Unsupervised Machine Learning

Types of machine learning

So far we have assumed an input space X and an output space Y and that there is some relation between X and Y

Two best known ML types: Supervised $\times$ unsupervised

Supervised: observations are of the form $(\mathbf{x}, y)$

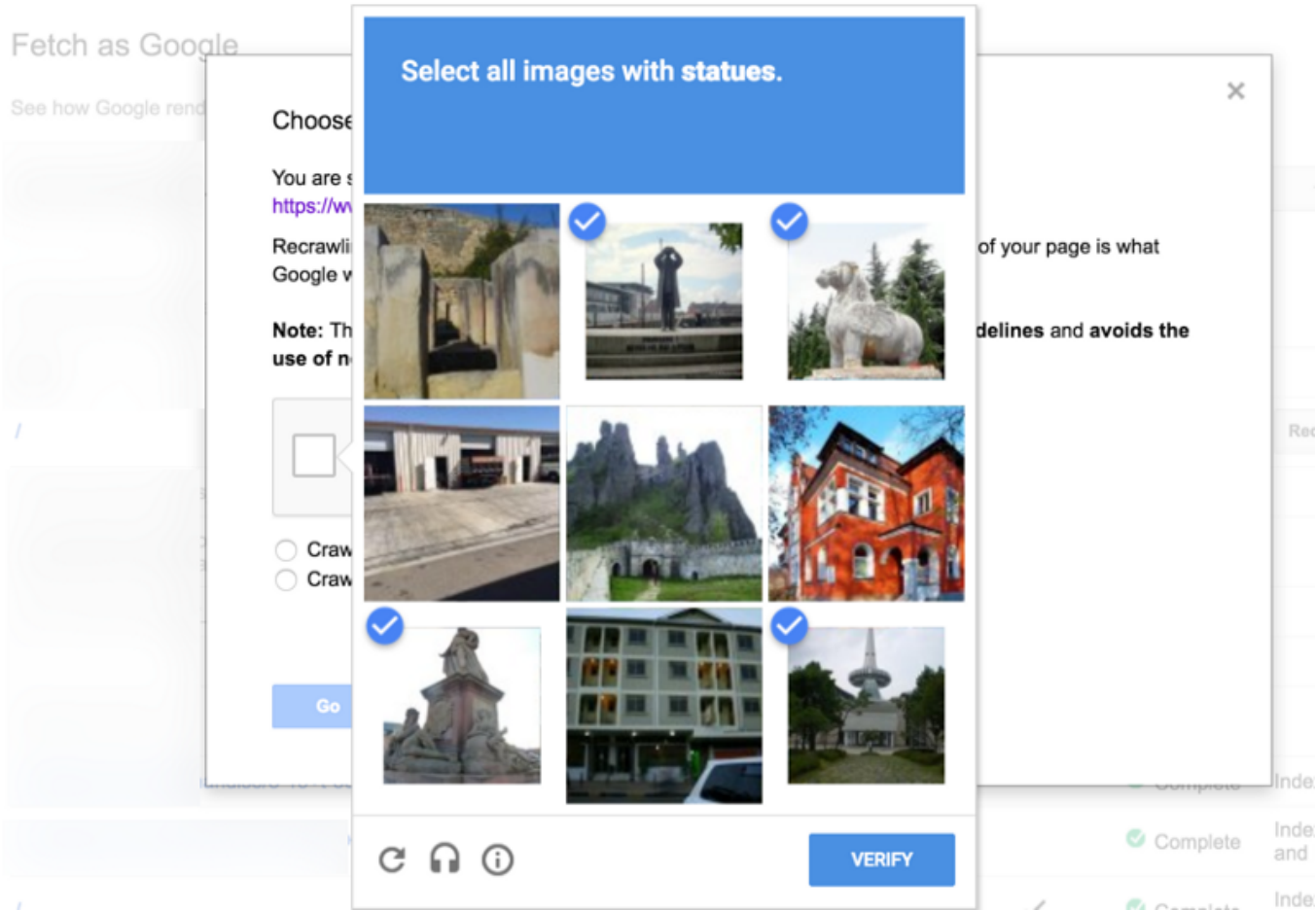
Data labeling is an expensive task

Unsupervised: observations are of the form $\mathbf{x}$

no target to guide us

Strategies for data labeling

Captcha



Strategies for data labeling

Mechanical Turks

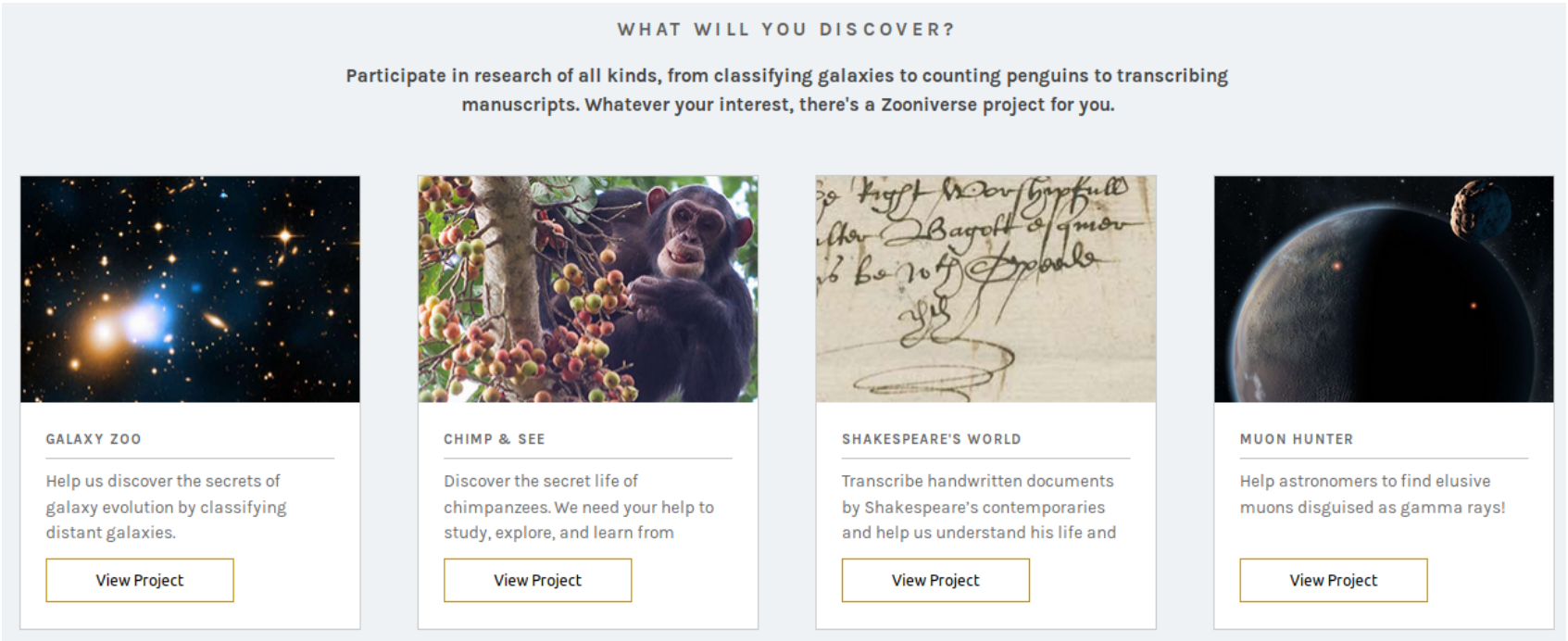
How it works

MTurk offers developers access to a diverse, on-demand workforce through a flexible user interface or direct integration with a simple API. Organizations can harness the power of crowdsourcing via MTurk for a range of use cases, such as microwork, human insights, and machine learning development.

https://www.mturk.com/

Strategies for data labeling

Crowdsourcing



https://www.zooniverse.org/

Unsupervised learning

What can we do in the absence of target labels ?

Group data based on similarity $\Longrightarrow$ **clustering**

Simplify/reduce data $\Longrightarrow$ **dimensionality reduction**

Visualize data $\Longrightarrow$ **data visualization**

Clustering

Grouping items into clusters
based on some similarity criteria

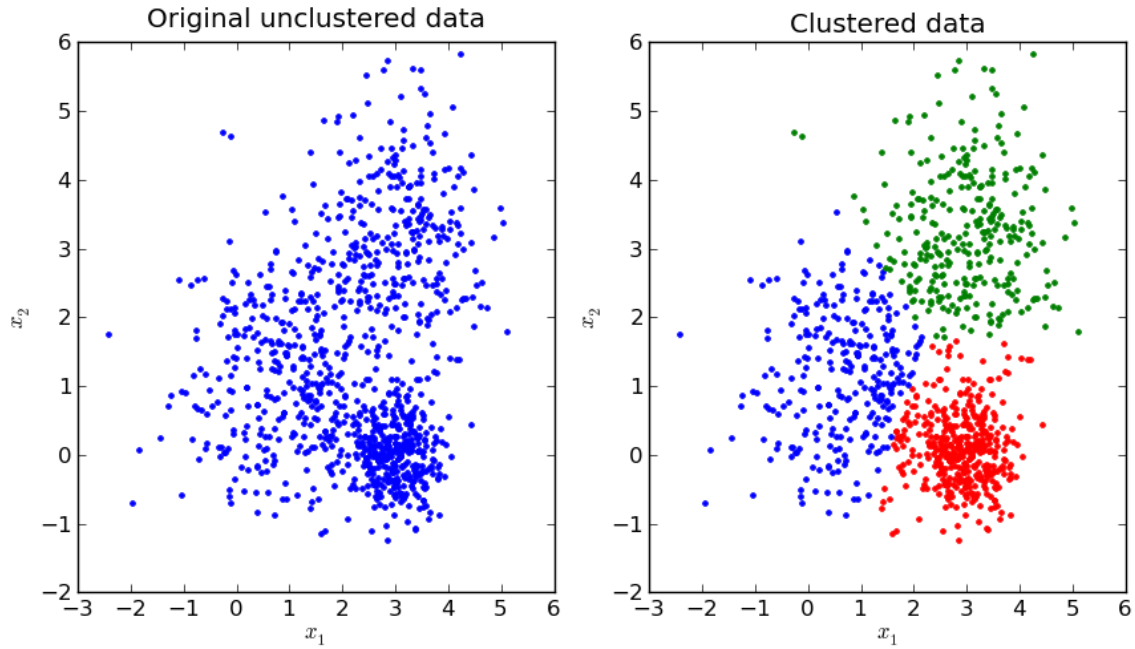
Standard clustering

Given: $D = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ (items without label)

Consists in partitioning D into c disjoint clusters $\{C_1, C_2, \ldots, C_c\}$

c : number of clusters (usually unknown)

- $C_j \neq \emptyset, \quad j = 1, \ldots, c$
- $\bigcup_{j=1}^{c} C_j = D$
- $C_i \cap C_j = \emptyset, \quad i, j = 1, 2, \ldots, c, \; i \neq j$



Source: https://i.stack.imgur.com/cIDB3.png

What criterion we should use to build the clusters ?

Principle: An ideal clustering is one in which items within a group are highly similar each other, while items in distinct groups are highly dissimilar

Similarity must be adequately characterized (and it may depend on the application)

Clustering can be modeled as as problem of finding an optimal partition of D based on some cost function

Given N items, how many bipartitions are possible ? $2^{N-1} - 1$

How many partitions are possible (number of Stirling):

$$\frac{1}{c!} \sum_{j=0}^{c} (-1)^{c-j} \binom{c}{j} j^N \sim \frac{c^N}{c!}$$

Impossible to build every one and compute their costs!

