Data Preparation

Mining Massive Datasets

Carlos Castillo

Topic 02

Universitat Pompeu Fabra Barcelona

Main Sources

- Data Mining, The Textbook (2015) by Charu Aggarwal (Chapter 2) + slides by Lijun Zhang
- Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 2)
- Data Mining Concepts and Techniques, 3rd edition (2011) by Han et al. (Chapter 3)

“凡事豫（预）则立，不豫（预）则废”——《礼记·中庸》

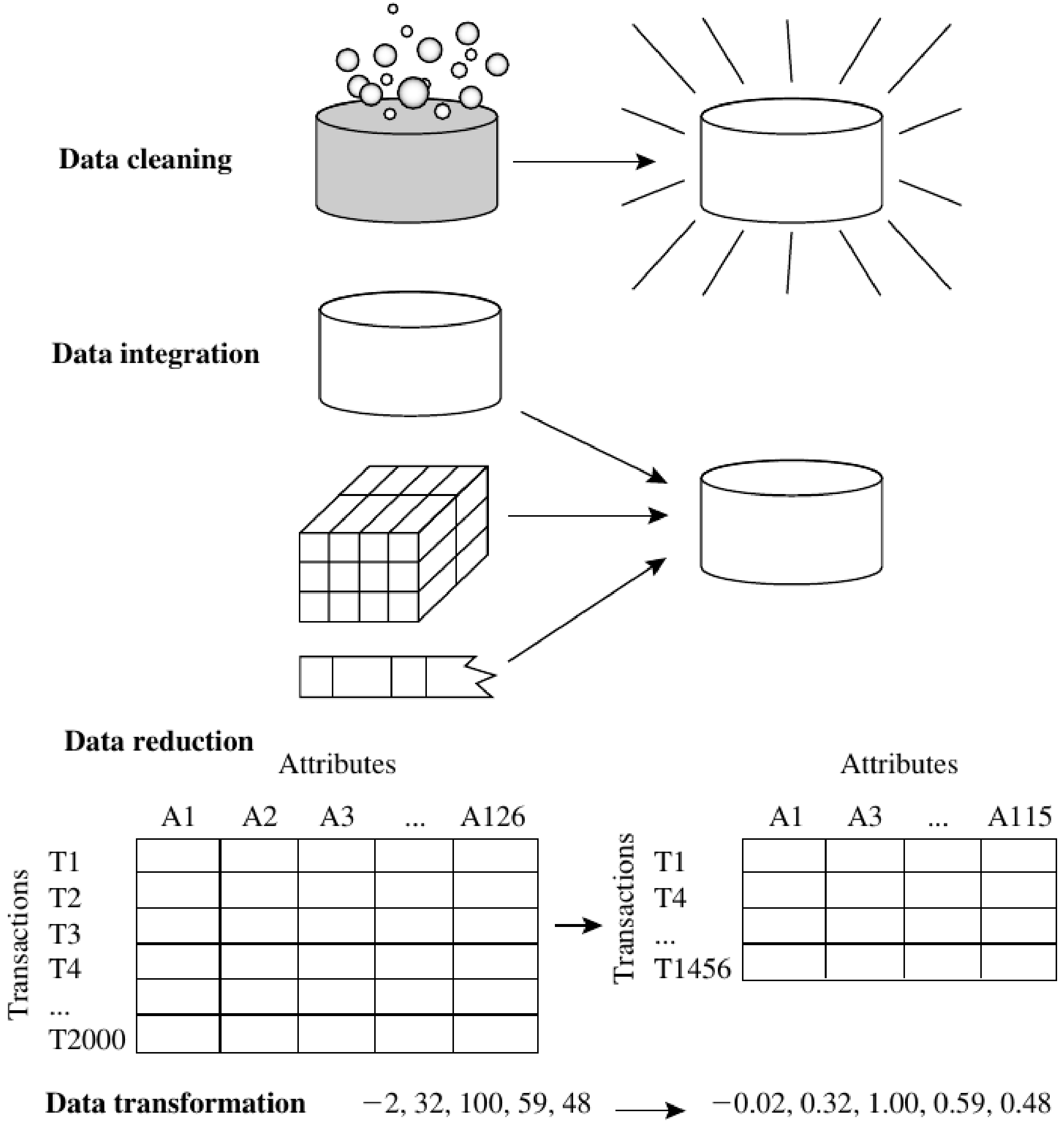
Success depends upon previous preparation, and without such preparation there is sure to be failure – Confucius

Typical datasets

- Records / Matrices
- Documents
- Transactions
- Graphs
- Temporal / Sequences
- Spatial

Data preparation

- Feature Extraction and Portability
  - Extract relevant elements for our analysis
  - Convert heterogeneous data types
- Data Cleaning
  - Deal with missing, erroneous, and inconsistent data
- Data Integration
  - Bring different data sources into a common framework
- Data Reduction, Selection, and Transformation
  - Done for both efficiency and effectiveness



Data Mining Concepts and Techniques, 3rd edition (2011) by Han et al. (page 87)

Feature extraction

Domain	Raw Data	Features
Sensor	Low-level signals	Wavelet or Fourier transforms
Image	Pixels	Color histograms Visual words
Web logs	Text strings	IP address Action
Network traffic	Characteristics of the network packets	Number of bytes transferred Network protocol
Document data	Text strings	Bag-of-words Entity extraction

This is both a skill and an art that the analyst develops over the years.

