

IFT 4030/7030,
Machine Learning for Signal Processing
**Week4: Machine Learning 1,
Decompositions**

Cem Subakan



UNIVERSITÉ
Laval



- Avez-vous regardé le document sur les proposals de projets?
 - ▶ Did you have a chance to read the project proposal document?
- On aura un deadline stricte pour les labos en commançant par labo 2.
 - ▶ The deadline for the labs will be strict from lab 2 on.
- Le devoir 1 va sortir bientot!
 - ▶ The first homework will be released soon!

- Avez-vous regardé le document sur les proposals de projets?
 - ▶ Did you have a chance to read the project proposal document?
- On aura un deadline stricte pour les labos en commançant par labo 2.
 - ▶ The deadline for the labs will be strict from lab 2 on.
- Le devoir 1 va sortir bientot!
 - ▶ The first homework will be released soon!
- Aujourd'hui on commence avec l'apprentissage automatique.
 - ▶ Today: We are starting with machine learning.

This week

- Today, our aim is to build the foundation for training machine learning models.
- Au'jourdhui le but est de batir le fondation pour l'entraînement des modèles.

This week

- Today, our aim is to build the foundation for training machine learning models.
- Au'jourdhui le but est de batir le fondation pour l'entraînement des modèles.
- More specifically, we will build a framework around learnable decompositions.
 - ▶ Plus spécifiquement on va batir un framework autour des décompositions apprises.

Table of Contents

The Decomposition Framework

Fixed Basis Decompositions (Linear Regression)

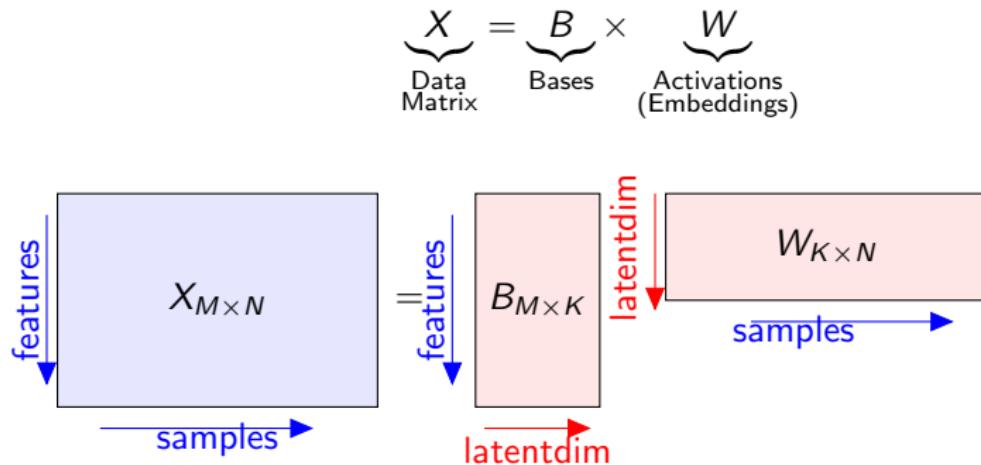
Learnable Basis Decompositions

Principal Component Analysis

Independent Component Analysis

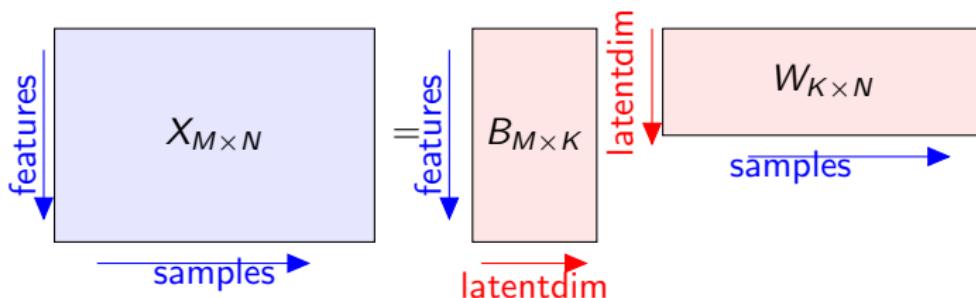
Non-Negative Matrix Factorization (NMF)

The framework



The framework

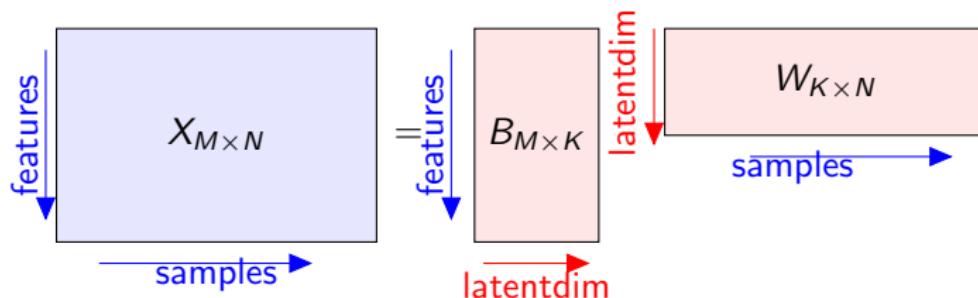
$$\underbrace{X}_{\text{Data Matrix}} = \underbrace{B}_{\text{Bases}} \times \underbrace{W}_{\text{Activations (Embeddings)}}$$



- Note that this framework embeds M dimensional data in K dimensions.
 - ▶ Notez qu'on est en train de trouver un embedding de K dimensions pour un data qui a M dimensions.

The framework

$$\underbrace{X}_{\text{Data Matrix}} = \underbrace{B}_{\text{Bases}} \times \underbrace{W}_{\text{Activations (Embeddings)}}$$



- Note that this framework embeds M dimensional data in K dimensions.
 - ▶ Notez qu'on est en train de trouver un embedding de K dimensions pour un data qui a M dimensions.
- We embed X , in the space defined by the columns of B .
 - ▶ On embed X dans une espace definit par les colonnes de B .

Table of Contents

The Decomposition Framework

Fixed Basis Decompositions (Linear Regression)

Learnable Basis Decompositions

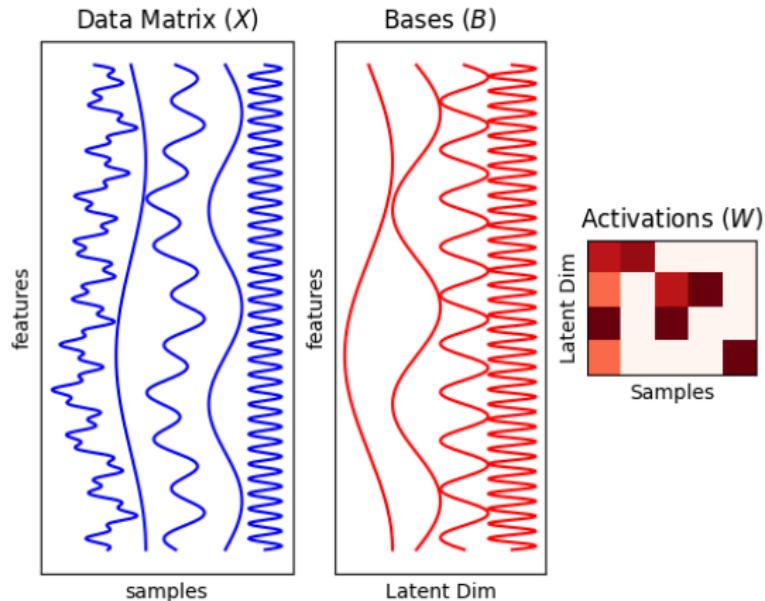
Principal Component Analysis

Independent Component Analysis

Non-Negative Matrix Factorization (NMF)

Example

- Remember this from last week?
 - ▶ Vous-vous en souvenez ça de la semaine passée?



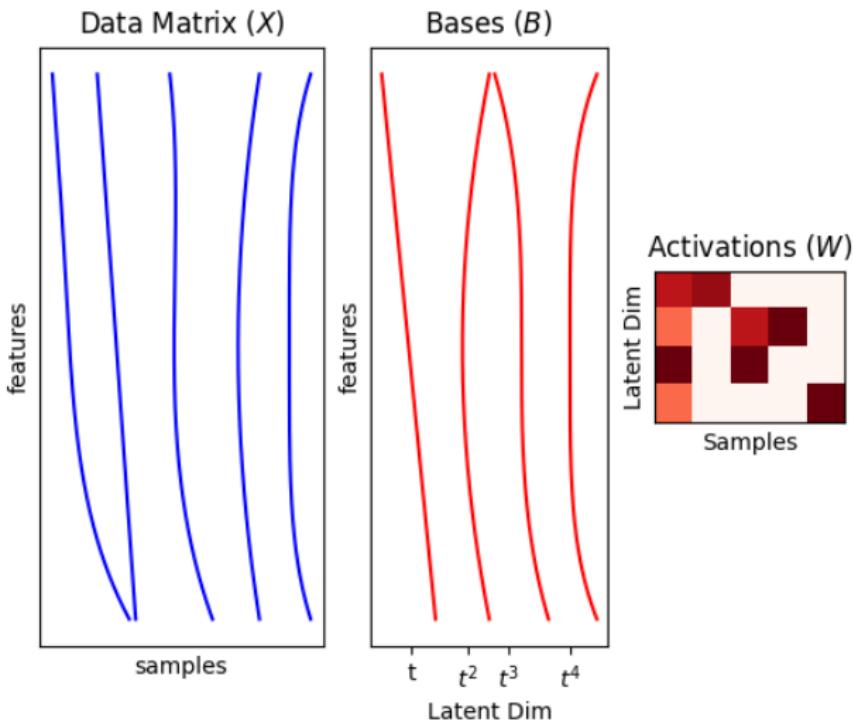
The goal

- We are trying to build a framework that can effectively reduce dimensionality, to explain data in a concise way.
 - ▶ On essaie de batir un framework qui peut effectivement reduire la dimensionalité et expliquer les données de manière parsimonieux.