Approaches for clustering

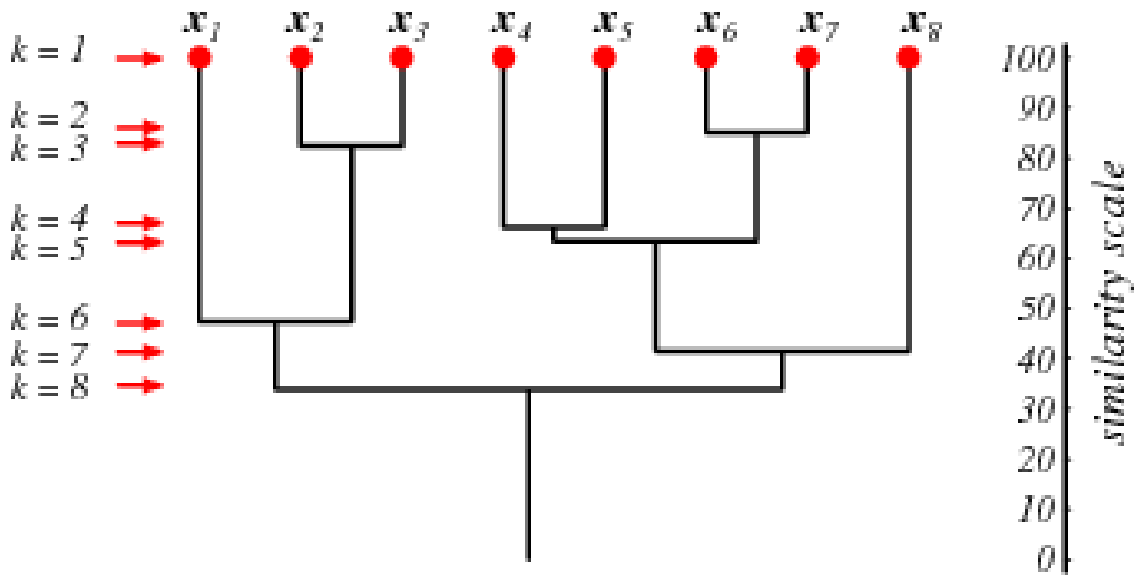
Hierarchical: either agglomerate or divisive techniques

Iterative: clusters are iteratively changed, based on some cost function optimization

Hierarchical Clustering

Agglomerative hierarchical clustering

Group items/clusters successively based on some similarity criterion (k: denotes the grouping step)



This tree-like structure is called *dendrogram*

We need to define similarity measures:

- between two items: $s(\mathbf{x}_i, \mathbf{x}_j)$
- between an item and a group: $s(\mathbf{x}_i, \mathbf{x}_j)$
- between two groups: $s(C_i, C_j)$

Some similarity measures:

http://www.scholarpedia.org/article/Similarity_measures

Similarity measures can be based on distance measures

Algorithm: Hierarchical agglomerative clustering

Initial partition: $\mathcal{C}_0 = \{C_i = \{\mathbf{x}_i\}, \quad i = 1, 2, \ldots, N\}$

$k = 0$

While $|\mathcal{C}_k| > 1$

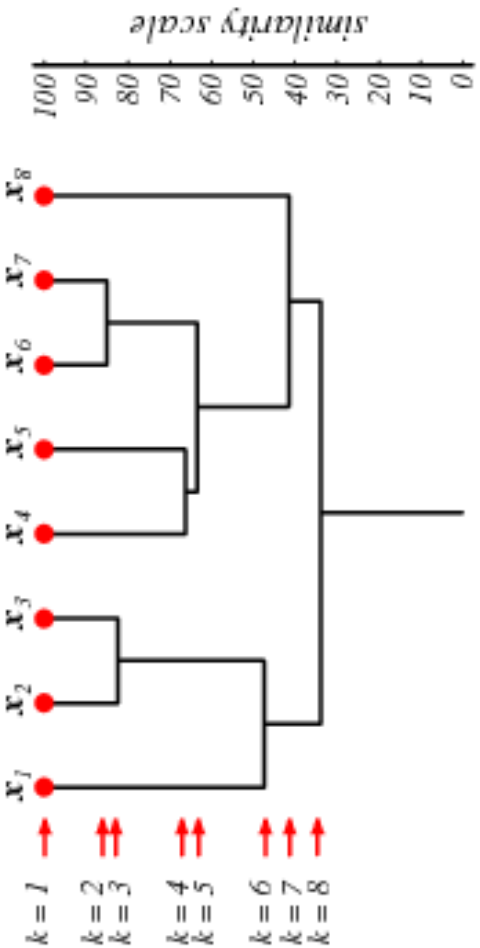
Among all pairs of clusters in $\mathcal{C}_k$, choose one that is most similar each other. Let (C_i, C_j) be such pair.

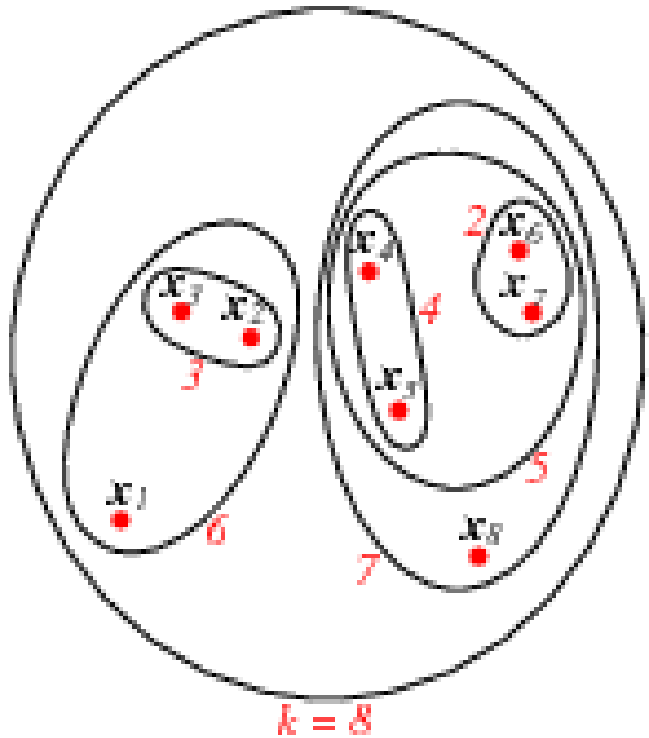
$C_q = C_i \cup C_j$

$\mathcal{C}_{k+1} = (\mathcal{C}_k \setminus \{C_i, C_j\}) \cup \{C_q\}$

$k = k + 1$

Cluster nesting

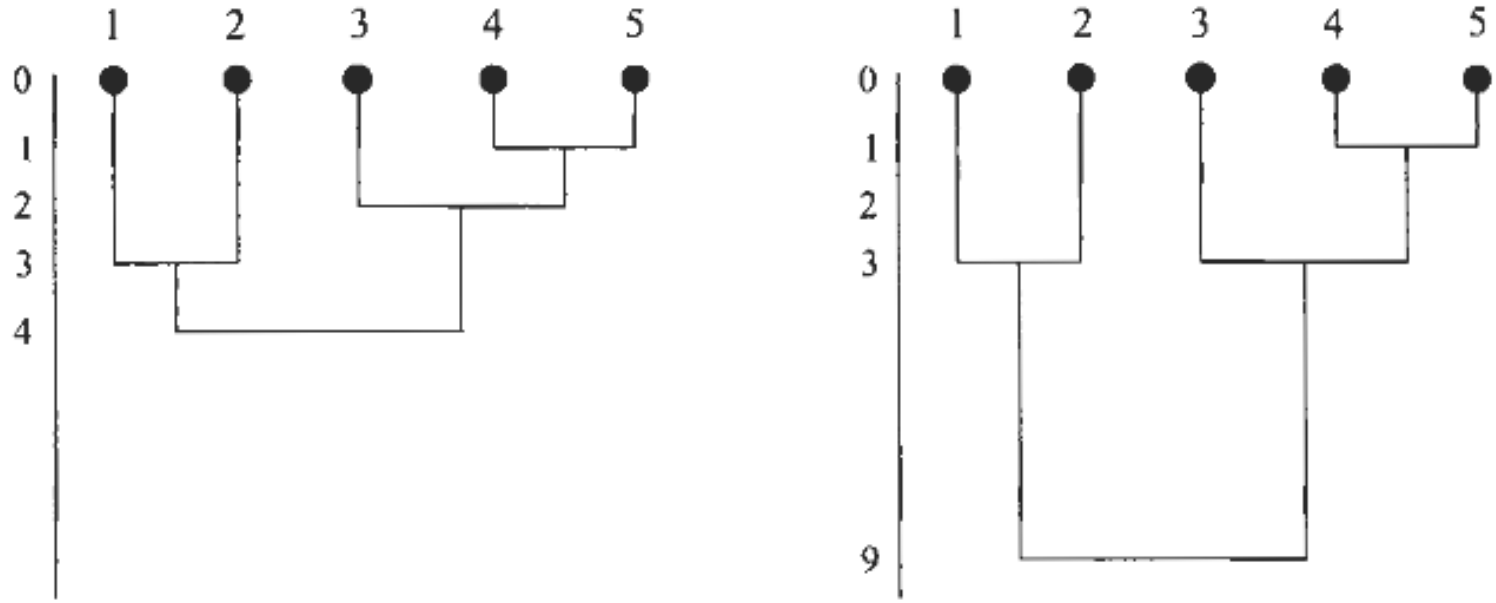




k: step of the grouping

How to define the final clusters ?

