Course 3: Unsupervised Learning

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

2018-11-05

Course 3: Unsupervised Learning

Course 3: Unsupervised Learning

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

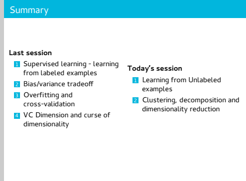
Summary

Last session

1. Supervised learning - learning from labeled examples
2. Bias/variance tradeoff
3. Overfitting and cross-validation
4. VC Dimension and curse of dimensionality

Today's session

1. Learning from Unlabeled examples
2. Clustering, decomposition and dimensionality reduction



Summary

Last session

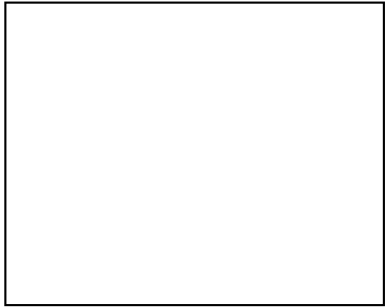
1. Supervised learning - learning from labeled examples
2. Bias/variance tradeoff
3. Overfitting and cross-validation
4. VC Dimension and curse of dimensionality

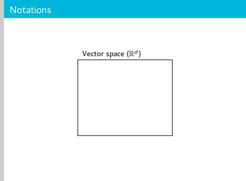
Today's session

1. Learning from Unlabeled examples
2. Clustering, decomposition and dimensionality reduction

Notations

Vector space ($\mathbb{R}^d$)

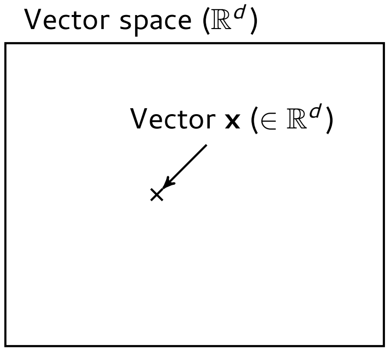


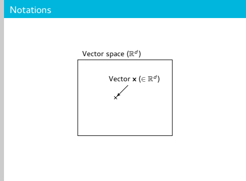


Notations

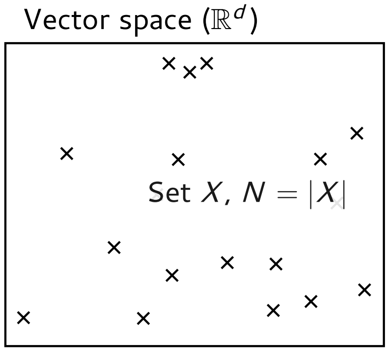
Vector space ($\mathbb{R}^d$)

Notations


Vector space ($\mathbb{R}^d$)
Vector $\mathbf{x}$ ($\in \mathbb{R}^d$)


Notations
Vector space ($\mathbb{R}^d$)
Vector $\mathbf{x}$ ($\in \mathbb{R}^d$)

Notations



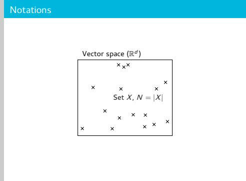
Vector space ($\mathbb{R}^d$)

Set X, $N = |X|$

2018-11-05

Course 3: Unsupervised Learning

└─Notations



Notations

Vector space ($\mathbb{R}^d$)

Set X, $N = |X|$

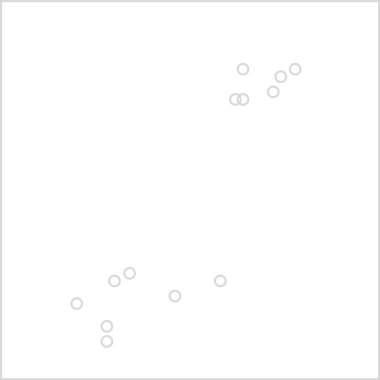
Unsupervised learning

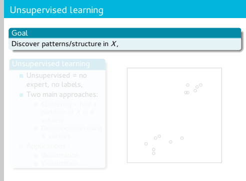
Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization. . .





Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:

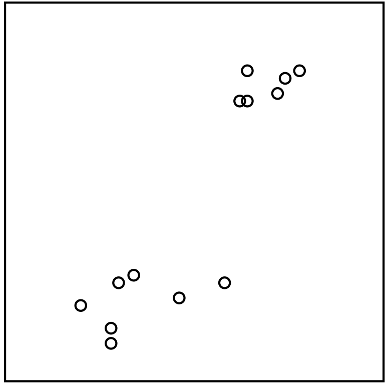
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

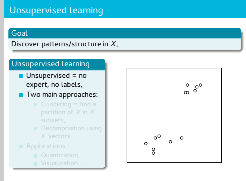
- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization...



2018-11-05

Course 3: Unsupervised Learning

└Unsupervised learning



Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization...

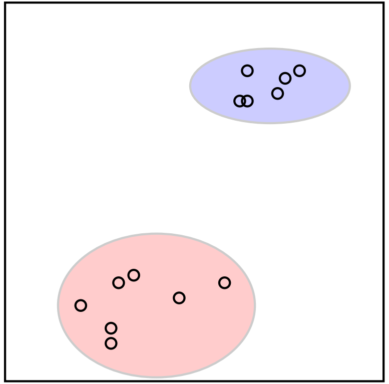
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

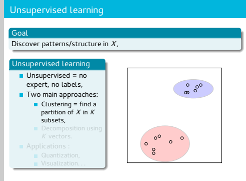
- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization. . .



2018-11-05

Course 3: Unsupervised Learning

└Unsupervised learning



Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization. . .

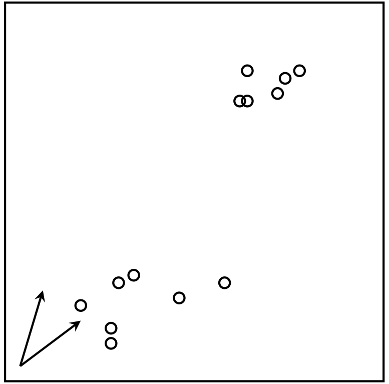
Unsupervised learning

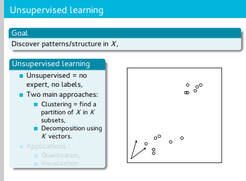
Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization...





Unsupervised learning
Goal
Discover patterns/structure in X,
Unsupervised learning
Unsupervised = no expert, no labels,
Two main approaches:
Clustering = find a partition of X in K subsets,
Decomposition using K vectors.
Applications :
Quantization,
Visualization...

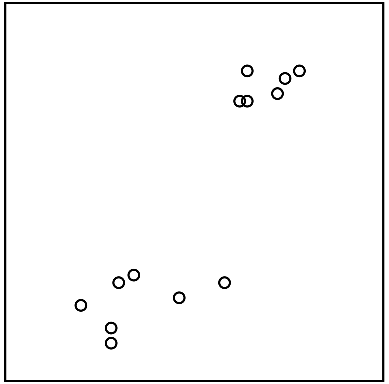
Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

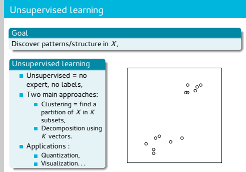
- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization. . .



2018-11-05

Course 3: Unsupervised Learning

└Unsupervised learning



Unsupervised learning

Goal

Discover patterns/structure in X,

Unsupervised learning

- Unsupervised = no expert, no labels,
- Two main approaches:
  - Clustering = find a partition of X in K subsets,
  - Decomposition using K vectors.
- Applications :
  - Quantization,
  - Visualization. . .

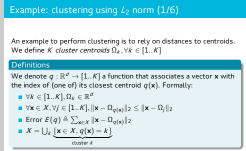
Example: clustering using L_2 norm (1/6)

An example to perform clustering is to rely on distances to centroids. We define K *cluster centroids* $\Omega_k, \forall k \in [1..K]$

Definitions

We denote $q : \mathbb{R}^d \to [1..K]$ a function that associates a vector $\mathbf{x}$ with the index of (one of) its closest centroid $q(\mathbf{x})$. Formally:

- $\forall k \in [1..K], \Omega_k \in \mathbb{R}^d$
- $\forall \mathbf{x} \in X, \forall j \in [1..K], \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2 \leq \|\mathbf{x} - \Omega_j\|_2$
- Error $E(q) \triangleq \sum_{\mathbf{x} \in X} \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2$
- $X = \bigcup_k \underbrace{\{\mathbf{x} \in X, q(\mathbf{x}) = k\}}_{\text{cluster } k}$



