

# Adaptation de domaines : application à la classification d'images aériennes historiques

Augustin Gagnon et Amaury Zarzelli

Sous la supervision de Clément Mallet et Sébastien Giordano

1er Février 2017



# Sommaire

## 1 Présentation de la problématique

- Cadre général
- Illustration du problème

## 2 Méthode employée

- Présentation
- Utilisation

## 3 Résultats obtenus

- Estimation de l'efficacité
- Estimation de l'utilité

# Définitions

**Classification** : opération consistant à "ranger" des objets dans des catégories.

**Classification supervisée** : type de classification dans lequel on connaît les catégories *a priori* (*via* des échantillons). Apprentissage => prédictions.

En télédétection, on classe **des pixels** ou des segments d'image.

**Adaptation de domaines** : méthode permettant de classifier une image à partir de l'apprentissage réalisé sur une autre image (ou un ensemble).

# Objectif de recherche

**Le but :** classifier des images aériennes d'archive de l'IGN.

Actuellement, 2 méthodes : **classification date à date** et apprentissage à partir de la dernière date.

D'après Giordano *et al* (2017), la première est la plus efficace.

**Problème :** sa mise en place est longue et coûteuse.

**Solution possible : adaptation de domaine**

adapter la méthode proposée par Tuia *et al* (2016) pour des données multi-modales.

