Regularized Mahalanobis Kernel for the Classification of Hyperspectral Images

M. Fauvel[1], A. Villa[2,3], J. Chanussot[2] and J. A. Benediktsson[3]

[1] DYNAFOR, INRA & ENSAT, INPT, Université de Toulouse - France
[2] GIPSA-Lab, Grenoble Institute of Technology - France
[3] University of Iceland, Reykjavik - Iceland

Atelier Astrostatistique, Grenoble 2011

Outline

High dimensional data

- **High number of measurements but limited number of samples.**

$$\boxed{\mathbf{x}_i \in \mathcal{X}^d \text{ with } d \gg 100,\ i \in \{1, \ldots, n\} \text{ and } n \approx d}$$

- Hyperspectral images : each pixel has thousands of spectral variables
- $\mathcal{X}$ can be sparse
- $\mathcal{X}$ can have different SNR
- Why:
  - A phenomenon depends on a lot of spectral variables
  - We don't know which variables will be useful
  - **Quality and quantity of information !**

Some properties of HD spaces 1/3

- The volume of an hypersphere tend to zero when the dimension grows

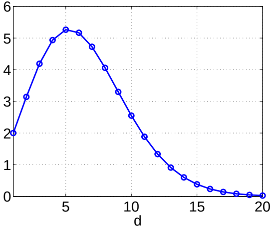
  No closed neighbors

- The volume of an hypersphere concentrates in an outside shell

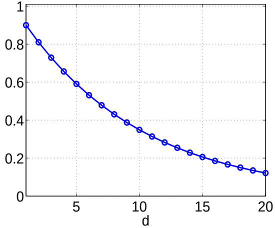
  Normally distributed data concentrates in the tails

- The volume of an hypersphere is negligible compare to the volume of an hypercube

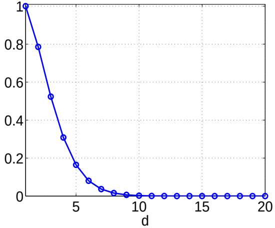
  Uniformly distributed data concentrates in the corners



$V_s(d),\ r = 1$



$(1 - \frac{\varepsilon}{r})^d,\ \varepsilon = 0.1r$



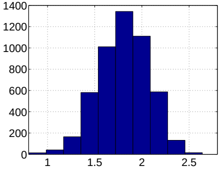
$\frac{V_s(d)}{V_c(d)}$

Some properties of HD spaces 2/3

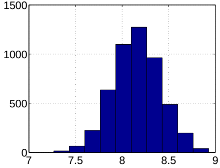
- pdf to have a sample $\|\mathbf{x}\| = t$ $(\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}))$

$$f(t) = \frac{dt^{d-1}\exp(-t^2/2)}{d^{(d/2)}\Gamma(d/2+1)}, \text{ maximum for } t^* = (d-1)^{0.5}$$

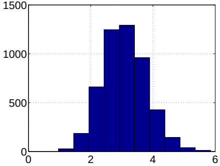
- Some simulations: $n = 5000$, $\|\mathbf{x}\|$.



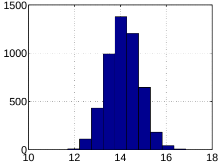
$\mathbf{x} \sim U([-1,1]),\ d = 10$



$\mathbf{x} \sim U([-1,1]),\ d = 200$



$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),\ d = 10$



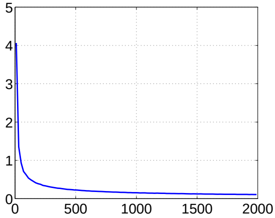
$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),\ d = 200$

Some properties of HD spaces 3/3

- **Concentration of measure phenomenon**: if $\mathbf{x}$ random vector with i.i.d. variables

$$\frac{d_M(\mathbf{x}) - d_m(\mathbf{x})}{d_m(\mathbf{x})} \underset{p}{\longrightarrow} 0$$

for all Minkowski norm: $\|\mathbf{x}\| = \left(\sum_{i=1}^{d} |x_i|^l \right)^{1/l}, \ l = 1, 2 \dots$



- **Empty space phenomenon**: most of the space is empty

A curse but also a blessing!

