Support Vector Machine basierte Klassifikation in der Geofernerkundung

*Andreas Rabe**
Sebastian van der Linden
Patrick Hostert

*andreas.rabe@geo.hu-berlin.de

Support Vector Machines (SVMs)
What would be of interest for the audience?

Good news:

- a SVM is a state-of-the-art classifier (fits arbitrary class boundaries)
- is widely used inside remote sensing applications
- works well in high-dimensional feature spaces (hyperspectral data)

Bad news:

- wrong usage leads to overfitting or underfitting
- mostly used as a black box (complex mathematics)
- nearly never used with one- or two-dimensional date

Take-home-message:

- you can always avoid overfitting or underfitting when using SVM
- you can use SVM as a black box, ...
- ... but you could gain a deeper understanding by looking at simple one- or two-dimensional examples

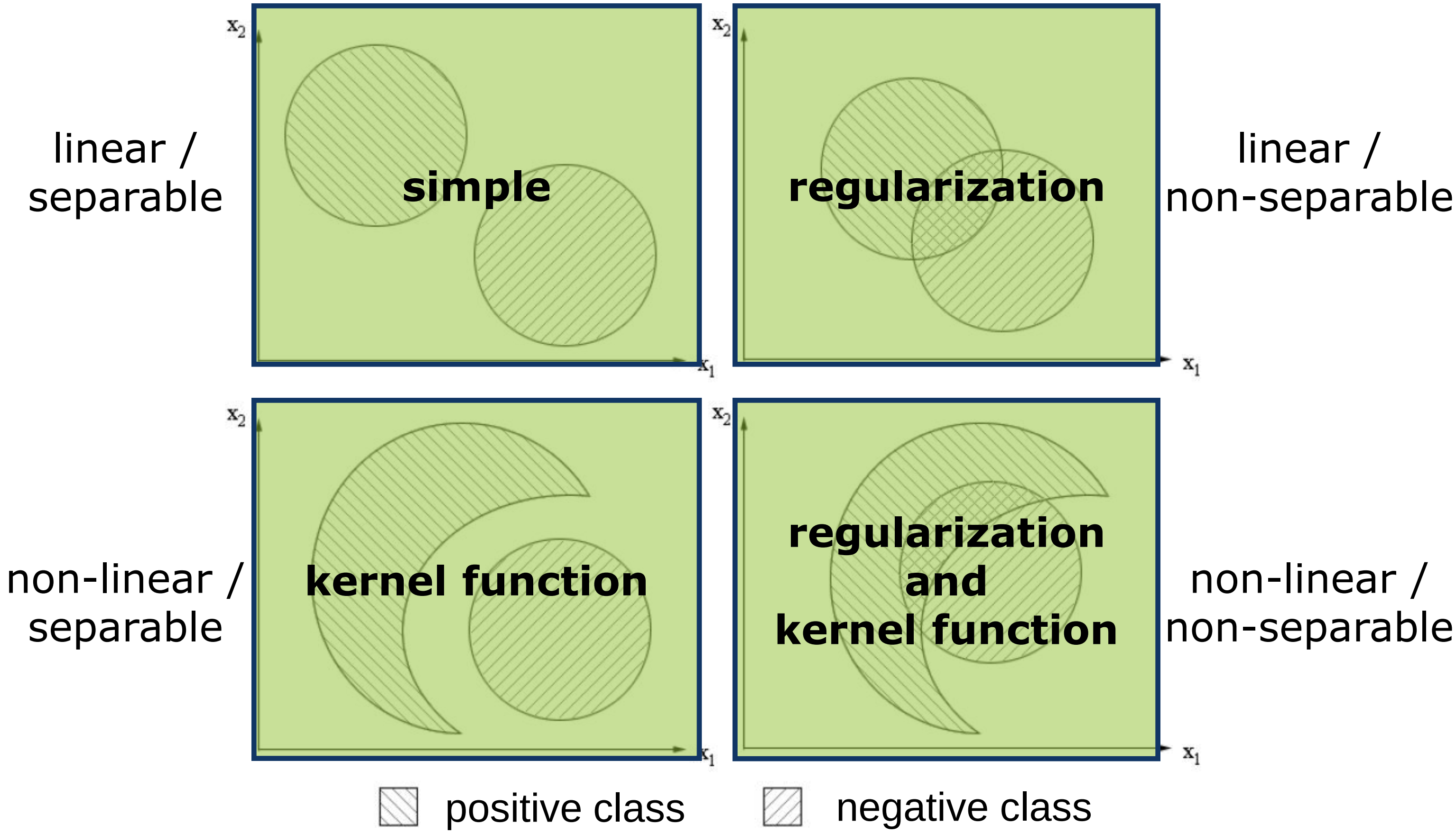
Support Vector Machines (SVMs)
What would be of interest for the audience?