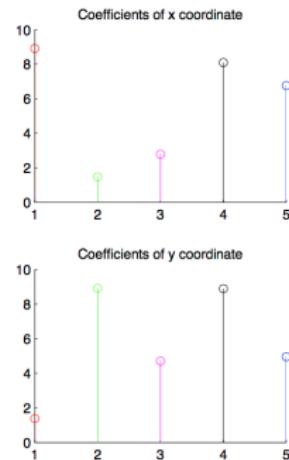
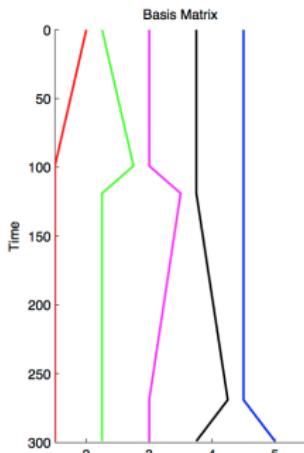
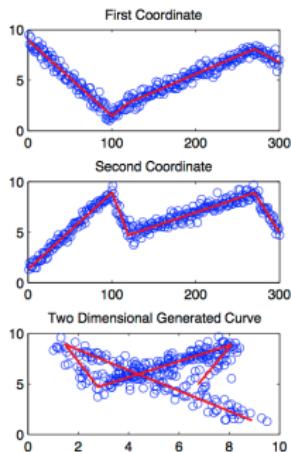
The goal

- We are trying to build a framework that can effectively reduce dimensionality, to explain data in a concise way.
 - ▶ On essaie de batir un framework qui peut effectivement reduire la dimensionalité et expliquer les données de manière parsimonieux.
- We can use basis functions other than sinusoids!
 - ▶ On peut utiliser des bases autres que les sinusoids!

Non-sinusoids (finally)

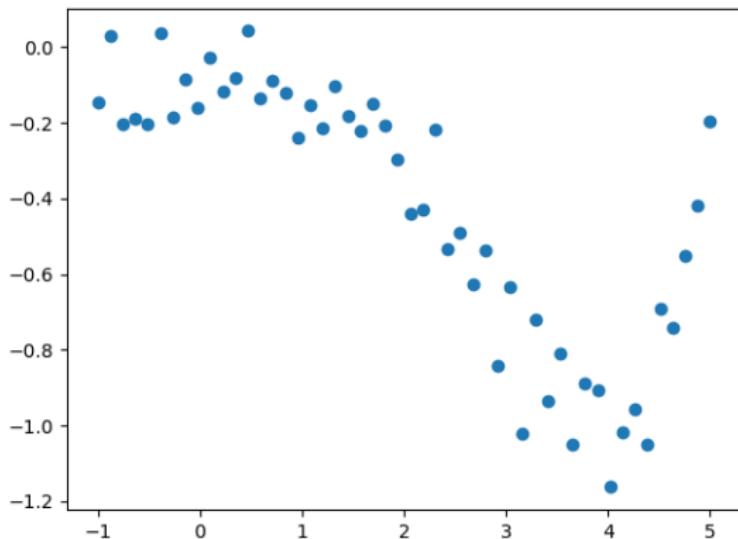


Piece-wise functions!!



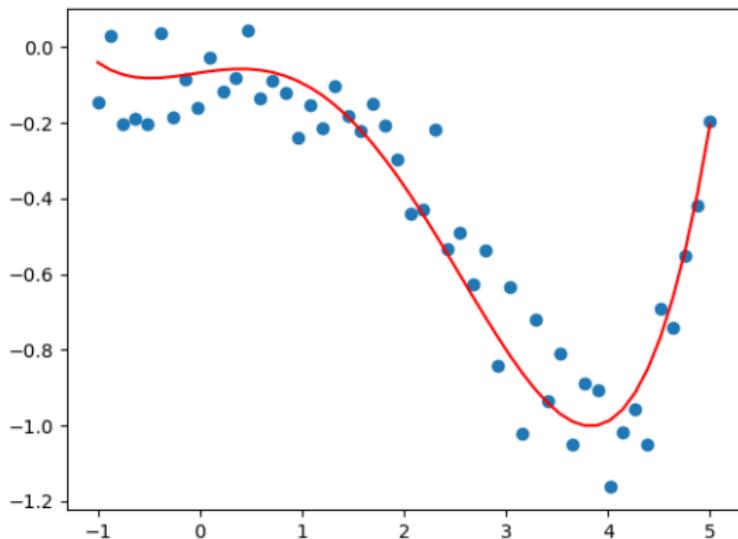
But why, I still don't get it?

- One application might be do to regression.
 - ▶ On peut faire de la regression avec ce framework.



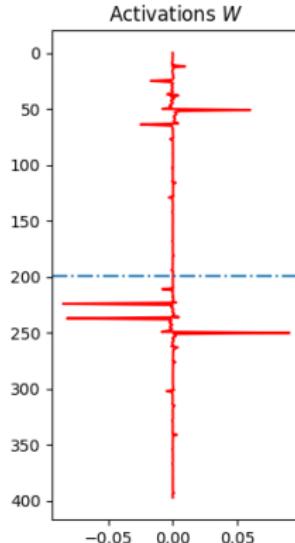
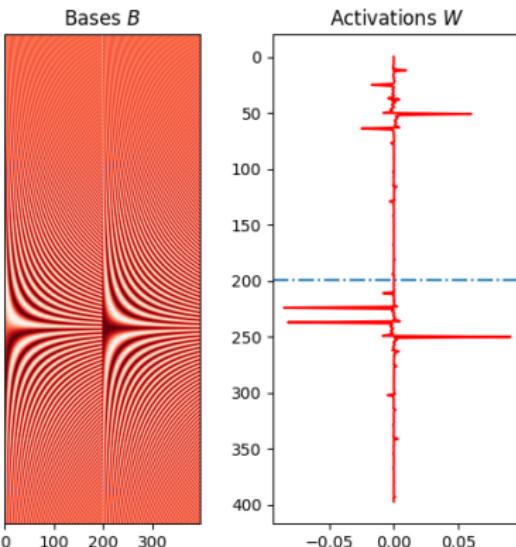
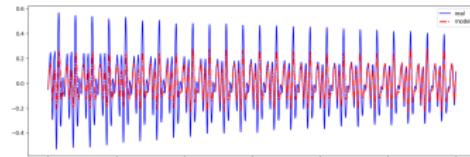
But why, I still don't get it?

- One application might be do to regression.
 - ▶ On peut faire de la regression avec ce framework.



Something a bit more real

- Modeling a guitar string Listen Real, Listen the Model
 - ▶ Modélisons un corde de guitare



Visualizing the model ingredients

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} b_1(1) & b_2(1) & \cdots & b_K(1) \\ b_1(2) & b_2(2) & \cdots & b_K(2) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(T) & b_2(T) & \cdots & b_K(T) \end{bmatrix}}_B \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}}_w$$

- $b_k(t)$ is the k 'th basis function in the basis (design) matrix B .
 - ▶ $b_k(t)$ est la fonction de base k 'eme dans la matrice de base.
- The output is a linear combination of the basis functions such that
 - ▶ La sortie du modèle est la combinaison linéaire des bases:

$$x_t = \sum_{k=1}^K w_k b_k(t) = w_1 b_1(t) + w_2 b_2(t) + \cdots + w_K b_K(t).$$

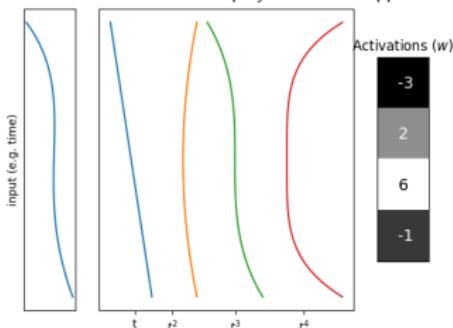
Visualizing the model ingredients

$$\underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_T \end{bmatrix}}_x = \underbrace{\begin{bmatrix} b_1(1) & b_2(1) & \cdots & b_K(1) \\ b_1(2) & b_2(2) & \cdots & b_K(2) \\ \vdots & \vdots & \ddots & \vdots \\ b_1(T) & b_2(T) & \cdots & b_K(T) \end{bmatrix}}_B \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}}_w$$

- $b_k(t)$ is the k 'th basis function in the basis (design) matrix B .
 - ▶ $b_k(t)$ est la fonction de base k 'eme dans la matrice de base.
- The output is a linear combination of the basis functions such that
 - ▶ La sortie du modèle est la combinaison linéaire des bases:

$$x_t = \sum_{k=1}^K w_k b_k(t) = w_1 b_1(t) + w_2 b_2(t) + \cdots + w_K b_K(t).$$

- Here's an example design matrix with polynomial basis functions. This particular choice is also called a Vandermonde matrix.
 - ▶ Voici un matrice de design exemplaire avec fonctions de bases polynomiel. On appelle ce choix la matrice Vandermonde.