Data type conversions

- Data is often heterogeneous
  - A demographic data set may contain both numeric and mixed attributes
- Possible solution
  - Designing an algorithm for an arbitrary combination of data types
    - Time-consuming and sometimes impractical
- Converting between various data types
  - Utilize off-the-shelf tools for processing

Data type conversions (cont.)

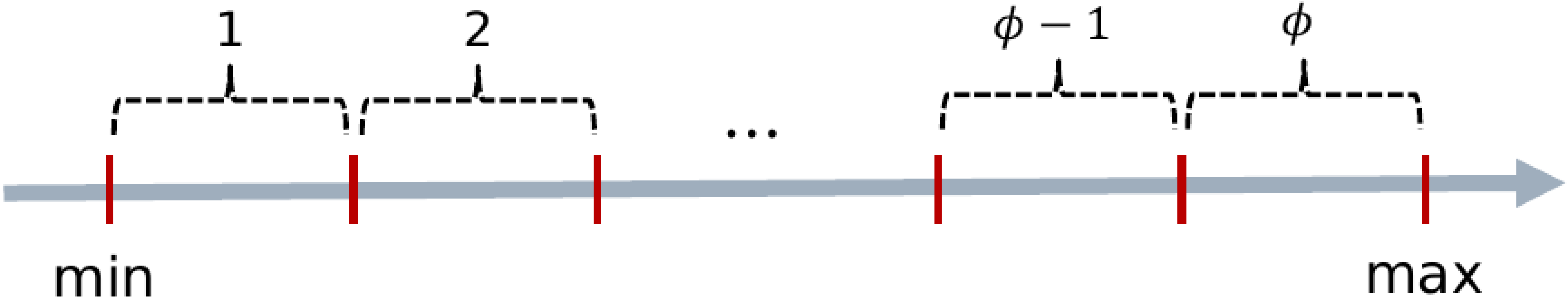
Some ways of converting between data types

Source data type	Destination data type	Methods
Numeric	Categorical	Discretization
Categorical	Numeric	Binarization
Text	Numeric	Latent semantic analysis (*LSA*)
Time series	Discrete sequence	*SAX*
Time series	Numeric multidimensional	*DWT*, *DFT*
Discrete sequence	Numeric multidimensional	*DWT*, *DFT*
Spatial	Numeric multidimensional	2-d *DWT*
Graphs	Numeric multidimensional	*MDS*, spectral
Any type	Graphs	Similarity graph (Restricted applicability)