This talk...

- is not about the mathematics and theory behind SVMs.
- is not about specific remote sensing applications → colored maps are not helpful!

- is about understanding the concepts behind SVM and the influence of parameters.
- is about learning from simple one- or two-dimensional examples, to be able to generalize to high-dimensional, real world problems.

Different settings for binary classification in 2D



To train a SVM we need to set appropriate parameter values for the kernel function (e.g.RBF kernel with parameter *g*) and for the regularization (parameter *C*).

SVM overview

A Support vector machine (SVM) ...

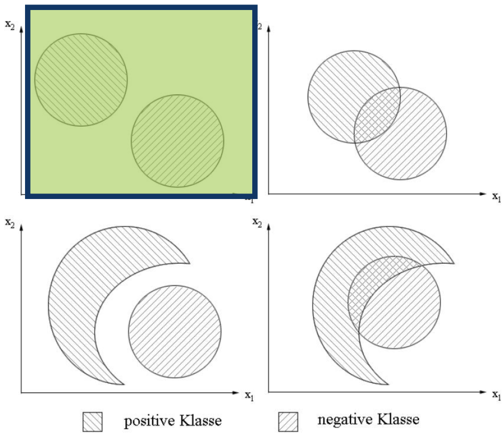
... is a **universal learning machine** for
- pattern recognition (classification),
- regression estimation and
- distribution estimation.

... can be seen as an implementation of Vapnik's **Structural Risk Minimisation** principle inside the context of **Statistical Learning Theory** (Vapnik1998).

not today

SVM classification overview

The optimal separating hyperplane.



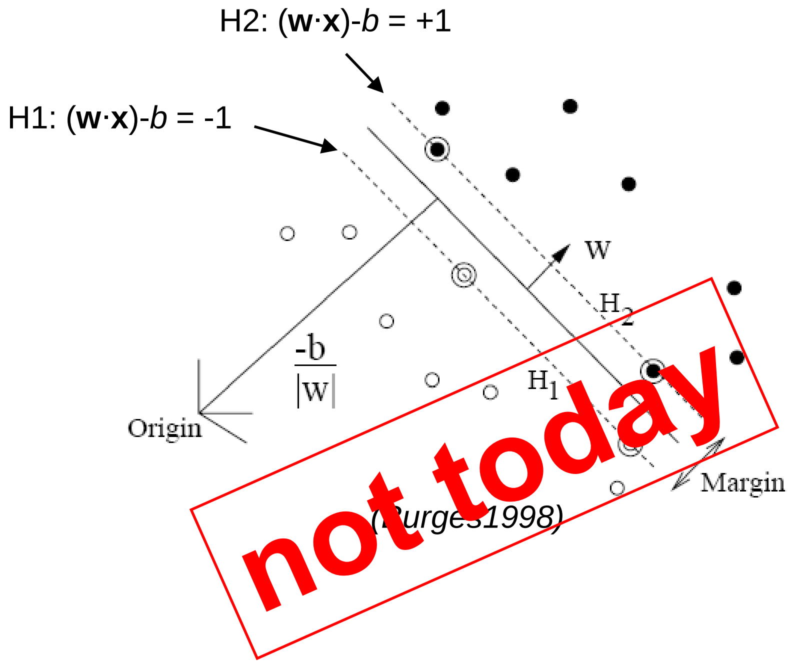
Suppose the training set:

$$(\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_l,y_l),\ \mathbf{x}\in R^n,\ y\in\{+1,-1\},$$