Example: clustering using L_2 norm (1/6)

An example to perform clustering is to rely on distances to centroids. We define K *cluster centroids* $\Omega_k, \forall k \in [1..K]$

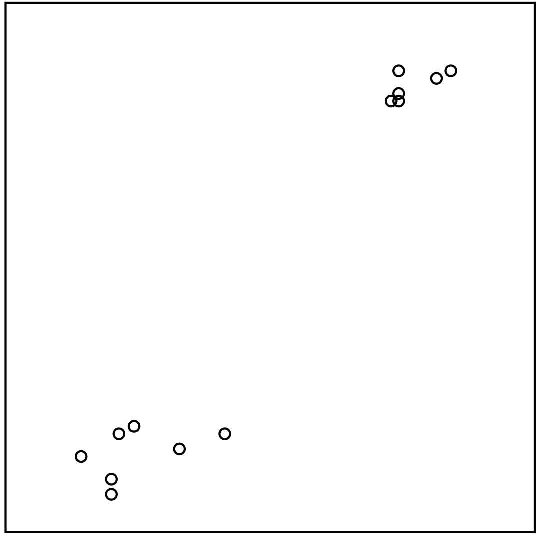
Definitions

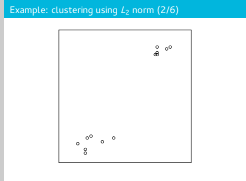
We denote $q : \mathbb{R}^d \to [1..K]$ a function that associates a vector $\mathbf{x}$ with the index of (one of) its closest centroid $q(\mathbf{x})$. Formally:

- $\forall k \in [1..K], \Omega_k \in \mathbb{R}^d$
- $\forall \mathbf{x} \in X, \forall j \in [1..K], \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2 \leq \|\mathbf{x} - \Omega_j\|_2$
- Error $E(q) \triangleq \sum_{\mathbf{x} \in X} \|\mathbf{x} - \Omega_{q(\mathbf{x})}\|_2$
- $X = \bigcup_k \underbrace{\{\mathbf{x} \in X, q(\mathbf{x}) = k\}}_{\text{cluster } k}$

Here, we provide a formal definition of clustering using centroids.
Note that there are other ways to define clustering, using regions, using density of spaces, using probabilities, etc...
The second point is the way to define the closest centroid.
The important point to note here is the definition of the error, which can be defined as the sum of all distances between points and their closest cluster centroid.

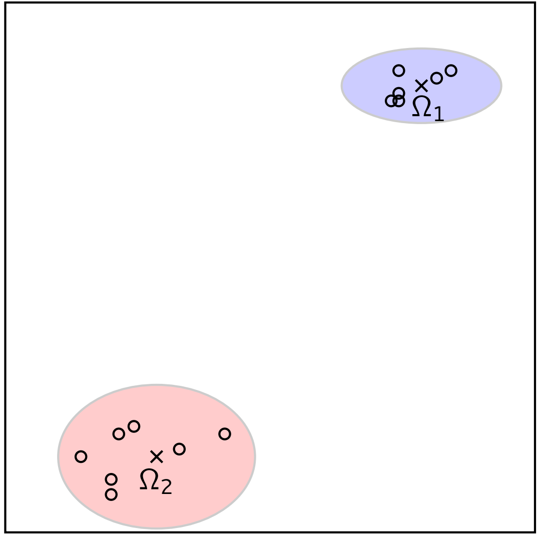
Example: clustering using L_2 norm (2/6)

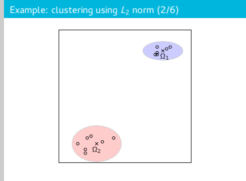




Here is a visual example. If we have the following set of points, then the following two centroids Ω_1 and Ω_2 would be reasonable candidates for a clustering with two clusters.

Example: clustering using L_2 norm (2/6)




Example: clustering using L_2 norm (2/6)

Here is a visual example. If we have the following set of points, then the following two centroids Ω_1 and Ω_2 would be reasonable candidates for a clustering with two clusters.