Lifetime of a cluster: difference between the similarity measure when it was created and when it was merged with another cluster



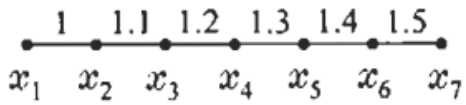
Subjective

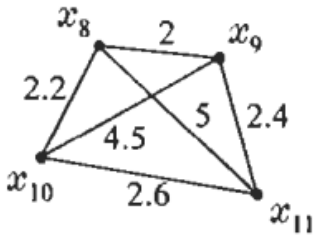
Visualization becomes problematic if there are too many items

Some characteristics of agglomerative clustering:

- easy to be understood
- requires large memory space
- not possible to undo previous grouping
- sensible to noise
- other similarity metrics can be used to group clusters
- resulting dendrogram depends on the metrics

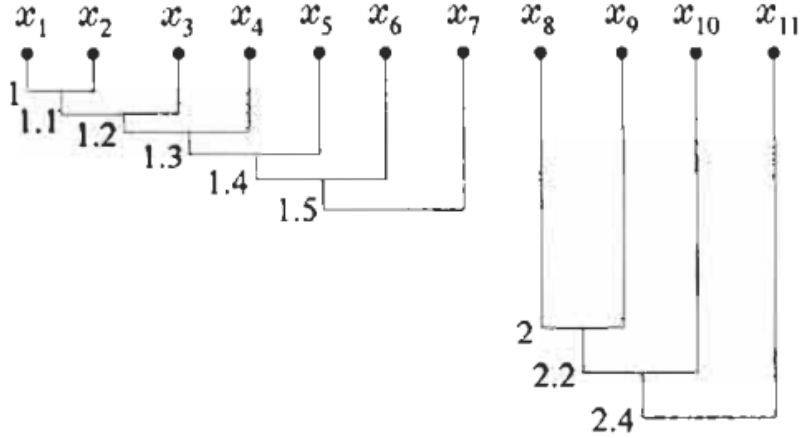
items and distances





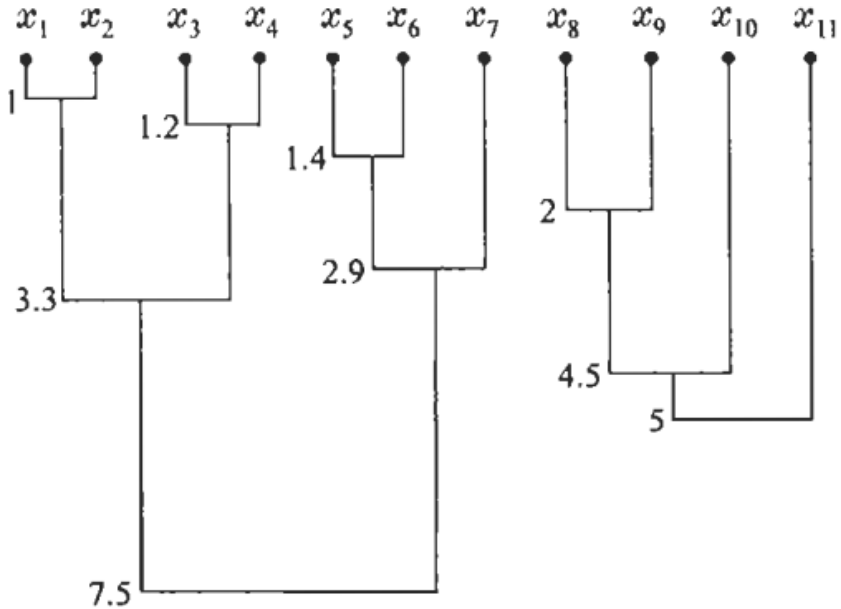
(a)

Single linkage



(b)

Complete linkage



Clustering

Iterative algorithms

(Best known: k-**means**)

Iterative algorithm mechanics

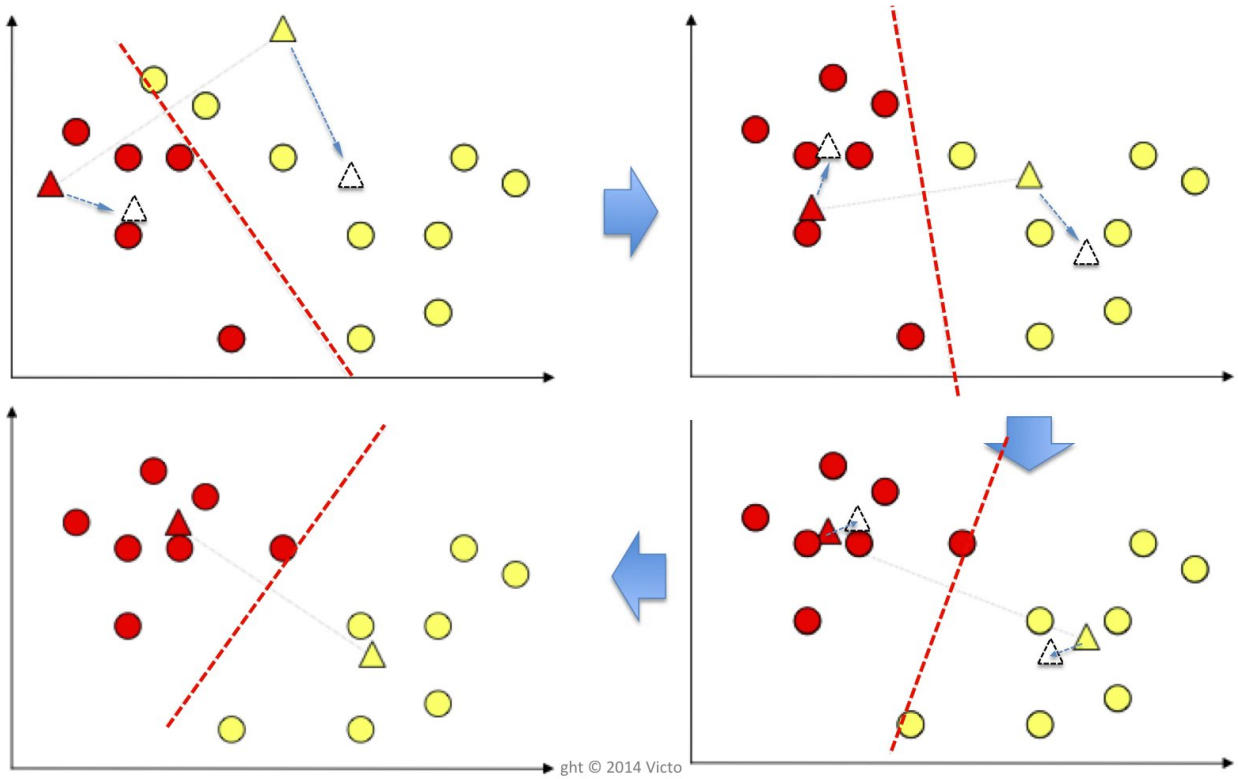
- Start the process with an arbitrary partition with K clusters
- Repeat until a stop criterion is met
  - update the partition
  - check if the updated partition has better cost and, if so, replace the previous partition with this one
- Return the current partition (the best obtained so far)

K-means algorithm

k, number of clusters, must be chosen a priori

1. Choose k items. They will be considered as the centroids (mean points) of the k initial clusters
2. Assign each item in D to the cluster of the closest centroid
3. Recompute the centroids, based on the assignment results
4. Repeat steps 2 and 3 until the centroids do not change or until some stop criterion is reached

K-means clustering example



Clustering

Random comments

Fuzzy clustering

Membership function:

$$u_j : D \to [0, 1], \quad j = 1, 2, \ldots, c$$

The sum must amount to 1:

$$\sum_{j=1}^{c} u_j(\mathbf{x}_i) = 1, \qquad i = 1, 2, \ldots, N$$

At least one item with non-null membership score in each class j:

$$0 < \sum_{i=1}^{N} u_j(\mathbf{x}_i) < N, \qquad j = 1, \ldots, c$$

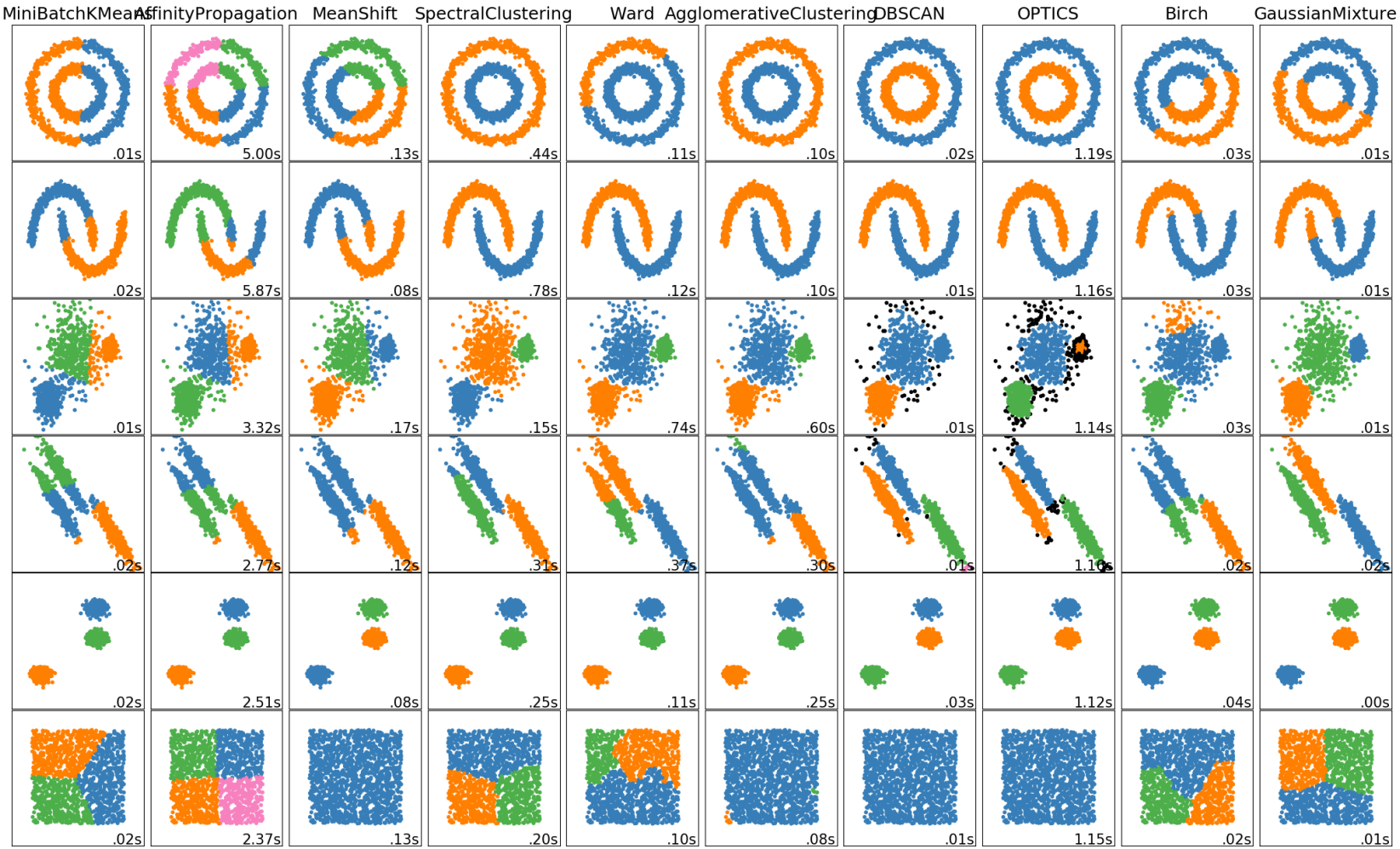
Standard clustering: for every $\mathbf{x} \in D$, there exists j such that $u_j(\mathbf{x}) = 1$ and $u_k(\mathbf{x}) = 0$ for all $k \neq j$

Clustering para grande volume de dados

Aplicação em Data Mining impulsionou a proliferação de algoritmos, principalmente voltados para o tratamento de grande volume de dados.

