Minería de Datos

U4 Reducción de Dimensionalidad

Héctor Maravillo

Section 1

Introduction

What is Data Science?

Data science is about solving **problems** based on observations of **factors** (referred to as co-variates, predictors, or just **features**) that may determine **a solution**.

Typical kinds of problems are:

- Classification.
- Prediction.
- Clustering

Classification

A **classification problem** is defined by a **partition** Π of a population Ω (into a number of **mutually disjoint classes** exhaustive of Ω) and the problem is to place an arbitrary element of the population into the right class.

A solution to the problem is a **model** that will make a (hopefully correct) decision for every element from the population (not just the sample) as to which class it belongs to.

Example

Biotaxonomic Classification

- INSTANCE: A DNA sequence representing a living organism x.
- QUESTION: What species in T does x belong in?

Prediction

A **prediction problem** is defined by a **function** f on a population that **associates** numerical values to every element in the population (usually hard to measure directly) and the problem is to find the value $f(x)$ of the function f for an arbitrary element x in the population Ω.

A solution is a **model** to determine that value based on other features identifying the input elements x from Ω.

Example

Provenance of a plant

- INSTANCE: A (long) DNA sequence x describing a plant in T.
- QUESTION: Where on Earth (latitude, longitude), was x grown?

Clustering

A **clustering problem** is defined by a **metric** (usually a **distance function**) function between elements in a population to capture degrees of **(dis)similarity**, and the problem is to produce a **partition** Π (unlike a classification problem, where the partition is given).

A solution is to find a **model** that will produce a partition such that elements in any one cluster are more similar among each other than to elements in the other clusters.

Example

Gene clustering

- INSTANCE: A DNA sequence representing a living organism x.
- QUESTION: How are DNA segments grouped according to their characteristics and functions?

Big Data

Big data is usually characterized by the so-called five V 's:

- **Volume:** Raw number of records/amount of data.
- **Variety**: How diverse is the type, nature, and format of the data.
- **Velocity**: Speed of data generation.
- **Veracity:** Quality of captured data.
- **Value:** Insight and impact of data on the general population.

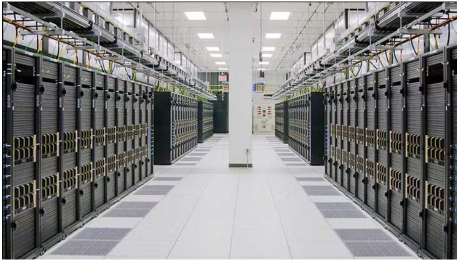


Figure: Data Hall at Meta Research Supercluster

Big Data

Computational resources today are not able to process the extraordinary full volume of data being generated in so many fields.

Because of the overwhelming **growth of data** collection, it can be challenging to **extract useful information** from big data with current computational resources.

Choosing a **good subset** of the data as the training sample to build some suitable models is a critical issue.

Data reduction is a critical step to turn large datasets into useful information, the overarching purpose of data science.

Section 2

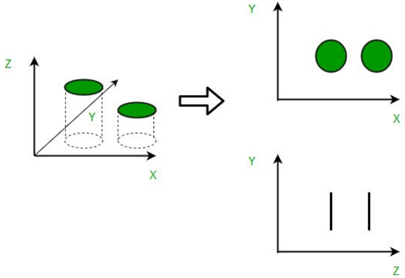
Dimensionality Reduction

Dimensionality reduction

In the simplest case, a **problem** (of Data Science) can be defined by a number of **features**, one of which is the so-called **response (random) variable** in statistics, or simply **target** in Data Mining context.

The remaining features are the so-called **predictors** or **independent variables** in statistics, or simply **input features** in Data Mining.

A **sample** of these features from the corresponding population constitutes the data to be used to solve the problem



Dimensionality reduction

The sample **X** can be regarded as a table consisting of a number n of observations ($n > 1$) given by vectors $(x_1, y_1), \ldots, (x_n, y_n)$ where $\mathbf{x}_i = (x_{i1}, \ldots, x_{il})$ is a l-**vector of input features** for the i-th observation and y_i is the **target variable**.

