

Unsupervised Learning

Clustering and Dimension Reduction

E. Le Pennec



Fall 2017

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 Clustering
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

1 Motivation, Supervised vs Unsupervised Learning

2 Clustering

- Partition Based
- Model Based
- Density based
- Agglomerative clustering
- Others
- Scalability

3 Dimension Reduction

- Dimensionality Curse, Inertia and PCA
- Reconstruction error
- Relationship preservation

- **Marketing:** finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
- **Biology:** classification of plants and animals given their features;
- **Libraries:** book ordering;
- **Insurance:** identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
- **City-planning:** identifying groups of houses according to their house type, value and geographical location;
- **Internet:** document classification; clustering weblog data to discover groups of similar access patterns.

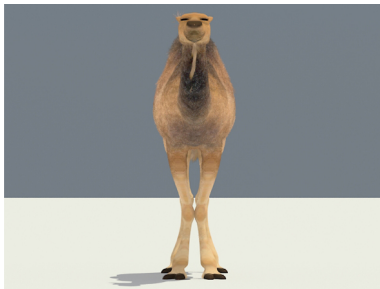


- **Data:** Base of customer data containing their properties and past buying records
- **Goal:** Use the customers *similarities* to find groups.
- **Two directions:**
 - **Clustering:** propose an explicit *grouping* of the customers
 - **Visualization:** propose a representation of the customers so that the groups are *visibles*

- How to view a high-dimensional dataset?
- High-dimension: dimension larger than 2!
- *Projection* in a 2D space.

Dimension Reduction

Motivation, Supervised
vs Unsupervised
Learning



- How to view a high-dimensional dataset?
- High-dimension: dimension larger than 2!
- *Projection* in a 2D space.

Dimension Reduction

Motivation, Supervised
vs Unsupervised
Learning



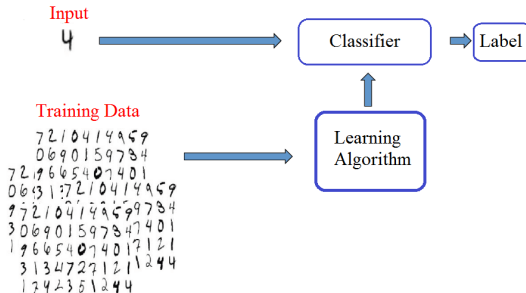
- How to view a high-dimensional dataset?
- High-dimension: dimension larger than 2!
- *Projection* in a 2D space.

Dimension Reduction

Motivation, Supervised
vs Unsupervised
Learning



- How to view a high-dimensional dataset?
- High-dimension: dimension larger than 2!
- *Projection* in a 2D space.



A definition by Tom Mitchell
(<http://www.cs.cmu.edu/~tom/>)

A computer program is said to learn from **experience E** with respect to some **class of tasks T** and **performance measure P**, if its performance at tasks in T, as measured by P, improves with experience E.

Experience, Task and Performance measure

- **Training data** : $\mathcal{D} = \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$ (i.i.d. $\sim \mathbf{P}$)
- **Predictor**: $f : \mathcal{X} \rightarrow \mathcal{Y}$ measurable
- **Cost/Loss function** : $\ell(f(\mathbf{X}), Y)$ measure how well $f(\mathbf{X})$ “predicts” Y
- **Risk**:

$$\mathcal{R}(f) = \mathbb{E} [\ell(Y, f(\mathbf{X}))] = \mathbb{E}_{\mathbf{X}} [\mathbb{E}_{Y|\mathbf{X}} [\ell(Y, f(\mathbf{X}))]]$$

- Often $\ell(f(\mathbf{X}), Y) = \|f(\mathbf{X}) - Y\|^2$ or $\ell(f(\mathbf{X}), Y) = \mathbf{1}_{Y \neq f(\mathbf{X})}$

Goal

- Learn a rule to construct a **classifier** $\hat{f} \in \mathcal{F}$ from the training data \mathcal{D}_n s.t. **the risk** $\mathcal{R}(\hat{f})$ is **small on average** or **with high probability with respect to \mathcal{D}_n** .

Experience, Task and Performance measure

- **Training data** : $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\}$ (i.i.d. $\sim \mathbf{P}$)
- **Task**: ???
- **Performance measure**: ???

- No obvious task definition!

Tasks for this lecture

- **Clustering (or unsupervised classification)**: construct a **grouping** of the data in **homogeneous** classes.
- **Dimension reduction**: construct a map of the data in a **low dimensional** space without **distorting** it too much.

- **Training data** : $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\} \in \mathcal{X}^n$ (i.i.d. $\sim \mathbf{P}$)
- Latent groups?

Clustering

- Construct a map f from \mathcal{D} to $\{1, \dots, K\}$ where K is a number of classes to be fixed:

$$f : \mathbf{X}_i \mapsto k_i$$

- Similar to classification except:
 - no ground truth (no given labels)
 - label only elements of the dataset!

Motivations

- Interpretation of the groups
- Use of the groups in further processing

- Need to define the **quality** of the cluster.
- No obvious measure! **this turns to be a great problem in reality**

Clustering quality

- Inner homogeneity: samples in the same group should be similar.
 - Outer inhomogeneity: samples in two different groups should be different.
-
- Several possible definitions of similar and different.
 - Often based on the distance between the samples.
 - Example based on the euclidean distance:
 - Inner homogeneity = intra class variance,
 - Outer inhomogeneity = inter class variance.
 - **Beware:** choice of the number of cluster K often complex!

- **Training data** : $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\} \in \mathcal{X}^n$ (i.i.d. $\sim \mathbf{P}$)
- Space \mathcal{X} of possibly high dimension.

