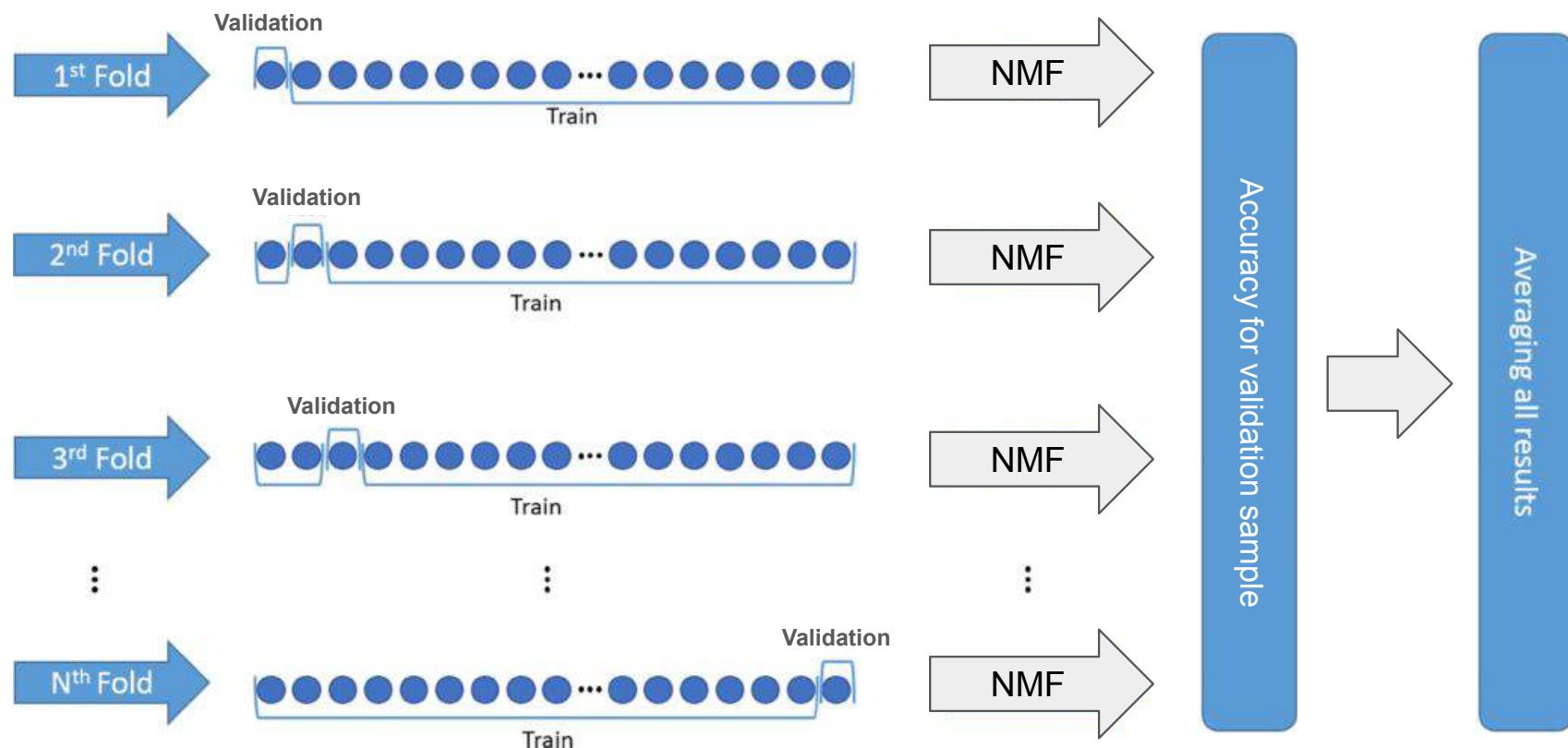
Use an independent validation set.

How to divide the data for cross-validation?



Cross-validation

Leave-one-out cross-validation (mask out one entry of \mathbf{V} during training)



Cross-validation

Leave-one-out cross-validation (mask out one entry of \mathbf{V} during training)

NaN	Validation	1	NaN	1
NaN	NaN	NaN	NaN	3
3	1	NaN	3	1
4	NaN	NaN	4	4

1st fold

Cross-validation

Leave-one-out cross-validation (mask out one entry of \mathbf{V} during training)

NaN	1	Validation	NaN	1
NaN	NaN	NaN	NaN	3
3	1	NaN	3	1
4	NaN	NaN	4	4

2nd fold

Cross-validation

Leave-one-out cross-validation (mask out one entry of \mathbf{V} during training)

NaN	1	1	NaN	Validation
NaN	NaN	NaN	NaN	3
3	1	NaN	3	1
4	NaN	NaN	4	4

3rd fold

Cross-validation

Leave-one-out cross-validation (mask out one entry of \mathbf{V} during training)

NaN	1	1	NaN	1
NaN	NaN	NaN	NaN	Validation
3	1	NaN	3	1
4	NaN	NaN	4	4

4th fold

Cross-validation

Leave-one-out cross-validation (mask out one entry of \mathbf{V} during training)

NaN	1	1	NaN	1
NaN	NaN	NaN	NaN	3
Validation	1	NaN	3	1
4	NaN	NaN	4	4

5th fold, etc...

Cross-validation

k-fold cross-validation (mask out mutually exclusive subsets of entries in \mathbf{V} for training)



k-fold cross-validation

For many tasks, our **data matrix** is an n -by- d matrix which has n samples, and d features.

d features

n samples

sepal length	sepal width	petal length	petal width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9
6.8	3	5.5	2.1
6.7	3.1	4.4	1.4
5.6	3	4.5	1.5
5.8	2.7	4.1	1

k-fold cross-validation

For many tasks, we can pick **whole ‘samples’ (rows)** as a holdout validation set (e.g. remember PCA).

***d* features**

sepal length	sepal width	petal length	petal width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9
6.8	3	5.5	2.1
6.7	3.1	4.4	1.4
5.6	3	4.5	1.5
5.8	2.7	4.1	1

1st fold

n samples

k-fold cross-validation

For many tasks, we can pick **whole ‘samples’ (rows)** as a holdout validation set (e.g. remember PCA).

***d* features**

sepal length	sepal width	petal length	petal width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9
6.8	3	5.5	2.1
6.7	3.1	4.4	1.4
5.6	3	4.5	1.5
5.8	2.7	4.1	1

2nd fold

n samples

k-fold cross-validation

For many tasks, we can pick **whole ‘samples’ (rows)** as a holdout validation set (e.g. remember PCA).

***d* features**

sepal length	sepal width	petal length	petal width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9
6.8	3	5.5	2.1
6.7	3.1	4.4	1.4
5.6	3	4.5	1.5
5.8	2.7	4.1	1

3rd fold

n samples

k-fold cross-validation

For many tasks, we can pick **whole ‘samples’ (rows)** as a holdout validation set (e.g. remember PCA).

***d* features**

sepal length	sepal width	petal length	petal width
5.1	3.5	1.4	0.2
4.9	3	1.4	0.2
6.5	3.2	5.1	2
6.4	2.7	5.3	1.9
6.8	3	5.5	2.1
6.7	3.1	4.4	1.4
5.6	3	4.5	1.5
5.8	2.7	4.1	1

4th fold

n samples

m toys



n babies

	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

How do we pick the cross-validation folds for NMF?

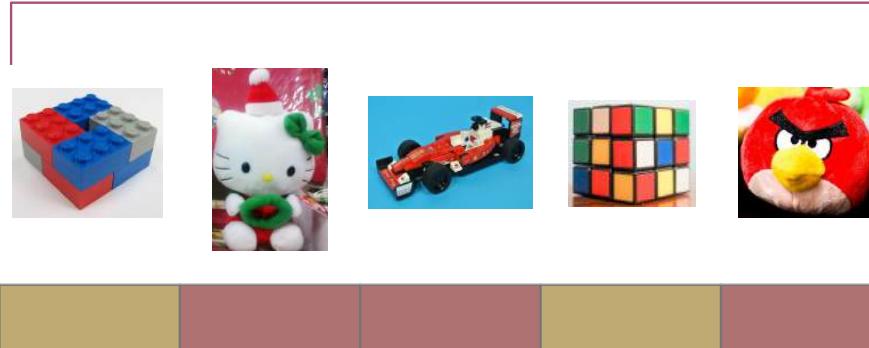
m toys



n babies	1	2	3	4	5
1	3	1	1	3	1
2	1	2	4	1	3
3	3	1	1	3	1
4	4	3	5	4	4

For multiplicative update algorithm,
we cannot mask
whole rows!

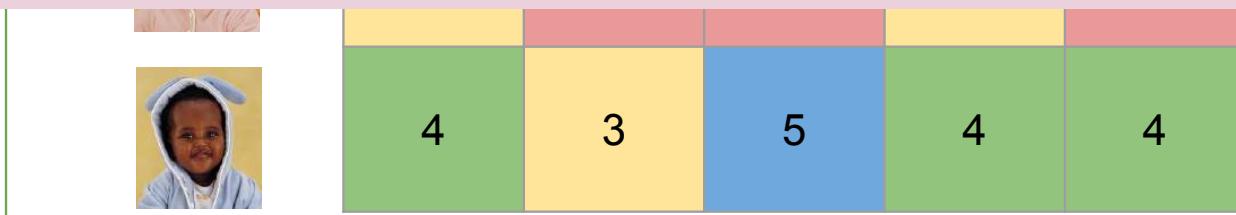
m toys



Remember: We cannot have whole row/column missing for the multiplicative update algorithm...

$$W_{ij} \leftarrow W_{ij} (\mathbf{V}\mathbf{H}^T)_{ij} / ((\mathbf{W}\mathbf{H}\mathbf{H}^T)_{ij} + \varepsilon) \longrightarrow \text{entries update to zero if missing } \textcolor{red}{\text{row}} \text{ in } \mathbf{V}$$

$$H_{ij} \leftarrow H_{ij} (\mathbf{W}^T \mathbf{V})_{ij} / ((\mathbf{W}^T \mathbf{W} \mathbf{H})_{ij} + \varepsilon) \longrightarrow \text{entries update to zero if missing } \textcolor{red}{\text{column}} \text{ in } \mathbf{V}$$



For multiplicative update algorithm, we cannot mask whole rows!

m toys



n babies

	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

Conceptually, we cannot perform **collaborative filtering** if we don't have any information about a user/item.

m toys



n babies

	3	1	1	3	1
	1	2	4	1	3
	3	1	1	3	1
	4	3	5	4	4

We can pick
mutually exclusive
cross-validation
folds from different
rows/columns.

Cross-validation

k-fold cross-validation (mask out mutually exclusive subsets of entries in \mathbf{V} for training)

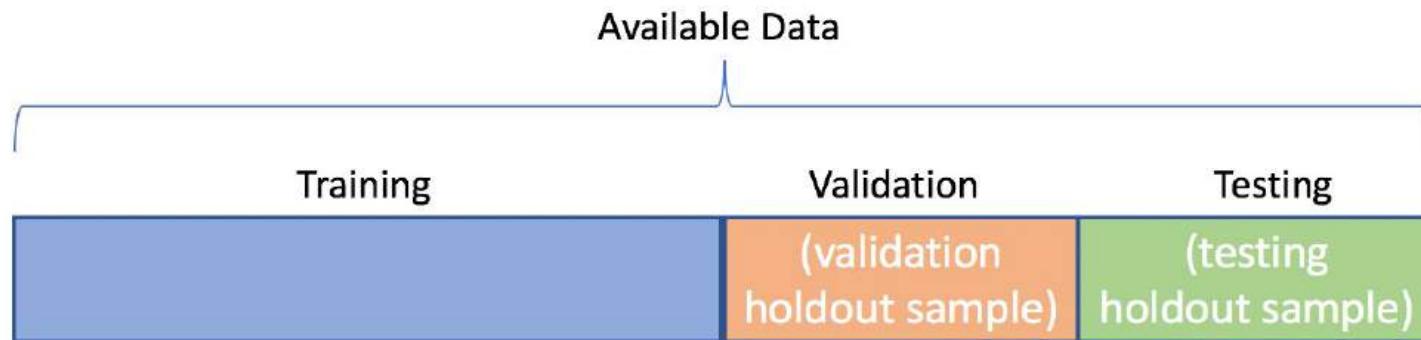
Be careful not to mask out **all entries** in a row/column!
(Multiplicative update will set values to zero!)



Cross-validation - Recommender systems

Data is large → - **leave-one-out**: computationally expensive

Data is sparse → - **k-fold with small k**: training data might be too small, not informative

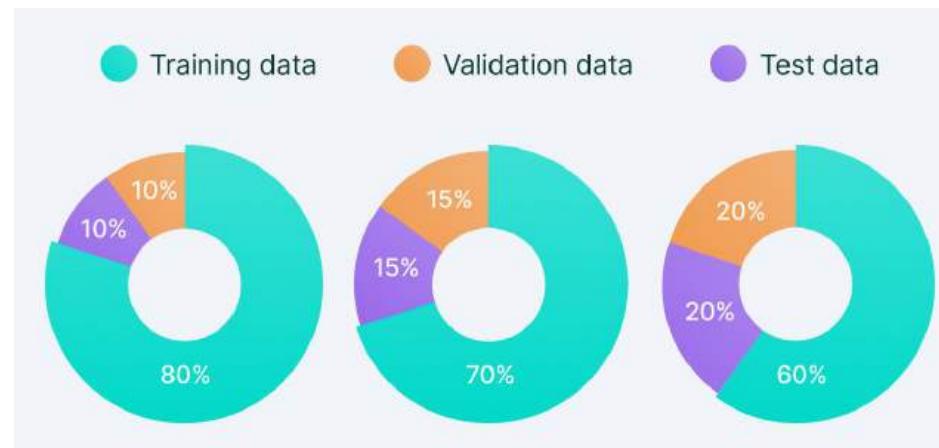


Cross-validation - Recommender systems

Data is large → - **leave-one-out**: computationally expensive

Data is sparse → - **k-fold with small k**: training data might be too small not informative

If it's computationally **too expensive** to use large number of folds, we can also use **one** independent validation set and **one** independent test set (e.g. lab assignment).



Questions?

Performance metrics

- Evaluation metrics for **predictions**

- Reconstruction error: Mean squared error ($1/N * ||\mathbf{V} - \mathbf{WH}||^2$),
root mean squared error

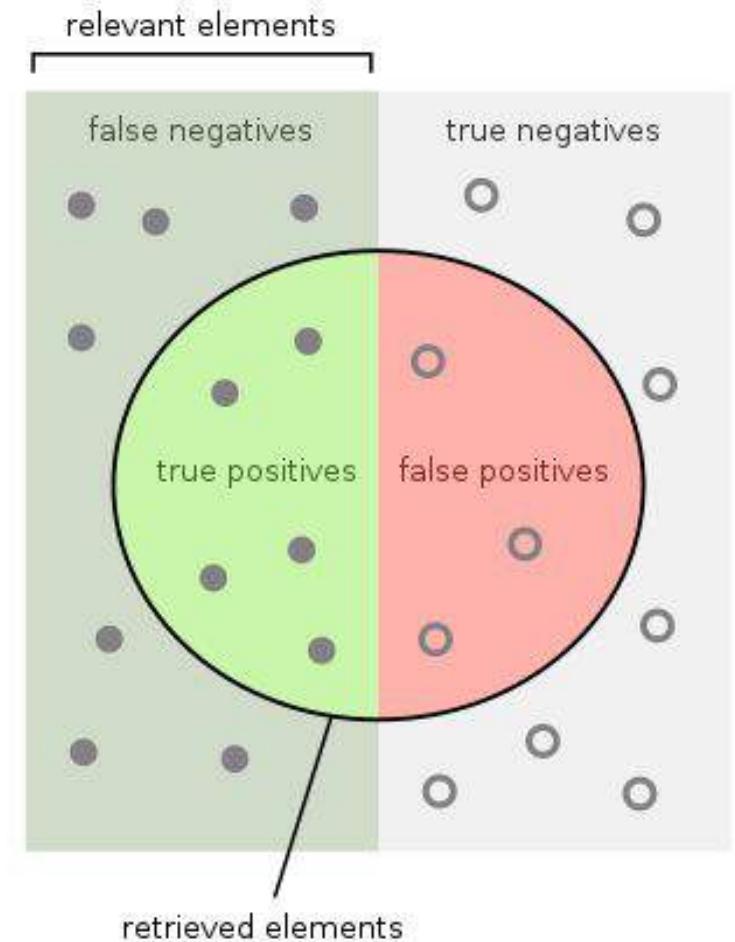
- Evaluation metrics for **recommendations**

- Recommendation accuracy: Percentage of correct recommendations.

How do we evaluate
NMF performance?

Performance metrics

- Further evaluation metrics for **recommendations**
 - True positive/false positive rates



Performance metrics

- Further evaluation metrics for **recommendations**

 - True positive/false positive rates

$$\text{Precision} = \frac{tp}{tp + fp}$$

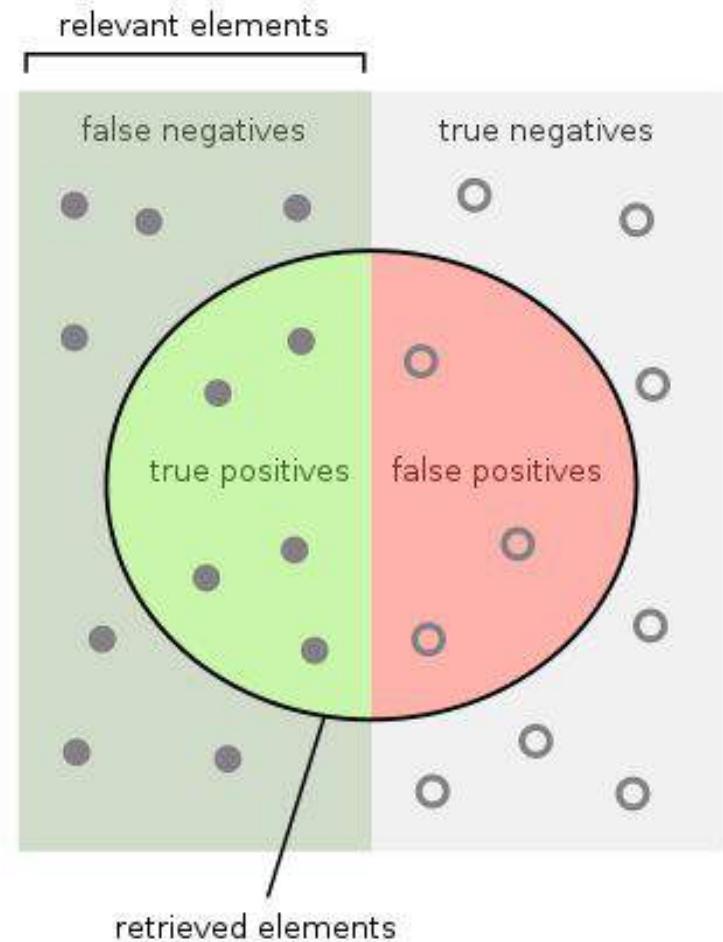
$$\text{Recall} = \frac{tp}{tp + fn}$$

How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{relevant elements}}{\text{retrieved elements}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{relevant elements}}{\text{true positives} + \text{false positives}}$$



Performance metrics

- Further evaluation metrics for **recommendations**

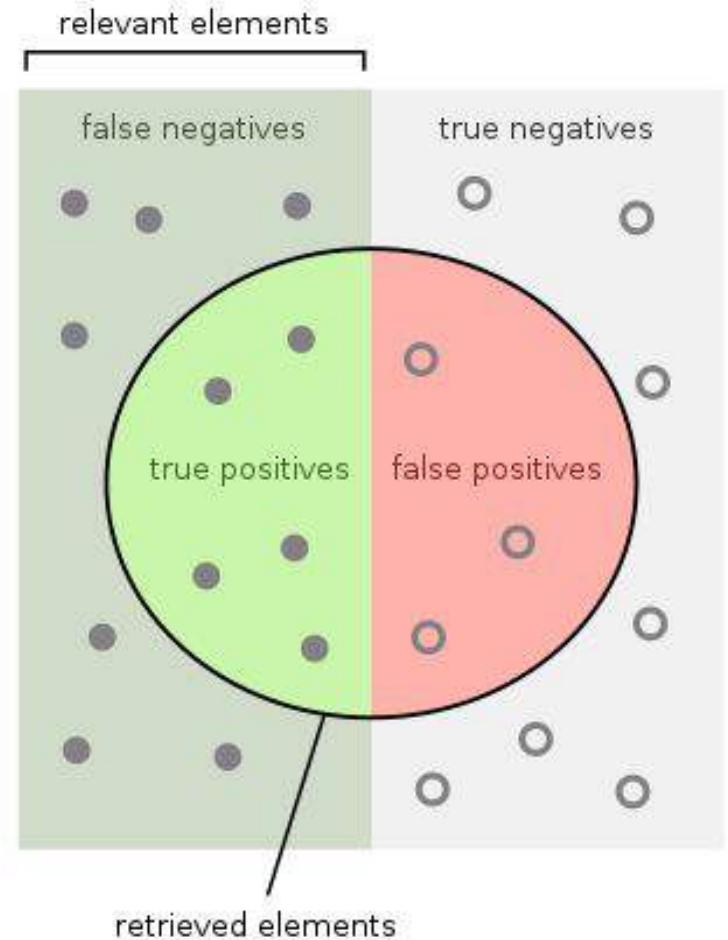
- True positive/false positive rates

$$\text{Precision} = \frac{tp}{tp + fp}$$

$$\text{Recall} = \frac{tp}{tp + fn}$$

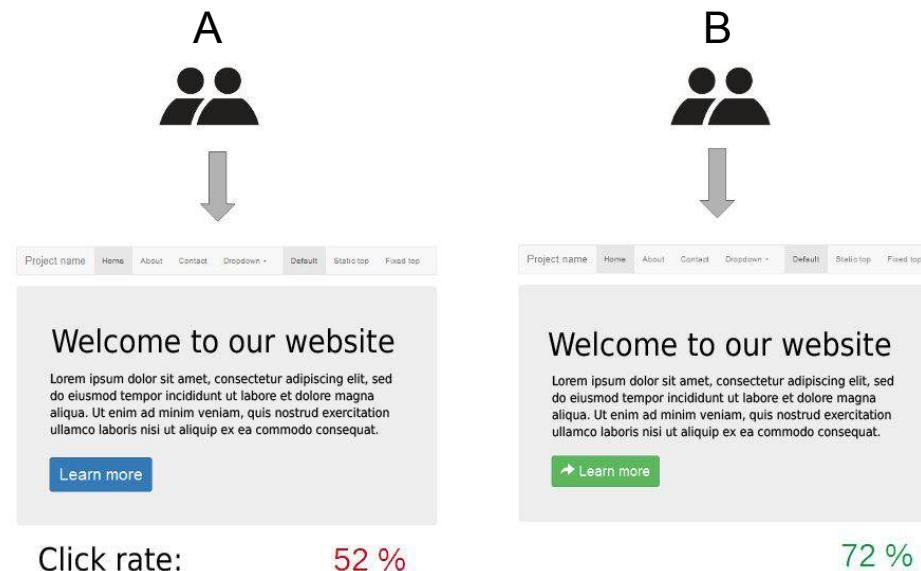
- F1-score:

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$



Performance metrics - Recommender systems

- **Online** evaluation: User's online reactions, e.g. the clicks or views a recommendation gets, etc.
 - A/B testing



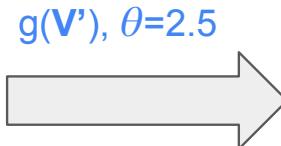
Performance metrics - Recommender systems

- Ranking based recommendations:

So far we only performed naive thresholding to get recommendations

1.18	1.00	1.00	1.28	1.00
4.92	3.00	3.23	4.82	3.00
3.00	1.00	1.30	3.00	1.00
4.00	4.00	3.88	4.00	4.00

Reconstruction \mathbf{V}'



0	0	0	0	0
1	1	1	1	1
1	0	0	1	0
1	1	1	1	1

Performance metrics - Recommender systems

- Ranking based recommendations:

If we want to recommend k items to each user, we have to consider rankings!

1.18	1.00	1.00	1.28	1.00
4.92	3.00	3.23	4.82	3.00
3.00	1.00	1.30	3.00	1.00
4.00	4.00	3.88	4.00	4.00

Reconstruction V'

Ranking based
→

2	-	-	1	-
1	4	3	2	-
-	-	1	-	-
-	1	2	-	-

Performance metrics - Recommender systems

- Ranking based evaluation metrics include

- Spearman's rank correlation:
Pearson's correlation between predicted ranks and ground truths
- Hit rate at K:
Percentage of users for which at least one hit takes place for the top K recommendations.

Ranking based


2	-	-	1	-
1	4	3	2	-
-	-	1	-	-
-	1	2	-	-

Other considerations

- **Diversity** (exploration) – Users tend to be more satisfied if recommendations are [diverse](#).
- **Recommender persistence** – It can be more effective to [re-show](#) recommendations than showing new items.

Other considerations

- **Diversity** (exploration) – Users tend to be more satisfied if recommendations are [diverse](#).
- **Recommender persistence** – It can be more effective to [re-show](#) recommendations than showing new items.
- **Robustness** – What if two users are using the same account? How to deal with [unreliable data](#)?
- **Serendipity** – Serendipity is a measure of "how surprising the recommendations are". If you're an e-commerce dairy farm, milk is not a [surprising recommendation](#), but biscuits might be.

Other considerations

- **Diversity** (exploration) – Users tend to be more satisfied if recommendations are [diverse](#).
- **Recommender persistence** – It can be more effective to [re-show](#) recommendations than showing new items.
- **Robustness** – What if two users are using the same account? How to deal with [unreliable data](#)?
- **Serendipity** – Serendipity is a measure of "how surprising the recommendations are". If you're an e-commerce dairy farm, milk is not a [surprising recommendation](#), but biscuits might be.
- **Privacy** – Recommender systems usually have to deal with privacy concerns because users might have to reveal [sensitive and personally identifying information](#). Building user profiles using collaborative filtering can be problematic from a privacy point of view.

Questions?

Summary

- Recommender systems are **ubiquitous**
- There are **multiple** possible **approaches** (collaborative, content-based, hybrid)
- Different **types of utility matrices** (ranking-based, preference-based, dense, sparse, etc.)

Summary

- Recommender systems are **ubiquitous**
- There are **multiple** possible **approaches** (collaborative, content-based, hybrid)
- Different **types of utility matrices** (ranking-based, preference-based, dense, sparse, etc.)
- Important to think about **cross-validation** when picking methods and making design choices

Summary

- Recommender systems are **ubiquitous**
- There are **multiple** possible **approaches** (collaborative, content-based, hybrid)
- Different **types of utility matrices** (ranking-based, preference-based, dense, sparse, etc.)
- Important to think about **cross-validation** when picking methods and making design choices

Next time:

- **Privacy** is important to consider when dealing with (big) user data
- **Alternatives** to NMF provide different methods for CF



Lecture 14 (Today)	Week 2.8	Jan 13	Recommender systems & Manifold learning
Lecture 15	Week 2.8	Jan 16	Dimensionality reduction
Lecture 16	Week 2.9	Jan 20	Exam preparation, Q&A
Lecture 17	Week 2.9	Jan 23	FREE
Exam	Week 2.10	Jan 27	Weblab exam

Recommender Systems

Harry Potter	The Triplets of Belleville	Shrek	The Dark Knight Rises	Memento

Continued from last time...

Performance metrics - Recommender systems

- Ranking based recommendations:

If we want to recommend k items to each user, we have to consider rankings!

1.18	1.00	1.00	1.28	1.00
4.92	3.00	3.23	4.82	3.00
3.00	1.00	1.30	3.00	1.00
4.00	4.00	3.88	4.00	4.00

Reconstruction V'

Ranking based
→

2	-	-	1	-
1	4	3	2	-
-	-	1	-	-
-	1	2	-	-

Performance metrics - Recommender systems

- Ranking based evaluation metrics include

- Spearman's rank correlation:
Pearson's correlation between predicted ranks and ground truths.
- Hit rate at K:
Percentage of users for which at least one hit takes place for the top K recommendations.

Ranking based


2	-	-	1	-
1	4	3	2	-
-	-	1	-	-
-	1	2	-	-

Performance metrics - Spearman's rank correlation

Rank predictions

2	-	-	1	-
1	4	3	2	-
-	-	1	-	-
-	1	2	-	-

Rank ground truth

2	-	-	1	-
3	2	1	4	-
-	-	1	-	-
-	2	1	-	-

Performance metrics - Spearman's rank correlation

Rank predictions

2	-	-	1	-
1	4	3	2	-
-	-	1	-	-
-	1	2	-	-

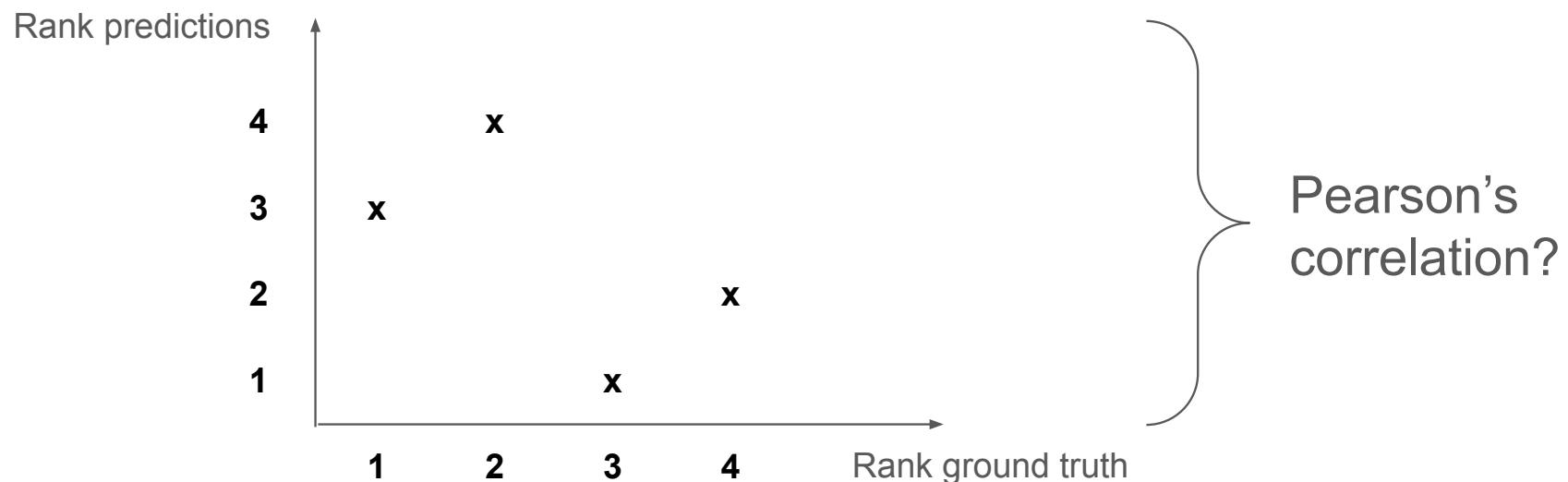
Rank ground truth

1	-	-	1	-
3	2	1	4	-
-	-	1	-	-
-	2	1	-	-

Spearman's rank correlation

- Convert predictions (reconstruction) and ground truths to **rank** → Then find the correlation!