can be separated by a hyperplane

$$(\mathbf{w}\cdot\mathbf{x})-b=0.$$

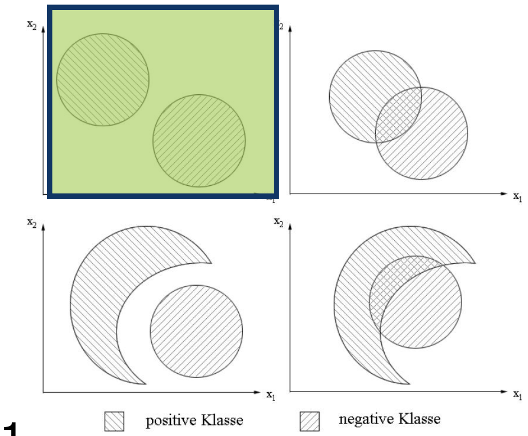
The **optimal separating hyperplane** separates the vectors without error and maximizes the **margin** between the closest vectors to the hyperplane.



(Burges1998)

SVM classification overview

The optimal separating hyperplane.



To construct the optimal separating hyperplane one has to solve a quadratic optimization problem:

Minimize the functional $$L\ \mathbf{w} = \frac{1}{2}\ \mathbf{w}\cdot\mathbf{w}$$

under the constraints:
$$\mathbf{w}\cdot\mathbf{x}_i + b \geq +1,\ \text{if } y_i = +1$$
$$\mathbf{w}\cdot\mathbf{x}_i + b \leq -1,\ \text{if } y_i = -1$$

Formulated as lagrange functional:

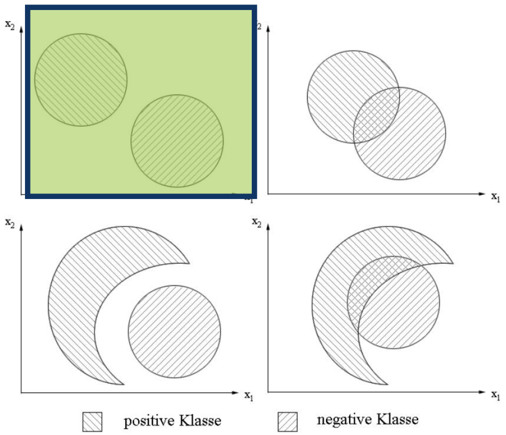
Maximize the functional $$W\ \mathbf{a} = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j=1}^{l} a_i a_j y_i y_j\ \mathbf{x}_i \mathbf{x}_j$$

under the constraints: $$\sum_{i=1}^{l} a_i y_i = 0 \text{ and } a_i \geq 0.$$

not today

SVM classification overview

The optimal separating hyperplane.



Let $\mathbf{a}^0 = (a_1^0, \ldots, a_l^0)$ be a solution to this quadratic optimization problem.

The optimal hyperplane $\mathbf{w}_0$ is a linear combination of the vectors of the training set.

$$\mathbf{w}_0 = \sum_{i=1}^{l} a_i^0 y_i \mathbf{x}_i$$

The decision rule $y(\mathbf{x})$ is based on the sign of the decision function $f(\mathbf{x})$:

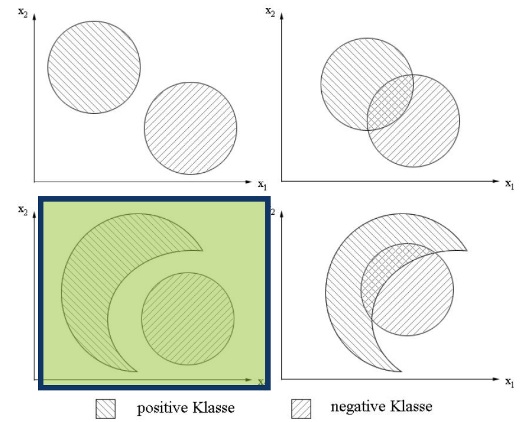
$$f(\mathbf{x}) = \mathbf{w}_0 \mathbf{x} - b_0 \qquad = \sum_{i=1}^{l} a_i^0 y_i (\mathbf{x}_i \mathbf{x}) - b_0$$

$$y(\mathbf{x}) = sign(f(\mathbf{x}))$$

not today

SVM classification overview

Kernel Function



When looking at the lagrange functional:

$$W(\mathbf{a}) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i,j=1}^{l} a_i a_j y_i y_j (\mathbf{x}_i \mathbf{x}_j)$$

it can be observed, that only **dot products** between vectors in the **input space** are calculated.

