Data pre-processing operations – part 2

M. Cristina

SCC5836/0252 Visualização Computacional

Previous class

- Missing data
- Data interpolation
- Data distribution
- Outlier identification
- Feature scaling

This class

- Assess correlation
- Assess data (dis)similarity
- High-dimensional data
- Dimension reduction - PCA

Correlation

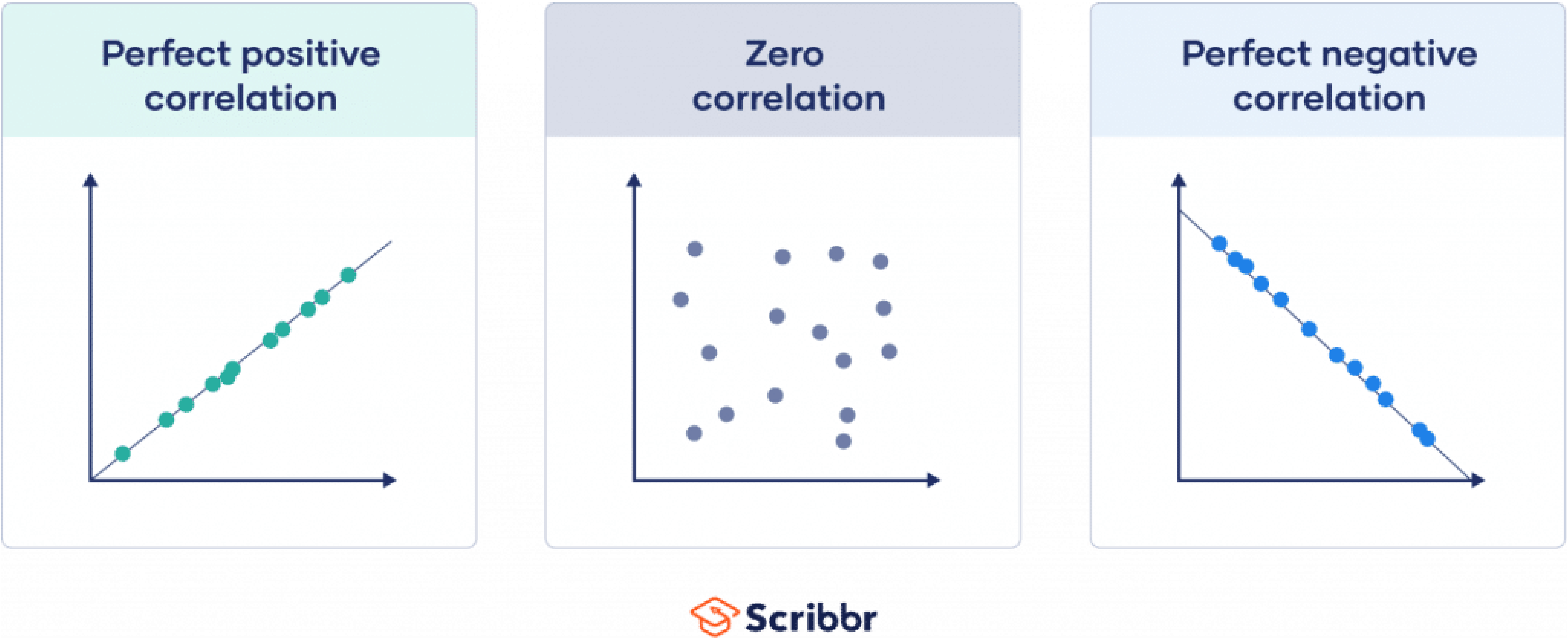
- Features within a dataset can be related for lots of reasons. For example:
  - One variable could cause or depend on the values of another variable.
  - One variable could be lightly associated with another variable.
  - Two variables could depend on a third unknown variable.

- Feature correlation may indicate redundancy in information

Correlation

- – **Positive Correlation**: both variables change in the same direction.
- – **Neutral Correlation**: no relationship in the change of the variables. (unrelated)
- – **Negative Correlation**: variables change in opposite directions.

Correlation



Measuring correlation

- Covariance

- Pearson correlation

- Spearman correlation

- …

Covariance

- between two variables (given *n* samples)

$$Cov(X, Y) = \frac{\sum(X_i - \bar{X})\left(Y_j - \bar{Y}\right)}{n - 1}$$

- Linear relationship, Gaussian-like distribution
- The sign of the covariance can be interpreted as whether the two variables change in the same direction (positive) or change in different directions (negative).
- A covariance value of zero indicates that both variables are completely independent.
- The magnitude of the covariance is not easily interpreted.

Ex. Covariance matrix of two variables

X = [0 1 2]

Y = [2 1 0]

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Pearson correlation

- named for Karl Pearson – a measure of linear correlation

$$Pearson_{corr} = \frac{Cov(X,Y)}{stdv(X) * stdv(Y)}$$

- returns value between -1 and 1 that represents the limits of correlation from a full negative correlation to a full positive correlation.

Pearson correlation

- Assumptions
  - Both variables quantitative and normally distributed with no outliers
  - Data from a random or representative sample
  - You expect a linear relationship

- If any of these assumptions are violated, should consider a rank correlation measure, e.g., Spearman, Kendall...

Pearson correlation

Correlation coefficient	Correlation strength	Correlation type
-.7 to -1	Very strong	Negative
-.5 to -.7	Strong	Negative
-.3 to -.5	Moderate	Negative
0 to -.3	Weak	Negative
0	None	Zero
0 to .3	Weak	Positive
.3 to .5	Moderate	Positive
.5 to .7	Strong	Positive
.7 to 1	Very strong	Positive

https://www.scribbr.com/statistics/correlation-coefficient/

Spearman correlation

- Two variables may be related by a nonlinear relationship
- Further, they may have a non-Gaussian distribution
  - use the Spearman’s correlation coefficient (named for Charles Spearman) to summarize the strength between the two data samples.
  - can also be used if there is a linear relationship between the variables, but will have slightly less power (e.g. may result in lower coefficient scores)

Spearman correlation

- The scores are between -1 and 1 for perfectly negatively correlated variables and perfectly positively correlated, respectively.

- These statistics are calculated from the relative rank of values on each sample.
  - common approach in non-parametric statistics, e.g. statistical methods where we do not assume data has a Gaussian distribution.

