laSalle
Universitat Ramon Llull

‘Unsupervised Learning’

Xavier Vilasís
Elisabet Golobardes

Comcha School 2019

Concepto de clustering

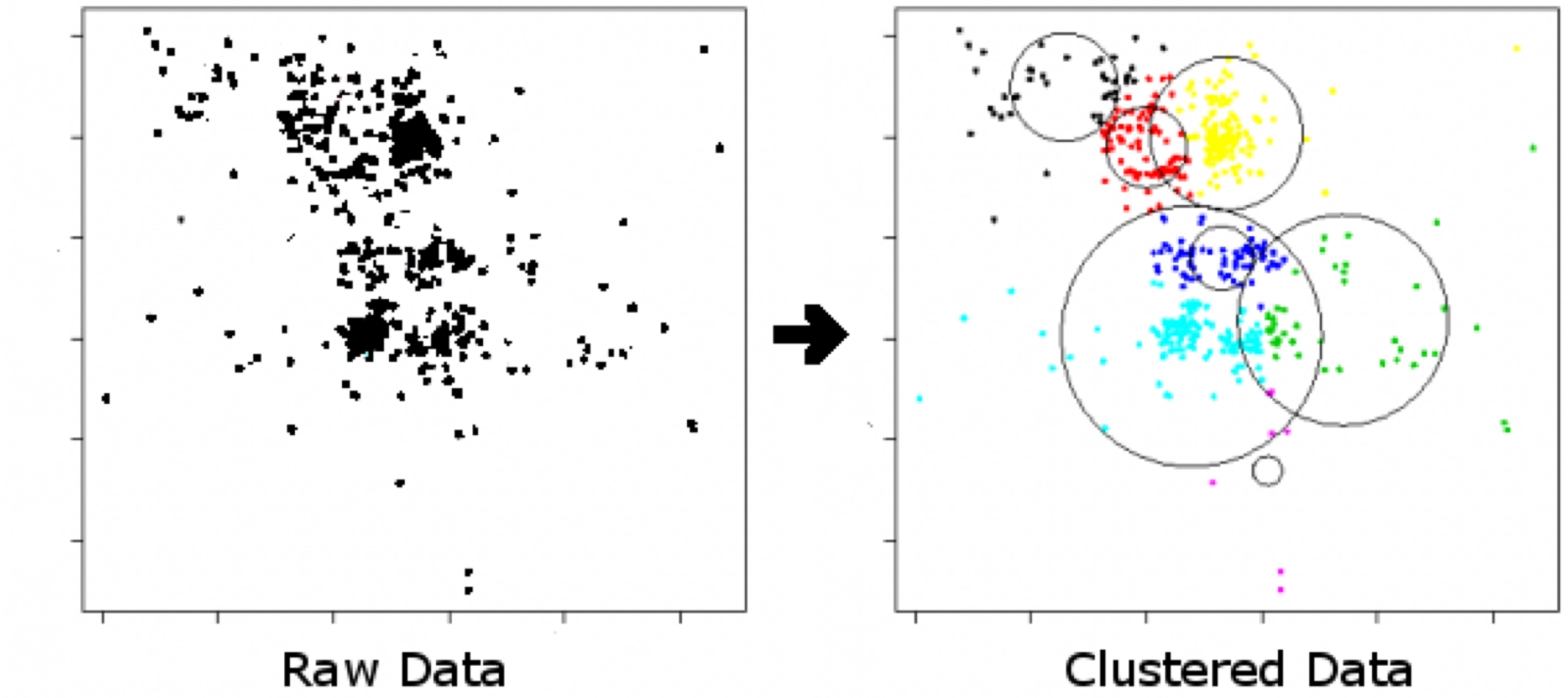
CONCEPTO

La técnica de clustering consiste en agrupar los datos en conjuntos de tal forma que:

- Cada conjunto sea lo más representativo posible para los objetos que pertenecen a él.
- Que los conjuntos se diferencien entre si.

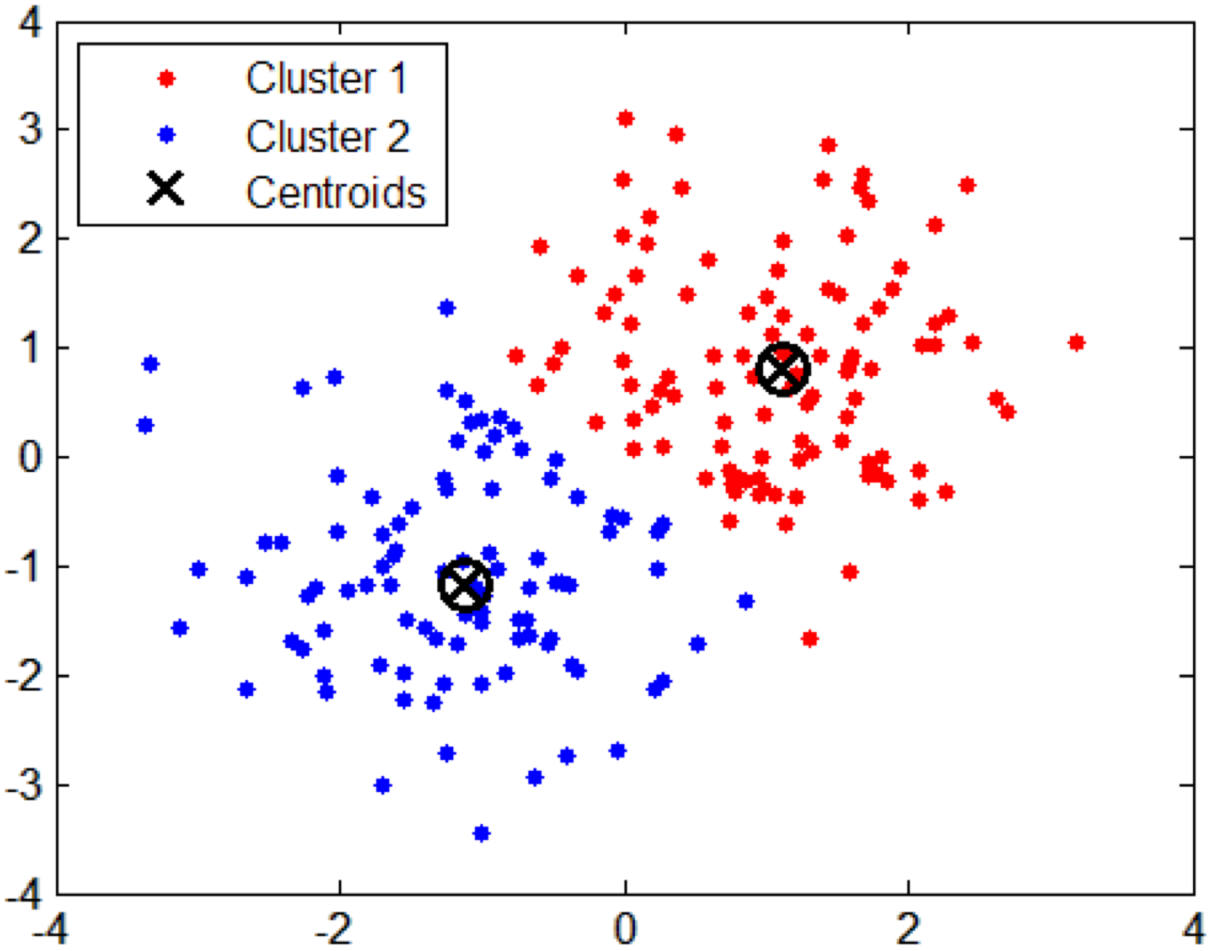
→ **APRENDIZAJE NO SUPERVISADO**

Concepto de clustering



Source:
http://stackoverflow.com/questions/1441319/whats-the-best-approach-to-recognize-patterns-in-data-and-whats-the-best-way