Implication for classification algorithms 1/2

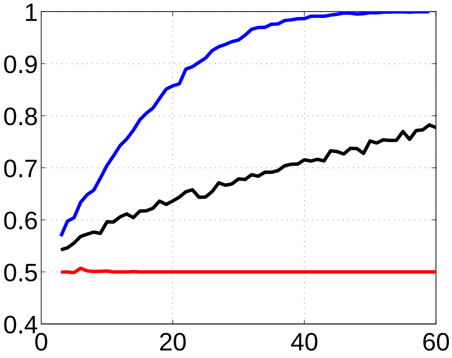
- **Generative methods**
  - Hughes phenomenon: For a fixed training set, there exits an optimal dimension
  - Statistical estimation very difficult: Emptiness + number of parameters
  - Gaussian mixture models
    - Number of parameters $\propto d^2$ by class
    - $\mathbf{\Sigma}^{-1}$ ill-posed
  - Non-parametric models
    - Number of samples to approximate a Gaussian law $\propto 10^{0.6d}$

- **Discriminative methods**
  - Number of points to uniformly sample a unit hypercube: 10^d
  - Methods based on nearest neighbors fail:
    - k-nn
    - Adjacency matrix (e.g. laplacian graph)
    - Local kernel machines
  - More generally, methods based on Euclidean distance fail

Implication for classification algorithms 2/2

- **Emptiness phenomenon**: the classes are more separable!

$$\mathbf{x}_1 \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \text{ and } \mathbf{x}_2 \sim \mathcal{N}(\boldsymbol{\varepsilon}, \mathbf{I})$$



Gaussian mixture, Minimum distance and Linear-SVM

Existing solutions

- **Simple models**:
  - Linear models
  - Gaussian models: $\mathbf{\Sigma}$ diagonal, equal for each class

- **Dimension reduction**: $\mathbf{x} \rightarrow \phi(\mathbf{x})$
  - Statistical approach: PCA, FDA, ICA
  - Local distance: Laplacian eigenmaps, LLE, CCA

- **Kernel methods**: expect local kernels (evaluation of a new sample depends on its neighbors in the training set)

- **Regularization**: Tikhonov $\mathbf{\Sigma}^{-1} \rightarrow (\mathbf{\Sigma} + \lambda \mathbf{I})^{-1}$

- **Subspace models**: Each class is located in a specific subspace: $\mathbf{\Sigma}$ is constrained
  - Probabilistic PCA
  - High Dimensional Discriminant Analysis (HDDA) models

Proposed approach

Subspace models and **kernel methods**

- Use **emptiness** property to construct the kernel
- **How**:
  - Mahalanobis distance for class c:

$$d_{\boldsymbol{\Sigma}_c}(\mathbf{x},\mathbf{z}) = \sqrt{(\mathbf{x}-\mathbf{z})^t\boldsymbol{\Sigma}_c^{-1}(\mathbf{x}-\mathbf{z})}$$

  - Gaussian Radial kernel:

$$k_g(\mathbf{x},\mathbf{z}) = \exp\left(-\frac{d(\mathbf{x},\mathbf{z})^2}{2\sigma^2}\right)$$

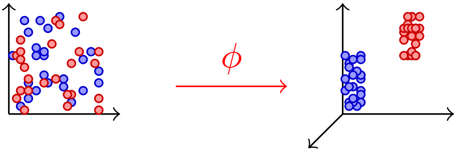
- **Mahalanobis kernel**:

$$k_m(\mathbf{x},\mathbf{z}|c) = \exp\left(-\frac{(\mathbf{x}-\mathbf{z})^t\boldsymbol{\Sigma}_c^{-1}(\mathbf{x}-\mathbf{z})}{2\sigma^2}\right)$$

Kernel methods

- **Kernel function**: It computes the similarity between two samples. It is equivalent to a dot product in some feature space:

$$k(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle_{\mathcal{H}}, \ \phi : \mathbb{R}^d \mapsto \mathcal{H}$$



- **Kernel methods**: The kernel is at the basis of the processing.

$$f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b$$

- **Some kernels**:
  - Linear: $k(\mathbf{x}, \mathbf{z}) = \langle \mathbf{x}, \mathbf{z} \rangle$
  - Polynomial: $k(\mathbf{x}, \mathbf{z}) = \big(\langle \mathbf{x}, \mathbf{z} \rangle + q\big)^p$