# Nos images de test



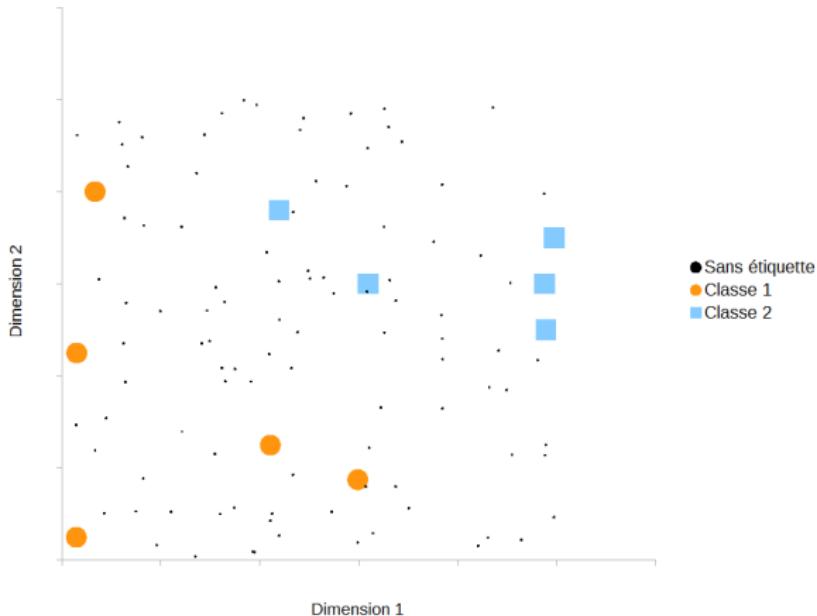
Prise de vue de 1966



Prise de vue de 1989

# Illustration du problème

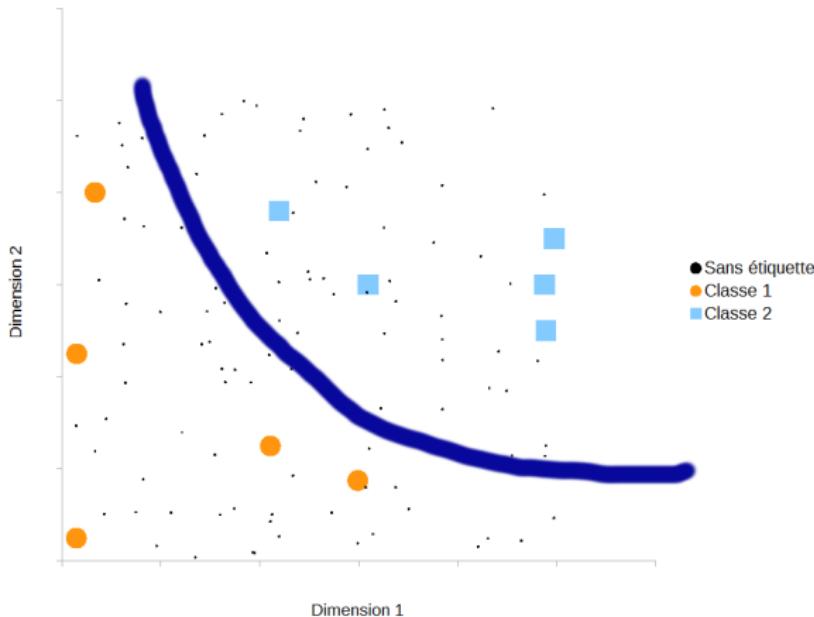
Dispersion pour l'image 1



Un exemple de distribution possible dans un espace à 2 dimensions  
Carrés et ronds : données d'apprentissage