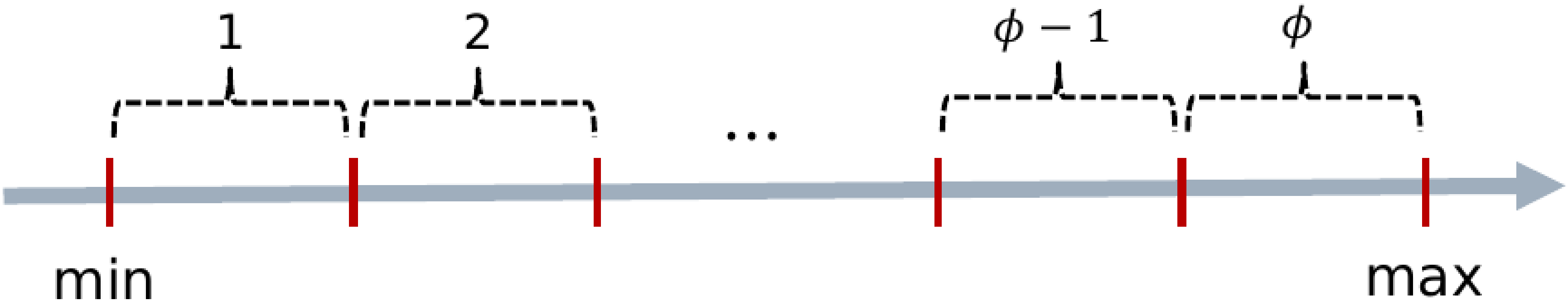
Numerical ⟷ Categorical

Numerical to categorical: discretization

- Divide the range for the numerical variable into Φ different ranges

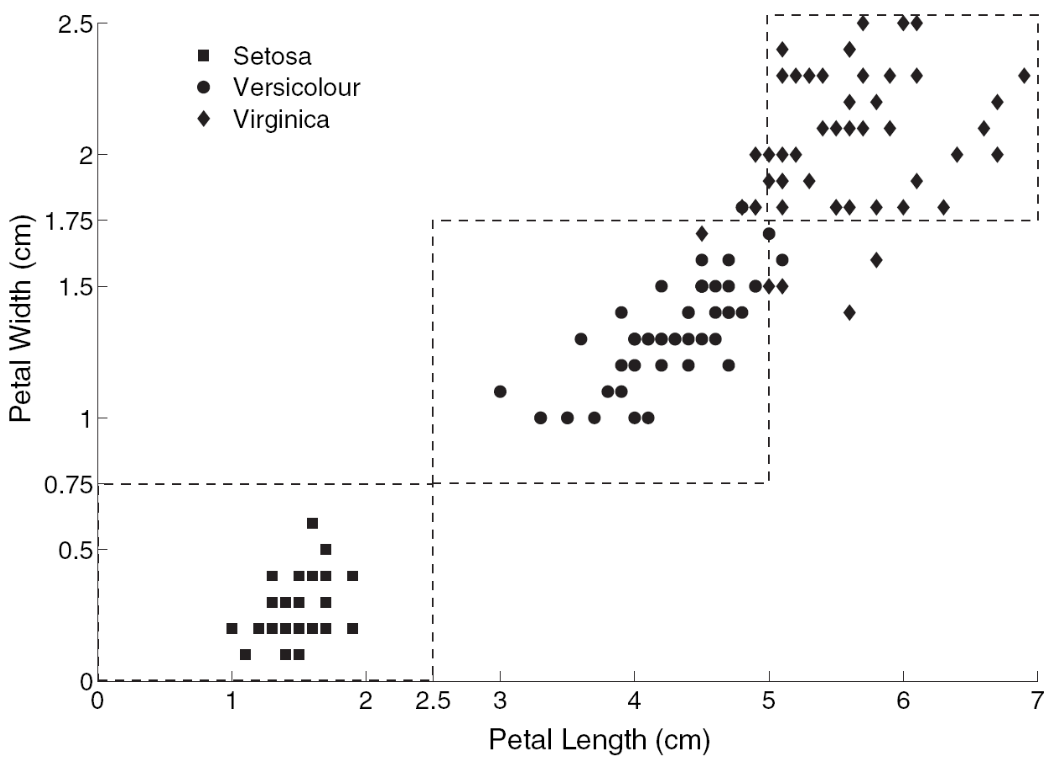


Numerical to categorical: discretization (cont.)



- **Equi-width** ranges (l_i – r_i is constant)
- **Equi-log** ranges ($log\ r_i$ – $log\ l_i$ is constant)
- **Equi-depth** ranges (num. items in $[l_i, r_i]$ constant)

Example discretization in IRIS dataset



Continuous variables are converted to three possible values per feature: small, medium, large

Try it!

- Given this database
- Create two categorical (ordinal) attributes
  - Salary_EW with salary binned into equi-width categories
  - Salary_ED with salary binned into equi-depth categories
- Each new attribute should have three values

Person	Salary
a	34,000
b	49,000
c	53,000
d	54,000
e	32,000
f	44,000
g	41,000
h	37,000
i	48,000

Categorical to numerical: binarization (one-hot encoding)

- One categorical value with K categories

  ⇒ indicator vector with K binary variables

User name	gender
alice	Female
bob	Male
cara	Female
...	...

alice	bob	cara	Female	Male	...
1	0	0	1	0	...
0	1	0	0	1	...
0	0	1	1	0	...
...	...	...	...	...	...

https://slideplayer.com/slide/14461691/

Series and sequences

Time series to discrete sequence

- Symbolic aggregate approximation (SAX)
  - Window-based averaging
    - Evaluate the average value in each window
  - Value-based discretization
    - Discretize the average value by equi-depth intervals
- How to ensure equi-depth without seeing the entire series?
  - Assume certain distribution, such as Gaussian
  - Estimate the distribution

Time series to numeric data

- Discrete Wavelet Transform (DWT)
- Discrete Fourier transform (DFT)

(Seen elsewhere, e.g., signal processing)

Discrete sequence to numeric

- Discrete sequence to a set of (binary) time series
  - ACACACTGTGACTG (4 Symbols)
  - 10101000001000 (A)
  - 01010100000100 (C)
  - 00000010100010 (T)
  - 00000001010001 (G)
- Map each of these time series into a multidimensional vector
- Features from the different series are combined

Graphs ⟷ Numerical

Convert any data type to a graph

- Determine distance $d(u,v)$ between all pairs of elements (u,v)
- All elements with $d(u,v) \leq \theta$ are connected

Graphs to numerical

- Graph embeddings
  - Each node is converted into a point in a low-dimensional space
  - Nearby nodes in the low-dimensional space are connected by short paths in the graph

We might see more on this on the *spectral graph clustering* topic, possibly (if we get to that topic)

Data integration

Data integration aspects

Remember: difference between DB schema and DB state

- Schema integration
  - Bring different schemata together
  - Equal concepts should be represented with equal types
- Object matching / Entity identification
  - Equal entities should be equally identified across datasets (unless re-identification forbidden by policy)

Data integration aspects (cont.)

- Redundancy analysis
  - Sometimes data needs to be integrated because different sets are row-incomplete
  - Sometimes those sets don't form a partition ⇒ there will be repeated entities to be removed
- Resolution of value conflicts
  - Same entity, different attribute values

Data cleaning

Why data cleaning?

- Data collection technologies are inaccurate
  - Sensors
  - Optical character recognition
  - Speech-to-text data
- Privacy reasons
- Manual errors
- Data collection is expensive and inaccurate

What is data cleaning?

It is a process by which data records are

modified or deleted

until each record passes

data validity criteria

Data validity criteria

- **Data-Type** constraints: values in a column must be of a particular datatype
- **Range** constraints: numbers or dates should fall within a certain range
- **Mandatory** constraints: certain columns cannot be empty.
- **Unique** constraints: a field, or a combination of fields, must be unique
- **Set-Membership** constraints: values in a column come from a set of discrete values or codes
- **Foreign-Key** constraints: set membership constraint where valid values in a column are defined in a column of another table that contains unique values
- **Regular expression patterns**: e.g., phone numbers [0-9]{9}
- **Cross-field validation**: certain conditions that utilize multiple fields must hold, e.g., percentages add up to 1.0 or to 100

Handling missing entries
Why is a value missing?

