Kernel Matrix Approximation for Learning the Kernel Hyperparameters

M. Fauvel and D. Sheeren

DYNAFOR, INRA & ENSAT, INPT, University de Toulouse - France

2e colloque scientifique de la SFTH

18 & 19 Juin 2012 - ONERA Toulouse

June 19, 2012

Kernel methods

Kernel matrix approximation

Experimental results

Conclusions and perspectives

Kernel methods

Kernel matrix approximation

Experimental results

Conclusions and perspectives

Kernel methods in hyperspectral imagery

Kernels methods are popular and effective algorithms, which are widely used for many applications [1]:

- Classification and detection,
- Biophysical parameter estimation,
- Unmixing, ...

They are well suitable for the processing of hyperspectral images:

- KM are robust to the high spectral dimension,
- Joint spatial and spectral processing are easy with KM,
- Few hyperparameters to tune,
- Very good results.

[1] G. Camps-Valls and L. Bruzzone, *Kernel Methods for Remote Sensing Data Analysis*, Wiley, 2009.

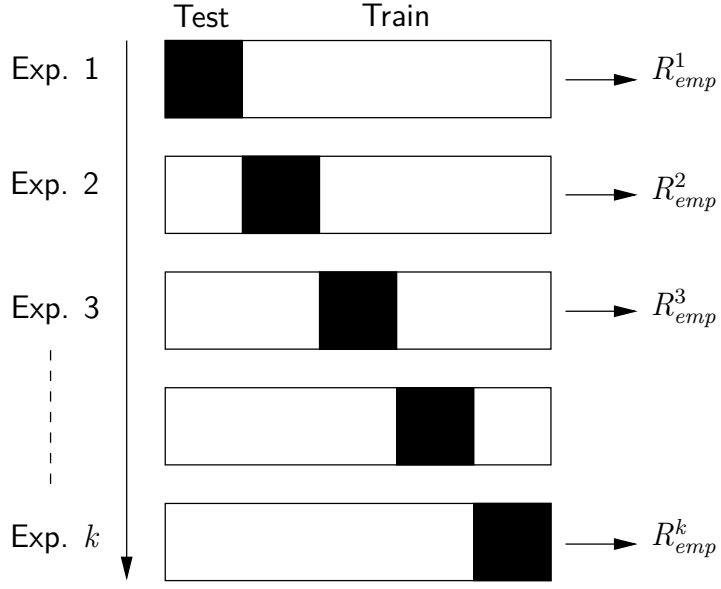
Kernel methods methodology

Kernel method can be decomposed into three steps:

1. Choose the kernel:
   - $k(\mathbf{x}_i, \mathbf{x}_j) = \Big(\langle \mathbf{x}_i, \mathbf{x}_j \rangle + 1\Big)^p$,
   - $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$.
2. Tune the hyperparameters $\mathbf{p}$ (e.g., p or σ^2).
3. Learn the parameters of the processing rule, i.e., solve a (constrained) linear optimization problem.
   - Ridge regression: $\hat{\boldsymbol{\alpha}} = \Big(\mathbf{K} + \lambda \mathbf{I}\Big)^{-1} \mathbf{y}$
   - Support vectors machines : $\hat{\boldsymbol{\alpha}} = \max_{\boldsymbol{\alpha}} \left[\boldsymbol{\alpha}^t \mathbf{1} - \frac{1}{2}\boldsymbol{\alpha}^t \mathbf{K} \boldsymbol{\alpha}\right]$ subject to $0 \preceq \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^t \mathbf{y} = 0$.

Choosing the hyperparameters 1/2

- Crucial step: improve or decrease drastically the performances of KM
- Cross validation is conventionally used. CV estimates the expected error R.



- $R(\mathbf{p}) \approx \frac{1}{k} \sum_{i=1}^{k} R^i_{emp}$
- Good behavior in various supervised learning problem but **high computational load**.