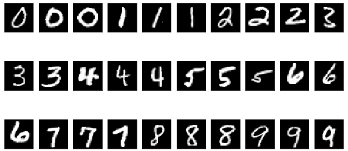
Clustering using L_2 norm (3/6)

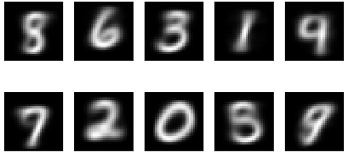
MNIST Dataset

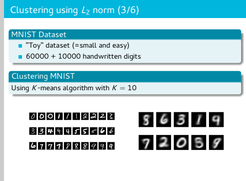
- "Toy" dataset (=small and easy)
- 60000 + 10000 handwritten digits

Clustering MNIST

Using K-means algorithm with $K = 10$







Clustering using L_2 norm (3/6)

MNIST Dataset

- "Toy" dataset (=small and easy)
- 60000 + 10000 handwritten digits

Clustering MNIST

Using K-means algorithm with $K = 10$

Let’s look at an example that looks a little bit more like real data. The MNIST dataset is small dataset of handwritten digits. It used to be an important benchmark, but it is considered too easy today to be a serious machine learning benchmark, so that is why we say it is a "toy" dataset.

MNIST is composed of 60000 examples of digits that are used for training, and 10000 that are used for test.

We can do a simple clustering test on this dataset, by using the K-Means algorithm.

Briefly, the K-means algorithm iterates between (a) assigning each point to a cluster by considering the distance to centroids, and (b) calculating the centroids for the next iteration by computing the average in each cluster. Centroid clusters can be initiliazed randomly.

The K-means algorithm stops when a certain criterion is met (number of iterations, or difference between iterations is small enough).

See here `https://upload.wikimedia.org/wikipedia/commons/f/fb/K-means.png` (picture is nice) or `https://en.wikipedia.org/wiki/K-means_clustering`

Maybe a very quick explanation of Kmeans on the board is good if the time enables it.

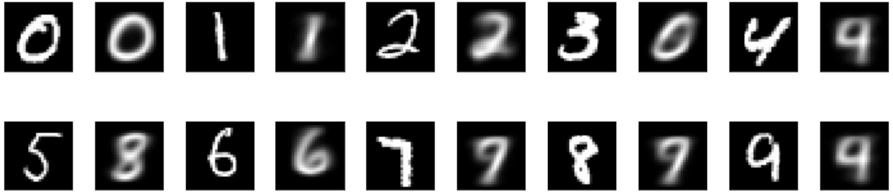
The bottom left figure represent original examples of MNIST. The bottom right figure shows the obtained cluster centroids with Kmeans. We can comment that some of the clusters seem to capture one digit (6, 1, 2, 0), but that other digits can correspond to several clusters (8, 4, 3).

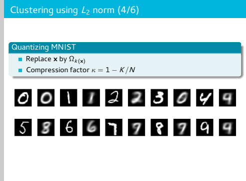
The next figure will illustrate this more precisely.

Clustering using L_2 norm (4/6)

Quantizing MNIST

- Replace $\mathbf{x}$ by $\Omega_{k(\mathbf{x})}$
- Compression factor $\kappa = 1 - K/N$





Clustering using L_2 norm (4/6)
Quantizing MNIST
- Replace $\mathbf{x}$ by $\Omega_{k(\mathbf{x})}$
- Compression factor $\kappa = 1 - K/N$

We have chosen here a random example of each digit, and we show the closest cluster centroid. We see that there are issues with 3, 4, 5, 7 and 8, even though we have tried to find 10 clusters.
In the top part of the slide, we also explain that we can actually use Clustering for compression; we just have to store the centroids, and the cluster label.

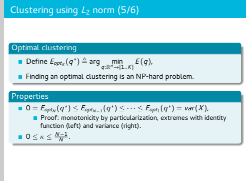
Clustering using L_2 norm (5/6)

Optimal clustering

- Define $E_{opt_K}(q^*) \triangleq \arg \min_{q:\mathbb{R}^d \rightarrow [1..K]} E(q)$,
- Finding an optimal clustering is an NP-hard problem.

Properties

- $0 = E_{opt_N}(q^*) \leq E_{opt_{N-1}}(q^*) \leq \dots \leq E_{opt_1}(q^*) = var(X)$,
  - Proof: monotonicity by particularization, extremes with identity function (left) and variance (right).
- $0 \leq \kappa \leq \frac{N-1}{N}$.



Clustering using L_2 norm (5/6)

Optimal clustering

- Define $E_{opt_K}(q^*) \triangleq \arg \min_{q:\mathbb{R}^d \rightarrow [1..K]} E(q)$,
- Finding an optimal clustering is an NP-hard problem.