The HDDA model 1/3

- Family of **parsimonious models** for HD data [Bouveyron et all, 2007]
- **Cluster assumption**: each class c lives in a specific subspace
- Covariance matrix of class c:

$$\boldsymbol{\Sigma}_c = \mathbf{Q}_c \boldsymbol{\Lambda}_c \mathbf{Q}_c^t = \sum_{i=1}^{d} \lambda_{ci} \mathbf{q}_{ci} \mathbf{q}_{ci}^t$$

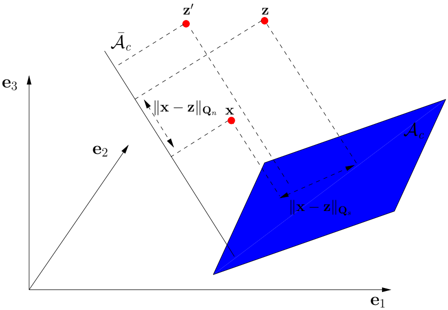
- HDDA: $\text{diag}(\boldsymbol{\Lambda}_c) = \Big[\underbrace{\lambda_{c1} \ldots \lambda_{cp_c}}_{p_c} \underbrace{b_c \ldots\ldots b_c}_{d-p_c}\Big]$ with $p_c \ll d$
- Covariance matrix of class c under HDDA:

$$\boldsymbol{\Sigma}_c = \underbrace{\sum_{i=1}^{p_c} \lambda_{ci} \mathbf{q}_{ci} \mathbf{q}_{ci}^t}_{\mathcal{A}_c} + b_c \underbrace{\sum_{i=p_c+1}^{d} \mathbf{q}_{ci} \mathbf{q}_{ci}^t}_{\bar{\mathcal{A}}_c}$$

- $\mathcal{A}_c$ is the signal subspace and $\bar{\mathcal{A}}_c$ is the noise subspace ($\mathbb{R}^d = \mathcal{A}_c \bigoplus \bar{\mathcal{A}}_c$)

The HDDA model 2/3

- In $\mathbb{R}^3$:



- The inverse can be computed explicitly:

$$\mathbf{\Sigma}_c^{-1} = \sum_{i=1}^{p_c} \frac{1}{\lambda_{ci}} \mathbf{q}_{ci}\mathbf{q}_{ci}^t + \frac{1}{b_c} \sum_{i=p_c+1}^{d} \mathbf{q}_{ci}\mathbf{q}_{ci}^t$$

- Using $\mathbf{I} = \sum_{i=1}^{d} \mathbf{q}_{ci}\mathbf{q}_{ci}^t$,

$$\mathbf{\Sigma}_c^{-1} = \sum_{i=1}^{p_c} \left(\frac{1}{\lambda_{ci}} - \frac{1}{b_c} \right) \mathbf{q}_{ci}\mathbf{q}_{ci}^t + \frac{1}{b_c}\mathbf{I}$$

The HDDA model 3/3

- **So what?**
  - Less parameters have to be estimated ($d = 100$ and $p_c = 10$)
    - Full $\mathbf{\Sigma}$: $d(d+3)/2$ parameters $\rightarrow$ 5150
    - HDDA: $d(p_c+1)+2-p_c(p_c-1)/2$ parameters $\rightarrow$ 1057
  - Better than PCA
    - $\mathbf{x}$ and $\mathbf{z}$ may be artificially closed in $\mathcal{A}_c$
    - An accurate estimation of p_c is necessary

- **Estimation**: From the sample covariance matrix

$$\hat{\mathbf{\Sigma}}_c = \frac{1}{n_c}\sum_{i=1}^{n_c}\left(\mathbf{x}_i - \bar{\mathbf{x}}_c\right)\left(\mathbf{x}_i - \bar{\mathbf{x}}_c\right)^t,\ \mathbf{x}_i \in c$$

  - $\left\{\hat{\lambda}_{ci}\right\}_{i=1}^{p_c}$ are estimated by the first p_c eigenvalues of $\hat{\mathbf{\Sigma}}_c$
  - $\left\{\hat{\mathbf{q}}_{ci}\right\}_{i=1}^{p_c}$ are estimated by the first p_c eigenvectors of $\hat{\mathbf{\Sigma}}_c$
  - $\hat{b}_c$ is estimated by $\left(\text{trace}(\hat{\mathbf{\Sigma}}_c) - \sum_{i=1}^{\hat{p}_c}\hat{\lambda}_{ci}\right)/(d-\hat{p}_c)$
  - $\hat{p}_c$ is estimated with the scree test of Catell