- Missing completely at random (MCAR)
  - Missingness of a value is independent of attributes
  - Fill in values based on the attribute
  - Analysis may be unbiased overall
- Missing at Random (MAR)
  - Missingness is related to other variables
  - Fill in values based other values
  - Almost always produces a bias in the analysis
- Missing Not at Random (MNAR)
  - Missingness is related to unobserved measurements
  - Informative or non-ignorable missingness
- In general, it is not possible to know the situation just from the data

Handling missing entries

- Delete the data record containing missing entries
- Estimate or Impute the Missing Values
  - Additional errors may be introduced
  - Good under certain conditions (e.g., Matrix Completion)
- Some algorithms can work with missing data

What would you do in cases of missing data? (be explicit on your assumptions)

- 5% of student records at a university have no “civil status” (single, married, ...)
  - Drop records? Impute value, how?
- 5% of smokers in a study of the effects of tobacco on health had no year of birth
  - Drop records? Impute value, how?
- 5% of records of sales of a company have zip code but no province
  - Drop records? Impute value, how?
- Temperature sensor at weather station was failing at random intervals during one day, total downtime 6 hours, max continuous downtime 15 minutes
  - Drop that day? Impute values, how?
- Same sensor failed during one night, downtime 6 hours continuous
  - Drop that day? Impute values, how?

Handling Incorrect and Inconsistent Entries

- Inconsistency detection
  - E.g., full name and abbreviation don’t match
- Domain knowledge
  - Human age cannot reach to 800 (yet?)
- Data-centric methods
  - Outlier detection

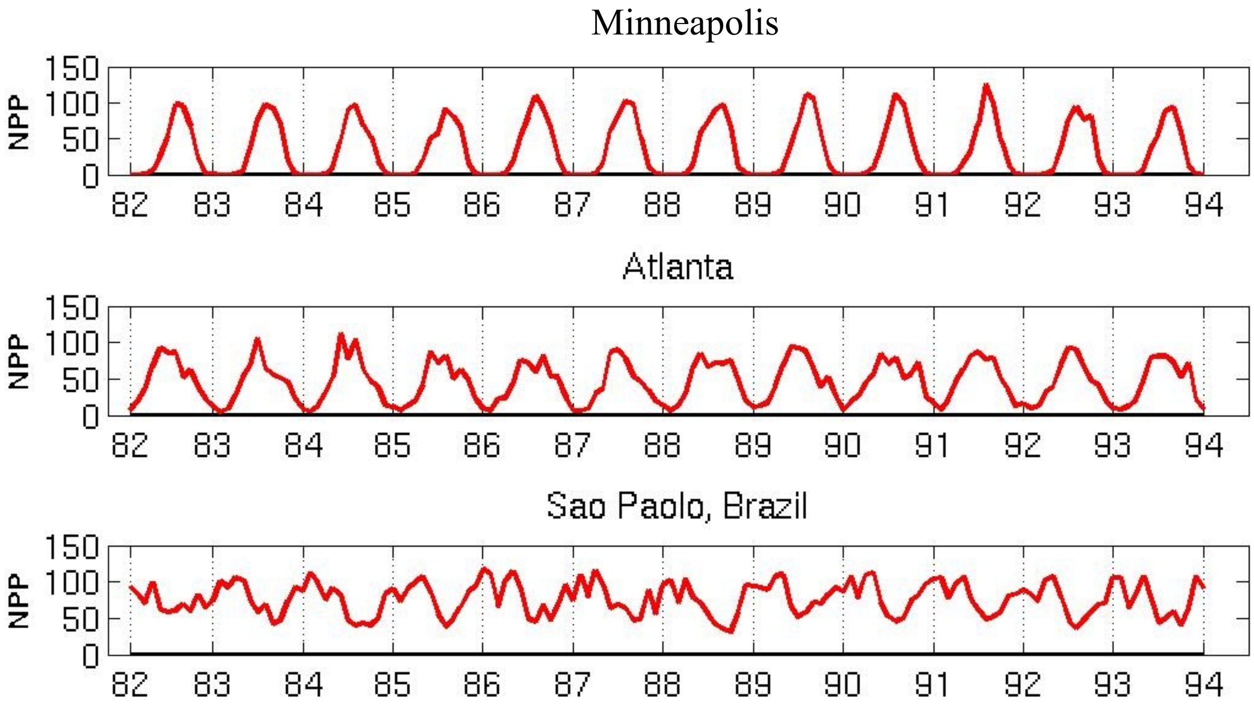
Scaling and normalization

- Features have different scales
  - Age versus Salary
- Standardization ("z-scoring")
  - Mean 0 and stdev 1
- Min-Max Scaling
  - Map to [0,1]
  - Sensitive to noise