# Illustration du problème

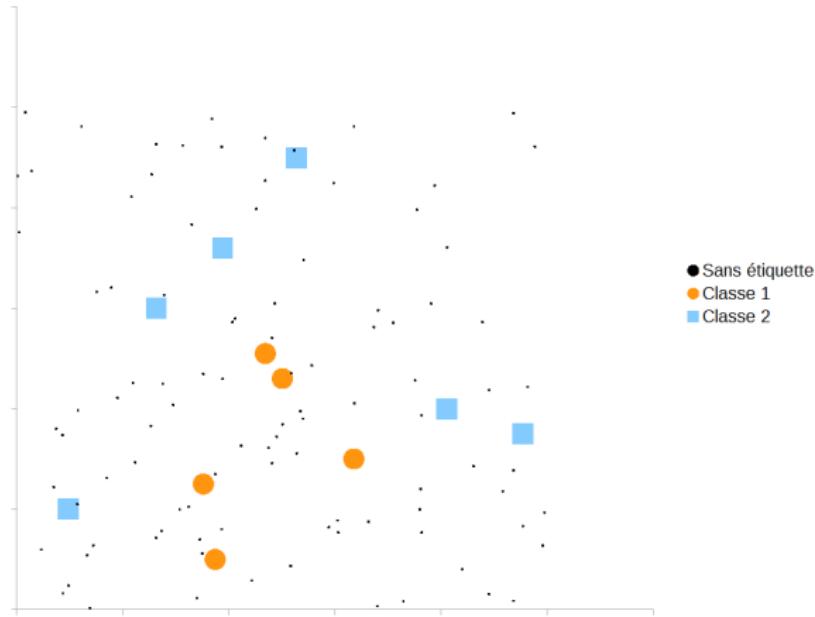
Dispersion pour l'image 1



On détermine la limite séparant les classes

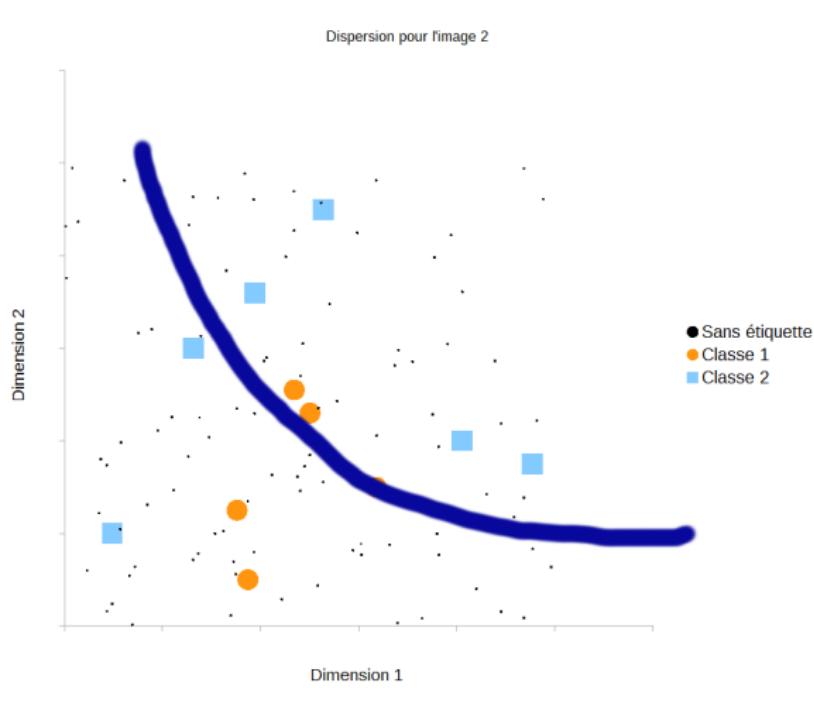
# Illustration du problème

Dispersion pour l'image 2



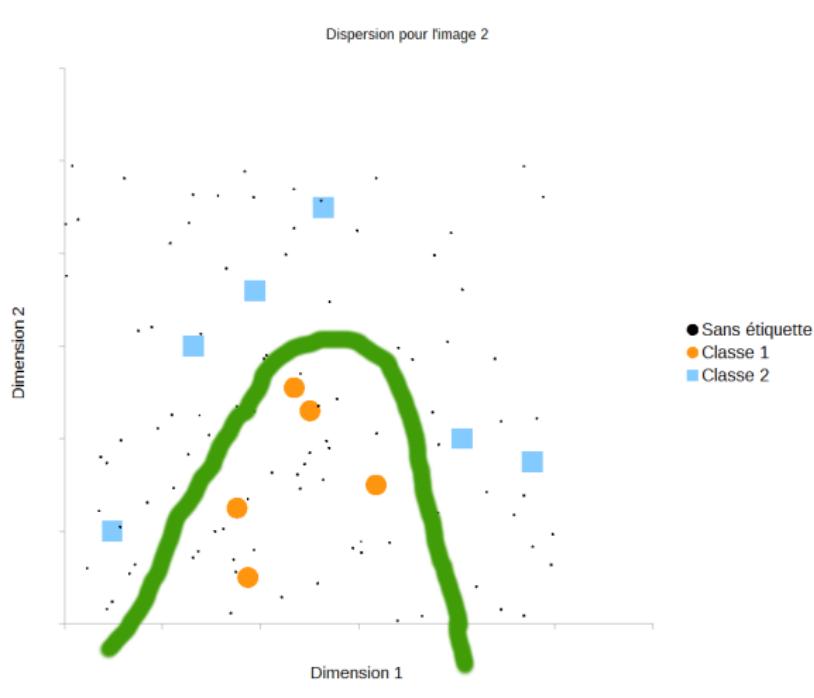
L'autre image, du fait de sa radiométrie, a une distribution différente

# Illustration du problème



Si on lui applique la classification déterminée précédemment, ça ne marche pas  
=> Modèle de l'image 1 : non discriminant pour l'image 2