Dimension Reduction Map

- Construct a map Φ from the space \mathcal{X} into a space \mathcal{X}' of **smaller dimension**:

$$\begin{aligned}\Phi : \mathcal{X} &\rightarrow \mathcal{X}' \\ \mathbf{X} &\mapsto \Phi(\mathbf{X})\end{aligned}$$

- Map can be defined only on the dataset.

Motivations

- Visualization of the data
- Dimension reduction before further processing

- Need to control the **distortion** between \mathcal{D} and $\Phi(\mathcal{D}) = \{\Phi(\mathbf{X}_1), \dots, \Phi(\mathbf{X}_n)\}$

Distortion(s)

- Reconstruction error:
 - Construct $\tilde{\Phi}$ from \mathcal{X}' to \mathcal{X}
 - Control the error between \mathbf{X} and its reconstruction $\tilde{\Phi}(\Phi(\mathbf{X}))$
 - Relationship preservation:
 - Compute a *relation* \mathbf{X}_i and \mathbf{X}_j and a *relation* between $\Phi(\mathbf{X}_i)$ and $\Phi(\mathbf{X}_j)$
 - Control the difference between those two *relations*.
- Leads to different constructions....

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 Clustering
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

- **Training data** : $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\} \in \mathcal{X}^n$ (i.i.d. $\sim \mathbf{P}$)
- Latent groups?

Clustering

- Construct a map f from \mathcal{D} to $\{1, \dots, K\}$ where K is a number of classes to be fixed:

$$f : \mathbf{X}_i \mapsto k_i$$

Motivations

- Interpretation of the groups
 - Use of the groups in further processing
-
- Several strategies possible!
 - Can use dimension reduction as a preprocessing.

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 Clustering
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

Partition Heuristic

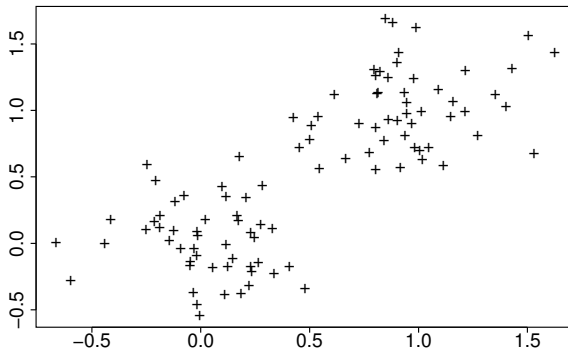
- Clustering is defined by a partition in K classes...
- that minimizes a homogeneity criterion.

K- Means

- Cluster k defined by a *center* μ_k .
- Each sample is associated to the closest center.
- Centers defined as the minimizer of
$$\sum_{i=1}^n \min_k \|\mathbf{x}_i - \mu_k\|^2$$
- Iterative scheme (Lloyd):
 - Start by a (pseudo) random choice for the centers μ_k
 - Assign each samples to its nearby center
 - Replace the center of a cluster by the mean on its assigned samples.
 - Repeat the last two steps until convergence.

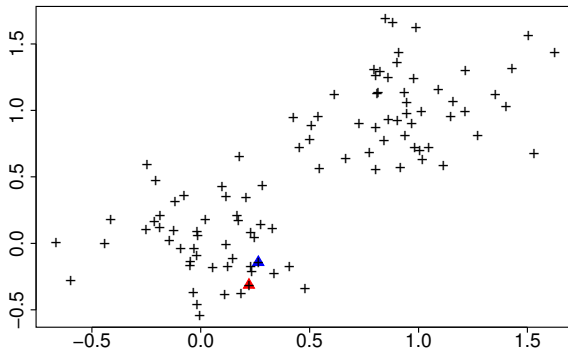
Partition based

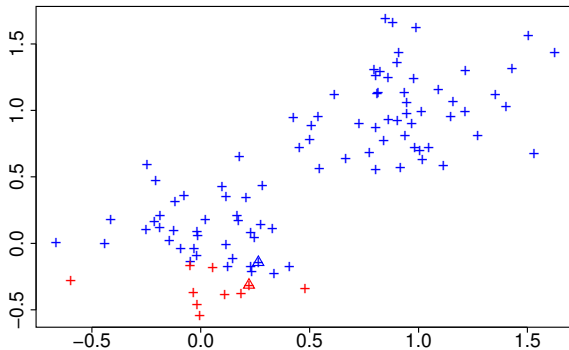
Clustering



Partition based

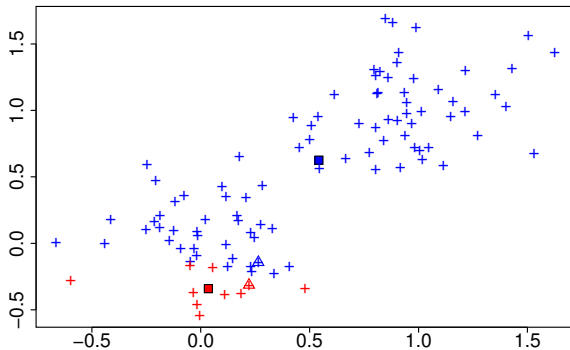
Clustering





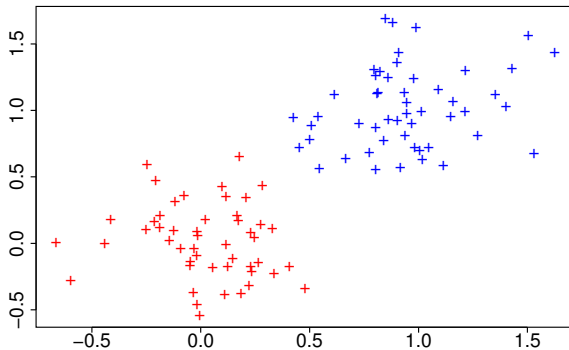
Partition based

Clustering



Partition based

Clustering



- Other schemes:
 - McQueen: modify the mean each time a sample is assigned to a new cluster.
 - Hartigan: modify the mean by removing the considered sample, assign it to the nearby center and recompute the new mean after assignment.
- A good initialization is crucial!
 - Initialize by samples.
 - k-Mean++: try to take them as separated as possible.
 - No guarantee to converge to a global optimum: repeat and keep the best result!
- Complexity : $O(n \times K \times T)$ where T is the number of step in the algorithm.