Thus the dataset can be represented as a matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & \ldots & x_{1l} \\ x_{21} & \ldots & x_{2l} \\ \ldots & & \\ x_{n1} & \ldots & x_{nl} \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_n \end{bmatrix}, \qquad \mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ \ldots \\ y_n \end{bmatrix}$$

where the input matrix can be viewed as n row vectors of dimension l $[\mathbf{x}_1, \ldots, \mathbf{x}_n]^T$ (the **data points**); or l column vectors of dimension n $[\mathbf{X}_1, \ldots, \mathbf{X}_l]$ (the **features**).

Dimensionality reduction methods

In general, the overarching goal of **dimensionality reduction (DR)** is to find **lower dimensional representations** of data that **preserve** their key properties for a given problem.

Historically, the **classical technique** to DR was **Principal Component Analysis**, commonly referred to as PCA.

Other techniques were eventually developed to include **feature/variable selection** (particularly including targets) rather than feature extraction.

In general, choosing between feature extraction with PCA or just feature/variable selection depends mostly on the problem being solved.

Statistical Approach

The classical **statistical approach** assumes that the solution to a given problem is **some function** f of l features in $\mathbf{X}$ that produces the response vectors $\mathbf{Y}$ (containing the responses for all the n input feature vectors in the data), where l is too large to efficiently build a good approximation of f.

Typically, $\mathbf{X}$ may have a **high correlation** among some of the columns, or some columns may have a nonlinear relationship.

Therefore, one can first attempt **to reduce the number of features** to a much smaller number k ($k \ll l$).

Data transformation function

This reduction can be viewed as a **transformation** Ψ of the original dataset $\mathbf{X}$ into another matrix $\mathbf{X}^*$ with k columns,

$$\Psi(X) = \mathbf{X}^* = [\mathbf{X}_1^*, \mathbf{X}_2^*, \ldots, \mathbf{X}_k^*]$$

This reduced feature set could then be used to solve the problem of determining the response from the features as a relation between $\mathbf{Y}$ and $\mathbf{X}^*$ given by

$$\mathbf{Y} = f(\Psi(\mathbf{X}^*)) + \epsilon$$

where ϵ is a random error term and $f(\cdot)$ is the response function.

The function Ψ could be a **selection** from the original (given raw) features in $\mathbf{X}$ or could be a **combination** of them into other new derived (or abstract) features.

Example

A **linear mapping** is most useful when some of the column vectors in $\mathbf{X}$ are **linearly correlated** with each other.

In this case,

$$\Psi(X) = \mathbf{XC}$$

where $\mathbf{C}$ is a matrix representing **variable selection** and/or **dimensionality reduction**.

Example

Various choices for the matrix $\mathbf{C}$ in the model produce several cases:

- When $\mathbf{C} = \mathbf{I}$, the l-dimensional identity matrix, it is reduced to the full set of features $\mathbf{X}$ and a model $\mathbf{Y} = f(\mathbf{X}) + \epsilon$.
- When $\mathbf{C}$ is a subset of columns in $\mathbf{I}$, it is **reduced** to the usual sub-model with only a few columns/variables that can be **selected** by various methods.
- The **PCA method** can be considered a special case with $\mathbf{C}$ being a weight matrix consisting of a certain column combination of features in $\mathbf{X}$ that **maximizes the variation** of the data in $\mathbf{X}$.

Section 3

Conventional Statistical Approaches

Conventional Statistical Approaches

From a **statistical perspective**, a most important metric of the data is its **variability**, as captured by common measures of dispersion such as variance/standard deviation and correlations between its various features.

Preserving such variability is used as the primary criterion to search for and evaluating methods to reduce dimensionality.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is clearly the most popular DR method.

When the dimensionality I is large, PCA is commonly used in **exploratory data analysis** and for **extracting features** and developing models.

The key concept here is **principal component (PC)** of **variability**. The first principal component is the **direction** that **maximizes the variance** of a **projection** of the data onto a single line in feature space.

The second principal component can be taken as a direction **orthogonal** to the first principal component that maximizes the variance of the remaining components of the projected data,

Principal Component Analysis (PCA)

Iterating the process, PCA computes the third principal component **orthogonally** to the first two PC's.