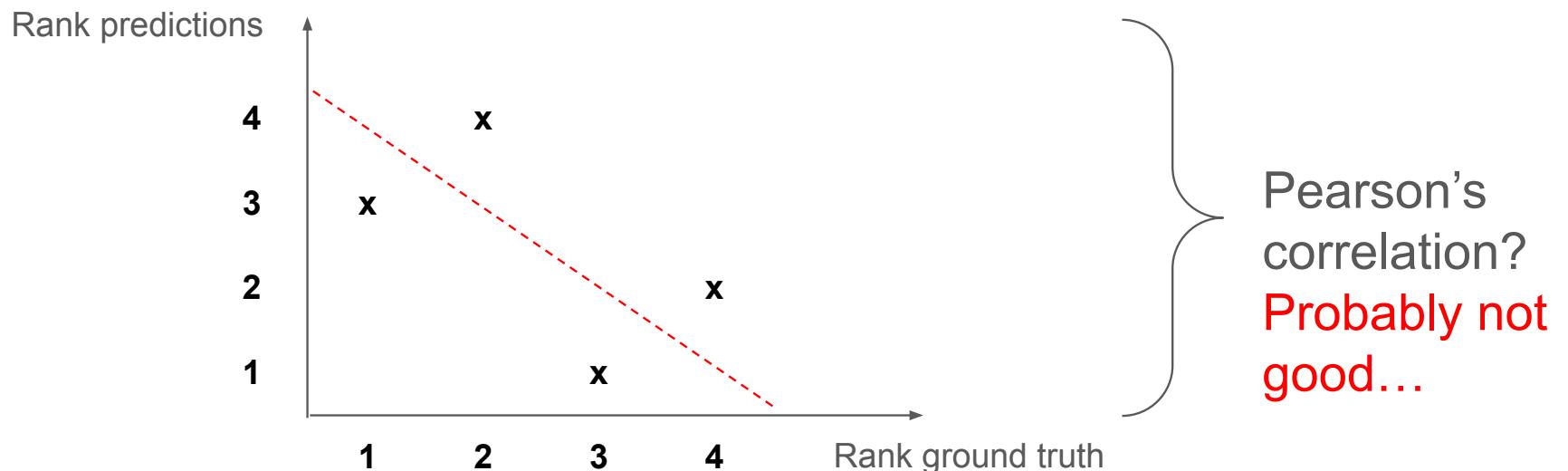
Rank predictions					Rank ground truth				
1	4	3	2	-	3	2	1	4	-



Spearman's rank correlation

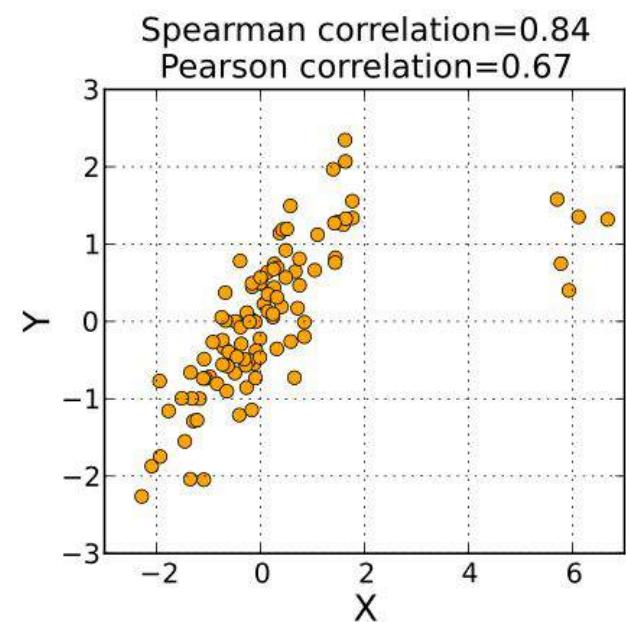
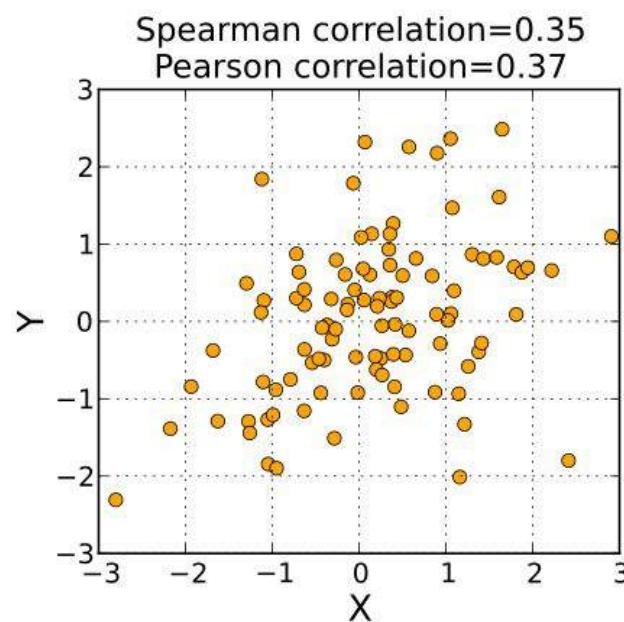
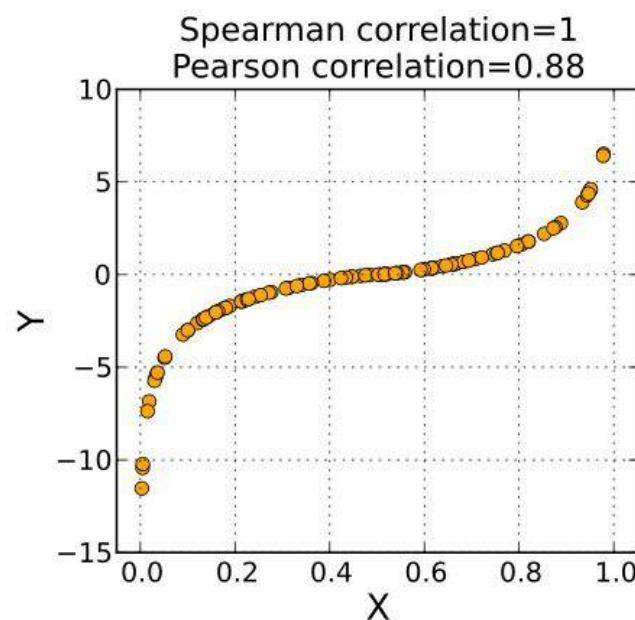
- Convert predictions (reconstruction) and ground truths to **rank** → Then find the correlation!

Rank predictions					Rank ground truth				
1	4	3	2	-	3	2	1	4	-



Spearman's rank correlation

Just a measure of monotonicity!



Other considerations

- **Diversity** (exploration) – Users tend to be more satisfied if recommendations are [diverse](#).
- **Recommender persistence** – It can be more effective to [re-show](#) recommendations than showing new items.
- **Robustness** – What if two users are using the same account? How to deal with [unreliable data](#)?
- **Serendipity** – Serendipity is a measure of "how surprising the recommendations are". If you're an e-commerce dairy farm, milk is not a [surprising recommendation](#), but biscuits might be.

Hybrid recommender systems

Sponsored ::

							
Bayer Chic 2000 612 25 - Joggin... €35.81 Amazon.nl Free shipping By Kelkoo	Bebeconfort Peps Buggy met... €49.99 bol.com Free shipping By Producthero	Ding Nora Buggy - Zwart -... €49.90 bol.com Free shipping By Producthero	YOYO Beensteun Stokke NL €30.00 +€7.90 shipping ★★★★★ (442) By Google	Maxi-Cosi Lara2 Kinderwagen, 0... €134.99 Amazon.nl Free shipping By Kelkoo	Bebies First Buggy... €59.99 bol.com Free shipping By Producthero	Bayer Chic 2000 601-71 Mini... €17.62 Amazon.nl +€2.99 shipping By Kelkoo	Bayer Design 30112AA... €19.99 Amazon.nl +€10.99 shipping By Kelkoo

Hybrid recommender systems

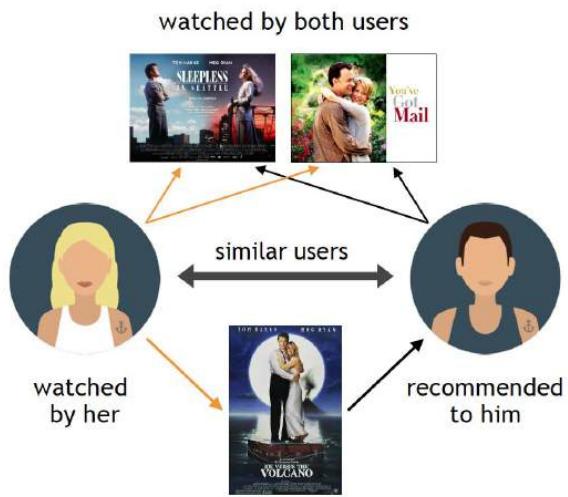
Sponsored ::

Rain cover: Diverse, surprising, useful!

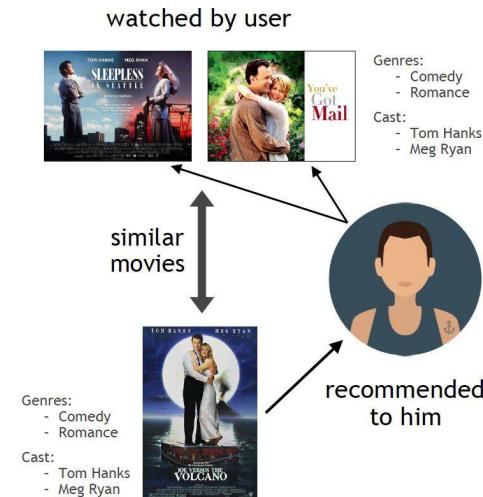
							
<p>Bayer Design - Poppenbuggy - Blauw met koeien</p> <p>€14.99 bol.com +€2.99 shipping</p> <p>By Producthero</p>	<p>Universele regenhoes voor kinderwagen, weerbestendig...</p> <p>€15.99 Amazon.nl +€2.99 shipping</p> <p>By Kelkoo</p>	<p>Little dutch poppenbuggy</p> <p>€19.95 ilovespeelgoed.... +€4.95 shipping</p> <p>By Bigshopper</p>	<p>Ding Nora Black Buggy DI-902775</p> <p>€49.90 Mamaloes.nl Free shipping</p> <p>By Bigshopper</p>	<p>kinderwagen voor kinderen van 6 tot 36 maanden - licht en...</p> <p>€90.00 bol.com Free shipping</p> <p>By Producthero</p>	<p>Bayer - Poppenbuggy - Roze met prinses</p> <p>€14.99 bol.com +€2.99 shipping</p> <p>By Producthero</p>	<p>Cybex gb Gold Wandelwagen, buggy, Pockit Air All Terrain,...</p> <p>€152.22 Amazon.nl Free shipping</p> <p>By Kelkoo</p>	<p>Poppenbuggy Funny, poppenwagen, mini-buggy,...</p> <p>€16.30 Amazon.nl +€2.99 shipping</p> <p>By Kelkoo</p>
							

Recommender systems

Collaborative Filtering



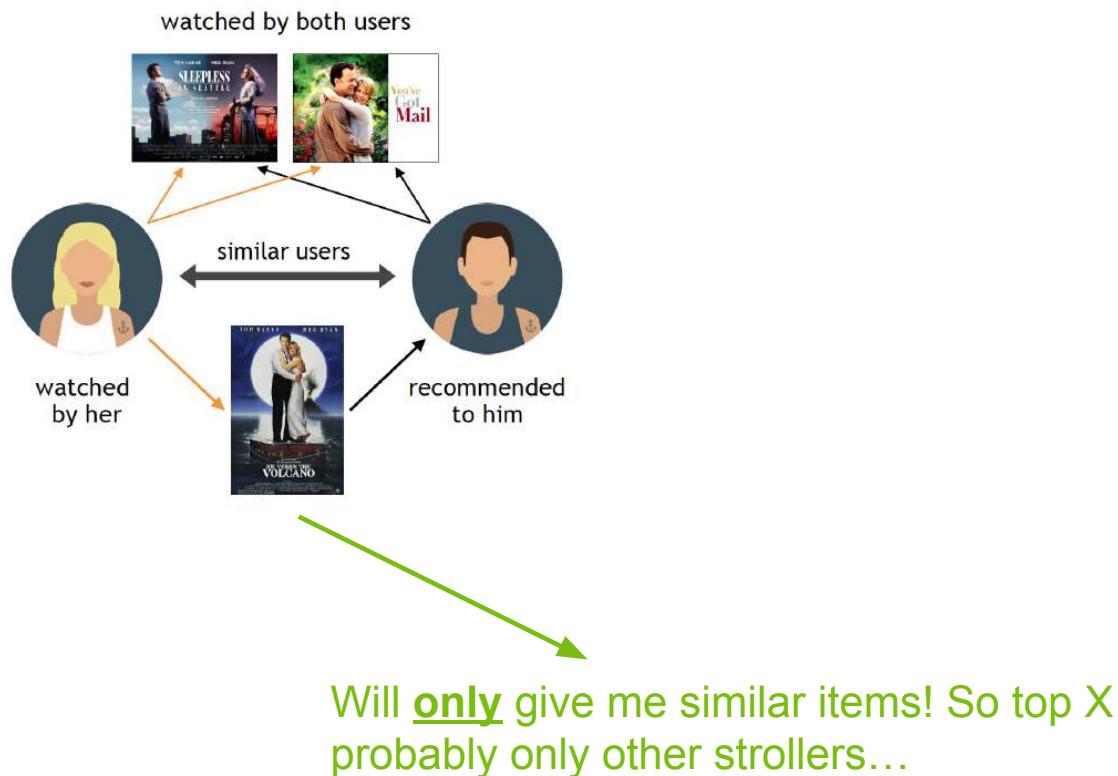
Content-based Filtering



Often the two approaches are combined.

Recommender systems

Collaborative Filtering



Recommender systems

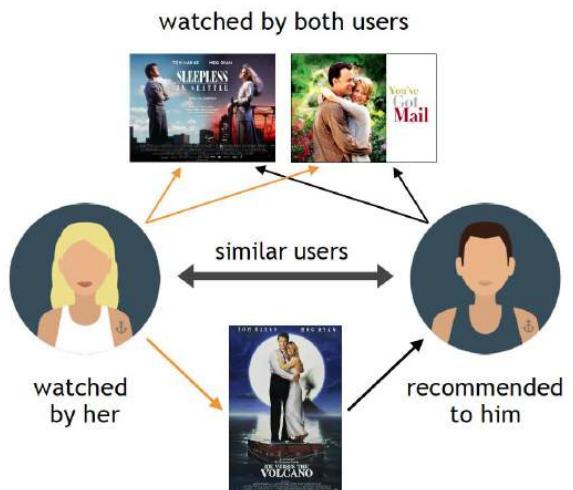
Content-based Filtering



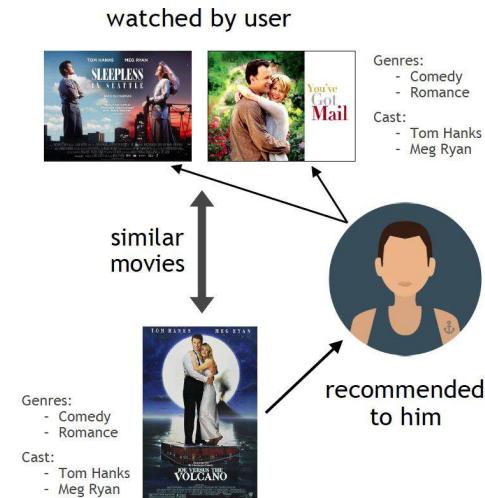
I can use **item descriptions** to find items in the category of “accessories” (all accessory items, not just for strollers...)

Recommender systems

Collaborative Filtering



Content-based Filtering



Combine: I can search for similar items (highest collaborative filtering prediction) in the category of “accessories”!

Questions?

Other considerations

- **Diversity** (exploration) – Users tend to be more satisfied if recommendations are [diverse](#).
- **Recommender persistence** – It can be more effective to [re-show](#) recommendations than showing new items.
- **Robustness** – What if two users are using the same account? How to deal with [unreliable data](#)?
- **Serendipity** – Serendipity is a measure of "how surprising the recommendations are". If you're an e-commerce dairy farm, milk is not a [surprising recommendation](#), but biscuits might be.
- **Privacy** – Recommender systems usually have to deal with privacy concerns because users might have to reveal [sensitive and personally identifying information](#). Building user profiles using collaborative filtering can be problematic from a privacy point of view.

Last time: Long-tail phenomenon

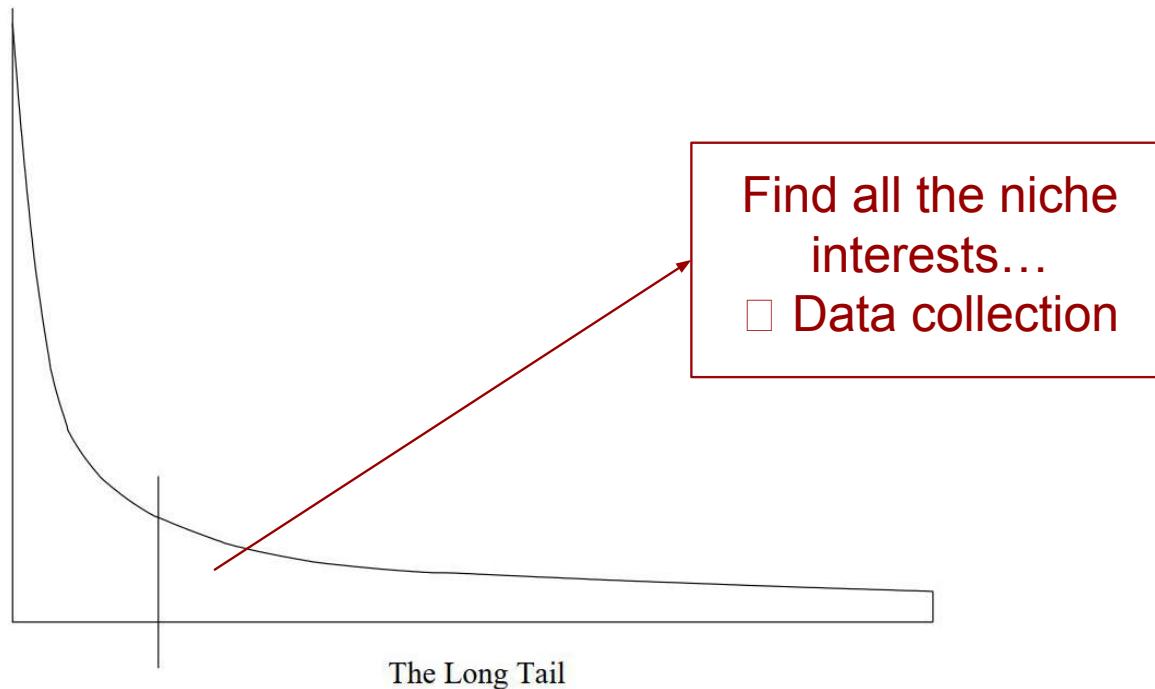


Figure 9.2: The long tail: physical institutions can only provide what is popular, while on-line institutions can make everything available

Privacy

What might be some **privacy concerns** for
(recommender system) data collection?

Privacy

Cross-referencing (anonymized) datasets can reveal personally identifying information.

Table 20.1: Example of a data table

SSN	Age	ZIP Code	Disease
012-345-6789	24	10598	HIV
823-627-9231	37	90210	Hepatitis C
987-654-3210	26	10547	HIV
382-827-8264	38	90345	Hepatitis C
847-872-7276	36	89119	Diabetes
422-061-0089	25	02139	HIV

Table 20.2: Example of a snapshot of fictitious voter registration rolls

Name	Age	ZIP Code
Mary A.	38	90345
John S.	36	89119
Ann L.	31	02139
Jack M.	57	10562
Joy M.	26	10547
Victor B.	46	90345
Peter P.	25	02139
Diana X.	24	10598
William W.	37	90210
Sue G.	26	10547

Privacy

Cross-referencing (anonymized) datasets can reveal personally identifying information.

Table 20.1: Example of a data table

SSN	Age	ZIP Code	Disease
012-345-6789	24	10598	HIV
823-627-9231	37	90210	Hepatitis C
987-654-3210	26	10547	HIV
382-827-8264	38	90345	Hepatitis C
847-872-7276	36	89119	Diabetes
422-061-0089	25	02139	HIV

Table 20.2: Example of a snapshot of fictitious voter registration rolls

Name	Age	ZIP Code
Mary A.	38	90345
John S.	36	89119
Ann L.	31	02139
Jack M.	57	10562
Joy M.	26	10547
Victor B.	46	90345
Peter P.	25	02139
Diana X.	24	10598
William W.	37	90210
Sue G.	26	10547

Privacy

What is "**information privacy**"?

Privacy

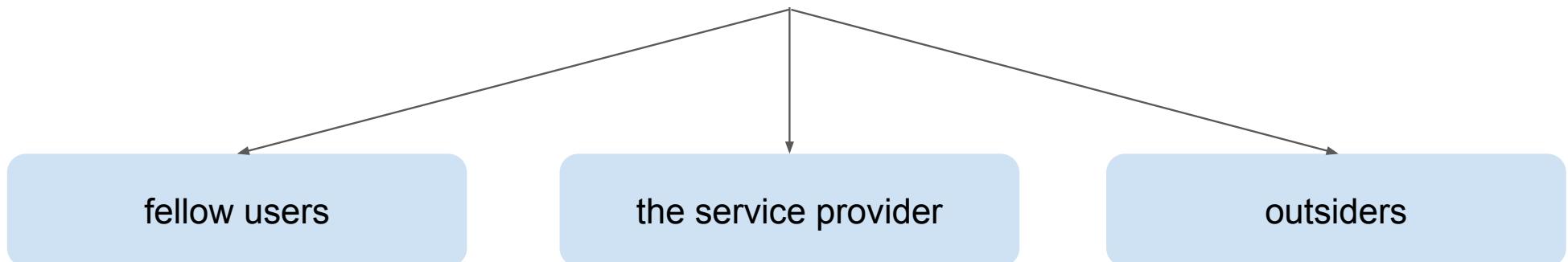
"**Information privacy** is an individual's claim to **control** the terms under which personal information – information identifiable to the individual – is acquired, disclosed or used."

Consent is important!

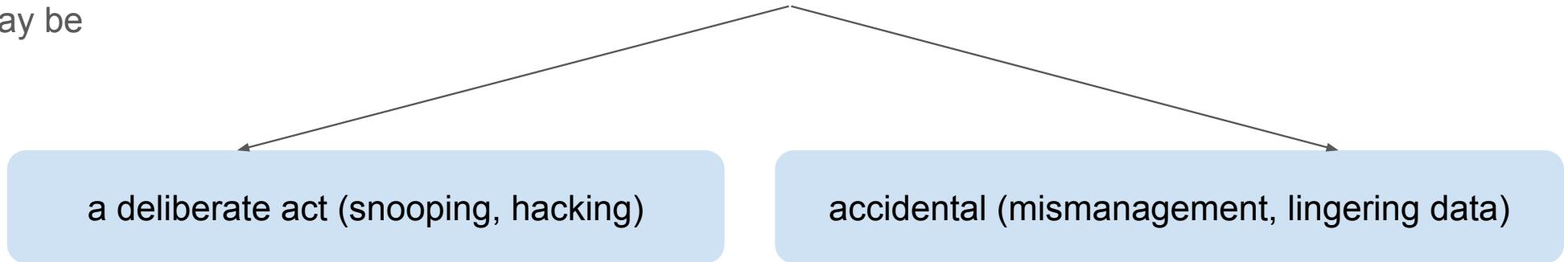
[1] Jeckmans AJ, et al. Privacy in recommender systems. Social media retrieval. 2013:263-81.

Privacy

Information privacy breaches by:



may be



Question? What are some potential problems with information privacy?

Data Collection.

Data Retention.

Data Sales (or failed anonymization).

**Employee
Browsing
Private
Information.**

**Recommendations Revealing
Information.**

Information privacy - consent

Data Collection. Many users are not aware of the amount and extent of information that a service provider is able to collect, and what can be derived from this information.

Data Retention. Online information is often difficult to remove because there is commercial value in user information. Furthermore, erased information may still reside somewhere else in the system, for example in backups, to be found by others.

Information privacy - consent

Data Collection. Many users are not aware of the amount and extent of information that a service provider is able to collect, and what can be derived from this information.

Data Retention. Online information is often difficult to remove because there is commercial value in user information. Furthermore, erased information may still reside somewhere else in the system, for example in backups, to be found by others.

Data Sales (or failed anonymization).

Data sales usually conflicts with the privacy expectations of users. Cross-referencing (anonymized) datasets can reveal personally identifying information.

Employee Browsing Private Information.

Recommendations Revealing Information. For example, if you belong to the same "recommendation cluster" as another user of a known gender, ethnicity etc.

Privacy



"The Netflix Prize" competition

Background: In 2006-2009, Netflix sponsored a recommender competition on 100M movie ratings: \$1,000,000 to the team which could beat the company's baseline by 10%.

Netflix's dataset was **anonymized**, however...

Privacy



"The Netflix Prize" competition

Background: In 2006-2009, Netflix sponsored a recommender competition on 100M movie ratings: \$1,000,000 to the team which could beat the company's baseline by 10%.

Netflix's dataset was **anonymized**, however there are other movie ratings datasets to **cross-reference**.



Arvind Narayanan



Vitaly Shmatikov



Two researchers were able to **identify individual users** based on **rating similarity and time of rating** on the Internet Movie Database (IMDb).

How to break anonymity of the Netflix Prize dataset. arXiv:cs/0610105v2, 2007.

As a result: Netflix was sued in Doe v. Netflix (2009). The dataset was withdrawn and the competition was cancelled.



Ok so based on Netflix data, 3rd parties can find my
IMDb profile... So what?

Two researchers were able to **identify individual users** based on **rating similarity and time of rating** on the Internet Movie Database (IMDb).

How to break anonymity of the Netflix Prize dataset. arXiv:cs/0610105v2, 2007.

As a result: Netflix was sued in Doe v. Netflix (2009). The dataset was withdrawn and the competition was cancelled.

Privacy



How to break anonymity of the Netflix Prize dataset. arXiv:cs/0610105v2, 2007.

Consent

“Given a user’s **public** IMDb ratings, which the user posted voluntarily to selectively reveal some of his movie likes and dislikes, we discover the ratings that he entered **privately** into the Netflix system, expecting that they will remain private.”

Privacy



How to break anonymity of the Netflix Prize dataset. arXiv:cs/0610105v2, 2007.

Consent

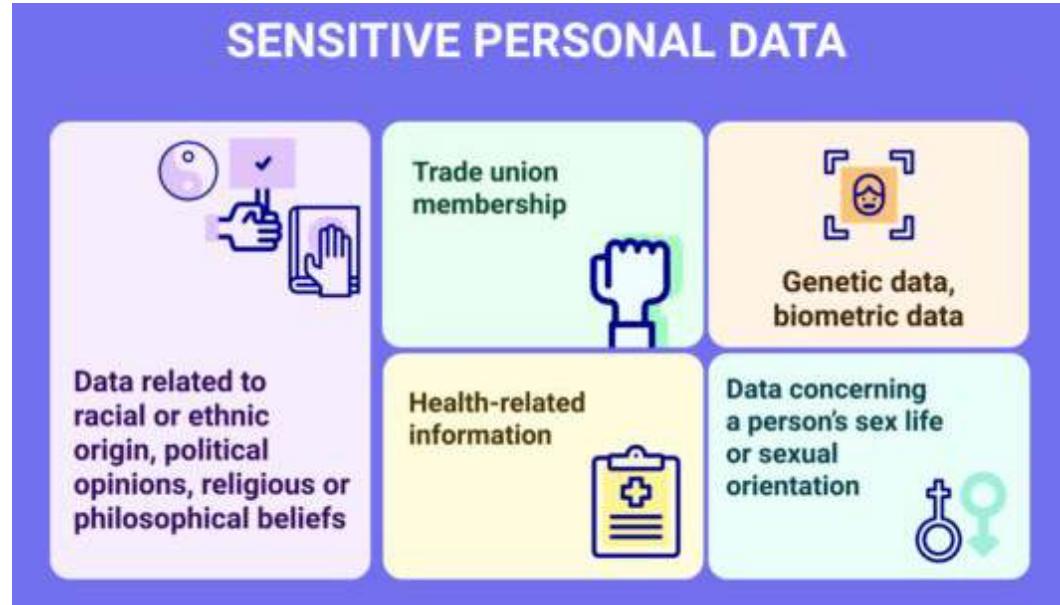
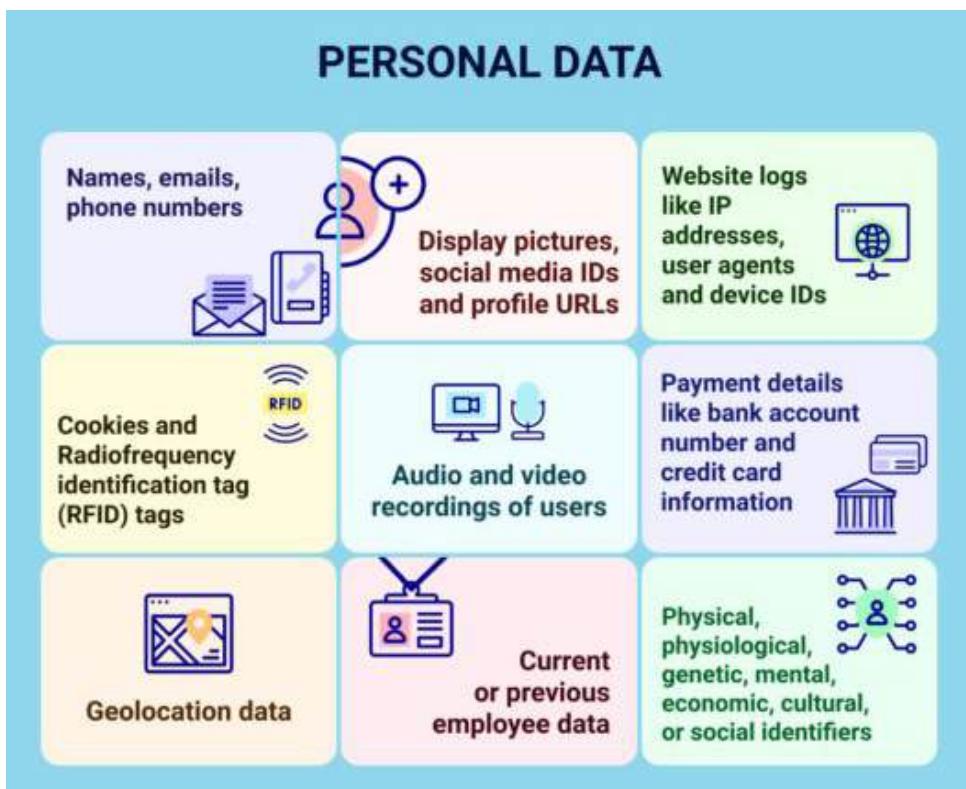
“Given a user’s **public** IMDb ratings, which the user posted voluntarily to selectively reveal some of his movie likes and dislikes, we discover the ratings that he entered **privately** into the Netflix system, expecting that they will remain private.”

