Operational Feature Selection in Gaussian Mixture Models

Final internship ENSTA - M2 AIC

A. Lagrange

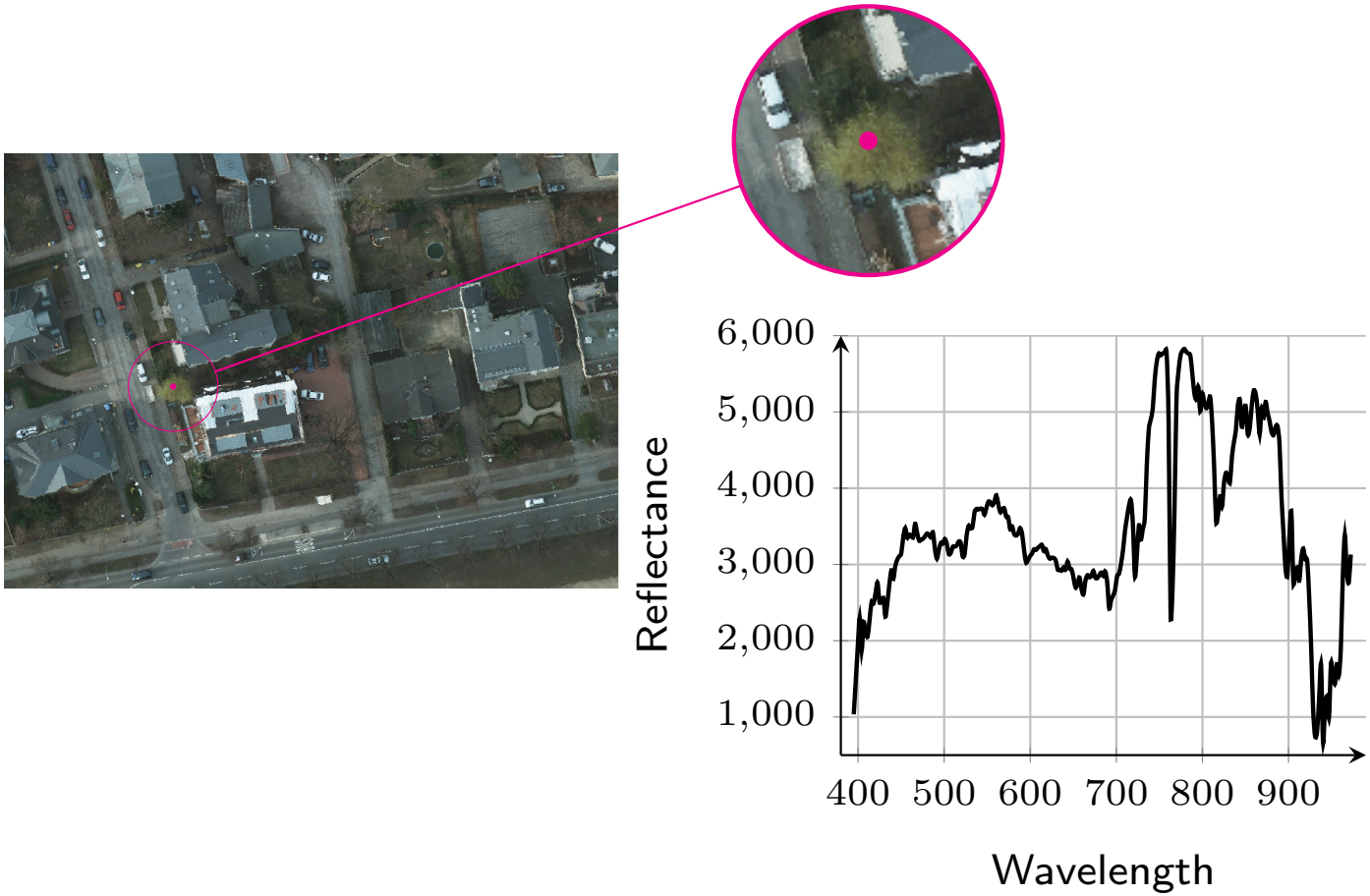
Supervisor: Mathieu Fauvel (DYNAFOR - INRA/ENSAT), Manuel Grizonnet (CNES)