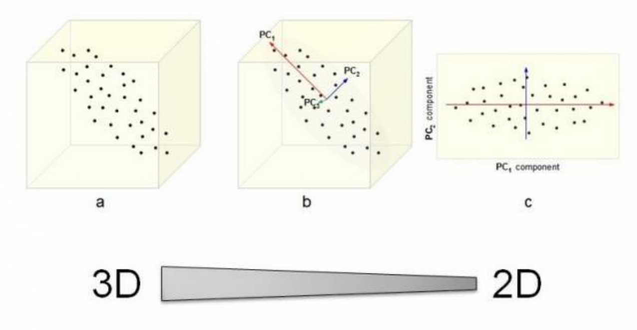
PCA reduces dimensionality by **projecting** each data point onto only the first few principal components to obtain lower dimensional data representations, while **preserving** as much of the data's variation as possible.

Overall, PCS is equivalent to a **change of basis** of the coordinate axes in feature space (e.g., by appropriate rotations) in order to capture the **most variability** of the data along the new axes.

Principal Component Analysis (PCA)

Thus, PCA extracts features that retain the most amount of the variance/covariance in the high dimension data.

Since these projections are **linear operators**, they can be easily implemented using a **singular value decomposition (SVD)** of its variance–covariance matrix with optimized and very fast **linear algebra** software libraries.

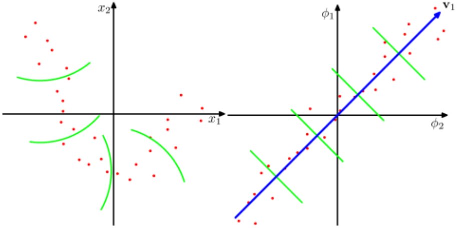


Kernel PCA (KPCA)

Kernel PCA (KPCA) is an extension of PCA obtained by choosing a transformation of the data by a so-called **kernel** that defines some **weighted distance measure**.

PCA is recovered as a particular case when the **kernel is linear**. Other kernels can be considered, such as

- Polynomial kernels.
- Gaussian kernels
- Laplacian kernels.



Independent Component Analysis (ICA)

Independent Component Analysis (ICA): Both PCA and ICA share the common feature of finding a set of vectors as a **basis** to re-code the data.

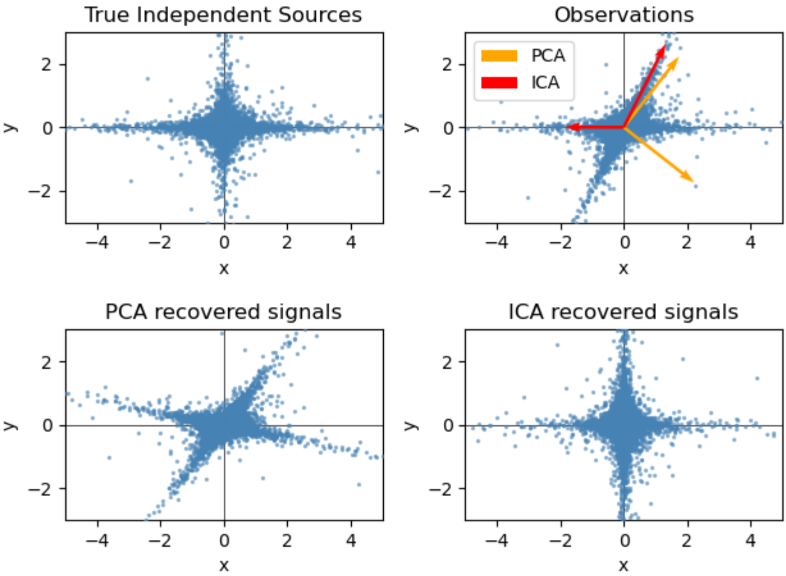
PCA can greatly **compress** the data into fewer dimensions.

On the other hand, ICA is useful to find a representation of high-dimensional data as **independent subelements** that can be used to separate data.

Therefore, ICA is useful when the data (usually image data) is a **mixture** of multiple signals for the separation of the various independent components.

ICA is used mostly for the purpose **image processing** or voice processing.

Independent Component Analysis (ICA)



Section 4

Geometric Approaches

Geometric Approaches

From a **geometric perspective**, PCA can be viewed as **fitting a linear (flat) subspace** to the data so as to **minimize the error** given by the total sum of squared distances of the data points from the subspace of reduced dimension.

In a more general setting, the data may not be flat, so a **curved surface** may best fit the data. To address these surfaces the concept of **manifolds** is used.

Roughly speaking, a **manifold** is a geometric object that appears to be flat (**locally**) like a Euclidean object (line, plane, affine hyperplane).