Modelos basados en centroides: K-means clustering



Source: http://mines.humanoriented.com/classes/2010/fall/csci568/portfolio_exports/mvoget/cluster/cluster.html

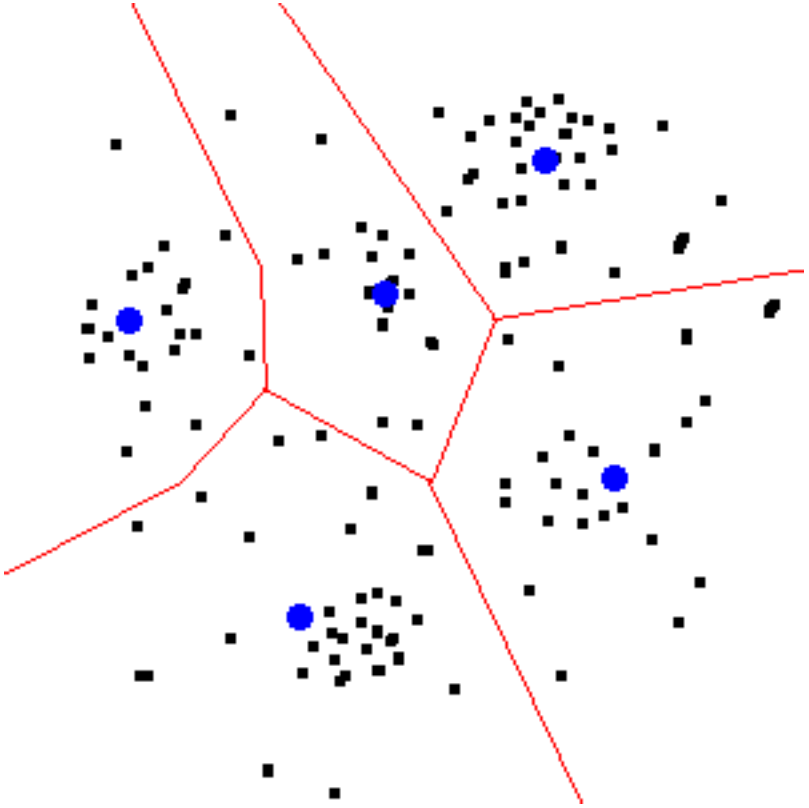
K-MEANS. Basic Algorithm

Basic K-means algorithm

1. Select K points as initial centroids
2. **Repeat**
3. **From** K clusters by assigning each point to its closest centroid
4. **Recompute** the centroid of each cluster
5. **Until** Centroids do not change

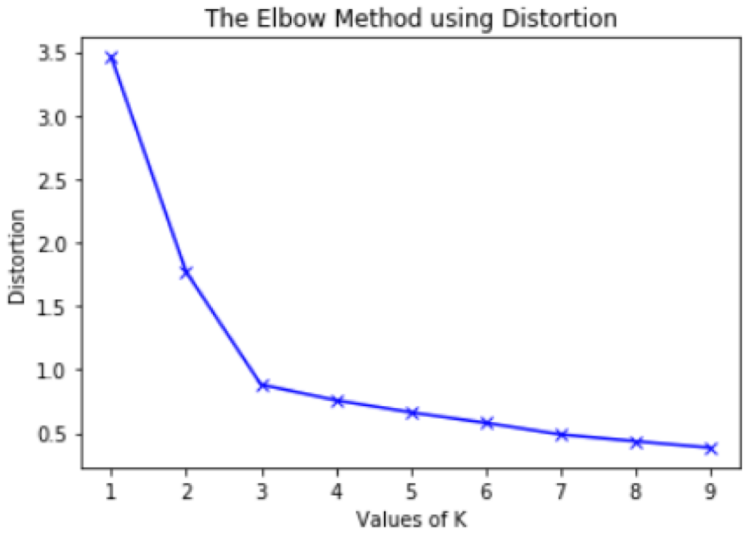
Space shaped by k-means

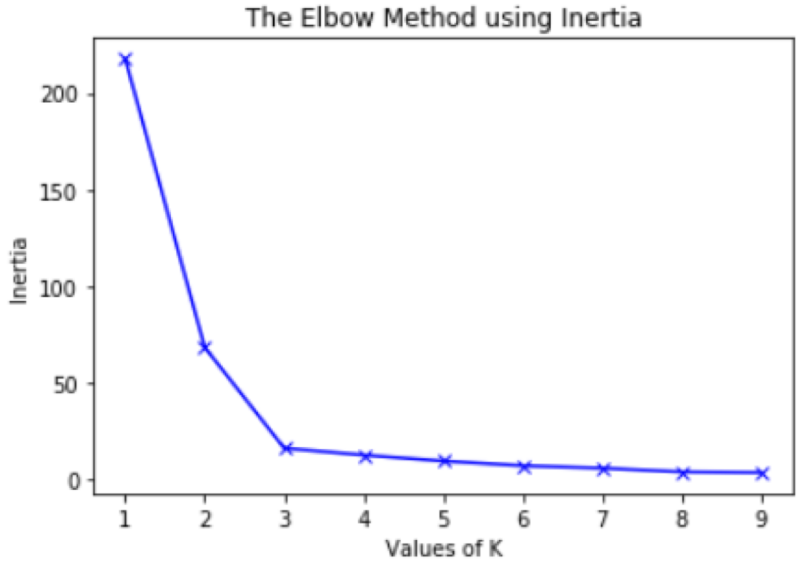
Different distances imply different clusterings



Elbow method to fix k

- **Distortion:** average of the squared distances from the cluster centers.
- **Inertia:** sum of squared distances of samples to their closest cluster center.