September 13, 2016

Outline

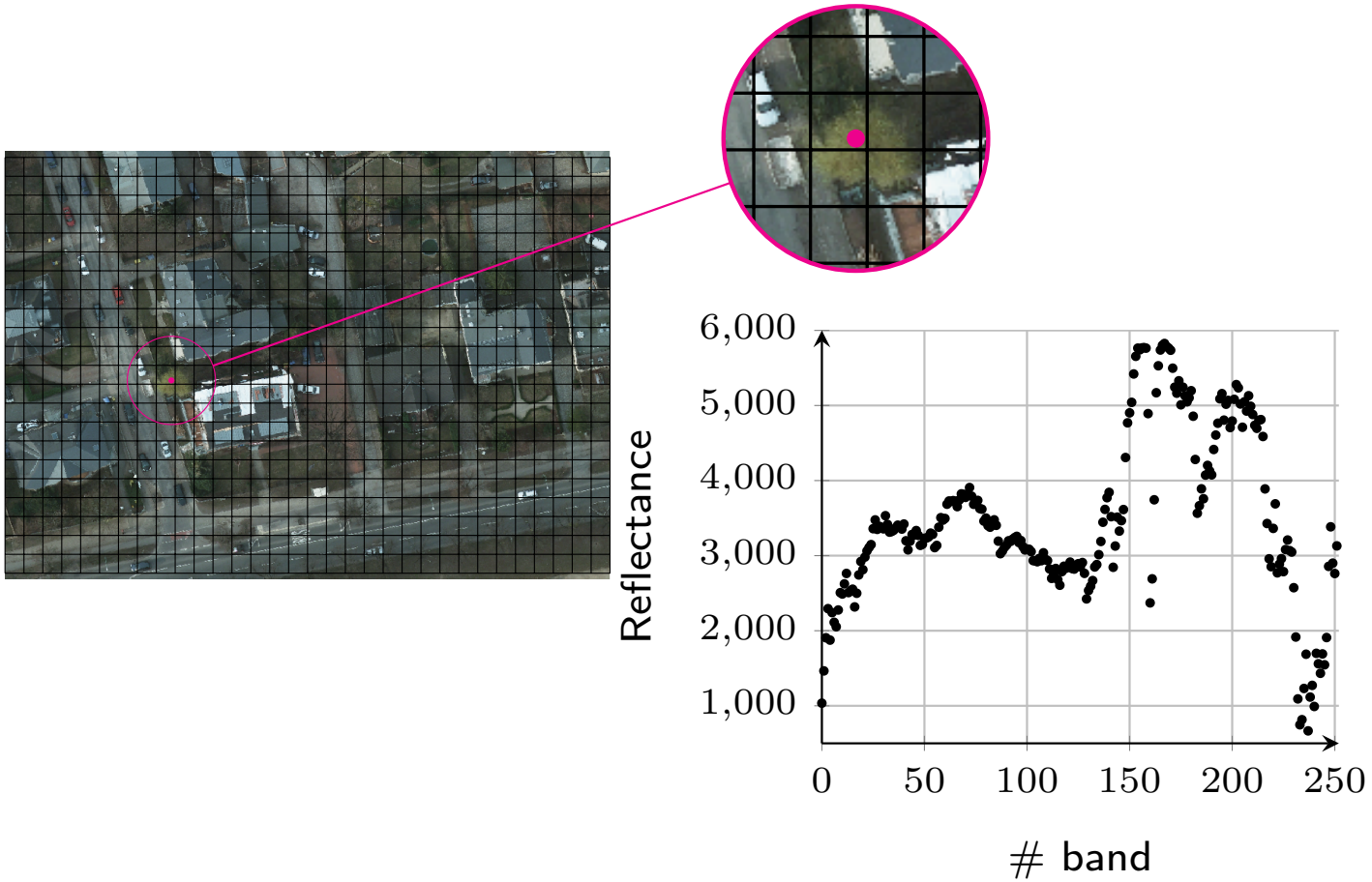
Remote sensing images

A remote sensing hyperspectral image is a sampling of a spatial, spectral and temporal signal.



Remote sensing images

A remote sensing hyperspectral image is a sampling of a spatial, spectral and temporal signal.



Pixelwise image classification in high dimensional space

- Statistical issue (curse of dimensionality): **high number of parameters** to estimate. For Gaussian Mixture Model, $\sim d^2$ parameters by class.

  Example: for $d = 250$, 31625 *parameters*

- Computational issue: **huge quantity of data**. Several terabytes of images produced every day.

Pixelwise image classification in high dimensional space

- Statistical issue (curse of dimensionality): **high number of parameters** to estimate. For Gaussian Mixture Model, $\sim d^2$ parameters by class.

  Example: for $d = 250$, 31625 *parameters*

- Computational issue: **huge quantity of data**. Several terabytes of images produced every day.

Considered approach

- Feature selection to reduce dimension.

Gaussian mixture models

- A sample is the realization of a random vector which distribution is a mixture of C classes conditioned Gaussian distribution:
$$p(\mathbf{x}) = \sum_{c=1}^{C} \pi_c \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{\Sigma}_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^t \mathbf{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)\right).$$
- Parameters computed with the conventional unbiased empirical estimator:
$$\hat{\pi}_c = \frac{n_c}{n}, \quad \hat{\boldsymbol{\mu}}_c = \frac{1}{n_c}\sum_{\{i|y_i=c\}} \mathbf{x}_i,$$
$$\hat{\mathbf{\Sigma}}_c = \frac{1}{(n_c-1)}\sum_{\{i|y_i=c\}} (\mathbf{x}_i-\boldsymbol{\mu}_c)(\mathbf{x}_i-\boldsymbol{\mu}_c)^t.$$
- The decision rule uses the Maximum A Posteriori:
$$\mathbf{x} \text{ belongs to } c \Leftrightarrow c = \arg\max_{c \in C} p(y=c|\mathbf{x}).$$
- Quadratic decision rule is rewritten:
$$Q_c(\mathbf{x}) = -(\mathbf{x}-\boldsymbol{\mu}_c)^t \mathbf{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c) - \log(|\mathbf{\Sigma}_c|) + 2\log(\pi_c) - d\log(2\pi).$$

Gaussian mixture models

- A sample is the realization of a random vector which distribution is a mixture of C classes conditioned Gaussian distribution:
$$p(\mathbf{x}) = \sum_{c=1}^{C} \pi_c \frac{1}{(2\pi)^{\frac{d}{2}} |\mathbf{\Sigma}_c|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^t \mathbf{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)\right).$$
- Parameters computed with the conventional unbiased empirical estimator:
$$\hat{\pi}_c = \frac{n_c}{n}, \quad \hat{\boldsymbol{\mu}}_c = \frac{1}{n_c}\sum_{\{i|y_i=c\}} \mathbf{x}_i,$$
$$\hat{\mathbf{\Sigma}}_c = \frac{1}{(n_c-1)}\sum_{\{i|y_i=c\}} (\mathbf{x}_i-\boldsymbol{\mu}_c)(\mathbf{x}_i-\boldsymbol{\mu}_c)^t.$$
- The decision rule uses the Maximum A Posteriori:
$$\mathbf{x} \text{ belongs to } c \Leftrightarrow c = \arg\max_{c \in C} p(y=c|\mathbf{x}).$$
- Quadratic decision rule is rewritten:
$$Q_c(\mathbf{x}) = -(\mathbf{x}-\boldsymbol{\mu}_c)^t \mathbf{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c) - \log(|\mathbf{\Sigma}_c|) + 2\log(\pi_c) - d\log(2\pi).$$

Problem: $\mathbf{\Sigma}_c$ is often badly-conditioned.