The idea is to replace the dot product in the **input space** by the dot product in a higher dimensional **feature space**, defined by a kernel function $K(\mathbf{x},\mathbf{x}_i)$.

Polynomial kernel: $K(\mathbf{x},\mathbf{x}_i) = ((\mathbf{x} \cdot \mathbf{x}_i) + 1)^d$

Gaussian RBF kernel: $K(\mathbf{x},\mathbf{x}_i) = \exp(-g|\mathbf{x} - \mathbf{x}_i|^2)$

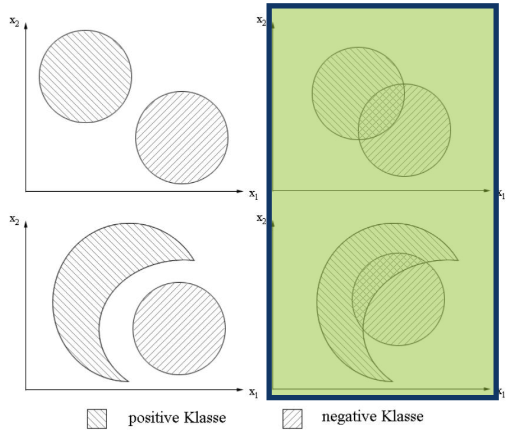
This leads to a non-linear decision function:

$$f(\mathbf{x}) = \sum_{i=1}^{l} a_i^0 y_i K(\mathbf{x}_i, \mathbf{x}) - b_0$$

not today

SVM classification overview

Regularization



The concept of maximizing the margin between classes must be modified, to be able to handle **non-separable** classes.

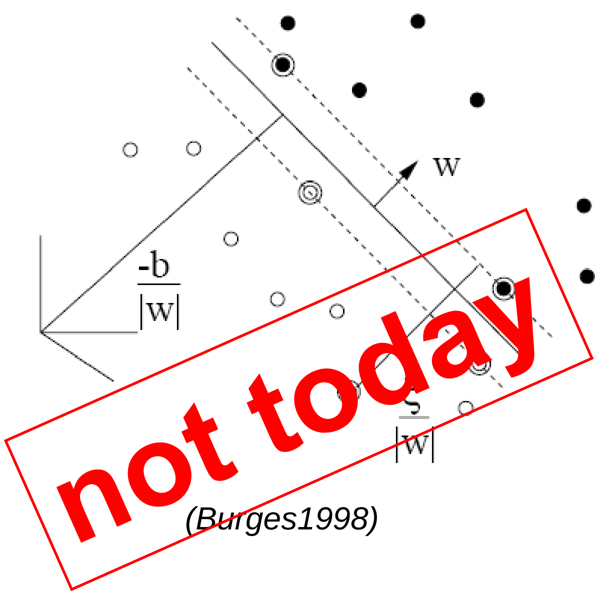
We introduce so-called slack variables $\xi = (\xi_1, \ldots, \xi_l)$, one for each vector in the training set.

Minimize the functional

$$L(\mathbf{w}) = \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) + C\sum_{i=1}^{l}\xi_i$$

under the constraints:

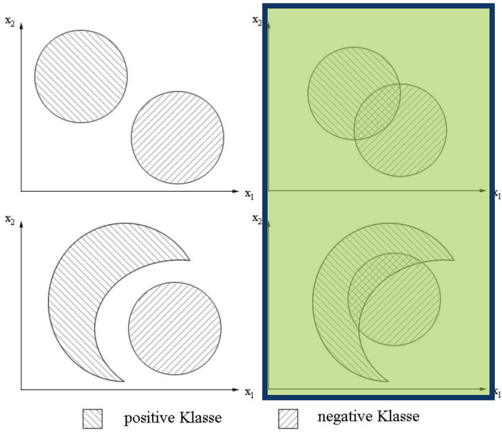
$$\mathbf{w}\cdot\mathbf{x}_i + b \geq +1 - \xi_i, \quad \text{if } y_i = +1$$
$$\mathbf{w}\cdot\mathbf{x}_i + b \leq -1 + \xi_i, \quad \text{if } y_i = -1$$
$$\xi_i \geq 0 \quad \forall i$$