Market Segmentation

96% accurate segmentation is how the right marketing reaches the right people

Figure 1. Clusters

96% the right marketing to the right people

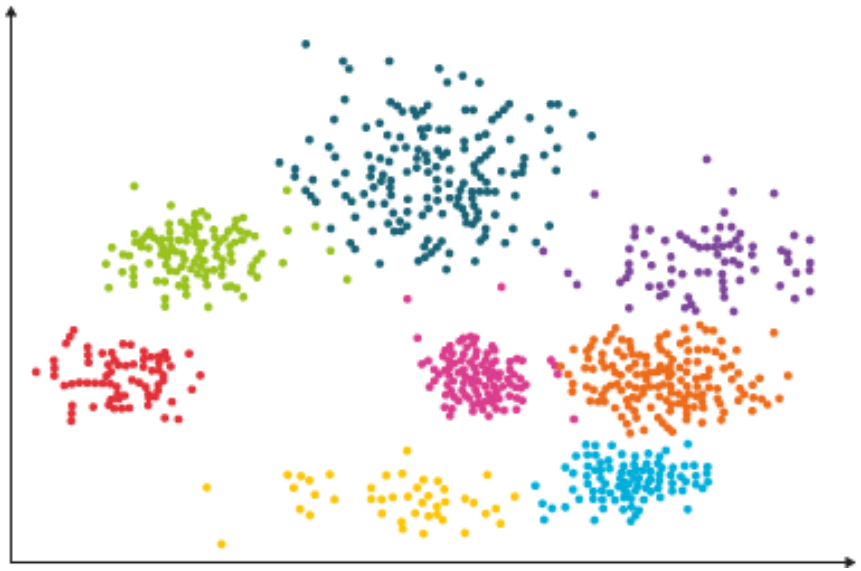
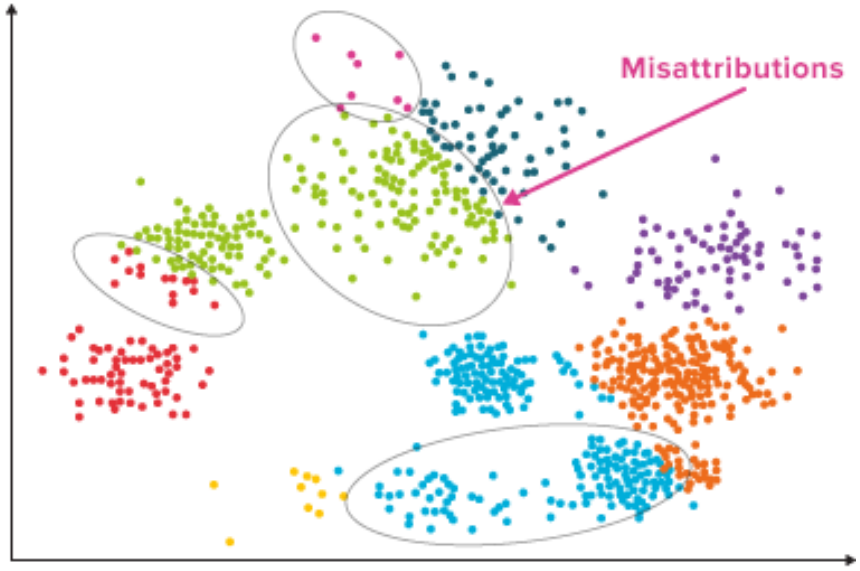


Figure 2. K-Means

62% the right marketing to the right people



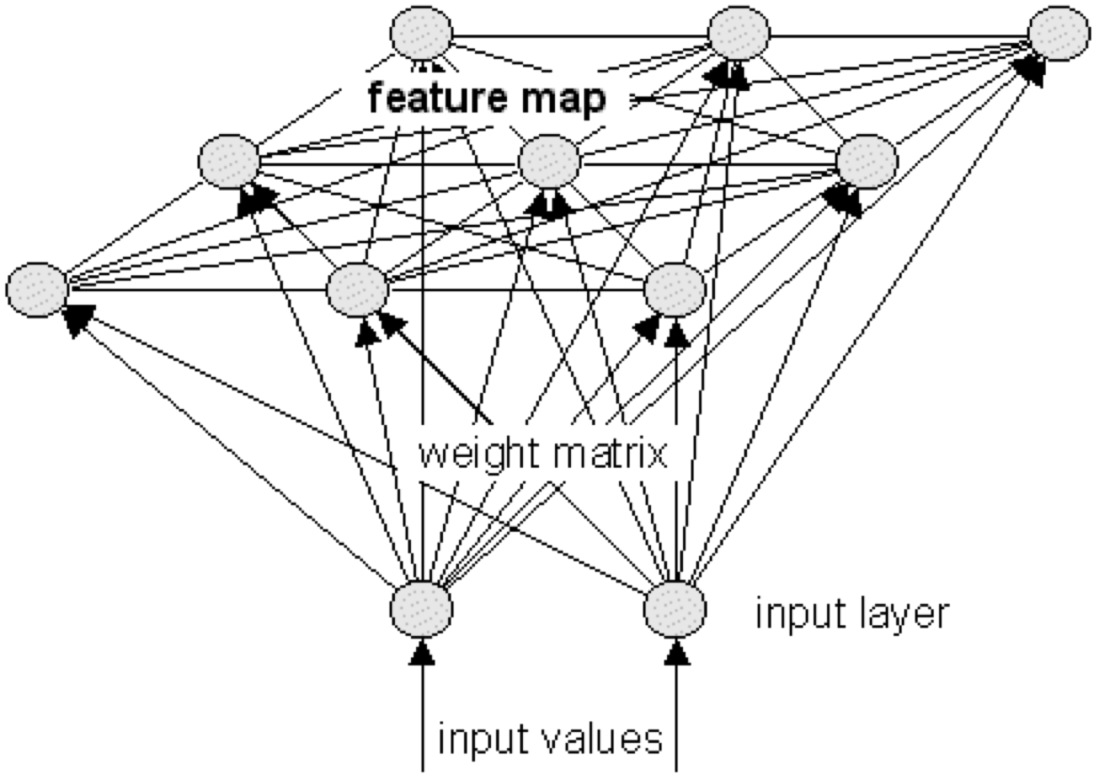
Source: <http://www.clusters.uk.com>

Mapas auto-organizativos (SOM). Arquitectura (1/2)

- Una SOM consiste en componentes llamados **nodos o neuronas**.
- Asociado a cada neurona hay un **vector de pesos**, de la misma dimensión de los vectores de entrada y una posición en el mapa.

Mapas auto-organizativos (SOM). Arquitectura (2/2)

ARQUITECTURA



Mapas auto-organizativos (SOM). Algoritmo de entrenamiento (1/2)