Assess model quality: criterion function

Divergence measure

- Kullback–Leibler divergence:

$$\mathrm{SKL}_{ij} = \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_{c_i}^{-1}\boldsymbol{\Sigma}_{c_j} + \boldsymbol{\Sigma}_{c_j}^{-1}\boldsymbol{\Sigma}_{c_i}) - d + \frac{1}{2}(\boldsymbol{\mu}_{c_i} - \boldsymbol{\mu}_{c_j})^t(\boldsymbol{\Sigma}_{c_i}^{-1} + \boldsymbol{\Sigma}_{c_j}^{-1})(\boldsymbol{\mu}_{c_i} - \boldsymbol{\mu}_{c_j})$$

- Bhattacharyya distance:

$$\mathrm{B}_{ij} = \frac{1}{8}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^t \left(\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2}\ln\left(\frac{|\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j|}{\sqrt{|\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j|}}\right)$$

- Jeffries–Matusita distance:

$$\mathrm{JM}_{ij} = \sqrt{2\{1 - \exp[-B_{ij}]\}}$$

Classification rate

- Overall accuracy:

$$\mathrm{OA} = \frac{\mathrm{TP}}{n}$$

- Cohen's kappa:

$$\mathrm{Kappa} = \frac{OA - \sum_{c=1}^{C} \frac{TP_c}{FP_c}\frac{TP_c}{FN_c}}{1 - \sum_{c=1}^{C} \frac{TP_c}{FP_c}\frac{TP_c}{FN_c}}$$

- Mean of F1-score:

$$\text{F1 Mean} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

where n is the number of samples, TP stands for True Positive, FN False Negative and FP False Positive.

Assess model quality: criterion function

Divergence measure

- Kullback–Leibler divergence:

$$\mathrm{SKL}_{ij} = \frac{1}{2}\mathrm{Tr}(\boldsymbol{\Sigma}_{c_i}^{-1}\boldsymbol{\Sigma}_{c_j} + \boldsymbol{\Sigma}_{c_j}^{-1}\boldsymbol{\Sigma}_{c_i}) - d + \frac{1}{2}(\boldsymbol{\mu}_{c_i} - \boldsymbol{\mu}_{c_j})^t(\boldsymbol{\Sigma}_{c_i}^{-1} + \boldsymbol{\Sigma}_{c_j}^{-1})(\boldsymbol{\mu}_{c_i} - \boldsymbol{\mu}_{c_j})$$

- Bhattacharyya distance:

$$\mathrm{B}_{ij} = \frac{1}{8}(\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)^t \left(\frac{\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j}{2}\right)^{-1} (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j) + \frac{1}{2}\ln\left(\frac{|\boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_j|}{\sqrt{|\boldsymbol{\Sigma}_i||\boldsymbol{\Sigma}_j|}}\right)$$

- Jeffries–Matusita distance:

$$\mathrm{JM}_{ij} = \sqrt{2\{1 - \exp[-B_{ij}]\}}$$

Classification rate

- Overall accuracy:

$$\mathrm{OA} = \frac{\mathrm{TP}}{n}$$

- Cohen's kappa:

$$\mathrm{Kappa} = \frac{OA - \sum_{c=1}^{C} \frac{TP_c}{FP_c}\frac{TP_c}{FN_c}}{1 - \sum_{c=1}^{C} \frac{TP_c}{FP_c}\frac{TP_c}{FN_c}}$$

- Mean of F1-score:

$$\text{F1 Mean} = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FN} + \mathrm{FP}}$$

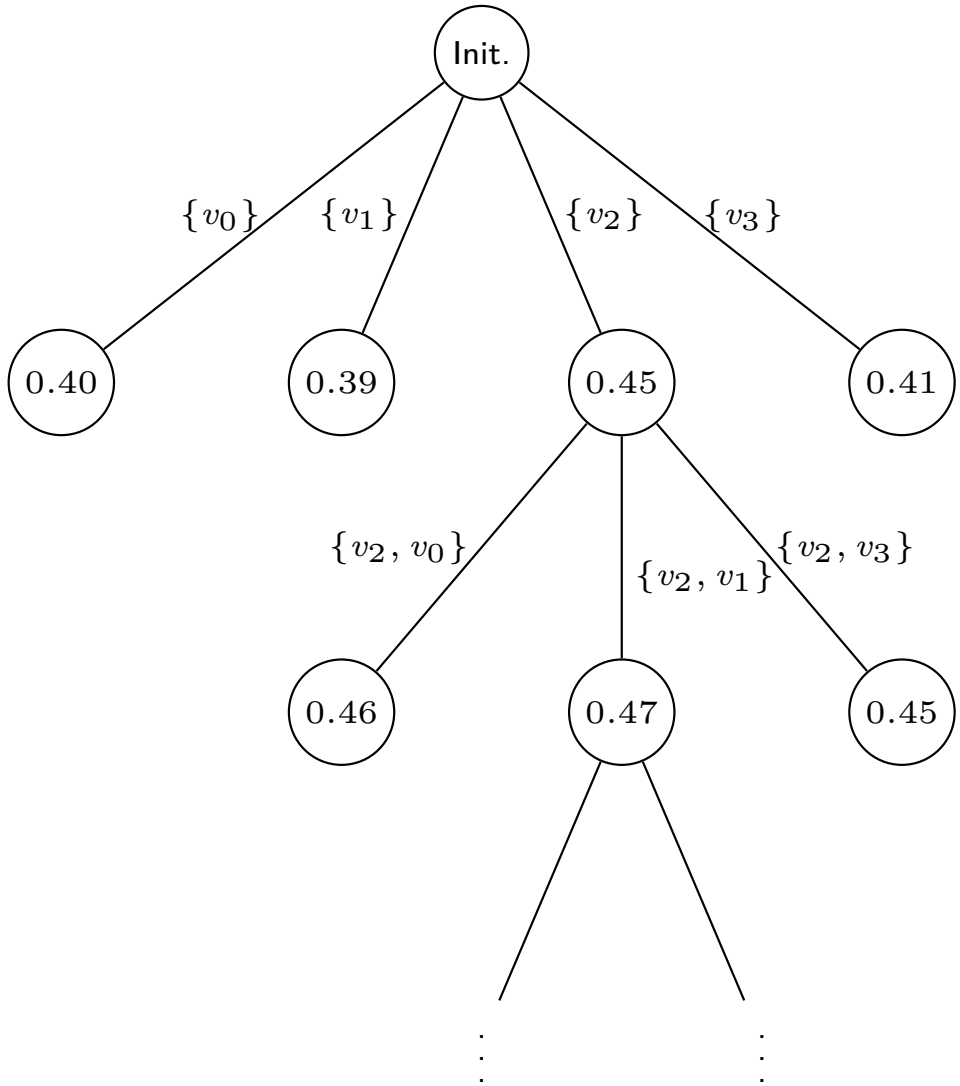
where n is the number of samples, TP stands for True Positive, FN False Negative and FP False Positive.