Spearman correlation

- A correlation coefficient is a descriptive (bivariate or multivariate) statistic
- Values obtained based on sample data do not necessarily generalize to the population
  - Test statistics (F test, t-test) to learn about statistical significance

Spearman correlation

- Test of statistical significance
- P-value: a measure of how likely or probable it is that any observed correlation is due to chance.
  - P-values range between 0 (0%) and 1 (100%). A p-value close to 1 suggests no correlation other than due to chance, p-values close to 0 indicate the observed correlation is unlikely to be due to chance.

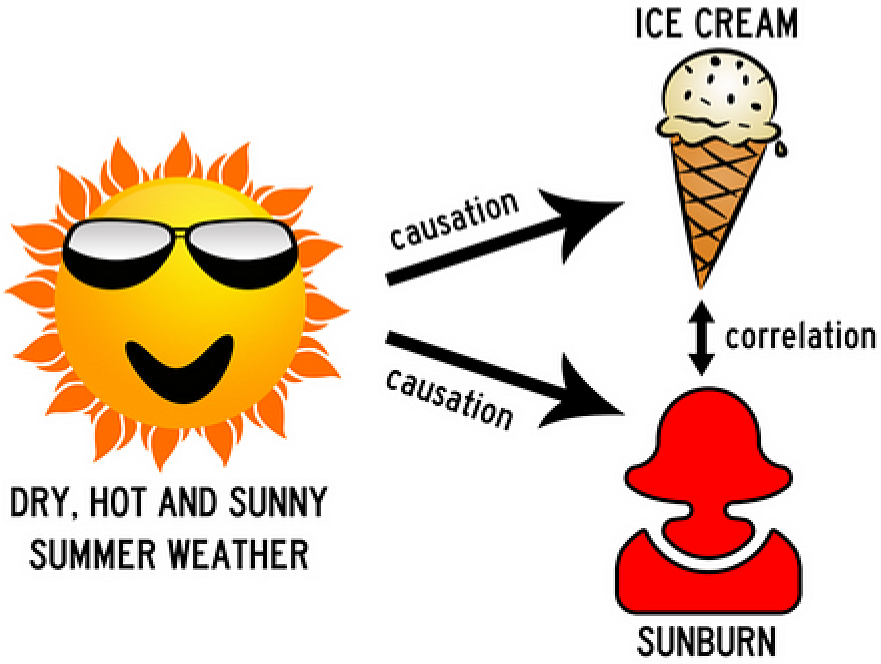
Correlation vs causation

- “Correlation does not imply causation”
  - **Correlation** means there is a statistical association between variables.
  - **Causation** means that a change in one variable causes a change in another variable.

https://www.scribbr.com/methodology/correlation-vs-causation/

Correlation vs causation

- https://www.tylervigen.com/spurious-correlations
- https://towardsdatascience.com/correlation-is-not-causation-ae05d03c1f53



Assessing data (dis)similarity

- Data items described by multiple (quantitative attributes)
  - Spatialization: items as points in a (multi)dimensional space
  - Geometry => graphical representation = visualization
  - Ex. https://www.youtube.com/watch?v=wvsE8jm1GzE

Distances & similarities

- Proximity in space as a proxy to `similarity´ (or `dissimilarity´)
  - Data `points´ that are close in space are similar
  - Data `points´ that are far away in space are dissimilar
- Similar to `clustering´ algorithms in machine learning

- Distance between points to quantify (dis)similarity
  - Short distances (low values of distance function) => higher similarity
  - High distances (high values of distance function) => higher dissimilarity

Distances & similarities

- A distance function is a metric if it satisfies the properties of
  - Non-negativity
  - Identity
  - Symetry
  - Triangular inequality

- All metrics are distances, but not all distance functions are metric

Distances & similarities

- Once we measure (dis)similarity
  - We can identify groups of data items (cluster, visualization)
  - Analyze patterns in groups, explain behavior of groups
  - Infer an organization for the data
  - Consider a new item in relation to existing groups
  - Employ machine learning algorithms to generate descriptive/predictive models

Computing distances

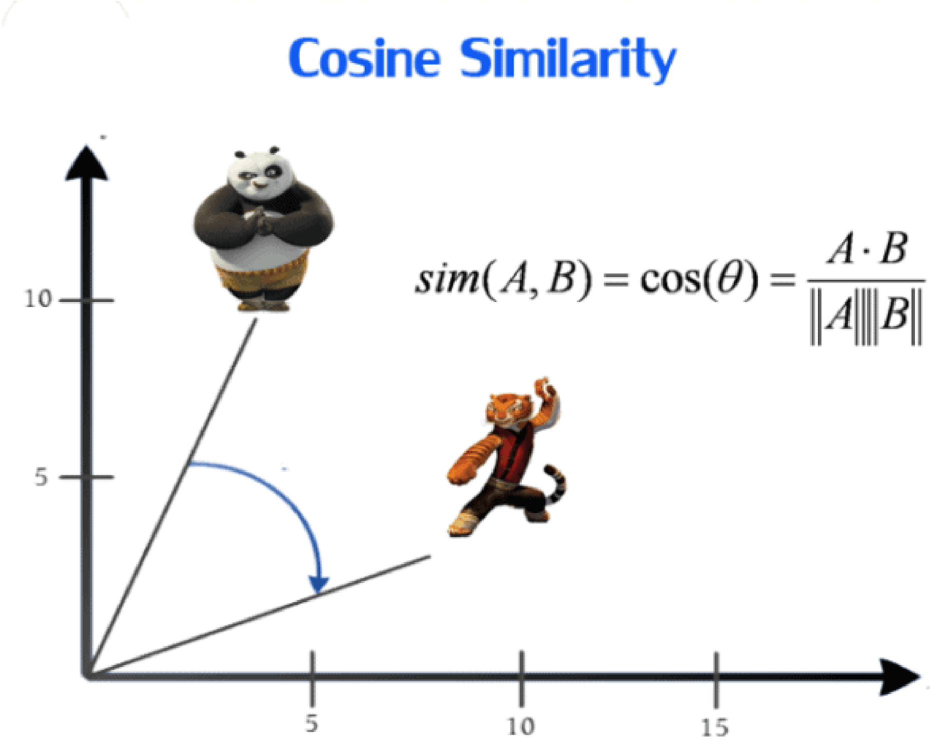
- Depends on the type of variable
  - Quantitative
  - Binary
  - Categorical: nominal/ordinal