- k-Medoid: use a sample as a center
 - PAM: for a given cluster, use the sample that minimizes the intra distance (sum of the squared distance to the other points)
 - Approximate medoid: for a given cluster, assign the point that is the closest to the mean.
- Complexity:
 - PAM: $O(n^2 \times T)$ in the worst case!
 - Approximate medoid: $O(n \times K \times T)$ where T is the number of step in the algorithm.
- **Remark:** Any distance can be used...

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 **Clustering**
 - Partition Based
 - **Model Based**
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

Model Heuristic

- Use a generative model of the data:

$$\mathbb{P} \{ \mathbf{X} \} = \sum_{k=1}^K \pi_k \mathbb{P}_{\theta_k} \{ \mathbf{X} | k \}$$

where π_k are proportions and $\mathbb{P}_{\theta} \{ \mathbf{X} | k \}$ are parametric probability models.

- Estimate those parameters (often by a ML principle).
- Assign each observations to the class maximizing the a posteriori probability (obtained by Bayes formula)

$$\frac{\widehat{\pi}_k \mathbb{P}_{\widehat{\theta}_k} \{ \mathbf{X} | k \}}{\sum_{k'=1}^K \widehat{\pi}_{k'} \mathbb{P}_{\widehat{\theta}_{k'}} \{ \mathbf{X} | k' \}}$$

- Link with Generative model in supervised classification!

- Large choice of parametric models.

Gaussian Mixture Model

- Use

$$\mathbb{P}_{\theta_k} \{\mathbf{X}|k\} \sim \mathcal{N}(\mu_k, \Sigma_k)$$

with $\mathcal{N}(\mu, \Sigma)$ the Gaussian law of mean μ and covariance matrix Σ .

- Efficient optimization algorithm available (EM)
- Often some constrain on the covariance matrices: identical, with a similar structure...
- Strong connection with K -means when the covariance matrices are assumed to be the same multiple of the identity.

Probabilistic latent semantic analysis (PLSA)

- Documents described by their word counts w
- Model:

$$\mathbb{P}\{w\} = \sum_{k=1}^K \mathbb{P}\{k\} \mathbb{P}_{\theta_k}\{w|k\}$$

with k the (hidden) topic, $\mathbb{P}\{k\}$ a topic probability and $\mathbb{P}\{w|k\}$ a multinomial law for a given topic.

- Clustering according to

$$\mathbb{P}\{k|w\} = \frac{\widehat{\mathbb{P}\{k\}} \widehat{\mathbb{P}_{\theta_k}\{w|k\}}}{\sum_{k'} \widehat{\mathbb{P}\{k'\}} \widehat{\mathbb{P}_{\theta_{k'}}\{w|k'\}}}$$

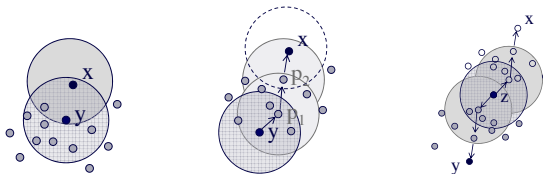
- Same idea than GMM!
- Bayesian variant called LDA.

- Framework based on density estimation principle.
- Assign a probability of membership.
- Lots of theoretical studies...
- Model selection principle can be used to select K the number of class:
 - AIC / BIC /MDL penalization
 - Cross Validation is also possible!
- Complexity: $O(n \times K \times T)$

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 **Clustering**
 - Partition Based
 - Model Based
 - **Density based**
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

Density heuristic

- Cluster are connected dense zone separated by low density zone.
 - Not all points belong to a cluster.
-
- Basic bricks:
 - Estimate the density.
 - Find points with high densities.
 - Gather those points according to the density
 - Density estimation:
 - Classical kernel density estimate...
 - Gathering:
 - Link points of high density and use the resulted component.
 - Move them toward top of density *hill* by following the gradient and gather all the points arriving at the same *summit*.



- Examples:
 - DBSCAN: link point of high densities using a very simple kernel.
 - PdfCLuster: find connected zone of high density.
 - Mean-shift: move points toward top of density *hill* following an evolving kernel density estimate.
- Complexity: $O(n^2 \times T)$ in the worst case.
- Can be reduced to $O(n \log(n) T)$ if samples can be encoded in a tree structure (n-body problem type approximation).

1 Motivation, Supervised vs Unsupervised Learning

2 Clustering

- Partition Based
- Model Based
- Density based
- **Agglomerative clustering**
- Others
- Scalability

3 Dimension Reduction

- Dimensionality Curse, Inertia and PCA
- Reconstruction error
- Relationship preservation

Agglomerative Clustering Heuristic

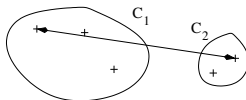
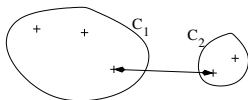
- Start with very small clusters (a sample by cluster?)
 - Sequential merging of the most similar clusters...
 - according to some *greedy* criterion Δ .
-
- Generates a hierarchy of clustering instead of a single one.
 - Need to select the number of cluster afterwards.
 - Several choice for the merging criterion...
 - Examples:
 - Minimum Linkage: merge the closest cluster in term of the usual distance
 - Ward Indice: merge the two clusters yielding the less inner inertia loss (k-means criterion)

Algorithm

- Start with $(\mathcal{C}_i^{(0)}) = (\{\mathbf{X}_i\})$ the collection of all singletons.
- At step s , we have $n - s$ clusters $(\mathcal{C}_i^{(s)})$:
 - Find the two clusters the most similar according to a criterion Δ :

$$(i, i') = \underset{(j, j')}{\operatorname{argmin}} \Delta(\mathcal{C}_j^{(s)}, \mathcal{C}_{j'}^{(s)})$$

- Merge $\mathcal{C}_i^{(s)}$ and $\mathcal{C}_{i'}^{(s)}$ into $\mathcal{C}_i^{(s+1)}$
 - Keep the $n - s - 2$ other clusters $\mathcal{C}_{i''}^{(s+1)} = \mathcal{C}_{i''}^{(s)}$
- Repeat until there is only one cluster.
- Complexity: $O(n^3)$ if no restriction on the merging possibilities.
- Can be reduced to $O(n^2)$ if only a bounded number of merging is possible for a given cluster.