Strongly dependant of Gaussian hypothesis

Need classification for estimation

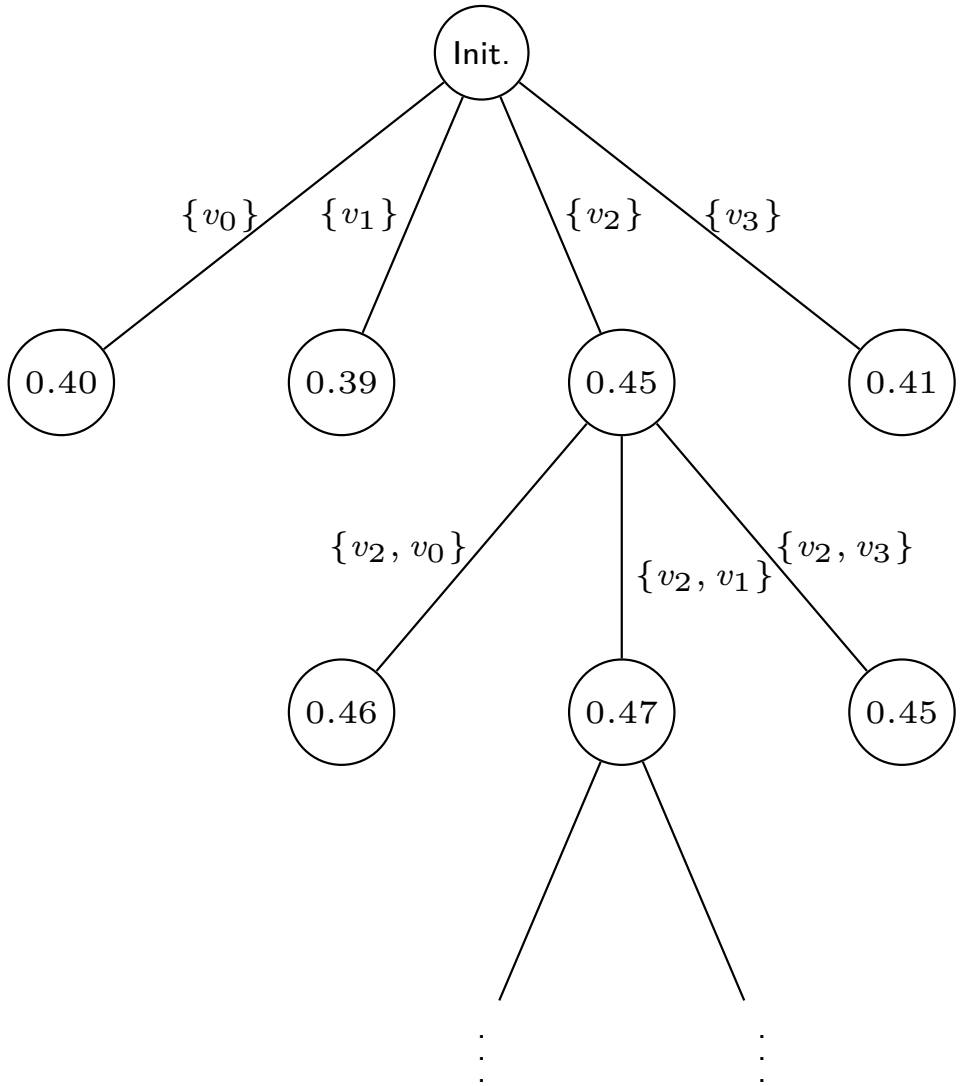
Sequential Forward Features Selection

- A forward iterative selection method
- Two key elements:
  - A criterion function to rank subsets of features
  - A search strategy picking features one by one



Sequential Forward Features Selection

- A forward iterative selection method
- Two key elements:
  - A criterion function to rank subsets of features
  - A search strategy picking features one by one
- Floating forward variation: **addition of backward steps** after each forward step



Update rules for criterion computation

Update rules for criterion computation

Inverse update for augmented feature set

$$(\mathbf{\Sigma}^{(k)})^{-1} = \begin{bmatrix} (\mathbf{\Sigma}^{(k-1)})^{-1} + \frac{1}{\alpha}(\mathbf{\Sigma}^{(k-1)})^{-1}\mathbf{u}\mathbf{u}^t(\mathbf{\Sigma}^{(k-1)})^{-1} & -\frac{1}{\alpha}(\mathbf{\Sigma}^{(k-1)})^{-1}\mathbf{u} \\ -\frac{1}{\alpha}\mathbf{u}^t(\mathbf{\Sigma}^{(k-1)})^{-1} & \frac{1}{\alpha} \end{bmatrix}$$

where $\mathbf{\Sigma}^{(k)} = \begin{bmatrix} \mathbf{\Sigma}^{(k-1)} & \mathbf{u} \\ \mathbf{u}^t & \sigma_{kk} \end{bmatrix}$ and $\alpha = \sigma_{kk} - \mathbf{u}^t(\mathbf{\Sigma}^{(k-1)})^{-1}\mathbf{u}$.