Alguns algoritmos existentes (circa 2013):

> Hierárquico, CURE, ROCK, Chameleon, k-means, k-medoids, Fuzzy, PAM, CLARA, CLARANS, SOM, DBSCAN, DENCLUE, CLIQUE, etc, etc



Source https:/https://scikit-learn.org/stable/modules/clustering.html

Validação de clusterings

- Rodar um algoritmo várias vezes, usando diferentes parâmetros
- Rodar diferentes algoritmos de clustering
- Confrontar com conhecimentos a priori (principalmente de especialistas da área)

Comparação de clusterings

Índices (valores entre 0 e 1) que indicam quão distintos ou similares são duas partições:

- Pureza
- Rand index
- Informação mútua
- etc

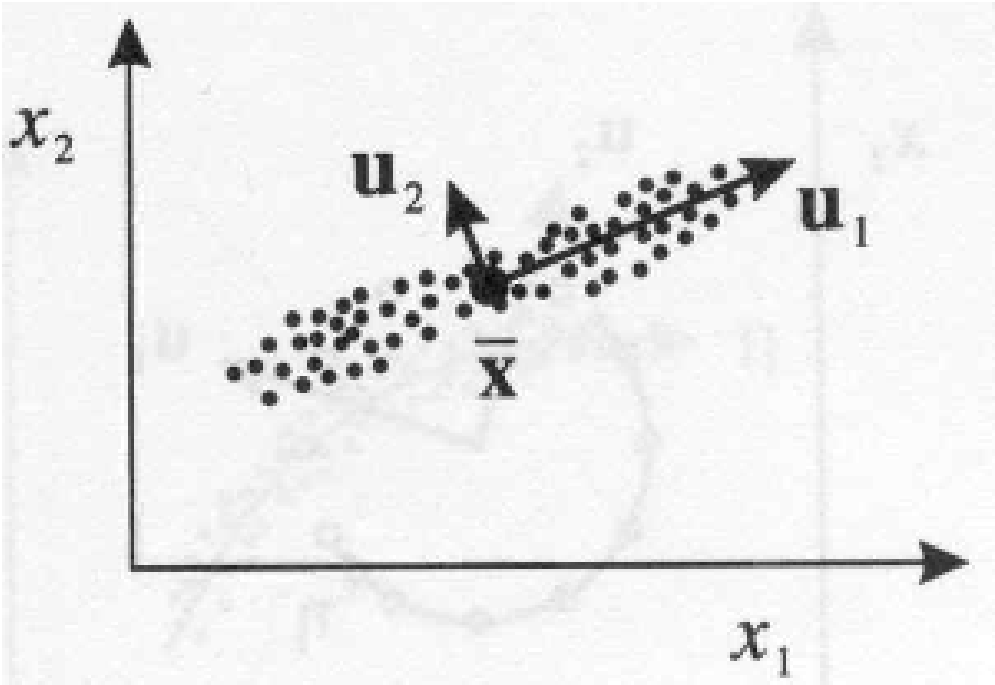
Referências bibliográficas:

- Seção 25.1 do livro do Kevin P. Murphy (Machine Learning: a Probabilistic Perspective)
- Davies, David L.; Bouldin, Donald W., "A Cluster Separation Measure," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.PAMI-1, no.2, pp.224,227, April 1979 (doi: 10.1109/TPAMI.1979.4766909)
- Marina Meila, Comparing clusterings – an information based distance, Journal of Multivariate Analysis, Volume 98, Issue 5, May 2007, Pages 873-895, http://dx.doi.org/10.1016/j.jmva.2006.11.013.

Data simplification / reduction

PCA – Principal component analysis

PCA – Intuition



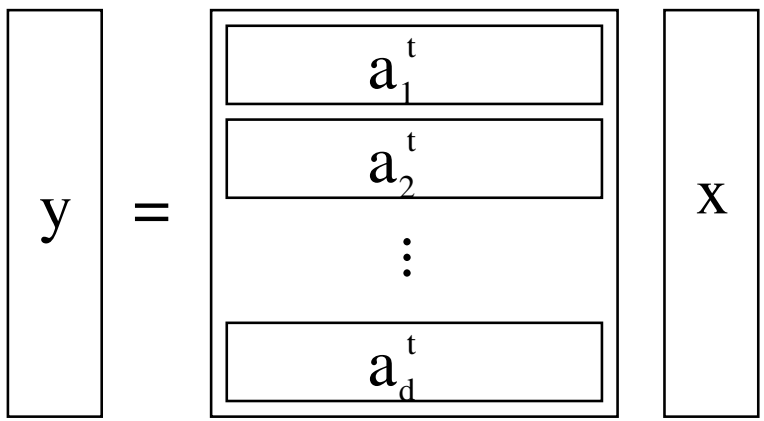
Change the basis, to align the axis to the directions with largest dispersion of the points.

PCA

The goal is to find a linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$ that

(1) maximizes the variance of the transformed variables $y_i = \mathbf{a}_i^T \mathbf{x}$
(y_i can be seen as a projection of $\mathbf{x}$ on $\mathbf{a}_i$)

(2) any two vectors $\mathbf{a}_i$ and $\mathbf{a}_j$ are orthogonal each other



The desired linear transformation can be obtained from the eingenvectors of the covariance matrix of D

Covariance matrix

Example for $d = 3$:

$$S = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{12} & s_{22} & s_{23} \\ s_{13} & s_{23} & s_{33} \end{bmatrix}$$

Diagonals: variance of x_1, x_2, and x_3

Remaining elements: covariance

$s_{ij} = s_{ji} = \frac{1}{N-1} \sum_{n=1}^{N} (x_i^{(n)} - \overline{x}_i)(x_j^{(n)} - \overline{x}_j)$

S : covariance matrix of dataset D

The covariance matrix S is symmetric and positive semi-definite ($\mathbf{x}^t S \mathbf{x} > 0, \forall \mathbf{x} \neq \mathbf{0}$)

Let $\lambda_1 > \lambda_2 > \ldots > \lambda_d > 0$ be the eingenvalues of S (they exist and are all real, non-negative and distinct since S is a symmetric positive definite matrix)

Let $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_d}$ be the unit eigenvectors of S corresponding to the eingenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_d$

Let M be the matrix whose columns are the eingenvectors $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_d}$ of S

S is diagonalizable : $\Lambda = M^{-1}SM$

If we set $\mathbf{y} = M^{-1}\mathbf{x}$, we have:

- $y_i = \mathbf{e}_i^t\mathbf{x}$ is the i-th **principal component**
- $Var(y_i) = \lambda_i$
- $Cov(y_i, y_j) = 0, \forall j < i$

PCA keeps the total variance

$$\sum_{i=1}^{d} Var(x_i) = s_{11} + s_{22} + \cdots + s_{dd} \overset{(1)}{=} tr(S)$$

$$\overset{(2)}{=} tr(M\Lambda M^{-1}) \overset{(3)}{=} tr(M\Lambda M^{T}) \overset{(4)}{=} tr(\Lambda M^{T} M)$$

$$\overset{(5)}{=} tr(\Lambda M^{-1} M) \overset{(6)}{=} \lambda_1 + \lambda_2 + \cdots + \lambda_d \overset{(7)}{=} \sum_{i=1}^{d} Var(y_i)$$

(1) s_{ii} is the variance of x_i; variances are in the diagonal of S

(2) $\Lambda = M^{-1} S M \implies S = M\Lambda M^{-1}$

(3,5) M is orthogonal; hence $M^{-1} = M^{T}$

(4) $tr(AB) = tr(BA)$

(6) the diagonal of Λ contains the eigenvalues of S

(7) property (previous page)

Interpretation

Total variance is equal to the sum of the eigenvalues

Thus, the **percentage of total variance explained by the k-th principal component** is

$$\frac{\lambda_k}{\lambda_1 + \lambda_2 + \cdots + \lambda_d}$$

Dimensionality reduction using PCA

How many components one should choose?

Choose the d' principal components such that

$$\frac{\sum_{i=1}^{d'} \lambda_i}{\sum_{i=1}^{d} \lambda_i} > T$$

In general, $T = 0.90$ or $T = 0.95$

In many cases, a few components is sufficient to explain most of the total variance.

What is the meaning of the selected components ?

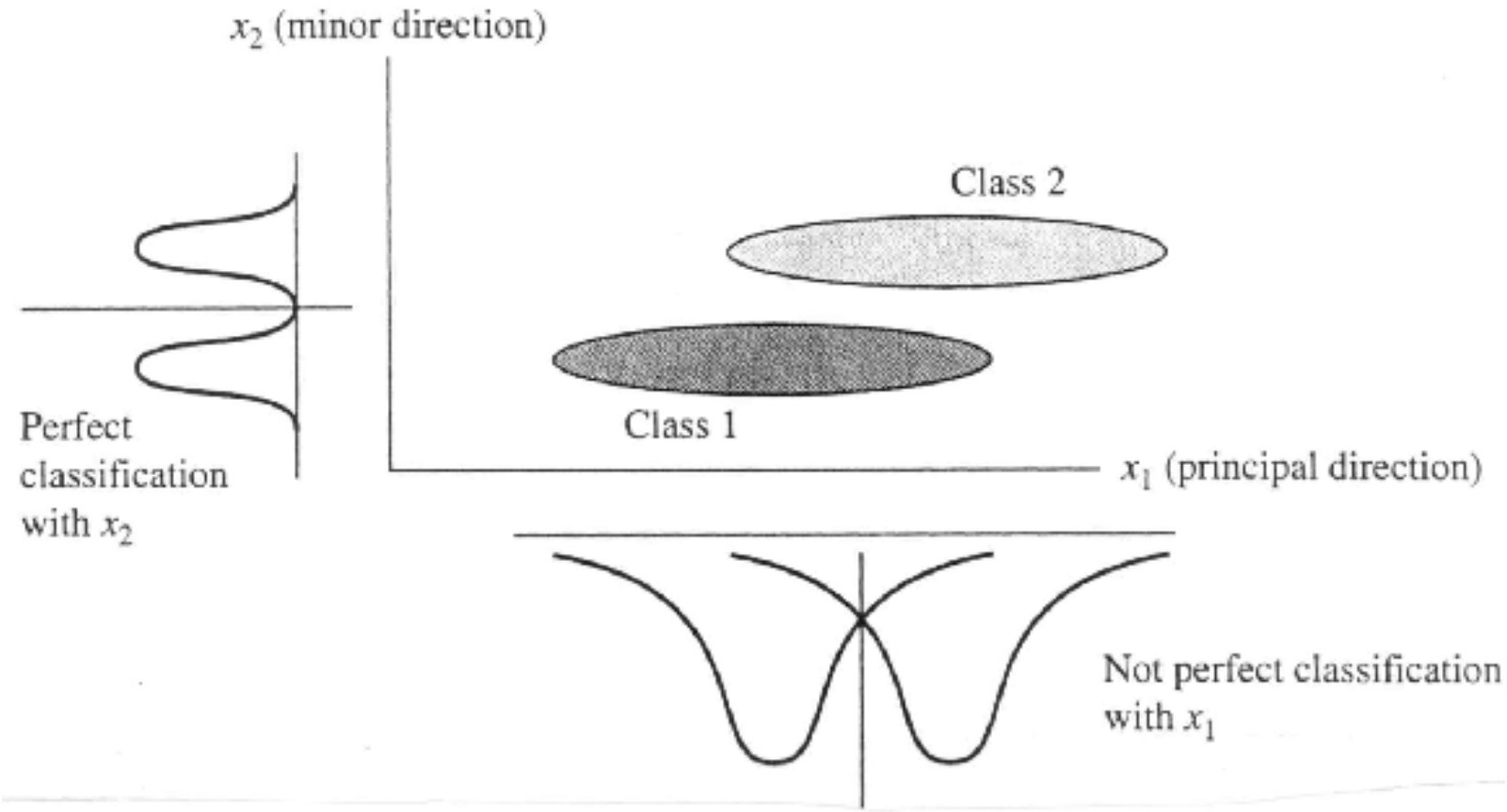
How do they relate to the original variables ?

We can examine the coefficients of the original variables in the projection $y_i = \mathbf{e}_i^t \mathbf{x}$

The magnitude of the Coefficients indicate the relevance or how each original variable contributes to the selected components.