Geometric Approaches

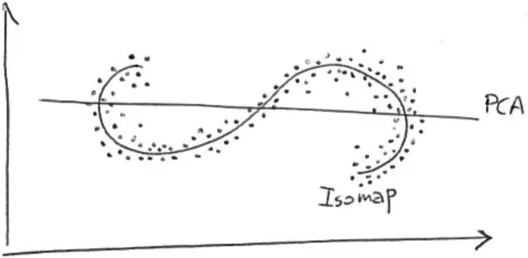
The general idea of a **geometric approach** is to reduce dimensionality while **minimizing** some appropriate **loss function** that measures discrepancies in the distances, or some other function of the distances is minimized.

The notion of distance being used defines a specific method.

Isometric Mapping (ISOMAP)

Isometric Mapping (ISOMAP) is a mapping that **preserves** a distance measure defined over a **lower dimensional manifold** that can fit the data reasonably well.

Its basic philosophy is to assume that learning this manifold is key to successful data analysis, and that the distances between the points are intrinsic or geodesic distances between the points on the manifold.

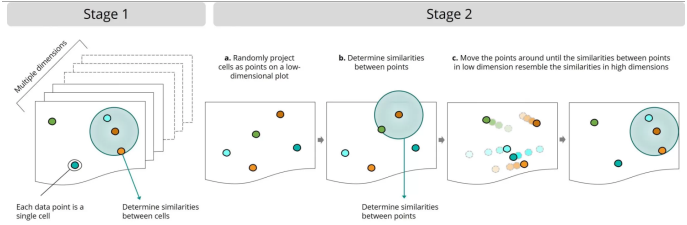


t-Stochastic Neighbor Embedding (*t*-SNE)

***t*-Stochastic Neighbor Embedding (*t*-SNE)** is considered the state of the art in **visualization algorithms**.

The *t*-SNE is a **probabilistic approach** transforming objects given by high-dimensional vectors or by pairwise dissimilarities, into a lower dimensional space in such a way that **neighbor identities** are preserved.

Unlike other dimensionality reduction methods, SNE can represent each object with a mixture of widely separated lower dimensional images.

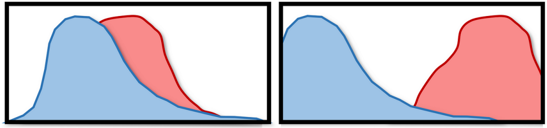

Stage 1
Stage 2
Multiple dimensions
a. Randomly project cells as points on a low-dimensional plot
b. Determine similarities between points
c. Move the points around until the similarities between points in low dimension resemble the similarities in high dimensions
Each data point is a single cell
Determine similarities between cells
Determine similarities between points

Information-Theoretic Approaches

One can also perform a dimensionality reduction procedure using Shannon's concept of **conditional entropy (CE)**.

The basic idea is to **select features/input variables** by **minimizing conditional entropy** or, equivalently, to select features/input variables by **maximizing mutual information**.

Specifically, **mutual information** measures the **reduction** in **uncertainty** for one variable when the value of another variable is known. This concept is closely related to the concept of **information gain** in machine learning, as a quantitative measure of the reduction in entropy, given the values of another random variable.



Low information gain
High entropy

High information gain
Low entropy

Summary

- **Principal Component Analysis (PCA)**: Creates key principal components capturing **maximum data variation** as combinations of **orthogonal linear input variables**.
- **Kernel PCA**: Generalizes of PCA to other possible weights for the distance variation.
- **Independent component analysis (ICA)**: Decomposes the data into several components for the purpose of **object separation** or clustering
- **Isometric Mapping (ISOMAP)**: Builds a mapping **preserving a distance measure** defined over a **lower dimensional manifold** that can fit the between-data distances reasonably well.
- ***t*-Stochastic Neighbor Embedding (*t*-SNE)**: Improves the probabilistic approach SNE transforming objects given by high-dimensional vectors or by pairwise dissimilarities, into a lower dimensional space **preserving neighbor identities**.
- **Conditional Entropy**: **Selects** features/input variables by **minimizing conditional entropy**, or equivalently, by maximizing mutual information or information gain

References

- Garzon, M., Yang, C.C., Venugopal, D., Kumar, N., Jana, K., Deng, L. *Dimensionality Reduction in Data Science*, Cap. 1, 2 & 3, Springer, 2022.