$$z_i = \frac{x_i - \mu}{\sigma}$$

$$z_i = \frac{x_i - \min}{\max - \min}$$

Example: seasonal standardization



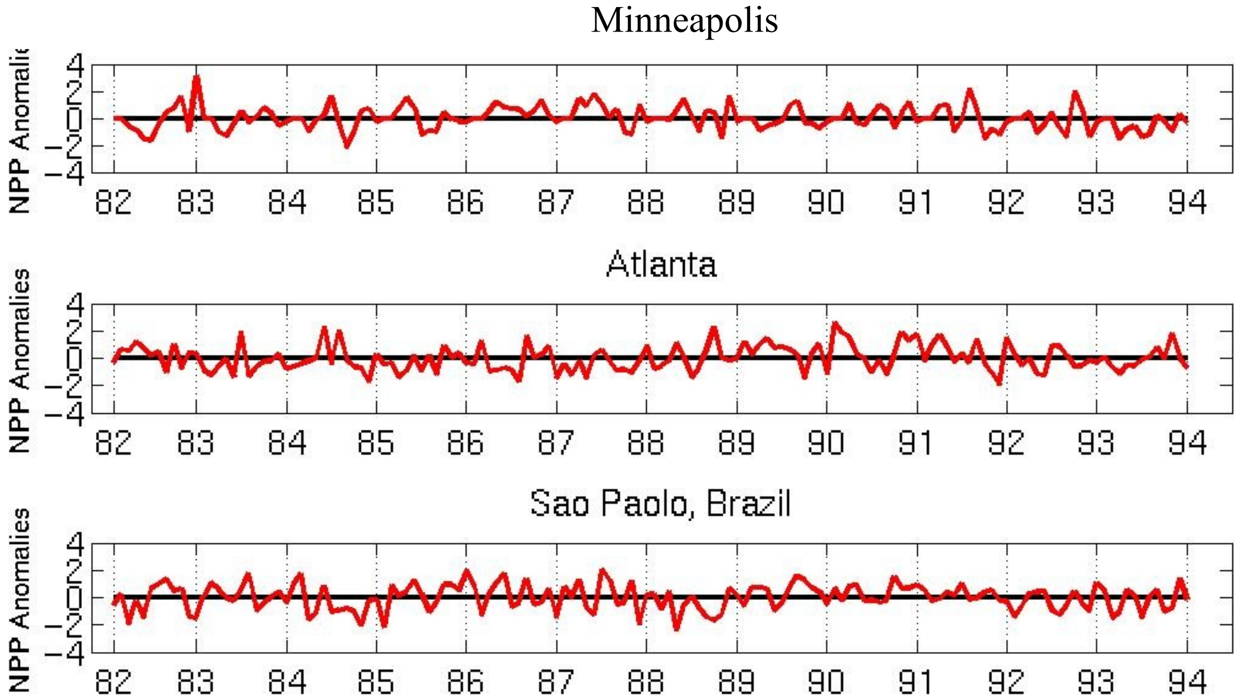
Net Primary Production (NPP) is a measure of plant growth used by ecosystem scientists.

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.7591	-0.7581
Atlanta	0.7591	1.0000	-0.5739
Sao Paolo	-0.7581	-0.5739	1.0000

Spurious correlations between time series

Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 2)

Example: seasonal standardization



Normalized using monthly Z Score:

Subtract off monthly mean and divide by monthly standard deviation

	Minneapolis	Atlanta	Sao Paolo
Minneapolis	1.0000	0.0492	0.0906
Atlanta	0.0492	1.0000	-0.0154
Sao Paolo	0.0906	-0.0154	1.0000

Adjusted correlations between time series

Introduction to Data Mining 2nd edition (2019) by Tan et al. (Chapter 2)

Data Reduction and Transformation

Data reduction and transformation

- Sampling

  ≃ “Less rows”

- Dimensionality Reduction

  ≃ “Less columns”

Why reduce/transform data?

- The Advantages
  - Reduce space complexity
  - Reduce time complexity
  - Reduce noise
  - Reveal hidden structures
    - E.g., manifold learning
- The Disadvantages
  - Information loss

Sampling for static data

- **Uniform** random sampling
  - with/without replacement
- **Biased** sampling
  - e.g., emphasize recent items
- **Stratified** sampling
  - Partition data in strata, sample in each stratum

Sampling example

- There are 10000 people which contain 100 millionaires
- Uniform random sample of 100 people
  - In expectation, one millionaire will be sampled
  - There is $\simeq$ 37% chance no millionaires are sampled, why?
- Stratified Sampling
  - Unbiased Sampling 1 from 100 millionaires
  - Unbiased Sampling 99 from remaining

Sampling from data streams

- The setting
  - Data arrive sequentially
  - We want sample of them uniformly
  - There is a reservoir that can hold k data points
- The algorithm: reservoir sampling
  - The first k data points are kept
  - Insert the n-th data point with probability k/n
  - Drop one of the existing data points uniformly at random
  - More on this in the sequence mining lecture …

Reducing data dimensionality

Note: PCA/SVD covered well in other courses, won’t be part of our exam

Feature selection

- **Unsupervised** Feature Selection
  - Using the performance of unsupervised learning (e.g, clustering) to guide the selection
- **Supervised** Feature Selection
  - Using the performance of supervised learning (e.g., classification) to guide the selection

