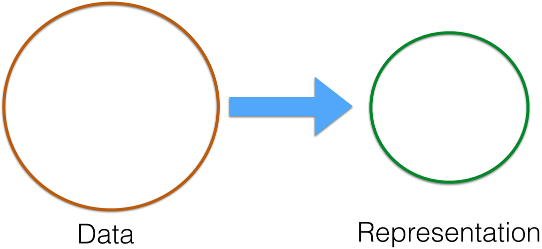
RegML 2020
Class 7
Dictionary learning

Lorenzo Rosasco
UNIGE-MIT-IIT

Data representation

A mapping of data in new **format** better suited for further processing



Data representation (cont.)

$\mathcal{X}$ data-space, a **data representation** is a map

$$\Phi : \mathcal{X} \to \mathcal{F},$$

to a **representation space** $\mathcal{F}$.

Different names in different fields:

- **machine learning**: feature map
- **signal processing**: analysis operator/transform
- **information theory**: encoder
- **computational geometry**: embedding

Outline

Part II: Data representation by learning

Dictionary learning

Metric learning

Supervised or Unsupervised?

Supervised (labelled/annotated) data are *expensive!*

Ideally a good data representation should reduce the need of (human) annotation...

$\rightsquigarrow$ **Unsupervised** learning of Φ

Unsupervised representation learning

Samples

$$S = \{x_1, \dots, x_n\}$$

from a distribution ρ on the input space $\mathcal{X}$ are available.

What are the **principles** to learn "good" representation in an unsupervised fashion?

Unsupervised representation learning principles

Two main concepts

1. **Reconstruction**, there exists a map $\Psi : \mathcal{F} \to \mathcal{X}$ such that

$$\Psi \circ \Phi(x) \sim x, \quad \forall x \in \mathcal{X}$$

2. **Similarity preservation**, it holds

$$\Phi(x) \sim \Phi(x') \Leftrightarrow x \sim x', \quad \forall x \in \mathcal{X}$$

Unsupervised representation learning principles

Two main concepts

1. **Reconstruction**, there exists a map $\Psi : \mathcal{F} \to \mathcal{X}$ such that

$$\Psi \circ \Phi(x) \sim x, \quad \forall x \in \mathcal{X}$$

2. **Similarity preservation**, it holds

$$\Phi(x) \sim \Phi(x') \Leftrightarrow x \sim x', \quad \forall x \in \mathcal{X}$$

Most unsupervised work has focused on reconstruction rather than on similarity
⇢ We give an overview next

Reconstruction based data representation

Basic idea: the quality of a representation Φ is measured by the **reconstruction error** provided by an associated reconstruction Ψ

$$\|x - \Psi \circ \Phi(x)\|,$$

Empirical data and population

Given $S = \{x_1, \ldots, x_n\}$ minimize the **empirical reconstruction error**

$$\widehat{\mathcal{E}}(\Phi, \Psi) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2 ,$$

Empirical data and population

Given $S = \{x_1, \dots, x_n\}$ minimize the **empirical reconstruction error**

$$\widehat{\mathcal{E}}(\Phi, \Psi) = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2,$$

as a proxy to the **expected reconstruction error**

$$\mathcal{E}(\Phi, \Psi) = \int d\rho(x) \|x - \Psi \circ \Phi(x)\|^2,$$

where ρ is the data distribution (fixed but uknown).

Empirical data and population

$$\min_{\Phi,\Psi} \mathcal{E}(\Phi,\Psi), \quad \mathcal{E}(\Phi,\Psi) = \int d\rho(x) \left\| x - \Psi \circ \Phi(x) \right\|^2,$$

Caveat...

But reconstruction alone is **not enough**...
copying data, i.e. $\Psi \circ \Phi = I$, gives zero reconstruction error!

Dictionary learning

$$\|x - \Psi \circ \Phi(x)\|$$

Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{F} = \mathbb{R}^p$

1. **linear reconstruction**

$$\Psi \in \mathcal{D},$$

with $\mathcal{D}$ a subset of the space of linear maps from $\mathcal{X}$ to $\mathcal{F}$.

Dictionary learning

$$\|x - \Psi \circ \Phi(x)\|$$

Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{F} = \mathbb{R}^p$

1. **linear reconstruction**

$$\Psi \in \mathcal{D},$$

   with $\mathcal{D}$ a subset of the space of linear maps from $\mathcal{X}$ to $\mathcal{F}$.

2. **nearest neighbor representation**,

$$\Phi(x) = \Phi_\Psi(x) = \arg\min_{\beta \in \mathcal{F}_\lambda} \|x - \Psi\beta\|^2, \qquad \Psi \in \mathcal{D},$$

   where $\mathcal{F}_\lambda$ is a subset of $\mathcal{F}$.

Linear reconstruction and dictionaries

Each reconstruction $\Psi \in \mathcal{D}$ can be identified a **dictionary** matrix with columns

$$a_1, \dots, a_p \in \mathbb{R}^d.$$

Linear reconstruction and dictionaries

Each reconstruction $\Psi \in \mathcal{D}$ can be identified a **dictionary** matrix with columns

$$a_1, \dots, a_p \in \mathbb{R}^d.$$

The reconstruction of an input $x \in \mathcal{X}$ corresponds to a suitable **linear expansion** on the dictionary

$$x = \sum_{j=1}^{p} a_j \beta_j, \qquad \beta_1, \dots, \beta_p \in \mathbb{R}.$$

Nearest neighbor representation

$$\Phi(x) = \Phi_\Psi(x) = \operatorname*{arg\,min}_{\beta \in \mathcal{F}_\lambda} \left\| x - \Psi\beta \right\|^2, \qquad \Psi \in \mathcal{D},$$

The above representation is called **nearest neighbor (NN)** since, for

$$\Psi \in \mathcal{D}, \quad \mathcal{X}_\lambda = \Psi \mathcal{F}_\lambda,$$

the representation $\Phi(x)$ provides the **closest** point to x in $\mathcal{X}_\lambda$,

$$d(x, \mathcal{X}_\lambda) = \min_{x' \in \mathcal{X}_\lambda} \left\| x - x' \right\|^2 = \min_{\beta \in \mathcal{F}_\lambda} \left\| x - \Psi\beta \right\|^2.$$

Nearest neighbor representation (cont.)