Privacy of sensitive information

- There is information in a user’s **entire** movie viewing history on Netflix which is not in his **public** IMDb ratings.

“First, we can immediately find his *political orientation* based on his strong opinions about “Power and Terror: Noam Chomsky in Our Times” and “Fahrenheit 9/11.” Strong guesses about his *religious views* can be made based on his ratings on “Jesus of Nazareth” and “The Gospel of John”. He did not like “Super Size Me” at all; perhaps this implies something about his *physical size*? ... This is far from all we found about this one person, but having made our point, we will spare the reader further lurid details.”

Types of personal data in the GDPR

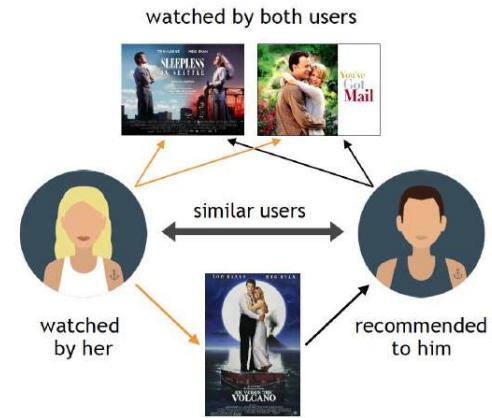


Questions?

Collaborative filtering: Alternative approaches

Matrix factorization (NMF) is just **one way** of doing collaborative filtering (CF).

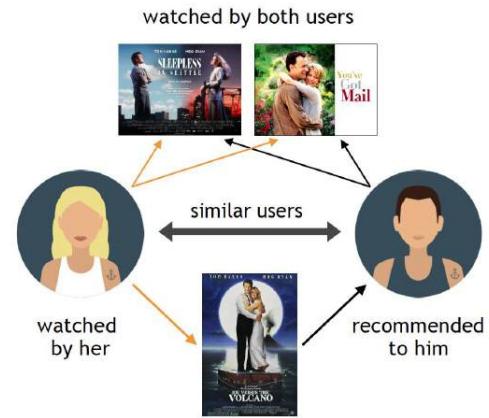
CF simply uses ratings or implicit data on a **user level**, so it's just a description of the problem, rather than any specific model.



Collaborative filtering: Alternative approaches

Matrix factorization (NMF) is just **one way** of doing collaborative filtering (CF).

CF simply uses ratings or implicit data on a **user level**, so it's just a description of the problem, rather than any specific model.



Matrix factorization is a way to generate latent features by multiplying two different matrices.

CF is the **application** of matrix factorization to identify the relationship between items and users.

CF can be performed by: e.g. Matrix Factorization, neighborhood models, neural networks, restricted Boltzmann machines, graph-based methods etc.

Collaborative filtering: Alternative approaches

Neighborhood-Based Methods for Collaborative Filtering

Compute simple **similarity measures**, e.g. Pearson correlation coefficient between **two users** x and y:

$$\text{Pearson}(\bar{X}, \bar{Y}) = \frac{\sum_{i=1}^s (x_i - \hat{x}) \cdot (y_i - \hat{y})}{\sqrt{\sum_{i=1}^s (x_i - \hat{x})^2} \cdot \sqrt{\sum_{i=1}^s (y_i - \hat{y})^2}}$$

Collaborative filtering: Alternative approaches

Neighborhood-Based Methods for Collaborative Filtering

Compute simple **similarity measures**, e.g. Pearson correlation coefficient between **two users** x and y:

$$\text{Pearson}(\bar{X}, \bar{Y}) = \frac{\sum_{i=1}^s (x_i - \hat{x}) \cdot (y_i - \hat{y})}{\sqrt{\sum_{i=1}^s (x_i - \hat{x})^2} \cdot \sqrt{\sum_{i=1}^s (y_i - \hat{y})^2}}$$

or the cosine similarity between **two users** or **two items** u and v:

$$\text{Cosine}(\bar{U}, \bar{V}) = \frac{\sum_{i=1}^s u_i \cdot v_i}{\sqrt{\sum_{i=1}^s u_i^2} \cdot \sqrt{\sum_{i=1}^s v_i^2}}.$$

Collaborative filtering: Alternative approaches

Neighborhood-Based Methods for Collaborative Filtering

Compute simple **similarity measures**, e.g. Pearson correlation coefficient between **two users** x and y:

$$\text{Pearson}(\bar{X}, \bar{Y}) = \frac{\sum_{i=1}^s (x_i - \hat{x}) \cdot (y_i - \hat{y})}{\sqrt{\sum_{i=1}^s (x_i - \hat{x})^2} \cdot \sqrt{\sum_{i=1}^s (y_i - \hat{y})^2}}$$

or the cosine similarity between **two users** or **two items** u and v:

$$\text{Cosine}(\bar{U}, \bar{V}) = \frac{\sum_{i=1}^s u_i \cdot v_i}{\sqrt{\sum_{i=1}^s u_i^2} \cdot \sqrt{\sum_{i=1}^s v_i^2}}.$$

□□ Then we can recommend **similar items to similar users** (choose k-nearest neighbors, cluster, etc.).

Collaborative filtering: Alternative approaches

Neighborhood-Based Methods for Collaborative Filtering

Compute simple **similarity measures**, e.g. Pearson correlation coefficient between **two users** x and y:

or the cosine similarity between

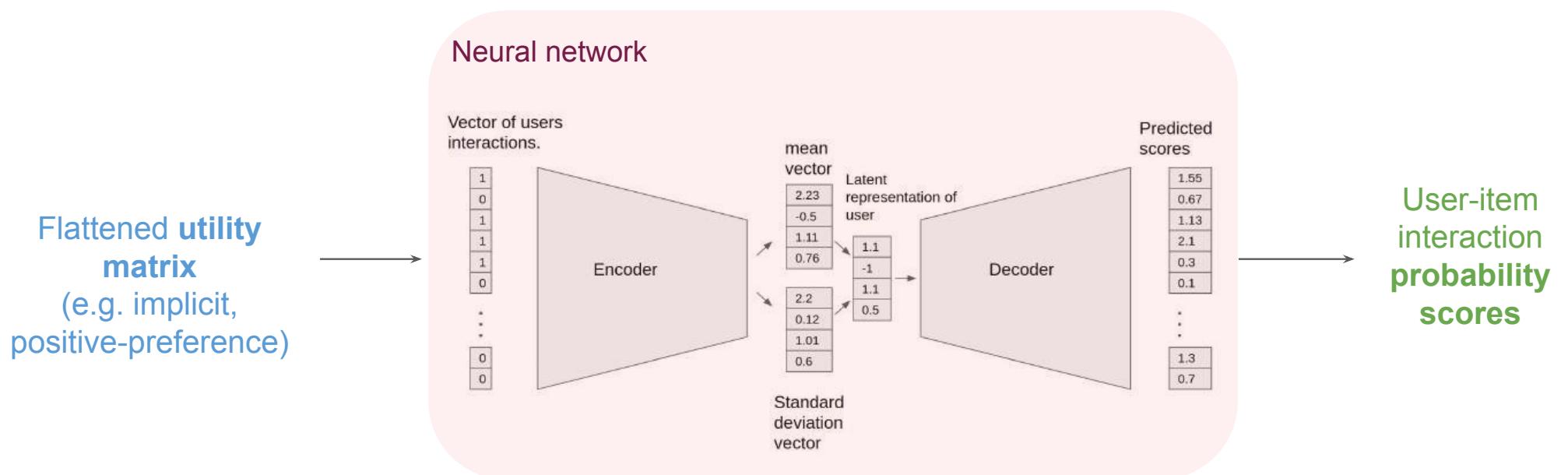
Or use hashing + Jaccard distance...

- □ Then we can recommend **similar items to similar users** (choose k-nearest neighbors, cluster, etc.).

Collaborative filtering: Alternative approaches

Neural network based models

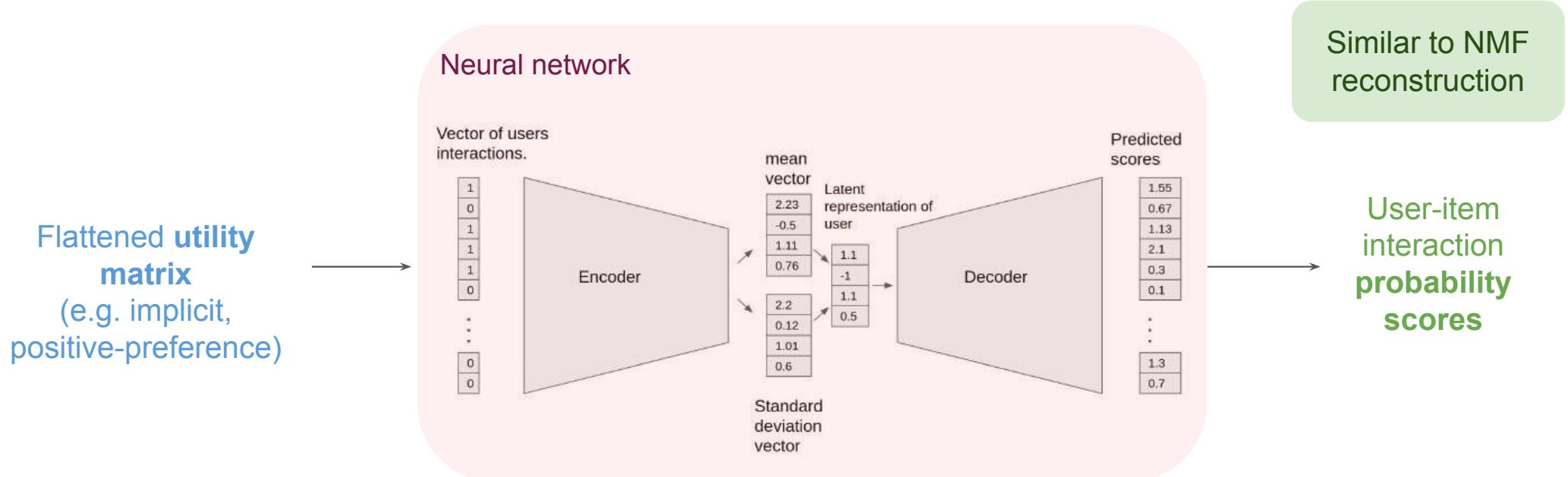
Variational Autoencoder for Collaborative Filtering



Collaborative filtering: Alternative approaches

Neural network based models

Variational Autoencoder for Collaborative Filtering



Collaborative filtering: Alternative approaches

Graph based methods

Co-clustering of user-item graph

	INTEREST GROUP A CO-CLUSTER	GLADIATOR	BEN-HUR	SPARTACUS	GODFATHER	GOODFELLS	SCARFACE
U ₁	1			1		1	
U ₄			1	1			
U ₆	1	1					
U ₂					1		1
U ₃	1				1	1	
U ₅						1	1

(a) Co-cluster

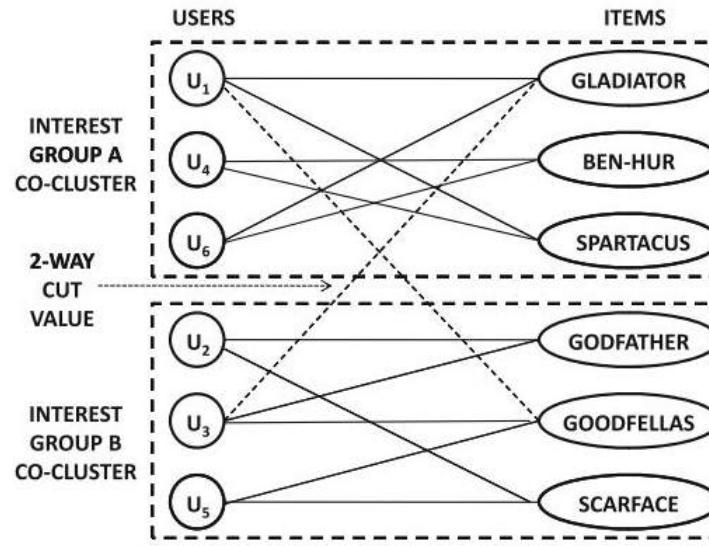
Collaborative filtering: Alternative approaches

Graph based methods

Co-clustering of user-item graph

	INTEREST GROUP A CO-CLUSTER	GLADIATOR	BEN-HUR	SPARTACUS	GODFATHER	GOODFELLAS	SCARFACE
INTEREST GROUP B CO-CLUSTER							
U ₁	1			1		1	
U ₄			1	1			
U ₆	1	1					
U ₂					1		1
U ₃	1				1	1	
U ₅						1	1

(a) Co-cluster



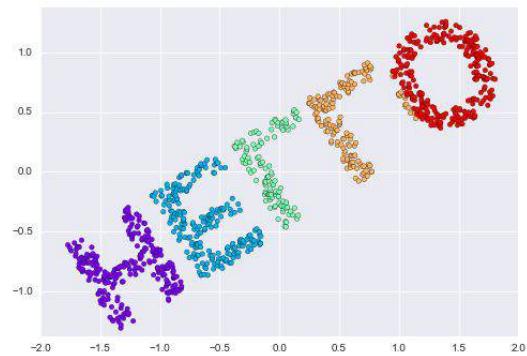
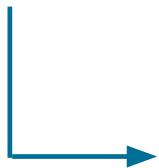
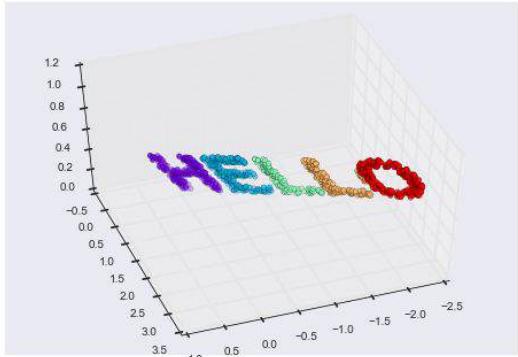
(b) User-item graph

Summary

- Recommender systems are **ubiquitous**
- There are **multiple** possible **approaches** (collaborative, content-based, hybrid)
- Different **types of utility matrices** (ranking-based, preference-based, dense, sparse, etc.)
- Important to think about **cross-validation** when picking methods and making design choices
- **Privacy** is important to consider when dealing with (big) user data
- **Alternatives** to NMF provide different methods for CF



Questions?



Dimensionality Reduction and Manifold Learning

CSE2525, Data Mining

Nergis Tomen

13.01.2025

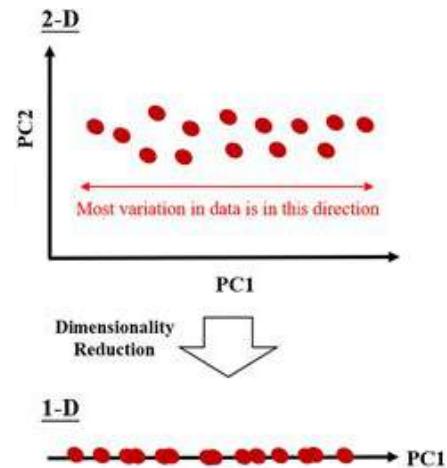
Overview

Dimensionality reduction

- When is PCA not enough?

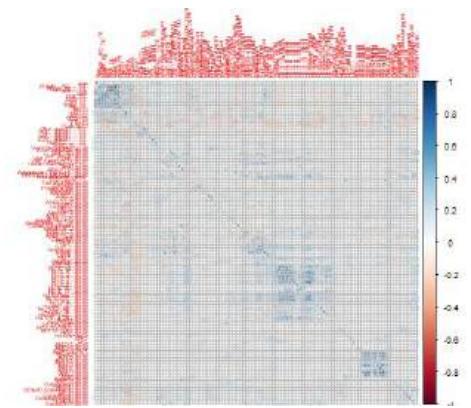
Manifold learning

- Manifolds
- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection (UMAP)



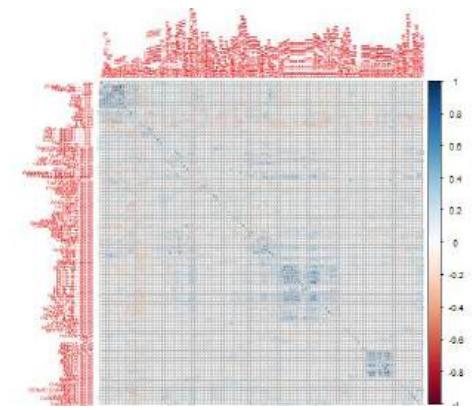
Dimensionality reduction

Why dimensionality reduction?



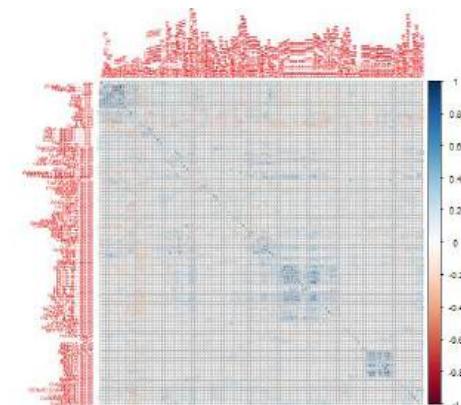
Why dimensionality reduction?

- Interesting real world data is often **high dimensional**, e.g.
 - Genetic data
 - Images and videos (dimensionality = [# of pixels \times # of frames])



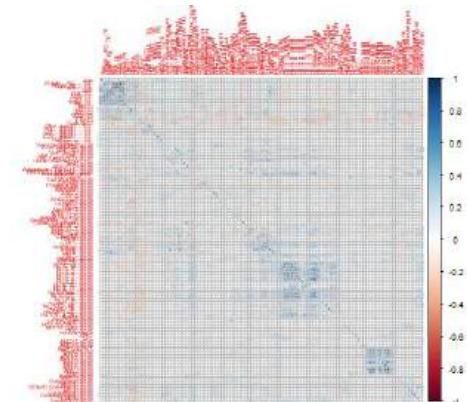
Why dimensionality reduction?

- Interesting real world data is often **high dimensional**, e.g.
 - Genetic data
 - Images and videos (dimensionality = [# of pixels × # of frames])
- It is always **useful to visualize the data** you're working with, but often **it's not possible** to meaningfully visualize high-dimensional data (beyond 3-dimensional).



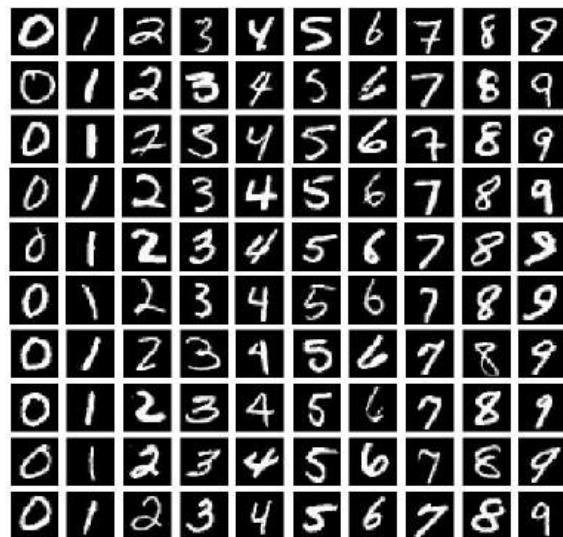
Why dimensionality reduction?

- Interesting real world data is often **high dimensional**, e.g.
 - Genetic data
 - Images and videos (dimensionality = [# of pixels × # of frames])
- It is always **useful to visualize the data** you're working with, but often **it's not possible** to meaningfully visualize high-dimensional data (beyond 3-dimensional).
- For **computational reasons**, we may want to extract and work with only the "most informative" features of a dataset ("compress" the data by getting rid of redundant or non-informative dimensions).



High-dimensional image dataset

MNIST [1]

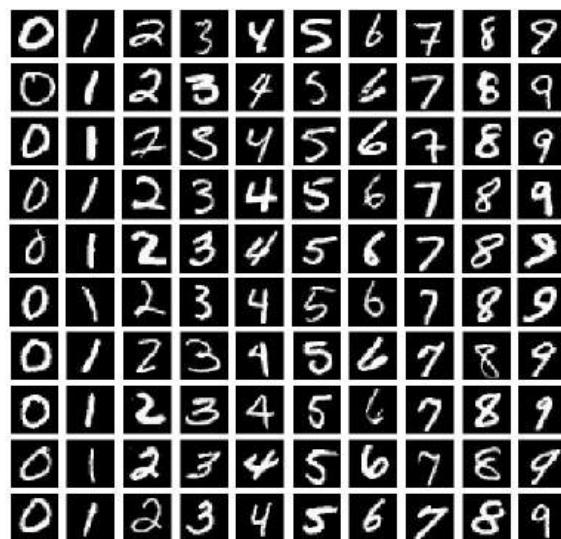


Each sample:
 28×28 pixels = 784 dimensional

Do we need 784 dimensions?

High-dimensional image dataset

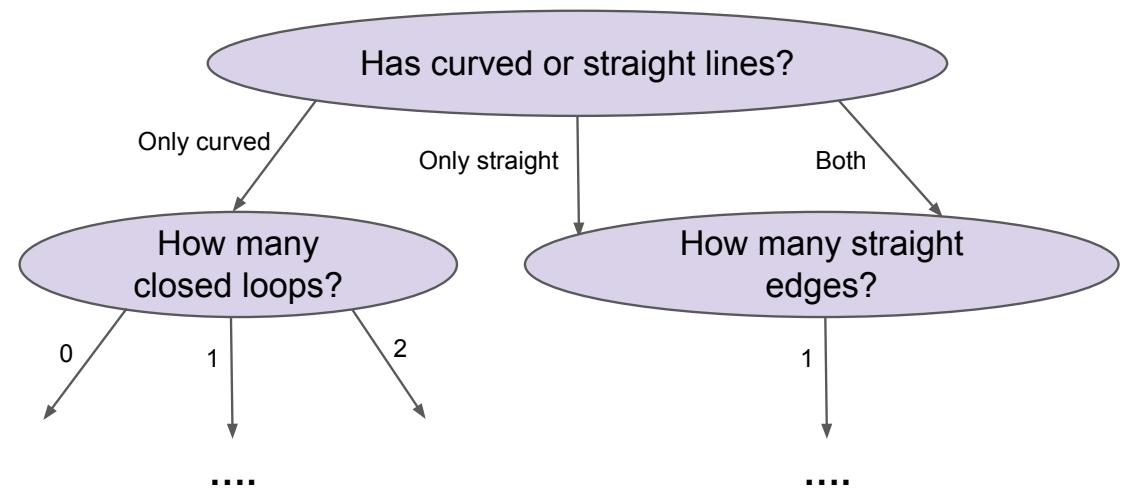
MNIST [1]



Each sample:

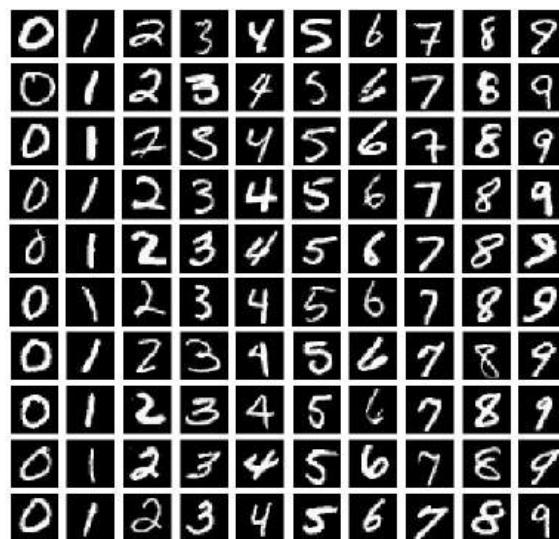
$28 \times 28 \text{ pixels} = 784 \text{ dimensional}$

As humans, we can probably find a few semantic features, which can uniquely identify each digit, e.g.



High-dimensional image dataset

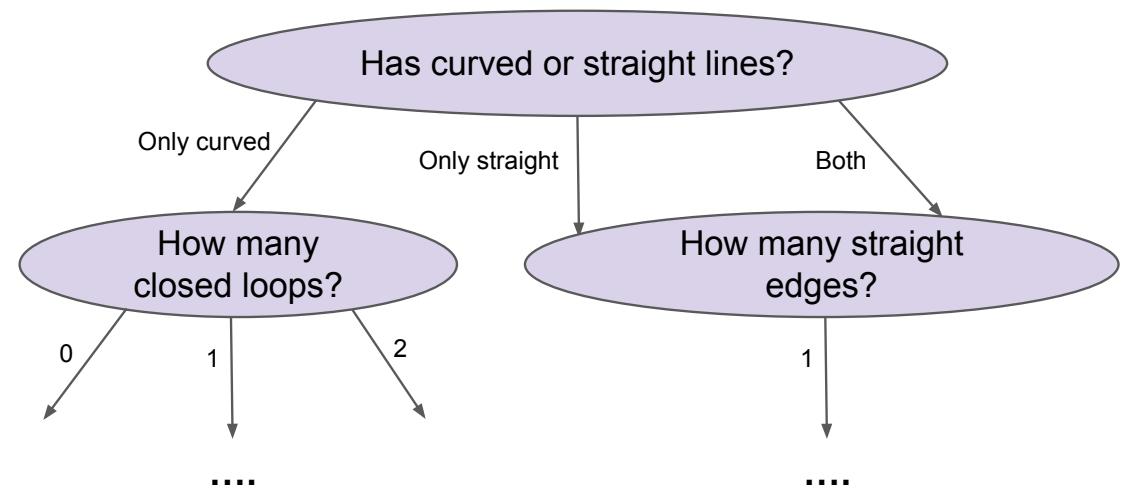
MNIST [1]



Each sample:

$28 \times 28 \text{ pixels} = 784 \text{ dimensional}$

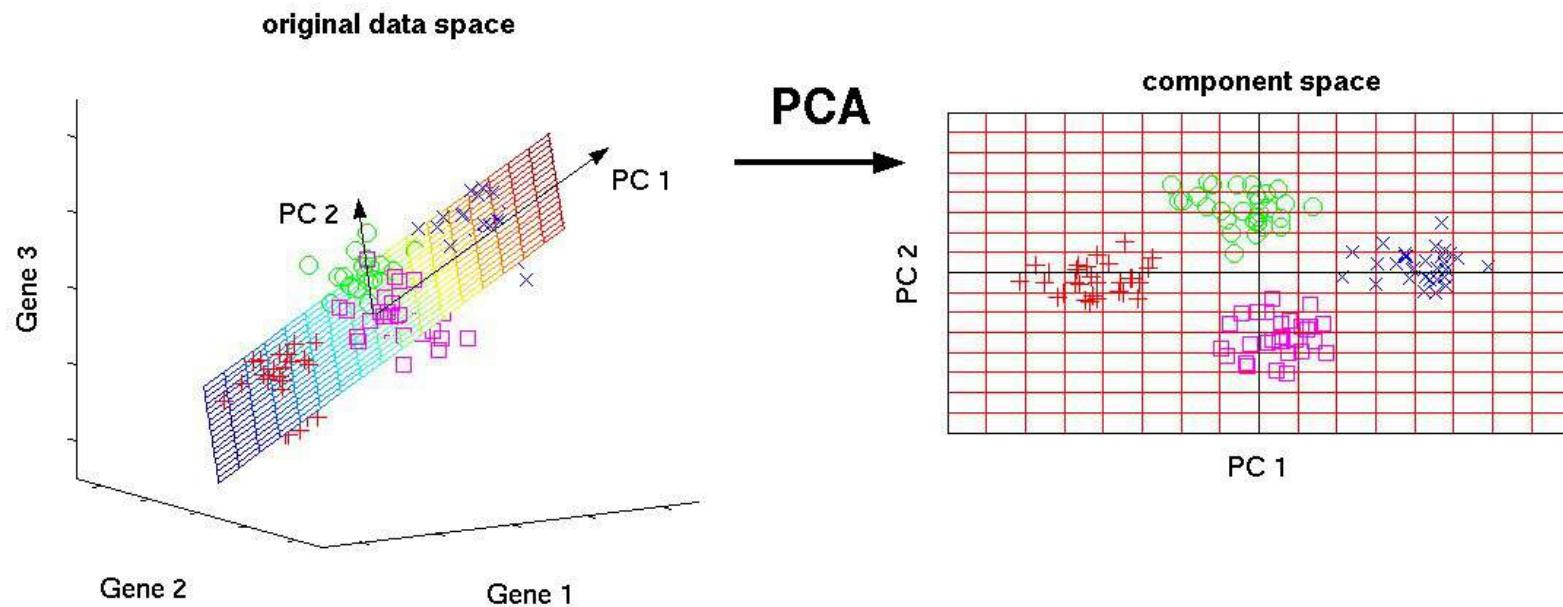
As humans, we can probably find a few semantic features, which can uniquely identify each digit, e.g.



If we can build a flow chart, or "decision tree", which can uniquely identify each digit using only a few features, do we really need 784 dimensions?

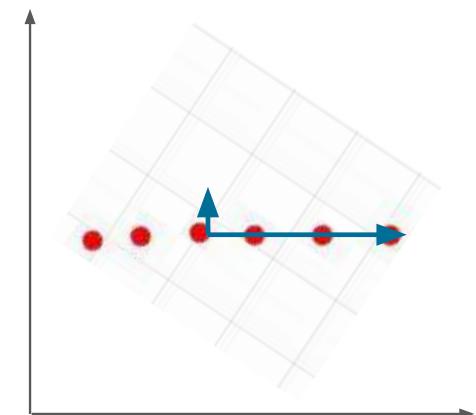
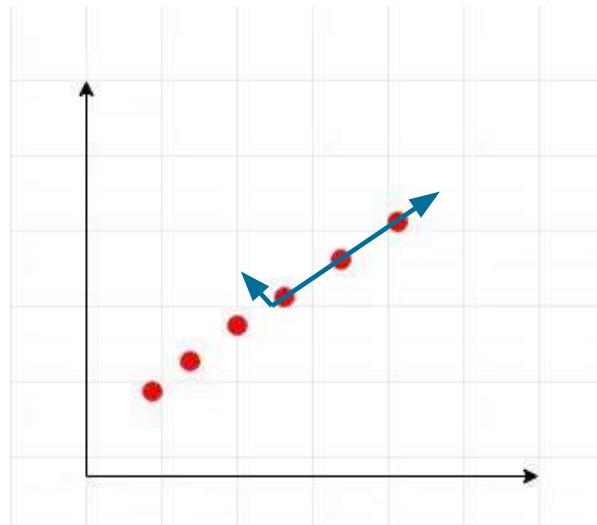
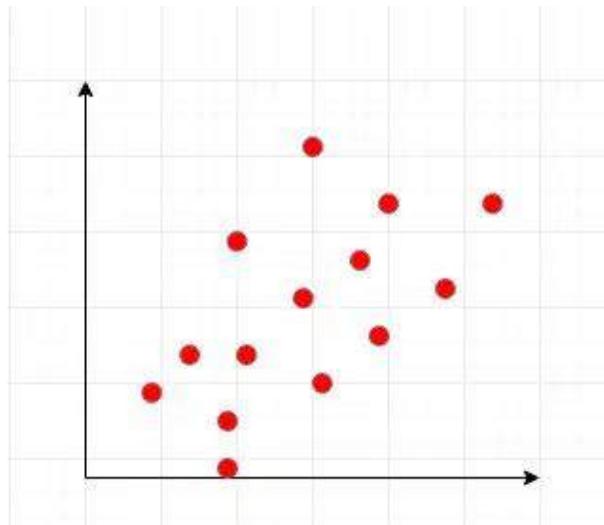
Intrinsic dimension < 784. Which dimensions are non-informative?