Even autoregressive modeling

Autoregressive Modeling

$$\underbrace{\begin{bmatrix} x_{K+1} \\ x_{K+2} \\ \vdots \\ x_T \end{bmatrix}}_y = \underbrace{\begin{bmatrix} x_K & x_{K-1} & \dots & x_2 & x_1 \\ x_{K+1} & x_K & \dots & x_3 & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{t-2} & x_{t-3} & \dots & x_{t-K+2} & x_{t-K+1} \\ x_{t-1} & x_{t-2} & \dots & x_{t-K+1} & x_{t-K} \end{bmatrix}}_B \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}}_w$$

Even autoregressive modeling

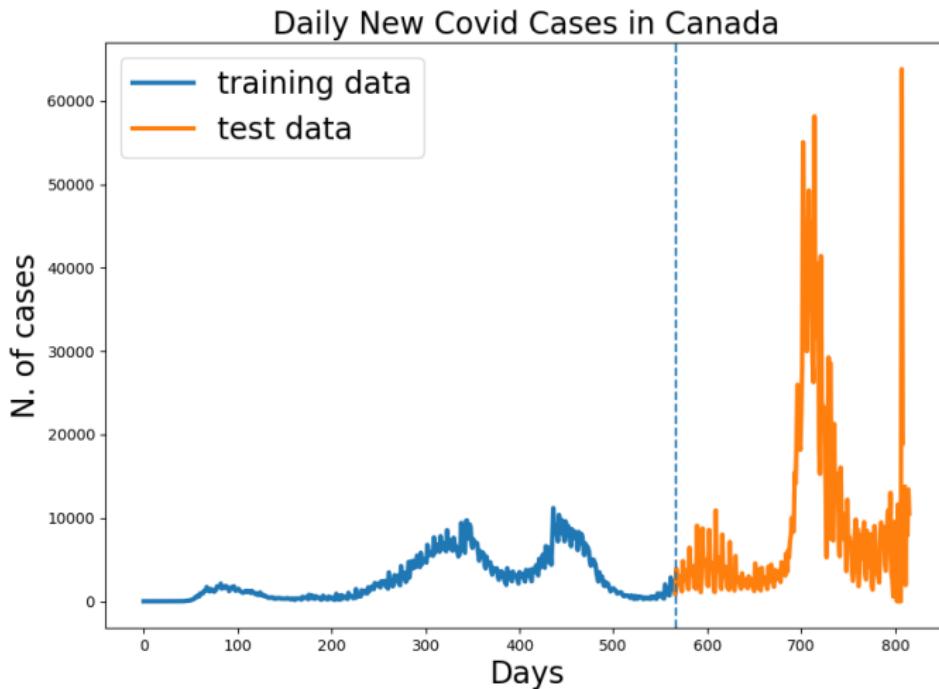
Autoregressive Modeling **LLM:Linear Language Model** (I am joking)

$$\underbrace{\begin{bmatrix} x_{K+1} \\ x_{K+2} \\ \vdots \\ x_T \end{bmatrix}}_y = \underbrace{\begin{bmatrix} x_K & x_{K-1} & \dots & x_2 & x_1 \\ x_{K+1} & x_K & \dots & x_3 & x_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{t-2} & x_{t-3} & \dots & x_{t-K+2} & x_{t-K+1} \\ x_{t-1} & x_{t-2} & \dots & x_{t-K+1} & x_{t-K} \end{bmatrix}}_B \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_K \end{bmatrix}}_w$$

Btw, do see that we have a series of convolutions? / En passant, vous voyez vous qu'on fait des convolutions?

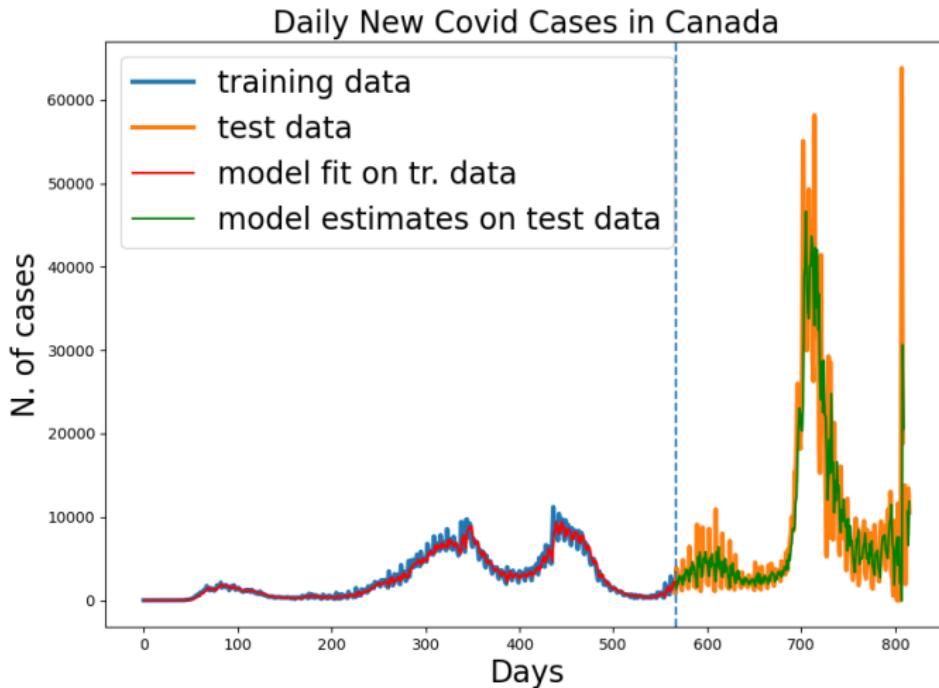
Real Real-life data

- We try to fit a regression model with autoregressive design matrix on nbr. of cases data with $K = 3$. / On utilise un matrice de design qui est autoregressive. On utilise un filtre de $K = 3$.



Real Real-life data

- We try to fit a regression model with autoregressive design matrix on nbr. of cases data with $K = 3$. / On utilise un matrice de design qui est autoregressive. On utilise un filtre de $K = 3$.



How to learn W ?

- Consider the following model (Linear Regression):
 - ▶ Considérons la modèle suivante (Regression Linéaire):

$$w_n \sim \mathcal{N}(w_n; 0, \sigma_0^2 I)$$

$$x_{t,n}|w_n \sim \mathcal{N}(x_t; B(t)w_n, \sigma^2 I)$$

n is the signal index, t is the time index / n est l'indice du temps, t est l'indice du data.

How to learn W ?

- Consider the following model (Linear Regression):
 - ▶ Considérons la modèle suivante (Regression Linéaire):

$$w_n \sim \mathcal{N}(w_n; 0, \sigma_0^2 I)$$

$$x_{t,n}|w_n \sim \mathcal{N}(x_t; B(t)w_n, \sigma^2 I)$$

n is the signal index, t is the time index / n est l'indice du temps, t est l'indice du data.

- Let's write the model likelihood / Écrivons le likelihood du modèle,

$$\begin{aligned}\mathcal{L} := \log p(x_{1:T,n}, w_n) &= \sum_t p(x_{t,n}|w_n) + \log p(w_n) \\ &= \sum_t \log \mathcal{N}(x_t; B(t)w_n, \sigma^2 I) + \log \mathcal{N}(w_n; 0, \sigma_0^2 I)\end{aligned}$$

How to learn W ?

- Consider the following model (Linear Regression):
 - ▶ Considérons la modèle suivante (Regression Linéaire):

$$w_n \sim \mathcal{N}(w_n; 0, \sigma_0^2 I)$$

$$x_{t,n}|w_n \sim \mathcal{N}(x_t; B(t)w_n, \sigma^2 I)$$

n is the signal index, t is the time index / n est l'indice du temps, t est l'indice du data.

- Let's write the model likelihood / Écrivons le likelihood du modèle,

$$\begin{aligned}\mathcal{L} := \log p(x_{1:T,n}, w_n) &= \sum_t p(x_{t,n}|w_n) + \log p(w_n) \\ &= \sum_t \log \mathcal{N}(x_t; B(t)w_n, \sigma^2 I) + \log \mathcal{N}(w_n; 0, \sigma_0^2 I)\end{aligned}$$

- What should we do next to estimate $w_{1:N}$?
 - ▶ Qu'est qu'on fait maintenant pour estimer $w_{1:N}$?

Finding the best w_n

- Now, we will take the gradient of \mathcal{L} with respect to w_n , set it equal to zero and solve for w_n . We switch to matrix-vector notation, and drop the n index to reduce clutter.

$$\begin{aligned}\mathcal{L} &\propto (Bw - x)^\top (Bw - x) \\ &= -\frac{1}{2\sigma^2} \left(w^\top B^\top B w - 2w^\top B^\top x + x^\top x \right) - \frac{1}{2\sigma_0^2} (w^\top w)\end{aligned}$$

Finding the best w_n

- Now, we will take the gradient of \mathcal{L} with respect to w_n , set it equal to zero and solve for w_n . We switch to matrix-vector notation, and drop the n index to reduce clutter.

$$\begin{aligned}\mathcal{L} &\propto (Bw - x)^\top (Bw - x) \\ &= -\frac{1}{2\sigma^2} \left(w^\top B^\top B w - 2w^\top B^\top x + x^\top x \right) - \frac{1}{2\sigma_0^2} (w^\top w)\end{aligned}$$

- And the gradient,

$$\frac{\partial \mathcal{L}}{\partial w} = -\frac{1}{\sigma^2} (B^\top B w - B^\top x) - \frac{1}{\sigma_0^2} w \quad (1)$$

Finding the best w_n

- Now, we will take the gradient of \mathcal{L} with respect to w_n , set it equal to zero and solve for w_n . We switch to matrix-vector notation, and drop the n index to reduce clutter.

$$\begin{aligned}\mathcal{L} &\propto (Bw - x)^\top (Bw - x) \\ &= -\frac{1}{2\sigma^2} \left(w^\top B^\top B w - 2w^\top B^\top x + x^\top x \right) - \frac{1}{2\sigma_0^2} (w^\top w)\end{aligned}$$

- And the gradient,

$$\frac{\partial \mathcal{L}}{\partial w} = -\frac{1}{\sigma^2} \left(B^\top B w - B^\top x \right) - \frac{1}{\sigma_0^2} w \quad (1)$$

- Solve for w ,

$$\begin{aligned}\frac{1}{\sigma^2} \left(B^\top B w - B^\top x \right) - \frac{1}{\sigma_0^2} w &= 0 \\ \left(B^\top B + \frac{\sigma^2}{\sigma_0^2} I \right) w &= B^\top x \\ \rightarrow \hat{w} &= \left(B^\top B + \frac{\sigma^2}{\sigma_0^2} I \right)^{-1} B^\top x\end{aligned}$$