Mahalanobis kernel 1/2

- $\left\{\hat{\lambda}_{ci}\right\}_{i=1}^{p_c}$ and $\hat{b}_c$ are switched to kernel hyperparameters $\left\{\sigma_i\right\}_{i=1}^{p_c+1}$
- **The kernel**:

$$k_m(\mathbf{x},\mathbf{z}|c) = \exp\left(-\frac{1}{2}\left(\sum_{i=1}^{\hat{p}_c}\frac{(\mathbf{x}-\mathbf{z})^t\hat{\mathbf{q}}_{ci}\hat{\mathbf{q}}_{ci}^t(\mathbf{x}-\mathbf{z})}{\sigma_i^2} + \frac{\|\mathbf{x}-\mathbf{z}\|^2}{\sigma_{\hat{p}_c+1}^2}\right)\right)$$

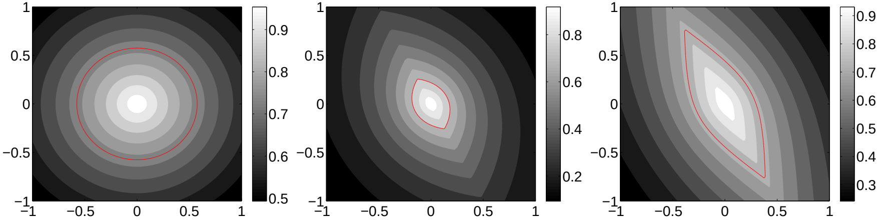
- Another formulation: **product of Gaussian kernels**

$$k_m(\mathbf{x},\mathbf{z}|c) = k_g(\mathbf{x},\mathbf{z}) \times \prod_{i=1}^{\hat{p}_c} k_g(\hat{\mathbf{q}}_{ci}^t\mathbf{x}, \hat{\mathbf{q}}_{ci}^t\mathbf{z})$$

- The Mahalanobis kernel constructs with the HDDA model is a mixture of a Gaussian kernel on the original data and a Gaussian kernel on the p_c first principal components of the considered class

Mahalanobis kernel 2/2

$k_m(\mathbf{0}, \mathbf{x}|c)$ with $\mathbf{0} = [0, 0]$ and $\mathbf{x} \in [-1, 1]^2$



(a) (b) (c)

- $\mathbf{\Sigma}_c = [0.6 \ \ -0.2; -0.2 \ 0.6]$ and $p_c = 1$
- Red contour line $\rightarrow k_m = 0.75$
- (a): Gaussian kernel
- (b): Mahalanobis kernel with $\sigma_1^2 = \sigma_2^2 = 0.5$
- (c): Mahalanobis kernel with $\sigma_1^2 = 1.5$ and $\sigma_2^2 = 0.5$

L2-Support Vectors Machines 1/2

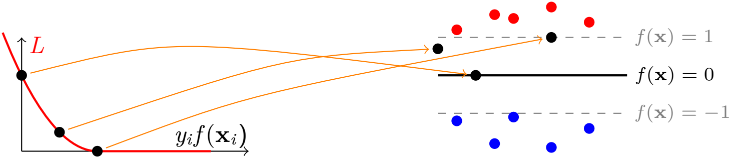
- Supervised method: $\mathcal{S} = \big\{(\mathbf{x}_i, y_i)\big\}_{i=1}^{n}$, $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$

$$h(\mathbf{z}) = \text{sign}\big(f(\mathbf{z})\big) \text{ with } f(\mathbf{z}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{z}, \mathbf{x}_i) + b$$

- Hyperparameters $\big(\{\alpha_i\}_{i=1}^{n}, b\big)$ learn by solving:

$$\min_{\boldsymbol{\alpha}, b} \left[\frac{1}{C}\|f\|^2 + \sum_{i=1}^{n} L\Big(y_i, f(\mathbf{x}_i)\Big)^2\right]$$