NN representation are defined by a **constrained inverse problem**,

$$\min_{\beta \in \mathcal{F}_\lambda} \|x - \Psi\beta\|^2 .$$

Nearest neighbor representation (cont.)

NN representation are defined by a **constrained inverse problem**,

$$\min_{\beta \in \mathcal{F}_\lambda} \|x - \Psi\beta\|^2 .$$

Alternatively let $\mathcal{F}_\lambda = \mathcal{F}$ and adding a regularization term $R_\lambda : \mathcal{F} \to \mathbb{R}$

$$\min_{\beta \in \mathcal{F}} \left\{ \|x - \Psi\beta\|^2 + R_\lambda(\beta) \right\} .$$

Dictionary learning

Then

$$\min_{\Psi,\Phi} \frac{1}{n}\sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2$$

becomes

$$\underbrace{\min_{\Psi\in\mathcal{D}}}_{\text{Dictionary learning}} \quad \frac{1}{n}\sum_{i=1}^{n} \underbrace{\min_{\beta_i\in\mathcal{F}_\lambda} \|x_i - \Psi\beta_i\|^2}_{\text{Representation learning}}.$$

Dictionary learning

- learning a **regularized representation** on a dictionary. . .
- **while** simultaneously **learning the dictionary** itself.

Examples

The framework introduced above encompasses a large number of approaches.

- PCA (& kernel PCA)
- KSVD
- Sparse coding
- K-means
- K-flats
- ...

Example 1: Principal Component Analysis (PCA)

Let $\mathcal{F}_\lambda = \mathcal{F}_k = \mathbb{R}^k$, $k \leq \min\{n, d\}$, and

$$\mathcal{D} = \{\Psi : \mathcal{F} \to \mathcal{X},\ \text{linear} \mid \Psi^*\Psi = I\}.$$

Example 1: Principal Component Analysis (PCA)

Let $\mathcal{F}_\lambda = \mathcal{F}_k = \mathbb{R}^k$, $k \le \min\{n, d\}$, and

$$\mathcal{D} = \{\Psi : \mathcal{F} \to \mathcal{X}, \text{ linear} \mid \Psi^*\Psi = I\}.$$

- Ψ is a $d \times k$ matrix with **orthogonal, unit norm** columns,

$$\Psi\beta = \sum_{j=1}^{k} a_j \beta_j, \quad \beta \in \mathcal{F}$$

Example 1: Principal Component Analysis (PCA)

Let $\mathcal{F}_\lambda = \mathcal{F}_k = \mathbb{R}^k$, $k \leq \min\{n, d\}$, and

$$\mathcal{D} = \{\Psi : \mathcal{F} \to \mathcal{X},\ \text{linear} \mid \Psi^*\Psi = I\}.$$

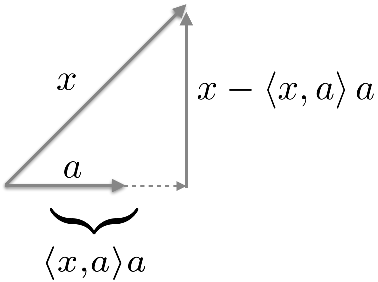
- Ψ is a $d \times k$ matrix with **orthogonal, unit norm** columns,

$$\Psi\beta = \sum_{j=1}^{k} a_j \beta_j, \quad \beta \in \mathcal{F}$$

- $\Psi^* : \mathcal{X} \to \mathcal{F}, \quad \Psi^* x = (\langle a_1, x \rangle, \ldots, \langle a_k, x \rangle), \quad x \in \mathcal{X}$

PCA & best subspace

- $\Psi\Psi^* : \mathcal{X} \to \mathcal{X}, \quad \Psi\Psi^* x = \sum_{j=1}^k a_j \langle a_j, x \rangle, \quad x \in \mathcal{X}.$



- $P = \Psi\Psi^*$ is the **projection** $(P = P^2)$ on the subspace of $\mathbb{R}^d$ **spanned** by $a_1, \ldots, a_k$.

Rewriting PCA

Note that,

$$\Phi(x) = \Psi^* x = \operatorname*{arg\,min}_{\beta \in \mathcal{F}_k} \left\| x - \Psi\beta \right\|^2, \quad \forall x \in \mathcal{X},$$

so that we can rewrite the PCA minimization as

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \left\| x - \Psi\Psi^* x_i \right\|^2.$$

Rewriting PCA

Note that,

$$\Phi(x) = \Psi^* x = \underset{\beta \in \mathcal{F}_k}{\arg\min} \|x - \Psi\beta\|^2, \quad \forall x \in \mathcal{X},$$

so that we can rewrite the PCA minimization as

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \|x - \Psi\Psi^* x_i\|^2.$$

Subspace learning

The problem of finding a $k-$dimensional orthogonal projection giving the best reconstruction.

PCA computation

Let $\widehat{X}$ the $n \times d$ data matrix and $C = \frac{1}{n}\widehat{X}^T\widehat{X}$.