The MAP solution for w_n

- Note this is the MAP solution for w_n for $n \in \{1, \dots, N\}$ that we saw before:
 - ▶ Notez que c'est la solution MAP pour w_n , $n \in \{1, \dots, N\}$:

$$\hat{w}_n = \left(B^\top B + \frac{\sigma^2}{\sigma_0^2} I \right)^{-1} B^\top x$$

The MAP solution for w_n

- Note this is the MAP solution for w_n for $n \in \{1, \dots, N\}$ that we saw before:
 - ▶ Notez que c'est la solution MAP pour w_n , $n \in \{1, \dots, N\}$:

$$\hat{w}_n = \left(B^\top B + \frac{\sigma^2}{\sigma_0^2} I \right)^{-1} B^\top x$$

- You can also easily show that the Maximum-Likelihood solution is very similar (Uniform Prior):
 - ▶ On peut aussi très facilement montrer que la solution Maximum-Likelihood est très similaire (si on utilise un prior uniforme):

$$\hat{w}_n = \left(B^\top B \right)^{-1} B^\top$$

The MAP solution for w_n

- Note this is the MAP solution for w_n for $n \in \{1, \dots, N\}$ that we saw before:
 - ▶ Notez que c'est la solution MAP pour w_n , $n \in \{1, \dots, N\}$:

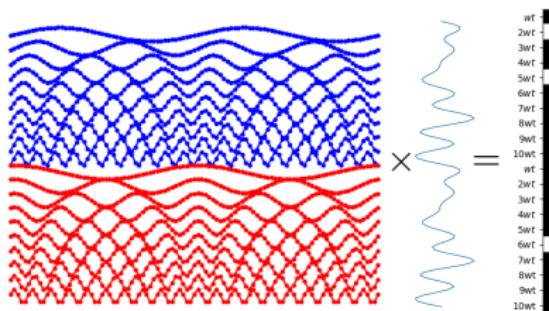
$$\hat{w}_n = \left(B^\top B + \frac{\sigma^2}{\sigma_0^2} I \right)^{-1} B^\top x$$

- You can also easily show that the Maximum-Likelihood solution is very similar (Uniform Prior):
 - ▶ On peut aussi très facilement montrer que la solution Maximum-Likelihood est très similaire (si on utilise un prior uniforme):

$$\hat{w}_n = \underbrace{\left(B^\top B \right)^{-1}}_{B^\dagger} B^\top x$$

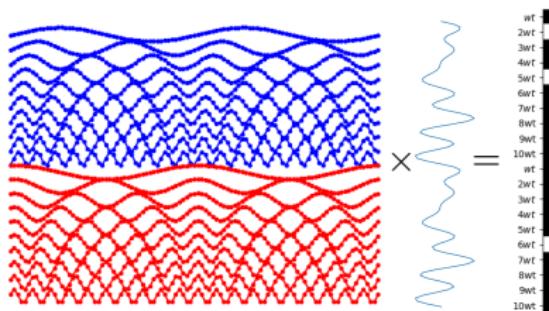
Ok, but how about the DFT stuff we talked about last week?

- Ok, but were we doing this last week? Is this optimal?
- D'accord, on faisait ça la semaine dernière? Est-ce optimale?



Ok, but how about the DFT stuff we talked about last week?

- Ok, but were we doing this last week? Is this optimal?
- D'accord, on faisait ça la semaine dernière? Est-ce optimale?



- Yes!/Oui!

Fourier Transform Maximizes Gaussian Likelihood

- Let's see / Voyons:

$$\hat{w}_n = (B^\top B)^{-1} B^\top$$

Fourier Transform Maximizes Gaussian Likelihood

- Let's see / Voyons:

$$\hat{w}_n = (B^\top B)^{-1} B^\top$$

Fourier Transform Maximizes Gaussian Likelihood

- Let's see / Voyons:

$$\hat{w}_n = \underbrace{(B^\top B)^{-1} B^\top}_{B^\dagger} x$$

- Note that in this case $B = F$, and $F^*F = FF^* = I$. Let's substitute.
 - F^* , is the Hermitian adjoint (transpose and flip the imaginary part)
- Notez que dans ce cas-ci $B = F$, et $F^*F = FF^* = I$. Substitutions:
 - F^* , est le Hermitian adjoint (on transpose et on flippe la partie imaginaire)
- DFT Matrix is orthogonal so,

$$\hat{w}_n = Fx$$

Fourier Transform Maximizes Gaussian Likelihood

- Let's see / Voyons:

$$\hat{w}_n = \underbrace{(B^\top B)^{-1} B^\top}_{B^\dagger} x$$

- Note that in this case $B = F$, and $F^*F = FF^* = I$. Let's substitute.

- ▶ F^* , is the Hermitian adjoint (transpose and flip the imaginary part)
- Notez que dans ce cas-ci $B = F$, et $F^*F = FF^* = I$. Substitutions:

- ▶ F^* , est le Hermitian adjoint (on transpose et on flippe la partie imaginaire)

- DFT Matrix is orthogonal so,

$$\hat{w}_n = Fx$$

- We can then deduce that DFT maximizes the Gaussian Likelihood under this linear regression / decomposition model! (or minimizes ℓ_2 error)
 - ▶ On peut alors déduire que DFT maximise le likelihood Gaussian sous ce modèle. (ou il minimise l'erreur ℓ_2 .)

Table of Contents

The Decomposition Framework

Fixed Basis Decompositions (Linear Regression)

Learnable Basis Decompositions

Principal Component Analysis

Independent Component Analysis

Non-Negative Matrix Factorization (NMF)

Let's learn B too!

- Note that earlier we were only learning the activations W for a fixed basis matrix B :

- ▶ On apprenait juste les activations W pour des bases fixes B :

$$\min_W \|X - BW\|$$

- ▶ But, we can learn B too!
 - ▶ Mais, on peut apprendre B aussi!

$$\min_{B,W} \|X - BW\|$$

But how?

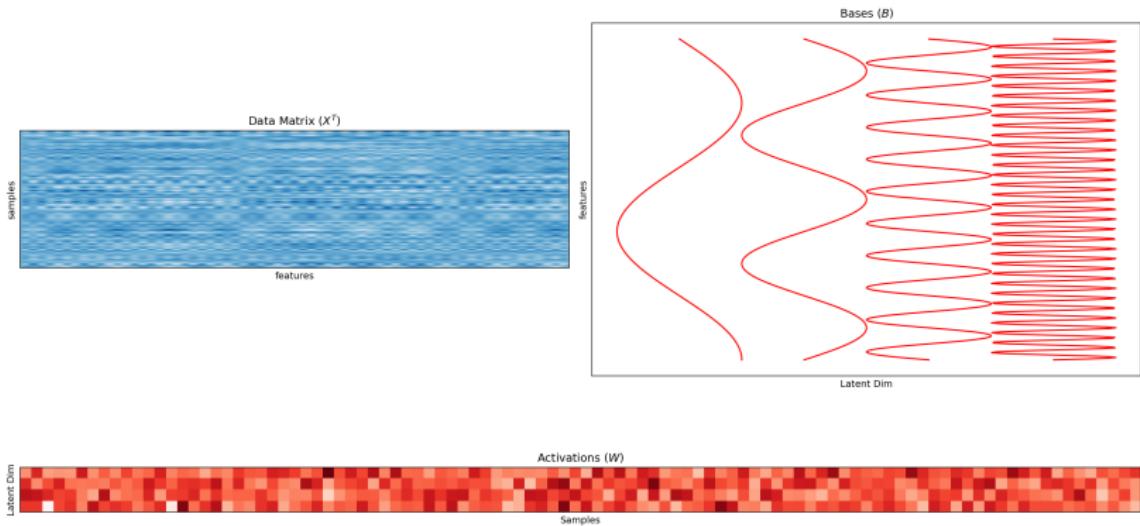
- We can alternate the least squares solution such that,
 - ▶ On peut juste alterner entre les solutions least-squares,

Algorithm 1 Alternating Least Squares

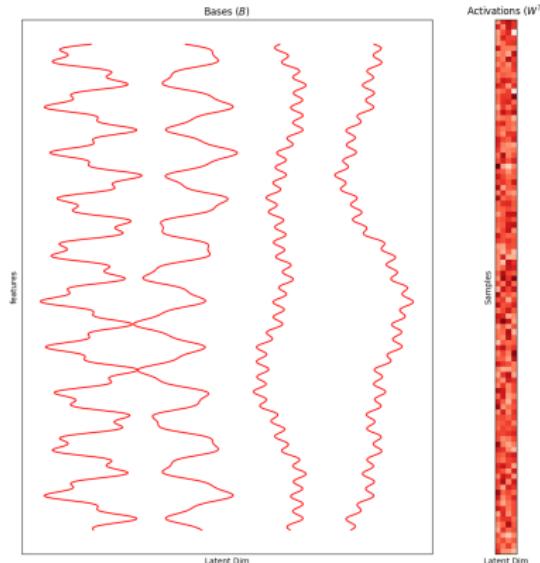
```
1: procedure ALTERNATING LEAST SQUARES
   Input: Input Data Matrix  $X$ . Threshold value  $\epsilon$ .
   Output: Estimated Basis and Activation Matrices  $\hat{B}$ ,  $\hat{W}$ .
2:   Initialize  $\hat{B}$ ,  $\hat{W}$ .
3:   while  $\|X - \hat{B}\hat{W}\| \geq \epsilon$  do
4:      $\hat{W} = \hat{B}^\dagger X$ 
5:      $\hat{B} = X\hat{W}^\dagger$ 
6:   end while
7: end procedure
```

Alternating Least Squares Dataset

- Let's try alternating least squares on this dataset
 - Essayons cette méthode sur ce dataset



The result



- Kinda good, but we can do better.
 - ▶ Ça fait quelque chose, mais pas idéale.

Table of Contents

The Decomposition Framework

Fixed Basis Decompositions (Linear Regression)

Learnable Basis Decompositions

Principal Component Analysis

Independent Component Analysis

Non-Negative Matrix Factorization (NMF)

Principal Component Analysis

- The goal is to find uncorrelated latent components W .
 - ▶ Le but est de trouver des composants W qui ne sont pas co-reliés.