Dimensionality reduction using PCA

In classification problems, not necessarily good



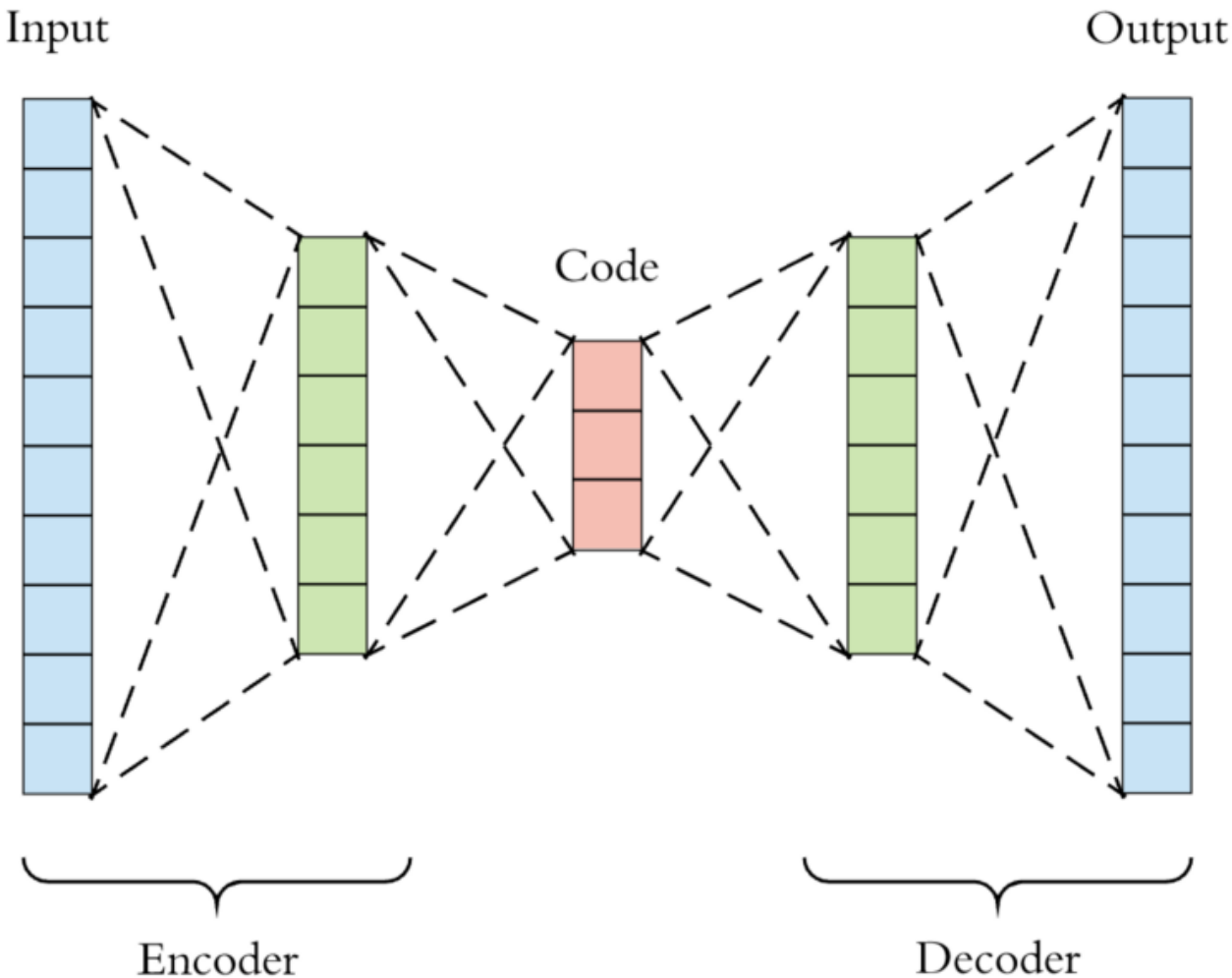
Some references for PCA

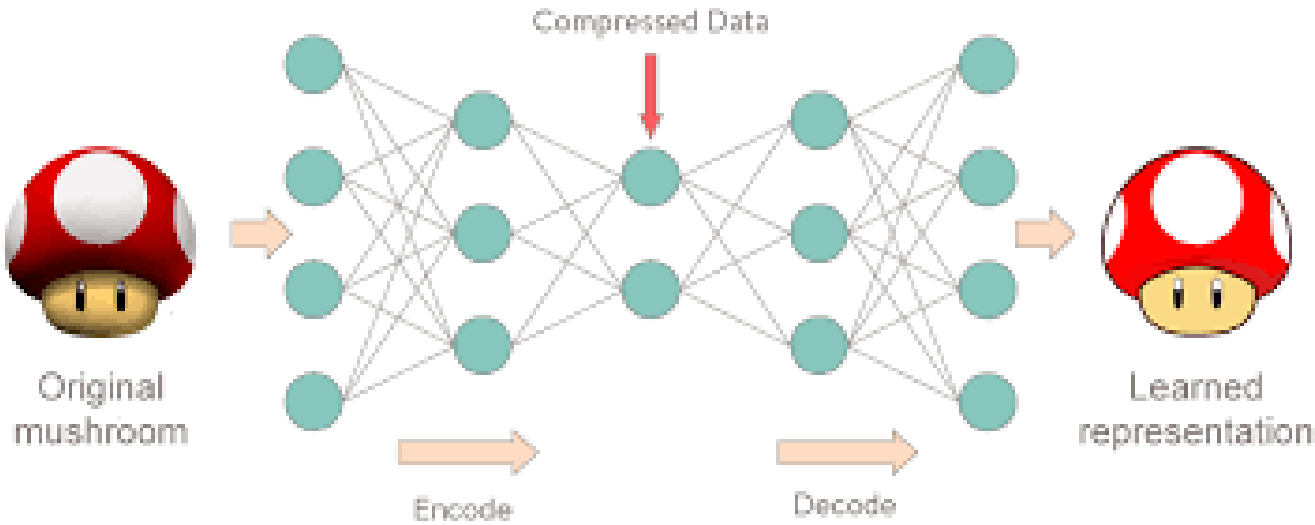
- Richard A. Johnson and Dean W. Wichern, *Applied Multivariate Statistical Analysis*, Prentice Hall
- Jon Shlens, *Tutorial on Principal Component Analysis*, December 2005.
- Lindsay I. Smith, *A tutorial on Principal Components Analysis*, February 2002 (some application examples in its final part)

Data simplification / reduction

Auto-encoders

Auto-encoders




Compressed Data
Original mushroom
Encode
Decode
Learned representation

Data simplification / reduction

Data visualization

SOM - Self organizing maps

(Kohonen maps)

SOM: Idea

Idea: to map points from a high dimensional space to a low dimension space, keeping the proximity relation among items

- The new space is usually a 2D map with a discrete grid of nodes
- Each node of the map has a coordinate and a weight vector of dimension d assigned to it
- **OBS.**: it is frequently referred to as a type of neural network, but it is not similar to the networks we have seen previously

Self-organizing maps (SOM)

Dados originais: exemplos $\mathbf{x} \in \mathbb{R}^d$

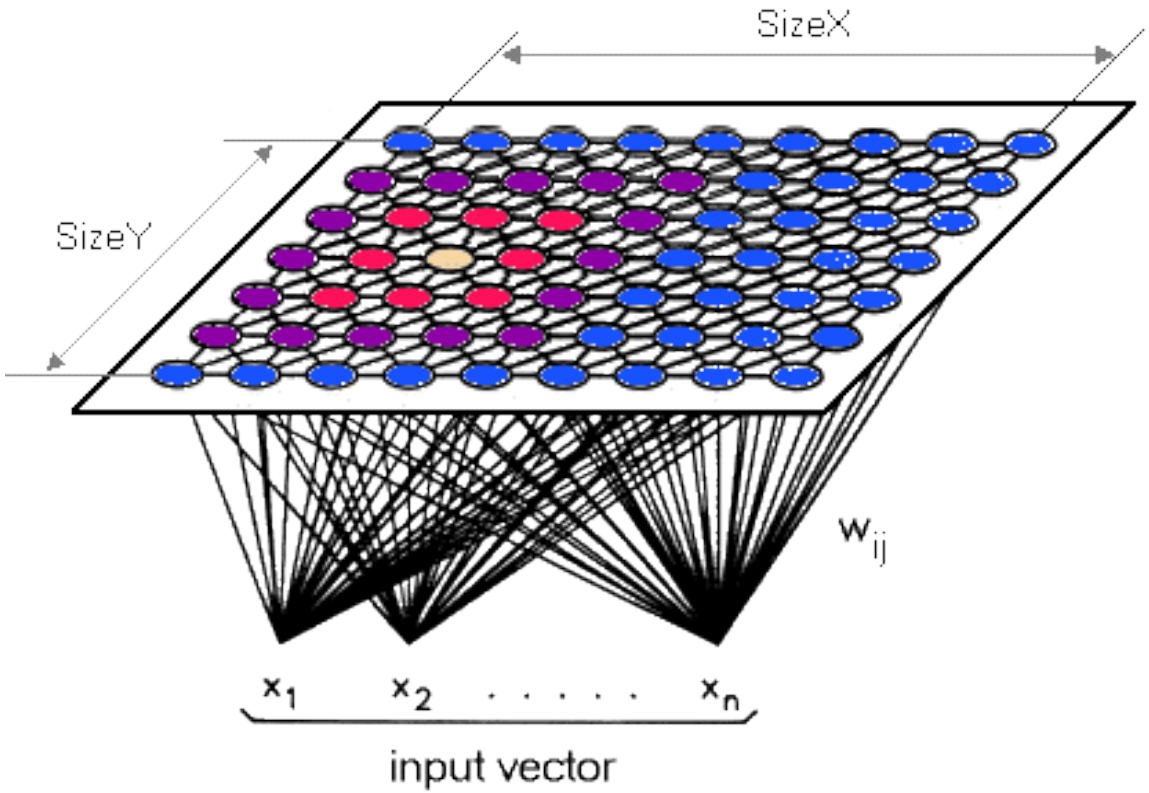
O mapa no qual esses pontos serão "projetados" está em uma grade no plano 2D

A cada nó $\mathbf{p}_k$ do mapa 2D está associado um vetor de pesos $\mathbf{w}_k \in \mathbb{R}^d$

Pode-se calcular a similaridade entre $\mathbf{x}$ e $\mathbf{w}_k$

Os exemplos $\mathbf{x}$ são fixos, mas os pesos $\mathbf{w}_k$ podem ser alterados iterativamente

Arquitetura de uma SOM



BMU (best matching unit) (nó amarelo): dado $\mathbf{x}$, nó $\mathbf{p}^*$ no mapa que possui o vetor de pesos mais próximo (similar a) $\mathbf{x}$

Vizinhos do BMU (nós vermelhos e roxos): $V(\mathbf{p}^*)$, definidos por um kernel (Lembram-se de Parzen window ?)