(Burges1998)

SVM classification overview

Regularization



Formulated as lagrange functional:

Maximize the functional

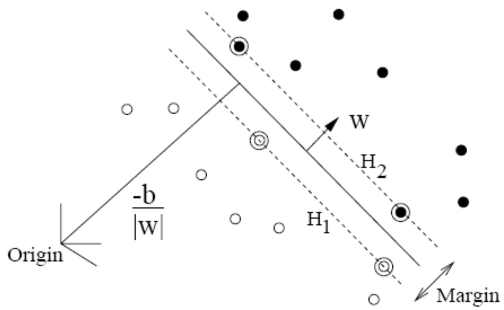
$$W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

under the constraints:

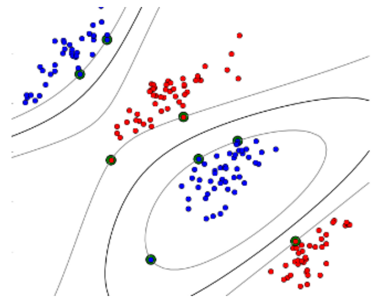
$$\sum_{i=1}^{l} \alpha_i y_i = 0 \text{ and } 0 \leq \alpha_i \leq C$$

not today

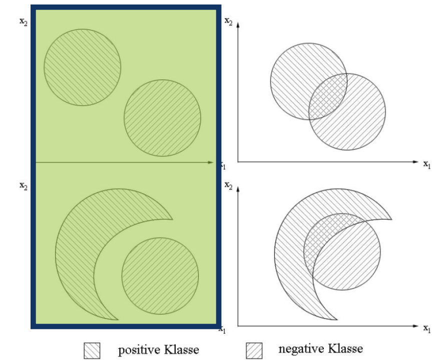
simple separable example



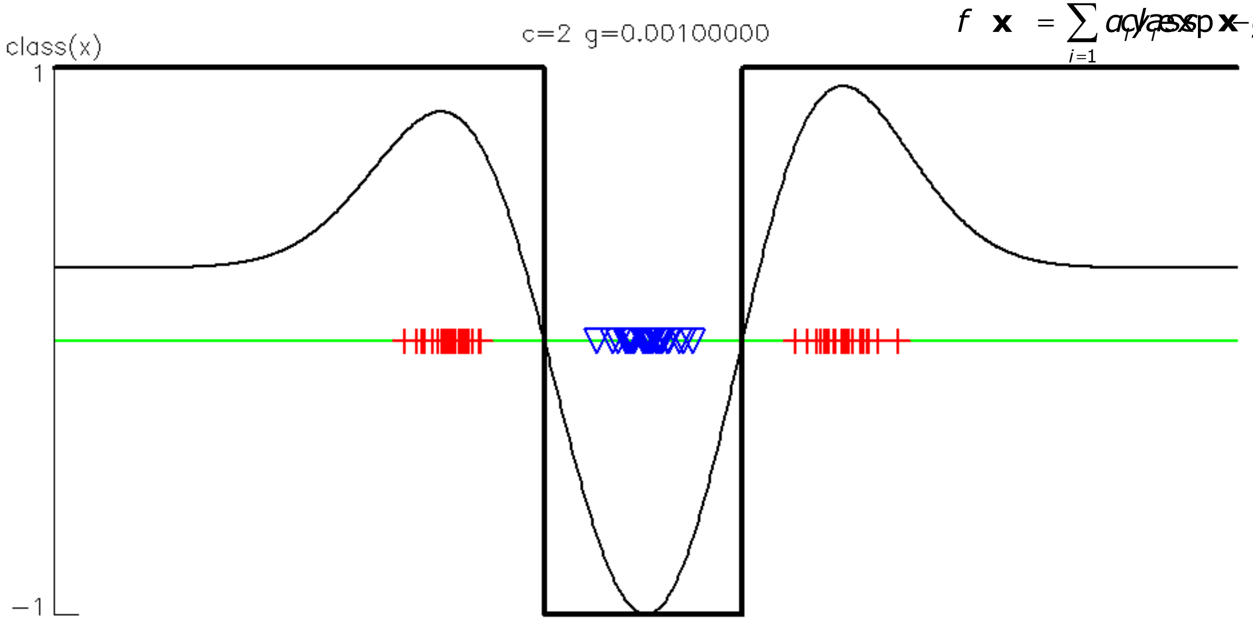
2D example - separable, linear (Burges1998)