Properties

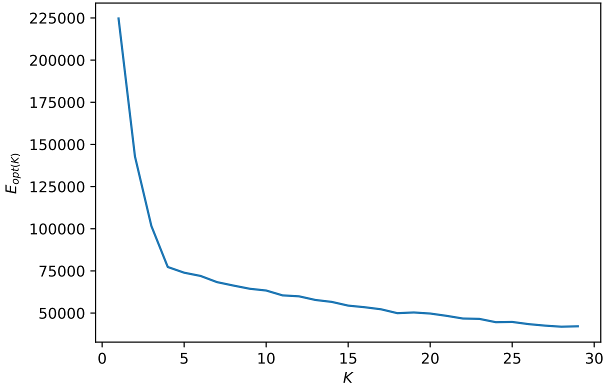
- $0 = E_{opt_N}(q^*) \leq E_{opt_{N-1}}(q^*) \leq \dots \leq E_{opt_1}(q^*) = var(X)$,
  - Proof: monotonicity by particularization, extremes with identity function (left) and variance (right).
- $0 \leq \kappa \leq \frac{N-1}{N}$.

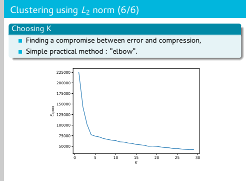
About the properties :
On the left side, if we take a cluster for each point in the space (N cluster centroids), then obviously the error is 0.
On the right side, if we take only one cluster, then the best cluster that can be chosen is the average of all points, in which case the error is exactly the variance across X.

Clustering using L_2 norm (6/6)

Choosing K

- Finding a compromise between error and compression,
- Simple practical method : "elbow".




Clustering using L_2 norm (6/6)

Choosing K

- Finding a compromise between error and compression,
- Simple practical method : "elbow".

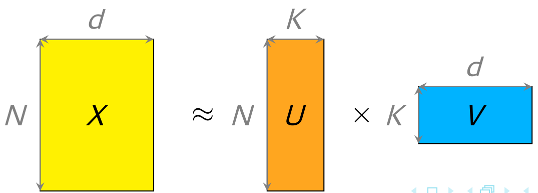
It is important to say that this is the ideal case! Here, we clearly see a value of K after which it is not necessary to add more clusters.

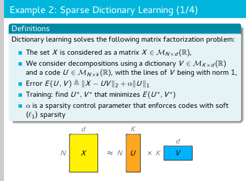
Example 2: Sparse Dictionary Learning (1/4)

Definitions

Dictionary learning solves the following matrix factorization problem:

- The set X is considered as a matrix $X \in \mathcal{M}_{N \times d}(\mathbb{R})$,
- We consider decompositions using a dictionary $V \in \mathcal{M}_{K \times d}(\mathbb{R})$ and a code $U \in \mathcal{M}_{N \times k}(\mathbb{R})$, with the lines of V being with norm 1,
- Error $E(U, V) \triangleq \|X - UV\|_2 + \alpha\|U\|_1$
- Training: find U^*, V^* that minimizes $E(U^*, V^*)$
- α is a sparsity control parameter that enforces codes with soft (ℓ_1) sparsity




Example 2: Sparse Dictionary Learning (1/4)
Definitions
Dictionary learning solves the following matrix factorization problem:
The set X is considered as a matrix $X \in \mathcal{M}_{N \times d}(\mathbb{R})$,
We consider decompositions using a dictionary $V \in \mathcal{M}_{K \times d}(\mathbb{R})$ and a code $U \in \mathcal{M}_{N \times k}(\mathbb{R})$, with the lines of V being with norm 1,
Error $E(U, V) \triangleq \|X - UV\|_2 + \alpha\|U\|_1$
Training: find U^*, V^* that minimizes $E(U^*, V^*)$
α is a sparsity control parameter that enforces codes with soft (ℓ_1) sparsity

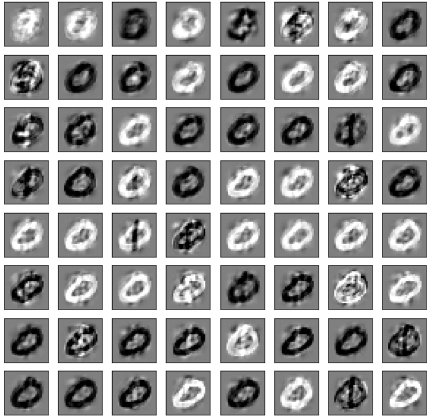
Here, just unroll the definition, by saying that Dictionary Learning is one way (among others) to perfom matrix factorization. It takes advantage of targetting a sparse code U. We will not explain here how to solve the optimization problem.