Distances & similarities

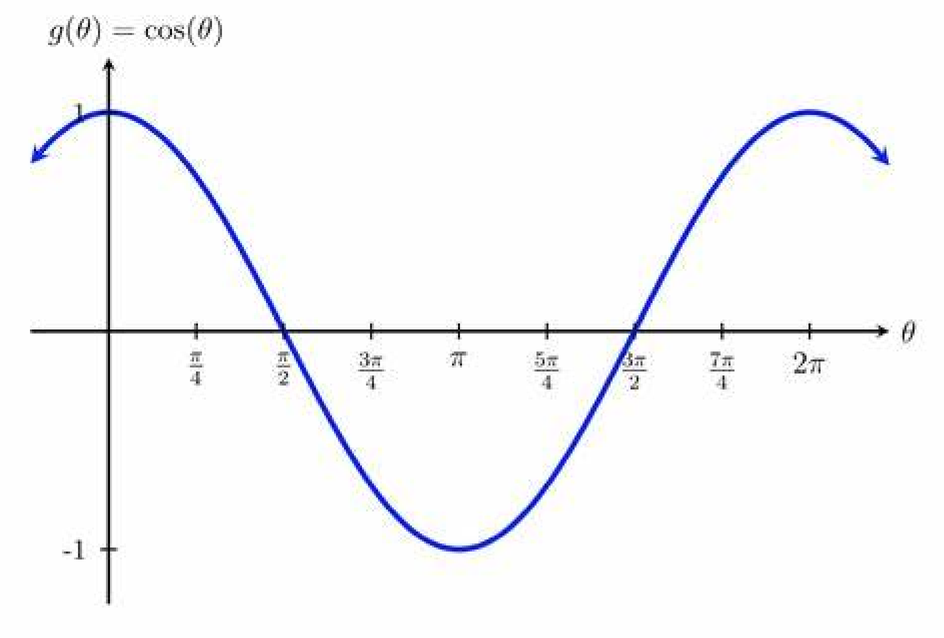
- Quantitative
  - Minkowski family of distances
    - Euclidean, Manhattan, Chebyshev
  - Cosine similarity
  - Correlation coefficient
  - ...

Distances & similarities

$$0 \leq \theta \leq \frac{\pi}{2}$$

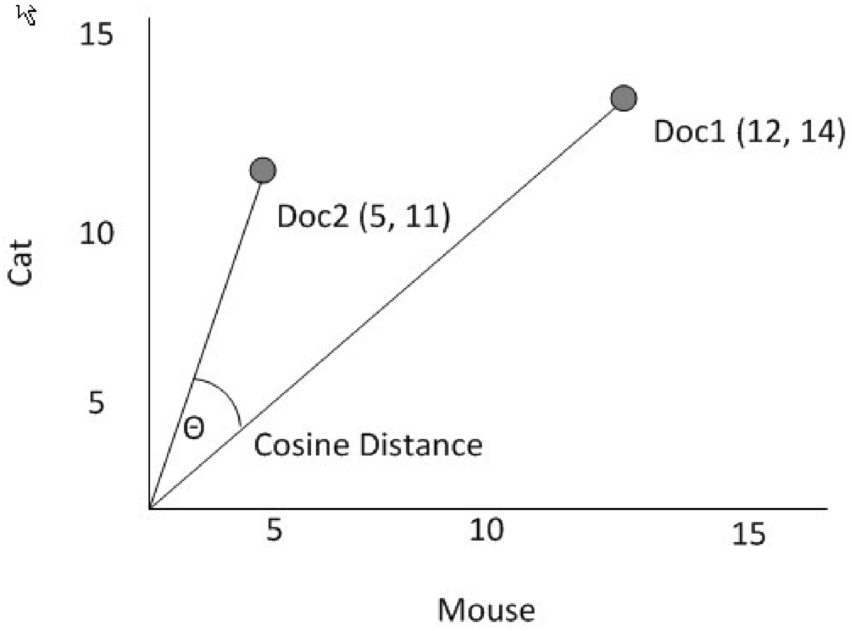


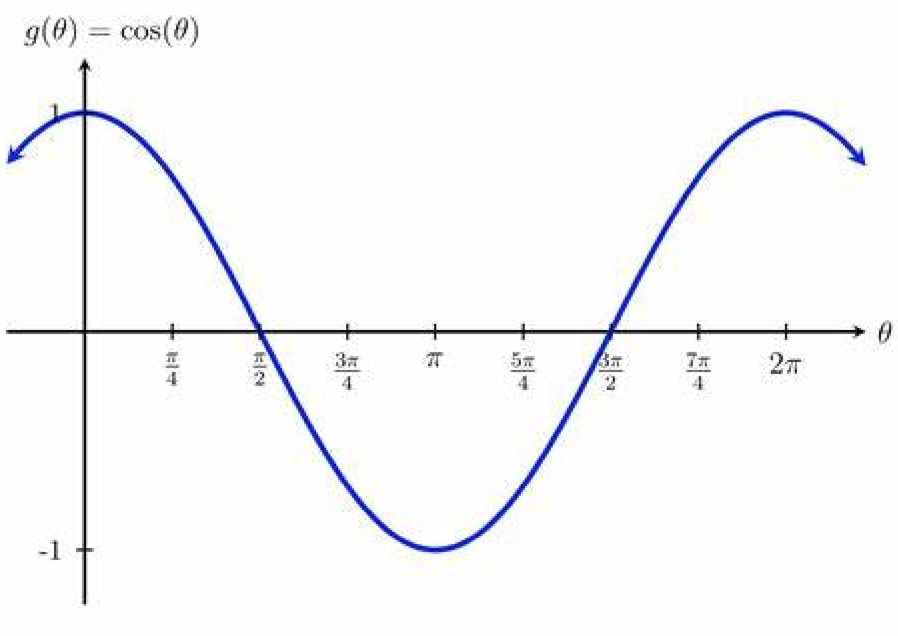
$$sim(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$