2D example - separable, non-linear (www.mblondel.org)

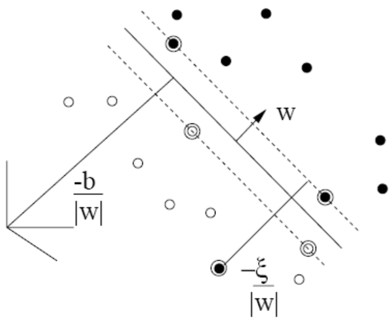


$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \exp\left(-g\left|\mathbf{x}-\mathbf{x}_i\right|^2\right) + b \qquad class = sign\, f(\mathbf{x})$$

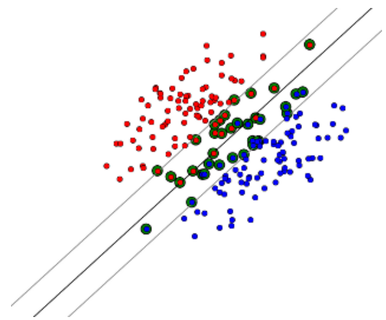


1D example - separable, non-linear

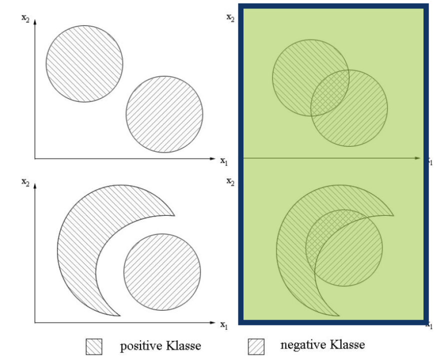
simple non-separable example



2D example - non-separable, linear (Burges1998)

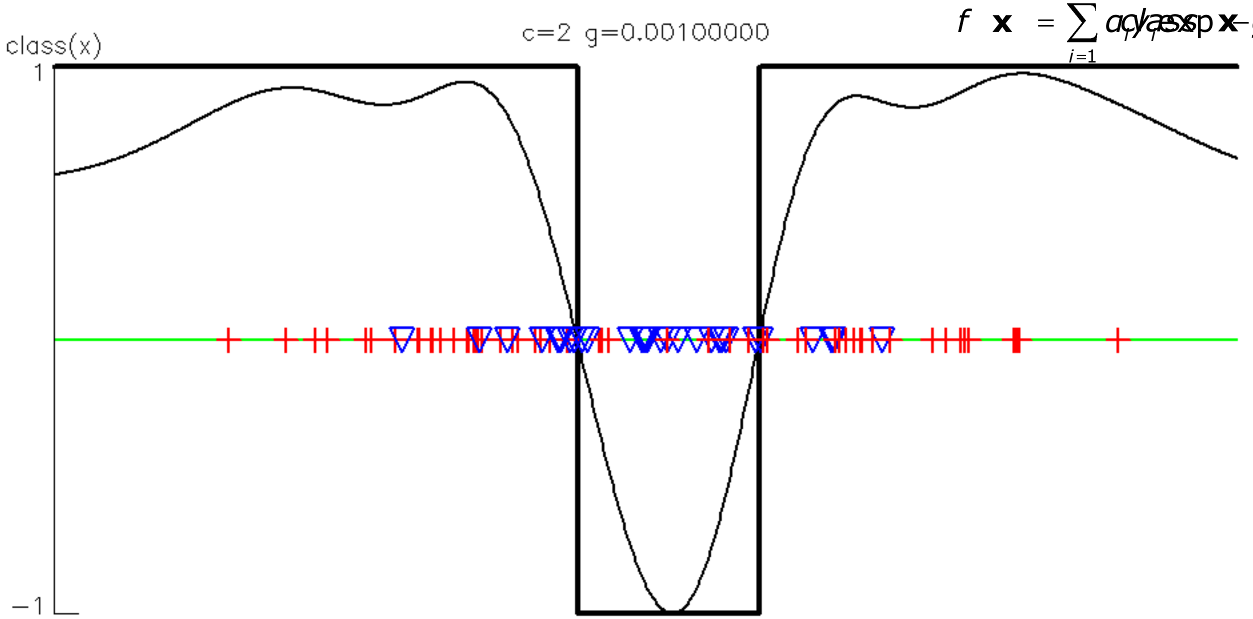


2D example - non-separable, linear (www.mblondel.org)



$$f(\mathbf{x}) = \sum_{i=1}^{l} \alpha_i y_i \exp\left(-g\left|\mathbf{x}-\mathbf{x}_i\right|^2\right) + b$$