  - $\|f\|^2 = \sum_{i,j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$
  - $L\Big(y_i, f(\mathbf{x}_i)\Big)^2 = \max\Big(0, 1 - y_i f(\mathbf{x}_i)\Big)^2$



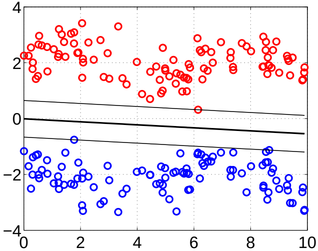
L2-Support Vectors Machines 2/2

- Equivalently: with $\tilde{k}(\mathbf{x}_i, \mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j) + C^{-1}\delta_{ij}$

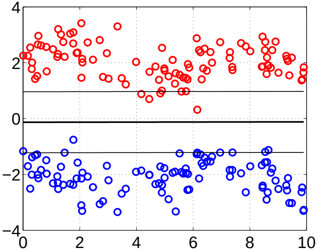
$$\max_{\alpha} \; g(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{n} \alpha_i\alpha_j y_i y_j \tilde{k}(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{subject to} \quad 0 \leq \alpha_i \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Toy examples:



$C = 100$



$C = 0.01$

Radius-margin bound 1/2

- In our setting $\mathbf{p} = [\sigma_1^2, \ldots, \sigma_{\hat{p}+1}^2, C]$
- Estimate of the generalization error: Radius-margin bound (upper bound of LOO)

$$\mathcal{T}(\mathbf{p}) := \mathcal{R}^2 \tilde{g}$$

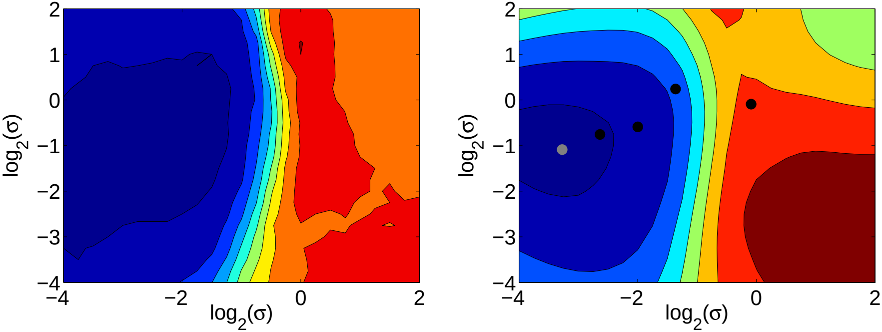
- $\tilde{g}$ depends on $(\tilde{\alpha}, \mathbf{p})$ and $\tilde{\alpha}$ depends on $\mathbf{p}$. But, **since $\tilde{g}$ depends on α via an optimization problem, the gradient of α w.r.t. $\mathbf{p}$ does not enter into the computation of $\tilde{g}$**.

$$\begin{aligned}
\tilde{g}(\mathbf{p}) &= \max_{\alpha} g(\mathbf{p}, \alpha) &= g\big(\mathbf{p}, \tilde{\alpha}(\mathbf{p})\big) \\
\boldsymbol{\nabla} \tilde{g} &= \left(\frac{\partial g}{\partial \mathbf{p}}, \frac{\partial g}{\partial \tilde{\alpha}}\right) & \\
&= \left(\frac{\partial g}{\partial \mathbf{p}}, \frac{\partial g}{\partial \alpha}\Big|_{\alpha=\tilde{\alpha}} \frac{\partial \alpha}{\partial \mathbf{p}}\right) &= \left(\frac{\partial g}{\partial \mathbf{p}}, \mathbf{0}\right)
\end{aligned}$$

- Gradient descent on the radius margin bound: $\boldsymbol{\nabla}\mathcal{T} = \frac{\partial \mathcal{R}^2}{\partial \mathbf{p}} g + \mathcal{R}^2 \frac{\partial g}{\partial \mathbf{p}}$
- Training: $\min \max$ problem (**non-convex**)