Distances & similarities

$$0 \leq \ \theta \ \leq \frac{\pi}{2}$$



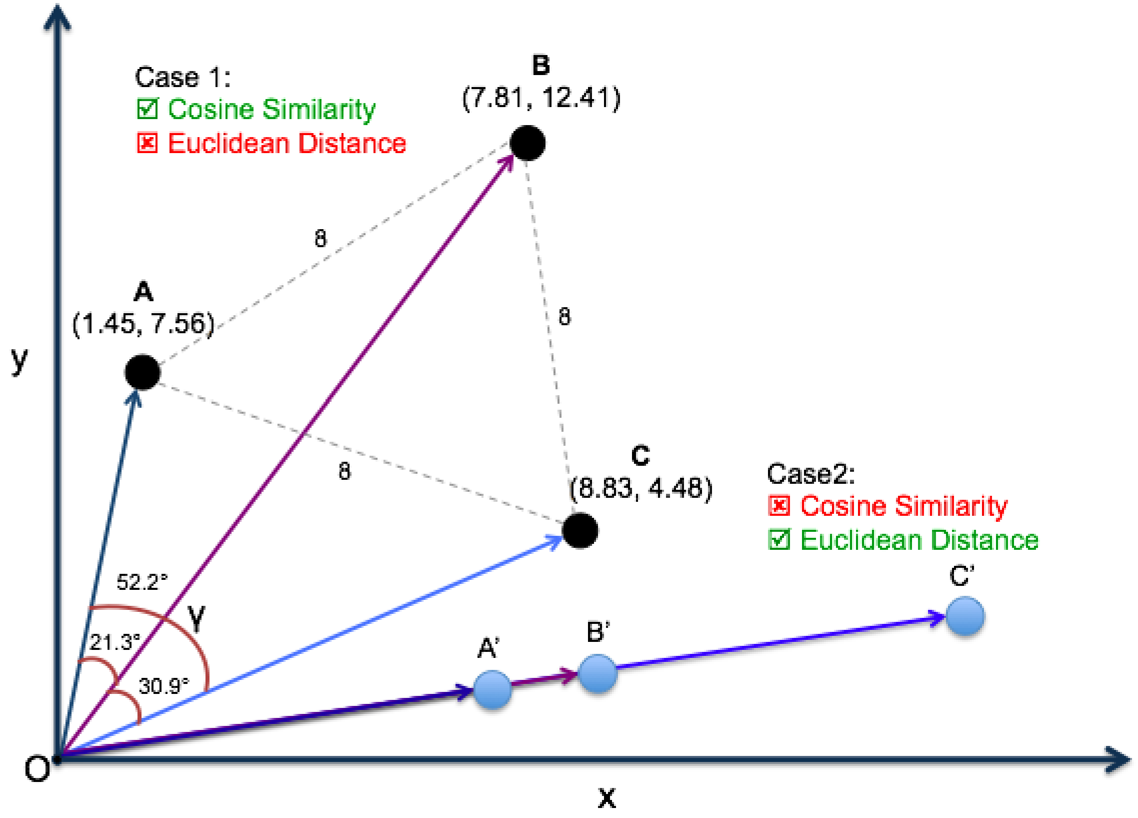


$$\cos(\theta) = sim(Doc1, Doc2)$$
$$\mathrm{disim}(\mathrm{Doc1}, \mathrm{Doc2}) = 1 \ - \mathrm{sim}(\mathrm{Doc1}, \mathrm{Doc2})$$

Euclidean vs cosine

- Euclidean distances between points **A**, **B**, **C** all equal (8), cosine similarities are different

- Cosine similarities between points **A'**, **B'**, **C'** all equal (1, or 0 distance), Euclidean distances are different



Source: https://medium.com/@sasi24/cosine-similarity-vs-euclidean-distance-e5d9a9375fc8

Euclidean vs cosine

- When to use cosine?
  - Great illustrated discussion here: https://cmry.github.io/notes/euclidean-v-cosine
  - “Cosine similarity is generally used as a metric for measuring distance when the magnitude of the vectors does not matter. This happens for example when working with text data represented by word counts.”
  - See section Cosine in action

Categorical variables

- Many algorithms take numerical variables only => encoding categorical variables
  - **Ordinal** values can be directly mapped to discrete numbers
  - Non-ranked values require specific strategies, e.g. “**one-hot encoding**”

- See https://www.analyticsvidhya.com/blog/2020/08/types-of-categorical-data-encoding/

Categorical variables

one hot encoding
creates `dummy variables´

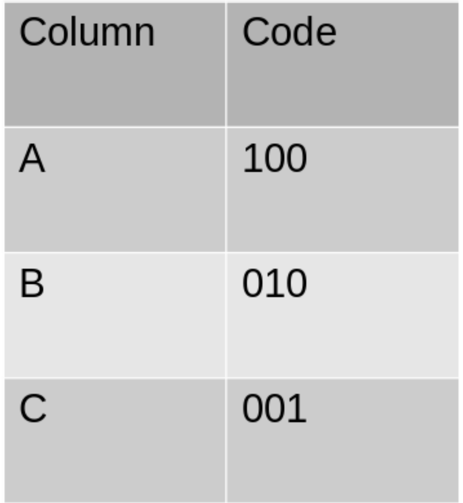
Index	Animal
0	Dog
1	Cat
2	Sheep
3	Horse
4	Lion

One-Hot code

Index	Dog	Cat	Sheep	Lion	Horse
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	0	1
4	0	0	0	1	0

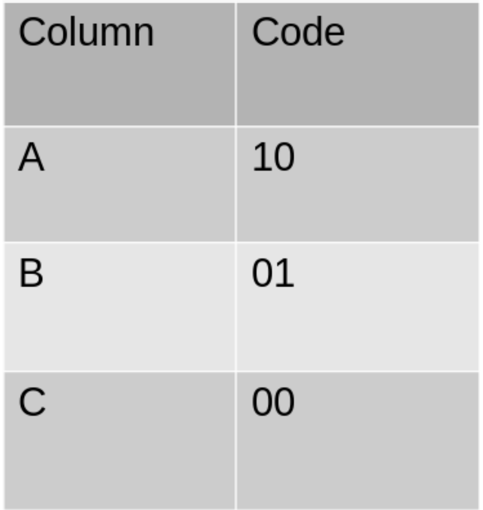
Categorical variables

dummy encoding: N-1 features to represent N labels/categories



Column	Code
A	100
B	010
C	001

One- Hot Coding



Column	Code
A	10
B	01
C	00

Dummy Code

Distances & similarities

- Binary variables
  - Simple matching
  - Jaccard
  - Hamming
  - ...