VARIABLES

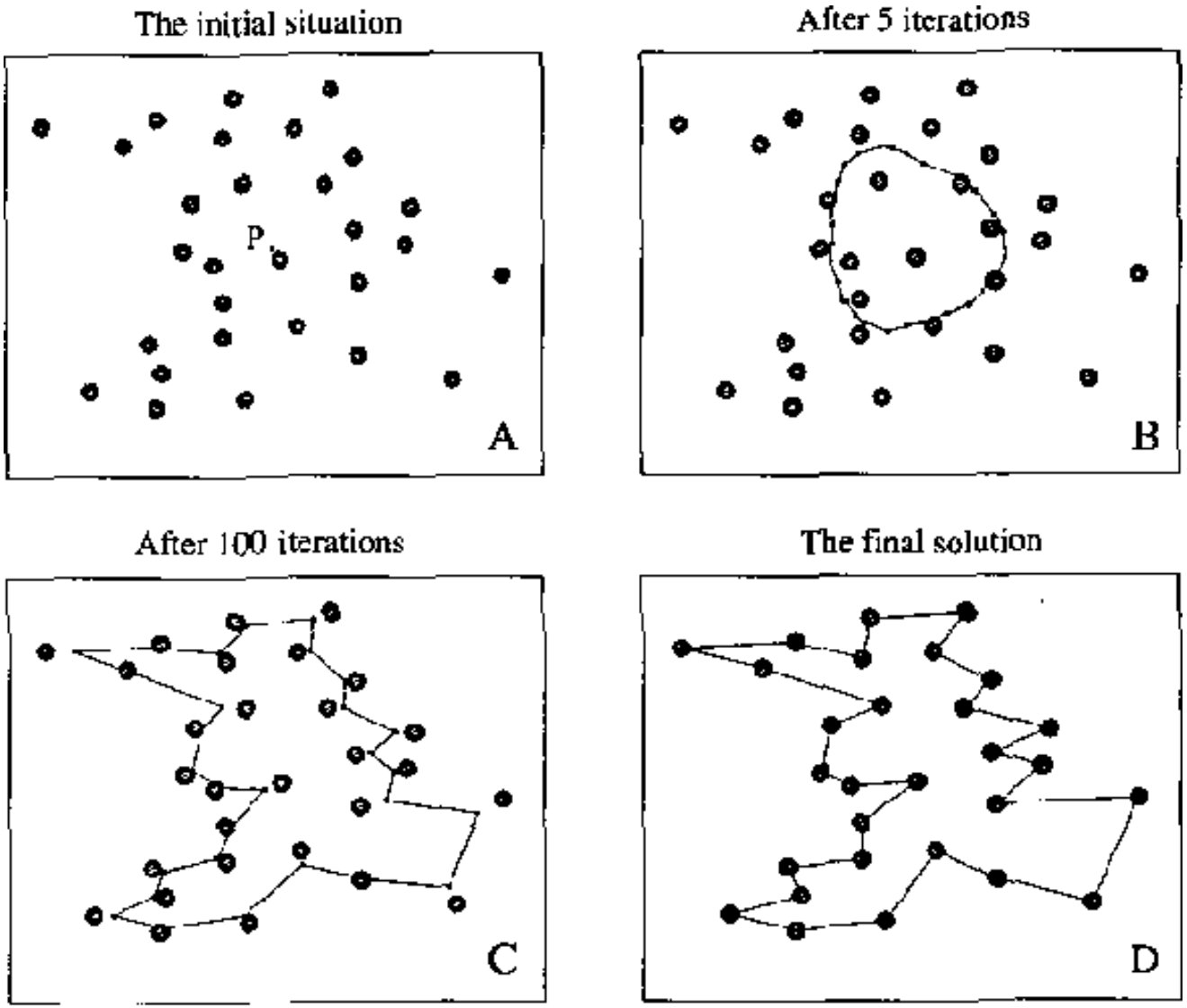
- BMU : Unidad de mejor correspondencia (*Best Matching Unit*)
- ***s*** : es la iteración actual.
- **λ** : es la cantidad total de iteraciones.
- ***t*** : es el índice del vector de entrada en el conjunto de datos de entrada ***D***.
- ***D(t)*** : es un vector de entrada de índice ***t*** del conjunto de datos de entrada ***D***.
- ***v*** : es el índice de una neurona en el mapa.
- ***Wv*** : es el vector de pesos de la neurona ***v***.
- ***u*** : es el índice del BMU en el mapa.
- **Θ*(u, v, s)*** : es la función de vecindad.
- **α(*s*) :** es un restrictor de aprendizaje debido al progreso de las iteraciones. Factor de aprendizaje.

Mapas auto-organizativos (SOM). Algoritmo de entrenamiento (2/2)

ALGORITMO

1. Hacer un mapa de neuronas con vectores de pesos aleatorios.
2. Tomar un vector de entrada D(t).
   21. Iterar por cada neurona del mapa.
      211. Calcular la distancia entre el vector de entrada y los vectores de pesos de las neuronas del mapa (Distancia euclidiana).
      212. Mantener la neurona que ha tenido la menor distancia, esta neurona será el BMU.
   22. Actualizar las neuronas en la vecindad del BMU.
      221. $Wv(s + 1) = Wv(s) + \Theta(u, v, s)\, \alpha(s)(D(t) - Wv(s))$
3. Incrementar s y volver al paso 2 mientras $s < \lambda$

Solving the TSP with SOM



Zhu, Yang DOI:10.1109/RISSP.2003.1285655

Principal Component Analysis

3. Principal Component Analysis and Gaussian Variables

3.1. PCA basics

The first principal component of a data set is the direction with largest variance projection,

$$\vec{w}^1 = \max_{\|\vec{w}\|=1} < (\vec{w}^T \vec{x})^2 > . \tag{14}$$

Recuersively, one obtains further principal components by looking at directions with largest variance projection, once the previous components are substracted.

$$\vec{w}^k = \max_{\|\vec{w}\|=1} < \left(\vec{w}^T \left(\vec{x} - \sum_{i=1}^{k-1} \vec{w}^i \vec{w}^{iT} \vec{x} \right) \right)^2 > . \tag{15}$$

Actually one should remark that equation 14, is equivalent to

$$\vec{w}^1 = \max_{\|\vec{w}\|=1} \vec{w}^T < x x^T > \vec{w} = \max_{\|\vec{w}\|=1} \vec{w}^T C \vec{w}. \tag{16}$$

This constrained equation has extrema on every unitary eigenvector of the correlation matrix. To obtain the maximum, we must choose the eigenvector with larger eigenvalue. If we order the eigenvectors of C from larger to lower eigenvalue, we have,

$$\vec{w}^1 = \vec{e}_1.$$

PCA

Equation for $\vec{w}^2$ then becomes,

$$\vec{w}^2 = \max_{\|\vec{w}\|=1} < \left(\vec{w}^T\vec{x} - \vec{w}^T\vec{e}_1\vec{e}_1\vec{x}\right)^2 >, \tag{17}$$

which can be rewritten,

$$\vec{w}^2 = \max_{\|\vec{w}\|=1} \left(\vec{w}^T C\vec{w} - \lambda_1(\vec{w}^T\vec{e}_1)^2\right). \tag{18}$$

Since $\lambda_1 > 0$ and $(\vec{w}^T\vec{e}_1)^2 > 0$, the maximum can only be obtained for $\vec{w}^T\vec{e}_1 = 0$ and so, we must resort to the eigenvector of C which has largest eigenvalue after $\vec{e}_1$, namely $\vec{e}_2$. This argument can be expanded to the rest of the principal component definition to simply recover the covariance matrix unitary eigenvectors.