Principal Component Analysis

- The goal is to find uncorrelated latent components W .
 - ▶ Le but est de trouver des composants W qui ne sont pas co-reliés.
- We want to find an orthogonal transformation $B^\top B = I$, so that $W = B^\top X$.
 - ▶ On veut trouver une transformation orthogonale.
- Let's calculate:

$$\begin{aligned}\text{covar}(w) &= \text{covar}(B^\top x) \\ &= B^\top \underbrace{\mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top]}_{:= C} B\end{aligned}$$

Principal Component Analysis

- The goal is to find uncorrelated latent components W .
 - ▶ Le but est de trouver des composants W qui ne sont pas co-reliés.
- We want to find an orthogonal transformation $B^\top B = I$, so that $W = B^\top X$.
 - ▶ On veut trouver une transformation orthogonale.
- Let's calculate:

$$\begin{aligned}\text{covar}(w) &= \text{covar}(B^\top x) \\ &= B^\top \underbrace{\mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top]}_{:= C} B\end{aligned}$$

- So basically want to solve for B such that $B^\top C B = I$, that is C is whitened.
 - ▶ Dans le fond on veut blanchir C .

Principal Component Analysis

- The goal is to find uncorrelated latent components W .
 - ▶ Le but est de trouver des composants W qui ne sont pas co-reliés.
- We want to find an orthogonal transformation $B^\top B = I$, so that $W = B^\top X$.
 - ▶ On veut trouver une transformation orthogonale.
- Let's calculate:

$$\begin{aligned}\text{covar}(w) &= \text{covar}(B^\top x) \\ &= B^\top \underbrace{\mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top]}_{:= C} B\end{aligned}$$

- So basically want to solve for B such that $B^\top C B = I$, that is C is whitened.
 - ▶ Dans le fond on veut blanchir C .
- Any ideas? (Ei.. SV.. ?)

Eigenvectors to the rescue

- Consider the SVD of C , s.t. $C = U\Sigma U^\top$. (**Same as eigenvalue decomp. why?**)
 - ▶ Considérons le SVD de C .

Eigenvectors to the rescue

- Consider the SVD of C , s.t. $C = U\Sigma U^\top$. (**Same as eigenvalue decomp. why?**)
 - ▶ Considérons le SVD de C .
- Let's substitute. Can you see the solution now? / Voyez-vous la solution?

$$B^\top U \Sigma U^\top B = I$$

- We can easily see that $B = U\Sigma^{-1/2}$ does the job!
 - ▶ On a trouvé la solution $B = U\Sigma^{-1/2}$!

Eigenvectors to the rescue

- Consider the SVD of C , s.t. $C = U\Sigma U^\top$. (**Same as eigenvalue decomp. why?**)
 - ▶ Considérons le SVD de C .
- Let's substitute. Can you see the solution now? / Voyez-vous la solution?

$$B^\top U \Sigma U^\top B = I$$

- We can easily see that $B = U\Sigma^{-1/2}$ does the job!
 - ▶ On a trouvé la solution $B = U\Sigma^{-1/2}$!
- The columns of U are the eigenvectors of C !
 - ▶ Les colonnes de U sont les vecteurs propres de C .

A note on variance

- Note that this way we maximize the variance along the direction of b_1 . / Cette solution maximise la variance sur la direction de b_1 .

$$\mathcal{V} := \text{var}(b_1^\top x) = b_1^\top \mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top] b_1$$

- Let's maximize this variance such that $b_1^\top b_1 = 1$. / Maximisons la variance telle que b_1 est de norme unitaire.

$$\begin{aligned}\mathcal{V} &= b_1^\top C b_1 - \lambda b_1^\top b_1 \\ \frac{\partial \mathcal{V}}{\partial b_1} &= 2C b_1 - \lambda b_1 \\ \rightarrow C b_1 &= \lambda b_1\end{aligned}$$

- So, we have the definition of an eigenvector... / Donc c'est la définition du vecteur propre de C .

A note on variance

- Note that this way we maximize the variance along the direction of b_1 . / Cette solution maximise la variance sur la direction de b_1 .

$$\mathcal{V} := \text{var}(b_1^\top x) = b_1^\top \mathbb{E}[(x - \mathbb{E}[x])(x - \mathbb{E}[x])^\top] b_1$$

- Let's maximize this variance such that $b_1^\top b_1 = 1$. / Maximisons la variance telle que b_1 est de norme unitaire.

$$\begin{aligned}\mathcal{V} &= b_1^\top C b_1 - \lambda b_1^\top b_1 \\ \frac{\partial \mathcal{V}}{\partial b_1} &= 2C b_1 - \lambda b_1 \\ \rightarrow C b_1 &= \lambda b_1\end{aligned}$$

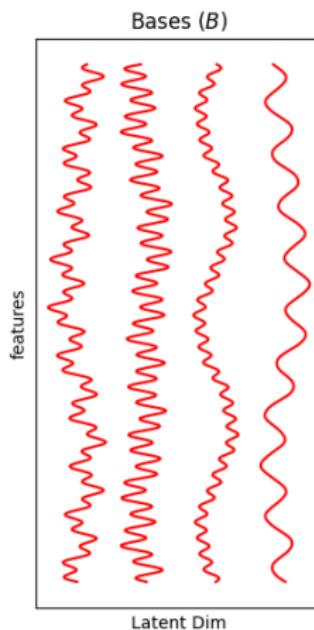
- So, we have the definition of an eigenvector... / Donc c'est la définition du vecteur propre de C .
- Similarly the other principal components are found..
 - ▶ Similairement, les autres composants principaux sont trouvés..

The recipe for PCA

- $X - \mathbb{E}[x] = U\Sigma V^\top$
- $(X - \mathbb{E}[x])(X - \mathbb{E}[x])^\top = C = U\Sigma^2 U^\top.$
- We said that we need the eigenvectors of C , which are the columns of U . / On a besoin de calculer les vecteurs propres de C , qui sont les colonnes de U .
- We also note that the left singular vectors of $X - \mathbb{E}[x]$ also give the same result.
 - ▶ On note aussi que les vecteurs singulaires gauche de $X - \mathbb{E}[x]$ donnent la même résultat.

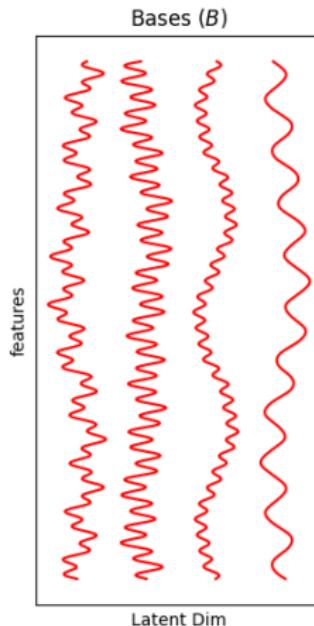
PCA on our sinusoid basis problem

- A bit better!



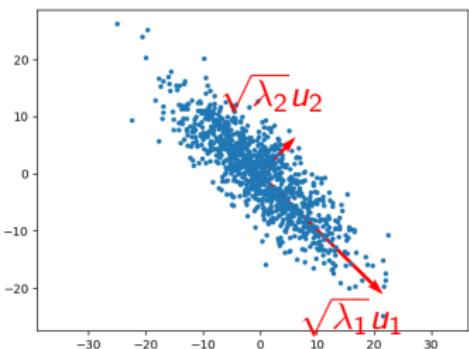
PCA on our sinusoid basis problem

- A bit better!

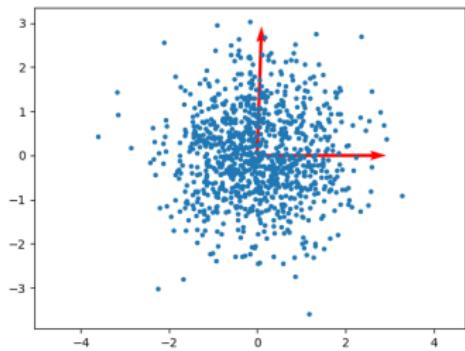


- We can improve this! (more on this later ICA)
 - ▶ On peut améliorer ça. (On verra)

Interpretation of PCA



$B^\top X$
→



Note that $B^\top = \text{diag}([\sqrt{\lambda_1}, \sqrt{\lambda_2}])^{-1} U^\top$.