# Illustration du problème



Il faut donc recalculer un modèle à chaque image

## 1 Présentation de la problématique

- Cadre général
- Illustration du problème

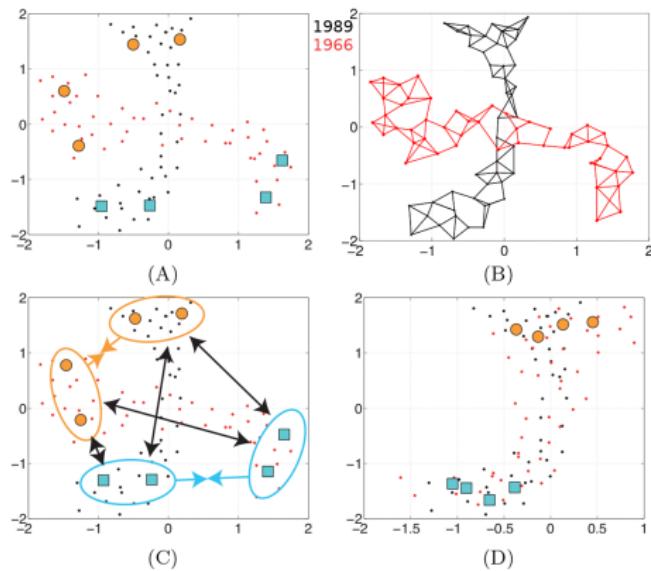
## 2 Méthode employée

- Présentation
- Utilisation

## 3 Résultats obtenus

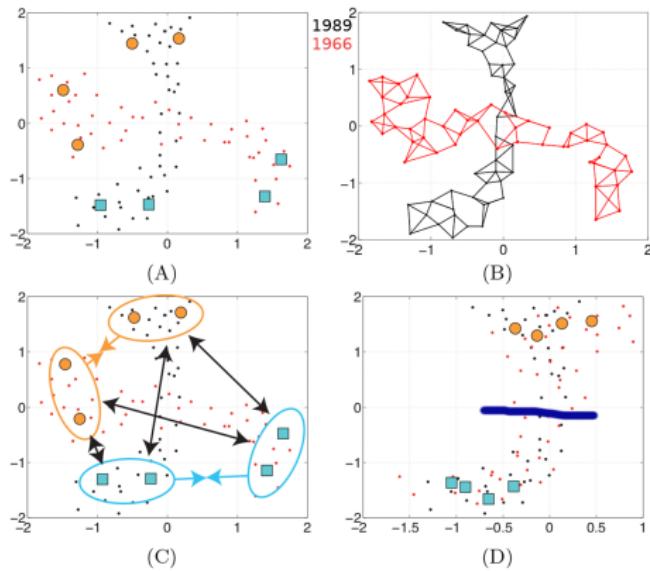
- Estimation de l'efficacité
- Estimation de l'utilité

# La solution : l'adaptation de domaines



Principe de l'alignement de variété. (*Tuia et Camps-Valls, 2016*)

# La solution : l'adaptation de domaines



Principe de l'alignement de variété. (*Tuia et Camps-Valls, 2016*)

# Méthode choisie : l'alignement par noyau de variété (KEMA)

Tuia et Camps-Valls, 2016)

Method	Properties			
	$D \geq 2$	Unpaired	$d_S \neq d_T$	Nonlinear
KEMA	✓	✓	✓	✓

$D$ : number of domains.

$d_S, d_T$ : number of features in source and target.

$\hat{y}$ : semilabels predicted by a classifier.

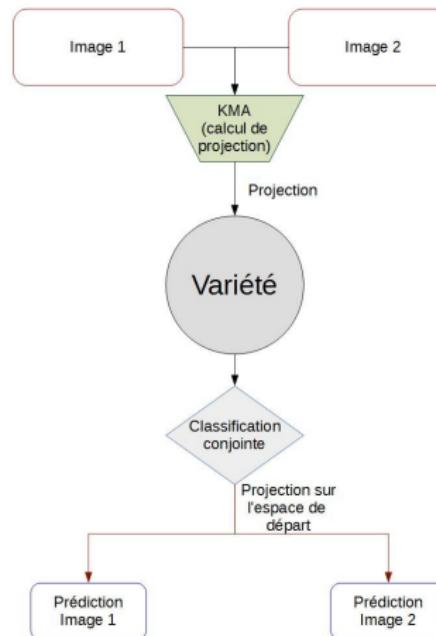
$p$ : known corresponding samples, but no labels.

doi:10.1371/journal.pone.0148655.t001

Caractéristiques de l'alignement de variété. (*Tuia et Camps-Valls, 2016*)

- Prise en compte de  $N$  domaines,  $N > 2$ , correspondant aux diverses images d'archive
- Disparité possible dans le nombre d'attributs entre les domaines (par exemple, présence ou non d'un MNS)
- Paramètres : nombre de pixels par classe (ppc) dans chaque image

# Principe de l'alignement par noyau de variété



Principe de l'alignement de variété.