Merging criterion based on the distance between points

- Minimum linkage:

$$\Delta(\mathcal{C}_i, \mathcal{C}_j) = \min_{\mathbf{x}_i \in \mathcal{C}_i} \min_{\mathbf{x}_j \in \mathcal{C}_j} d(\mathbf{x}_i, \mathbf{x}_j)$$

- Maximum linkage:

$$\Delta(\mathcal{C}_i, \mathcal{C}_j) = \max_{\mathbf{x}_i \in \mathcal{C}_i} \max_{\mathbf{x}_j \in \mathcal{C}_j} d(\mathbf{x}_i, \mathbf{x}_j)$$

- Average linkage:

$$\Delta(\mathcal{C}_i, \mathcal{C}_j) = \frac{1}{|\mathcal{C}_i||\mathcal{C}_j|} \sum_{\mathbf{x}_i \in \mathcal{C}_i} \sum_{\mathbf{x}_j \in \mathcal{C}_j} d(\mathbf{x}_i, \mathbf{x}_j)$$

- Clustering based on the proximity...

Merging criterion based on the inertia (distance to the mean)

- Ward Indice:

$$\begin{aligned}\Delta(C_i, C_j) = & \sum_{\mathbf{x}_i \in C_i} (d^2(\mathbf{x}_i, \mu_{C_i \cup C_j}) - d^2(\mathbf{x}_i, \mu_{C_i})) \\ & + \sum_{\mathbf{x}_j \in C_j} (d^2(\mathbf{x}_j, \mu_{C_i \cup C_j}) - d^2(\mathbf{x}_j, \mu_{C_j}))\end{aligned}$$

- If d is the euclidian distance:

$$\Delta(C_i, C_j) = \frac{2|C_i||C_j|}{|C_i| + |C_j|} d^2(\mu_{C_i}, \mu_{C_j})$$

- Same criterion than in the k -means algorithm but greedy optimization.

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 **Clustering**
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - **Others**
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

Grid heuristic

- Split the space in pieces
 - Group those of high density according to their proximity
-
- Similar to density based estimate (with partition based initial clustering)
 - Space splitting can be fixed or adaptive to the data.
 - Examples:
 - STING (Statistical Information Grid): Hierarchical tree construction plus DBSCAN type algorithm
 - AMR (Adaptive Mesh Refinement): Adaptive tree refinement plus k -means type assignment from high density leaves.
 - CLIQUE: Tensorial grid and 1D detection.
 - Linked to Divisive clustering (DIANA)

Graph based

- Spectral clustering: dimension reduction + k-means
 - Message passing:
 - Max Flow / Min Flow:
-
- Evolutionary algorithm,
 - ...

1 Motivation, Supervised vs Unsupervised Learning

2 Clustering

- Partition Based
- Model Based
- Density based
- Agglomerative clustering
- Others
- Scalability

3 Dimension Reduction

- Dimensionality Curse, Inertia and PCA
- Reconstruction error
- Relationship preservation

Large dataset issue

- When n is large, a $O(n^\alpha \log n)$ with $\alpha > 1$ is not acceptable!
- How to deal with such a situation?
- **Beware:** Computing all the pairwise distance requires $O(n^2)$ operations!

Ideas

- Sampling
- Online processing
- Simplification
- Parallelization

Sampling heuristic

- Use only a subsample to construct the clustering.
 - Assign the other points to the constructed clusters afterwards.
-
- Requires a clustering method that can assign new points (partition, model...)
 - Often repetition and choice of the best clustering
 - Example:
 - CLARA: K-medoid with sampling and repetition

Online heuristic

- Modify the current clusters according to the value of a single observation.
- Requires compactly described clusters.
- Examples:
 - Add to an existing cluster (and modify it) if it is close enough and create a new cluster otherwise (k -means without reassignment)
 - Stochastic descent gradient (GMM)
- May leads to far from optimal clustering.
- Often used in a sequential way:
 - Several passes
 - Two step algorithm:
 - Generate a large number n' of clusters using the online algorithm (with $n' \ll n$)
 - Cluster the clusters with a more accurate algorithm.

Simplification heuristic

- Simplify the algorithm to be more efficient at the cost of some precision.
- Algorithm dependent!
- Examples:
 - Replace groups of observation (preliminary cluster) by the (approximate) statistics.
 - Approximate the distances by cheaper ones.
 - Use n-body type techniques.

Parallelization heuristic

- Split the computation on several computers.
- Algorithm dependent!
- Examples:
 - Distance computation in k -means, parameter gradient in model based clustering
 - Grid density estimation, Space splitting strategies
- Classical batch sampling not easy to perform as partitions are not easily merged...

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 Clustering
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

- **Training data** : $\mathcal{D} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\} \in \mathcal{X}^n$ (i.i.d. $\sim \mathbf{P}$)
- Space \mathcal{X} of possibly high dimension.