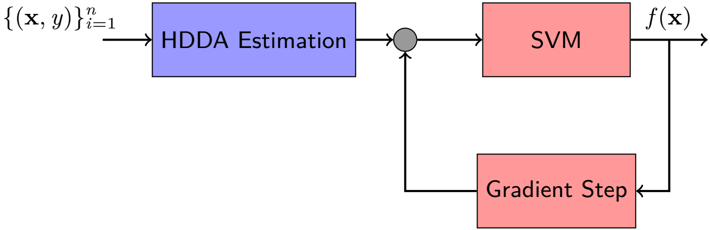
Radius-margin bound 2/2

- Toy example: $\{\mathbf{x} \big| \mathsf{var}(x_1) \ll \mathsf{var}(x_2)\}$



Test errors   Radius-margin bound

Block diagram



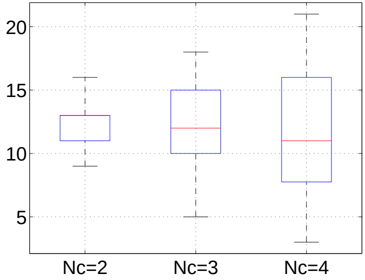
- Multiclass: one classifier per class (but $\text{SVM}_{c_i \text{ vs } c_j} \neq \text{SVM}_{c_j \text{ vs } c_i}$)
- Complexity:
  - HDDA: $\frac{2d^3}{3}$ or $p^2 d$, computation of the eigenvalues/eigenvectors
  - SVM: $\approx dn^3$, CQP solver
  - Gradient step: $\approx (p+1)n^2$

Simulated data 1/3

- Experimental setup: Mixture of Gaussian following HDDA model

$$\mathbf{x} = \sum_{i=1}^{c} \alpha_i \mathbf{s}_i + \mathbf{b},\ \ y = j \text{ such as } \alpha_j = \max_i \alpha_i \text{ and } \mathbf{s}_i \sim \text{HDDA}$$

  - $d = 413,\ p = 10,\ n = 1000,\ n_t = 1500$ and $SNR = 1$
  - Mean values were extracted from spectral library
  - Number of classes $N_c = 2$, 3 and 4
  - 50 tries

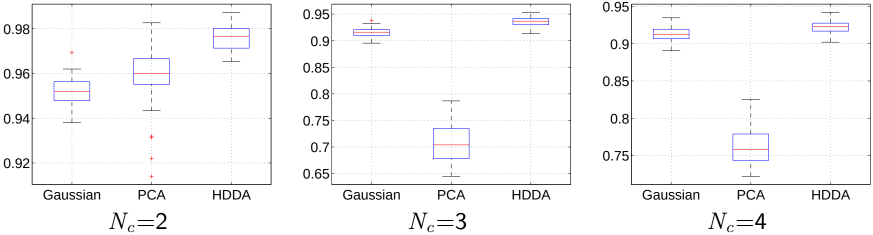


Simulated data 1/3

- Experimental setup: Mixture of Gaussian following HDDA model

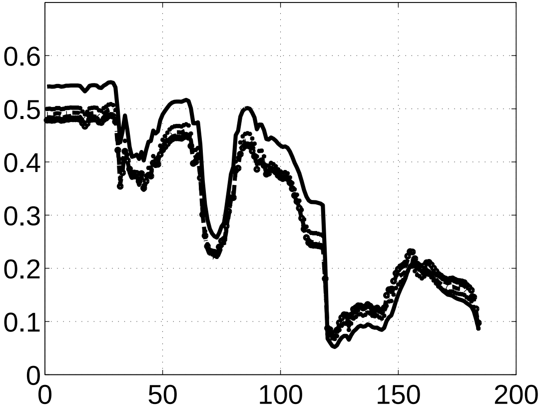
$$\mathbf{x} = \sum_{i=1}^{c} \alpha_i \mathbf{s}_i + \mathbf{b},\ \ y = j \text{ such as } \alpha_j = \max_i \alpha_i \text{ and } \mathbf{s}_i \sim \text{HDDA}$$

  - $d = 413,\ p = 10,\ n = 1000,\ n_t = 1500$ and $SNR = 1$
  - Mean values were extracted from spectral library
  - Number of classes N_c = 2, 3 and 4
  - 50 tries



Simulated data 2/3

- The model has 5 parameters (Sylvain Douté): the grain size of water and CO_2 ice, the proportion of water, CO_2 ice and dust.
- $\mathbf{x} \in \mathbb{R}^{184}$ and $n = 31500$.
- Fives classes according to the grain size of water, $n = n_t = 15750$