# Axes de recherche

## Question possibles

- L'alignement par noyau de variété est-il plus efficace qu'une classification SVM classique ?
- Quelle est la taille optimale de l'échantillon servant à l'alignement ?
- Quelle est l'influence d'une différence dans le nombre d'attributs entre les images ?
- La précision de la classification dépend-elle du nombre d'images traitées ?

# Axes de recherche

## Question possibles

- L'alignement par noyau de variété est-il plus efficace qu'une classification SVM classique ?
- Quelle est la taille optimale de l'échantillon servant à l'alignement ?
- Quelle est l'influence d'une différence dans le nombre d'attributs entre les images ?
- La précision de la classification dépend-elle du nombre d'images traitées ?

# Mode opératoire

- **Compréhension et adaptation du programme informatique MATLAB** proposé par Tuia *et al* (2016).
- **Extraction** des données traitées par S. Giordano : 2 images ayant **9 attributs** par pixel (dont 2 issus d'un MNS), avec un apprentissage comportant **5 classes** (bâti, routes, sol nu, cultures, végétation haute, végétation basse).
- **Programmation** de routines permettant de tester et fournir des **statistiques** sur les caractéristiques des classifications réalisées après l'adaptation de domaines (matrices de confusion)

## ① Présentation de la problématique

- Cadre général
- Illustration du problème

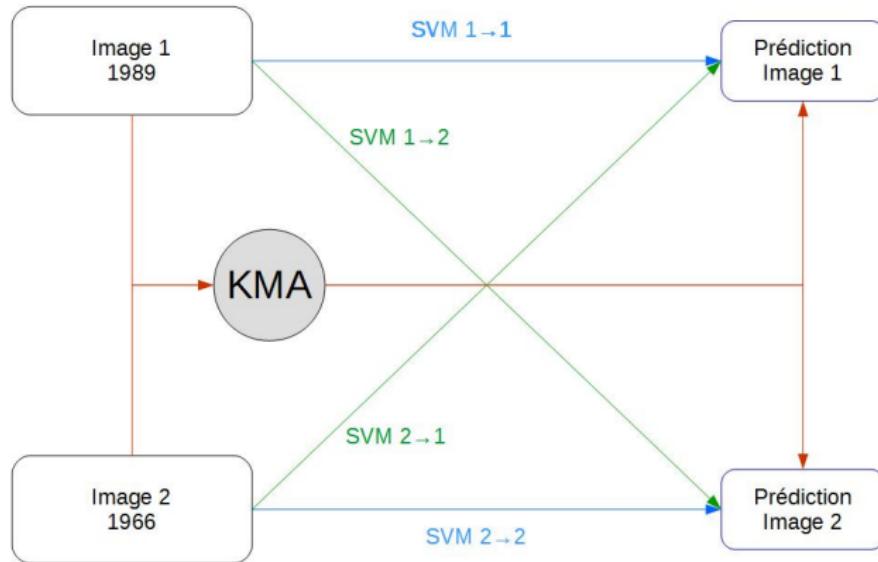
## ② Méthode employée

- Présentation
- Utilisation

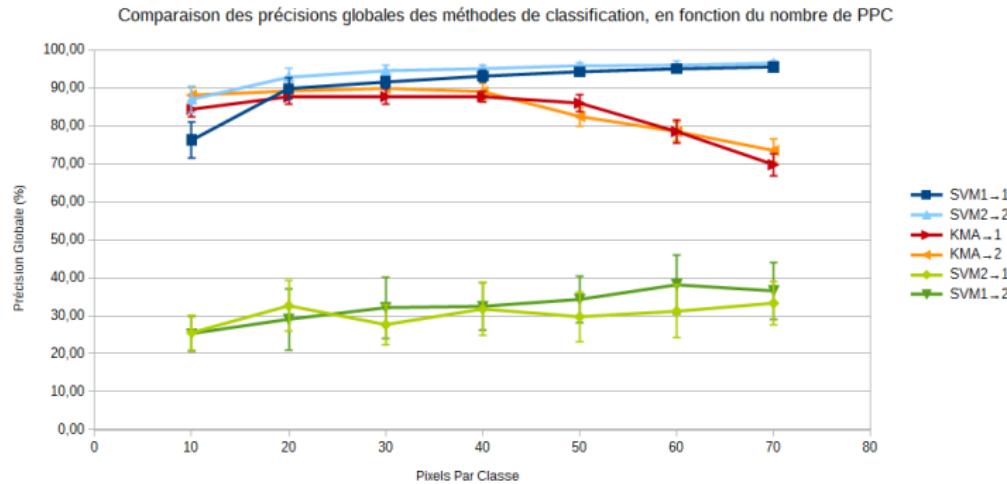
## ③ Résultats obtenus

- Estimation de l'efficacité
- Estimation de l'utilité

# Protocole pour estimer l'efficacité de la méthode



# Efficacité de la méthode d'alignement par noyau de variété



Valeurs obtenues à l'aide du programme MATLAB que nous avons développé

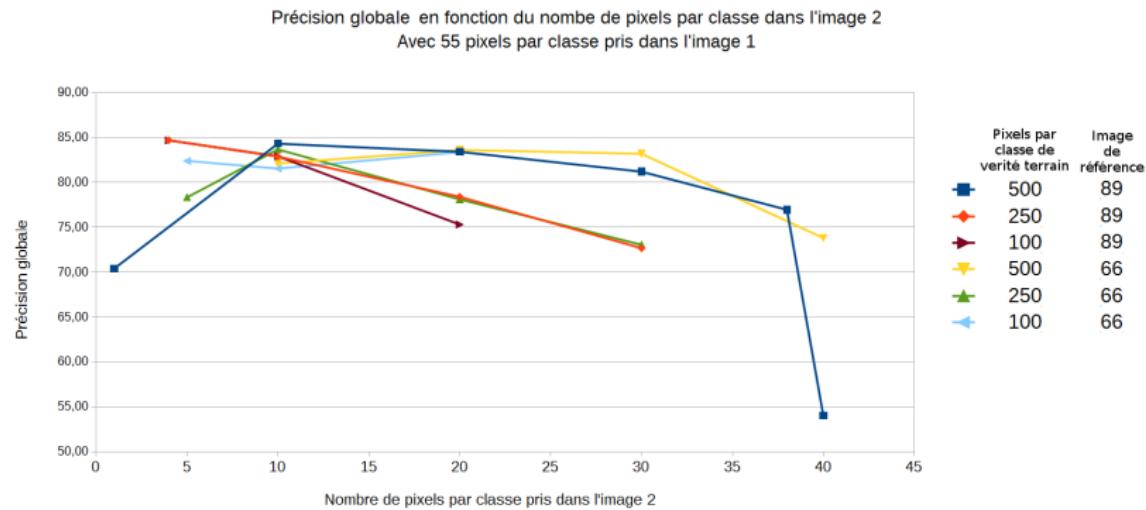
On constate que la méthode d'alignement de variétés (rouge) permet d'obtenir de meilleurs résultats que l'application d'un classificateur de manière croisée (vert).

# Protocole pour estimer l'utilité de la méthode

Toutes choses égales par ailleurs :

- Image de référence : vérité terrain de **1000 pixels par classe** (soit 5000 pixels au total) : grande vérité terrain correspondant à la dernière date.  
Utilisation de **55 pixels par classe pour l'alignement** de variété.
- **Alternance** de la référence entre 1989 et 1966.
- **Variation de la taille de l'apprentissage** dans l'image peu connue (de 100 à 500 ppc)
- Variation du nombre de pixels par classe pour l'alignement de variété.