Dimension Reduction Map

- Construct a map Φ from the space \mathcal{X} into a space \mathcal{X}' of **smaller dimension**:

$$\begin{aligned}\Phi : \mathcal{X} &\rightarrow \mathcal{X}' \\ \mathbf{X} &\mapsto \Phi(\mathbf{X})\end{aligned}$$

Criterion

- Reconstruction error
- Relationship preservation

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 Clustering
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

High Dimension Geometry Curse

- Folks theorem: In high dimension, everyone is alone.
- Theorem: If $\mathbf{X}_1, \dots, \mathbf{X}_n$ in the hypercube of dimension d such that their coordinates are i.i.d then

$$d^{-1/p} (\max \|\mathbf{X}_i - \mathbf{X}_j\|_p - \min \|\mathbf{X}_i - \mathbf{X}_j\|_p) = 0 + O\left(\sqrt{\frac{\log n}{d}}\right)$$
$$\frac{\max \|\mathbf{X}_i - \mathbf{X}_j\|_p}{\min \|\mathbf{X}_i - \mathbf{X}_j\|_p} = 1 + O\left(\sqrt{\frac{\log n}{d}}\right).$$

- When d is large, all the points are almost equidistant...
- Nearest neighbors are meaningless!

- $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbf{R}^d$
- $m = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$

Two views on inertia

- Inertia:

$$\begin{aligned} I &= \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - m\|^2 \\ &= \frac{1}{2n^2} \sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 \end{aligned}$$

- 2 times the mean squared distance to the mean = Mean squared distance between individual
- Heuristic: a good representation is a representation with a large inertia
- Large dispersion \sim Large average separation!

- What if we replace \mathbf{X} by its projection $\widetilde{\mathbf{X}} = P(\mathbf{X} - m) + m$?

Two views on inertia

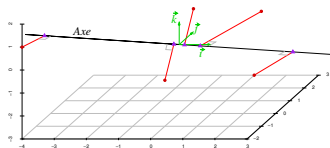
- Inertia:

$$\begin{aligned}\tilde{l} &= \frac{1}{n} \sum_{i=1}^n \|\widetilde{\mathbf{x}}_i - m\|^2 \\ &= \frac{1}{2n^2} \sum_{i,j} \|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\|^2\end{aligned}$$

- Inertia:

$$\begin{aligned}\tilde{l} &= l - \frac{1}{n} \sum_{i=1}^n \|\widetilde{\mathbf{x}}_i - \mathbf{x}_i\|^2 \\ &= l - \frac{1}{2n^2} \sum_{i,j} \left(\|\widetilde{\mathbf{x}}_i - \widetilde{\mathbf{x}}_j\|^2 - \|\mathbf{x}_i - \mathbf{x}_j\|^2 \right)\end{aligned}$$

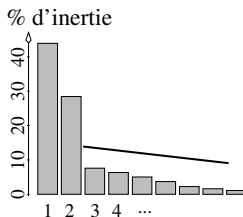
- Four different way to obtain a large inertia!



- 1D case: $\tilde{\mathbf{X}} = m + a^t(\mathbf{X} - m)a$ with $\|a\| = 1$
- Inertia: $\tilde{I} = \frac{1}{n} \sum_{i=1}^n a^t(\mathbf{X}_i - m)(\mathbf{X}_i - m)^t a$

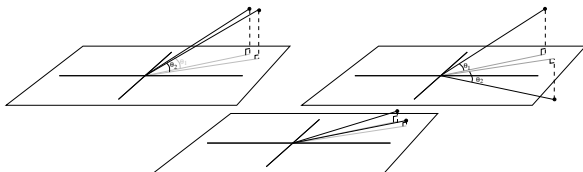
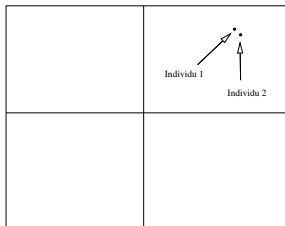
Principal Component Analysis : optimization of the projection

- Maximization of $\tilde{I} = \frac{1}{n} \sum_{i=1}^n a^t(\mathbf{X}_i - m)(\mathbf{X}_i - m)^t a = a^t \Sigma a$ with $\Sigma = \frac{1}{n} \sum_{i=1}^n (\mathbf{X}_i - m)(\mathbf{X}_i - m)^t$ the empirical covariance matrix.
- Explicit optimal choice given by the eigenvector of the largest eigenvalue of Σ .



Principal Component Analysis : optimization of the projection

- Explicit optimal solution obtain by the projection on the eigenvectors of the largest eigenvalues of Σ .
- Projected inertia given by the sum of those eigenvalues.
- Often fast decay of the eigenvalues: some dimensions are much more important than other.
- Not exactly the curse of dimensionality setting...
- Yet a lot of *small* dimension can drive the distance!



Close projection doesn't mean close individuals!

- Same projections but different situations.
- Quality of the projection measured by the angle!

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 Clustering
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - **Reconstruction error**
 - Relationship preservation

Goal

- Construct a map Φ from the space \mathcal{X} into a space \mathcal{X}' of **smaller dimension**:

$$\Phi : \mathcal{X} \rightarrow \mathcal{X}'$$

$$\mathbf{X} \mapsto \Phi(\mathbf{X})$$

- Construct $\tilde{\Phi}$ from \mathcal{X}' to \mathcal{X}
 - Control the error between \mathbf{X} and its reconstruction $\tilde{\Phi}(\Phi(\mathbf{X}))$
- Canonical example for $\mathbf{X} \in \mathbb{R}^d$: find Φ and $\tilde{\Phi}$ in a parametric family that minimize

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \tilde{\Phi}(\Phi(\mathbf{x}_i))\|^2$$

- $\mathcal{X} \in \mathbb{R}^d$ and $\mathcal{X}' = \mathbb{R}^{d'}$
- Affine model $\mathbf{X} \sim m + \sum_{l=1}^{d'} \mathbf{X}'^{(l)} V^{(l)}$ with $(V^{(l)})$ an orthonormal family.