The principal components of a data point are then

$$s_i = \vec{w}^{iT}\vec{x} = \vec{e}_i^T\vec{x}, \tag{19}$$

namely, the projections over the correlation matrix eigenvectors. This means that the data point can be written in the eigenvector base as

$$\vec{x} = \sum_{i=1}^{n} s_i\vec{e}_i \tag{20}$$

We can define the representation error as

$$\vec{\varepsilon} = \vec{x} - \sum_{i=1}^{n} s_i\vec{w}_i = \vec{x} - \sum_{i=1}^{n} \vec{w}_i^T\vec{x}\vec{w}_i. \tag{21}$$

PCA

The average representation error is then,

$$E =< \vec{\varepsilon}^T \vec{\varepsilon} >, \tag{22}$$

operating, we can obtain,

$$E =< \vec{x}^T \vec{x} > - < \sum_{i=1}^{n} (\vec{x}^T \vec{w}_i)^2 >, \tag{23}$$

Since $< \vec{x}^T \vec{x} >$ is constant, minmising E is equivalent to maximising

$$\sum_{i=1}^{n} < (\vec{x}^T \vec{w}_i)^2 > .$$

The solution for unitary $\vec{w}_i$ is, we know, the set of unitary eigenvectors of the correlation matrix, namely the principal components of the data set.

Therefore, PCA gives a faithfull data representation in the mean square sense.

PCA

3.2. PCA for Gaussian Variables

PCA finds the rotation for which data variables are decorrelated. In effect, if we build up a matrix

$$W = (\vec{e}_1 \ \vec{e}_2 \dots \ \vec{e}_n) \tag{24}$$

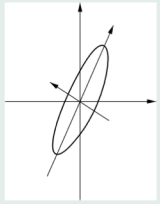
where the $\vec{e}_i$ are the unitary principal components, their orthonormal properties provide

$$W^T W = \mathbf{1}. \tag{25}$$

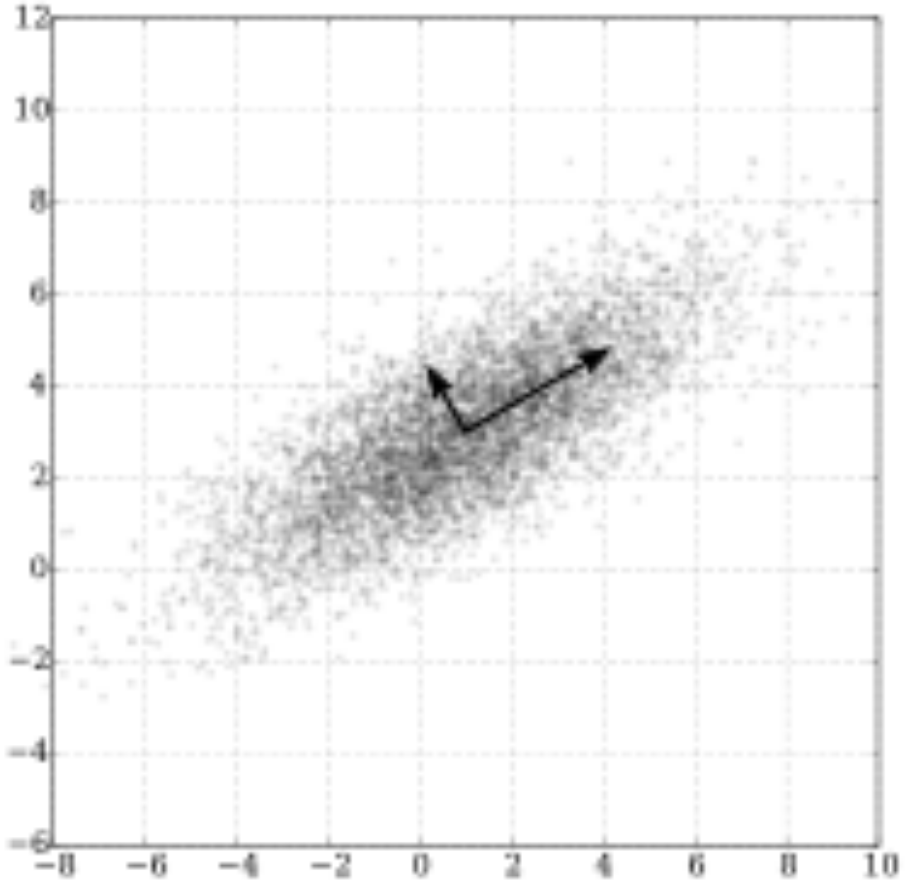
The principal component vector for a data sample is the rotation under W of the data

$$\vec{s} = W^T \vec{x}. \tag{26}$$

In the case of joint zero mean Gaussian distributed data, the whole distribution is characterised just by the covariance matrix. Higher order cumulants are zero. PCA gives then, the directions for which gaussian variables are independent, since decorrelated gaussian variables are independent.



PCA



wikipedia

Independent Component Analysis

4. Independent Component Analysis

4.1. ICA basics

PCA decorrelates the input data, but is not sufficient to separate into independent components non-gaussian data. Such task requires a different approach.

Our representation problem is to find a matrix W such that

$$\vec{s} = W\vec{x}, \tag{32}$$

being the components of $\vec{s}$ independent. Actually, we are assuming that our data $\vec{x}$ is a linear mixing of originally independent $\vec{s}$,

$$\vec{x} = A\vec{s}, \tag{33}$$

so that $A = W^{-1}$. Actually there is a large ambiguity in this definition since any rescaling of data $\vec{x}$ or of independent components $\vec{s}$ can be absorbed into matrices W and A.

The working principle of ICA is the fact that the sum of two independent components is *more gaussian* than the separate components. This is a first step into the central limit theorem.

ICA tries to find the rotation W making $\vec{s}$ as non-gaussian as possible. For so, we need to find out non-gaussianity measures.

Kurtosis is the fourth cumulant of a distribution. Recall odd order cumulants are zero for symmetric distributions and Gaussians have all their

ICA

higher (than second) order cumulants equal to zero. For zero mean variables, we have,

$$\text{kurt}(x) =< x^4 > -3(< x^2 >)^2. \tag{34}$$

However, non-gaussian variables may have zero kurtosis. Its computation on sample data is sensitive to outliers.

Negentropy The entropy of a random variable is a measure of the information coded into it. The Shannon entropy is defined by

$$S(\vec{x}) = -\int d\vec{x}\ f(\vec{x}) \log f(\vec{x}), \tag{35}$$

being $f(\cdot)$ the probability density function. A Gaussian variable has the largest entropy among all random variables of equal variance. As such, a measure for non-gaussianity is the negentropy, namely,

$$J(\vec{x}) = S(\vec{x}_{Gauss}) - S(\vec{x}), \tag{36}$$

where $S(\vec{x}_{Gauss})$ is a gaussian random variable with the same mean and variance than $\vec{x}$.