Distances & similarities

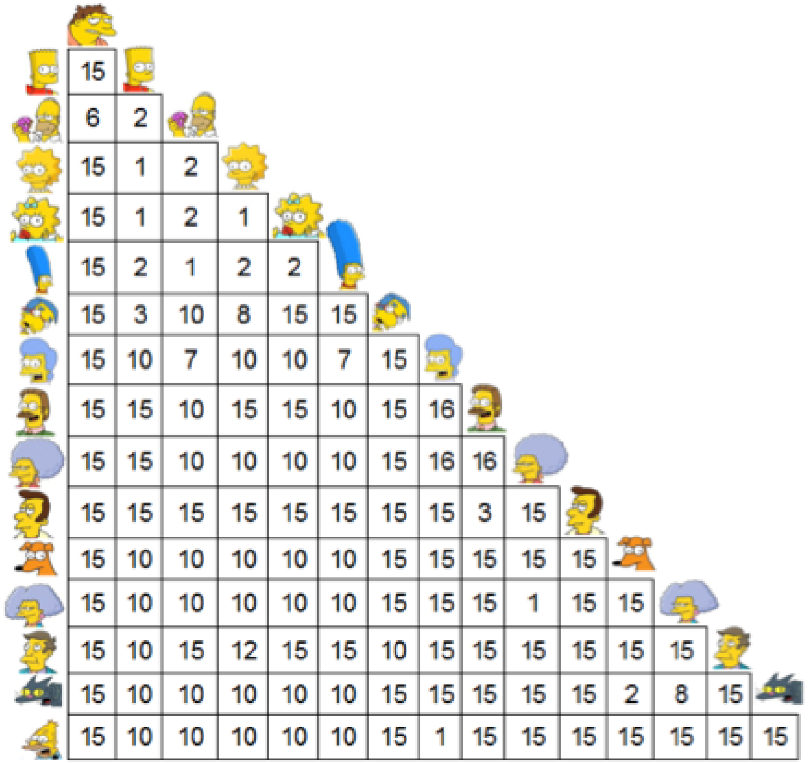
- Good tutorial:

- Similarity Measurement (revoledu.com)

Distances & similarities

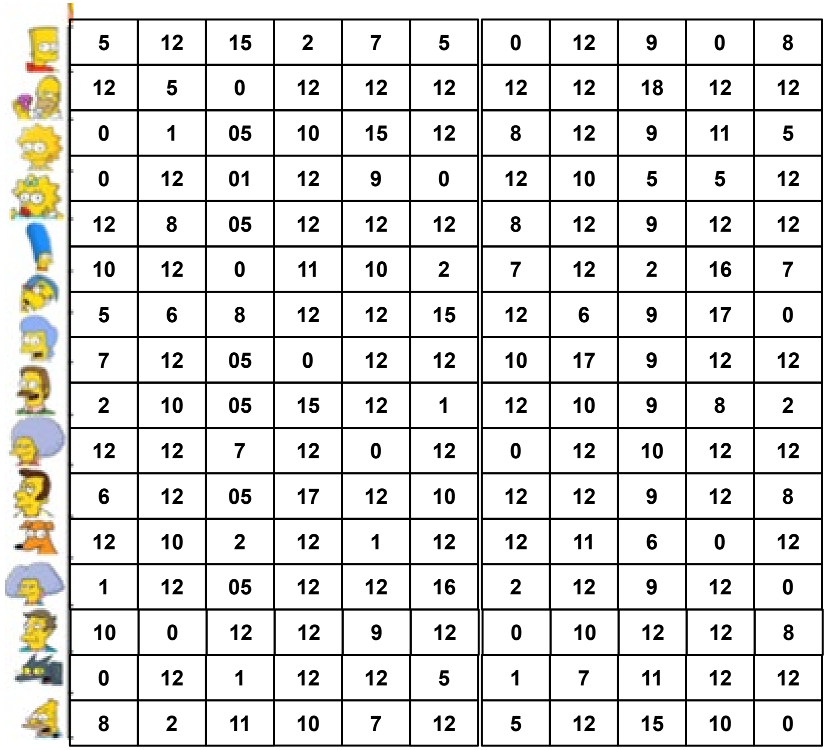
- If you have *n* data points described by d features: *d*-dimensional space representation given by *n* x *d* matrix
- Eventually, you may compute a dissimilarity measure in this space
  - Yields another representation of your data: *n* x *n* dissimilarity matrix
  - This computation is computationally expensive $O(n^2)$
- Space embedding: obtain the embedding matrix from the distance matrix

Data representation



15														
6	2													
15	1	2												
15	1	2	1											
15	2	1	2	2										
15	3	10	8	15	15									
15	10	7	10	10	7	15								
15	15	10	15	15	10	15	16							
15	15	10	10	10	10	15	16	16						
15	15	15	15	15	15	15	15	3	15					
15	10	10	10	10	10	15	15	15	15	15				
15	10	10	10	10	10	15	15	15	1	15	15			
15	10	15	12	15	15	10	15	15	15	15	15	15		
15	10	10	10	10	10	15	15	15	15	15	2	8	15	
15	10	10	10	10	10	15	1	15	15	15	15	15	15	15

n x *n* (symmetric) distance matrix



5	12	15	2	7	5	0	12	9	0	8
12	5	0	12	12	12	12	12	18	12	12
0	1	05	10	15	12	8	12	9	11	5
0	12	01	12	9	0	12	10	5	5	12
12	8	05	12	12	12	8	12	9	12	12
10	12	0	11	10	2	7	12	2	16	7
5	6	8	12	12	15	12	6	9	17	0
7	12	05	0	12	12	10	17	9	12	12
2	10	05	15	12	1	12	10	9	8	2
12	12	7	12	0	12	0	12	10	12	12
6	12	05	17	12	10	12	12	9	12	8
12	10	2	12	1	12	12	11	6	0	12
1	12	05	12	12	16	2	12	9	12	0
10	0	12	12	9	12	0	10	12	12	8
0	12	1	12	12	5	1	7	11	12	12
8	2	11	10	7	12	5	12	15	10	0

n x *d* data matrix (embedding)