Simulated data 3/3

- Estimated subspace size: $s = 10^{-5}$

c	1	2	3	4	5
$\hat{p}$	15	14	12	13	14

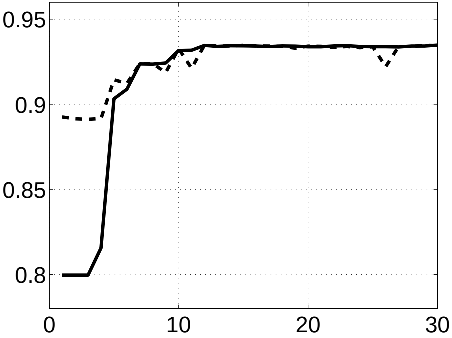
- Classification accuracies:

Kernel	**Gaussian**	**PCA-Mahalanobis**	**HDDA-Mahalanobis**
$y = 50$	99.7	99.7	99.8
$y = 150$	97.6	98.2	98.3
$y = 250$	94.7	96.0	96.1
$y = 350$	89.4	93.4	93.4
$y = 450$	95.0	95.3	95.4
OA	78.3	91.1	91.3
K	85.4	88.9	89.1

- McNemar(HDDA/PCA) $\rightarrow$ 2.58

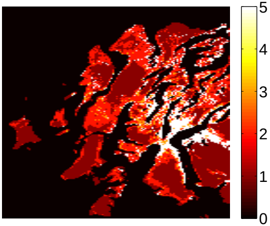
Influence of the parameter $\hat{p}_c$

- **OA vs $\hat{p}_c$ (class *y=350*)**:

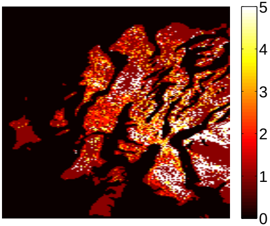


Real data

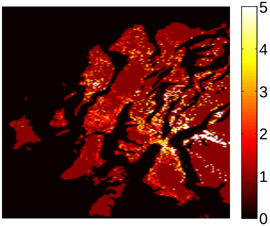
- Data from the imaging spectrometer OMEGA (visible and infra red, 0.95-4.15, 184 wavelengths). Atmospherically corrected (S. Douté).
- Parameters learn with the simulated data.
- Colormap:
  - 0: no data
  - 1: $y = 50$
  - 2: $y = 150$
  - 3: $y = 250$
  - 4: $y = 350$
  - 5: $y = 450$



Gaussian



PCA



HDDA

Conclusion

- Classification of hyperspectral images
- A Mahalanobis kernel based on HDDA was proposed:
  - Cluster assumption
  - Multiple hyperparameters
- Link with mixture kernels
- SVM Classification framework
- Good classification results on three data sets
  - Better than the conventional RBF
  - As good as PCA + RBF

Perspectives 1/2

- Implementation: Optimization of the hyperparameters
- Estimation of $\hat{p}_c$
- Construction of others kernel:

$$k(\mathbf{x}, \mathbf{z}) = \left(\mathbf{x}^t \mathbf{\Sigma}^{-1} \mathbf{z} + 1\right)^p$$

- Investigate mixture of kernels :

$$k_m(\mathbf{x}, \mathbf{z}|c) = \mu_o k_g(\mathbf{x}, \mathbf{z}) + \sum_{i=1}^{\hat{p}_c} \mu_i k_g(\hat{\mathbf{q}}_{ci}^t \mathbf{x}, \hat{\mathbf{q}}_{ci}^t \mathbf{z})$$

- Discriminative subspaces (Fisher ...)

Perspectives 2/2

- Supervised - VS - Unsupervised
- Model transfert : From simulated data to real data
- Semi-supervised methods
- Face the strong non-linearity of the physical model (saturation of the parameters).

Regularized Mahalanobis Kernel for the Classification of Hyperspectral Images

M. Fauvel[1], A. Villa[2,3], J. Chanussot[2] and J. A. Benediktsson[3]

[1] DYNAFOR, INRA & ENSAT, INPT, Université de Toulouse - France
[2] GIPSA-Lab, Grenoble Institute of Technology - France
[3] University of Iceland, Reykjavik - Iceland

Atelier Astrostatistique, Grenoble 2011