PCA computation

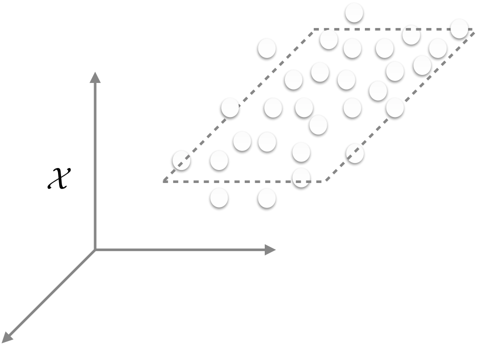
Let $\widehat{X}$ the $n \times d$ data matrix and $C = \frac{1}{n}\widehat{X}^T\widehat{X}$.

...PCA optimization problem is solved by the eigenvector of C associated to the K largest eigenvalues.

Learning a linear representation with PCA

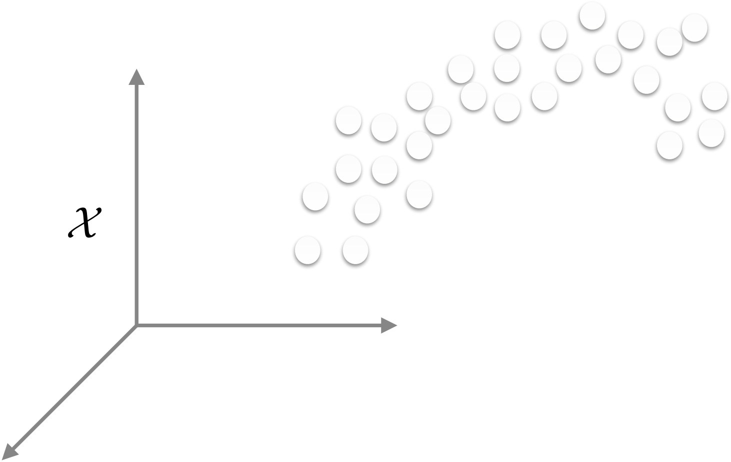
Subspace learning

The problem of finding a $k-$dimensional orthogonal projection giving the best reconstruction.

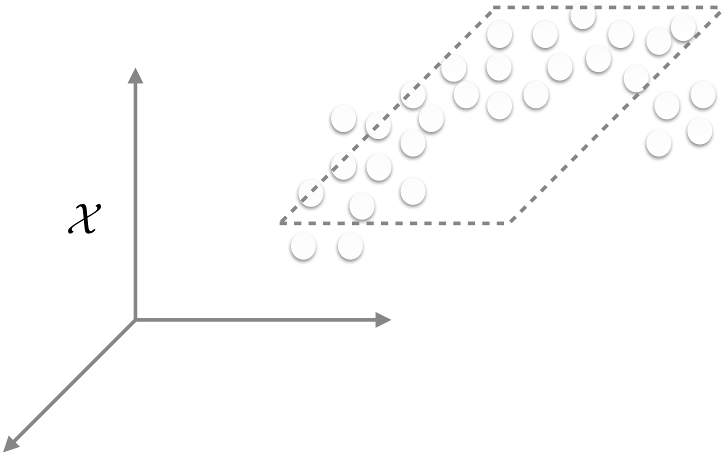


PCA assumes the support of the data distribution to be well approximated by a low dimensional *linear* subspace

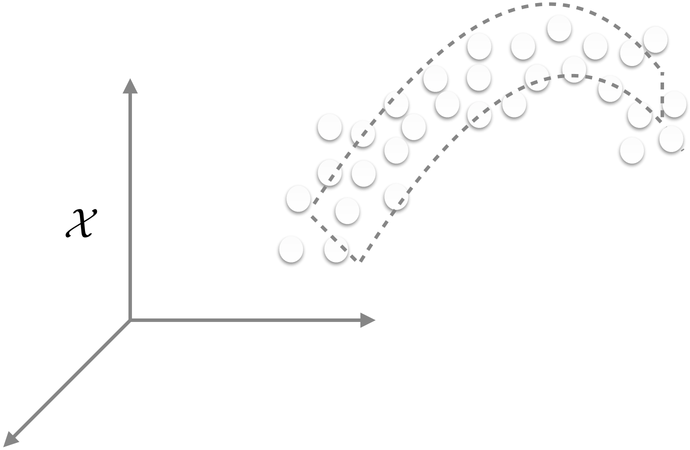
PCA beyond linearity



PCA beyond linearity



PCA beyond linearity



Kernel PCA

Consider

$$\phi : \mathcal{X} \to \mathcal{H}, \quad \text{and} \quad K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

a **feature map and associated (reproducing) kernel**.
We can consider the empirical **reconstruction in the feature space**,

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \min_{\beta_i \in \mathcal{H}} \|\phi(x_i) - \Psi \beta_i\|_{\mathcal{H}}^2 .$$

Connection to manifold learning...

Examples 2: Sparse coding

One of the first and most famous dictionary learning techniques.

Examples 2: Sparse coding

One of the first and most famous dictionary learning techniques.

It corresponds to

- $\mathcal{F} = \mathbb{R}^p$,

Examples 2: Sparse coding

One of the first and most famous dictionary learning techniques.

It corresponds to

- $\mathcal{F} = \mathbb{R}^p$,
- $p \geq d$, $\mathcal{F}_\lambda = \{\beta \in \mathcal{F} \ : \ \|\beta\|_1 \leq \lambda\}, \quad \lambda > 0,$

Examples 2: Sparse coding

One of the first and most famous dictionary learning techniques.