Distances & similarities

- Issue to pay attention
  - Data in high-dimensional spaces
  - Scenarios where *d* >> *n* (sparse spaces)
  - Distances in high-dimensional spaces do not behave as our intuition tells (from observing low-dimensional Cartesian spaces)

Distances & similarities

- High-dimensional spaces display properties that are against our intuition
  - Highly sparse... lots of empty spaces
  - **Relative contrast**: difference between the maximum and the minimum (Euclidean) distances tends to zero as dimensionality increases
  - Concentration of distances: pairwise distances nearly the same for all points...
  - Concept of nearest-neighbor not meaningful: small perturbation can change the nearest point into the farthest one

  - http://www.mariovalle.name/CrystalFp/uploads/CrystalFpLib/high-dim-spaces.pdf

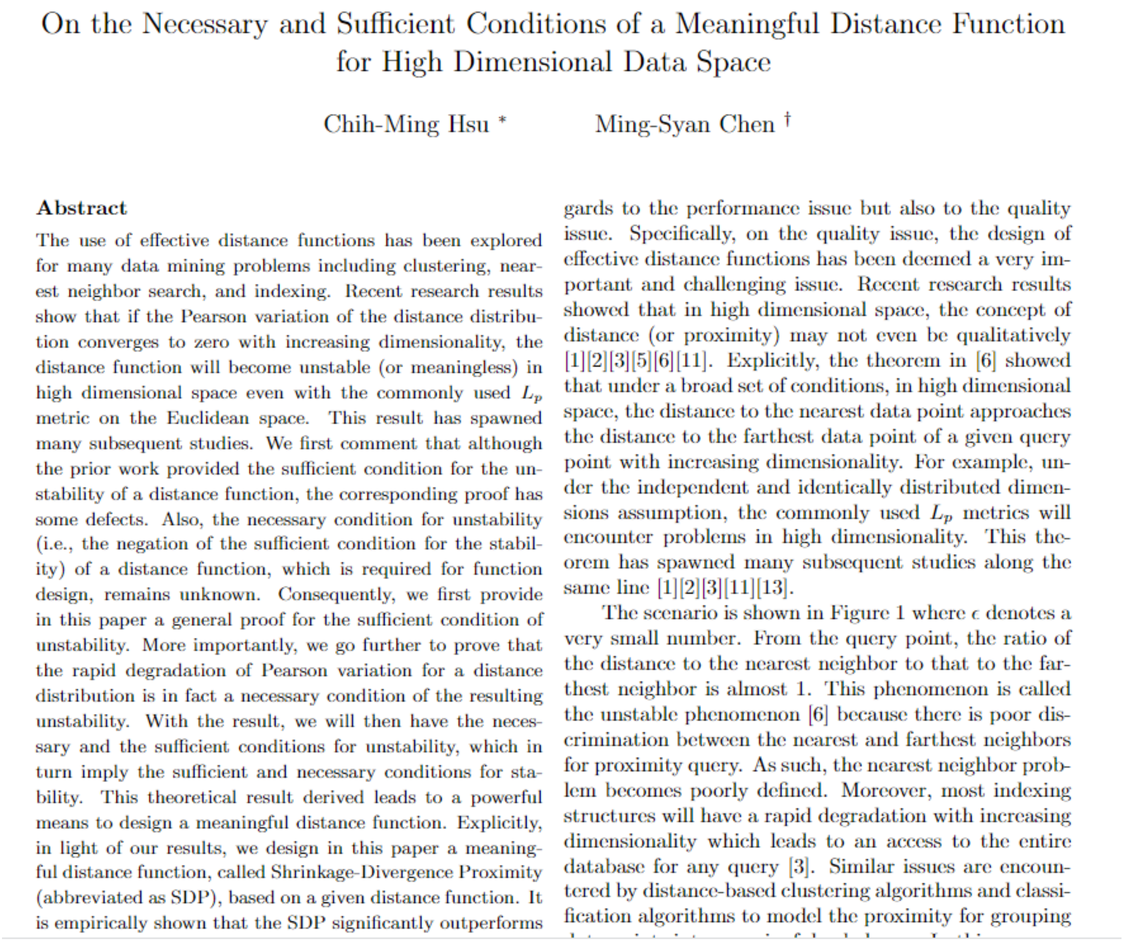
Distances & similarities

- Euclidean distance behaves poorly in high-dimensions
    - Select a distance function appropriate
    - Minkowski norm, cosine distance, ...

- Know your data: high-dimensional data is not always hd
    - effective/intrinsic dimensionality vs embedding dimensionality

Distances & similarities

- When a distance function is a good one?



On the Necessary and Sufficient Conditions of a Meaningful Distance Function for High Dimensional Data Space

Chih-Ming Hsu [*]   Ming-Syan Chen [†]