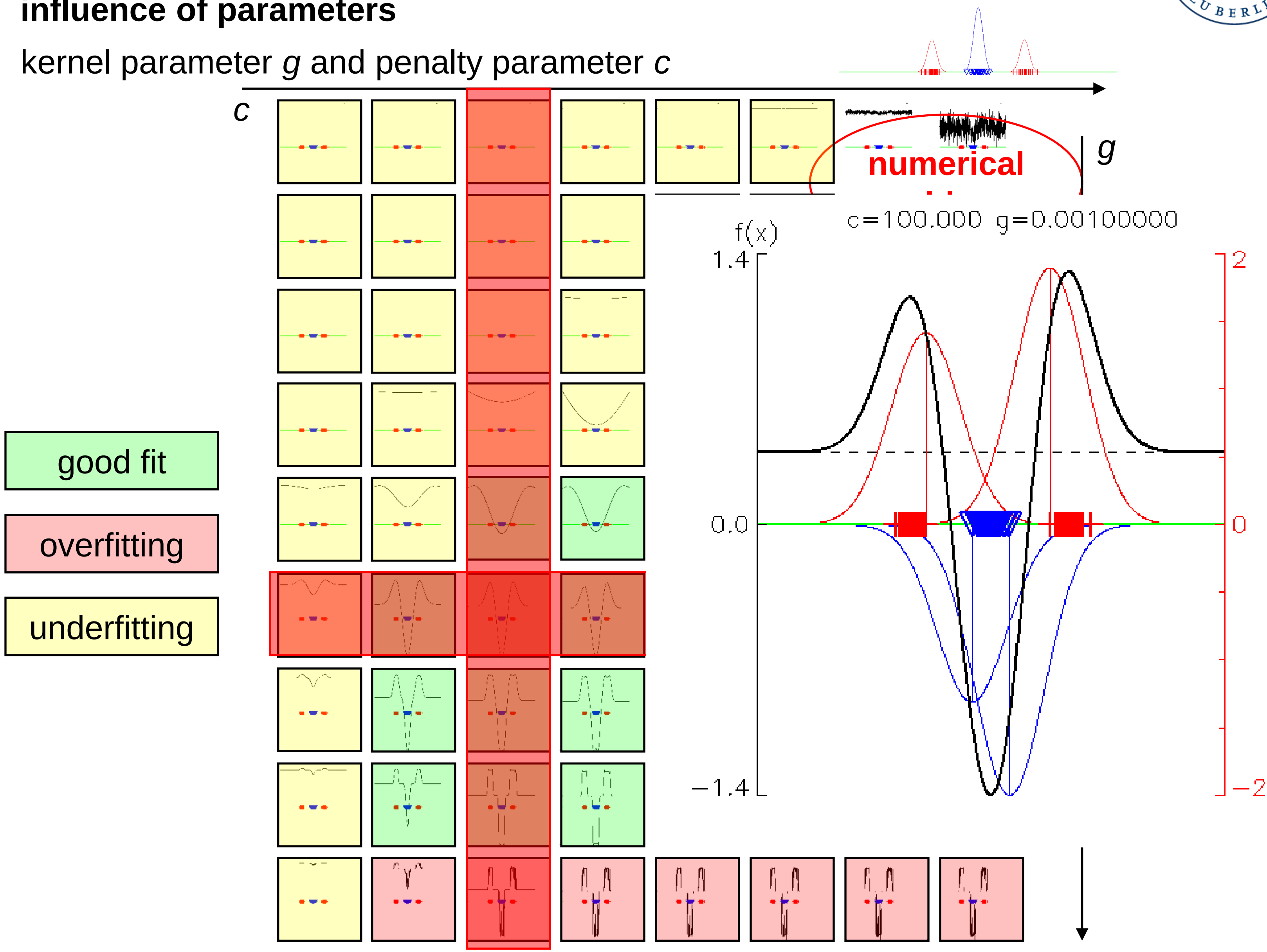
$$class = sign\, f(\mathbf{x})$$



1D example - non-separable, non-linear