Choosing the hyperparameters 2/2

Others strategies: Optimization of an upper bound of the expected error, e.g., the radius-margin bound or the span bound[2].

- Gradient based approaches,
- Genetic approaches.

However:

- Non convex optimization problem,
- Cannot manage a lot of training samples.

[2] Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S., *Choosing Multiple Parameters for Support Vector Machines*, Machine Learning, 2002.

Choosing the hyperparameters 2/2

Others strategies: Optimization of an upper bound of the expected error, e.g., the radius-margin bound or the span bound[2].

- Gradient based approaches,
- Genetic approaches.

However:

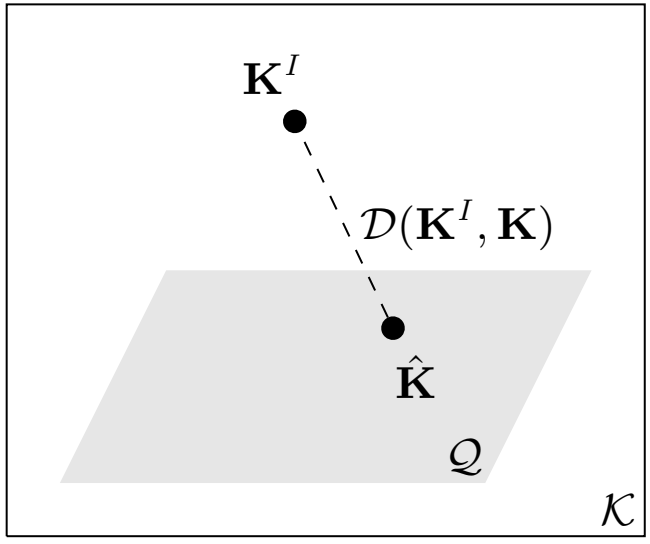
- Non convex optimization problem,
- Cannot manage a lot of training samples.

The upper bound depends on $\hat{\alpha}$

[2] Chapelle, O., Vapnik, V., Bousquet, O. and Mukherjee, S., *Choosing Multiple Parameters for Support Vector Machines*, Machine Learning, 2002.

Kernel target alignment

- Kernel target alignment **measures the degree of agreement between a kernel and a learning task**.
- No need to compute $\hat{\alpha}$.
- Exhaustive search or optimization of the alignment.
- Positively applied to remote sensing.
- Interesting formulation of the problem:
  Approximation of an ideal kernel matrix.



Kernel methods

Kernel matrix approximation

Experimental results

Conclusions and perspectives

Kernel matrix approximation principles for classification

- Training set $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n$, $\mathbf{x}_i \in \mathbb{R}^d$.
- Gaussian kernel: $k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$ $\rightsquigarrow$ similarity measure between $\mathbf{x}_i$ and $\mathbf{x}_j$.
- In the ideal situation: $k(\mathbf{x}_i, \mathbf{x}_j) \approx 1$ if $y_i = y_j$; $k(\mathbf{x}_i, \mathbf{x}_j) \approx 0$ otherwise.
- **Empirical ideal kernel**:

$$k^I(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} 1 \text{ if } y_i = y_j, \\ 0 \text{ otherwise.} \end{cases}$$

- **KMA principle**: Find the hyperparameter σ^2 such as the ideal conditions are fulfilled (as much as possible) for all $(\mathbf{x}_i, \mathbf{x}_j)$.