It corresponds to

- $\mathcal{F} = \mathbb{R}^p$,
- $p \geq d$, $\mathcal{F}_\lambda = \{\beta \in \mathcal{F} \; : \; \|\beta\|_1 \leq \lambda\}, \quad \lambda > 0$,
- $\mathcal{D} = \{\Psi : \mathcal{F} \to \mathcal{X} \mid \|\Psi e_j\|_{\mathcal{F}} \leq 1\}$.

Examples 2: Sparse coding

One of the first and most famous dictionary learning techniques.

It corresponds to

- $\mathcal{F} = \mathbb{R}^p$,
- $p \geq d$, $\mathcal{F}_\lambda = \{\beta \in \mathcal{F} \; : \; \|\beta\|_1 \leq \lambda\}, \quad \lambda > 0,$
- $\mathcal{D} = \{\Psi : \mathcal{F} \to \mathcal{X} \mid \|\Psi e_j\|_{\mathcal{F}} \leq 1\}$.

Hence,

$$\underbrace{\min_{\Psi \in \mathcal{D}}}_{\text{dictionary learning}} \quad \frac{1}{n} \sum_{i=1}^{n} \underbrace{\min_{\beta_i \in \mathcal{F}_\lambda} \|x_i - \Psi \beta_i\|^2}_{\text{sparse representation}}$$

Sparse coding (cont.)

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \min_{\beta_i \in \mathbb{R}^p, \|\beta_i\| \le \lambda} \|x_i - \Psi \beta_i\|^2$$

- The problem is **not convex**... but it is **separately convex** in the β_i's and Ψ.
- An alternating minimization is fairly natural (other approaches possible–see e.g. [Schnass '15, Elad et al. '06])

Representation computation

Given a dictionary, the problems

$$\min_{\beta \in \mathcal{F}_\lambda} \left\| x_i - \Psi \beta \right\|^2, i = 1, \ldots, n$$

are convex and correspond to a **sparse representation** problems.

They can be solved using **convex optimization** techniques.

Splitting/proximal methods

$$\beta_0, \quad \beta_{t+1} = T_{\gamma,\lambda}(\beta_t - \gamma \Psi^*(x_i - \Psi \beta_t)), \quad t = 0, \ldots, T_{\mathsf{max}}$$

with T_λ the soft-thresholding operator,

Dictionary computation

Given $\Phi(x_i) = \beta_i$, $i = 1, \dots, n$, we have

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2 = \min_{\Psi \in \mathcal{D}} \frac{1}{n} \left\| \widehat{X} - B^* \Psi \right\|_F^2,$$

where B is the $n \times p$ matrix with rows β_i, $i = 1, \dots, n$ and we denoted by $\|\cdot\|_F$, the Frobenius norm.

Dictionary computation

Given $\Phi(x_i) = \beta_i$, $i = 1, \dots, n$, we have

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2 = \min_{\Psi \in \mathcal{D}} \frac{1}{n} \left\| \widehat{X} - B^* \Psi \right\|_F^2,$$

where B is the $n \times p$ matrix with rows β_i, $i = 1, \dots, n$ and we denoted by $\|\cdot\|_F$, the Frobenius norm.

It is a convex problem, solvable via standard techniques.

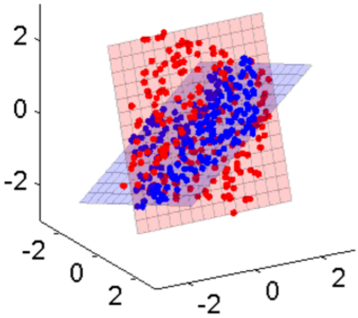
Splitting/proximal methods

$$\Psi_0, \quad \Psi_{t+1} = P(\Psi_t - \gamma_t B^*(X - \Psi B)), \quad t = 0, \dots, T_{\max}$$

where P is the projection corresponding to the constraints,

$$\begin{aligned} P(\Psi^j) &= \Psi^j / \left\|\Psi^j\right\|, \quad \text{if } \left\|\Psi^j\right\| > 1 \\ P(\Psi^j) &= \Psi^j, \quad \text{if } \left\|\Psi^j\right\| \leq 1. \end{aligned}$$

Sparse coding model



- Sparse coding assumes the support of the data distribution to be a union of $\binom{p}{s}$ subspaces, i.e. all possible s dimensional subspaces in $\mathbb{R}^p$, where s is the sparsity level.
- More general penalties, more general geometric assumptions.

Example 3: K-means & vector quantization

K-means is typically seen as a **clustering** algorithm in machine learning...

Example 3: K-means & vector quantization

K-means is typically seen as a **clustering** algorithm in machine learning. . . but it is also a classical **vector quantization** approach.

Example 3: K-means & vector quantization

K-means is typically seen as a **clustering** algorithm in machine learning... but it is also a classical **vector quantization** approach.

Here we revisit this point of view from a **data representation** perspective.

Example 3: K-means & vector quantization

K-means is typically seen as a **clustering** algorithm in machine learning... but it is also a classical **vector quantization** approach.

Here we revisit this point of view from a **data representation** perspective.

K-means corresponds to

- $\mathcal{F}_\lambda = \mathcal{F}_k = \{e_1, \dots, e_k\}$, the canonical basis in $\mathbb{R}^k$, $k \leq n$

Example 3: K-means & vector quantization

K-means is typically seen as a **clustering** algorithm in machine learning... but it is also a classical **vector quantization** approach.

Here we revisit this point of view from a **data representation** perspective.