Principal Component Analysis (PCA)



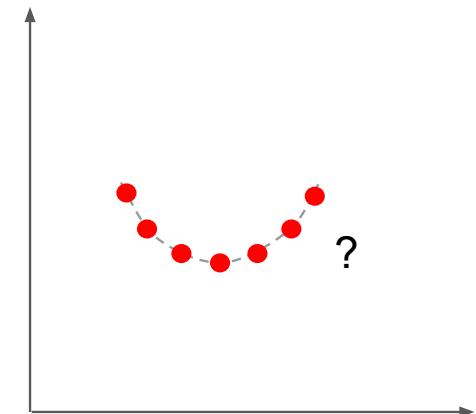
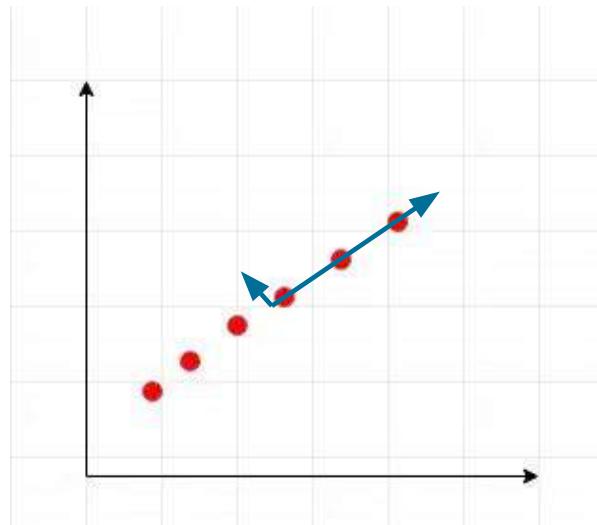
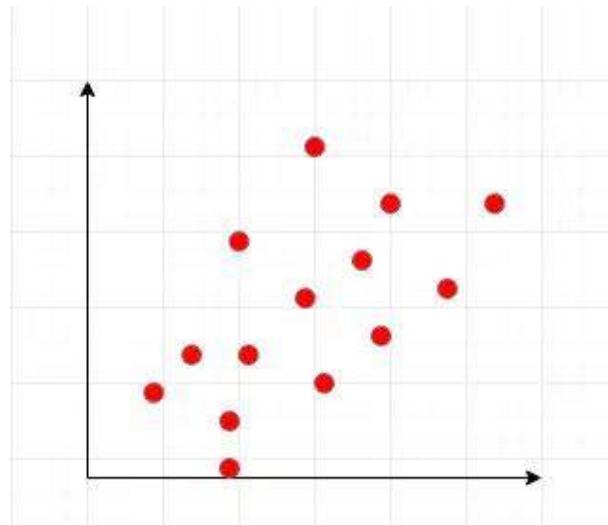
PCA

Dimensions vs. intrinsic dimensions...



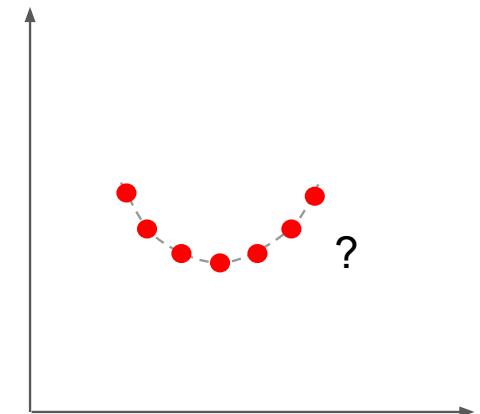
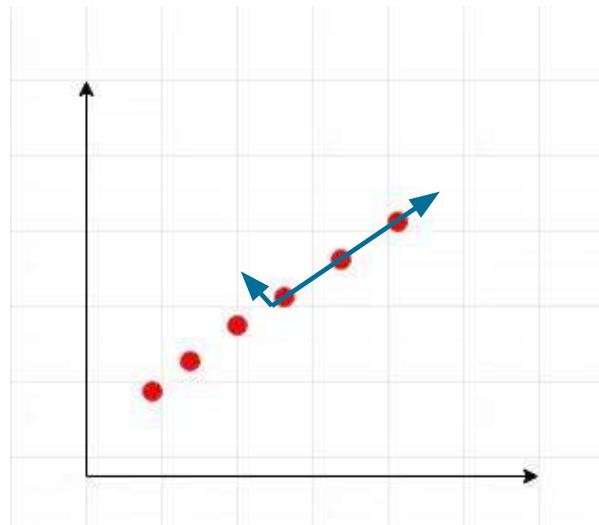
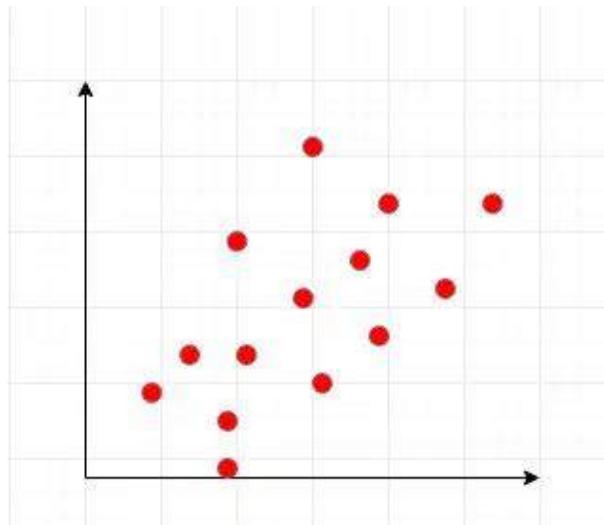
PCA

Dimensions vs. intrinsic dimensions...



PCA

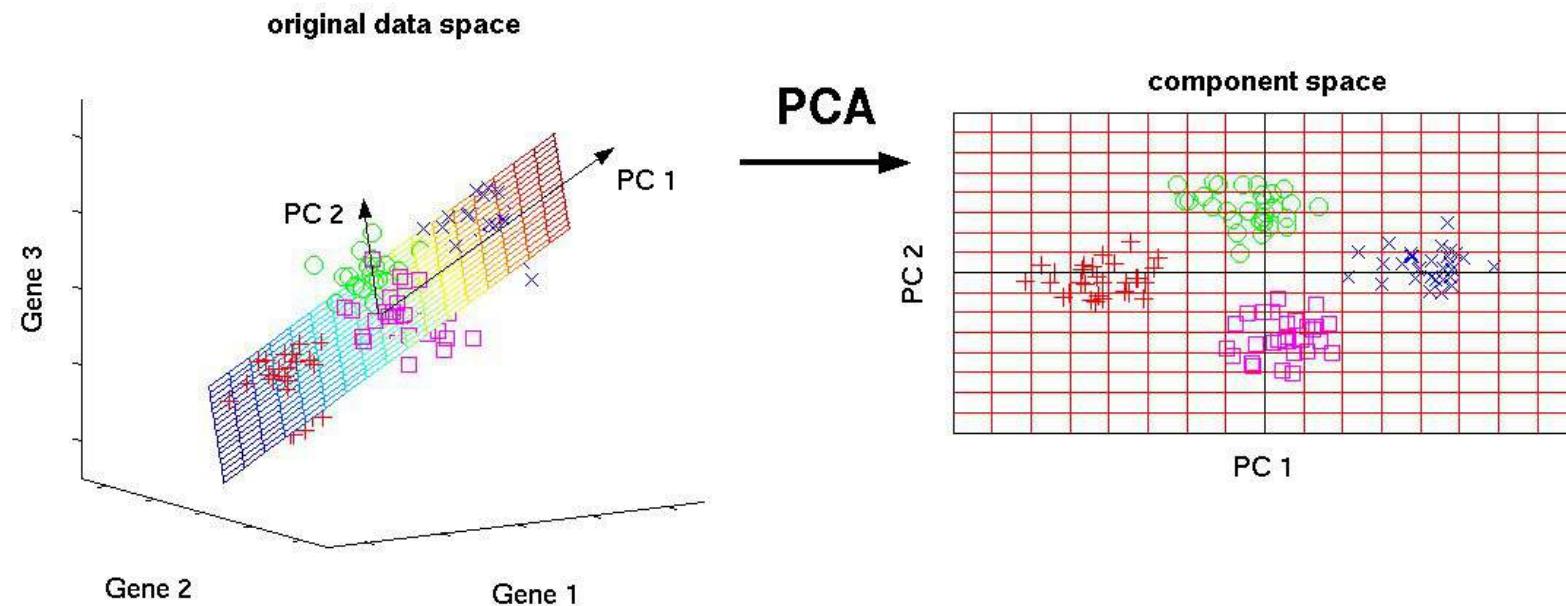
Dimensions vs. intrinsic dimensions...



Data on 1-D manifold,
PCA cannot capture it.

PCA - Assumptions

PCA performs an **orthogonal, linear transformation** (XK) to a new coordinate system (the component space).



PCA - Assumptions

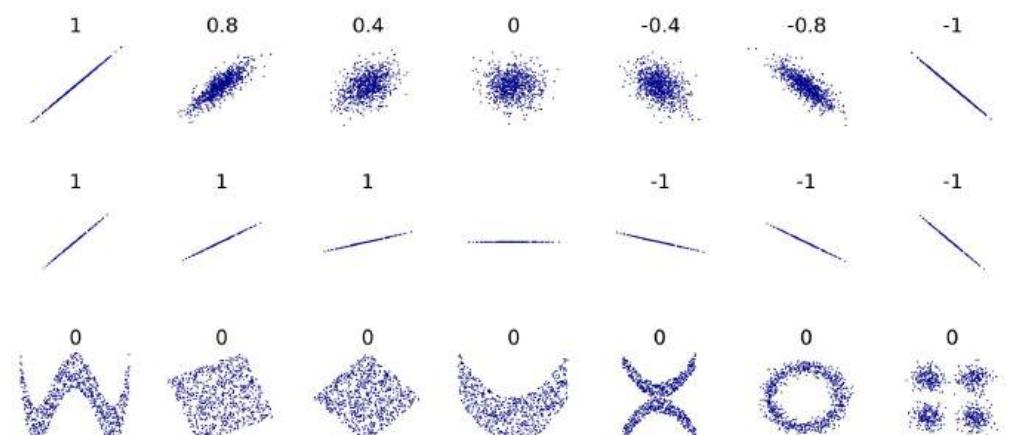
PCA performs an **orthogonal, linear transformation** (XK) to a new coordinate system (the component space).

It assumes that:

- the relationships between the variables/features are **linear** (covariance is a bi-linear operator).

However, variables can still be statistically correlated ("statistical dependence" in probability theory), **not captured by PCA!**

- [for dimensionality reduction] the direction with largest variance is the **most informative**... is that always true?



PCA

Advantages

Fast, easy to interpret, linear correlation assumption **works well** in many practical cases.

Disadvantages

Strict **assumptions** (e.g. assumes orthogonality of the 'most important' components), linear transform which **cannot capture nonlinear correlations** between variables/features, thus doesn't work very well for more complex problems.

PCA

Advantages

Fast, easy to interpret, linear correlation assumption works well in many practical cases.

Disadvantages

Strict assumptions (e.g. assumes orthogonality of the 'most important' components), linear transform which cannot capture nonlinear correlations between variables/features, thus doesn't work very well for more complex problems.

Possible alternatives (other than manifold learning)

- Independent component analysis (ICA): can capture statistical independence, can solve the cocktail-party problem,
- Kernel principal component analysis (KPCA): can capture nonlinear relationships between variables,
- Linear discriminant analysis (LDA): specialized for dimensionality reduction for classification tasks.

PCA

Advantages

Fast, easy to interpret, linear correlation assumption **works well** in many practical cases.

Disadvantages

Strict **assumptions** (e.g. assumes orthogonality of the 'most important' components), linear transform which **cannot capture nonlinear correlations** between variables/features, thus doesn't work very well for more complex problems.

Possible alternatives (other)

- Independent component analysis
- Kernel principal component analysis
- Linear discriminant analysis

Today: Manifold learning!

Flexible, relaxes most assumptions,

cocktail-party problem,
variables,
tasks.

Questions?

A class of alternative methods: Manifold learning

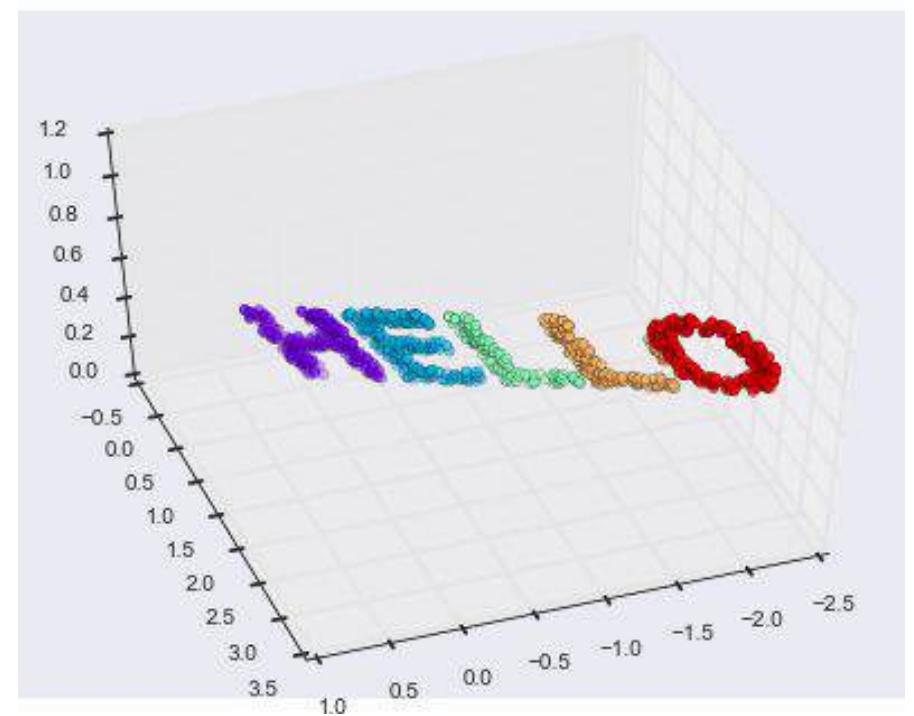
Manifold learning: Can capture **nonlinear** correlations between variables/features!

A class of alternative methods: Manifold learning

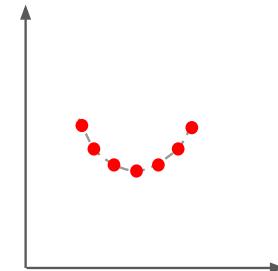
Manifold learning: Can capture **nonlinear** correlations between variables/features!

Hypothesis: The dataset we are working with contains a lot of "unnecessary dimensions".

Data on 2-D manifold embedded in 3-D space



A class of alternative methods: Manifold learning

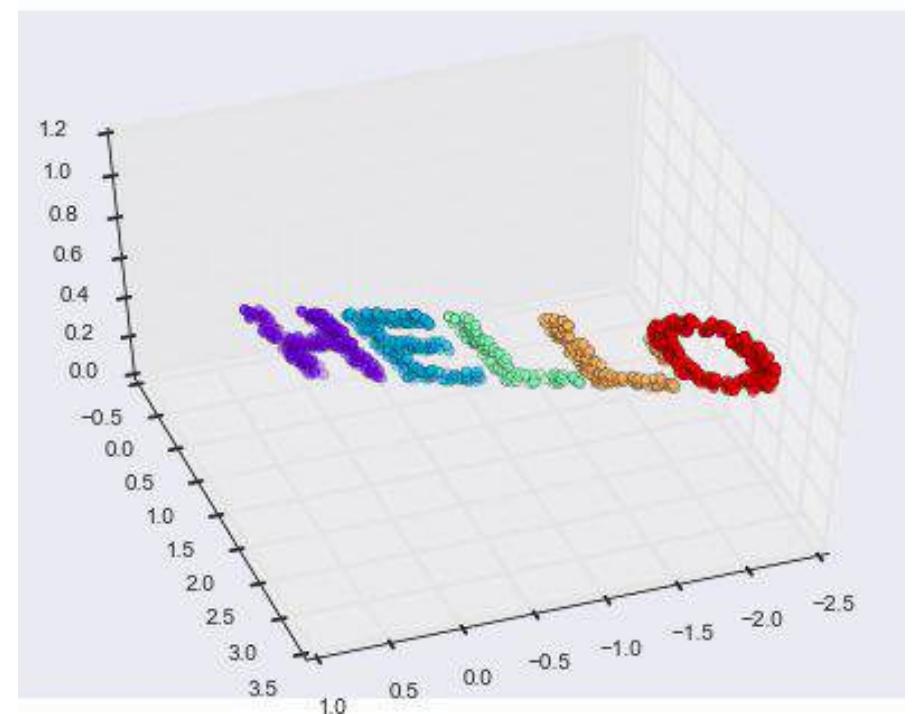


Manifold learning: Can capture **nonlinear** correlations between variables/features!

Hypothesis: The dataset we are working with contains a lot of "unnecessary dimensions".

Specifically, we assume that our data lies on a **low-dimensional** manifold (or set of manifolds) embedded in a high-dimensional space.

Data on 2-D manifold embedded in 3-D space



What is a "manifold"?

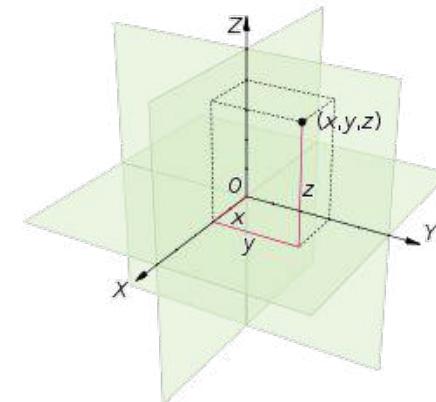
"In mathematics, a manifold is a topological space that **locally resembles** Euclidean space near each point.

What is a "manifold"?

"In mathematics, a manifold is a topological space that **locally resembles** Euclidean space near each point. More precisely, an n -dimensional manifold is a topological space with the property that each point has a neighborhood that is homeomorphic (**continuous and invertible mapping**) to an open subset of n -dimensional Euclidean space."

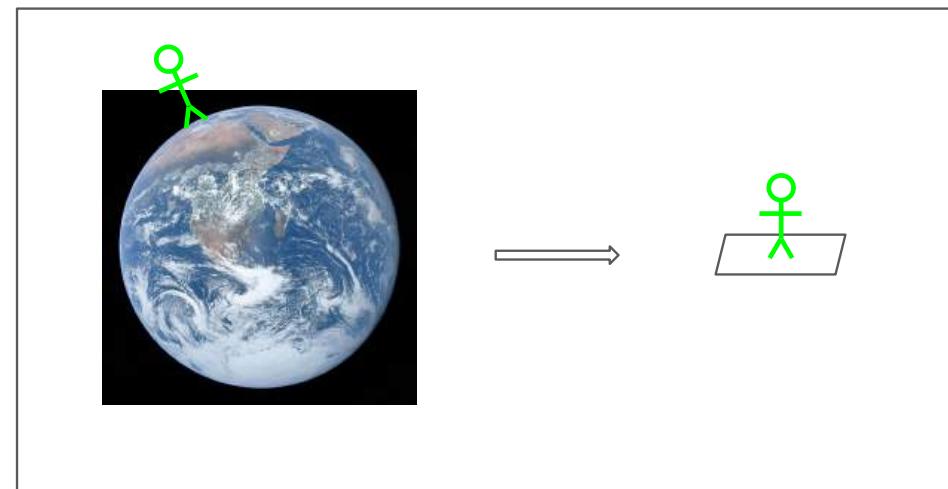
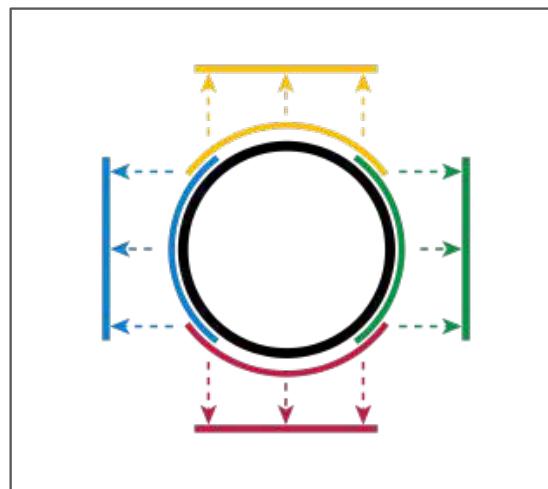
Euclidean space:

- "Real-world" space
- Cartesian or polar coordinates
- Only 1 line passing through 2 points
- Parallel lines don't cross
- Distances satisfy the triangle inequality
- Inner product of orthogonal vectors is 0
- ... etc.



Manifolds

"In mathematics, a manifold is a topological space that **locally resembles** Euclidean space near each point. More precisely, an n -dimensional manifold is a topological space with the property that each point has a neighborhood that is homeomorphic (**continuous and invertible mapping**) to an open subset of n -dimensional Euclidean space."



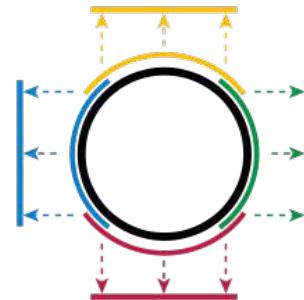
Manifolds

Examples:

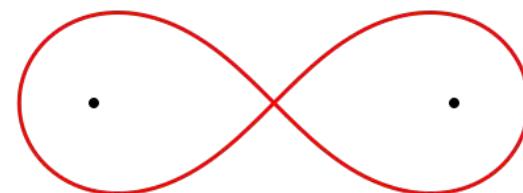
- (1-D manifold) Line segment



- (1-D manifold) Circle



- (NOT 1-D manifold) Crossing lines



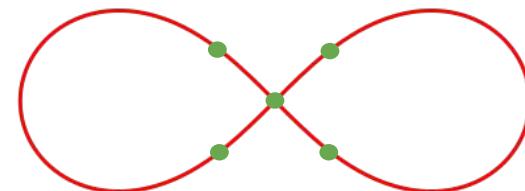
Manifolds

Examples:

- 1-D Euclidean space



No homeomorphic mapping



- (NOT 1-D manifold) Crossing lines

“Manifold hypothesis”

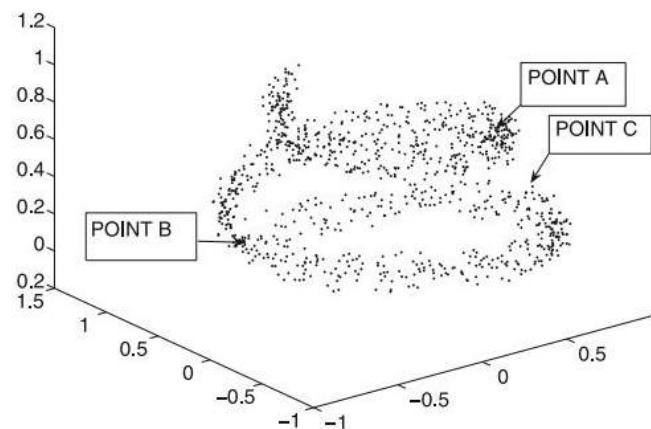
"In general, high-dimensional data are aligned along nonlinear low-dimensional shapes, which are also referred to as **manifolds**.

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

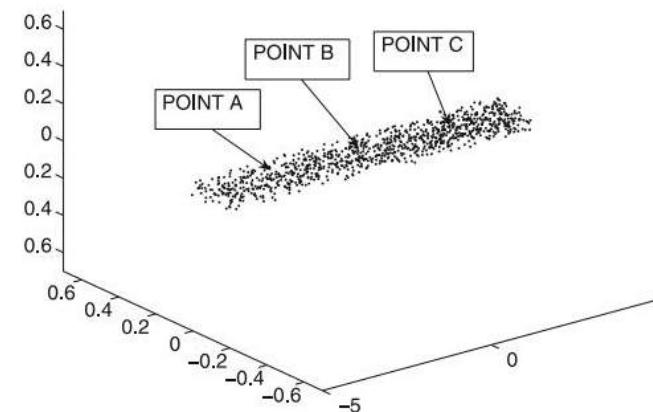
Do we need 784 dimensions?
Probably not...

“Manifold hypothesis”

"In general, high-dimensional data are aligned along nonlinear low-dimensional shapes, which are also referred to as **manifolds**. [Idea:] These manifolds can be “**flattened out**” to a new representation where metric distances can be used effectively."



(a) A and C seem close
(original data)



(b) A and C are actually far away
(ISOMAP embedding)

“Manifold hypothesis”



“Manifold hypothesis”

Useful for:

e.g.

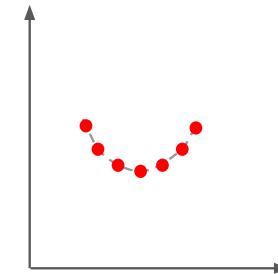
dimensionality reduction,
data visualization,
generative models...



Summary and outlook

- › Dimensionality reduction/visualization of **high-dimensional data** is important!
- › PCA can only capture **linear relationships**.

- › **Manifold learning** can capture non-linear relationships.



- › **Next time:** → t-Distributed Stochastic Neighbor Embedding (t-SNE)
 - Uniform Manifold Approximation and Projection (UMAP)

Evaluation

Dear Student,

Your opinion counts!

We hope you are willing to letting us know what you think about the teaching methods, online tools, course organization, assessment, etc. of this course. With your feedback we can further improve our education. Therefore, we kindly ask you to take 5-10 minutes time to fill in the questionnaire.

Please follow the link or scan the QR code below to open the questionnaire.

In order to obtain a reliable evaluation result, we hope that many of you will complete this questionnaire. Your answers will be processed anonymously and handled confidentially.

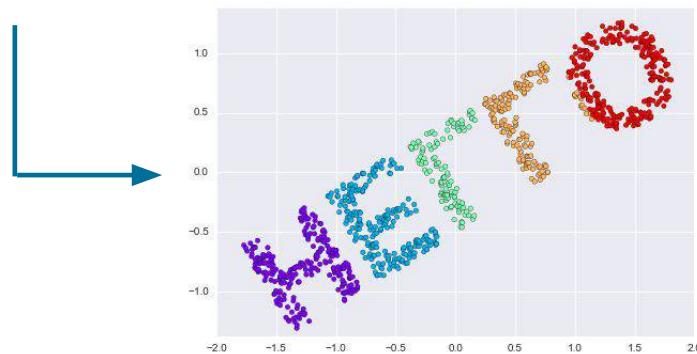
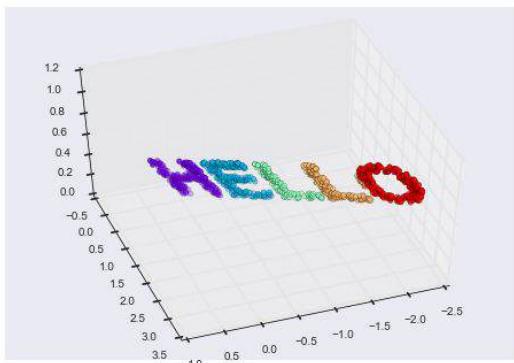
<https://evasys-survey.tudelft.nl/evasys/online.php?p=4MRLK>

Many thanks in advance for your feedback! Later on, a summary of the evaluation results will be published on the Brightspace page of your program (go to Content tab – Course evaluations).

Questions about the survey? Mail to QualityAssurance-EEMCS@tudelft.nl



Lecture 14	Week 2.8	Jan 13	Recommender systems & Manifold learning
Lecture 15 (Today)	Week 2.8	Jan 16	Dimensionality reduction
Lecture 16	Week 2.9	Jan 20	Exam preparation, Q&A
Lecture 17	Week 2.9	Jan 23	FREE
Exam	Week 2.10	Jan 27	Weblab exam



Manifold Learning

CSE2525, Data Mining

Nergis Tomen

16.01.2025

Overview

Manifold learning

- Manifolds
- t-Distributed Stochastic Neighbor Embedding (t-SNE)
- Uniform Manifold Approximation and Projection (UMAP)

Manifolds

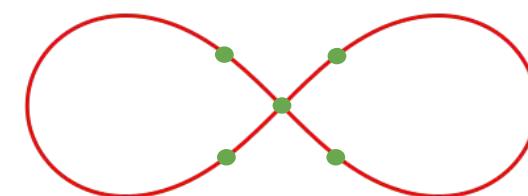
Examples:

- 1-D Euclidean space



No homeomorphic mapping

- (NOT 1-D manifold) Crossing lines



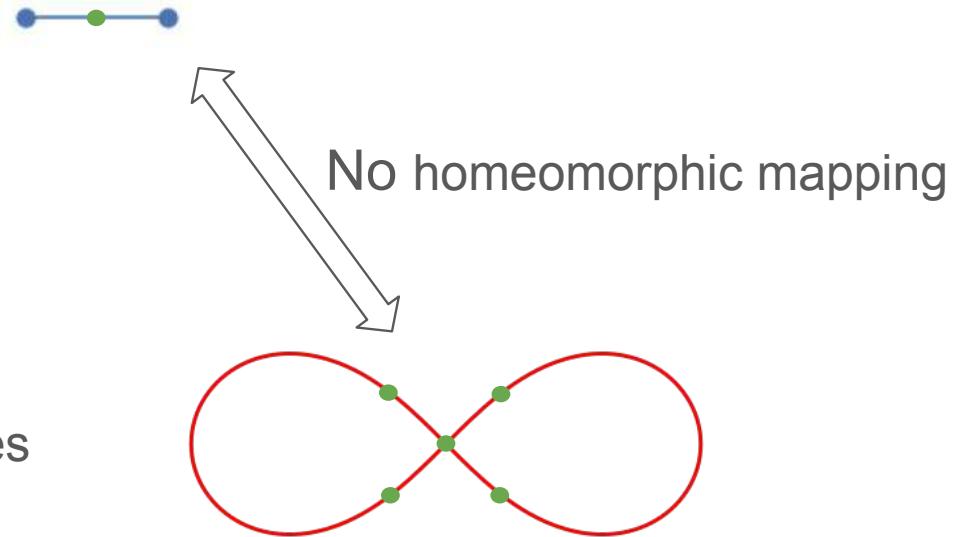
"In mathematics, a manifold is a topological space that **locally resembles** Euclidean space near each point. More precisely, an n -dimensional manifold is a topological space where each point has a neighborhood that is homeomorphic (**continuous and invertible mapping**) to an open subset of n -dimensional Euclidean space."