Update rules for criterion computation

Inverse update for augmented feature set

$$(\mathbf{\Sigma}^{(k)})^{-1} = \begin{bmatrix} (\mathbf{\Sigma}^{(k-1)})^{-1} + \frac{1}{\alpha}(\mathbf{\Sigma}^{(k-1)})^{-1}\mathbf{u}\mathbf{u}^t(\mathbf{\Sigma}^{(k-1)})^{-1} & -\frac{1}{\alpha}(\mathbf{\Sigma}^{(k-1)})^{-1}\mathbf{u} \\ -\frac{1}{\alpha}\mathbf{u}^t(\mathbf{\Sigma}^{(k-1)})^{-1} & \frac{1}{\alpha} \end{bmatrix}$$

where $\mathbf{\Sigma}^{(k)} = \begin{bmatrix} \mathbf{\Sigma}^{(k-1)} & \mathbf{u} \\ \mathbf{u}^t & \sigma_{kk} \end{bmatrix}$ and $\alpha = \sigma_{kk} - \mathbf{u}^t(\mathbf{\Sigma}^{(k-1)})^{-1}\mathbf{u}$.

Quadratical term update for augmented feature set

$$(\mathbf{x}^{(k)})^t(\mathbf{\Sigma}^{(k)})^{-1}\mathbf{x}^{(k)} = (\mathbf{x}^{(k-1)})^t(\mathbf{\Sigma}^{(k-1)})^{-1}\mathbf{x}^{(k-1)} + \alpha(\begin{bmatrix} \mathbf{v}^t & \frac{1}{\alpha} \end{bmatrix}\mathbf{x}^{(k)})^2$$

where $(\mathbf{\Sigma}^{(k)})^{-1} = \begin{bmatrix} A & \mathbf{v} \\ \mathbf{v}^t & \frac{1}{\alpha} \end{bmatrix}$

Determinant update for augmented feature set

$$\log\left(|\mathbf{\Sigma}^{(k)}|\right) = \log\left(|\mathbf{\Sigma}^{(k-1)}|\right) + \log\alpha$$

Example: update rule for decision function

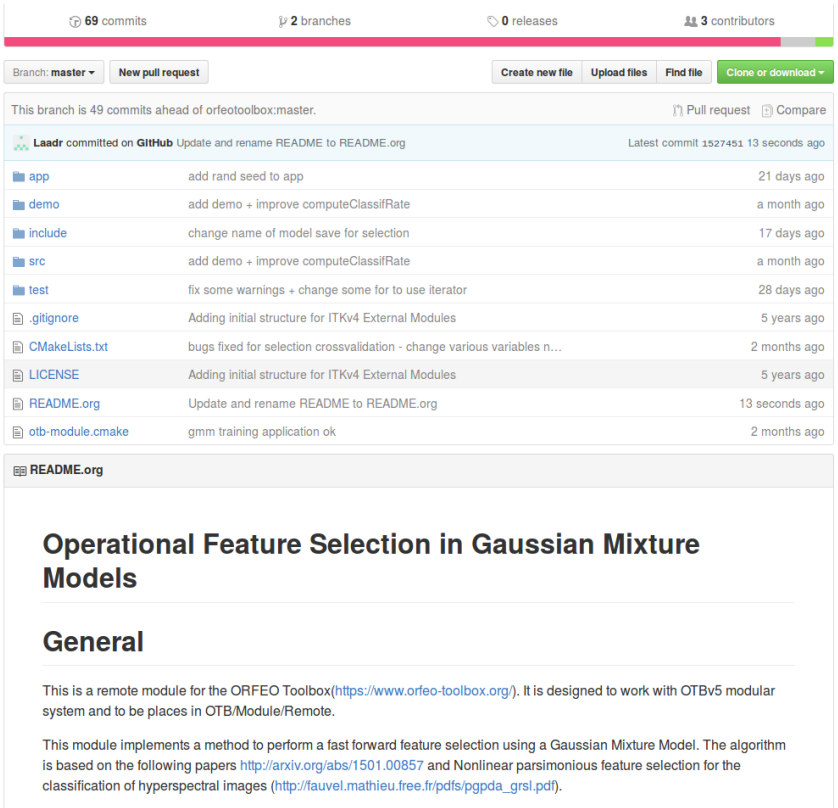
$$
\begin{aligned}
Q_c(\mathbf{x}) = & -(\mathbf{x}^{(k-1)} - \boldsymbol{\mu_c}^{(k-1)})^t (\boldsymbol{\Sigma_c}^{(k-1)})^{-1} (\mathbf{x}^{(k-1)} - \boldsymbol{\mu_c}^{(k-1)}) \\
& - \log\left(|\boldsymbol{\Sigma_c}^{(k-1)}|\right) + 2\log(\pi_c) + k\log(2\pi) \\
& - \alpha(\begin{bmatrix} \mathbf{v}^t & \frac{1}{\alpha} \end{bmatrix} (\mathbf{x}^{(k)} - \boldsymbol{\mu_c}^{(k)}))^2 - \log\alpha
\end{aligned}
$$