- Equivalent to:

$$\Phi(\mathbf{X}) = V^t(\mathbf{X} - m) \quad \text{and} \quad \tilde{\Phi}(\mathbf{X}') = m + V\mathbf{X}'$$

- Reconstruction error criterion:

$$\frac{1}{n} \sum_{i=1}^n \|\mathbf{X}_i - (m + VV^t(\mathbf{X}_i - m))\|^2$$

- **Explicit solution:** m is the empirical mean and V is any orthonormal basis of the space spanned by the d' first eigenvectors (the one with largest eigenvalues) of the empirical covariance matrix $\frac{1}{n} \sum_{i=1}^n (\mathbf{X}_i - m)(\mathbf{X}_i - m)^t$.

PCA Algorithm

- Compute the empirical mean $m = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i$
 - Compute the empirical covariance matrix $\frac{1}{n} \sum_{i=1}^n (\mathbf{X}_i - m)(\mathbf{X}_i - m)^t$.
 - Compute the d' first eigenvectors of this matrix: $V^{(1)}, \dots, V^{(d')}$
 - Set $\Phi(\mathbf{X}) = V^t(\mathbf{X} - m)$
-
- Complexity: $O(n(d + d^2) + d'd^2)$
 - Interpretation:
 - $\Phi(\mathbf{X}) = V^t(\mathbf{X} - m)$: coordinates in the restricted space.
 - $V^{(i)}$: influence of each original coordinates in the i th new one.
 - **Scaling:** This method is not invariant to a scaling of the variables! It is custom to normalize the variables (at least within groups) before applying PCA.

- PCA assumes $\mathcal{X} = \mathbb{R}^d$!
- How to deal with categorical values?
- MFA = PCA with clever coding strategy for categorical values.

Categorical value code for a single variable

- Classical redundant dummy coding:

$$\mathbf{X} \in \{1, \dots, V\} \mapsto P(\mathbf{X}) = (\mathbf{1}_{\mathbf{X}=1}, \dots, \mathbf{1}_{\mathbf{X}=V})^t$$

- Compute the mean (i.e. the empirical proportion) $\bar{P} = \frac{1}{n}P(\mathbf{X})$

- Renormalize $P(\mathbf{X})$ by $1/\sqrt{(V-1)\bar{P}}$:

$$P(\mathbf{X}) \mapsto P^r(\mathbf{X})$$

$$(\mathbf{1}_{\mathbf{X}=1}, \dots, \mathbf{1}_{\mathbf{X}=V}) \mapsto \left(\frac{\mathbf{1}_{\mathbf{X}=1}}{\sqrt{(V-1)\bar{P}_1}}, \dots, \frac{\mathbf{1}_{\mathbf{X}=V}}{\sqrt{(V-1)\bar{P}_V}} \right)$$

- χ^2 type distance!

- PCA becomes the minimization of

$$\frac{1}{n} \sum_{i=1}^n \|P^r(\mathbf{X}_i) - (m + VV^t(P^r(\mathbf{X}_i) - m))\|^2$$
$$= \frac{1}{n} \sum_{i=1}^n \sum_{v=1}^V \frac{\left| \mathbf{1}_{\mathbf{X}_i=v} - (m' + \sum_{l=1}^{d'} V^{(l)t}(P(\mathbf{X}_i) - m')V^{(l,v)}) \right|^2}{(V-1)\bar{P}_v}$$

- Interpretation:
 - $m' = \bar{P}$
 - $\Phi(\mathbf{X}) = V^t(P^r\mathbf{X} - m)$: coordinates in the restricted space.
 - $V^{(l)}$ can be interpreted as a probability profile.
- Complexity: $O(n(V + V^2) + d'V^2)$
- Link with Correspondence Analysis (CA)

MFA Algorithm

- Redundant dummy coding of each categorical variable.
 - Renormalization of each block of dummy variable.
 - Classical PCA algorithm on the resulting variables
-
- Interpretation as a reconstruction error with a rescaled/ χ^2 metric.
 - Interpretation:
 - $\Phi(\mathbf{X}) = V^t(P^r(\mathbf{X}) - m)$: coordinates in the restricted space.
 - $V^{(l)}$: influence of each modality/variable in the i th new coordinates.
 - **Scaling:** This method is not invariant to a scaling of the continuous variables! It is custom to normalize the variables (at least within groups) before applying PCA.

PCA Model

- PCA: Linear model assumption

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'^{(l)} V^{(l)} = m + V \mathbf{X}'$$

- with
 - $V^{(l)}$ orthonormal
 - $\mathbf{X}'^{(l)}$ without constraints.
- Two directions of extension:
 - Other constraints on V (or the coordinates in the restricted space): ICA, NMF, Dictionary approach
 - PCA on a non linear image of \mathbf{X} : kernel-PCA
- Much more complex algorithm!

ICA (Independent Component Analysis)

- Linear model assumption

$$\mathbf{X} \simeq \mathbf{m} + \sum_{l=1}^{d'} \mathbf{X}'^{(l)} \mathbf{V}^{(l)} = \mathbf{m} + \mathbf{V} \mathbf{X}'$$

- with
 - $\mathbf{V}^{(l)}$ without constraints.
 - $\mathbf{X}'^{(l)}$ independent

NMF (Non Negative Matrix Factorization)

- (Linear) Model assumption

$$\mathbf{X} \simeq \mathbf{m} + \sum_{l=1}^{d'} \mathbf{X}'^{(l)} \mathbf{V}^{(l)} = \mathbf{m} + \mathbf{V} \mathbf{X}'$$

- with
 - $\mathbf{V}^{(l)}$ non negative
 - $\mathbf{X}'^{(l)}$ non negative.