Exemplos de kernel unidimensional

- Retangular

$$\phi(u) = \begin{cases} \frac{1}{2}, & \text{se } |u| < 1, \\ 0, & \text{c.c.} \end{cases}$$

- Triangular

$$\phi(u) = \begin{cases} 1 - |u|, & \text{se } |u| < 1, \\ 0, & \text{c.c.} \end{cases}$$

- Biweight

$$\phi(u) = \begin{cases} \frac{15}{16}(1 - u^2)^2, & \text{se } |u| < 1, \\ 0, & \text{c.c.} \end{cases}$$

- Normal (um dos mais usados)

$$\phi(u) = \frac{1}{\sqrt{2\pi}} e^{\frac{-u^2}{2}}$$

- Bartlett-Epanechnenko

$$\begin{cases} \frac{3}{4}\left(1 - \frac{u^2}{\sqrt{5}}\right), & \text{se } |u| < \sqrt{5}, \\ 0, & \text{c.c.} \end{cases}$$

Algoritmo SOM

$t = 0$
Inicializar $\mathbf{w}(0)$, $\forall \mathbf{p}$ no mapa
Repetir
    Para cada $\mathbf{x} \in D$
        Seja $\mathbf{p}^*$ a BMU
        Para cada $\mathbf{p} \in \{\mathbf{p}^*\} \cup V(\mathbf{p}^*)$
            Seja $\mathbf{w}$ o vetor de pesos de $\mathbf{p}$
            $\mathbf{w}(t+1) = \mathbf{w}(t) + \eta(t)\phi(\mathbf{p}^* - \mathbf{p})(\mathbf{x} - \mathbf{w}(t))$
até convergir

ϕ é a window function (ou kernel function) — vale 1 no centro e diminui à medida que se afasta do centro

Isto é, o $\mathbf{w}$ do BMU e de seus vizinhos ficam "mais parecidos" com $\mathbf{x}$

Suponha que D é formado por 3 clusters bem separados

Qual aspecto terá o mapa?

Suponha que D é formado por 3 clusters bem separados

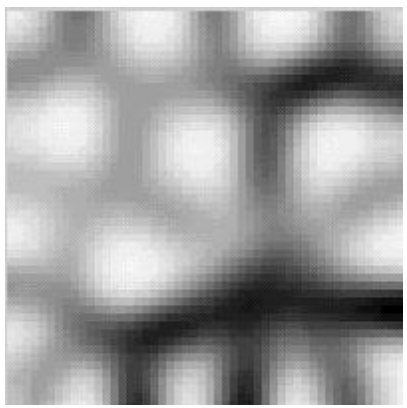
Qual aspecto terá o mapa?

Os BMUs de exemplos que estão próximos uns aos outros em $\mathbb{R}^D$ tendem a estar próximos no mapa

Que tipo de informação podemos extrair do mapa ?

SOM – Visualização

U-matrix (unified distance matrix): para cada nó do mapa calcula-se uma distância aos nós vizinhos (por exemplo, distância média)

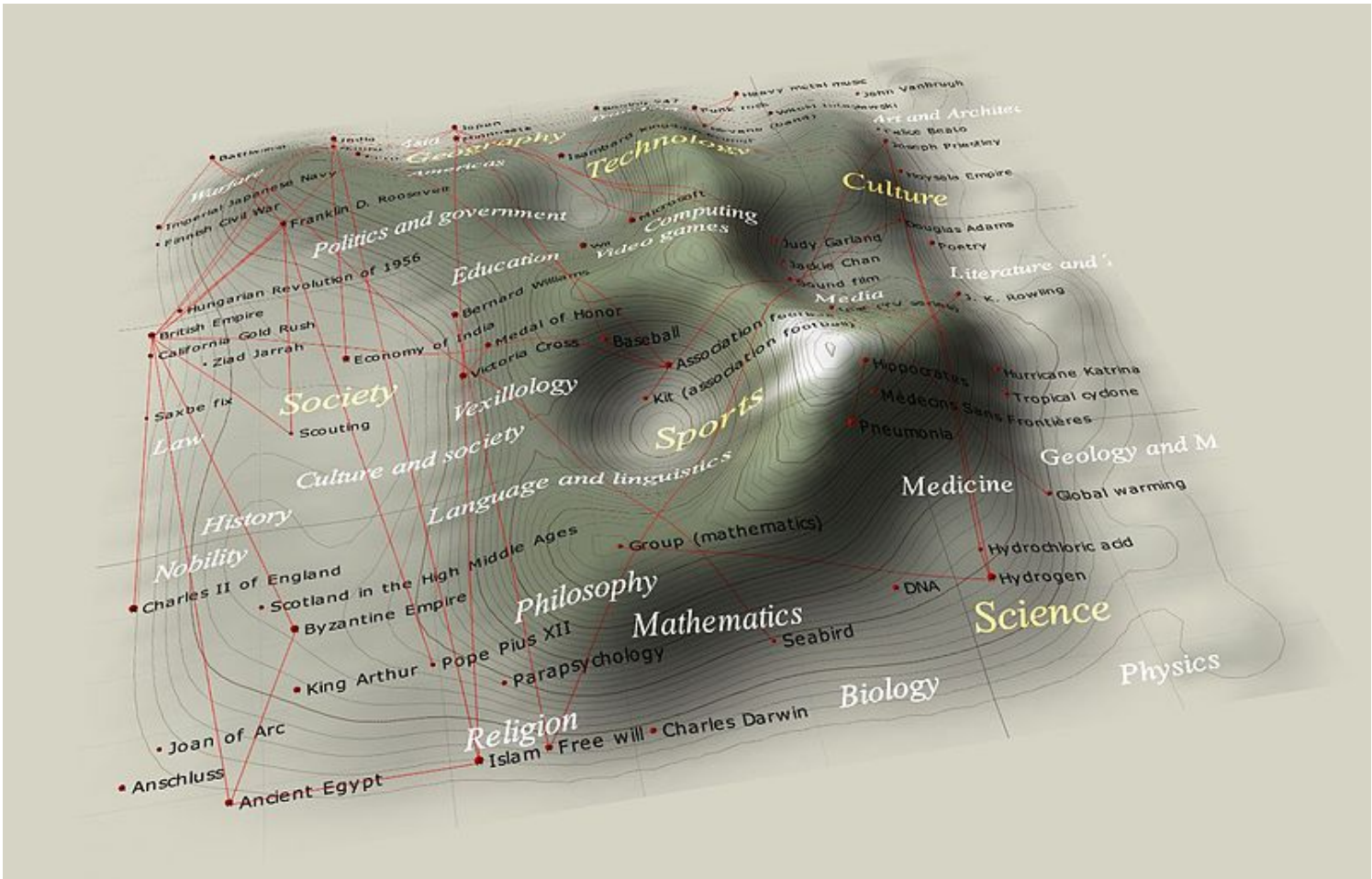


Cor dos nós no mapa: intensidade de cinza representa a similaridade aos nós vizinhos (no exemplo ao lado, quanto mais escura, maior a distância)

Regiões escuras correspondem a descontinuidades; regiões claras correspondem a nós com pesos similares.

Cada região clara pode ser interpretada como um grupo de dados similares no espaço original

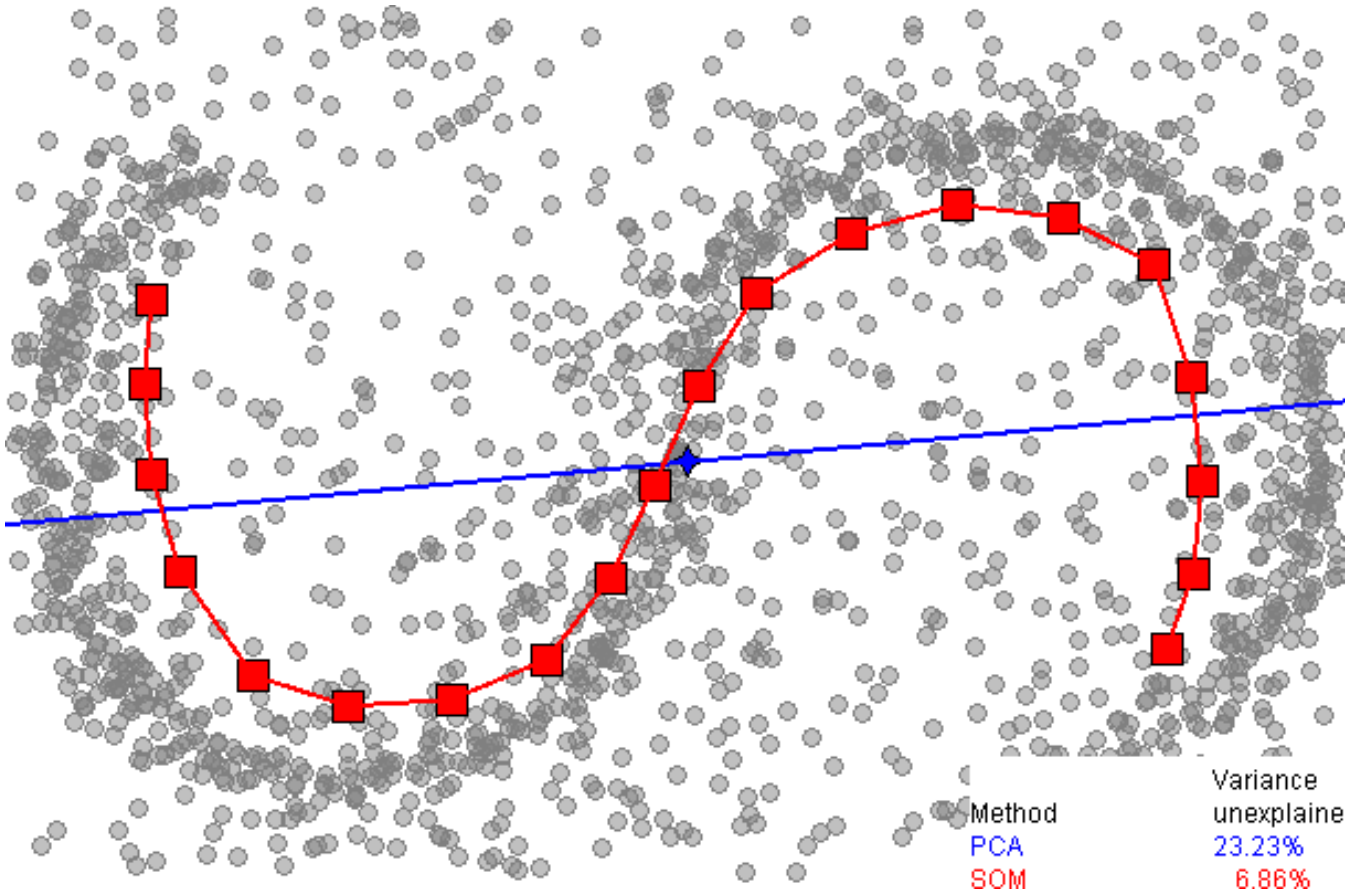
SOM – interpretação



Dados: artigos da wikipedia, representados pela frequência de palavras; as linhas vermelhas indicam links entre os artigos

SOM – interpretação

Aproximação discreta da distribuição no $\mathbb{R}^d$: cada vetor de pesos do mapa "aponta" para uma região do $\mathbb{R}^d$