} Computed once for all

} Computed for each set

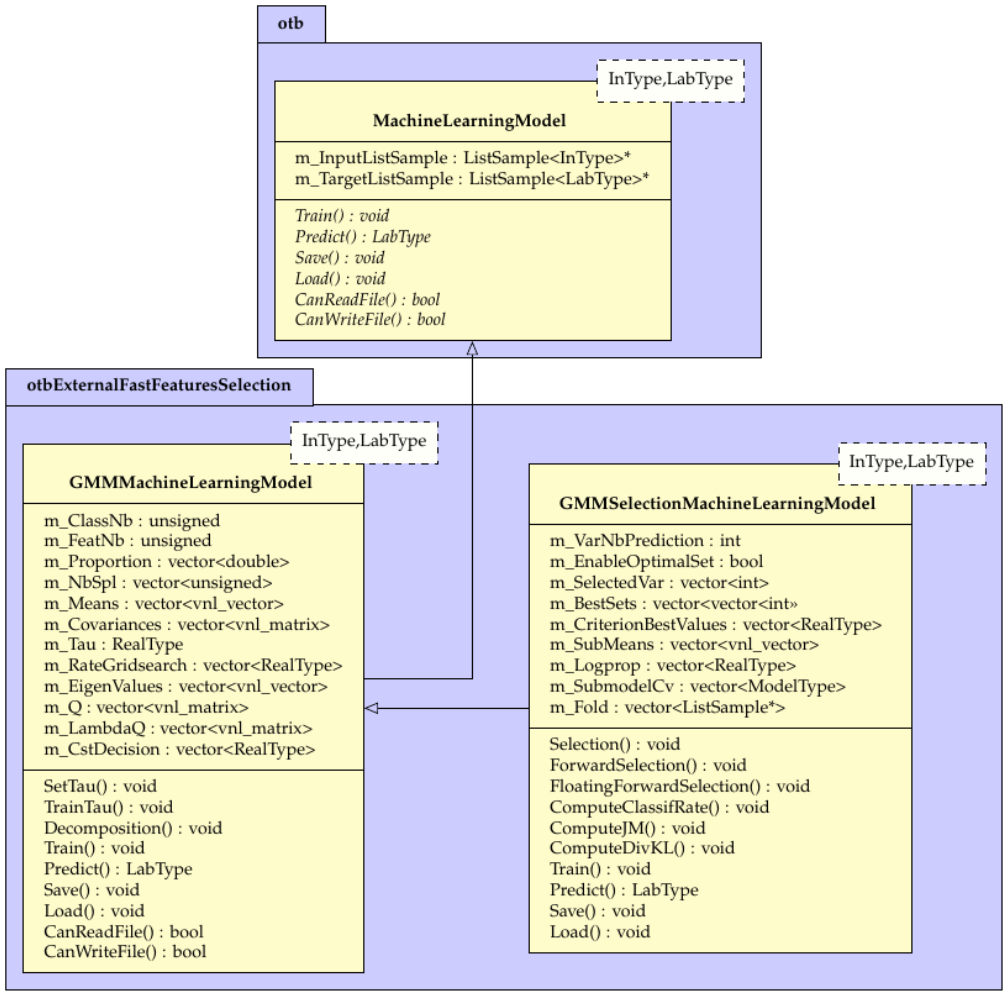
Production

- A Python module: `https://github.com/Laadr/FFFS`.
- A C++ module: a remote module for Orfeo Toolbox v5.4 (CNES) `https://github.com/Laadr/otbExternalFastFeaturesSelection`.



Module OTB: code

- Fork from otb external module template.
- Inherit from MachineLearningModel.



Module OTB: applications

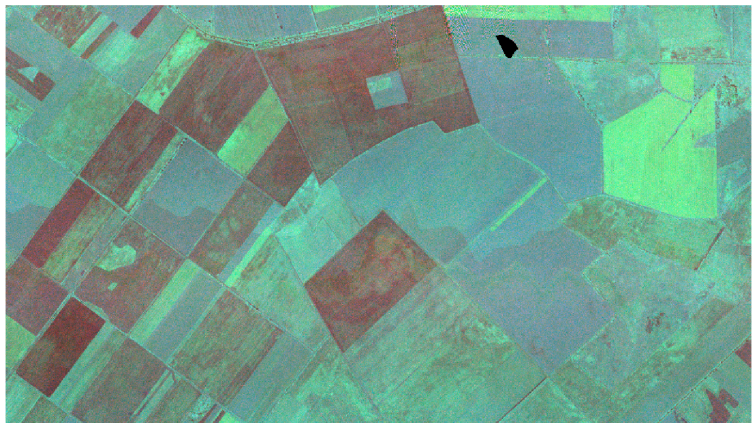
- **otbcli_TrainGMMApp**: train a GMM classifier and optionally perform a ridge regularization with an embedded selection of the regularization parameter;
- **otbcli_TrainGMMSelectionApp**: train a GMM classifier and perform a feature selection algorithm (SFS, SFFS) with various possible criterion (J-M dist., KL div., Cohen's kappa, overall accuracy, mean F1-score);
- **otbcli_PredictGMMApp**: perform a classification of the input image using a GMM model.