Negentropy can be approximated by kurtosis

$$J(x) \simeq \frac{1}{12}(< y^3 >)^2 + \frac{1}{48}\text{kurt}(x)^2 \tag{37}$$

ICA

but off course this approximation suffers from the very same problems than the kurtosis itself. In general, it can be proved that

$$J(x) \sim (< G(x) > - < G(x_{Gauss}) >)^2 \tag{38}$$

for any quadratic function of x. Well known examples are

$$G(x) = \frac{1}{a} \log \cosh au, \quad G(x) = -\exp -\frac{x^2}{2}.$$

Mutual Information compares the entropy of the joint distribution with the entropy of the individual components,

$$I(\vec{x}) = \sum_{i=1}^{n} S(x_i) - S(\vec{x}). \tag{39}$$

For any invertible linear transformation, $x' = Wx$,

$$I(\vec{x}) = \sum_{i=1}^{n} S(x_i) - S(\vec{x}') + \log \| \det W \|. \tag{40}$$

Using this property, mutual information can be related to negentropy for uncorrelated unit variance random variables,

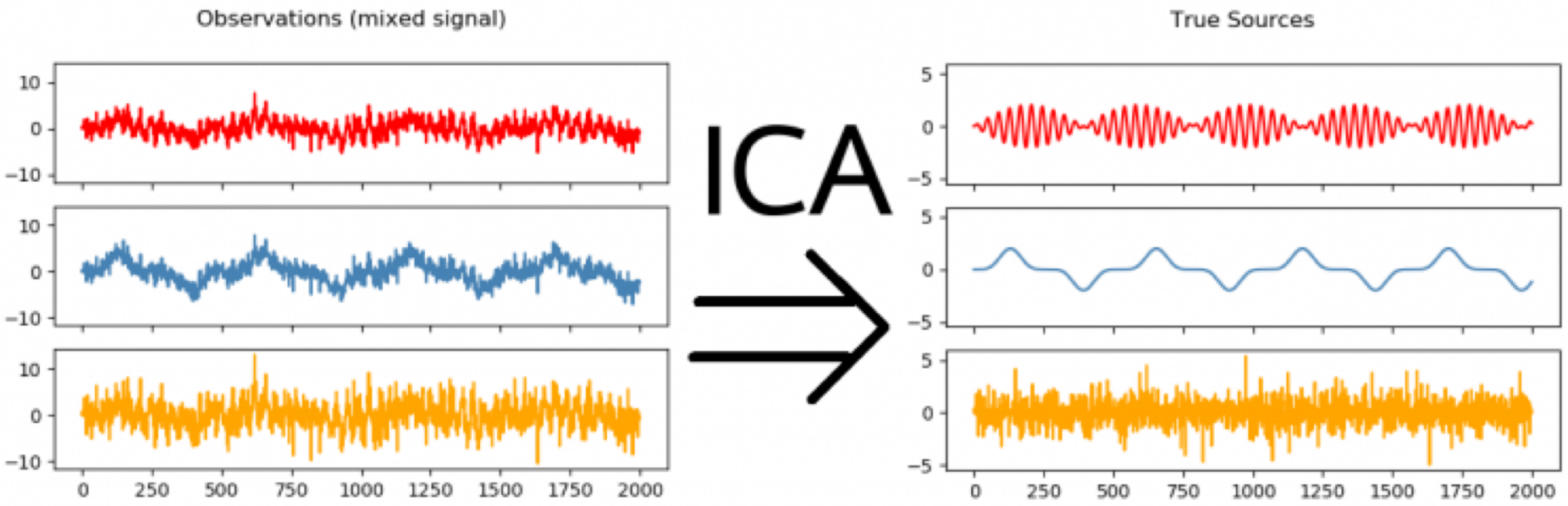
$$I(\vec{x}) = c - \sum_{i=1}^{n} J(x_i), \tag{41}$$

where c is a constant.

ICA

Therefore, a *simple* method to achieve independent component analysis, is finding a transformation minimising the mutual information of the mixing, computed through the formula related to negentropy.

In this sense, ICA, just as PCA looks for a faithfull representation of data, but with different objective function.



From INRIA website

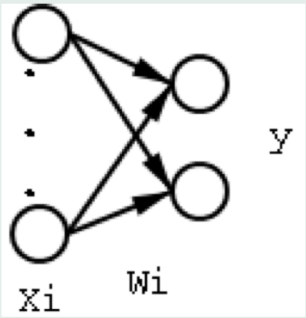
Fast ICA

4.2. FastICA

The FastICA algorithm is a particularly successful algorithm to perform Independent Component Analysis. It is based on finding the rotation minimising the mutual information of the data set.

- mutual information is computed using the negentropy formula.
- negentropy is estimated using non-quadratic functions, as shown before.
- data needs some preprocessing.

FastICA trains a linear perceptron with as many outputs as independent components we want to find,



$$y_i = \sum_{j=1}^{n} w_{ij} x_j = \vec{w}_i^T \vec{x}. \tag{42}$$

Fast ICA

Preprocessing To use the FastICA algorithm data need to be

- Zero mean : we substract the mean,
- Decorrelated : can be done by PCA,
- Unit variance : compute the variance and rescale data.

One unit FastICA If we denote by G the function approximating the negentropy, the Newton optimisation method (with some approximations for the jacobian) on the mutual information constrained to find unitary solutions, delivers the following process,

1. Choose an initial random weight vector $\vec{w}$.
2. Let
$$\vec{w}^+ =< \vec{x}G(\vec{w}^T x) > - < G'(\vec{w}^T x) > \vec{w} \tag{43}$$
3. Let
$$\vec{w} = \frac{\vec{w}^+}{\|\vec{w}^+\|} \tag{44}$$
4. Return to point (2) until convergence.