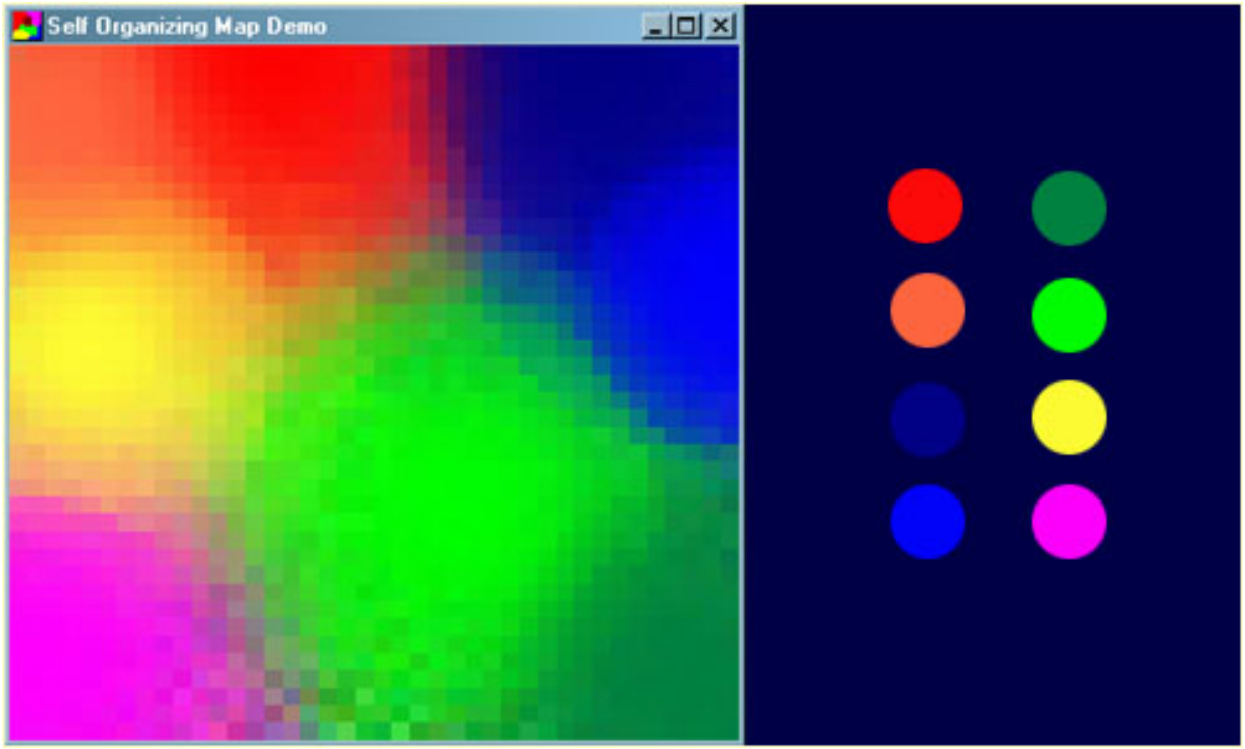


SOM unidimensional (20 nós);

Azul: componente principal Vermelho: os vetores de peso do SOM

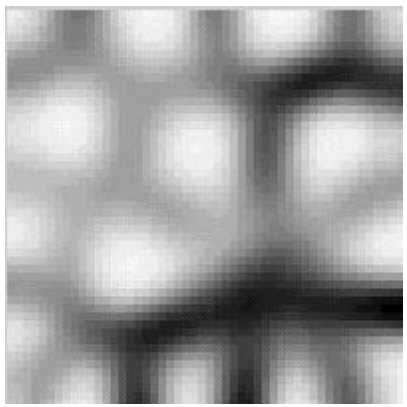
SOM – exemplo

Exemplo: representar cores RGB ($d = 3$) em um plano 2D



À direita: as cores RGB como pontos em $\mathbb{R}^3$. À esquerda: mapa SOM resultante. As cores do mapa são obtidas do próprio vetor de pesos (interpretadas como RGB).

SOM – processamento do mapa



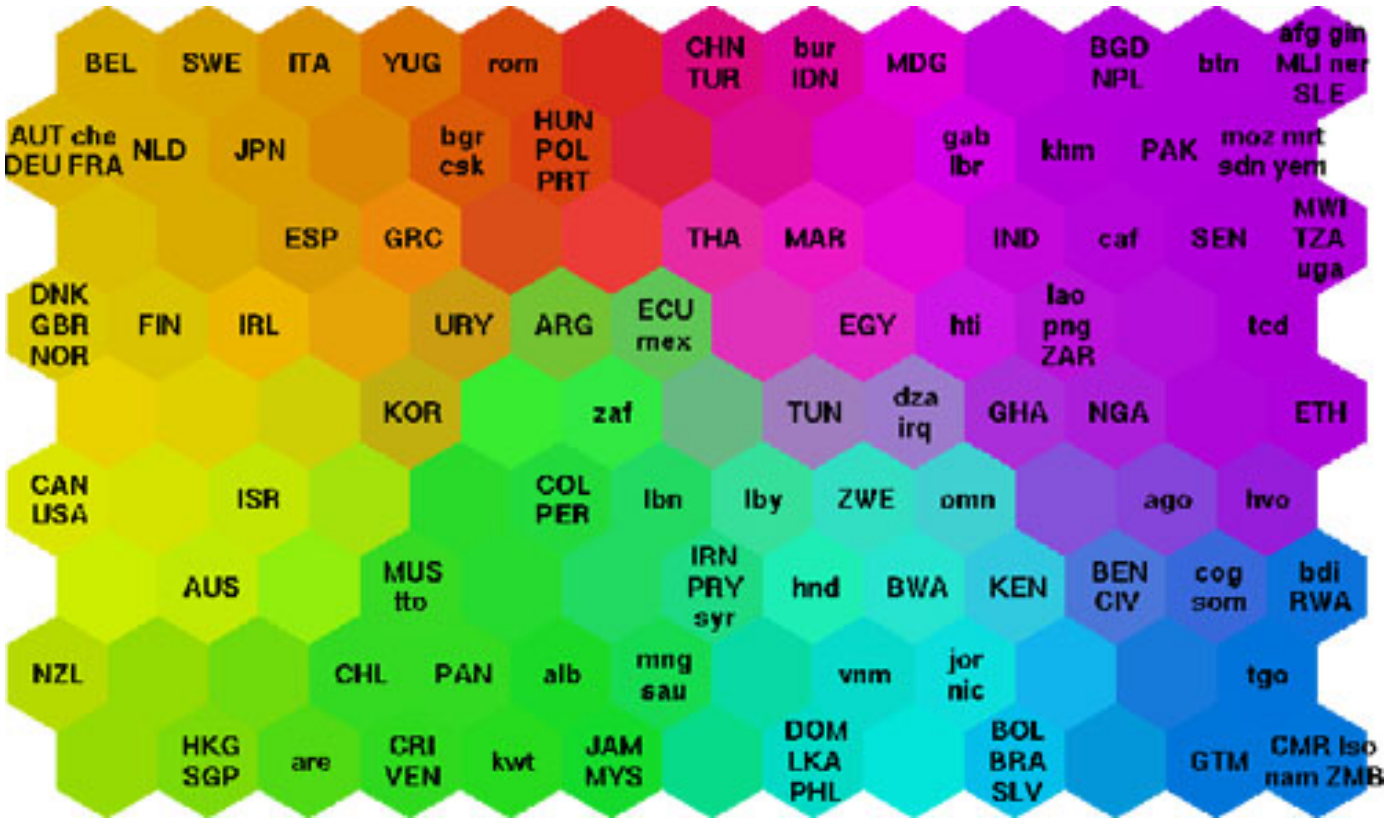
Quantos clusters ?

A qual dos clusters pertence um nó que está na região escura ?

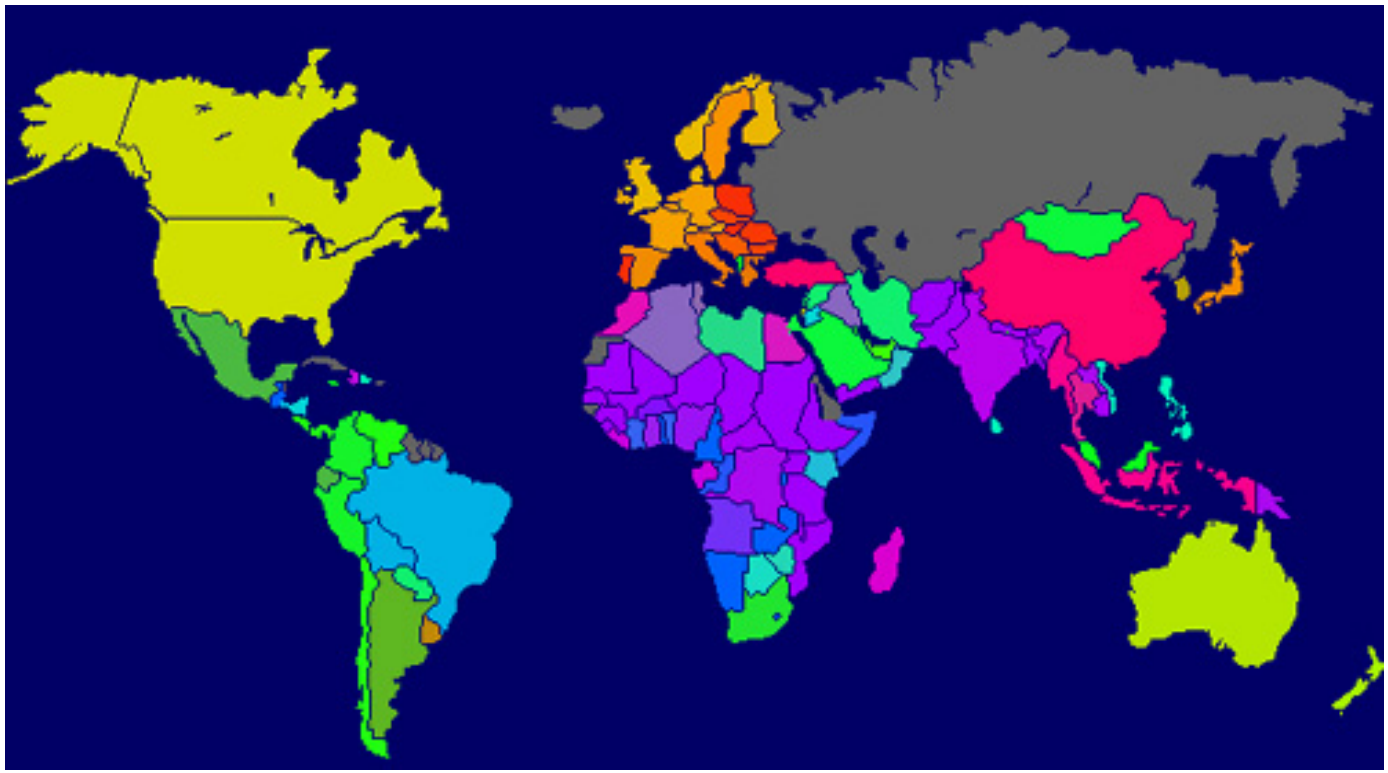
No exemplo anterior da cor, quantas cores (grupos) são ? A qual grupo pertencem os nós que ficam na fronteira de dois grupos (por exemplo, entre verde e azul?)

SOM – exemplo

Exemplo: espaço original consiste de várias estatísticas de países (índice de educação, saúde, etc).