# Utilité de la méthode d'alignement par noyau de variété



Valeurs obtenues à l'aide du programme MATLAB que nous avons développé

Le nombre de pixels par classe optimal dans l'image qui ne sert pas de référence est de 10, ce qui montre que l'apprentissage peut être petit sur cette image.

# Conclusion et perspectives

## Avantages :

- grande flexibilité dans la classification inter-images
- bonne précision dans la classification d'images

Method	Properties			
	$D \geq 2$	Unpaired	$d_S \neq d_T$	Nonlinear
KEMA	✓	✓	✓	✓

$D$ : number of domains.

$d_S, d_T$ : number of features in source and target.

$\hat{y}$ : semilabels predicted by a classifier.

$p$ : known corresponding samples, but no labels.

doi:10.1371/journal.pone.0148655.t001

Caractéristiques de l'alignement de variété. (*Tuia et Camps-Valls, 2016*)

## Inconvénients :

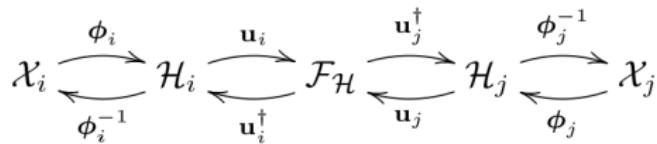
- difficulté d'implémentation informatique
- très grande consommation de mémoire (quadratique)
- trop grand temps de calcul (cubique)

# Références

-  S. Giordano, A. Le Bris, C. Mallet (2017)  
Fully Automatic Analysis of Archival Aerial Images Current status and challenges  
*JURSE 2017*
-  Devis Tuia, Claudio Persello, Lorenzo Bruzzone (2016)  
Recent Advances in Domain Adaptation for the Classification of Remote Sensing Data  
*IEEE Geoscience and Remote Sensing Magazine* 4(2), 41-57
-  Devis Tuia, Gusteau Camps-Valls (2016)  
Kernel Manifold Alignment for Domain Adaptation  
*PLoS ONE* 11(2)
-  Devis Tuia, Diego Marcos, Gustau Camps-Valls (2016)  
Multi-temporal and Multi-source Remote Sensing Image Classification by Nonlinear Relative Normalization  
*ISPRS Journal of Photogrammetry and Remote Sensing* 120, 1-12

# Merci pour votre attention

# Modèle mathématique



Modèle de l'alignement de variété. (*Tuia et Camps-Valls, 2016*)

- $X$  : espace des données
- $H$  : espace de Hilbert
- $F$  : domaine latent (variété)
- $\phi$  : données projetées sur  $H$
- $u$  : vecteur de projection de  $H$  vers  $F$

# Matrices impliquées

$$Y = \begin{pmatrix} y_{1,1} \\ y_{1,2} \\ \dots \\ y_{1,50} \\ 0 \\ \dots \\ 0 \\ y_{2,1} \\ \dots \\ y_{2,50} \\ 0 \\ \dots \\ 0 \end{pmatrix} \quad X_1 = \begin{pmatrix} x_{1,1} & \dots & x_{1,50} & x_{1,51} & \dots & \dots & x_{1,3e6} \\ x_{2,1} \\ x_{3,1} \\ x_{4,1} \\ x_{5,1} \\ x_{6,1} \\ x_{7,1} \\ x_{8,1} \\ x_{9,1} & \dots & x_{9,50} & x_{9,51} & \dots & \dots & x_{9,3e6} \end{pmatrix}$$

$Y$  est dans notre cas de taille  $(3M + 1.5M, 1)$ , soit  $(4.5M, 1)$  et  $X_1$  de taille  $(9, 3M)$

Or on calcule un k-NN graphe avec  $X_1$  puis  $X_2$ , ce qui est donc extrêmement coûteux en mémoire