Manifolds

Examples: → 1-D Euclidean space

→ (NOT 1-D manifold) Crossing lines

Q: What if one line goes
“under” the other line?



"In mathematics, a manifold is a topological space that **locally resembles** Euclidean space near each point. More precisely, an n -dimensional manifold is a topological space where each point has a neighborhood that is homeomorphic (**continuous and invertible mapping**) to an open subset of n -dimensional Euclidean space."

Extra information:

See link below for a more formal definition of manifolds:

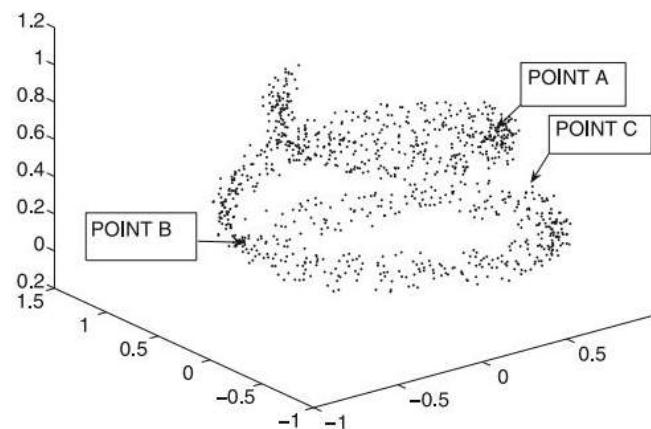
An n-dimensional **topological manifold** M is a topological Hausdorff space with a countable base which is locally homeomorphic to \mathbb{R}^n . This means that for every point p in M there is an open neighbourhood U of p and a homeomorphism $\varphi : U \rightarrow V$ which maps the set U onto an open set $V \subset \mathbb{R}^n$. Additionally:

- The mapping $\varphi : U \rightarrow V$ is called a **chart** or **coordinate system**.
- The set U is the **domain** or **local coordinate neighbourhood** of the chart.
- The image of the point $p \in U$, denoted by $\varphi(p) \in \mathbb{R}^n$, is called the **coordinates** or **local coordinates** of p in the chart.
- A set of charts, $\{\varphi_\alpha | \alpha \in \mathbb{N}\}$, with domains U_α is called the **atlas** of M , if $\bigcup_{\alpha \in \mathbb{N}} U_\alpha = M$.

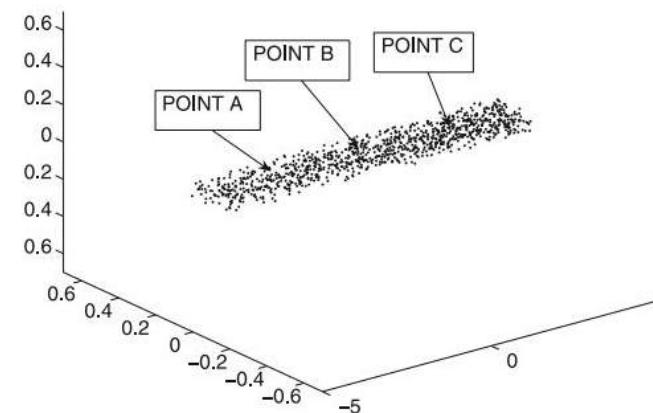
“Manifold hypothesis”

Concept: High-dimensional data might lie on nonlinear low-dimensional **manifolds**.

Application: These manifolds can be “**flattened out**” to a new representation where metric distances can be used effectively.”



(a) A and C seem close
(original data)



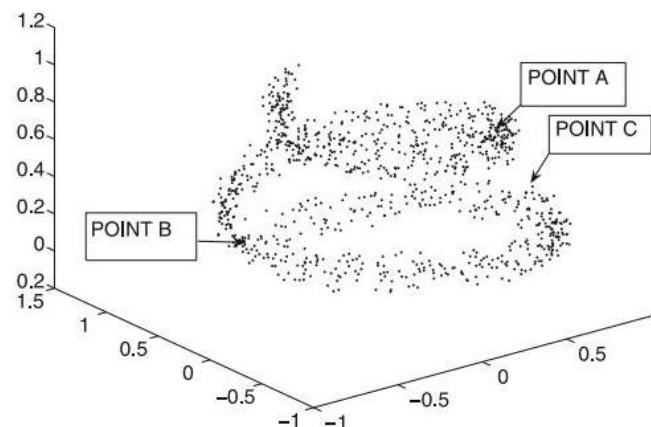
(b) A and C are actually far away
(*ISOMAP* embedding)

“Manifold hypothesis”

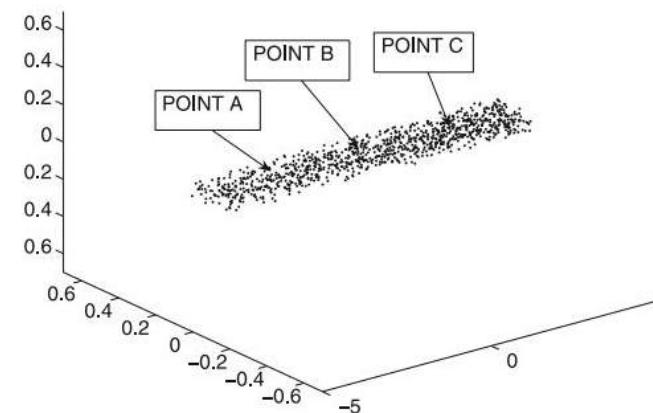
Concept: High-dimensional data might lie on nonlinear low-dimensional **manifolds**.

Application: These manifolds can be “**flattened out**” to a new representation where metric distances can be used effectively.”

Points which
are nearby in
high-D space
might not be
related
semantically.



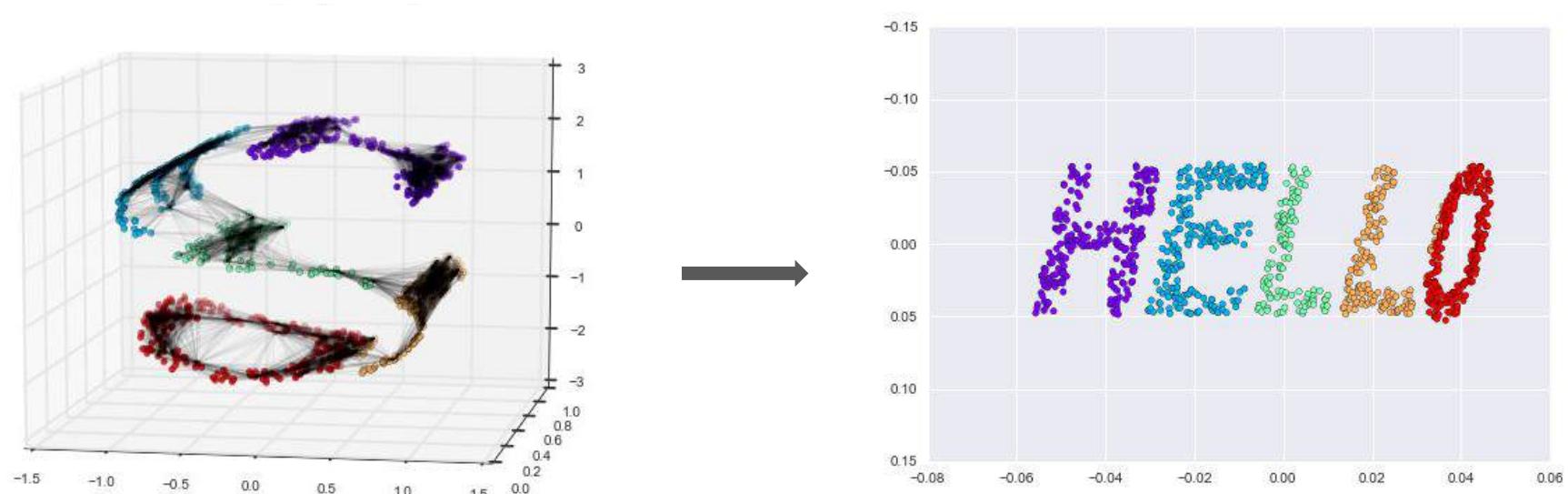
(a) A and C seem close
(original data)



(b) A and C are actually far away
(ISOMAP embedding)

“Manifold hypothesis”

Manifolds can be “**flattened out**” to a new representation where metric distances can be used effectively.

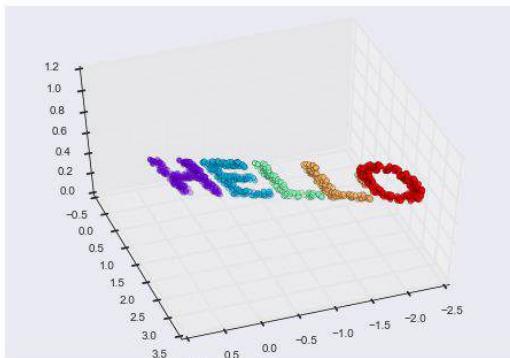


Questions?

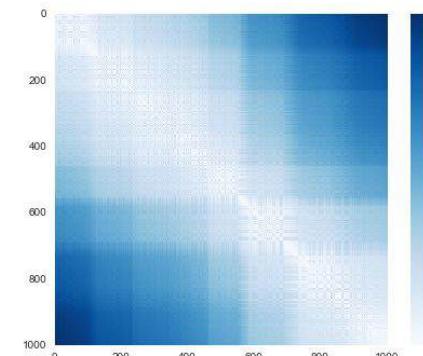
A class of alternative methods: Manifold learning

What does it do?: Manifold learning "estimates" a **low-dimensional manifold**, based **only on distances** between the high-dimensional data.

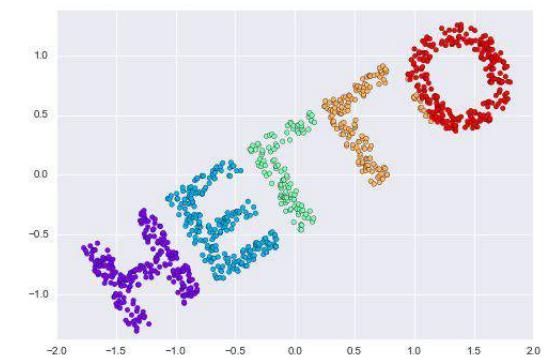
Data embedded in
3-D space



Pairwise distance matrix



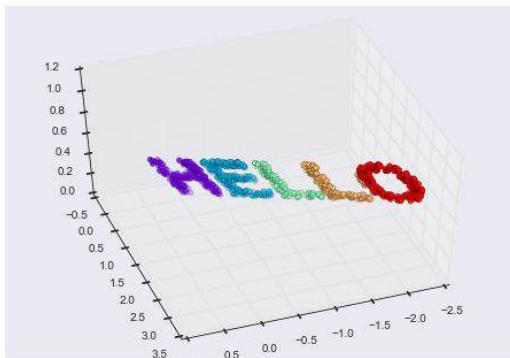
Estimated 2-D embedding



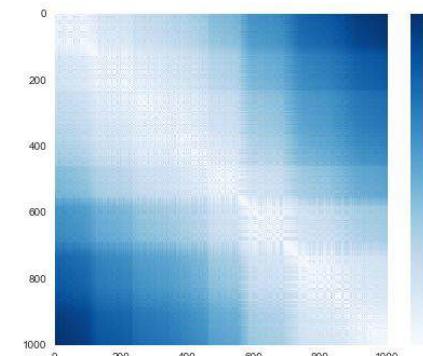
A class of alternative methods: Manifold learning

What does it do?: Manifold learning "estimates" a **low-dimensional manifold**, based **only on distances** between the high-dimensional data.

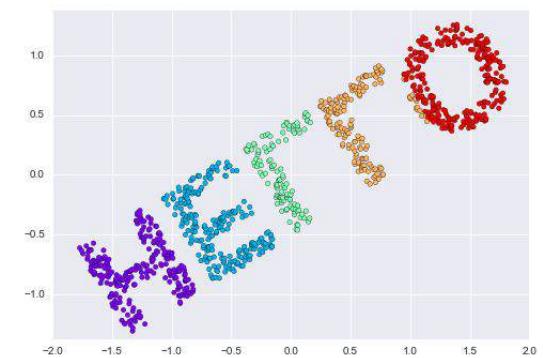
Data embedded in
3-D space



Pairwise distance matrix



Estimated 2-D embedding

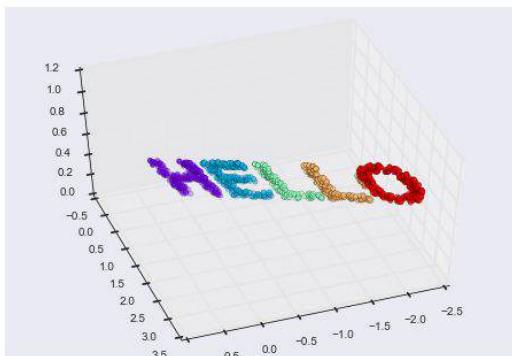


(Different algorithms might use different distance metrics, consider different 'neighborhoods' in high-dimensional space, etc. depending on their specific goals.)

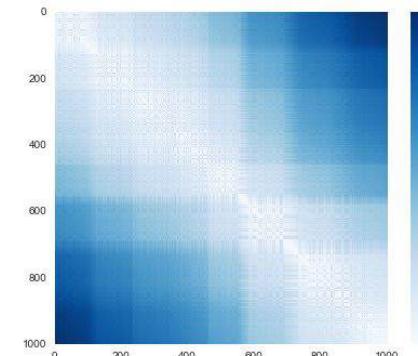
A class of alternative methods: Manifold learning

Question: What do you notice in the 2-D embedding below?

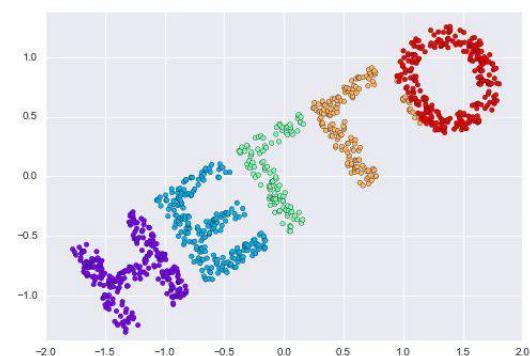
Data embedded in
3-D space



Pairwise distance matrix



Estimated 2-D embedding

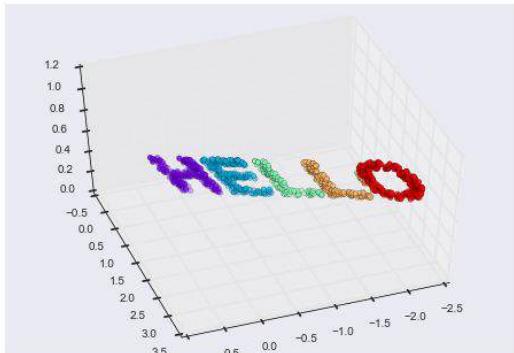


A class of alternative methods: Manifold learning

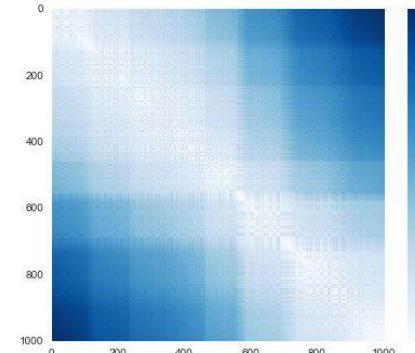
Question: What do you notice in the 2-D embedding below?

Translation invariance, rotation invariance, flipping invariance, etc.

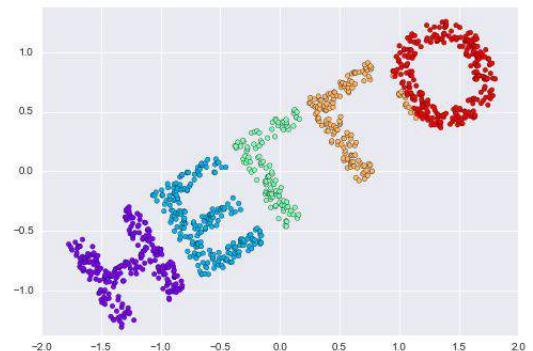
Data embedded in
3-D space



Pairwise distance matrix

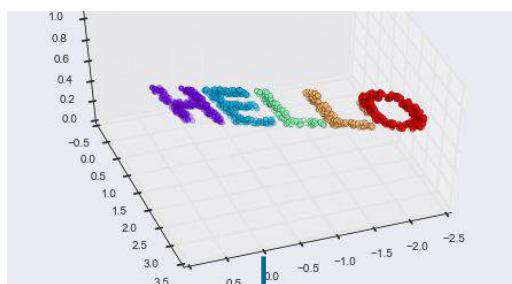


Estimated 2-D embedding

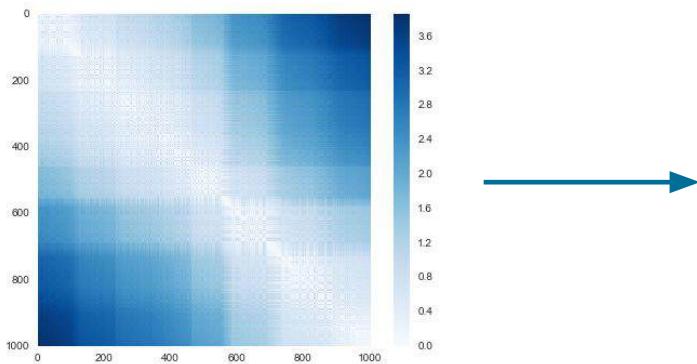


Manifold learning algorithms

Data embedded in
3-D space

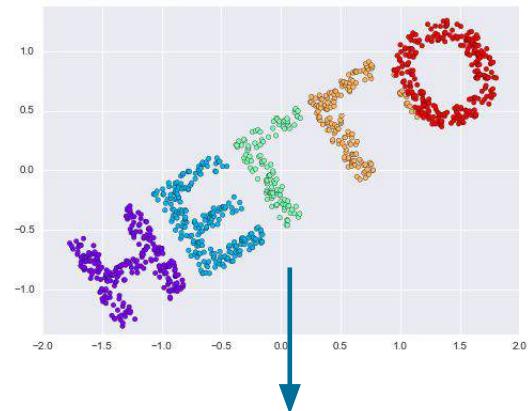


Pairwise distance in 3-D

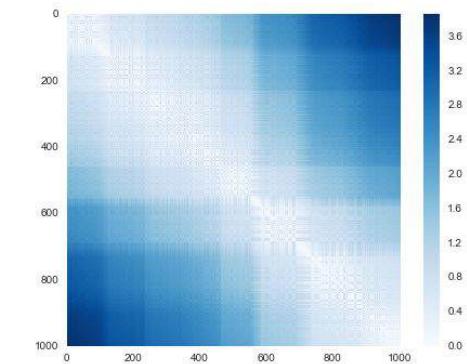


Minimize the difference!

Estimated 2-D embedding



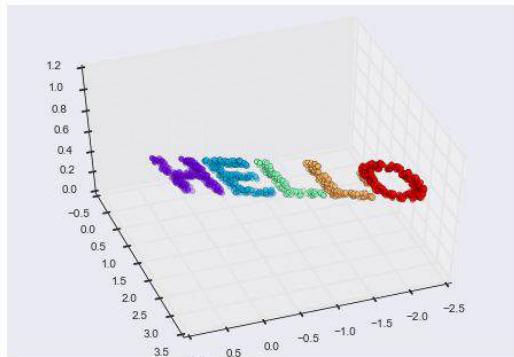
Pairwise distance in 2-D



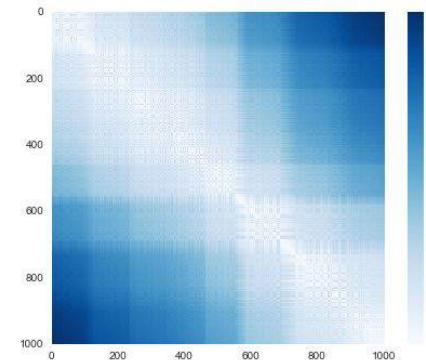
A class of alternative methods: Manifold learning

Disclaimer: 1) In manifold learning, **we assume** that our data lies on a low-dimensional manifold.

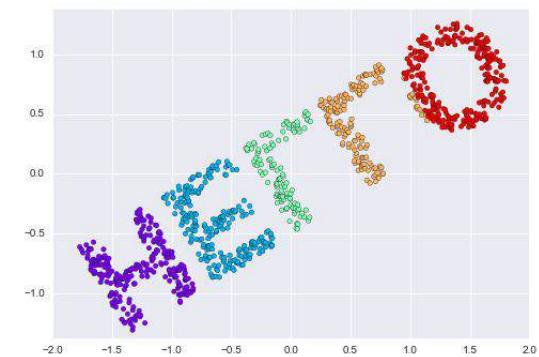
Data embedded in
3-D space



Pairwise distance matrix

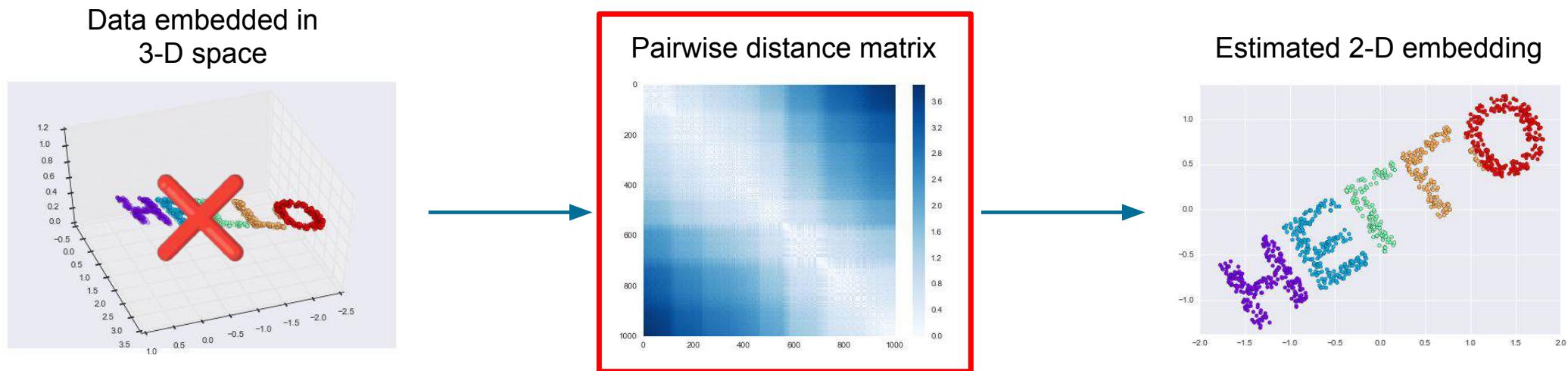


Estimated 2-D embedding



A class of alternative methods: Manifold learning

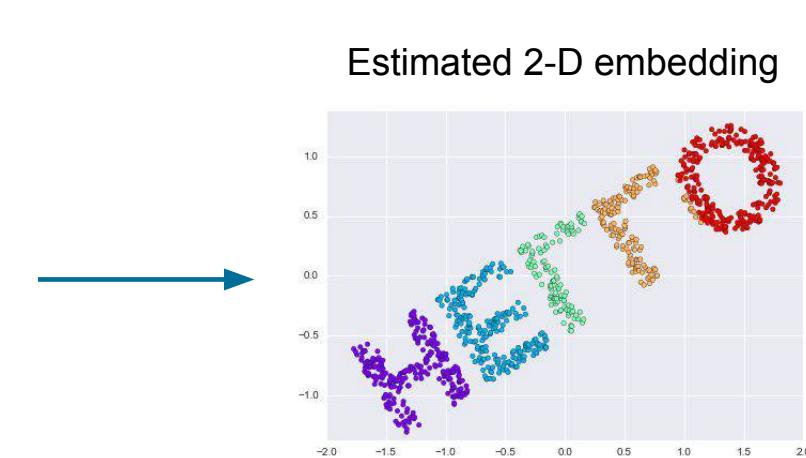
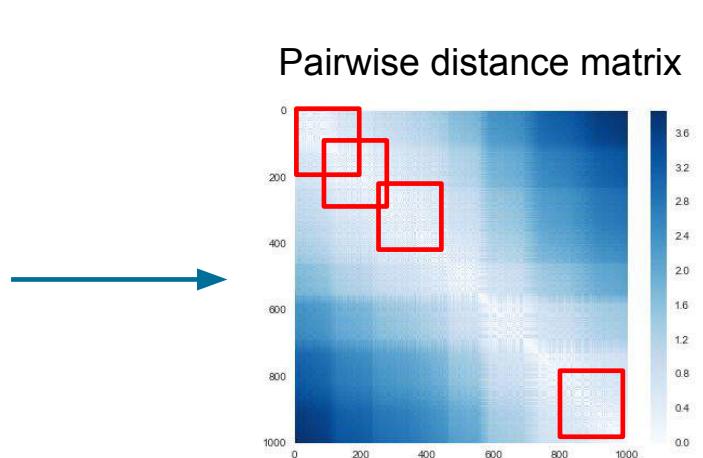
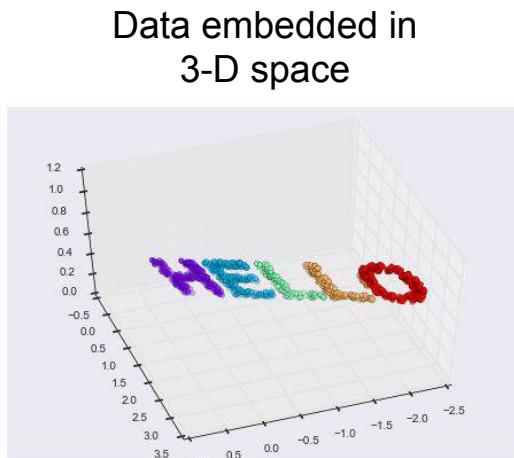
- Disclaimer:** 1) In manifold learning, **we assume** that our data lies on a low-dimensional manifold.
- 2) The algorithm only "sees" **the distance matrix**.



A class of alternative methods: Manifold learning

Disclaimer: 1) In manifold learning, **we assume** that our data lies on a low-dimensional manifold.

2) The algorithm only "sees" **the distance matrix**. It might only focus on "local" rather than "global" distances/neighbors. Low-dimensional embeddings might preserve only local, but not global, distances.

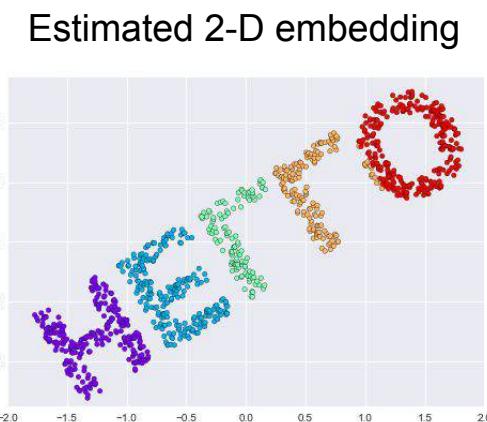
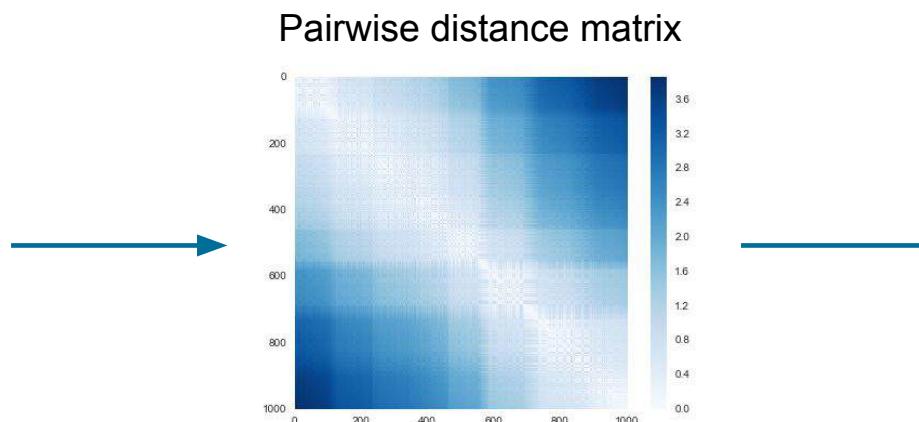
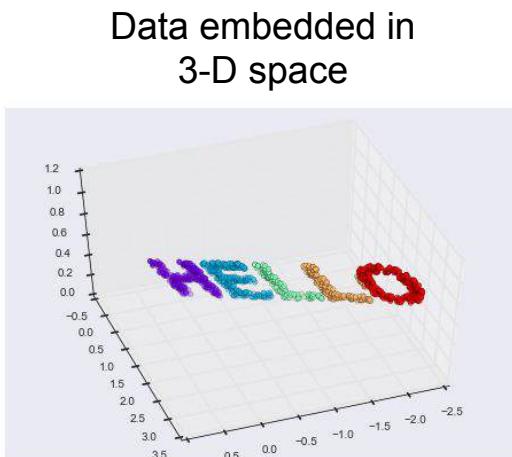


A class of alternative methods: Manifold learning

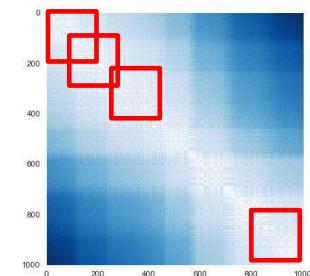
Disclaimer: 1) In manifold learning, **we assume** that our data lies on a low-dimensional manifold.

2) The algorithm only "sees" **the distance matrix**. It might only focus on "local" rather than "global" distances/neighbors. Low-dimensional embeddings might preserve only local, but not global, distances.

Therefore, manifold methods are typically recommended only for visualization and not downstream applications.