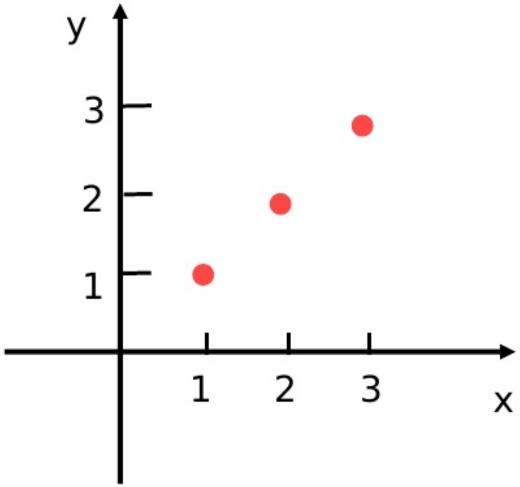
Dimensionality reduction with axis rotation (perfect case)

- Motivation: three points in a line in two-dimensional space

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$



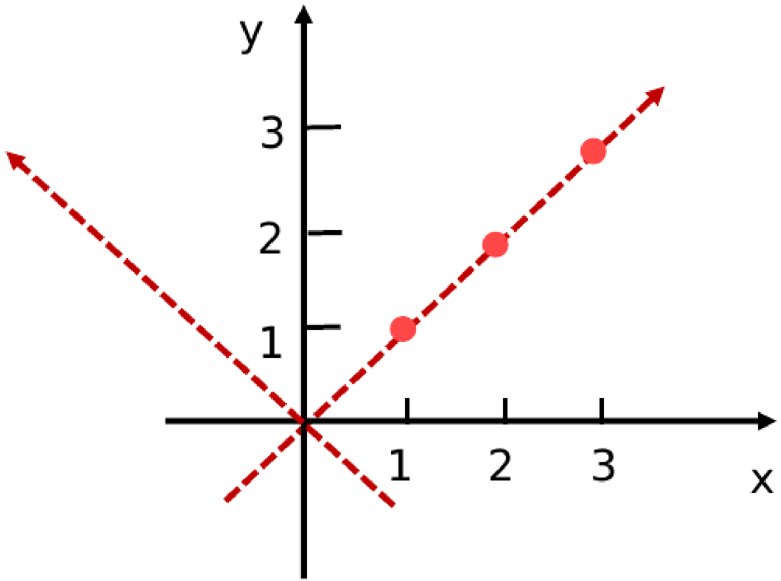
Dimensionality reduction with axis rotation (perfect case, cont.)

- Coordinates after axes rotation

$$\mathbf{x}_1 = \begin{bmatrix} \sqrt{2} \\ 0 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2\sqrt{2} \\ 0 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 3\sqrt{2} \\ 0 \end{bmatrix}$$



Dimensionality reduction with axis rotation (perfect case, cont.)

- Coordinates after axes rotation

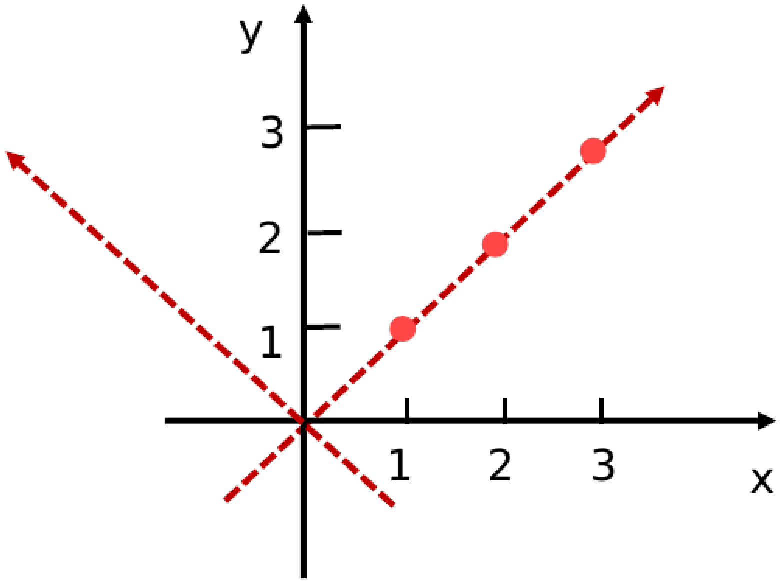
Drop second coordinate, **no** information is lost.

2D data reduced to 1D data

$$\mathbf{x}_1 = \begin{bmatrix} \sqrt{2} \\ \cancel{0} \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2\sqrt{2} \\ \cancel{0} \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 3\sqrt{2} \\ \cancel{0} \end{bmatrix}$$



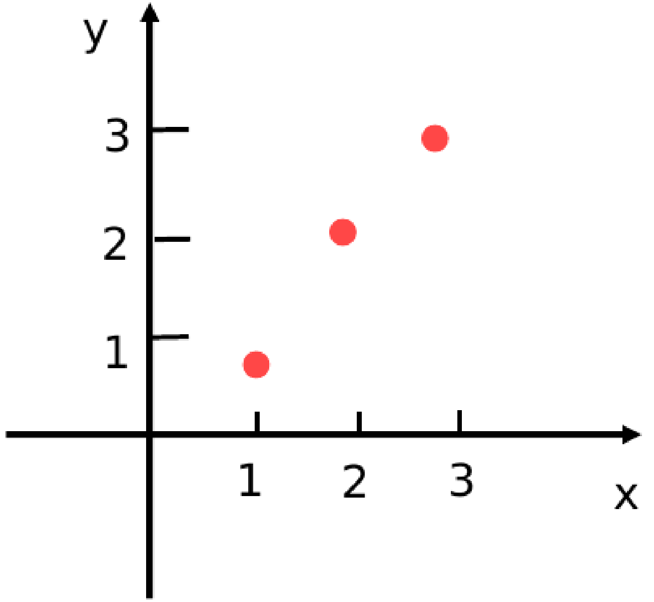
Dimensionality reduction with axis rotation (noisy case)