Abstract

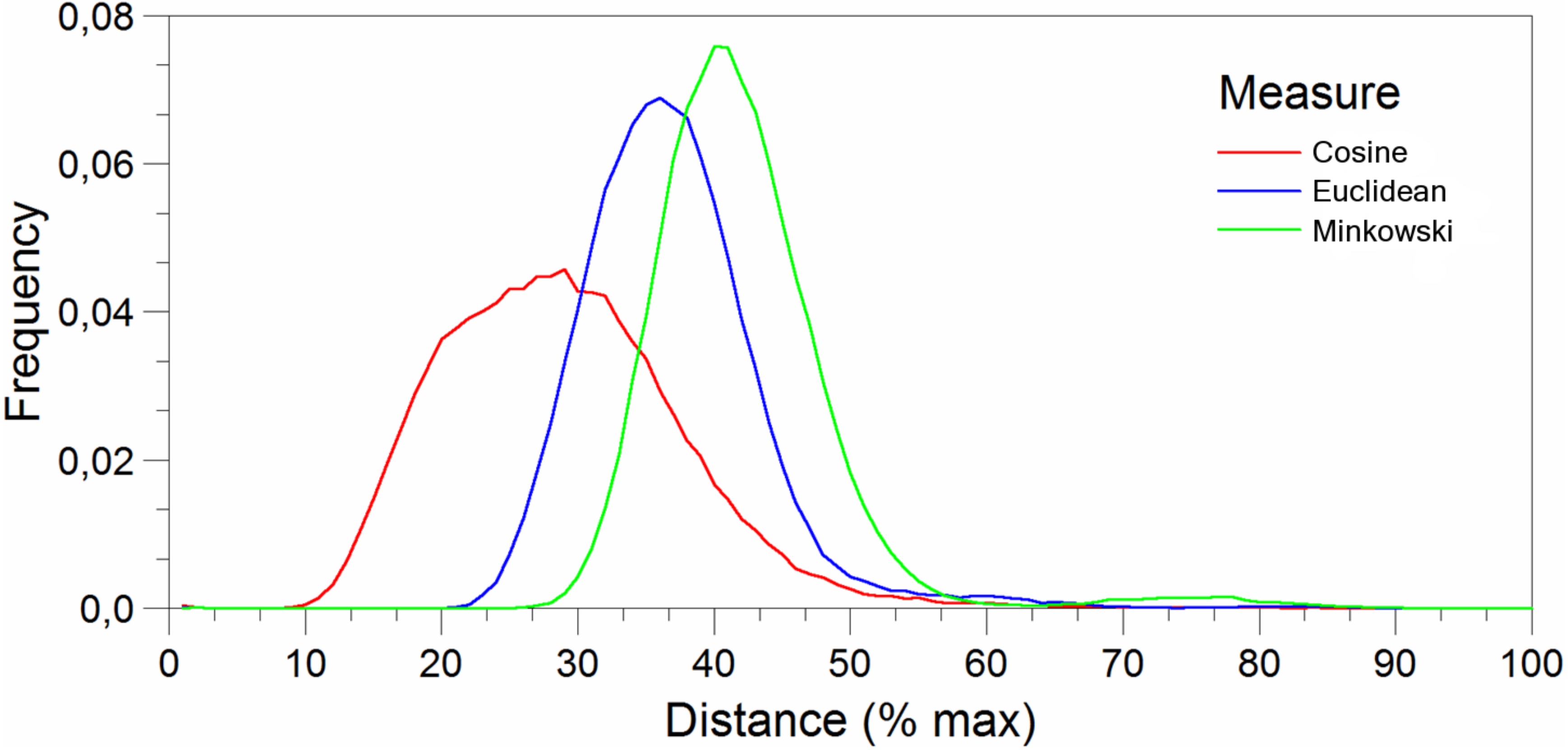
The use of effective distance functions has been explored for many data mining problems including clustering, nearest neighbor search, and indexing. Recent research results show that if the Pearson variation of the distance distribution converges to zero with increasing dimensionality, the distance function will become unstable (or meaningless) in high dimensional space even with the commonly used L_p metric on the Euclidean space. This result has spawned many subsequent studies. We first comment that although the prior work provided the sufficient condition for the unstability of a distance function, the corresponding proof has some defects. Also, the necessary condition for unstability (i.e., the negation of the sufficient condition for the stability) of a distance function, which is required for function design, remains unknown. Consequently, we first provide in this paper a general proof for the sufficient condition of unstability. More importantly, we go further to prove that the rapid degradation of Pearson variation for a distance distribution is in fact a necessary condition of the resulting unstability. With the result, we will then have the necessary and the sufficient conditions for unstability, which in turn imply the sufficient and necessary conditions for stability. This theoretical result derived leads to a powerful means to design a meaningful distance function. Explicitly, in light of our results, we design in this paper a meaningful distance function, called Shrinkage-Divergence Proximity (abbreviated as SDP), based on a given distance function. It is empirically shown that the SDP significantly outperforms

gards to the performance issue but also to the quality issue. Specifically, on the quality issue, the design of effective distance functions has been deemed a very important and challenging issue. Recent research results showed that in high dimensional space, the concept of distance (or proximity) may not even be qualitatively [1][2][3][5][6][11]. Explicitly, the theorem in [6] showed that under a broad set of conditions, in high dimensional space, the distance to the nearest data point approaches the distance to the farthest data point of a given query point with increasing dimensionality. For example, under the independent and identically distributed dimensions assumption, the commonly used L_p metrics will encounter problems in high dimensionality. This theorem has spawned many subsequent studies along the same line [1][2][3][11][13].

The scenario is shown in Figure 1 where ϵ denotes a very small number. From the query point, the ratio of the distance to the nearest neighbor to that to the farthest neighbor is almost 1. This phenomenon is called the unstable phenomenon [6] because there is poor discrimination between the nearest and farthest neighbors for proximity query. As such, the nearest neighbor problem becomes poorly defined. Moreover, most indexing structures will have a rapid degradation with increasing dimensionality which leads to an access to the entire database for any query [3]. Similar issues are encountered by distance-based clustering algorithms and classification algorithms to model the proximity for grouping

Distances & similarities

When a distance function is a good one?



Relative contrast depends on the distance algorithm chosen (here some example of pairwise distances distribution from CrystalFp). From: http://www.mariovalle.name/CrystalFp/uploads/CrystalFpLib/high-dim-spaces.pdf

Visualization issues

- Visualization cannot be **directly** used to understand high dimensional data, mainly because we can produce graphical images only in 2D and 3D.
- High-dimensional datasets are usually huge (to fight the sparseness of the high dimensional spaces), and a computer screen has a finite number of distinct pixels.
- On the converse, full automatic knowledge discovery approaches only work for well-defined and clearly specified problems
  - visualization can offer insights, even if its usage in high dimensional spaces is not intuitive or perceptually immediate.

Visualization issues

- Grasp partial idea of the high-dimensional space structure: visualiza it on 2D screen in a way that preserves at least approximately the distances between data points in the original space.
- Category of techniques known as multidimensional projection

- Later…

Dimension Reduction

- Approaches
  - Feature selection
  - Change the representation space
    - E.g., Principal Component Analysis

Dimension Reduction

- Common approach: Principal Component Analysis
  - Obtains new features (the principal components) as linear combinations of the original correlated) features/variables
  - The PCs are essentially (uncorrelated) linear combinations of the original variables, capturing most of the variance in the data (Jolliffe 2002)
  - the first PC explains the most variance, the second explains the second most variance, and so on, with each subsequent component explaining less

PCA for dimension reduction

- Goal is to find linearly independent dimensions (PCs) which can losslessly represent the data points.