Definitions

- Kernel matrix:

$$\mathbf{K} = \begin{pmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \dots & k(\mathbf{x}_1, \mathbf{x}_n) \\ k(\mathbf{x}_2, \mathbf{x}_1) & k(\mathbf{x}_2, \mathbf{x}_2) & \dots & k(\mathbf{x}_2, \mathbf{x}_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_n, \mathbf{x}_1) & k(\mathbf{x}_n, \mathbf{x}_2) & \dots & k(\mathbf{x}_n, \mathbf{x}_n) \end{pmatrix}$$

- Frobenius inner product:

$$\langle \mathbf{K}_1, \mathbf{K}_2 \rangle_F = \sum_{i,j=1}^{n} (\mathbf{K}_1)_{ij} \times (\mathbf{K}_2)_{ij} = \sum_{i,j=1}^{n} k_1(\mathbf{x}_i, \mathbf{x}_j) k_2(\mathbf{x}_i, \mathbf{x}_j)$$

Similarity measure between kernel matrices

- **Alignment**:

$$A(\boldsymbol{\sigma}) = \frac{\langle \mathbf{K}, \mathbf{K}^I \rangle_F}{\|\mathbf{K}\|_F \|\mathbf{K}^I\|_F}$$

- **Frobenius distance** (equivalent to mean square error):

$$D(\boldsymbol{\sigma}) = \frac{\|\mathbf{K} - \mathbf{K}^I\|_F^2}{n^2} = \frac{\|\mathbf{K}\|_F^2 + \|\mathbf{K}^I\|_F^2 - 2\langle \mathbf{K}, \mathbf{K}^I \rangle_F}{n^2}$$

- **Correlation**:

$$C(\boldsymbol{\sigma}) = \frac{\langle \mathbf{K} - \bar{\mathbf{K}}, \mathbf{K}^I - \bar{\mathbf{K}}^I \rangle_F}{\|\mathbf{K} - \bar{\mathbf{K}}\|_F \|\mathbf{K}^I - \bar{\mathbf{K}}^I\|_F}$$

where $\bar{\mathbf{K}} := \left[\frac{1}{n^2} \sum_{i,j=1}^{n} k(\mathbf{x}_i, \mathbf{x}_j)\right] \mathbf{1}$ and $\mathbf{1}$ is the n-square matrix of ones.