- Suppose points don't lie exactly on a line

$$\mathbf{x}_1 = \begin{bmatrix} 1 \\ 0.9 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2.1 \\ 2 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 2.9 \\ 3.1 \end{bmatrix}$$



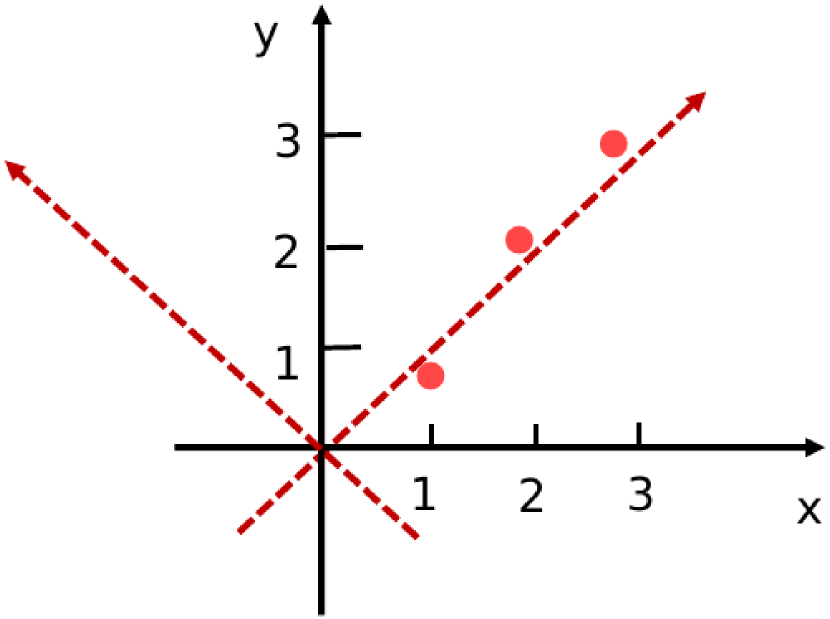
Dimensionality reduction with axis rotation (noisy case, cont.)

- Suppose points don't lie exactly on a line

$$\mathbf{x}_1 = \begin{bmatrix} 1.34 \\ 0.07 \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2.89 \\ 0.07 \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 4.24 \\ -0.14 \end{bmatrix}$$



Dimensionality reduction with axis rotation (noisy case, cont.)

- Suppose points don't lie exactly on a line

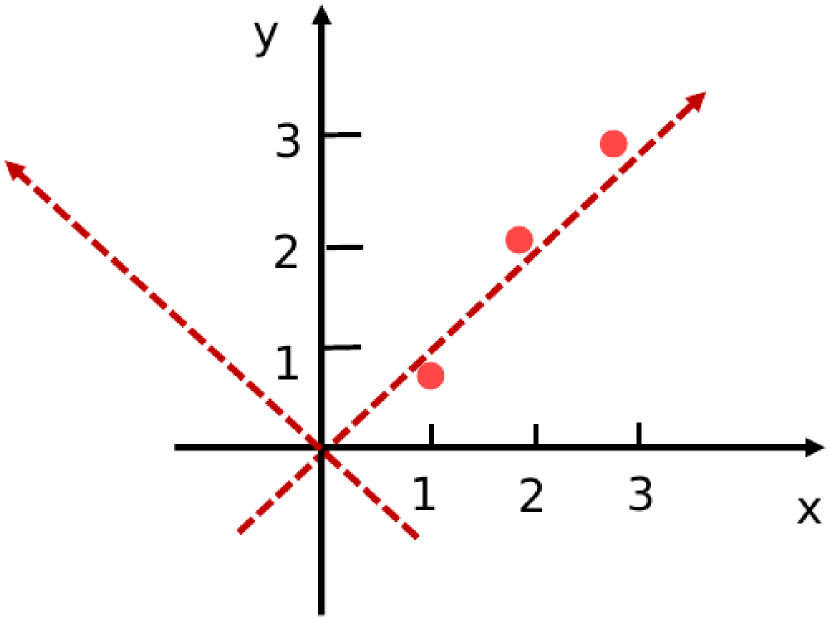
Drop second coordinate, **some** information is lost.

2D data reduced to 1D data

$$\mathbf{x}_1 = \begin{bmatrix} 1.34 \\ \cancel{0.07} \end{bmatrix}$$

$$\mathbf{x}_2 = \begin{bmatrix} 2.89 \\ \cancel{0.07} \end{bmatrix}$$

$$\mathbf{x}_3 = \begin{bmatrix} 4.24 \\ \cancel{-0.14} \end{bmatrix}$$



How does this work in reality?

- Change of axes removes correlations and reduces dimensionality
- Techniques
  - Principal Component Analysis (PCA)
  - Singular-Value Decomposition (SVD)

    (Seen in other courses)

Axis rotation - formulation

- Points are described with respect to the standard basis

$$\mathbf{x} = \begin{bmatrix} x^1 \\ x^2 \\ \vdots \\ x^d \end{bmatrix} \in \mathbb{R}^d \Longleftrightarrow \mathbf{x} = x^1\mathbf{e}_1 + x^2\mathbf{e}_2 + \cdots + x^d\mathbf{e}_d$$

Axis rotation – formulation (cont.)