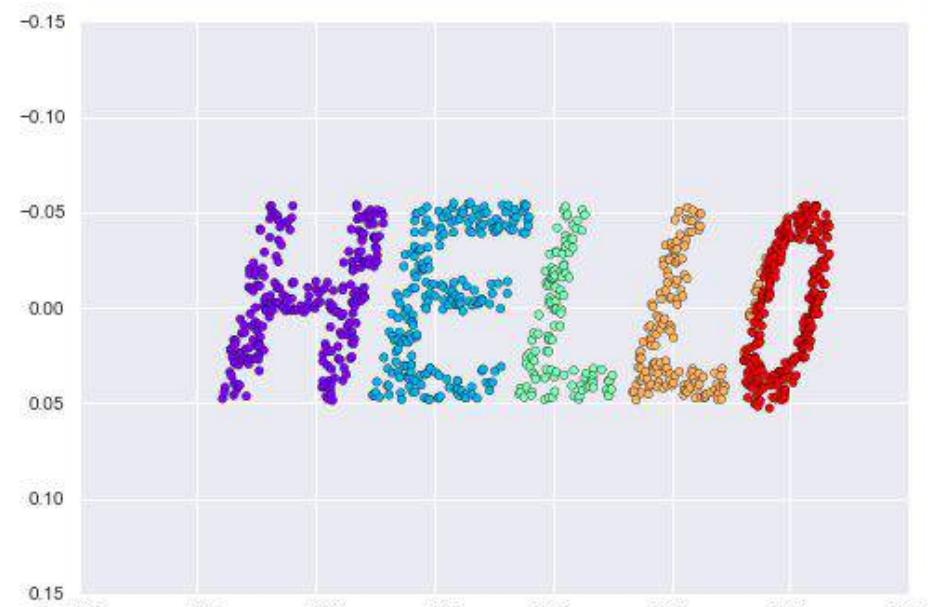
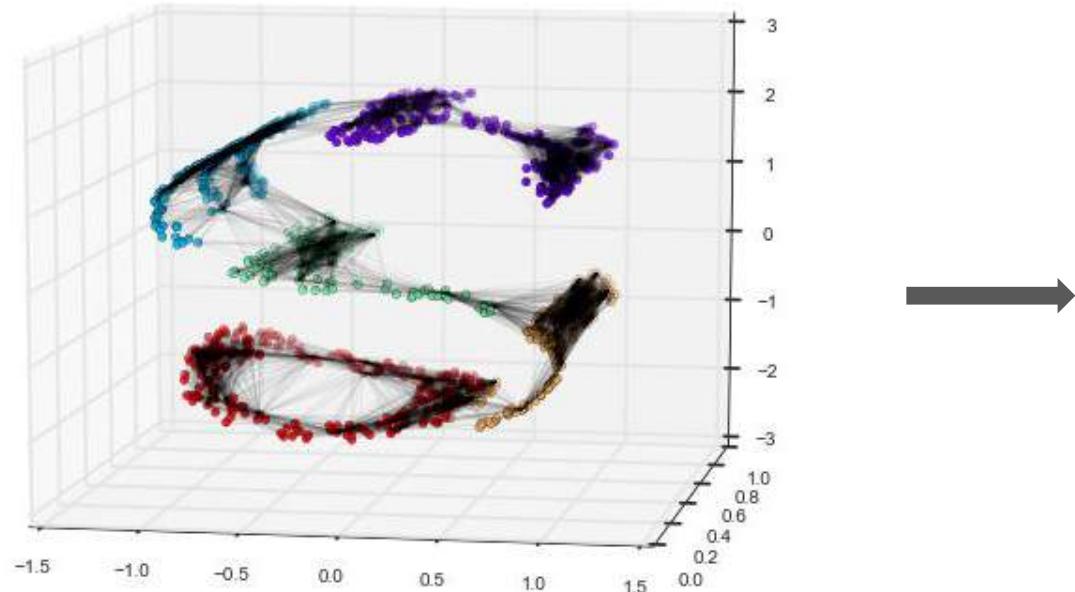


Learning a "curvy" manifold



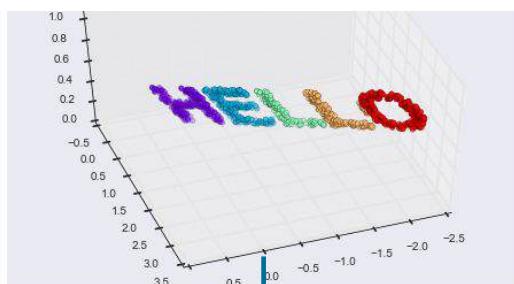
Try to preserve
100 nearest
neighbors...

LLE Linkages (100 NN)

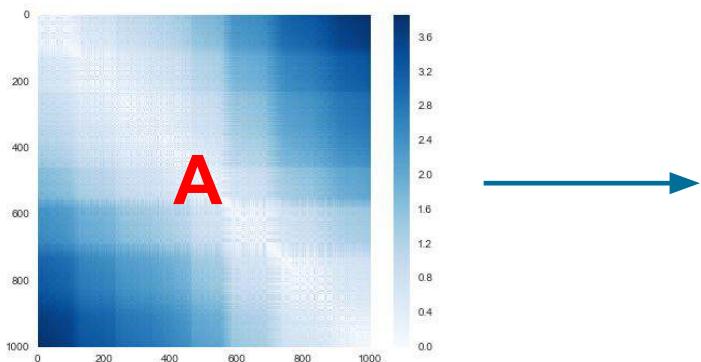


Questions?

Data embedded in
3-D space



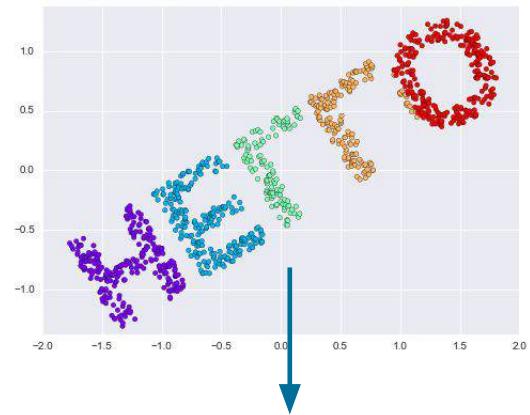
Pairwise distance in 3-D



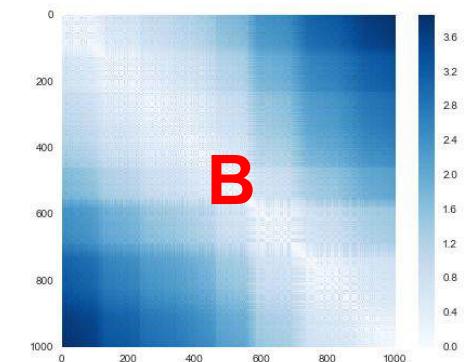
Older manifold learning algorithms:

Complex loss functions,
difficult to optimize...

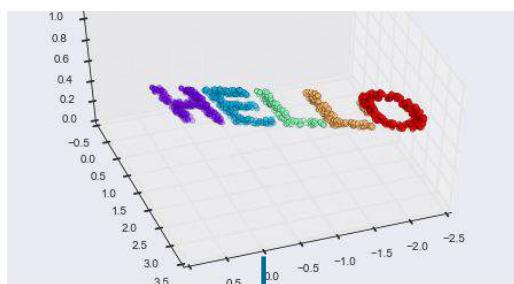
Estimated 2-D embedding



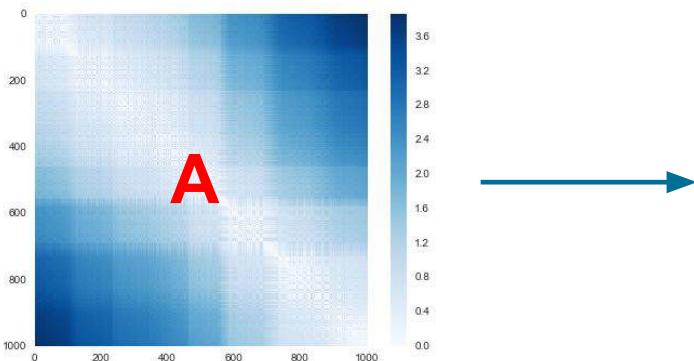
Pairwise distance in 2-D



Data embedded in
3-D space



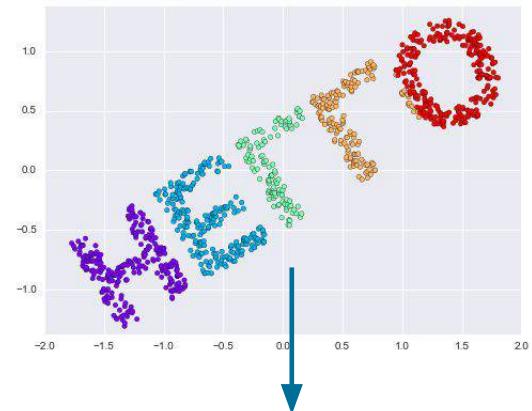
Pairwise distance in 3-D



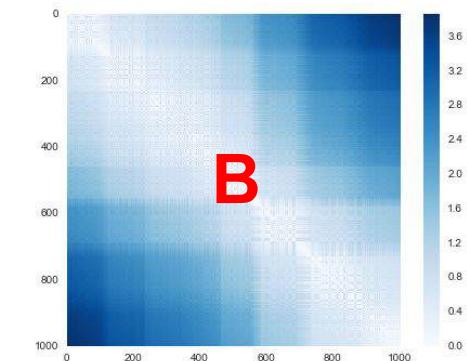
e.g. Stochastic
neighbor
embedding* (SNE):

Non-symmetric loss
function:
 $\|A-B\| \neq \|B-A\|$

Estimated 2-D embedding

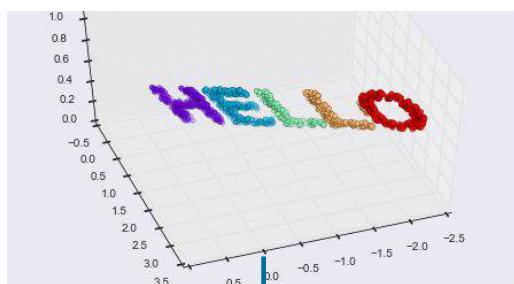


Pairwise distance in 2-D

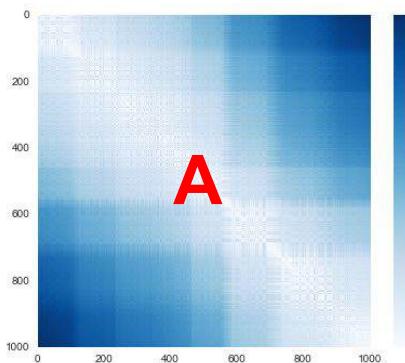


*Geoffrey Hinton, Sam Roweis. *Stochastic neighbor embedding*. Neural Information Processing Systems, 2002.

Data embedded in
3-D space



Pairwise distance in 3-D

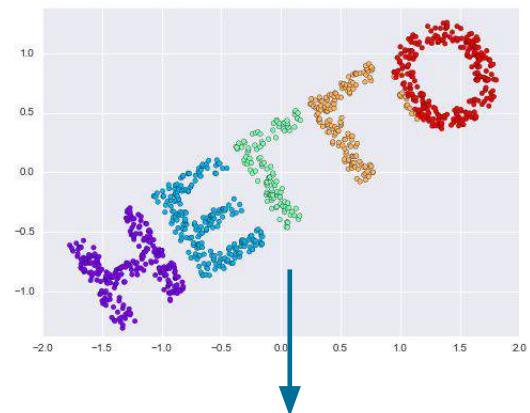


Solution:

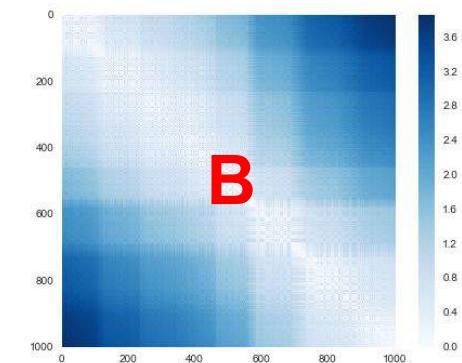
t-distributed
stochastic neighbor
embedding (t-SNE):

Symmetric loss function:
 $\|A-B\| = \|B-A\|$

Estimated 2-D embedding



Pairwise distance in 2-D



t-SNE - What does it do?

Stochastic Neighbor Embedding (SNE) had fundamental **problems**, e.g.:

- Cost function is difficult to optimize
- Crowding problem: 1-D



t-SNE - What does it do?

Stochastic Neighbor Embedding (**SNE**) had fundamental **problems**, e.g.:

- Cost function is difficult to optimize
- Crowding problem: 1-D



t-SNE - What does it do?

Stochastic Neighbor Embedding (**SNE**) had fundamental **problems**, e.g.:

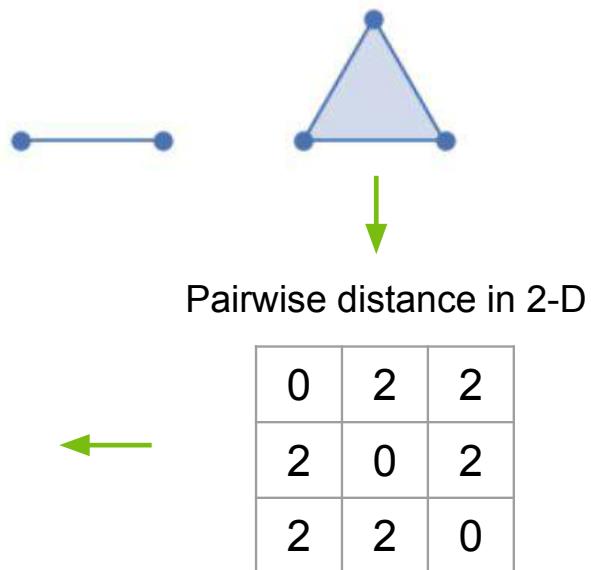
- Cost function is difficult to optimize
- Crowding problem: 1-D 2-D



t-SNE - What does it do?

Stochastic Neighbor Embedding (**SNE**) had fundamental **problems**, e.g.:

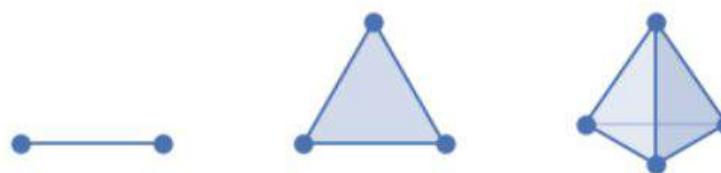
- Cost function is difficult to optimize
- Crowding problem: 1-D 2-D



t-SNE - What does it do?

Stochastic Neighbor Embedding (SNE) had fundamental **problems**, e.g.:

- Cost function is difficult to optimize
- Crowding problem*: 1-D 2-D 3-D

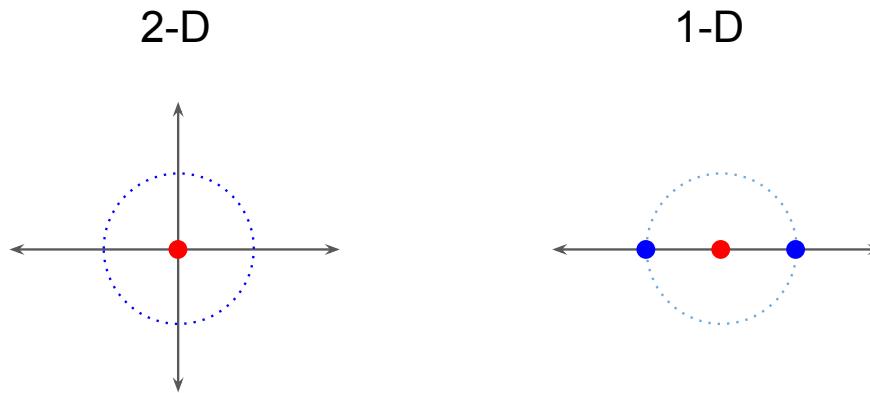


*When attempting to preserve medium-range distances between data points, we also attempt to preserve the volume spanned by them. There isn't enough area/volume available in lower-dimensional spaces to reliably represent very large volumes in high-dimensional space. So when we "pull together" more distant points, we are overcrowding densely populated areas in low-dimensional space and cannot separate distinct clusters very well.

t-SNE - What does it do?

Stochastic Neighbor Embedding (**SNE**) had fundamental **problems**, e.g.:

- Cost function is difficult to optimize
- Crowding problem:



t-SNE - What does it do?

Stochastic Neighbor Embedding (**SNE**) had fundamental **problems**, e.g.:

- Cost function is difficult to optimize
 - Crowding problem
-
- **t-SNE** proposes to **solve these problems!**^[2]
 - Cost function easier to optimize
 - Aims to *both* "pull together" neighboring points as much as possible, and "push apart" distant points as much as possible: better visualization of distinct clusters.

t-SNE - What does it do?

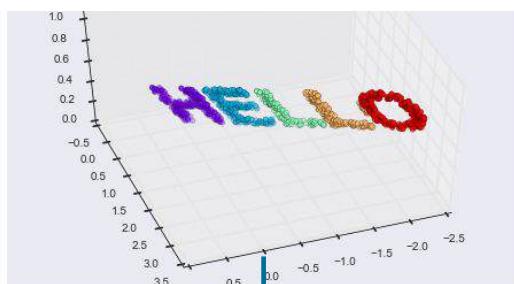


- t-SNE proposes to **solve these problems!**^[2]

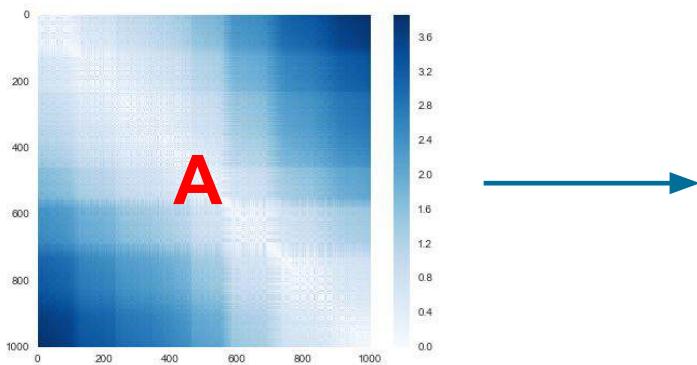
- Cost function easier to optimize
- Aims to *both* "pull together" neighboring points as much as possible, and "push apart" distant points as much as possible: better visualization of distinct clusters.



Data embedded in
3-D space

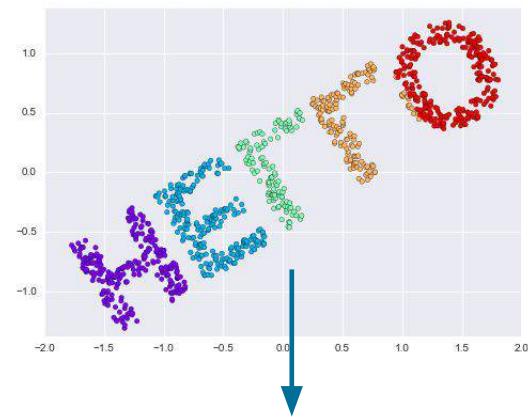


Pairwise distance in 3-D

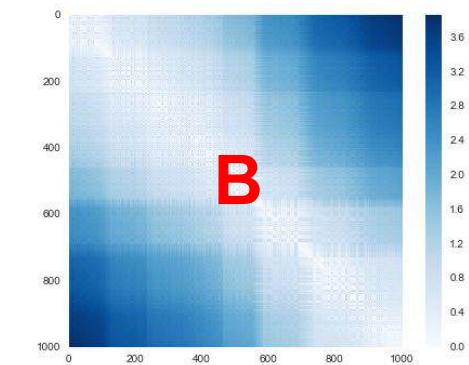


t-distributed
stochastic neighbor
embedding (t-SNE):

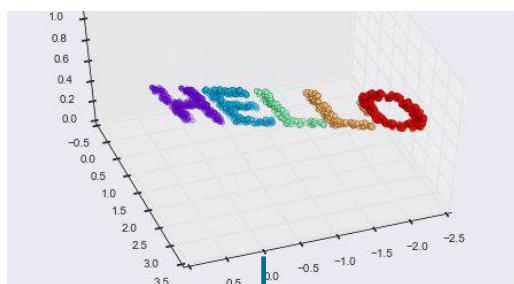
Estimated 2-D embedding



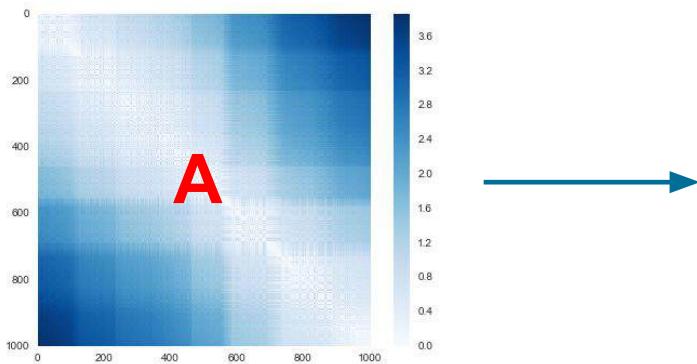
Pairwise distance in 2-D



Data embedded in
3-D space

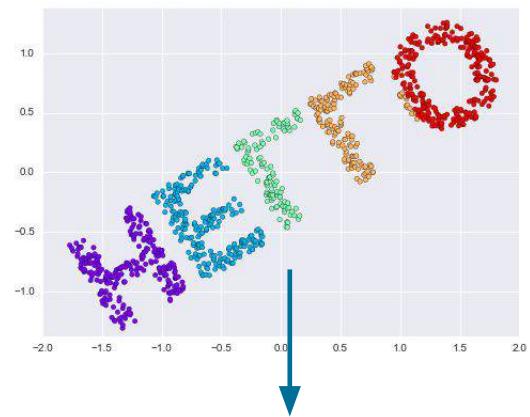


Pairwise distance in 3-D



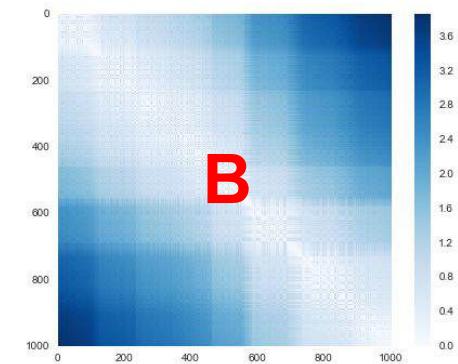
t-distributed
stochastic neighbor
embedding (t-SNE):

Estimated 2-D embedding



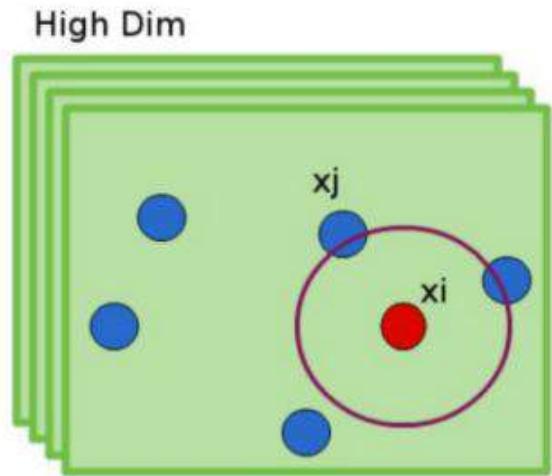
Define “similarity” score
between data points
(inversely proportional to
distance...)

Pairwise distance in 2-D



t-SNE - Algorithm

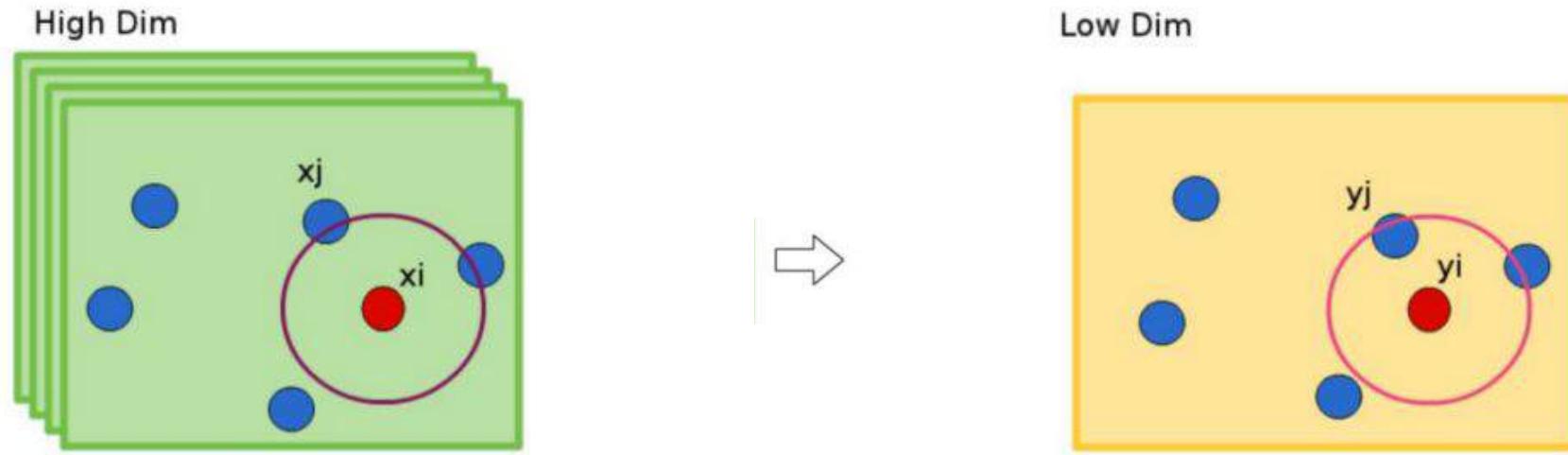
Gaussian distributed **similarities**



$$p_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma^2)},$$

t-SNE - Algorithm

Gaussian distributed **similarities**

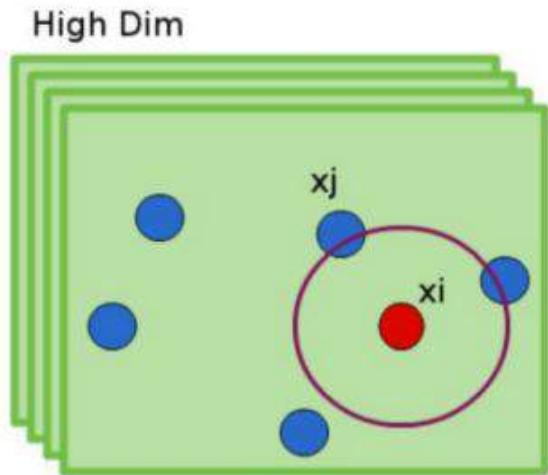


$$p_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma^2)},$$

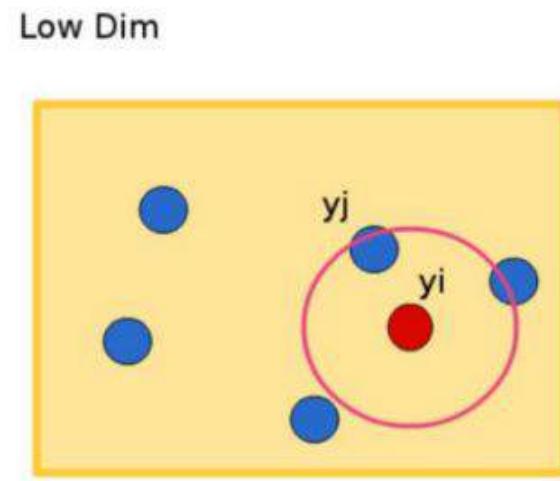
Should it also be Gaussian?

t-SNE - Algorithm

Gaussian distributed **similarities**



t-distributed **similarities**



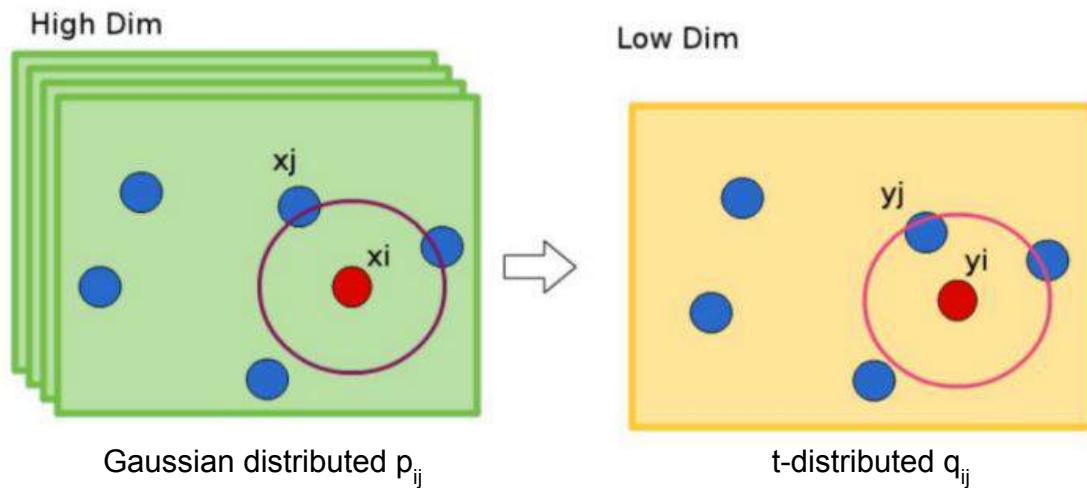
$$p_{ij} = \frac{\exp(-\|x_i - x_j\|^2 / 2\sigma^2)}{\sum_{k \neq l} \exp(-\|x_k - x_l\|^2 / 2\sigma^2)},$$

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l} (1 + \|y_k - y_l\|^2)^{-1}}$$

t-SNE - Cost function

Minimize the Kullback-Leibler divergence between the high dimensional similarity distribution P and the low dimensional similarity distribution Q.

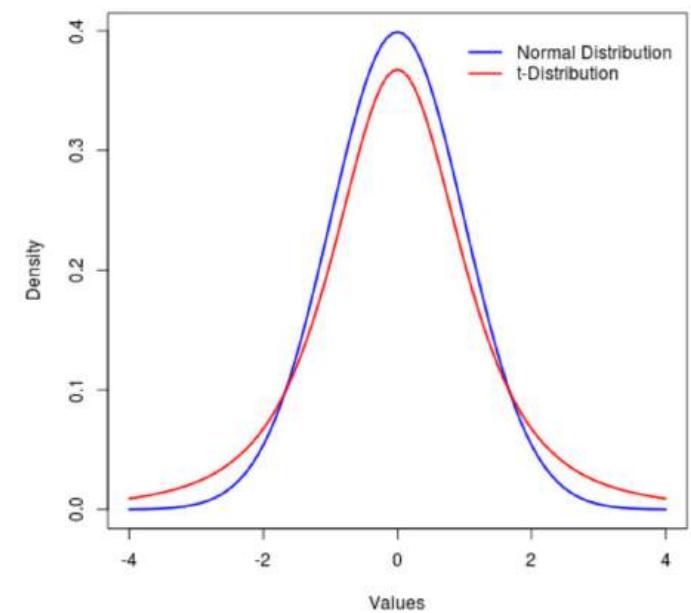
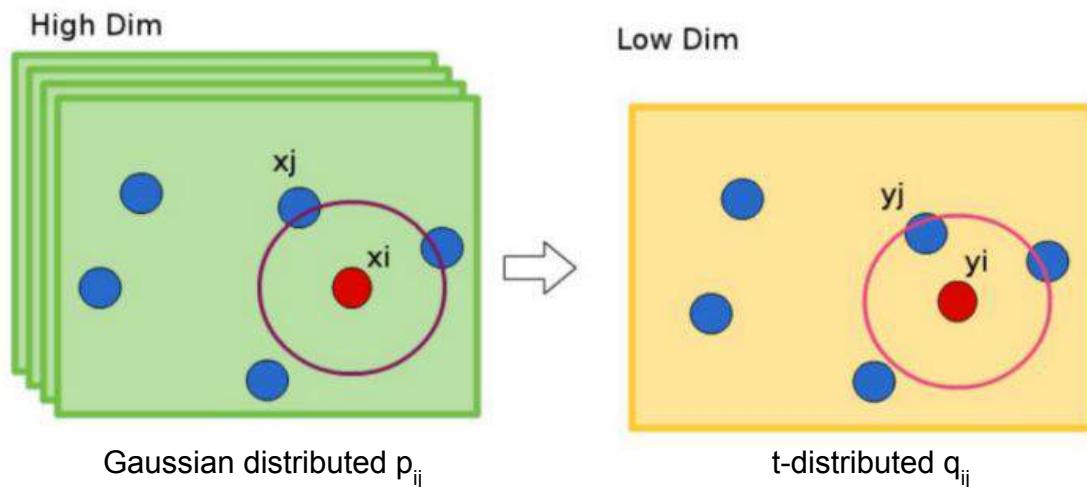
$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$



t-SNE - Cost function

Minimize the Kullback-Leibler divergence between the high dimensional similarity distribution P and the low dimensional similarity distribution Q.