K-means corresponds to

- $\mathcal{F}_\lambda = \mathcal{F}_k = \{e_1, \dots, e_k\}$, the canonical basis in $\mathbb{R}^k$, $k \leq n$
- $\mathcal{D} = \{\Psi : \mathcal{F} \to \mathcal{X} \mid \text{linear}\}$.

K-means computation

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \min_{\beta_i \in \{e_1,\dots,e_k\}} \|x_i - \Psi\beta_i\|^2$$

The K-means problem is not convex.

K-means computation

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \min_{\beta_i \in \{e_1, \dots, e_k\}} \|x_i - \Psi \beta_i\|^2$$

The K-means problem is not convex.

Alternating minimization

1. Initialize dictionary Ψ_0.

K-means computation

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \min_{\beta_i \in \{e_1,\dots,e_k\}} \|x_i - \Psi \beta_i\|^2$$

The K-means problem is not convex.

Alternating minimization

1. Initialize dictionary Ψ_0.
2. Let $\Phi(x_i) = \beta_i$, $i = 1, \dots, n$ be the solution of the problems

$$\min_{\beta \in \{e_1,\dots,e_k\}} \|x_i - \Psi \beta\|^2, \quad i = 1, \dots, n.$$

with $V_j = \{x \in S \mid \Phi(x) = e_j\}$, (multiple points have same representation since $k \leq n$).

K-means computation

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \min_{\beta_i \in \{e_1, \dots, e_k\}} \|x_i - \Psi \beta_i\|^2$$

The K-means problem is not convex.

Alternating minimization

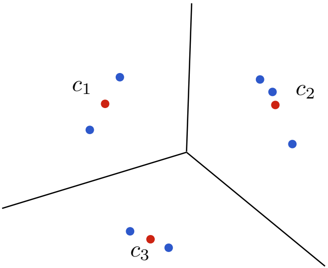
1. Initialize dictionary Ψ_0.
2. Let $\Phi(x_i) = \beta_i$, $i = 1, \dots, n$ be the solution of the problems

   $$\min_{\beta \in \{e_1, \dots, e_k\}} \|x_i - \Psi \beta\|^2, \quad i = 1, \dots, n.$$

   with $V_j = \{x \in S \mid \Phi(x) = e_j\}$, (multiple points have same representation since $k \leq n$).
3. Letting $a_j = \Psi e_j$, we can write

   $$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2 = \min_{a_1, \dots, a_k \in R^d} \frac{1}{n} \sum_{j=1}^{k} \sum_{x \in V_j} \|x - a_j\|^2.$$

Step 2: assignment

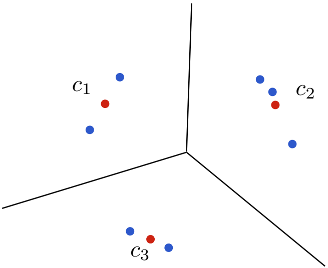


The discrete problem

$$\min_{\beta \in \{e_1,\dots,e_k\}} \|x_i - \Psi\beta\|^2, \quad i = 1,\dots,n.$$

can be seen as an **assignment** step.

Step 2: assignment



The discrete problem

$$\min_{\beta \in \{e_1,\dots,e_k\}} \|x_i - \Psi\beta\|^2, \quad i = 1, \dots, n.$$

can be seen as an **assignment** step.

Clusters

The sets

$$V_j = \{x \in S \mid \Phi(x) = e_j\},$$

are called **Voronoi** sets and can be seen as data **clusters**.

Step 3: centroid computation

Consider

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2 = \min_{a_1, \dots, a_k \in R^d} \frac{1}{n} \sum_{j=1}^{k} \sum_{x \in V_j} \|x - a_j\|^2,$$

where $a_j = \Psi e_j$.

Step 3: centroid computation

Consider

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2 = \min_{a_1,\dots,a_k \in R^d} \frac{1}{n} \sum_{j=1}^{k} \sum_{x \in V_j} \|x - a_j\|^2,$$

where $a_j = \Psi e_j$.

The minimization with respect to each column is **independent** to all others.

Step 3: centroid computation

Consider

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \|x_i - \Psi \circ \Phi(x_i)\|^2 = \min_{a_1, \dots, a_k \in R^d} \frac{1}{n} \sum_{j=1}^{k} \sum_{x \in V_j} \|x - a_j\|^2 ,$$

where $a_j = \Psi e_j$.

The minimization with respect to each column is **independent** to all others.

Centroid computation

$$c_j = \frac{1}{|V_j|} \sum_{x \in V_j} x = \arg\min_{a_j \in \mathbb{R}^d} \sum_{x \in V_j} \|x - a_j\|^2 , \quad j = 1, \dots, k.$$

K-means convergence

The computational procedure described before is known as **Lloyd's algorithm**.

K-means convergence

The computational procedure described before is known as **Lloyd's algorithm**.

- Since it is an **alternating minimization** approach, the value of the objective function can be shown to **decrease** with the iterations.

K-means convergence

The computational procedure described before is known as **Lloyd's algorithm**.

- Since it is an **alternating minimization** approach, the value of the objective function can be shown to **decrease** with the iterations.
- Since there is only a **finite** number of possible partitions of the data in k clusters, Lloyd's algorithm is ensured to **converge to a local minimum** in a finite number of steps.

K-means initialization

Convergence to a **global** minimum can be ensured (with high probability), provided a suitable initialization.

K-means initialization

Convergence to a **global** minimum can be ensured (with high probability), provided a suitable initialization.

K-means++ [Arthur, Vassilvitskii;07]

K-means initialization

Convergence to a **global** minimum can be ensured (with high probability), provided a suitable initialization.