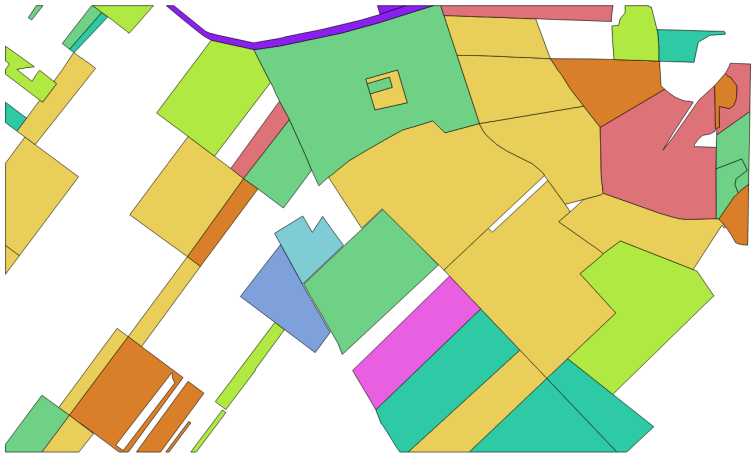
Module OTB: possible improvement

- Modify training applications to use the new sampling module;
- Parallelize code for training and selection:
  - model parameters estimation,
  - cross-validation,
  - criterion function estimation.

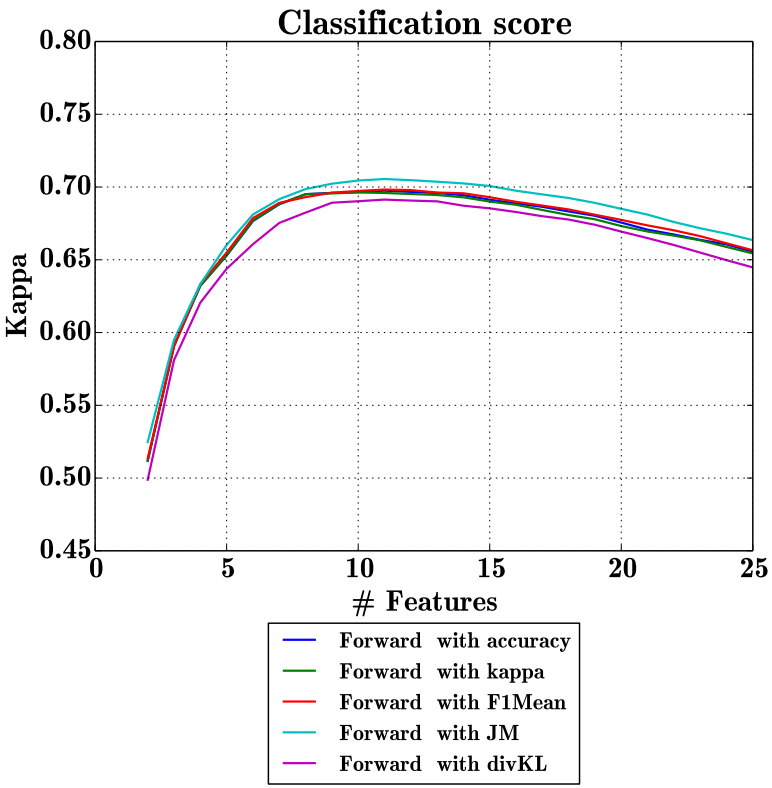
Aisa dataset

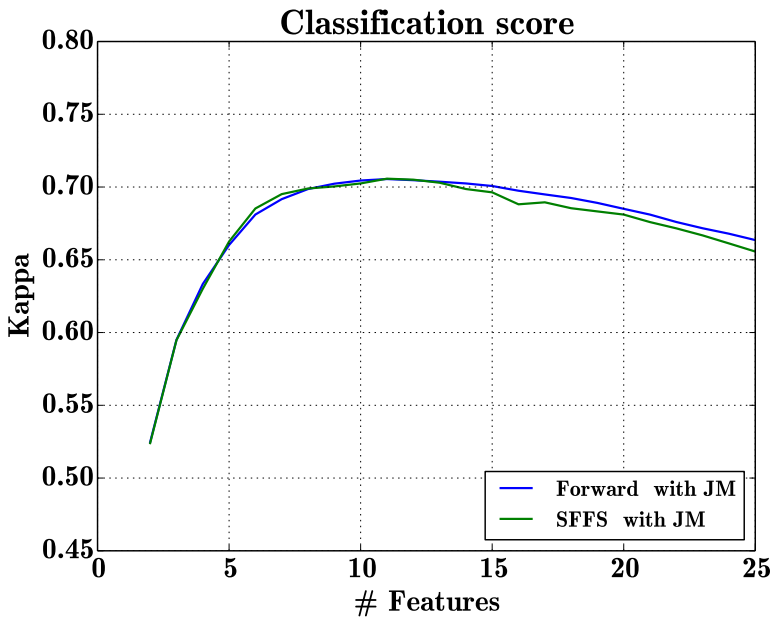
- Hyperspectral image acquired by the AISA Eagle sensor during a flight campaign over Heves, Hungary.
- 252 bands ranging from 395 to 975 nm.
- 16 classes with 361,971 referenced pixels.
- Classification with spatial stratification (separate polygons for training and validation).





Selection: method variation tested on Aisa





No advantage with floating forward variation.

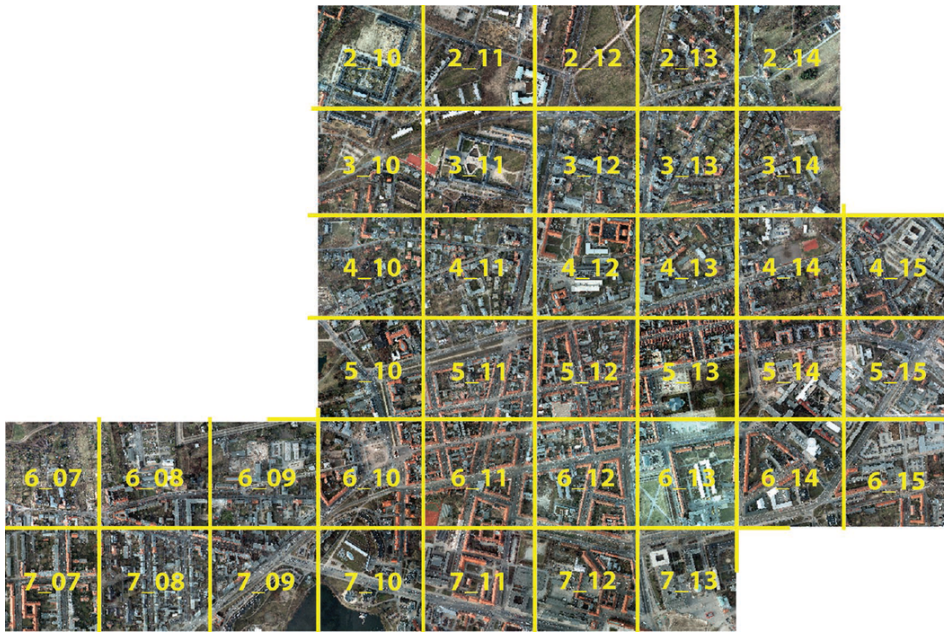
Best results with Jeffries-Matusita distance.