Dictionary

- (Linear) Model assumption

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'^{(l)} V^{(l)} = m + V\mathbf{X}'$$

- with
 - $V^{(l)}$ without constrains
 - \mathbf{X}' sparse (with a lot of 0)

kernel PCA

- Linear model assumption

$$\Psi(\mathbf{X} - m) \simeq \sum_{l=1}^{d'} \mathbf{X}'^{(l)} V^{(l)} = V\mathbf{X}'$$

- with
 - $V^{(l)}$ orthonormal
 - \mathbf{X}'_l without constrains.

- Linear model assumption:

$$\mathbf{X} \simeq m + \sum_{l=1}^{d'} \mathbf{X}'^{(l)} V^{(l)} = m + V \mathbf{X}'$$

- Vector rewriting

$$\mathbf{X}^t \simeq m^t + \mathbf{X}'^t V^t$$

Matrix Rewriting and Low Rank Factorization

- Matrix rewriting

$$\begin{array}{|c|} \hline \mathbf{X}_1^t - m^t \\ \vdots \\ \vdots \\ \mathbf{X}_n^t - m^t \\ \hline \end{array} \quad \simeq \quad \begin{array}{|c|} \hline \mathbf{X}_1'^t \\ \vdots \\ \vdots \\ \mathbf{X}_n'^t \\ \hline \end{array} \quad \begin{array}{|c|} \hline \mathbf{V}^t \\ \hline \end{array}$$

$(n \times d) \qquad (n \times d') \qquad (d' \times d)$

- Low rank matrix factorization! (Truncated SVD solution...)

SVD Decomposition

- Any matrix $n \times d$ matrix A can be decomposed as

$$\begin{array}{c} \boxed{\mathbf{A}} \\ (n \times d) \end{array} = \begin{array}{c} \boxed{\mathbf{U}} \\ (n \times n) \end{array} \begin{array}{c} \boxed{\Sigma} \\ (n \times d) \end{array} \begin{array}{c} \boxed{\mathbf{V}^t} \\ (d \times d) \end{array}$$

with U and V two orthonormal matrices and Σ a *diagonal* matrix with decreasing values.

Low Rank Approximation

- The best low rank approximation or rank r is obtained by restriction of the matrices to the first r dimensions:

$$\begin{array}{ccc}
 \boxed{\mathbf{A}} & \simeq & \boxed{\mathbf{U}_r} \quad \boxed{\Sigma_{r,r}} \quad \boxed{\mathbf{V}_r^t} \\
 (n \times d) & & (n \times r) \quad (r \times r) \quad (r \times d)
 \end{array}$$

for both the operator norm and the Frobenius norm!

- PCA: Frobenius norm, $d' = r$ and

$$\begin{pmatrix} \mathbf{x}_1^t - m^t \\ \vdots \\ \vdots \\ \mathbf{x}_n^t - m^t \end{pmatrix} \leftrightarrow A, \quad \begin{pmatrix} \mathbf{x}_1'^t \\ \vdots \\ \vdots \\ \mathbf{x}_n'^t \end{pmatrix} \leftrightarrow \mathbf{U}_r \Sigma_{r,r}, \quad \mathbf{v}^t \leftrightarrow \mathbf{v}_r^t$$

- 1 Motivation, Supervised vs Unsupervised Learning
- 2 Clustering
 - Partition Based
 - Model Based
 - Density based
 - Agglomerative clustering
 - Others
 - Scalability
- 3 Dimension Reduction
 - Dimensionality Curse, Inertia and PCA
 - Reconstruction error
 - Relationship preservation

- Different point of view!
- Focus on pairwise relation $\mathcal{R}(\mathbf{X}_i, \mathbf{X}_j)$.

Distance Preservation

- Construct a map Φ from the space \mathcal{X} into a space \mathcal{X}' of **smaller dimension**:

$$\Phi : \mathcal{X} \rightarrow \mathcal{X}'$$

$$\mathbf{X} \mapsto \Phi(\mathbf{X}) = \mathbf{X}'$$

- such that

$$\mathcal{R}(\mathbf{X}_i, \mathbf{X}_j) \sim \mathcal{R}'(\mathbf{X}'_i, \mathbf{X}'_j)$$

- Most classical version (MDS):
 - Scalar product relation: $\mathcal{R}(\mathbf{X}_i, \mathbf{X}_j) = (\mathbf{X}_i - m)^t(\mathbf{X}_j - m)$
 - Linear mapping $\mathbf{X}' = \Phi(\mathbf{X}) = V^t(\mathbf{X} - m)$.
 - Euclidean scalar product matching:

$$\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |(\mathbf{X}_i - m)^t(\mathbf{X}_j - m) - (\mathbf{X}'_i)^t \mathbf{X}'_j|^2$$

- Φ often defined only on \mathbf{D} ...

MDS Heuristic

- Match the *scalar* products:

$$\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |(\mathbf{X}_i - m)^t (\mathbf{X}_j - m) - \mathbf{X}_i'^t \mathbf{X}_j'|^2$$

- Linear method: $\mathbf{X}' = U^t(\mathbf{X} - m)$ with U orthonormal

- **Beware:** \mathbf{X} is unknown!
- Resulting criterion: minimization in $U^t(\mathbf{X}_i - m)$ of

$$\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |(\mathbf{X}_i - m)^t (\mathbf{X}_j - m) - (\mathbf{X}_i - m)^t U U^t (\mathbf{X}_j - m)|^2$$

without knowing explicitly \mathbf{X} ...

- Explicit solution obtained through the eigendecomposition of the known Gram matrix $(\mathbf{X}_i - m)^t (\mathbf{X}_j - m)$ by keeping only the d' largest eigenvalues.