- Number of PC = number of dimensions, but last PCs can often be discarded

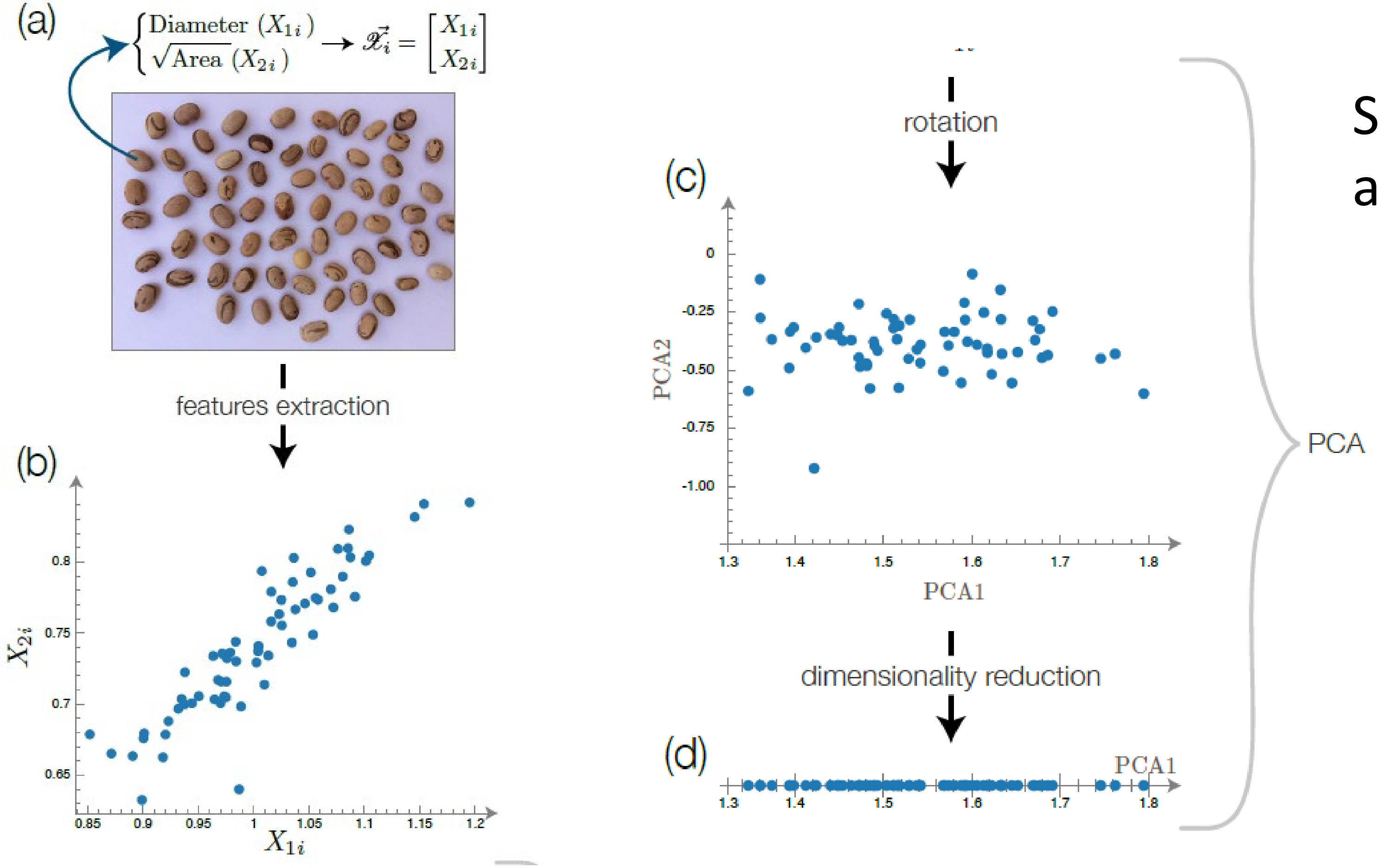
- Those newly found dimensions should allow us to predict/reconstruct the original dimensions. The reconstruction/projection error should be minimized.

PCA for dimension reduction

- particularly useful in processing data with many features, where multi-colinearity exists between the features

- beyond visualization, useful for denoising and data compression

How PCA works



Source: Gewers et al. 2018
arXiv 1804.02502v2

FIG. 1. PCA example on a real-world situation. Each bean (a) is characterized in terms of two measurements: diameter X_{1i} and square root of area X_{2i}. Though these two measurements are intrinsically related in a direct fashion, bean shape variations induce a dispersion of the objects when mapped into the features space (b). PCA allows the identification of the orientation of maximum data dispersion (c). As the dispersion in the resulting second axis is relatively small, this axis can be discarded (d).

How PCA works

- Calculate the covariance matrix *X* of data points
- Calculate eigenvectors and corresponding eigenvalues
- Sort the eigenvectors according to their eigenvalues in decreasing order
- Choose first *k* eigenvectors as the new *k* dimensions => these are the most important directions!
- Transform the original *d*-dimensional data points onto the new *k* dimensions

PCA explained@StaQuest

- https://www.youtube.com/watch?v=HMOI_lkzW08 ~5 min – how to interpret

- https://www.youtube.com/watch?v=FgakZw6K1QQ ~21 min – step by step

- https://www.youtube.com/watch?v=oRvgq966yZg ~8 min – practical tips

How PCA works

- Quite often, the dataset is *standardized* prior to PCA
  - advisable when the original variables have significantly different dispersions or scales, to avoid biasing the influence of certain variables
  - get misleading components if use data features of different scales

- If features are of different scales should use correlation matrix instead of covariance matrix

Notes

- Always normalize your data before doing PCA because if we use data features of different scales, you get misleading components

- If features are of different scales should use correlation matrix instead of covariance matrix

Data pre-processing tasks

- Checking for & handling missing values
- Checking for & handling categorical data (data mining)
- Verify distribution of variables - check for anomalies and outliers
- Feature scaling (normalization, standardization)
- Assessing data (dis)similarity
- Assess correlation between variables
- Dimension reduction (e.g. PCA transformation)
- **Data sub-sampling**
- Data aggregation
- **Data interpolation**
- Data splitting (e.g. for data mining model learning)

Sources/material

T. Munzner, Visualization Analysis & Design

Information Visualization Fundamentals, Enrico Bertini, online course in Coursera

M. Valle – a look at high-dimensional spaces
http://www.cscs.ch/~mvalle

Principal component analysis: a natural approach to data exploration. Gewers et al. 2018