K-means++ [Arthur, Vassilvitskii;07]

1. Choose a centroid uniformly at random from the data,

K-means initialization

Convergence to a **global** minimum can be ensured (with high probability), provided a suitable initialization.

K-means++ [Arthur, Vassilvitskii;07]

1. Choose a centroid uniformly at random from the data,
2. Compute distances of data to the nearest centroid already chosen.

K-means initialization

Convergence to a **global** minimum can be ensured (with high probability), provided a suitable initialization.

K-means++ [Arthur, Vassilvitskii;07]

1. Choose a centroid uniformly at random from the data,
2. Compute distances of data to the nearest centroid already chosen.
3. Choose a new centroid from the data using probabilities proportional to such distances (squared).

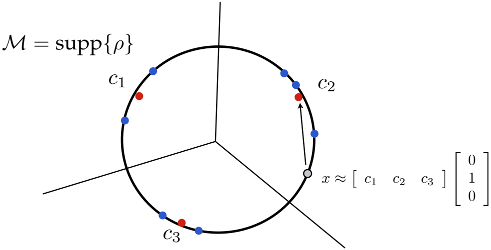
K-means initialization

Convergence to a **global** minimum can be ensured (with high probability), provided a suitable initialization.

K-means++ [Arthur, Vassilvitskii;07]

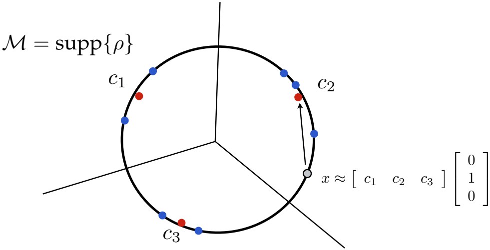
1. Choose a centroid uniformly at random from the data,
2. Compute distances of data to the nearest centroid already chosen.
3. Choose a new centroid from the data using probabilities proportional to such distances (squared).
4. Repeat steps 2 and 3 until k centers have been chosen.

K-means & piece-wise representation



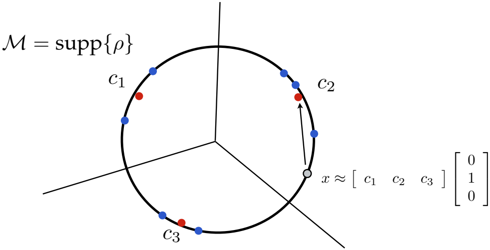
- k-means representation: **extreme sparse representation**, only one non zero coefficient (**vector quantization**).

K-means & piece-wise representation



- k-means representation: **extreme sparse representation**, only one non zero coefficient (**vector quantization**).
- k-means reconstruction: **piecewise constant** approximation of the data, each point is reconstructed by the nearest mean.

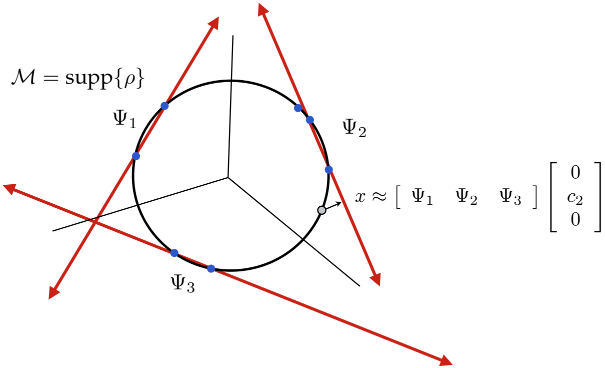
K-means & piece-wise representation



- k-means representation: **extreme sparse representation**, only one non zero coefficient (**vector quantization**).
- k-means reconstruction: **piecewise constant** approximation of the data, each point is reconstructed by the nearest mean.

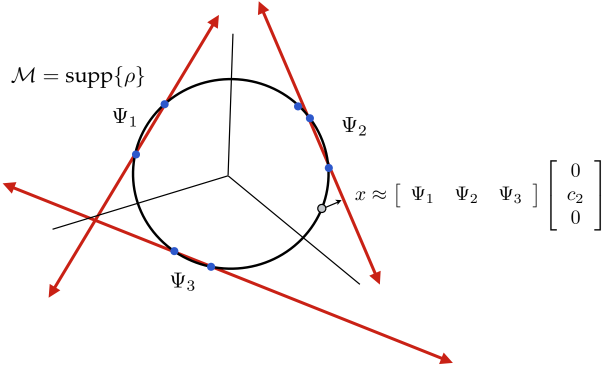
This latter perspective suggests extensions of k-means considering **higher order** data approximation such as, e.g. piecewise linear.

K-flats & piece-wise linear representation



[Bradley, Mangasarian '00, Canas, R.'12]

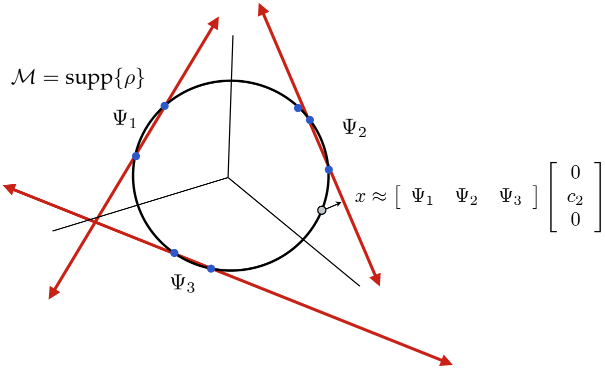
K-flats & piece-wise linear representation



[Bradley, Mangasarian '00, Canas, R.'12]

- k-flats representation: **structured sparse representation**, coefficients are projection on a *flat*.