- In this case, MDS yields the same result than the PCA (but with different inputs, distance between observation vs correlations)!
- **Explanation:** Same SVD problem up to a transposition:

- MDS

$$\bar{\mathbf{X}}_{(n)}^t \bar{\mathbf{X}}_{(n)} \sim \bar{\mathbf{X}}_{(n)}^t U U^t \bar{\mathbf{X}}_{(n)}$$

- PCA

$$\bar{\mathbf{X}}_{(n)} \bar{\mathbf{X}}_{(n)}^t \sim U^t \bar{\mathbf{X}}_{(n)} \bar{\mathbf{X}}_{(n)}^t U$$

- Complexity: PCA $O((n + d')d^2)$ vs MDS $O((d + d')n^2)$...

- Preserving the scalar products amounts to preserve the euclidean distance.
- Easier **generalization** if we work in term of distance!

Generalized MDS

- Generalized MDS:
 - Distance relation: $\mathcal{R}(\mathbf{X}_i, \mathbf{X}_j) = d(\mathbf{X}_i, \mathbf{X}_j)$
 - Linear mapping $\mathbf{X}' = \Phi(\mathbf{X}) = V^t(\mathbf{X} - m)$.
 - Euclidean matching:

$$\frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n |d(\mathbf{X}_i, \mathbf{X}_j) - d'(\mathbf{X}'_i, \mathbf{X}'_j)|^2$$

- Strong connection (but no equivalence) with MDS when $d(x, y) = \|x - y\|^2$!
- **Minimization:** Simple gradient descent can be used (can be stuck in local minima).

- MDS: equivalent to PCA (but more expensive) if $d(x, y) = \|x - y\|^2$!
- ISOMAP: use a *localized* distance instead to limit the influence of very far point.

ISOMAP

- For each point \mathbf{X}_i , define a neighborhood \mathcal{N}_i (either by a distance or a number of points) and let

$$d_0(\mathbf{X}_i, \mathbf{X}_j) = \begin{cases} +\infty & \text{if } \mathbf{X}_j \notin \mathcal{N}_i \\ \|\mathbf{X}_i - \mathbf{X}_j\|^2 & \text{otherwise} \end{cases}$$

- Compute the shortest path distance for each pair.
- Use the MDS algorithm with this distance

Random Projection Heuristic

- Draw at random d' unit vector (direction) U_i .
- Use $\mathbf{X}' = U^t(\mathbf{X} - m)$ with $m = \frac{1}{n} \sum_{i=1}^n \mathbf{X}_i$
- **Property:** If \mathbf{X} lives in a space of dimension d'' , then, as soon as, $d' \sim d'' \log(d'')$,
$$\|\mathbf{X}_i - \mathbf{X}_j\|^2 \sim \frac{d}{d'} \|\mathbf{X}'_i - \mathbf{X}'_j\|^2$$
- Do not really use the data!

LLE Heuristic

- For each point \mathbf{X}_i , define a neighborhood \mathcal{N}_i (either by a distance or a number of points).

- Compute some weights $W_{i,j}$ such that

$$W_{i,j} = 0 \quad \text{if } \mathbf{X}_j \notin \mathcal{N}_i$$

$$\mathbf{x}_i \sim \sum_j W_{i,j} \mathbf{x}_j$$

- Find some \mathbf{x}'_i in a space \mathcal{X}' of **smaller dimension** such that

$$\mathbf{x}'_i \sim \sum_j W_{i,j} \mathbf{x}'_j$$

- LLE: use a least square metric for the fits.

SNE heuristic

- From $\mathbf{X}_i \in \mathcal{X}$, construct a set of conditional probability:

$$P_{j|i} = \frac{e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2}}{\sum_{k \neq i} e^{-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2}} \quad P_{i|i} = 0$$

- Find \mathbf{X}'_i in $\mathbb{R}^{d'}$ such that the set of conditional probability:

$$Q_{j|i} = \frac{e^{-\|\mathbf{x}'_i - \mathbf{x}'_j\|^2 / 2\sigma_i^2}}{\sum_{k \neq i} e^{-\|\mathbf{x}'_i - \mathbf{x}'_k\|^2 / 2\sigma_i^2}} \quad Q_{i|i} = 0$$

is close from P .

- t-SNE:** use a Student-t term $(1 + \|\mathbf{x}'_i - \mathbf{x}'_j\|^2)^{-1}$ for \mathbf{x}'_i
- Minimize the Kullback-Leibler divergence $(\sum_{i,j} P_{j|i} \log \frac{P_{j|i}}{Q_{j|i}})$ by a simple gradient descent (can be stuck in local minima).
- Parameters σ_i such that $H(P_i) = -\sum_{j=1}^n P_{j|i} \log P_{j|i} = \text{cst.}$

Graph heuristic

- Construct a graph with weighted edges $w_{i,j}$ measuring the *proximity* of \mathbf{X}_i and \mathbf{X}_j ($w_{i,j}$ large if close and 0 if there is no information).

- Find the points $\mathbf{X}'_i \in \mathbb{R}^{d'}$ minimizing

$$\frac{1}{n} \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^n w_{i,j} \|\mathbf{x}'_i - \mathbf{x}'_j\|^2$$

- Need of a constraint on the size of \mathbf{X}'_i ...
- Explicit solution through linear algebra: d' eigenvectors with smallest eigenvalues of the Laplacian of the graph $D - W$, where D is a diagonal matrix with $D_{i,i} = \sum_j w_{i,j}$.
- Variation on the definition of the Laplacian...



F. Husson, S. Le and J. Pags (2017)

Exploratory Multivariate Analysis by Example Using R, Second Edition

Chapman and Hall/CRC



Ch. Aggarwal and Ch. Reddy (2013)

Data Clustering: Algorithms and Applications

Chapman and Hall/CRC



Ch. Hennig, M. Meila, F. Murtagh and R. Rocci (2015)

Handbook of Cluster Analysis

Chapman and Hall/CRC