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$



t-SNE - MNIST visualizations

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

28 x 28 =
784 dimensions

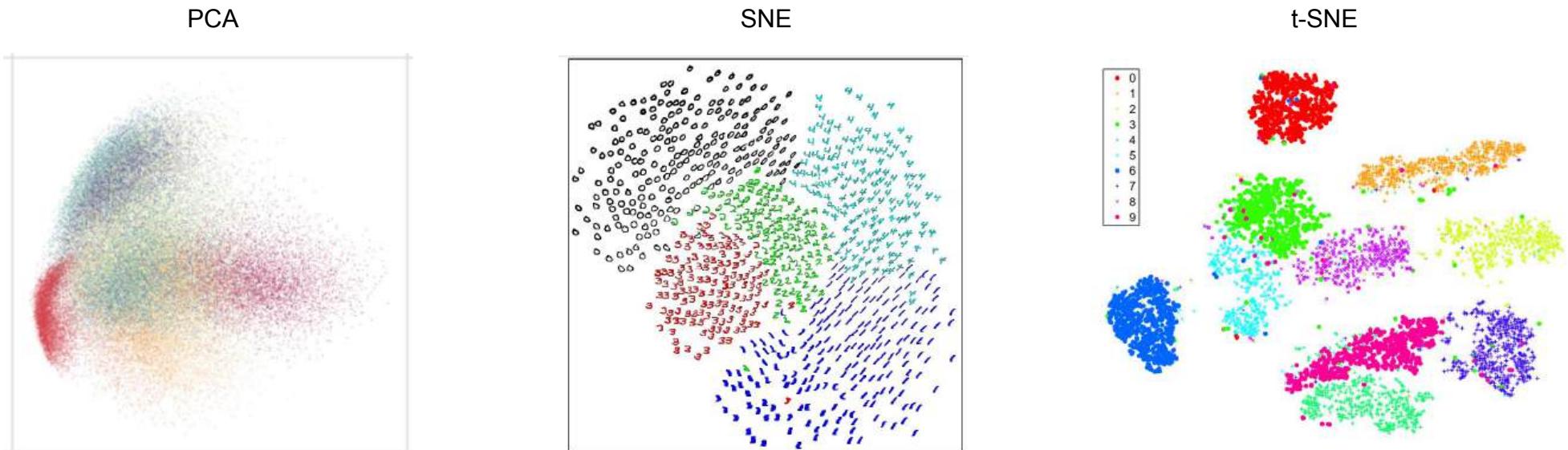
t-SNE - MNIST visualizations

PCA cannot capture the **nonlinear structure** of MNIST!

SNE suffers from the **overcrowding problem**.

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

$28 \times 28 =$
784 dimensions



t-SNE

Advantages

- Can capture **nonlinear** relationships in data
- Preserves local and global structures well for visualization (i.e. **well separated clusters** in low-dimensional maps)
- **Effective in practice** (solves optimization and crowding problems of SNE)

t-SNE

Advantages

- Can capture **nonlinear** relationships in data
- Preserves local and global structures well for visualization (i.e. **well separated clusters** in low-dimensional maps)
- **Effective in practice** (solves optimization and crowding problems of SNE)

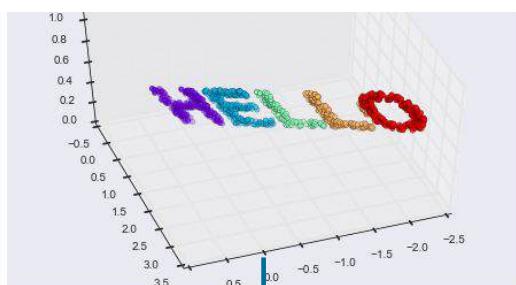
Disadvantages

- Recommended to only **use for visualization** (downstream tasks may be misled by the change in global neighborhoods or **non-preservation of distances**)
- Model and optimization **parameters** (e.g. σ and learning rate) need to be set by hand (cost function not convex, **sensitive** to initial conditions)
- Computational complexity high: $O(N^2)$, so relatively **slow** for large datasets

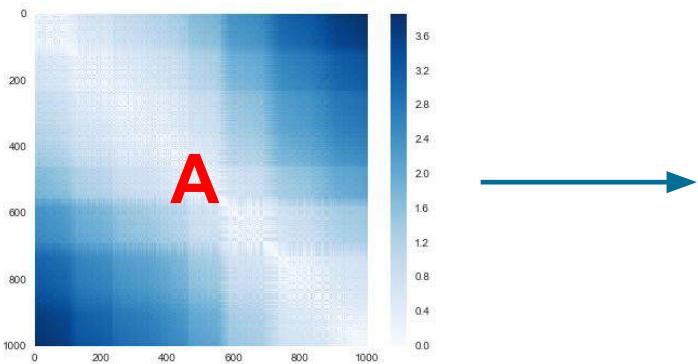
	t-SNE	PCA
Type	Non-linear	Linear
Goal	Preserve local pairwise distances	Preserve global variance
Best used for	Complex, high-dimensional data	Data with linear structure
Output	Low-dimensional embeddings	Principle components
Use cases	Cluster visualization	Anomaly detection, feature extraction
Computational intensity	High (In many manifold learning methods, PCA is used to first reduce dimensionality (e.g. to 30-50 dimensions))	Low
Interpretation	Harder to interpret	Easier to interpret

Questions?

Data embedded in
3-D space



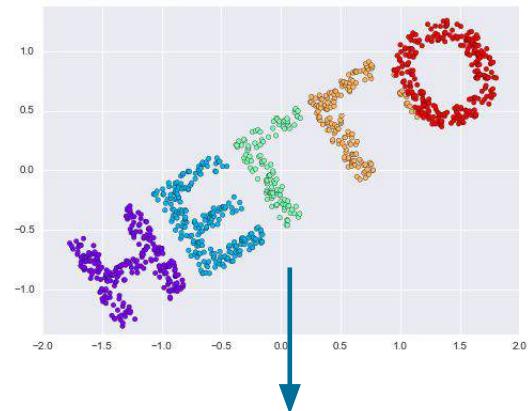
Pairwise distance in 3-D



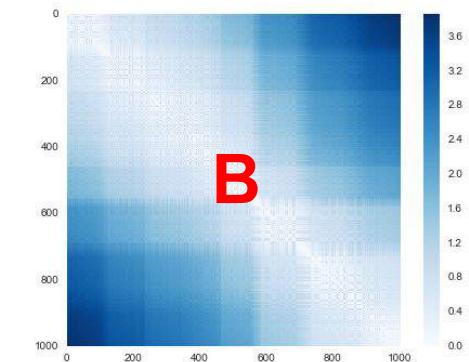
remember:

Hard or impossible to get
 $\|A-B\|=0$

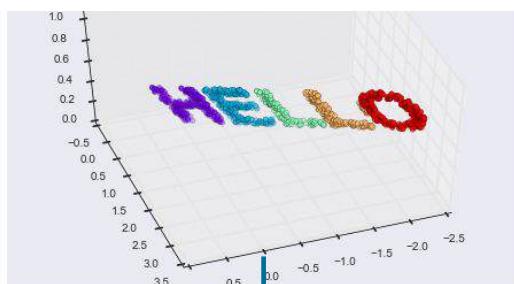
Estimated 2-D embedding



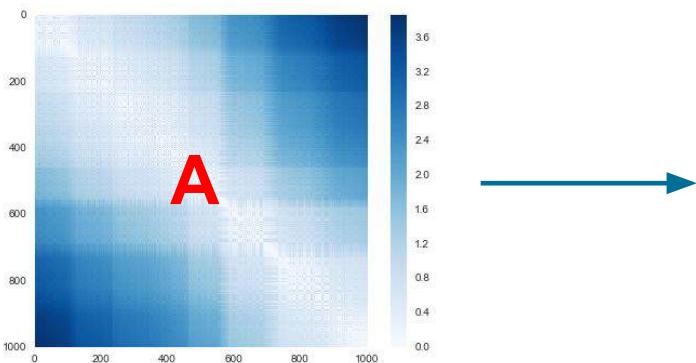
Pairwise distance in 2-D



Data embedded in
3-D space



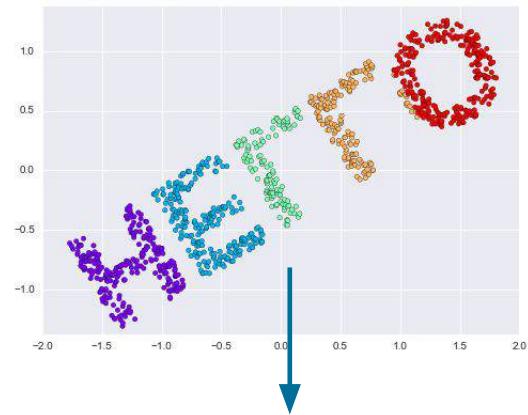
Pairwise distance in 3-D



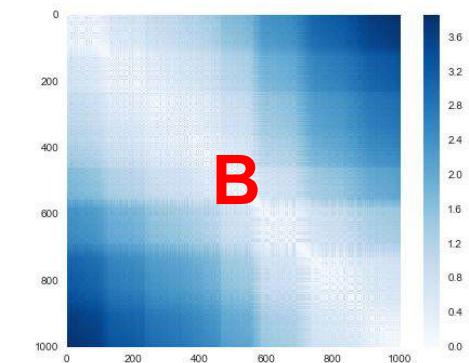
A different
approach:

Define data points as
nodes in a graph!
(then minimize the
difference between the
graph embeddings...)

Estimated 2-D embedding



Pairwise distance in 2-D



UMAP - What does it do?

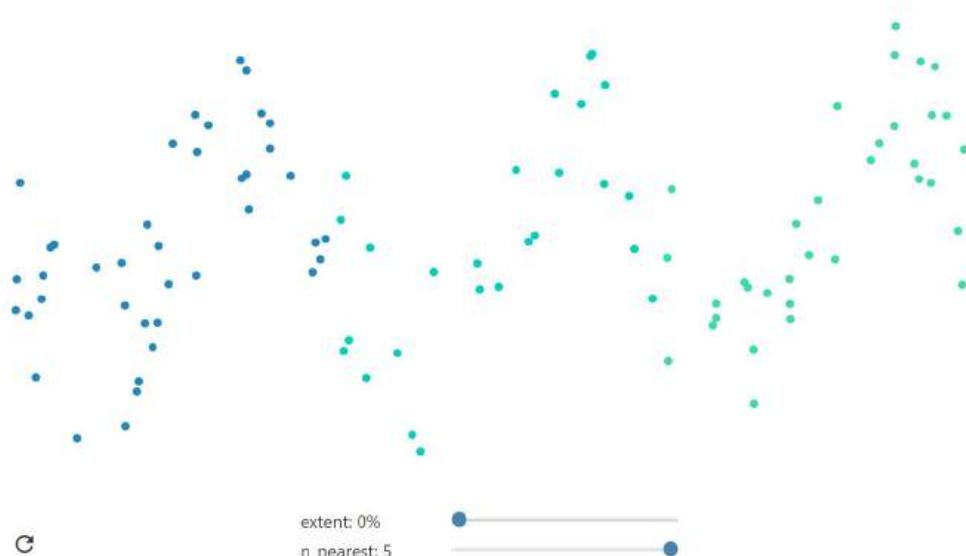
Uniform Manifold Approximation and Projection (UMAP)

UMAP is a "k-neighbour based **graph learning** algorithm"*.

*See e.g. https://umap-learn.readthedocs.io/en/latest/how_umap_works.html for more details.

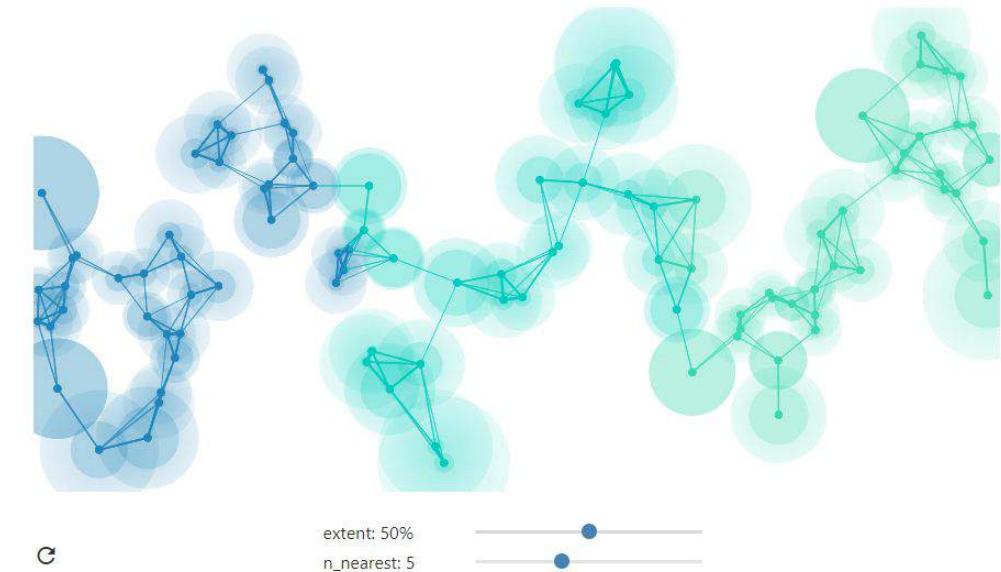
UMAP - What does it do?

Uniform Manifold Approximation and Projection (UMAP)

UMAP is a "k-neighbour based **graph learning** algorithm"*. 

Main idea:

- 1) Build a graph based representation of the data in high dimensional space based on **k** nearest neighbors.



UMAP - What does it do?

Uniform Manifold Approximation and Projection (UMAP)

UMAP is a "k-neighbour based **graph learning** algorithm"*.

Main idea:

- 1) Build a graph based representation of the data in high dimensional space based on **k** nearest neighbors.

Example:

<https://pair-code.github.io/understanding-umap/#:~:text=refresh-,Figure%203%3A,-Adjust%20the%20slider>

C

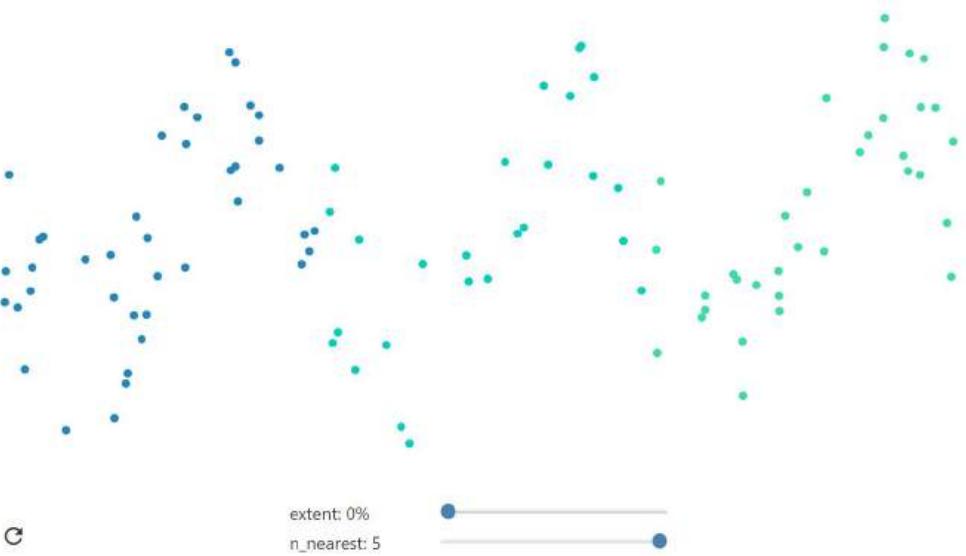
extent: 0%
n_nearest: 5

C

extent: 50%
n_nearest: 5

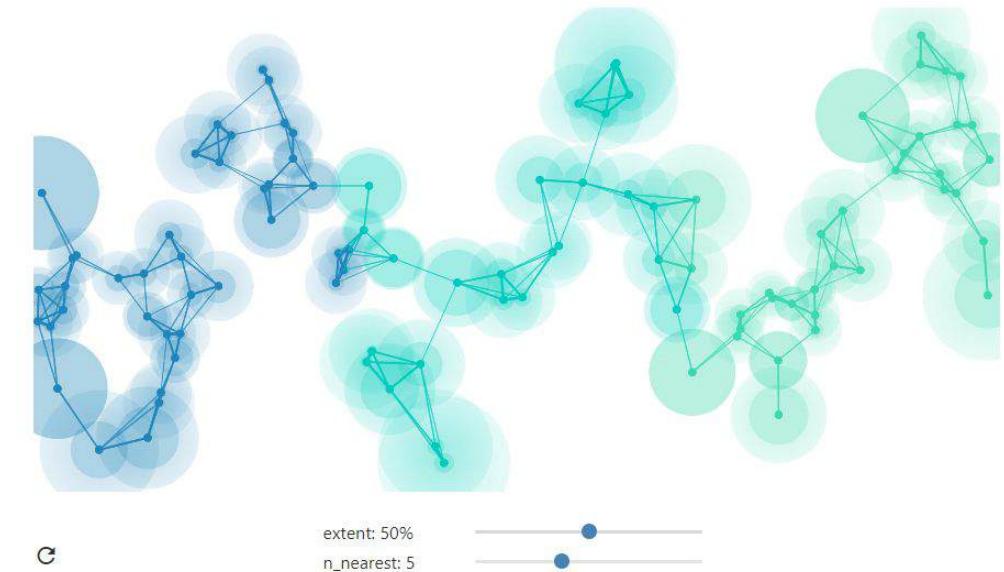
UMAP - What does it do?

Uniform Manifold Approximation and Projection (UMAP)

UMAP is a "k-neighbour based **graph learning** algorithm"*. 

Main idea:

- 1) Build a graph based representation of the data in high dimensional space based on **k** nearest neighbors.
- 2) Find a **lower dimensional** embedding with a similar graph.



UMAP - How does it work?

Assumptions^[3]

- 1) There exists a **manifold** on which the data is uniformly distributed (provides theoretical benefits).
- 2) The manifold of interest is **locally connected** (each data point has **at least one** nearest neighbor).

Goal: We try to preserve the **topological structure** of this manifold [in lower dimensions].

UMAP - How does it work?

Assumptions^[3]

- 1) There exists a **manifold** on which the data is uniformly distributed (provides theoretical benefits).
- 2) The manifold of interest is **locally connected** (each data point has **at least one** nearest neighbor).

Goal: We try to preserve the **topological structure** of this manifold [in lower dimensions].

Control over preserved information

The choice of parameters (extent + number of nearest neighbors **k**) defines the extent to which each data point will be connected to other points in the graph. For example: Larger k → preserves more global structure, smaller k → preserves more local structure^[5].

UMAP - MNIST visualizations

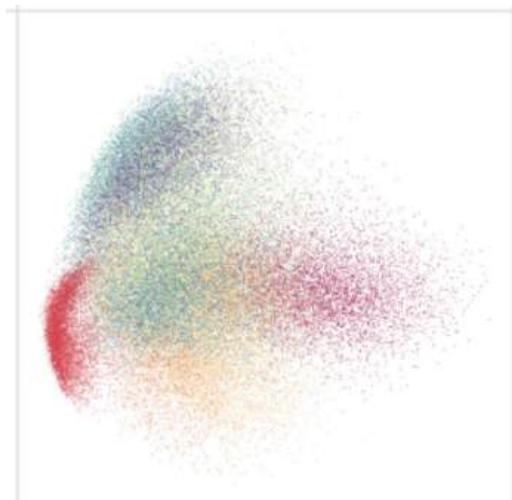
Advantage

Provides well separated clusters

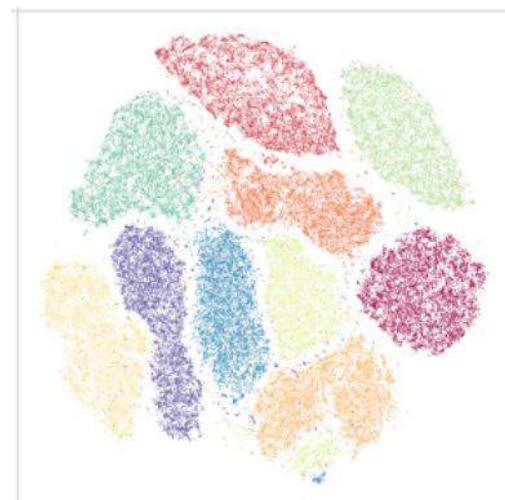
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

$28 \times 28 =$
784 dimensions

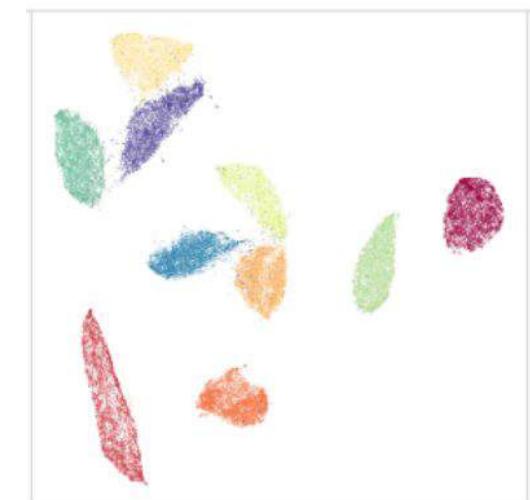
PCA



t-SNE



UMAP



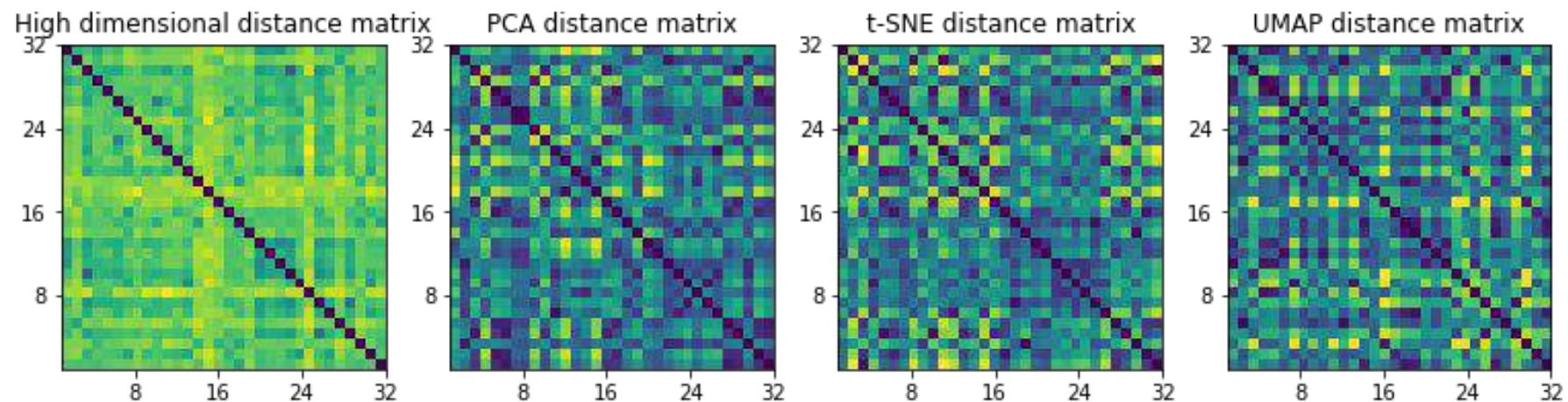
UMAP - MNIST visualizations

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

28 x 28 =
784 dimensions

Disadvantage

Distance information still lost

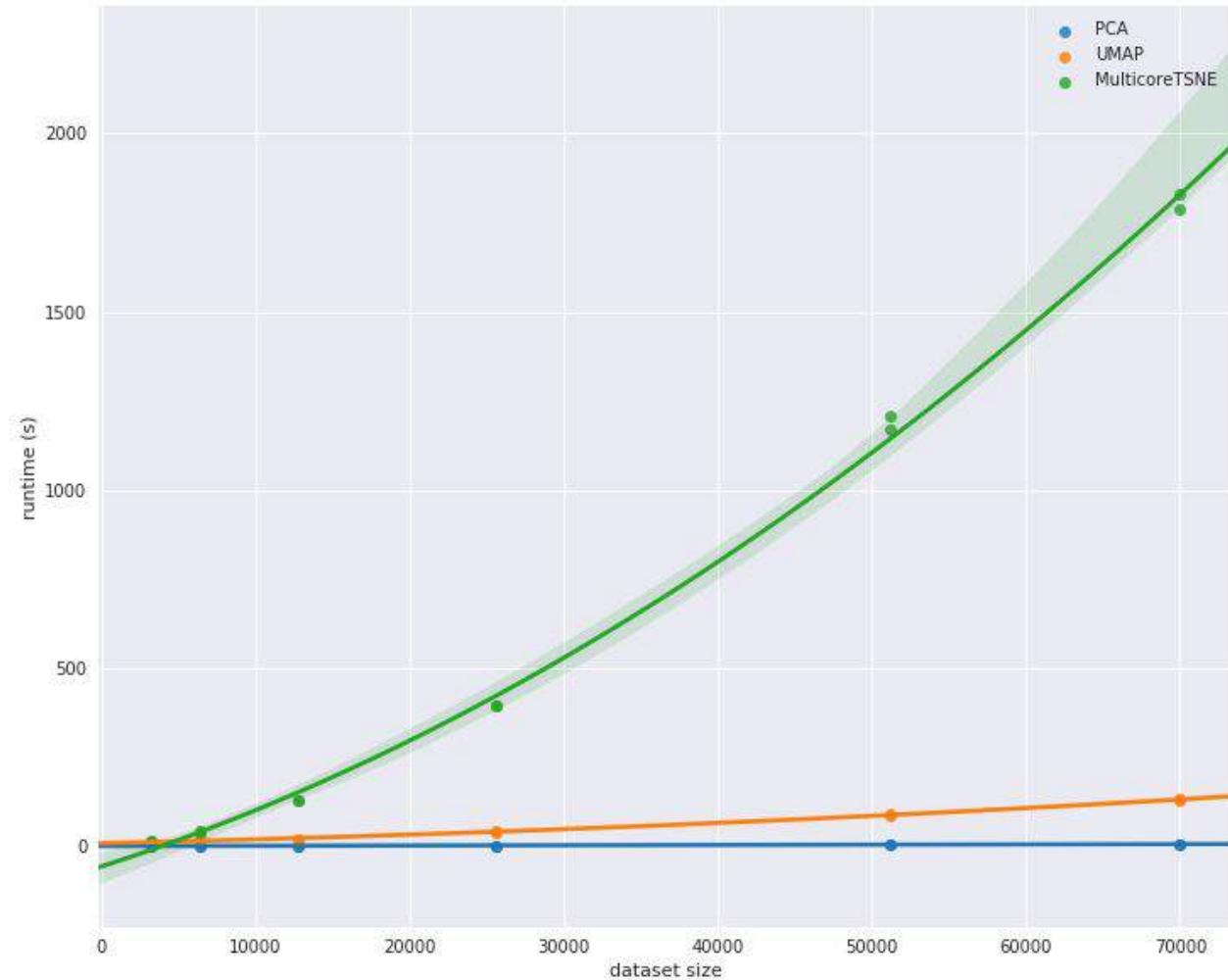


UMAP

Advantage

Much faster than t-SNE

Complexity
 $O(N \log N)$



UMAP - How does it work?

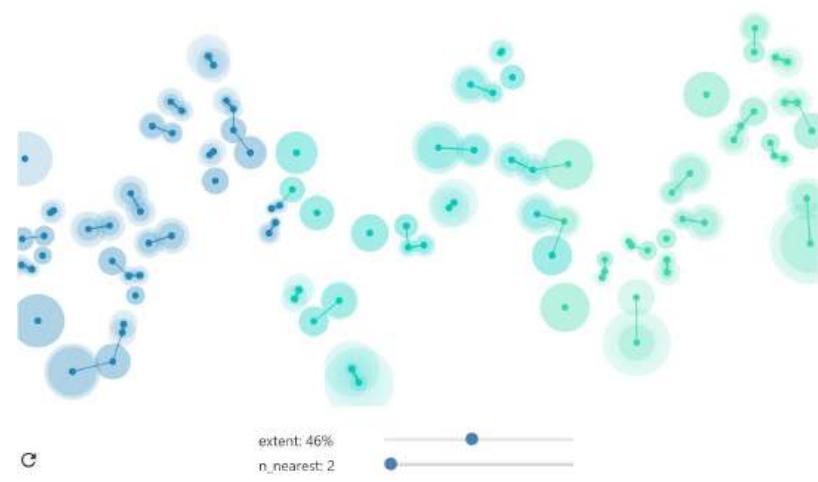
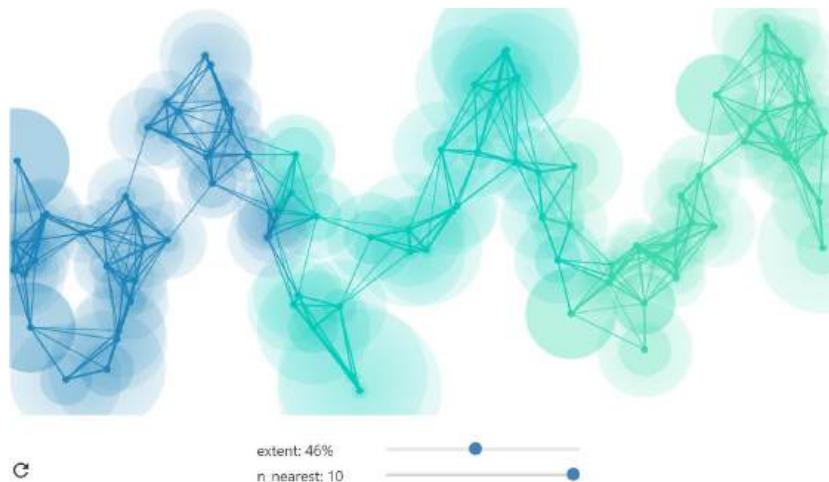
Advantage: Control over what information is preserved in low-D.

UMAP - How does it work?

Advantage: Control over what information is preserved in low-D.