Aisa dataset: results

	Cohen's kappa			**Training time (s)**			**Classification time (s)**		
# samples by class	250	500	1000	250	500	1000	250	500	1000
GMM SFS kappa	**0.684**	**0.698**	**0.711**	257	496	955	**7.7**	**8.6**	**8.7**
GMM SFS JM	**0.680**	**0.700**	**0.710**	**8.6**	**8.9**	**9.1**	**9.7**	**9.8**	**9.6**
GMM SFFS JM	**0.680**	**0.700**	**0.710**	**8.8**	**9.0**	**9.3**	**9.7**	**9.8**	**9.8**
GMM Ridge	0.611	0.620	0.642	71.7	105	167	530	530	530
KNN	0.551	0.563	0.574	**8.9**	19.6	59.7	387	639	887
Random Forest	0.645	0.673	**0.693**	24.5	49.3	105	33.0	41.7	45.9

Potsdam dataset

- Aerial images of Potsdam urban area distributed by the International Society for Photogrammetry and Remote Sensing (ISPRS).
- Patches of 6000x6000 pixels with a 5cm-resolution and 4 channels (Red, Blue, Green and Infrared) + Digital Surface Model (DSM).
- 6 classes: Low vegetation, High vegetation, Impervious surfaces, Buildings, Cars, Clutter.



Potsdam dataset: extracted features

- Radiometric indexes: NDVI, TNDVI, RVI, SAVI, TSAVI, MSAVI, MSAVI2, GEMI, IPVI, NDWI2, NDTI, RI, CI, BI, BI2 (15 features);
- Morphological profile with reconstruction of each band with a disk of radius 5, 9, 13, 17, 21, 25, 29, 33, 37 and 41 (80 features);
- Attribute profile of each band with area as attribute and 1000, 2000, 5000, 10000 and 15000 as thresholds (40 features);
- Attribute profile of each band with diagonal of bounding box as attribute and 100, 200, 500, 1000 and 20000 as thresholds (40 features);
- Textural features with neighborhood of 19x19 pixels: mean, standard deviation, range and entropy (16 features);
- DSM and raw image (5 features)

Potsdam dataset: extracted features

- Radiometric indexes: NDVI, TNDVI, RVI, SAVI, TSAVI, MSAVI, MSAVI2, GEMI, IPVI, NDWI2, NDTI, RI, CI, BI, BI2 (15 features);
- Morphological profile with reconstruction of each band with a disk of radius 5, 9, 13, 17, 21, 25, 29, 33, 37 and 41 (80 features);
- Attribute profile of each band with area as attribute and 1000, 2000, 5000, 10000 and 15000 as thresholds (40 features);
- Attribute profile of each band with diagonal of bounding box as attribute and 100, 200, 500, 1000 and 20000 as thresholds (40 features);
- Textural features with neighborhood of 19x19 pixels: mean, standard deviation, range and entropy (16 features);
- DSM and raw image (5 features)

196 features stacked to create new images with 196 bands.

Potsdam dataset: results

- Training set: 1000 samples by class

	Cohen's kappa of 5_11 (train)	Cohen's kappa of 5_12 (test)	Training time (s)	Classification time (s)	# features used to classify
GMM SFS kappa	**0.698** (0.008)	**0.679** (0.007)	100	**300**	13.8
GMM SFS JM	0.460 (0.221)	0.453 (0.226)	1	**400**	29.4
GMM SFFS JM	0.460 (0.221)	0.453 (0.226)	1	**400**	29.4
GMM ridge	0.521 (0.010)	0.478 (0.014)	10	2000	all
KNN	0.536 (0.005)	0.468 (0.002)	1	1000	all
Random Forest	**0.734** (0.003)	**0.680** (0.005)	20	800	all

- Training set: 50000 samples by class

	Cohen's kappa of 5_11 (train)	Cohen's kappa of 5_12 (test)	Training time (s)	Classification time (s)	# features used to classify
GMM SFS kappa	**0.710** (0.007)	**0.681** (0.007)	5000	**400**	24.5
GMM SFS JM	0.461 (0.084)	0.442 (0.088)	5	**400**	30
GMM SFFS JM	0.476 (0.111)	0.468 (0.108)	5	**400**	27.6
GMM ridge	0.508 (0.000)	0.467 (0.001)	500	2000	all
Random Forest	**0.854** (0.001)	**0.720** (0.001)	2000	2000	all

Conclusions and perspectives

- An algorithm for the classification of high dimension data was designed.
  - GMM model to select iteratively the most relevant features.
  - Few or no advantages with floating forward variation.
- An efficient implementation available as a remote module of the Orfeo toolbox.
  - Good performances compared to standard classifiers implemented in Orfeo toolbox.

- Need to investigate robustness to Gaussian hypothesis (Jeffries-Matusita vs kappa).
- Stability of selected bands could be improve by selecting continuous intervals.