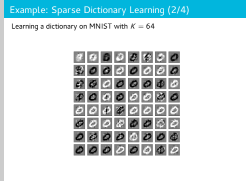
Note that the definition of the error here includes the sparsity term.

As a consequence, formally the error defined here is the optimization problem that is being solved, while the error (of reconstruction) regarding the original data is only the first term with the L2 norm.

Example: Sparse Dictionary Learning (2/4)

Learning a dictionary on MNIST with $K = 64$





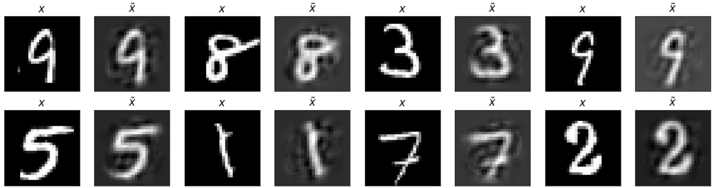
Example: Sparse Dictionary Learning (2/4)

Learning a dictionary on MNIST with $K = 64$

This is what a sparse dictionnary looks like, with 64 atoms in the dictionnary, on MNIST.

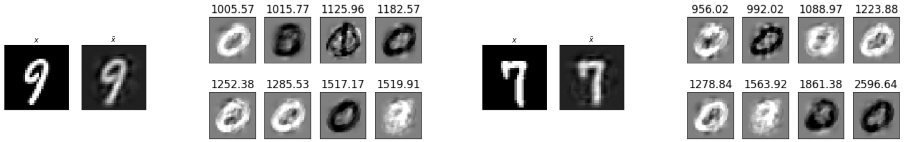
Example 2: Sparse Dictionary Learning (3/4)

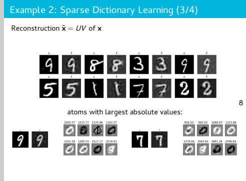
Reconstruction $\tilde{\mathbf{x}} = UV$ of $\mathbf{x}$



8

atoms with largest absolute values:




Example 2: Sparse Dictionary Learning (3/4)
Reconstruction $\tilde{\mathbf{x}} = UV$ of $\mathbf{x}$
8
atoms with largest absolute values:

In this slide we show the result of reconstructing the original vectors using the learnt dictionnary. In the top panel, we only show the results of reconstruction. In the bottom panel, we show some examples of how the atoms are combined, by showing the absolute values of atoms and the corresponding code (i.e. how the atoms are weighted to reconstruct the original vector).

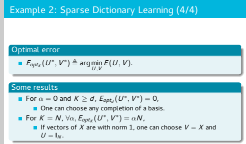
Example 2: Sparse Dictionary Learning (4/4)

Optimal error

- $E_{opt_K}(U^*, V^*) \triangleq \arg\min_{U,V} E(U, V)$.

Some results

- For $\alpha = 0$ and $K \geq d$, $E_{opt_d}(U^*, V^*) = 0$,
  - One can choose any completion of a basis.
- For $K = N$, $\forall \alpha$, $E_{opt_K}(U^*, V^*) = \alpha N$,
  - If vectors of X are with norm 1, one can choose $V = X$ and $U = \mathbf{I}_N$.



Example 2: Sparse Dictionary Learning (4/4)

Optimal error

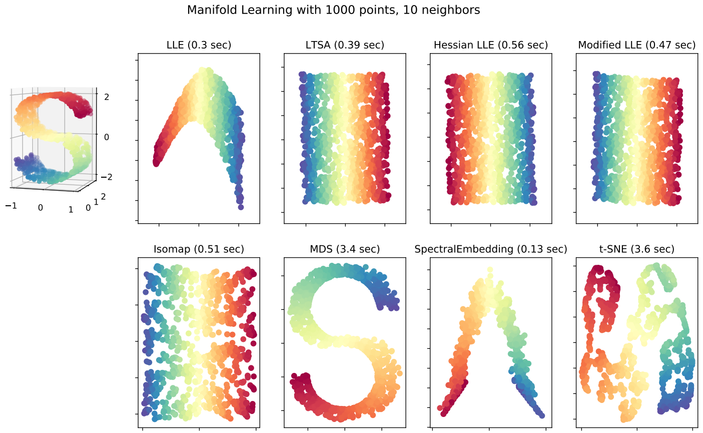
- $E_{opt_K}(U^*, V^*) \triangleq \arg\min_{U,V} E(U, V)$.