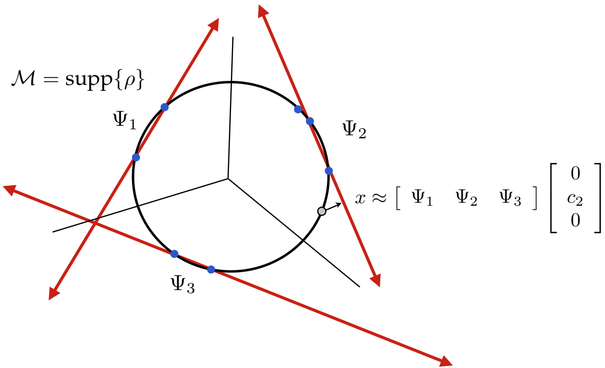
K-flats & piece-wise linear representation



[Bradley, Mangasarian '00, Canas, R.'12]

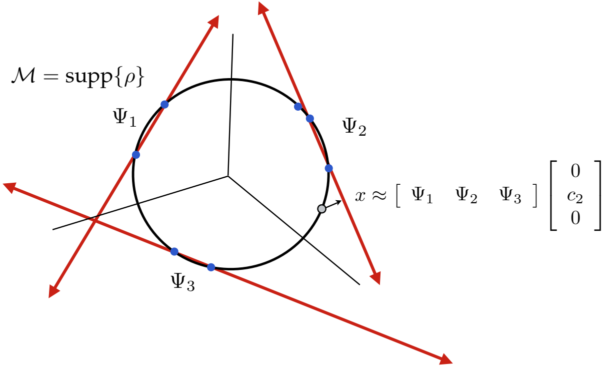
- k-flats representation: **structured sparse representation**, coefficients are projection on a *flat*.
- k-flats reconstruction: **piecewise linear** approximation of the data, each point is reconstructed by projection on the nearest flat.

Remarks on K-flats



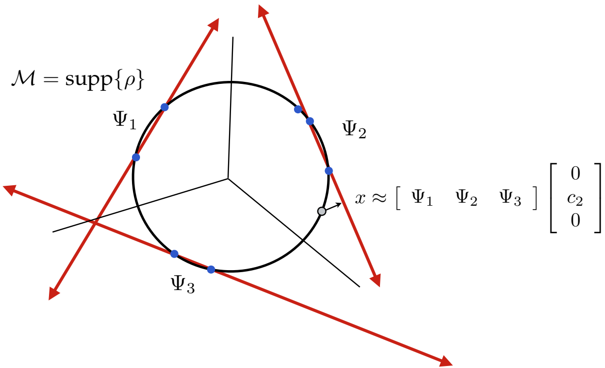
- Principled way to **enrich** k-means representation (cfr *softmax*).

Remarks on K-flats



- Principled way to **enrich** k-means representation (cfr *softmax*).
- **Geometric structured** dictionary learning.

Remarks on K-flats



- Principled way to **enrich** k-means representation (cfr *softmax*).
- **Geometric structured** dictionary learning.
- **Non-local** approximations.

K-flats computations

Alternating minimization

1. **Initialize** flats $\Psi_1, \dots, \Psi_k$.
2. **Assign** point to nearest flat,

$$V_j = \{x \in \mathcal{X} \mid \left\|x - \Psi_j \Psi_j^* x\right\| \leq \|x - \Psi_t \Psi_t^* x\|, \quad t \neq j\}.$$

3. **Update** flats by computing (local) PCA in each cell V_j, $j = 1, \dots, k$.

Kernel K-means & K-flats

It is easy to extend K-means & K-flats using **kernels**.

$$\phi : \mathcal{X} \to \mathcal{H}, \quad \text{and} \quad K(x, x') = \langle \phi(x), \phi(x') \rangle_{\mathcal{H}}$$

Consider the empirical reconstruction problem in the feature space,

$$\min_{\Psi \in \mathcal{D}} \frac{1}{n} \sum_{i=1}^{n} \min_{\beta_i \in \{e_1, \dots, e_k\} \subset \mathcal{H}} \|\phi(x_i) - \Psi \beta_i\|_{\mathcal{H}}^2 .$$

Note: Easy to see that computation can be performed in closed form

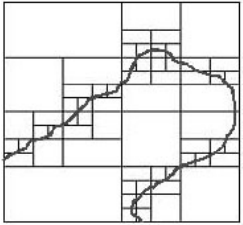
- Kernel k-means: **distance computation**.
- Kernel k-flats: **distance computation+local KPCA**.

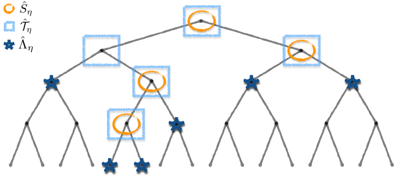
Geometric Wavelets (GW)- Reconstruction Trees

- **Select** (rather than compute) a partition of the data-space
- Approximate the point in each cell via a vector/plane.

multi-scale

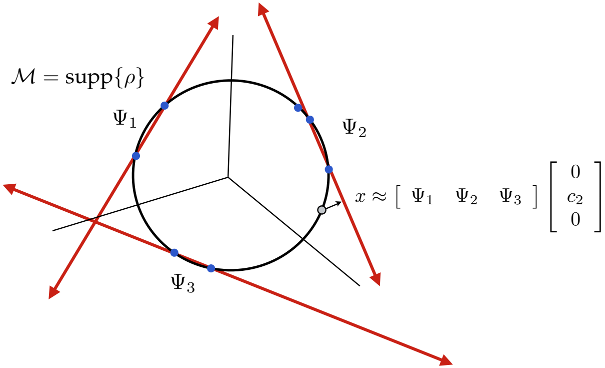
Selection via **multi-scale/coarse-to-fine** pruning of a partition tree [Maggioni et al. . . .]