Fast ICA

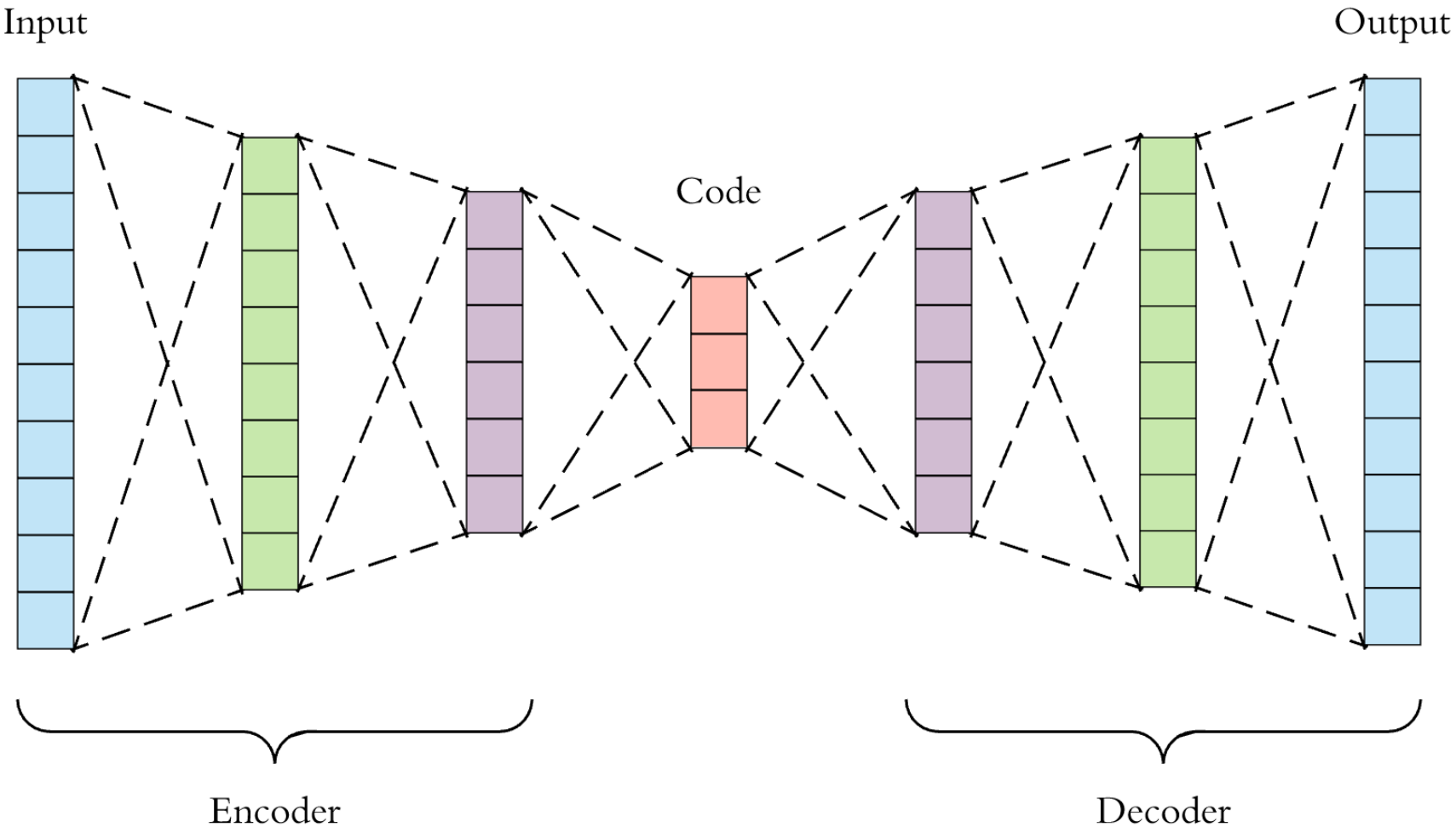
Extension to several units To obtain further independent components, we shall cast out the projections over the already known combinations. So, we estimate p independent component vectors $\vec{w}_1 \dots \vec{w}_p$. Then we run the one unit fixed point algorithm for $\vec{w}_{p+1}$, but at each iteration state we subtract the projection on the p previous vectors. That is, after point (3), we perform,

1. Let
$$\vec{w}_{p+1} = \vec{w}_{p+1} - \sum_{j=1}^{p} (\vec{w}_{p+1}^T \vec{w}_j)\vec{w}_j, \tag{45}$$
2. Let
$$\vec{w}_{p+1} = \frac{\vec{w}_{p+1}}{\|\vec{w}_{p+1}\|} \tag{46}$$
3. Return to point (2) until convergence.

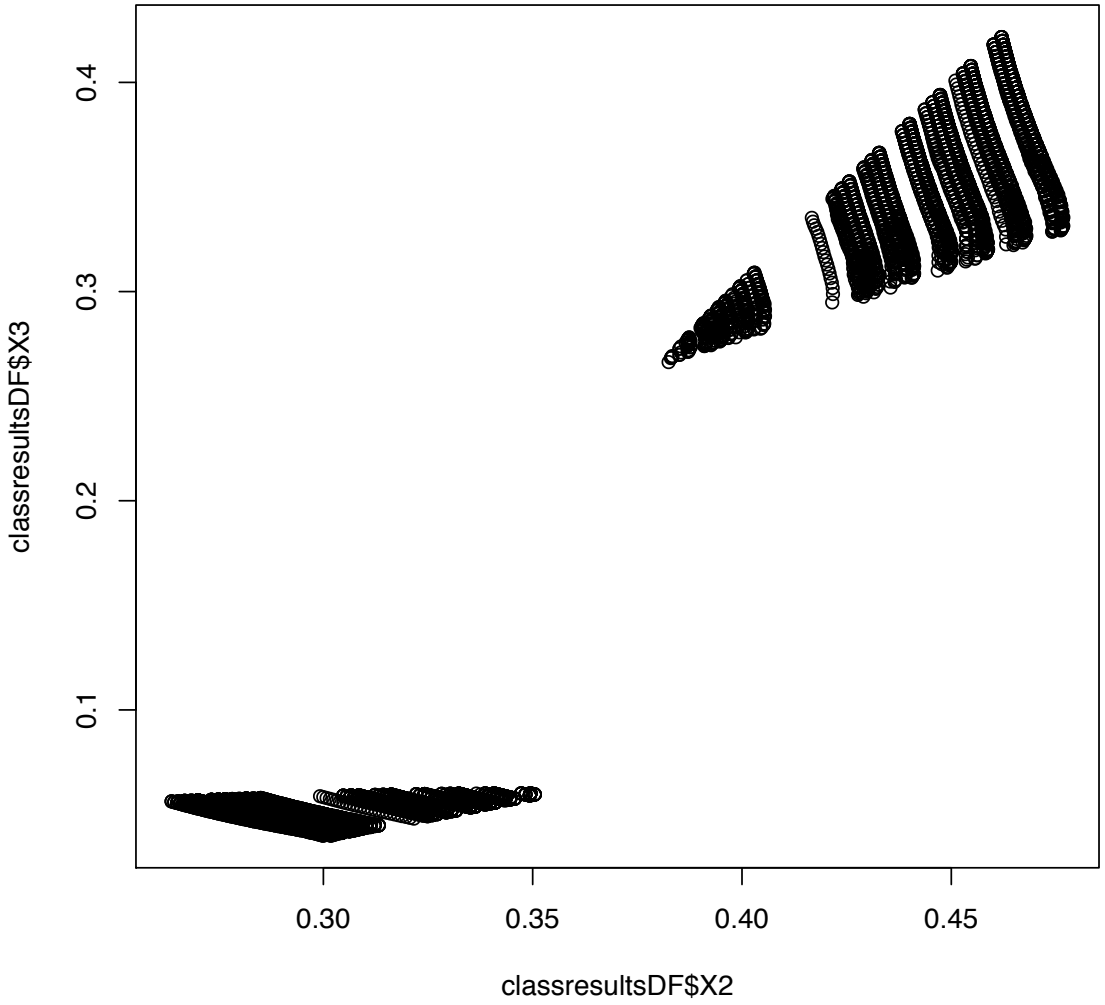
Examples of ICA applications:

- Medical Data : EEG, EEC, Brain activity from magnetic resonance images.
- Financial Data : Stock Market evolution.
- Telecomunications : Selecting CDMA signals.