Some results

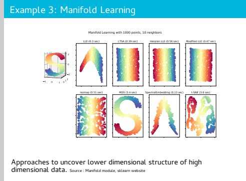
- For $\alpha = 0$ and $K \geq d$, $E_{opt_d}(U^*, V^*) = 0$,
  - One can choose any completion of a basis.
- For $K = N$, $\forall \alpha$, $E_{opt_K}(U^*, V^*) = \alpha N$,
  - If vectors of X are with norm 1, one can choose $V = X$ and $U = \mathbf{I}_N$.

Some comments about the results in the bottom block. If there is no sparsity, and for K higher than the number of dimension, then any basis of the space can be taken and the error is 0. This is a direct consequence of the fact that we are working in a orthonormal space. Regarding the second item, if taking as many atoms as points in the space, then the error is exactly αN, by simply normalizing vectors of X to norm 1, then choose X as dictionary, and the identity as code.

Example 3: Manifold Learning



Approaches to uncover lower dimensional structure of high dimensional data. Source : Manifold module, sklearn website

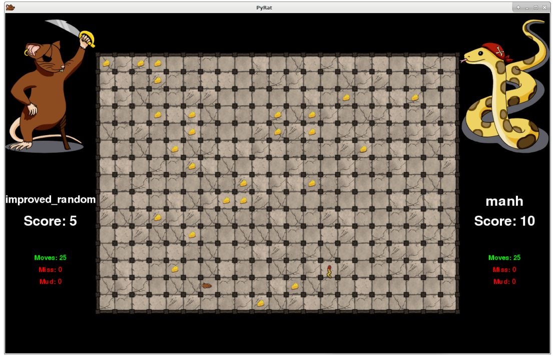

Example 3: Manifold Learning

Approaches to uncover lower dimensional structure of high dimensional data. Source : Manifold module, sklearn website

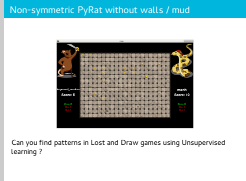
Tell them here that we don't have time to investigate in detail how these different methods work. The important thing is to explain the range of methods that can uncover the lower dimensional topology, in an unsupervised way.

Re-explain the original data (the swiss roll in the top right corner) and explain that there are methods that use different metrics (potentially non linear ones) that try to project in lower d.

Non-symmetric PyRat without walls / mud



Can you find patterns in Lost and Draw games using Unsupervised learning ?


Non-symmetric PyRat without walls / mud

Can you find patterns in Lost and Draw games using Unsupervised learning ?

We just state here the goal for the next lab session.

Lab Session 3 and assignments for Session 5

TP Unsupervised Learning (TP2)

- K-means, Dictionary Learning and Manifold Learning
- Application on Digits and PyRat

Project 2 (P2)

You will choose an unsupervised learning method. You have to prepare a Jupyter Notebook on this method, including:

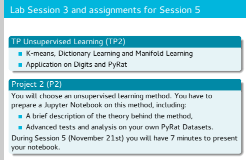
- A brief description of the theory behind the method,
- Advanced tests and analysis on your own PyRat Datasets.

During Session 5 (November 21st) you will have 7 minutes to present your notebook.

2018-11-05

Course 3: Unsupervised Learning

└─Lab Session 3 and assignments for Session 5



Lab Session 3 and assignments for Session 5

TP Unsupervised Learning (TP2)

- K-means, Dictionary Learning and Manifold Learning
- Application on Digits and PyRat

Project 2 (P2)

You will choose an unsupervised learning method. You have to prepare a Jupyter Notebook on this method, including:

- A brief description of the theory behind the method,
- Advanced tests and analysis on your own PyRat Datasets.

During Session 5 (November 21st) you will have 7 minutes to present your notebook.

Self explanatory!