K-means/flats and GW

- Can be seen as piecewise representations.
- The data model is a manifold– limit when the number of pieces goes to infinity
- GMRA is **local** (cells are connected) while K-Flats is not...
- ...but GMRA is **multi-scale** while K-flats is not...



Dictionary learning & matrix factorization

PCA,Sparse Coding, K-means/flats, Reconstruction trees are some examples of methods based on

$$(P1) \qquad \underbrace{\min_{\Psi \in \mathcal{D}}}_{\text{Dictionary learning}} \quad \frac{1}{n} \sum_{i=1}^{n} \underbrace{\min_{\beta_i \in \mathcal{F}_\lambda} \left\| x_i - \Psi \beta_i \right\|^2}_{\text{Representation learning}}.$$

In fact, under mild conditions the above problem is a special case of **Matrix Factorization**:

Dictionary learning & matrix factorization

PCA,Sparse Coding, K-means/flats, Reconstruction trees are some examples of methods based on

$$(P1) \qquad \underbrace{\min_{\Psi \in \mathcal{D}}}_{\text{Dictionary learning}} \qquad \frac{1}{n} \sum_{i=1}^{n} \underbrace{\min_{\beta_i \in \mathcal{F}_\lambda} \left\| x_i - \Psi \beta_i \right\|^2}_{\text{Representation learning}}.$$

In fact, under mild conditions the above problem is a special case of **Matrix Factorization**:

If the minimizations of the β_i's are independent, then

$$(P1) \Leftrightarrow \min_{B,\Psi} \left\| \widehat{X} - \Psi B \right\|_F^2$$

where B has columns $(\beta_i)_i$, $\widehat{X}$ data matrix, and $\|\cdot\|_F$ is the Frobenius norm.

The equivalence holds for all the methods we saw before!

From reconstruction to similarity

We have seen two concepts emerging

- **parsimonious reconstruction**
- **similarity preservation**

What about similarity preservation?

Randomized linear representation

Consider **randomized** representation/reconstruction given by a set of random templates smaller then data dimension, that is

$$a_1, \dots, a_k, \quad k < d.$$

Randomized linear representation

Consider **randomized** representation/reconstruction given by a set of random templates smaller then data dimension, that is

$$a_1, \dots, a_k, \quad k < d.$$

Consider $\Phi : \mathcal{X} \to \mathcal{F} = \mathbb{R}^k$ such that

$$\Phi(x) = Ax = (\langle x, a_1 \rangle, \dots, \langle x, a_k \rangle), \quad \forall x \in \mathcal{X},$$

with A random i.i.d. matrix, with rows $a_1, \dots, a_k$

Johnson-Lindenstrauss Lemma

The representation $\Phi(x) = Ax$ defines a **stable embedding**, i.e.

$$(1-\epsilon)\,\|x - x'\| \leq \|\Phi(x) - \Phi(x')\| \leq (1+\epsilon)\,\|x - x'\|$$

with high probability and for all $x, x' \in \mathcal{C} \subset \mathcal{X}$.
The precision ϵ depends on : 1) number of random atoms k, 2) the set $\mathcal{C}$

Johnson-Lindenstrauss Lemma

The representation $\Phi(x) = Ax$ defines a **stable embedding**, i.e.

$$(1-\epsilon)\|x-x'\| \le \|\Phi(x)-\Phi(x')\| \le (1+\epsilon)\|x-x'\|$$

with high probability and for all $x, x' \in \mathcal{C} \subset \mathcal{X}$.
The precision ϵ depends on : 1) number of random atoms k, 2) the set $\mathcal{C}$

Example:
If $\mathcal{C}$ is a finite set $|\mathcal{C}| = n$, then

$$\epsilon \sim \sqrt{\frac{\log n}{k}}.$$

Metric learning

Metric learning

Find $D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that

$$x \textit{ similar } x' \Leftrightarrow D(x, x')$$

1. **How to parameterize D?**
2. **How we know whether data points are similar?**
3. **How do we turn all into an optimization problem?**

Metric learning (cont.)

1. **How to parameterize D?**

$$\text{Mahalanobis} \quad D(x, x') = \langle x - x', M(x - x') \rangle$$

where M symmetric PD, or rather $\Phi(x) = Bx$ with $M = B^*B$ (using kernels possible).

Metric learning (cont.)

1. **How to parameterize D?**

   $$\text{Mahalanobis} \quad D(x, x') = \langle x - x', M(x - x') \rangle$$

   where M symmetric PD, or rather $\Phi(x) = Bx$ with $M = B^*B$ (using kernels possible).

2. **How to know whether points are similar?**
   Most works assume **supervised** data

   $$(x_i, x_j, y_{i,j})_{i,j}.$$

Metric learning (cont.)

1. **How to parameterize D?**

$$\text{Mahalanobis} \quad D(x, x') = \langle x - x', M(x - x') \rangle$$

   where M symmetric PD, or rather $\Phi(x) = Bx$ with $M = B^*B$ (using kernels possible).

2. **How to know whether points are similar?**
   Most works assume **supervised** data

$$(x_i, x_j, y_{i,j})_{i,j}.$$

3. **How to turn all into an optimization problem?**
   Extension of classification algorithms such as **support vector machines**.

This class

- dictionary learning
- metric learning

Next class

Deep learning!