Similarity vs Expected error

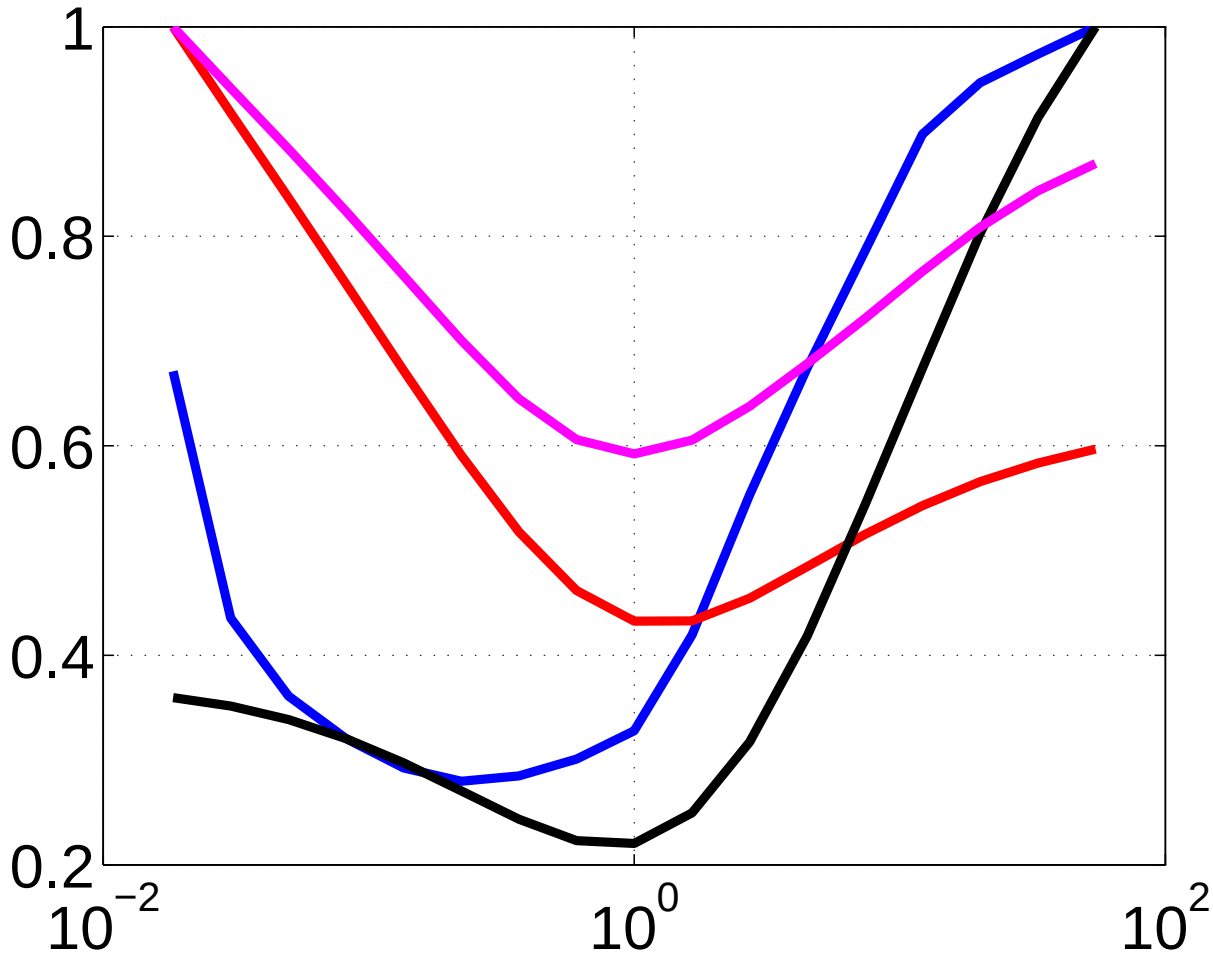


Figure: Normalized value of classification errors estimated with CV (in blue), A (in red), D (in black) and C (in magenta) for the *University Area*. The horizontal axis correspond to the value of the parameter σ^2 in log scale and the vertical axis correspond to the normalized value of CV, A, D and C. These values have been normalized for the purpose of visualization.

Optimization of the hyperparameters 1/2

- A, D and C are derivable w.r.t. σ^2:

$$\begin{aligned}
\frac{\partial A(\sigma^2)}{\partial \sigma^2} &= \frac{1}{\|\mathbf{K}^I\|_F}\left[\frac{\left\langle \mathbf{K}^I, \frac{\partial \mathbf{K}}{\partial \sigma^2}\right\rangle_F}{\|\mathbf{K}\|_F} - \frac{\langle \mathbf{K}, \mathbf{K}^I\rangle_F \left\langle \mathbf{K}, \frac{\partial \mathbf{K}}{\partial \sigma^2}\right\rangle_F}{\|\mathbf{K}\|_F^{3/2}}\right] \\
\frac{\partial D(\sigma^2)}{\partial \sigma^2} &= \frac{2}{n^2}\left\langle \mathbf{K} - \mathbf{K}^I, \frac{\partial \mathbf{K}}{\partial \sigma^2}\right\rangle_F \\
\frac{\partial C(\sigma^2)}{\partial \sigma^2} &= \frac{1}{\|\mathbf{K}^I - \bar{\mathbf{K}}^I\|_F}\left[\frac{\left\langle \mathbf{K}^I - \bar{\mathbf{K}}^I, \frac{\partial \mathbf{K}}{\partial \sigma^2} - \frac{\partial \bar{\mathbf{K}}}{\partial \sigma^2}\right\rangle_F}{\|\mathbf{K} - \bar{\mathbf{K}}\|_F}\right. \\
&\quad \left. - \frac{\langle \mathbf{K} - \bar{\mathbf{K}}, \mathbf{K}^I - \bar{\mathbf{K}}^I\rangle_F \left\langle \mathbf{K} - \bar{\mathbf{K}}, \frac{\partial \mathbf{K}}{\partial \sigma^2} - \frac{\partial \bar{\mathbf{K}}}{\partial \sigma^2}\right\rangle_F}{\|\mathbf{K} - \bar{\mathbf{K}}\|_F^{3/2}}\right]
\end{aligned}$$