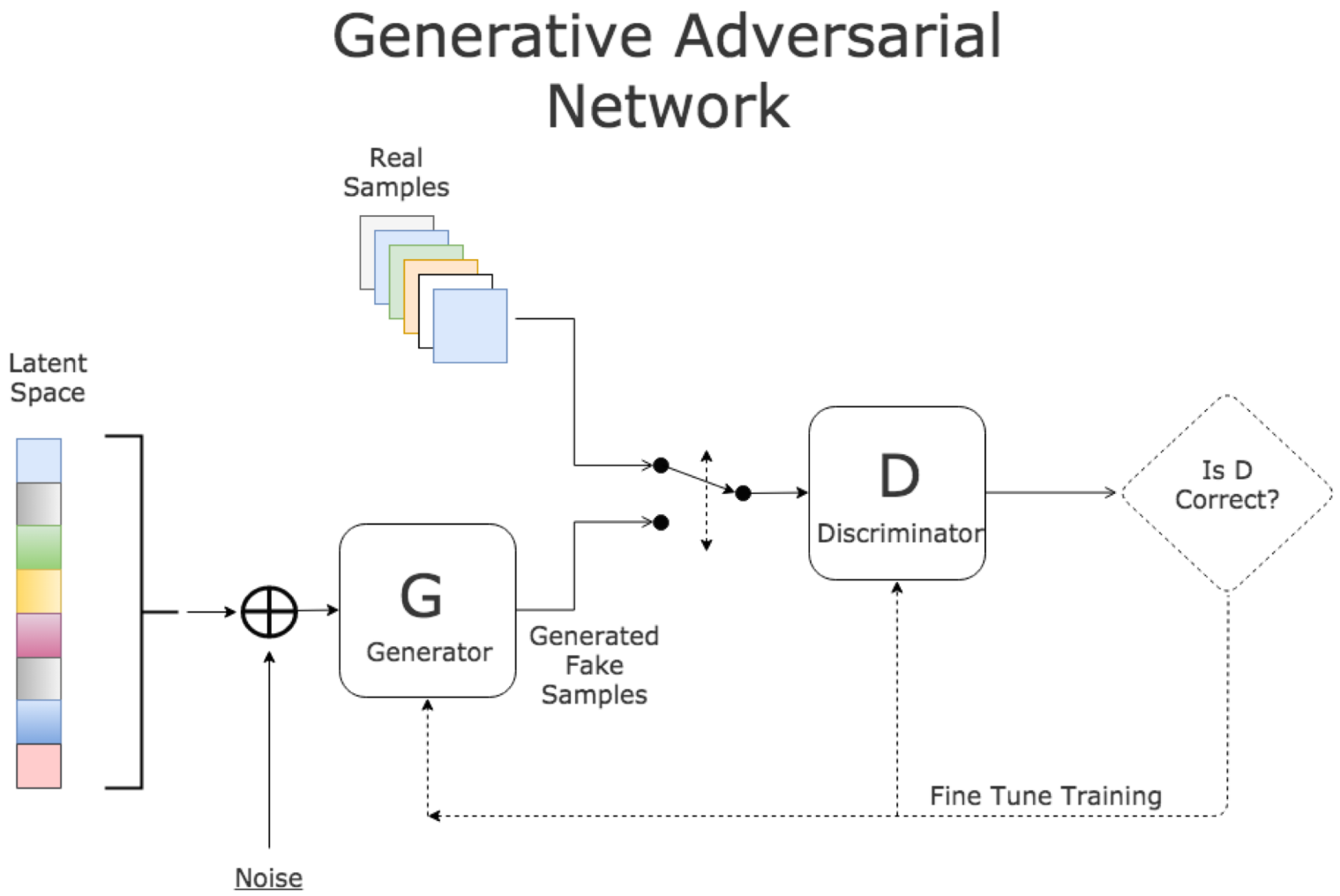
Autoencoders



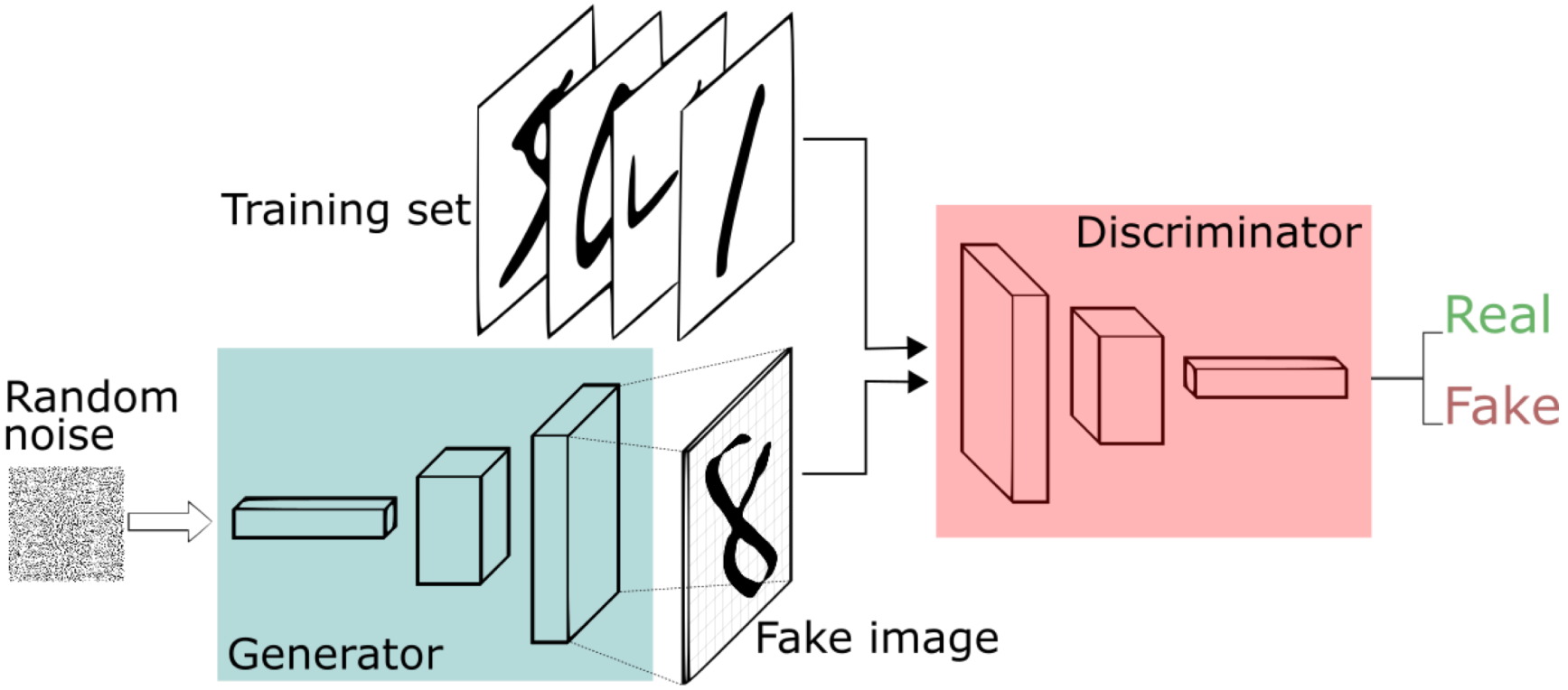
Using Autoencoders on a Time Series



Truly Fancy Stuff : GANs



More GANs


Training set
Random noise
Generator
Fake image
Discriminator
Real
Fake

laSalle
Universitat Ramon Llull

‘Unsupervised Learning’

Xavier Vilasís
Elisabet Golobardes

Comcha School 2019