Dimensionality Reduction with PCA

- Note that PCA makes the following decomposition / On fait la décomposition suivante:

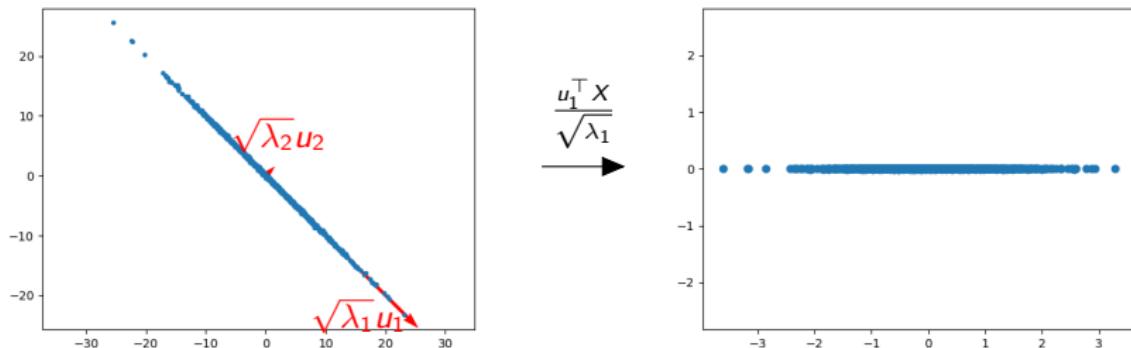
$$\text{var}(X) = \sum_{k=1}^K \lambda_k u_k u_k^\top$$

Dimensionality Reduction with PCA

- Note that PCA makes the following decomposition / On fait la décomposition suivante:

$$\text{var}(X) = \sum_{k=1}^K \lambda_k u_k u_k^\top$$

- Let's consider the following case / Considérons le cas suivant:

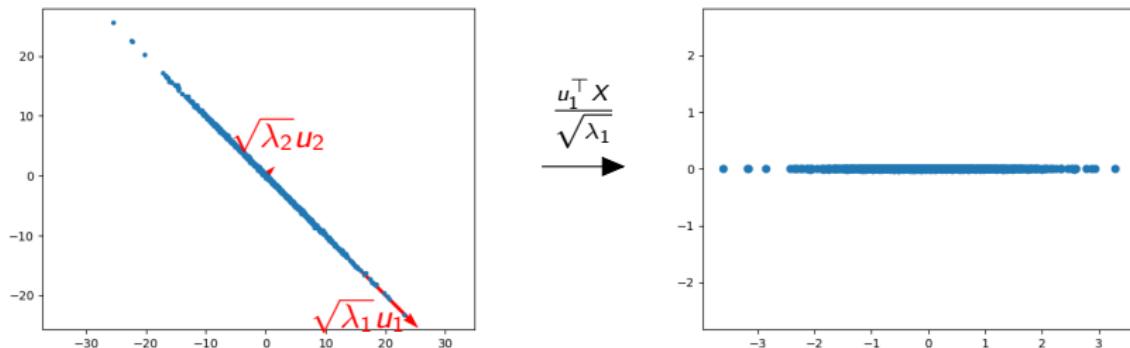


Dimensionality Reduction with PCA

- Note that PCA makes the following decomposition / On fait la décomposition suivante:

$$\text{var}(X) = \sum_{k=1}^K \lambda_k u_k u_k^\top$$

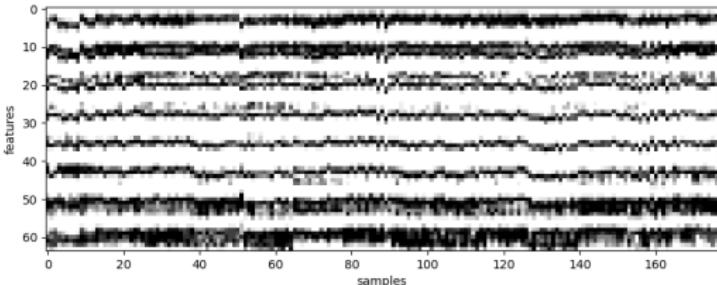
- Let's consider the following case / Considérons le cas suivant:



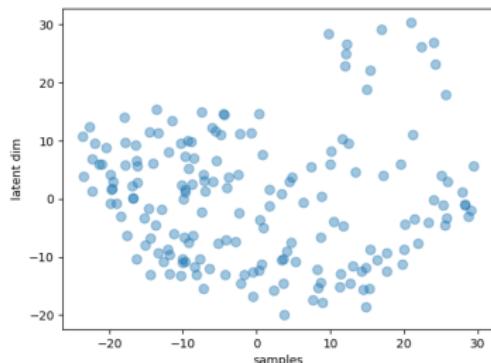
- $\lambda_1 = 10, \lambda_2 = 0.1$. Most of the variance is along one direction. We can only use one dim. / La variance est sur une direction. On peut s'en débarrasser d'une direction.

Embedding digits in 2 dimensions

- We only keep two dimensions / On garde juste 2 dimensions

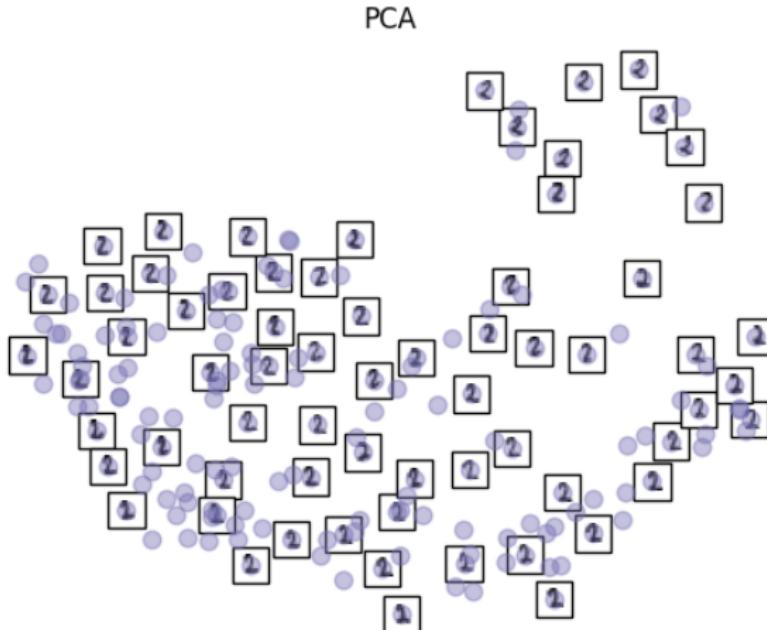


$$B^\top(X - \mathbb{E}[x])$$



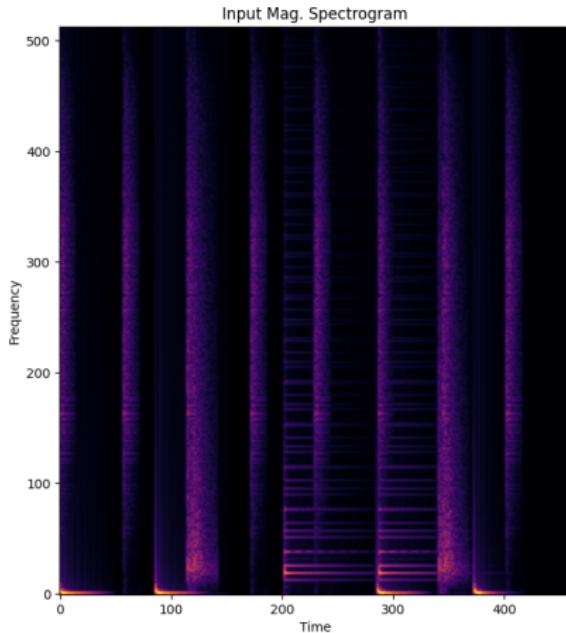
Embedding digits in 2 dimensions

- We only keep two dimensions / On garde juste 2 dimensions

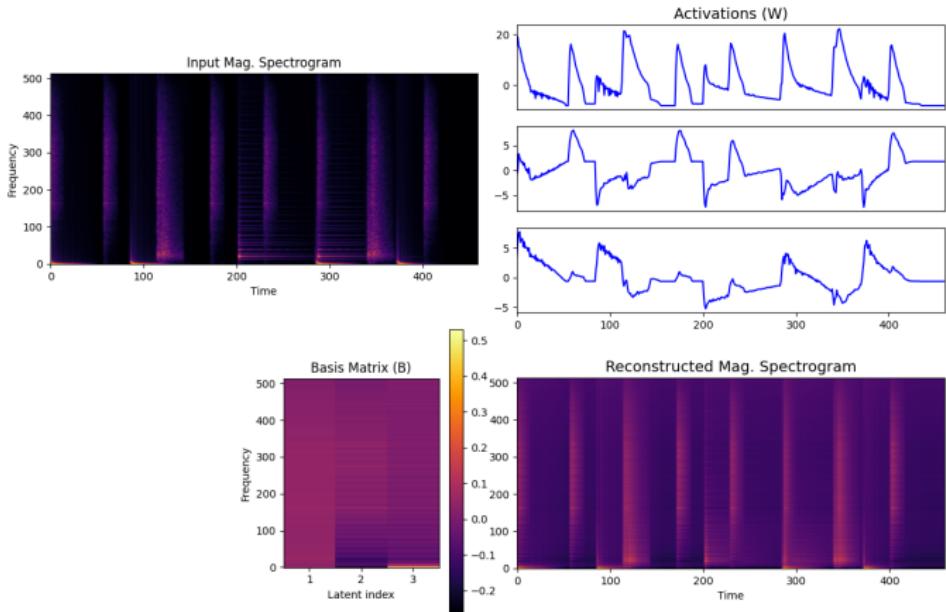


Embedding spectra

- Let's embed this spectrogram into a 3 dim. space / On va embedder ce spectrogram dans un 3 dim. espace. Listen



Embedding spectra with PCA



PCA on time-series

- Let's apply PCA on local windows of a time series / Appliqueons PCA on des fenetres d'un time series

$$x_1, x_2, \dots, x_T$$

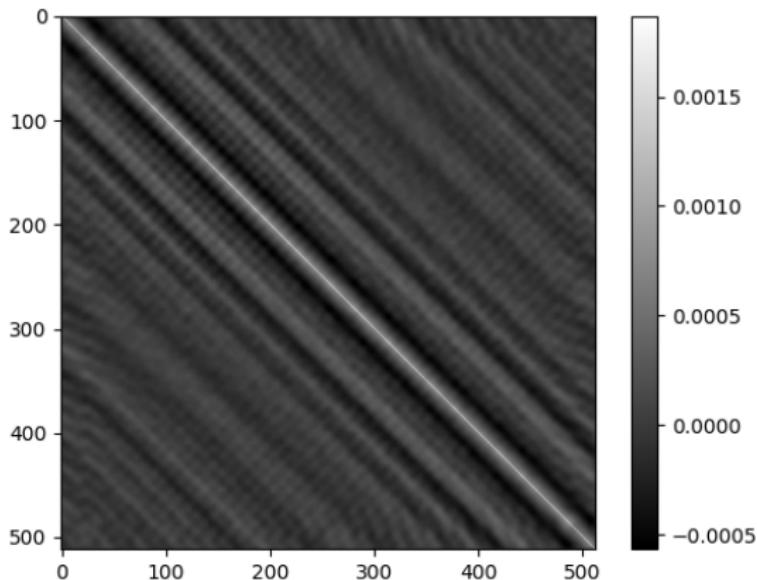
- Pack in a data matrix as follows

$$X = \begin{bmatrix} x_1 & x_{1+s} & x_{1+2s} & \dots \\ x_2 & x_{2+s} & x_{2+2s} & \dots \\ \vdots & \vdots & \vdots & \ddots \\ x_N & x_{N+s} & x_{N+2s} & \dots \end{bmatrix}$$

- Note that if we do $W = FX$, this is equal to Short-Time-Fourier-Transform that we saw last week.
 - ▶ Notez que si on utilise les bases de Fourier ça donne STFT.
 - ▶ s is the hopsize we saw for STFT. / s est le hopsize, même que STFT.