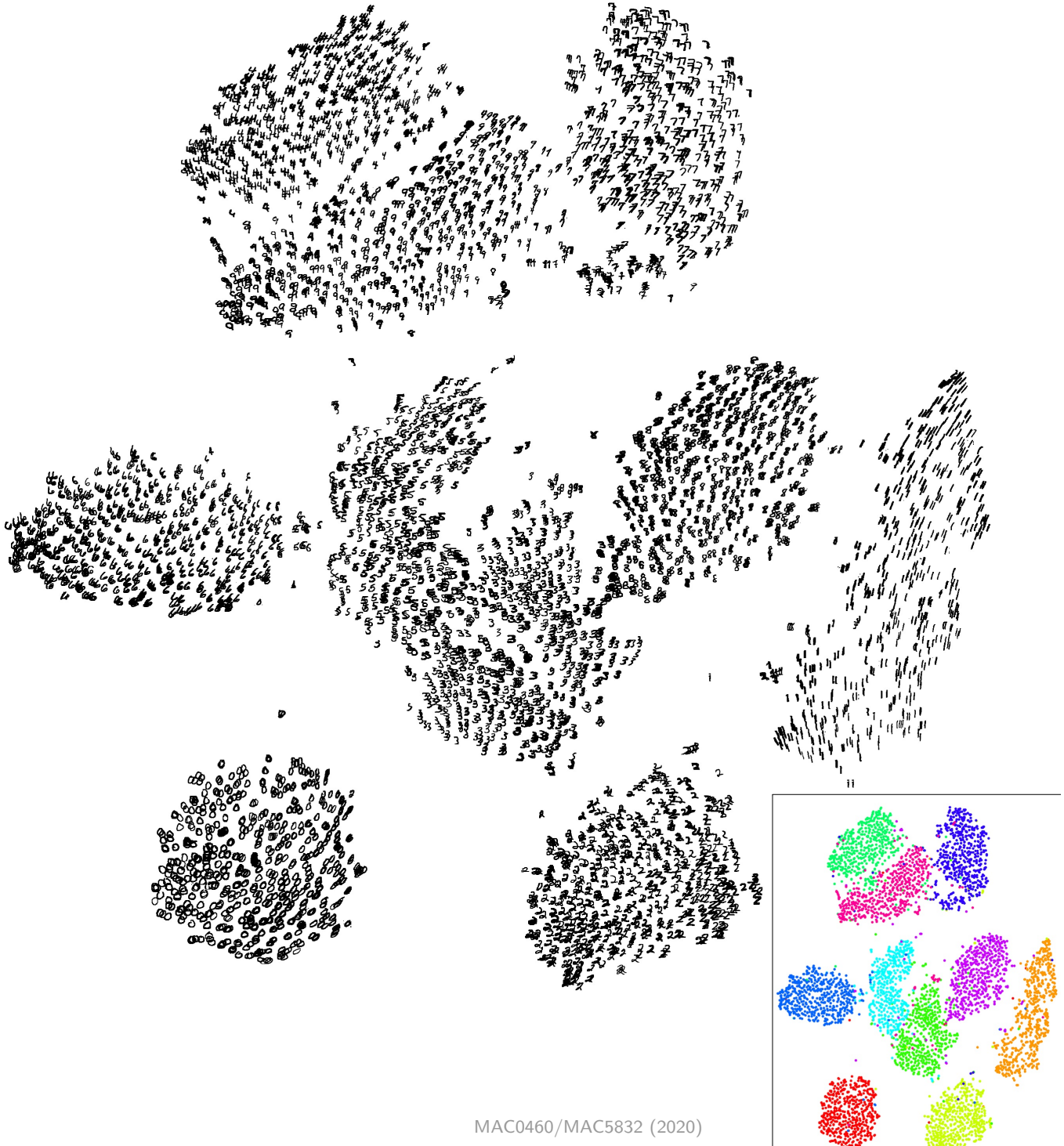
As cores do mapa podem ser atribuídas interpretando três componentes do vetor de pesos como RGB, ou usando pseudo-coloração



Data visualization

t-SNE (t-Distributed Stochastic Neighbor Embedding)

(Laurens van der Maaten)
https://lvdmaaten.github.io/tsne/



Data visualization is an active field of research

2D representation of data

Ways of visualizing multidimensional data

Color use

etc

Self-supervision

Principle: to learn features using a pretext task for which labels are freely or cheaply available

Example 1

To classify images, a pretext task could consist in rotating images slightly and training a regressor network that estimates the rotation angle

Then, the same network could be further trained using classification data

We expect that fewer classified data are needed for training the classification network

Example 2: **Word embedding**

In NLP, one of the first steps is to assign a code to each word (it could be a feature vector, but it is not very practical)

word2vec is a word embedding algorithm that uses a kind of self-supervision

Word codification is optimized based on tasks such as predicting the next word, or filling a missing word between two words, etc.

Data for these tasks is freely available (e.g., wikipedia)

Extras

Similarity between item and group

Distance between an item $\mathbf{x}$ and a cluster C:

- Smallest distance

$$d(\mathbf{x}, C) = \min_{\mathbf{y} \in C} d(\mathbf{x}, \mathbf{y})$$

- Largest distance

$$d(\mathbf{x}, C) = \max_{\mathbf{y} \in C} d(\mathbf{x}, \mathbf{y})$$

- Mean distance

$$d(\mathbf{x}, C) = \frac{1}{|C|} \sum_{\mathbf{y} \in C} d(\mathbf{x}, \mathbf{y})$$

Distance between an item and a group

Choose a representative item $\mathbf{m}$ in C and compute $d(\mathbf{x}, \mathbf{m})$

Possible representatives:

- **item** (spherical groups)
  - **mean item**: $\mathbf{m}_p = \frac{1}{|C|} \sum_{\mathbf{y} \in C} \mathbf{y}$
  - **mean item**: $\mathbf{m}_c \in C$ such that
    $\sum_{\mathbf{y} \in C} d(\mathbf{y}, \mathbf{m}_c) \leq \sum_{\mathbf{y} \in C} d(\mathbf{y}, \mathbf{m}), \quad \forall \mathbf{m} \in C$
  - **median item** : $\mathbf{m}_{med}$ such that
    $med\{d(\mathbf{y}, \mathbf{m}_{med}), \quad \mathbf{y} \in C\} \leq med\{d(\mathbf{y}, \mathbf{m}), \quad \mathbf{y} \in C\}, \quad \forall \mathbf{m} \in C$
- **hyperplane, hypercircle** : distance between item $\mathbf{x}$ and the curve that represents the cluster C

Distance/similarity between groups

- **Single linkage** (smallest)

$$d(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

- **Complete linkage** (largest)

$$d(C_i, C_j) = \max_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

- **Average linkage** (mean)

$$d(C_i, C_j) = \frac{1}{|C_i|\,|C_j|} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$

- distance between the representats of the groups

Clustering evaluation

An **ideal clustering** is one in which items within a group are highly similar each other, while items in disctinct groups are highly dissimilar

There are two common cluster similarity measures:

- within-class
- betweeb-class

Some notations:
Cluster j, $j = 1, 2, \ldots, c$:

$$C_j = \{\mathbf{x}_{j1}, \mathbf{x}_{j2}, \ldots, \mathbf{x}_{jn_j}\}, \quad j = 1, 2, \ldots, c$$

Center point of cluster C_j with n_j items:

$$\bar{\mathbf{x}}_j = \frac{1}{n_j} \sum_{\mathbf{x} \in C_j} \mathbf{x}$$

Global center point of the total of N items:

$$\bar{\mathbf{x}} = \frac{1}{N} \sum_{\mathbf{x} \in \mathbf{X}} \mathbf{x} = \frac{1}{N} \sum_{j=1}^{c} n_j \bar{\mathbf{x}}_j$$

Scatter matrices

Scatter matrix of cluster j:

$$S_j = \sum_{\mathbf{x} \in C_j} (\mathbf{x} - \overline{\mathbf{x}}_j)(\mathbf{x} - \overline{\mathbf{x}}_j)^t$$

Intra-class scatter matrix

$$S_W = \sum_{i=j}^{c} S_j$$

Between-class scatter matrix

$$S_B = \sum_{j=1}^{c} n_j(\overline{\mathbf{x}}_j - \overline{\mathbf{x}})(\overline{\mathbf{x}}_j - \overline{\mathbf{x}})^t$$

Total scatter matrix

$$S_T = \sum_{\mathbf{x} \in \mathbf{X}} (\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^t$$

Critérios baseados em matrizes de espalhamento

$$S_T = S_W + S_B$$

$$J_1 = \frac{\text{trace}\{S_T\}}{\text{trace}\{S_W\}}$$

$$J_2 = \frac{|S_T|}{|S_W|} = |S_W^{-1} S_T|$$

$$J_3 = \text{trace}\{S_W^{-1} S_T\}$$

Minimizar espalhamento intra-classes

$$tr[S_W] = \sum_{j=1}^{c} tr[S_j] = \sum_{j=1}^{c} \sum_{\mathbf{x} \in C_j} ||\mathbf{x} - \overline{\mathbf{x}}_j||^2$$

traço: soma dos elementos na diagonal da matriz

Minimizar o traço de S_W corresponde a minimizar o espalhamento intra-classes (k-means minimiza J_e)

$$tr[S_W] = J_e$$

Maximizar espalhamento entre-classes

Ao invés de minimizar o espalhamento intra-classes, podemos pensar em **maximizar o espalhamento entre-classes**.

Como $S_T = S_W + S_B$ e S_T não varia, **maximizar** $tr[S_B]$ **é equivalente a minimizar** $tr[S_W]$**.**

Minimizar volume de espalhamento

Critério do determinante:

$$J_d = |S_W| = |\sum_{j=1}^{c} S_j|$$

Determinante da matriz de espalhamento intra-classe pode ser pensado como o volume do espalhamento.

Em muitas situações resulta em agrupamentos similares ao do critério do traço.

Critério invariante

$S_W^{-1} S_B$: "razão entre o espalhamento entre e intra-classes"

λ_i auto-valores de $S_W^{-1} S_B$. Pode-se mostrar que

$$tr[S_W^{-1} S_B] = \sum_{i=1}^{d} \lambda_i$$

Logo, um bom critério seria maximizar $tr[S_W^{-1} S_B]$, ou seja, minimizar o critério invariante

$$J_d = tr[S_T^{-1} S_B] = \sum_{i=1}^{d} \frac{1}{1 + \lambda_i}$$

Um pouco de estatística amostral

Amostra

Conjunto $\mathbf{D}$ com N observações:

$$\mathbf{D} = \left\{ \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \right\},$$

$$\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{pmatrix} \in \mathbb{R}^d, i = 1, 2, \ldots, N$$

Vamos supor que cada componente x_j ($j = 1, \ldots, d$) é i.i.d.

Estatística amostral

Para cada componente x_j, $j = 1, 2, \ldots, d$:

média amostral:

$$\overline{x_j} = \frac{1}{N} \sum_{i=1}^{N} x_{ij}$$

covariância amostral:

$$s_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_{ij} - \overline{x_j})(x_{ik} - \overline{x_k})$$

variância amostral:

$$s_j^2 = s_{jj} = \frac{1}{N} \sum_{i=1}^{N} (x_{ij} - \overline{x_j})^2$$

correlação amostral:

$$r_{jk} = \frac{s_{jk}}{\sqrt{s_{jj}}\sqrt{s_{kk}}}$$

Em geral, no cálculo da variância e covariância, divide-se o valor do somatório por $N - 1$

Vetor de médias

$$\overline{\mathbf{x}} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_d)$$

Covariância é calculada entre duas variáveis x_j e x_k

$$s_{jk} = \frac{1}{N} \sum_{i=1}^{N} (x_{ij} - \overline{x_j})(x_{ik} - \overline{x_k})$$

covariância positiva: ambos crescem ou decrescem juntos

covariância negativa: um cresce e outro decresce ou vice-versa

covariância nula: nenhuma relação evidente