- For the Gaussian kernel:

$$\left(\frac{\partial \mathbf{K}}{\partial \sigma^2}\right)_{ij} = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^4} k(\mathbf{x}_i, \mathbf{x}_j).$$

Optimization of the hyperparameters 2/2

- Positivity is obtained by optimizing according to $\ln(\sigma)$:

$$\left(\frac{\partial \mathbf{K}}{\partial \ln(\sigma)}\right)_{ij} = \left(\frac{\partial \mathbf{K}}{\partial \sigma^2}\right)_{ij} \frac{\partial \sigma^2}{\partial \ln(\sigma)} = 2\sigma^2 \left(\frac{\partial \mathbf{K}}{\partial \sigma^2}\right)_{ij},$$

- Finally, the derivative is simply computed as:

$$\left(\frac{\partial \mathbf{K}}{\partial \ln(\sigma)}\right)_{ij} = -2 \log\left((\mathbf{K})_{ij}\right)(\mathbf{K})_{ij},$$

- Newton method for the optimization of the hyperparameter,
- Hessian matrix is computable at reduced cost.

Kernel methods

Kernel matrix approximation

Experimental results

Conclusions and perspectives

Experimental setup

- Data scaled between [-1,1] for each variable,
- 12.5%-25% of the total number of pixels used for training,
- Experiments have been repeated 20 times,
- Comparison with conventional CV($\sigma^2 \in [2^{-5}, 2^{-4.5}, \ldots, 2^3]$),
- LIBSVM solver,
- One vs one multiclass strategy.