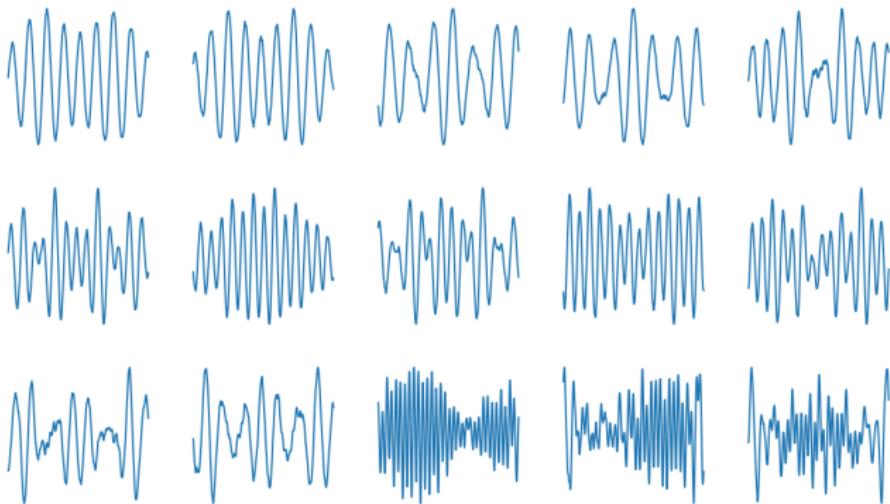
But let's do PCA instead

- Here's the covariance matrix



- Seems like we have high covariance in the neighborhood, then some periodicity.
 - Haut covariance locale, et un peu de périodicité.

Sinusoids!



Listen

So, it seems sinusoidal bases are kinda statistically optimal for local covariance as well.. / Les bases sinusoids sont optimale si on a une covariance locale!.

Table of Contents

The Decomposition Framework

Fixed Basis Decompositions (Linear Regression)

Learnable Basis Decompositions

Principal Component Analysis

Independent Component Analysis

Non-Negative Matrix Factorization (NMF)

Independent Component Analysis (ICA)

- ICA estimates a square mixing matrix $A \in \mathbb{R}^{K \times K}$, such that,
 - ▶ ICA estime un matrice carré A , telle que,

$$x = As + n$$

the elements of $s \in \mathbb{R}^K$ are statistically independent. / les éléments du vecteur s sont statisquement indépendent.

- We want to achieve $p(s) = p(s_1)p(s_2)\dots p(s_K)$.
 - ▶ On veut que le probabilité joint $p(s)$ se factorise.

Methods to solve ICA (high-level)

- Non linear decorrelation $\mathbb{E}[f(s_i)g(s_j)]$, for fixed f, g .
 - ▶ Décorrélation non-linéaire pour $\mathbb{E}[f(s_i)g(s_j)]$, f, g sont fixes..
 - ▶ Cichocki-Unbehauen algorithm

Methods to solve ICA (high-level)

- Non linear decorrelation $\mathbb{E}[f(s_i)g(s_j)]$, for fixed f, g .
 - ▶ Décorrélation non-linéaire pour $\mathbb{E}[f(s_i)g(s_j)]$, f, g sont fixes..
 - ▶ Cichocki-Unbehauen algorithm
- Higher order diagonalization.
 - ▶ Diagonalize

$$Q(s) := \mathbb{E}[s_i s_j s_k s_l] - \mathbb{E}[s_i s_j] \mathbb{E}[s_k s_l] - \mathbb{E}[s_i s_k] \mathbb{E}[s_j s_l] - \mathbb{E}[s_i s_l] \mathbb{E}[s_j s_k]$$

- ▶ Remember PCA diagonalizes $\mathbb{E}[s_i s_j]$.

Methods to solve ICA (high-level)

- Non linear decorrelation $\mathbb{E}[f(s_i)g(s_j)]$, for fixed f, g .
 - ▶ Décorrélation non-linéaire pour $\mathbb{E}[f(s_i)g(s_j)]$, f, g sont fixes..
 - ▶ Cichocki-Unbehauen algorithm
- Higher order diagonalization.
 - ▶ Diagonalize

$$Q(s) := \mathbb{E}[s_i s_j s_k s_l] - \mathbb{E}[s_i s_j] \mathbb{E}[s_k s_l] - \mathbb{E}[s_i s_k] \mathbb{E}[s_j s_l] - \mathbb{E}[s_i s_l] \mathbb{E}[s_j s_k]$$

- ▶ Remember PCA diagonalizes $\mathbb{E}[s_i s_j]$.
- Info-theoretic approach

$$\min \text{KL}(p(s) \| p(s_1)p(s_2)\dots p(s_K)) = \min \int p(s) \log \frac{p(s)}{\prod_k p(s_k)}$$

- ▶ We try to make the product of marginals become the joint / on essaie de faire la produit de marginales égale à joint.

Methods to solve ICA (high-level)

- Non linear decorrelation $\mathbb{E}[f(s_i)g(s_j)]$, for fixed f, g .
 - ▶ Décorrélation non-linéaire pour $\mathbb{E}[f(s_i)g(s_j)]$, f, g sont fixes..
 - ▶ Cichocki-Unbehauen algorithm
- Higher order diagonalization.
 - ▶ Diagonalize

$$Q(s) := \mathbb{E}[s_i s_j s_k s_l] - \mathbb{E}[s_i s_j] \mathbb{E}[s_k s_l] - \mathbb{E}[s_i s_k] \mathbb{E}[s_j s_l] - \mathbb{E}[s_i s_l] \mathbb{E}[s_j s_k]$$

- ▶ Remember PCA diagonalizes $\mathbb{E}[s_i s_j]$.
- Info-theoretic approach

$$\min \text{KL}(p(s) \| p(s_1)p(s_2)\dots p(s_K)) = \min \int p(s) \log \frac{p(s)}{\prod_k p(s_k)}$$

- ▶ We try to make the product of marginals become the joint / on essaie de faire la produit de marginales égale à joint.
- More: FastICA, Neural Nets, Negentropy (Measure of non-gaussianity), More...

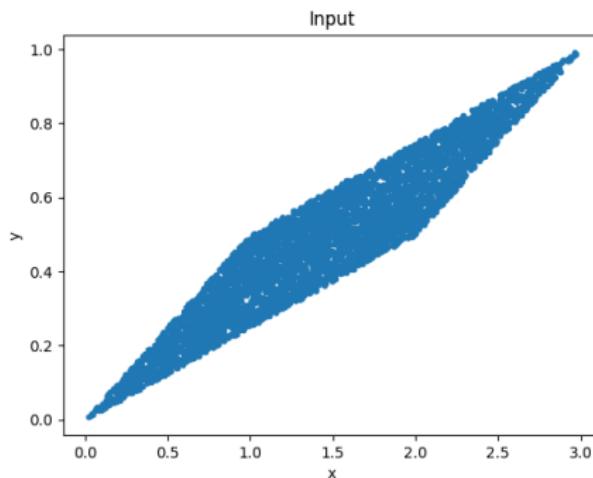
PCA vs ICA

- Let's consider this toy example

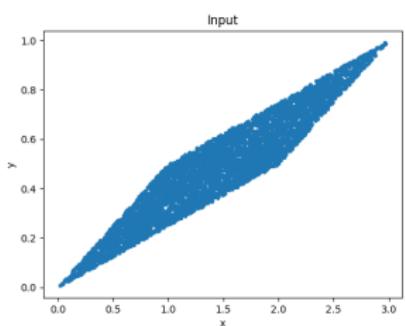
$$r_1, r_2 \sim \mathcal{U}(0, 1)$$

$$x = r_1 + r_2$$

$$y = 2r_1 + r_2$$

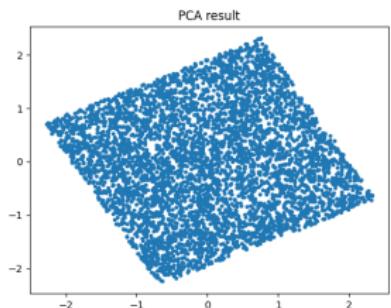


PCA vs ICA

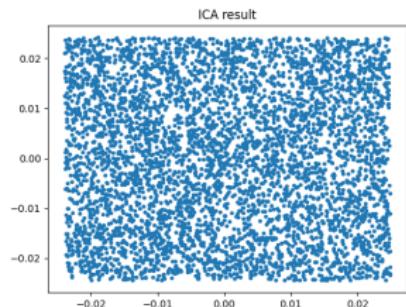
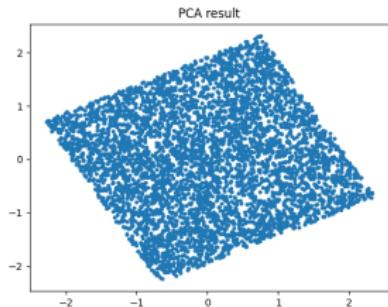
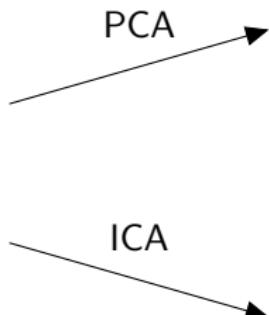
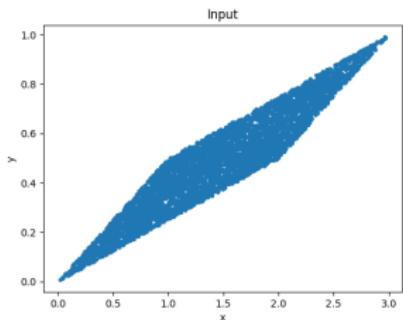


PCA

A black arrow points from the "Input" plot towards the "PCA result" plot, indicating the transformation process.