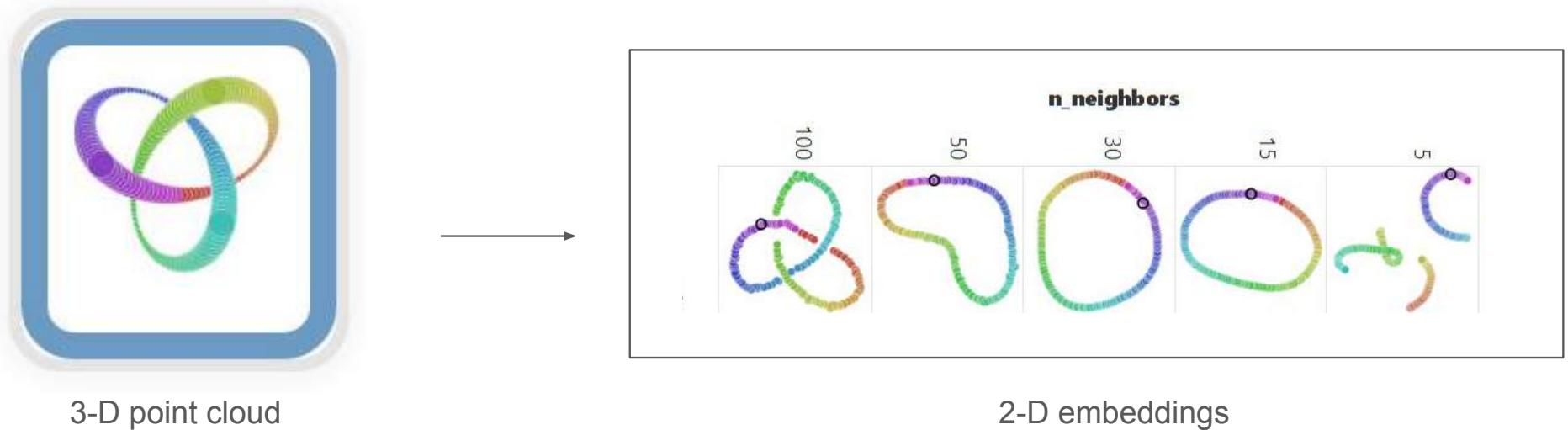
The choice of nearest neighbors **k** defines the connectivity of the graph.

Larger k → preserves more **global structure**, **smaller k** → preserves more **local structure**^[5].



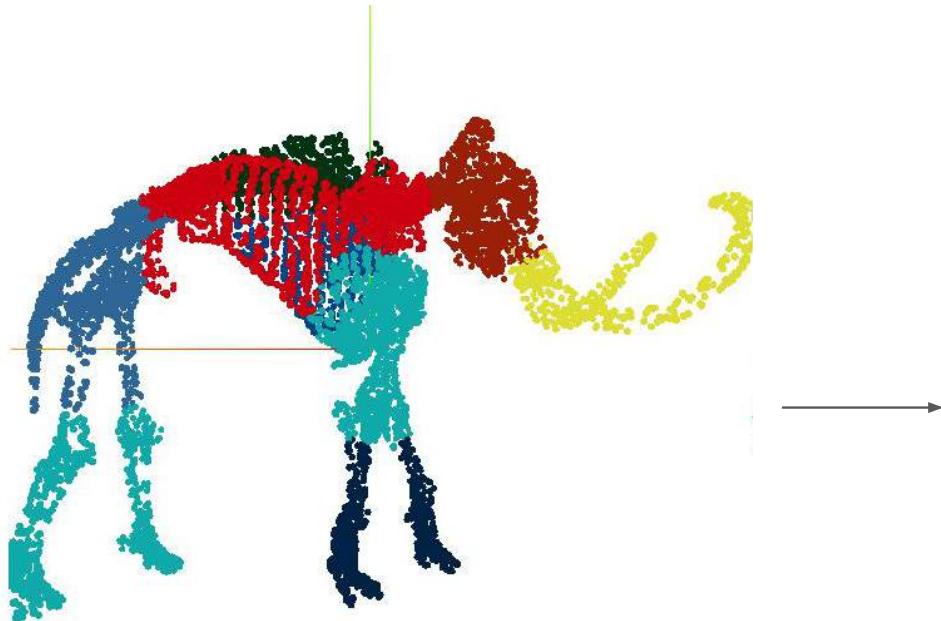
UMAP

Advantage Control over local vs. global information preservation.

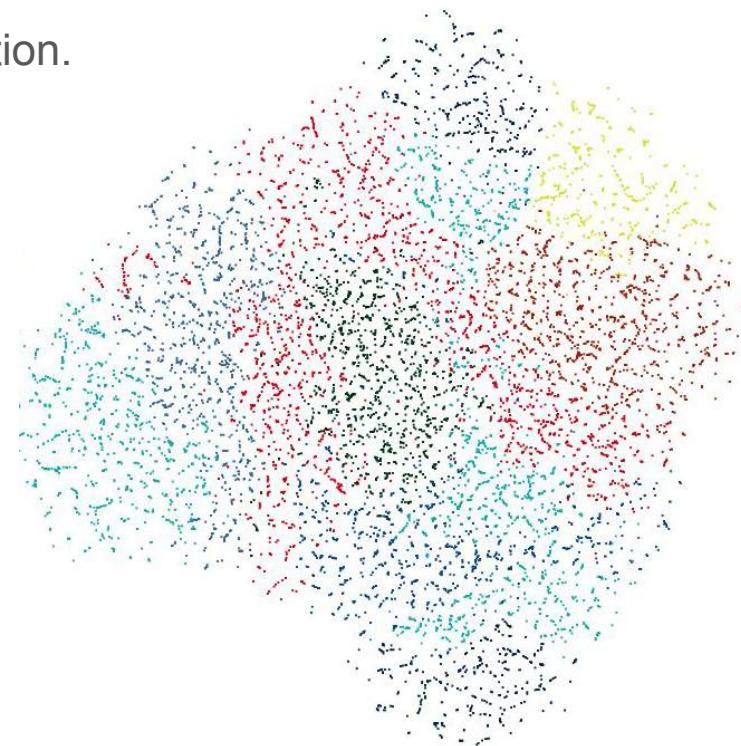


UMAP

Advantage Control over local vs. global information preservation.



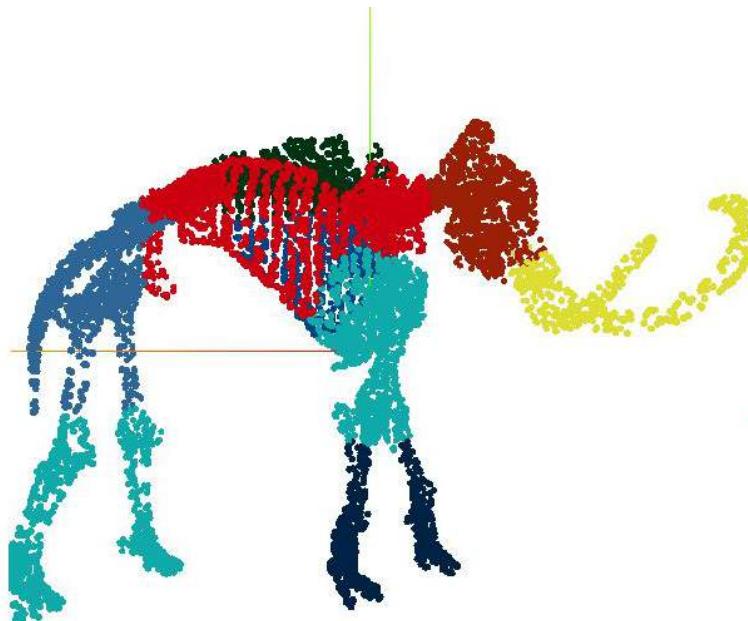
3-D woolly mammoth skeleton



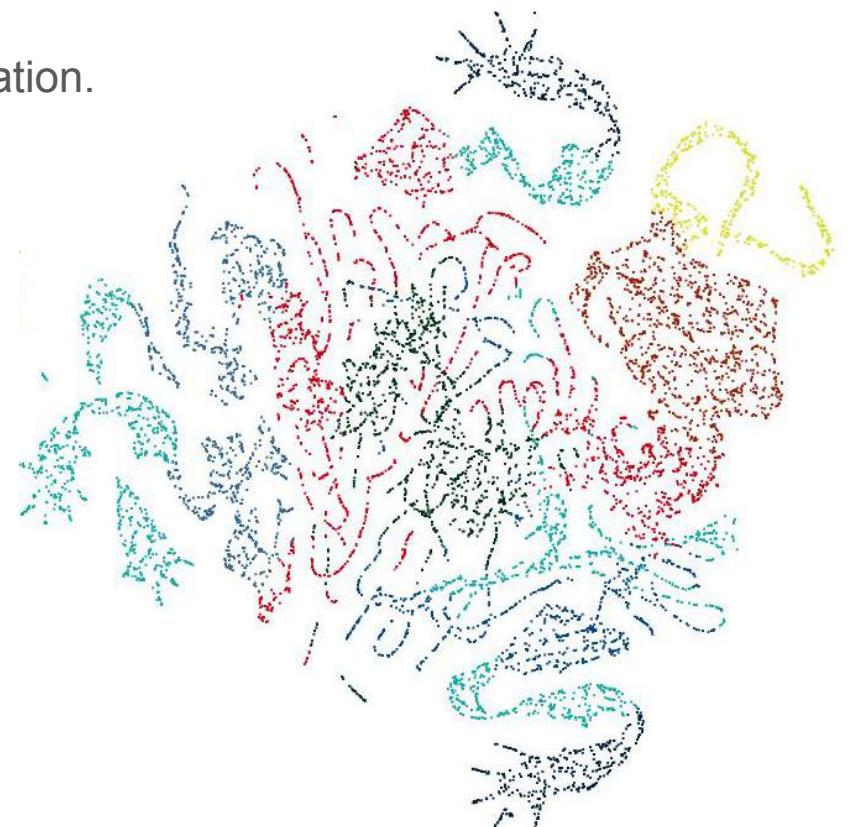
2-D embeddings ($k=3$)

UMAP

Advantage Control over local vs. global information preservation.



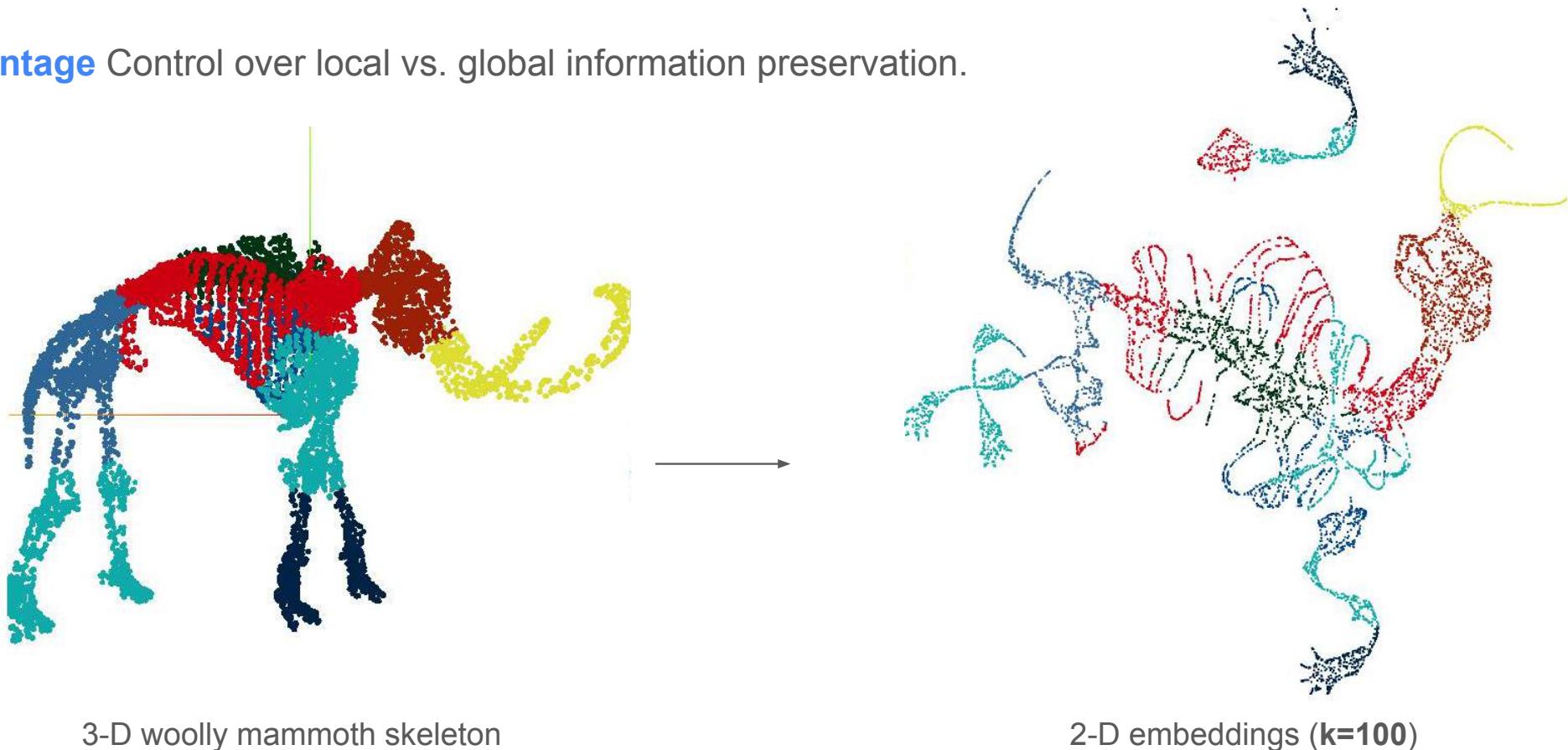
3-D woolly mammoth skeleton



2-D embeddings ($k=10$)

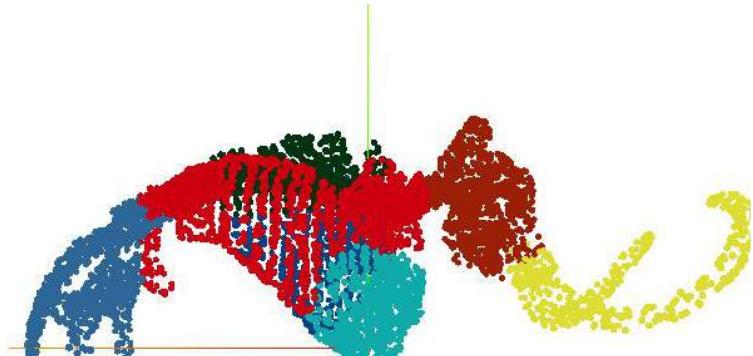
UMAP

Advantage Control over local vs. global information preservation.



UMAP

Advantage Control over local vs. global information preservation.



Example:

https://pair-code.github.io/understanding-umap/#:~:text=min_dist%3A%200.1-,Figure%205%3A,-UMAP%20projections%20of



3-D woolly mammoth skeleton

2-D embeddings (**k=100**)

UMAP

Advantages

- Control over **local** vs. **global** information preservation
- **Fast**
- Works well on complex, real-world data (well **separated clusters**)

UMAP

Advantages

- Control over **local** vs. **global** information preservation
- **Fast**
- Works well on complex, real-world data (well **separated clusters**)

Disadvantages

- **Free parameters** to be optimized (e.g. extent and number of nearest neighbors **k**)
- Global clusters preservation better than t-SNE, but distance information still lost (only **visualization**)
- Non-convex optimization (like t-SNE), sensitive to **optimization parameters** and initial conditions

Colab notebook

Feel free to have a look at the notebook:

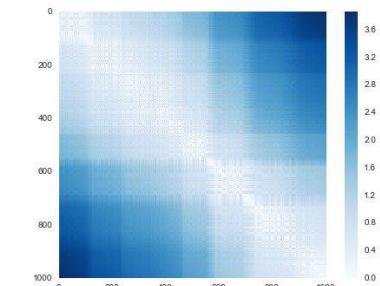
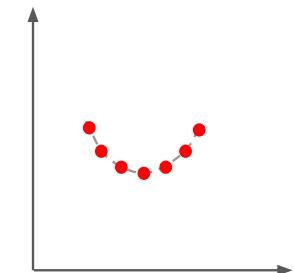
https://colab.research.google.com/drive/127VMpxVT-JFMC_JmNco0X1PWEWUUfwYY?usp=sharing

It contains example code using PCA, t-SNE and UMAP to visualize the MNIST dataset.

Summary and outlook

- › Dimensionality reduction/visualization of **high-dimensional data** is important!
- › PCA can only capture **linear relationships**.

- › **Manifold learning** can capture non-linear relationships.
- › Using the manifold hypothesis and the **pairwise distance matrix**, we can find lower dimensional data embeddings.

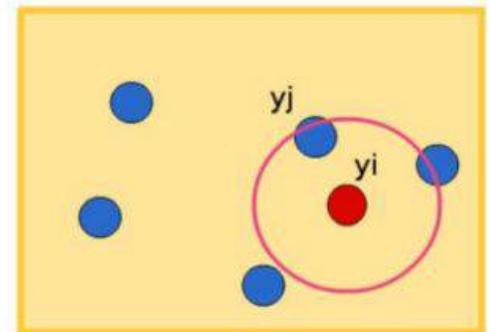


Summary and outlook

› **t-SNE** solves problems of earlier algorithms:

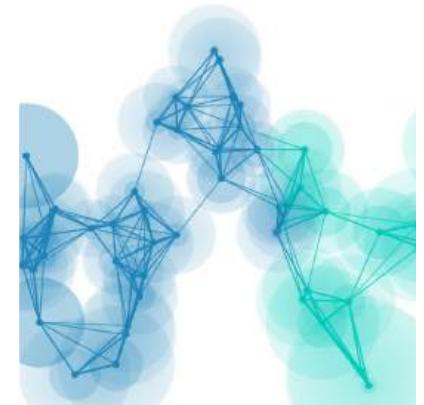
- Cost function **easier** to optimize
- Aims to solve the '**crowding problem**' using t-distributed similarities
- **Slow** and **sensitive** to hyperparameters

Low Dim



› **UMAP**:

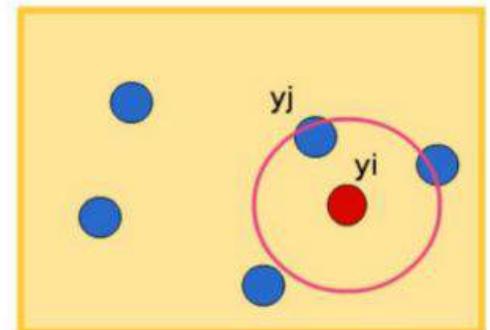
- **Fast**, provides control over **local** vs. **global** information preservation
- Need to optimize number of nearest neighbors **k**



Summary and outlook

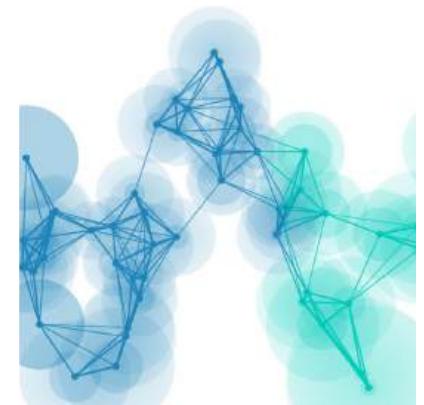
- › **t-SNE** solves problems of earlier algorithms:
 - Cost function **easier** to optimize
 - Aims to solve the '**crowding problem**' using t-distributed similarities
 - **Slow** and **sensitive** to hyperparameters

Low Dim



UMAP:

- **Fast**, provides control over **local** vs. **global** information preservation
- Need to optimize number of nearest neighbors **k**



For both **manifold learning** methods:

Well-separated clusters but **distances are not preserved!**

Evaluation

Dear Student,

Your opinion counts!

We hope you are willing to letting us know what you think about the teaching methods, online tools, course organization, assessment, etc. of this course. With your feedback we can further improve our education. Therefore, we kindly ask you to take 5-10 minutes time to fill in the questionnaire.

Please follow the link or scan the QR code below to open the questionnaire.

In order to obtain a reliable evaluation result, we hope that many of you will complete this questionnaire. Your answers will be processed anonymously and handled confidentially.

<https://evasys-survey.tudelft.nl/evasys/online.php?p=4MRLK>

Many thanks in advance for your feedback! Later on, a summary of the evaluation results will be published on the Brightspace page of your program (go to Content tab – Course evaluations).

Questions about the survey? Mail to QualityAssurance-EEMCS@tudelft.nl

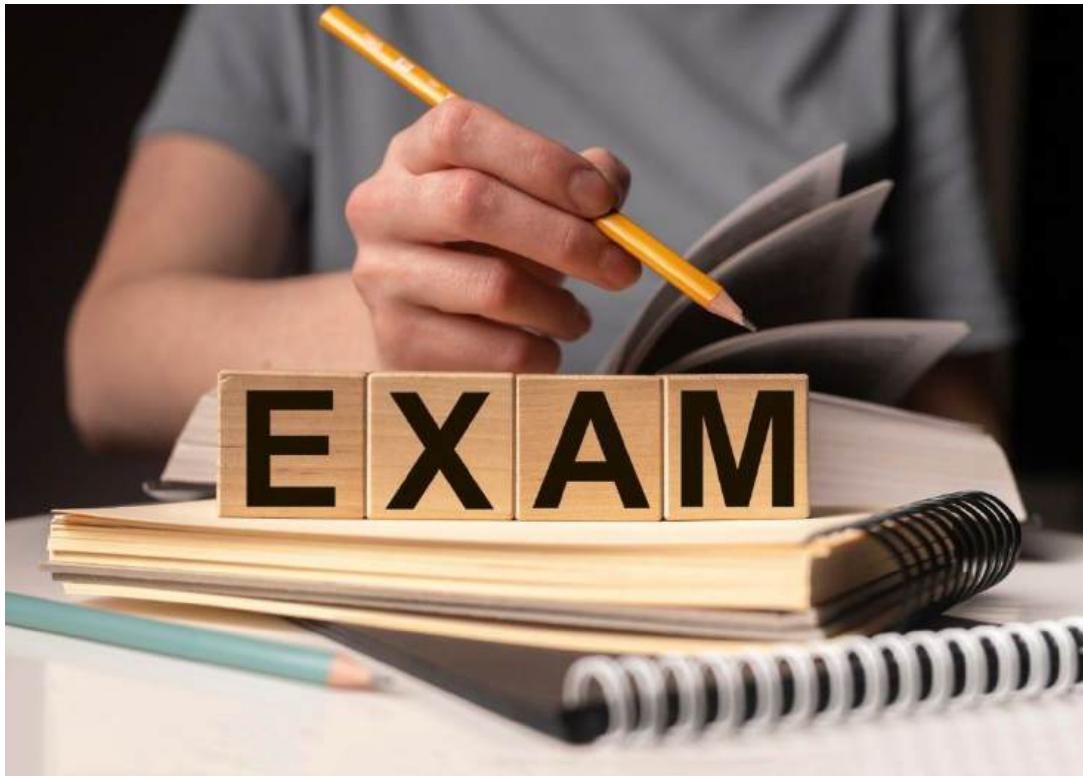


Questions?

References / Recommended reading

- [1] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [2] L. van der Maaten, and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research* 9, no. 11 (2008).
- [3] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint*, arXiv:1802.03426 (2018).
- [4] G. Hinton, and S. Roweis, "Stochastic neighbor embedding," NeurIPS 2002.
- [5] <https://www.youtube.com/watch?v=6BPI81wGGP8>
- [6] Vavasis SA. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*. 2010;20(3):1364-77.
<https://arxiv.org/abs/0708.4149>
- [7] Lee D, Seung HS. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*. 2000;13.

Today	Week 2.9	Jan 20	Exam preparation, Q&A
Thursday	Week 2.9	Jan 23	FREE
Next Monday: Exam	Week 2.10	Jan 27	Weblab exam



Exam recap

CSE2525, Data Mining

20.01.2025

Matrices & PCA

Concepts (understand):

- › Matrices are **linear transformations**. (Is translation a linear transformation?)
- › When multiplied with vectors, matrices can provide **geometric transformations** such as rotation.
- › Matrices can be used as **data tables** to store and manipulate important information.
- › Properties of the covariance matrix.
- › PCA is computed using the eigenvectors of the covariance matrix. First principle component points in the direction of **largest variance** of the dataset.
 - **Maximizing variance** is equivalent to **minimizing reprojection error**.
- › Dimensionality vs. ‘intrinsic’ dimensionality.
- › Reprojection error and ‘explained variance’.

Matrices & PCA

Algorithms (reproduce):

- › PCA.
- › Computing the covariance matrix.

Skills (apply):

- › Assumptions of PCA.
- › PCA can be helpful for dimensionality reduction and anomaly detection.
- › **Normalization can be crucial** for a given dataset and task!
 - e.g. A good method for clustering might be bad for anomaly detection.
- › Dimensionality reduction using PCA might not preserve important structures such as clusters in high dimensions.
- › Difference between using PCA with clean training set vs. one with anomalies.

Non-negative matrix factorization (NMF)

Concepts (understand):

- › What are the **assumptions and constraints** of NMF?
- › Rating/utility matrices don't have the **standard** sample-by-feature "data matrix" structure.
- › NMF can provide **compression**, and can find latent features, useful for interpretability and clustering.
- › NMF can be used for **data imputation** in recommender systems.
- › NMF is a **non-convex optimization** problem. Multiplicative update is a useful algorithm, with several **advantages** over gradient descent for computing NMF.
- › **Linear dependence** and statistical redundancies are useful for compression via NMF.

Non-negative matrix factorization (NMF)

Algorithms (reproduce):

- › NMF with a **dense matrix**: how to preprocess, initialize **W** and **H**, and implement the multiplicative update algorithm.
- › NMF with **missing values** (data imputation): How to modify the NMF routine to be able to handle missing values.

Skills (apply):

- › Decide if you want to use gradient descent or **multiplicative update**.
- › Decide on normalization.
- › Decide on the NMF design choices: Number of features k (for **W**, **H** initialization), error tolerance, maximum iterations.

Recommender systems

Concepts (understand):

- › **Collaborative vs. content-based filtering.**
- › Different types of utility matrices.
- › **Advantages and disadvantages** of collaborative filtering.
- › **Data imputation** problem.
- › **Cold start, scalability, sparsity** problems.
- › Performance metrics.
- › ‘Other considerations’ for building recommender systems

Recommender systems

Algorithms (reproduce):

- › NMF with missing values (**data imputation**).
- › **Thresholding** for obtaining recommendations.
- › Computing **recommendation accuracy**.

Skills (apply):

- › **Cross-validation:**
 - How to pick a cross-validation method (k-fold, leave-one-out, fixed validation set?)
 - How to divide the data for cross-validation (what happens if you remove whole rows/columns?)
- › How to pick a **performance metric** for recommendations (accuracy vs. ranking based)?

Manifold learning

Concepts (understand):

- › **Ranking-based** recommender performance metrics.
- › Diversity and serendipity in recommender systems.
- › **Hybrid** (collaborative+content-based) filtering
- › Privacy concerns for data collection. Specifically: What is **information privacy**? When is cross-referencing a concern?
- › NMF is only **one way** to perform collaborative filtering. There are alternative methods.

Manifold learning

Algorithms (reproduce):

- › Spearman's rank correlation.

Skills (apply):

- › When is PCA not enough?
 - What are PCAs **assumptions**?
 - What are PCAs **advantages and disadvantages**?
 - What **correlations** can PCA not capture?
- › What **correlations** can manifold learning capture (that PCA cannot)?

Dimensionality reduction

Concepts (understand):

- › What is the **manifold hypothesis**?
- › **Advantages/disadvantages** of manifold learning over linear methods (e.g. PCA).
- › Are **distances preserved** in manifold learning? Is it always useful to preserve distances?
- › Manifold learning uses **only** pairwise distances, and are translation, rotation, flip invariant, etc.
- › Assumptions of manifold learning methods.
- › Advantages of t-SNE over previous methods, the **crowding problem**.
- › **t-SNE similarity distributions**: Gaussian in high-D, t-distributed in low-D.
- › UMAP is a **graph learning** algorithm.

Dimensionality reduction

Skills (apply):

- Using UMAP parameters "extent" and “number of nearest neighbors **k**”, we can control what distances are preserved: larger **k** → preserves more global structure, smaller **k** → preserves more local structure.
- t-SNE, **advantages** and disadvantages.
- UMAP, **advantages** and disadvantages.

Anomaly Detection

Concepts (understand):

- › Three types of anomalies and how to detect them, for point x and context C :
 - Point - $F(x) \rightarrow \{\text{normal, anomalous}\}$
 - Contextual - $F(x, C) \rightarrow \{\text{normal, anomalous}\}$
 - Collective - $F(C) \rightarrow \{\text{normal, anomalous}\}$
- › Distance/density based anomalies
- › Spectral/dimensionality reduction techniques (PCA)
- › The pitfalls of temporal correlation when analyzing time series

Anomaly Detection

Algorithms (reproduce):

- › Isolation Forest
- › Dynamic Time Warping

Skills (apply):

- › Sliding windows to detect anomalies in time series
- › Minimizing positive space to detect anomalies using classifiers
- › Use distance or density based anomaly detection
- › Maintaining shape using alignment and normalization in time series

Distances

Concepts (understand):

Metric vs Measure

Properties of a Metric

Definition of metrics

Euclidean, Manhattan, Hamming, Jaccard, Cosine, Graph distances

Skills (apply):

Which metric to use for different datatypes ?

Proof whether a distance is a metric.

Clustering

Concepts (understand):

- › Batching as used in minibatch Kmeans
- › Prototyping as used in CURE
- › Initialization as used in KMeans++
- › Sufficient statistics as used in BFR
- › Evaluating anomaly detection (no need to remember formulas)

Clustering

Algorithms (reproduce):

- › DBScan
- › BFR

Skills (apply):

- › When to use a Euclidean distance in clustering
- › When a metric is required
- › What type of clustering can represent what type of data
- › The effect of normalization on distances
- › Speeding up clustering methods

Discrimination

Concepts (understand):

- › Redlining
- › The prosecutor's fallacy
- › Simpson's paradox
- › The effects of discrimination removal

Algorithms (reproduce):

- › Three ways to lessen discrimination in classification

Skills (apply):

- › Compute discrimination

Locality Sensitive Hashing

Concepts (understand):

- › Shingling (N-grams)
- › Locality sensitive hashing

Algorithms (reproduce):

- › MinHash

Skills (apply):

- › AND and OR constructions
- › Tuning LSH to a desired FP rate
- › How to apply LSH in near neighbor computations

Sketching

Concepts (understand):

- › Box-Cox (power) transforms
- › Morris counting
- › The guarantees of Bloom Filters and Count-Min Sketches (no need to remember formulas)

Algorithms (reproduce):

- › Bloom Filter
- › Count-Min Sketch

Skills (apply):

- › Change parameters to get a desired effect in approximation quality (no need to remember formulas)

Indexing

Concepts (understand):

- › Posting lists, and Inverted indexes
- › Query processing over indexes

Algorithms (reproduce):

- › Doc-at-a-time, and term-at-a-time algos
- › Document inversion

Skills (apply):

- › When to use a query processing algorithm

Embeddings

Concepts (understand):

- Sparse vs dense representations

Algorithms (reproduce):

- Word2vec training
- Random walk based embeddings - Deepwalk vs node2vec

Skills (apply):

- Effect of length of walk, dimensions

Graph Mining

Concepts (understand):

- › Simple Graph properties – radius, distances, eccentricity
- › Centrality properties
- › Graph spectral analysis

Algorithms (reproduce):

- › Connected components
- › Spectral Clustering

Skills (apply):

- › How to store graphs ? How to cluster nodes in the graph ?