New coordinates under orthonormal basis { w_1, w_2, …, w_d }:

$W = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_d]$ is a orthonormal matrix

$$\mathbf{x} = WW^\top \mathbf{x} = \left(\sum_{i=1}^{d} \mathbf{w}_i \mathbf{w}_i^\top \right) \mathbf{x} = \sum_{i=1}^{d} \mathbf{w}_i \left(\mathbf{w}_i^\top \mathbf{x} \right)$$

$$= (\mathbf{w}_1^\top \mathbf{x}) \mathbf{w}_1 + (\mathbf{w}_2^\top \mathbf{x}) \mathbf{w}_2 + \cdots + \left(\mathbf{w}_d^\top \mathbf{x} \right) \mathbf{w}_d$$

Thus, the new coordinates are

$$\mathbf{y} = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{x} \\ \mathbf{w}_2^\top \mathbf{x} \\ \vdots \\ \mathbf{w}_d^\top \mathbf{x} \end{bmatrix} \in \mathbb{R}^d$$

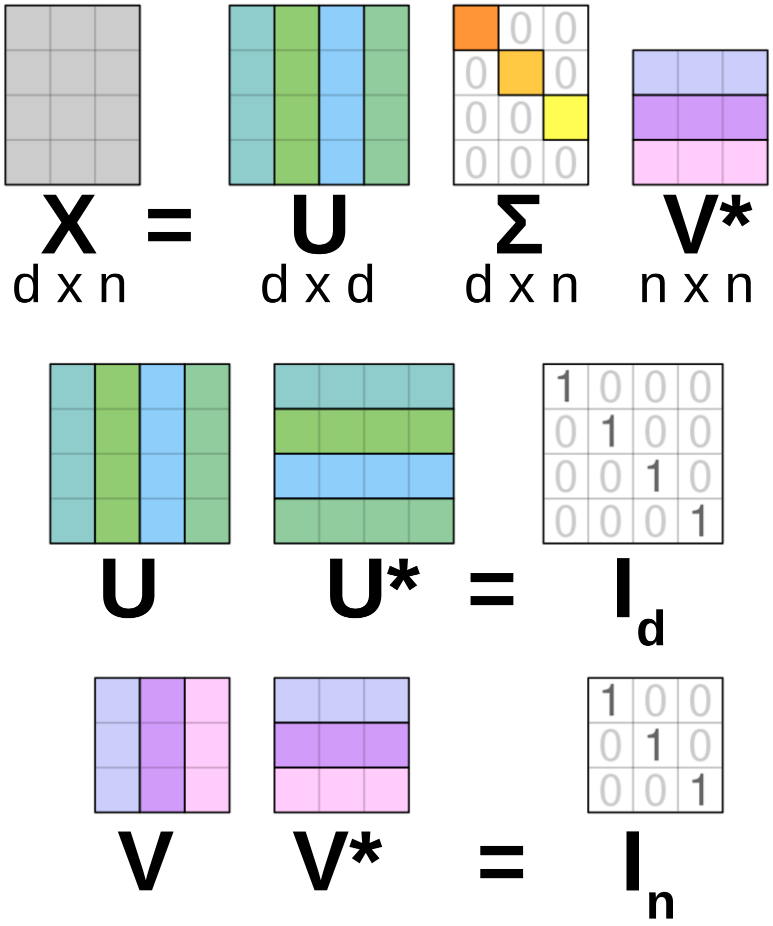
We will drop some dimensions from here, as we did previously

PCA formulation: optimization

- Find new basis { w_1, w_2, …, w_k }, with $k \leq d$ such that the variance of this set is maximized:

$$\left\{ \mathbf{y}_1 = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{x}_1 \\ \mathbf{w}_2^\top \mathbf{x}_1 \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_1 \end{bmatrix}, \mathbf{y}_2 = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{x}_2 \\ \mathbf{w}_2^\top \mathbf{x}_2 \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_2 \end{bmatrix}, \cdots, \mathbf{y}_n = \begin{bmatrix} \mathbf{w}_1^\top \mathbf{x}_n \\ \mathbf{w}_2^\top \mathbf{x}_n \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_n \end{bmatrix} \right\}$$

SVD formulation



- U and V are rotation matrices; Σ is a scaling matrix
- The rotated data is obtained by multiplying U^TX

Algorithms for PCA and SVD

- PCA
  1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$
  2. Calculate the covariance matrix $C = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\top$
  3. Calculate the k-largest eigenvectors of C

- SVD
  1. Calculate the mean vector $\bar{\mathbf{x}} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i$
  2. Calculate the k largest left singular vectors of $\bar{X} = [\mathbf{x}_1 - \bar{\mathbf{x}}, \dots, \mathbf{x}_n - \bar{\mathbf{x}}]$

Summary

Things to remember

- Converting across data types
- Data cleaning
  - Specially: when and how to impute missing values
- Data sampling methods
- Data transformations

Exercises for this topic

- Exercises 3.7 of Data Mining Concepts and Techniques, 3rd edition (2011) by Han et al.
- Exercises 2.6 of Introduction to Data Mining, Second Edition (2019) by Tan et al.
  - Mostly the first exercises, say 1-6