PCA vs ICA



PCA's uncorrelatedness criterion is not enough in this case / La décorrélation de PCA n'est pas suffisante ici!

PCA on steroids

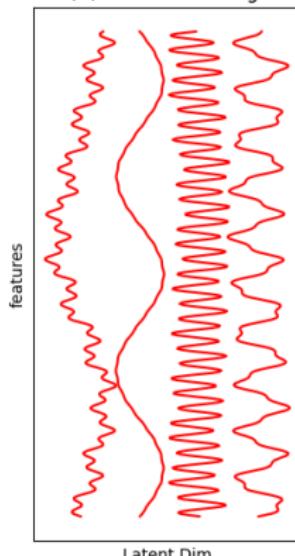
- We were doing the decomposition / On faisait la décomposition,

$$X = BW$$

- We can apply ICA to obtain / On peut appliquer ICA pour obtenir

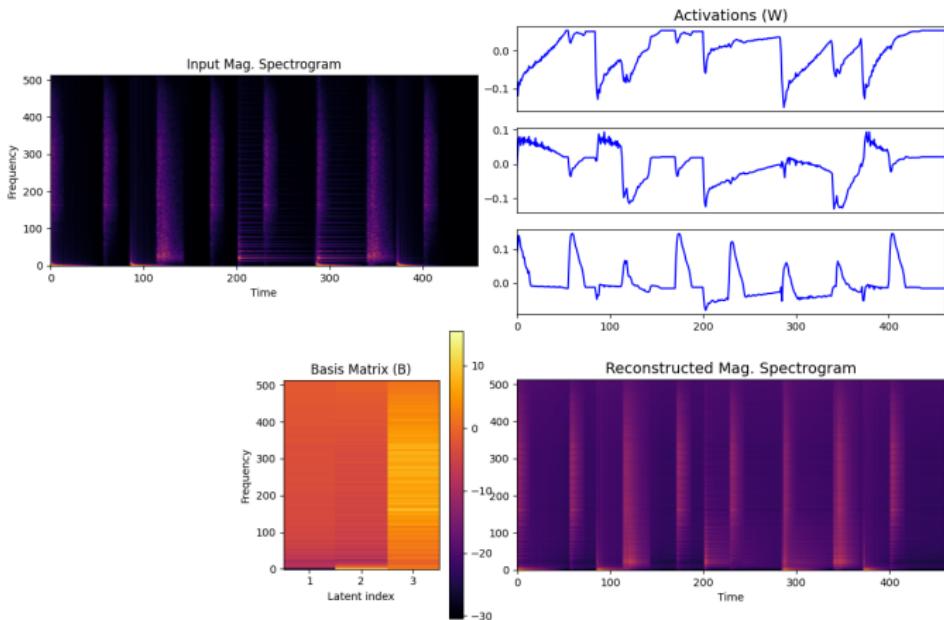
$$X = BAS = \tilde{B}S$$

Bases (B) after ICA mixing matrix



- Closer to sinusoids!

Embedding spectra with ICA



Bit better but we can do better..

ICA Summary

- PCA assumes that everything is Gaussian. (For Gaussian data it does return independent dimensions)
 - ▶ PCA suppose que le monde est Gaussienne.
- iCA does not assume Gaussian, and try to achieve independence.
 - ▶ ICA essaie d'obtenir l'independence.
- Most ICA estimators are approximate
 - ▶ La majorité des estimateurs ICA sont approximatives.
- We don't have an important ordering of components, so no dim. reduction
 - ▶ On n'a pas un ordering des composant, donc on ne peut faire une reduction de dimensions.
 - ▶ We can however combine it PCA to improve it. / On peut le combiner avec PCA pour l'améliorer.

Table of Contents

The Decomposition Framework

Fixed Basis Decompositions (Linear Regression)

Learnable Basis Decompositions

Principal Component Analysis

Independent Component Analysis

Non-Negative Matrix Factorization (NMF)

Non-Negative Matrix Factorization

- We want to again optimize for B, W , but for $W \geq 0, W \geq 0$. i

$$\begin{aligned} & \min_{B,W} \|X - BW\| \\ & \text{s.t. } B \geq 0, W \geq 0. \end{aligned}$$

- First proposed in 1999 Nature paper. / Proposé dans un papier Nature en 1999.
- Works pretty well on data non-négative. Fonctionne magiquement bien sur le data non-negative.
- We often work with non-negative data. (counts, pixels, energy...) / On travaille souvent avec du data non-négative.

Non-Negative Matrix Factorization

- We want to again optimize for B, W , but for $W \geq 0, W \geq 0$. i

$$\begin{aligned} & \min_{B, W} \|X - BW\| \\ & \text{s.t. } B \geq 0, W \geq 0. \end{aligned}$$

- First proposed in 1999 Nature paper. / Proposé dans un papier Nature en 1999.
- Works pretty well on data non-négative. Fonctionne magiquement bien sur le data non-negative.
- We often work with non-negative data. (counts, pixels, energy...) / On travaille souvent avec du data non-négative.
- If we have negative values in our estimates, they cancel out, harm interpretability. / Si on a des values négatives dans les parameters, ça nuit l'interprétabilité.

But how?

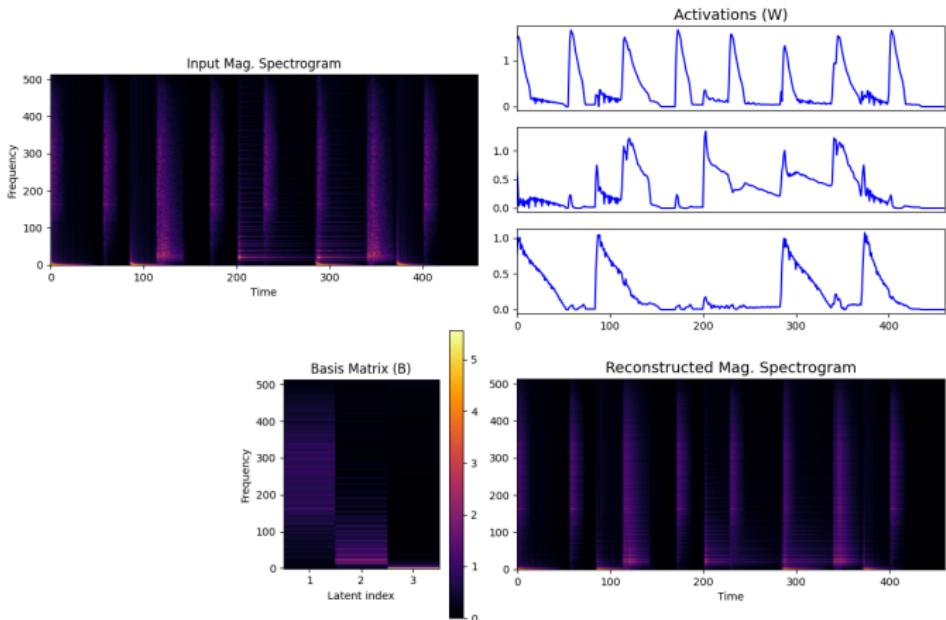
- We can alternate the least squares solutions and also project such,
 - ▶ On peut juste alterner entre les solutions least-squares avec une addition des projections,

Algorithm 2 Alternating Least Squares for NMF

```
1: procedure ALTERNATING LEAST SQUARES FOR NMF
  Input: Input Data Matrix  $X$ . Threshold value  $\epsilon$ .
  Output: Estimated Basis and Activation Matrices  $\hat{B}$ ,  $\hat{W}$ .
2:   Initialize  $\hat{B} \geq 0$ ,  $\hat{W} \geq 0$ .
3:   while  $\|X - \hat{B}\hat{W}\| \geq \epsilon$  do
4:      $\hat{W} = \hat{B}^\dagger X$ ;  $\hat{W} = \max(0, \hat{W})$ 
5:      $\hat{B} = X\hat{W}^\dagger$ ;  $\hat{B} = \max(0, \hat{B})$ 
6:   end while
7: end procedure
```

- There are also other algos. (e.g. Multiplicative Updates, probabilistic versions..)
 - ▶ Y a des autres algos. aussi.

NMF to rescue



PCA, NMF or ICA?

- It depends. / Ça depends.
- PCA is great for dim. reduction / PCA est très utile pour réduire la dimensionnalité.
- ICA gives more sparse/independent embeddings / ICA donne des embeddings plus parsimonieux.
- NMF gives interpretable results, but only for non-negative / NMF donne des résultats interprétables, mais juste pour des données non-négatives.

Recap

- We have introduced a framework that handles fixed basis regression, and learnable-basis regression.
 - ▶ On a introduit un framework qui peut gérer la regression avec des bases fixés, et regression avec des bases apprises.
- We have talked about very important latent variable methods such as PCA, ICA, NMF.
 - ▶ On a parlé des méthodes importants de variables latents comme PCA, ICA et NMF.

Suggested Reading

- Chapters 4, 12, Bishop
- The NMF Nature paper
<https://www.nature.com/articles/44565/>
- Eigenfaces <https://en.wikipedia.org/wiki/Eigenface>

Next Week

- What if we want to learn non-linear embeddings? Manifold methods!
 - ▶ Qu'est-ce qu'on fait si on veut apprendre des embeddings non-linéaires? Méthodes de manifolds!
- And Classification!