ROSIS-03

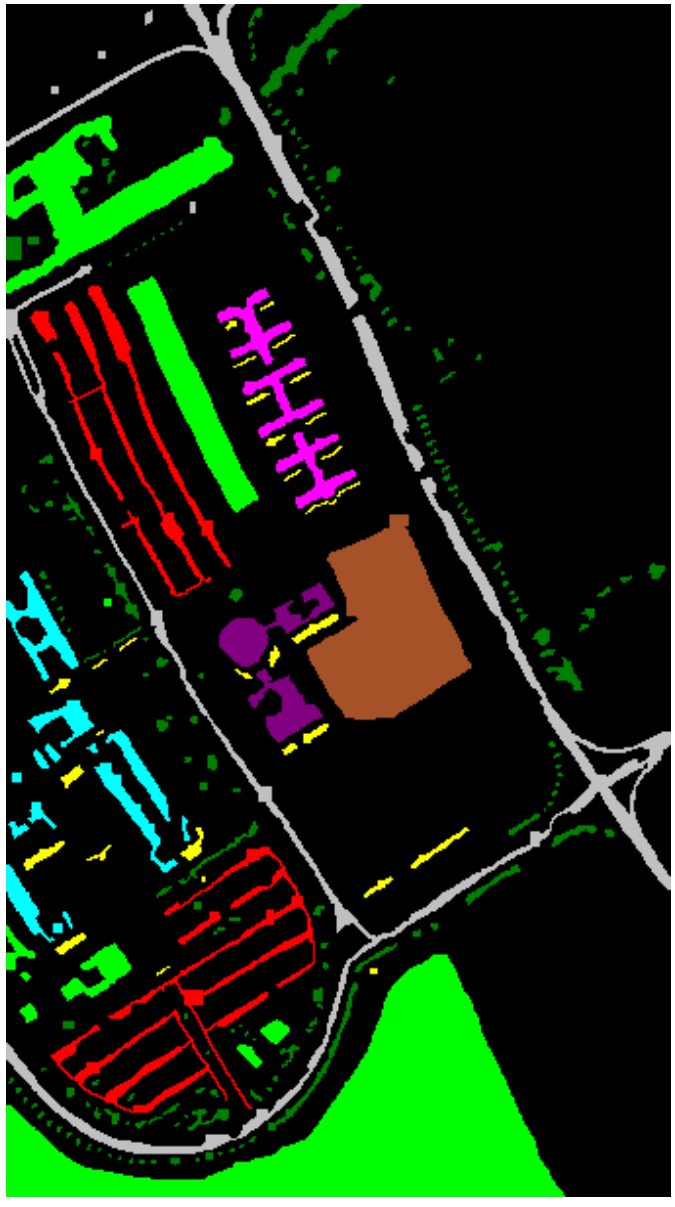
University Area, Pavia - Italy

- Airbone,
- [H W]=[610 340],
- 103 channels,
- 1.3 m/pixel,
- 42776 referenced samples,
- 9 classes : Asphalts, Meadow, Gravel, Tree, Metal Sheet, Bare Soil, Bitumen, Brick and Shadow.

ROSIS-03

University Area, Pavia - Italy

- Airbone,
- [H W]=[610 340],
- 103 channels,
- 1.3 m/pixel,
- 42776 referenced samples,
- 9 classes : Asphalts, Meadow, Gravel, Tree, Metal Sheet, Bare Soil, Bitumen, Brick and Shadow.



Results

- Global accuracies & processing time

Method	OA	κ	Proc. time (s)
CV	**94.1 (0.13)**	**0.92 (1.8$\times 10^{-3}$)**	325.0 (6.1)
A	92.4 (0.23)	0.90 (3.2$\times 10^{-3}$)	**58.9 (12.0)**
D	93.3 (0.19)	0.91 (2.5$\times 10^{-3}$)	70.3 (37.6)
C	93.0 (0.24)	0.90 (3.2$\times 10^{-3}$)	114.0 (41.4)

- Optimal hyperparameter:

	CV	A	D	C
σ^2	0.17	1.28	0.81	1.02

- Default hyperparameter of LIBSVM ($\sigma^2 = 0.5 * d \approx 50$) $\rightsquigarrow$ OA=78%.

HySpex

Village of Villelongue, France

- Airbone,
- [H W]=[1000 2000],
- 160 channels,
- 0.5 m/pixel,
- 32016 referenced samples,
- 10 woody classes: Ash tree, Chestnut tree, Lime tree, Hazel tree ...

Results

- Global accuracies & processing time

Method	OA	κ	Proc. time (s)
CV	**95.6 (0.12)**	**0.94 (1.4$\times 10^{-3}$)**	2249.4 (48.9)
A	95.4 (0.13)	0.94 (1.6$\times 10^{-3}$)	**68.9 (8.2)**
D	95.6 (0.13)	0.95 (1.6$\times 10^{-3}$)	177.3 (28.2)
C	95.6 (0.13)	0.95 (1.6$\times 10^{-3}$)	82.6 (46.0)

- Default hyperparameter: OA $\approx 34\%$,
- Alignment is about 30 times faster.

Kernel methods

Kernel matrix approximation

Experimental results

Conclusions and perspectives

Conclusions and perspectives

- **Conclusions**
  - The approach is effective for tuning the hyperparameters,
  - Fast and accurate,
  - Multiple hyperparameters (ellipsoidal Gaussian kernel) have been also investigated but results are not convincing.

- **Perspectives**
  - Ideal kernel for regression, inversion?
  - Other kernels?
  - Optimization for multiple hyperparameters?

Kernel Matrix Approximation for Learning the Kernel Hyperparameters

M. Fauvel and D. Sheeren

DYNAFOR, INRA & ENSAT, INPT, University de Toulouse - France

2e colloque scientifique de la SFTH

18 & 19 Juin 2012 - ONERA Toulouse

June 19, 2012