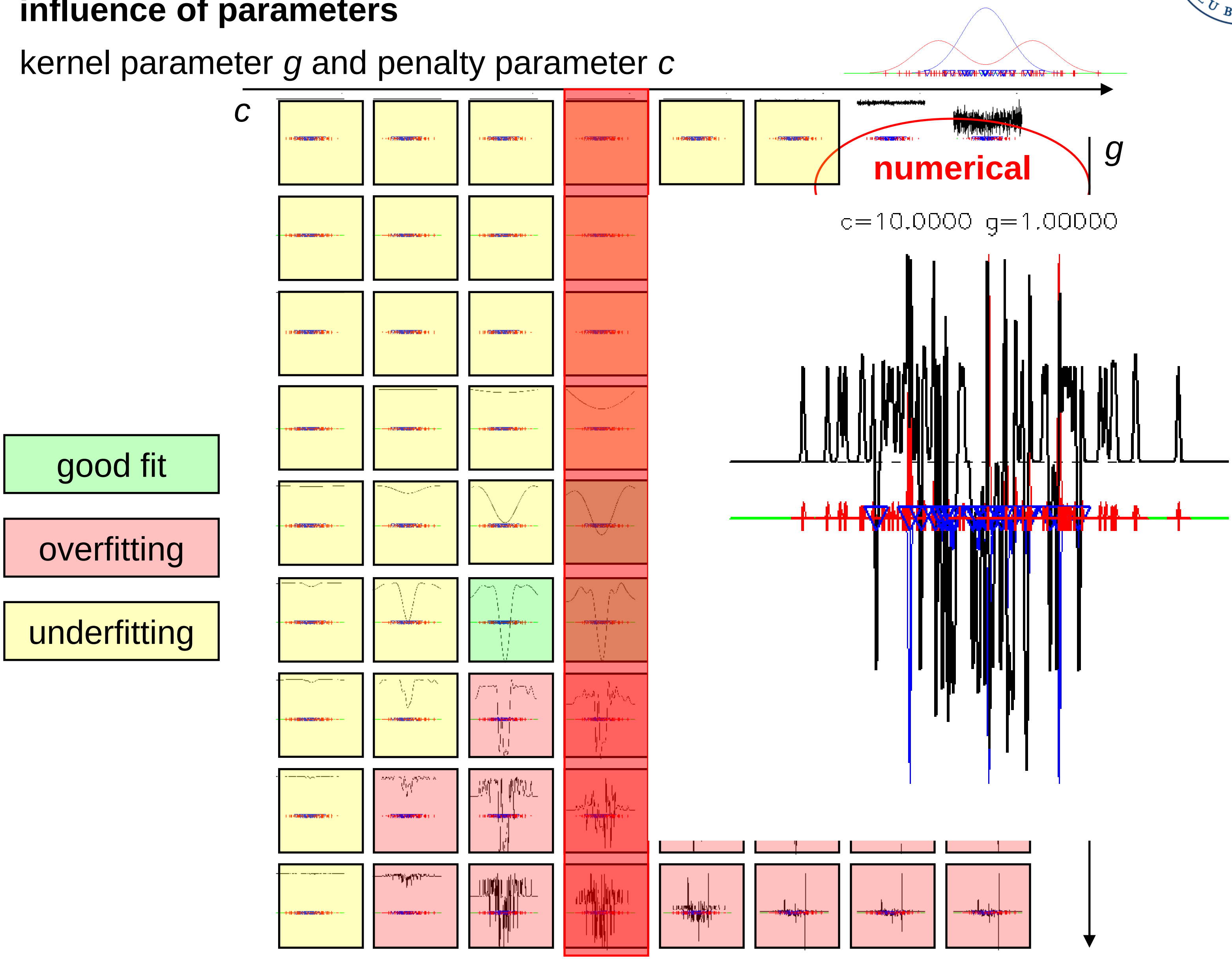
influence of parameters

kernel parameter *g* and penalty parameter *c*



influence of parameters

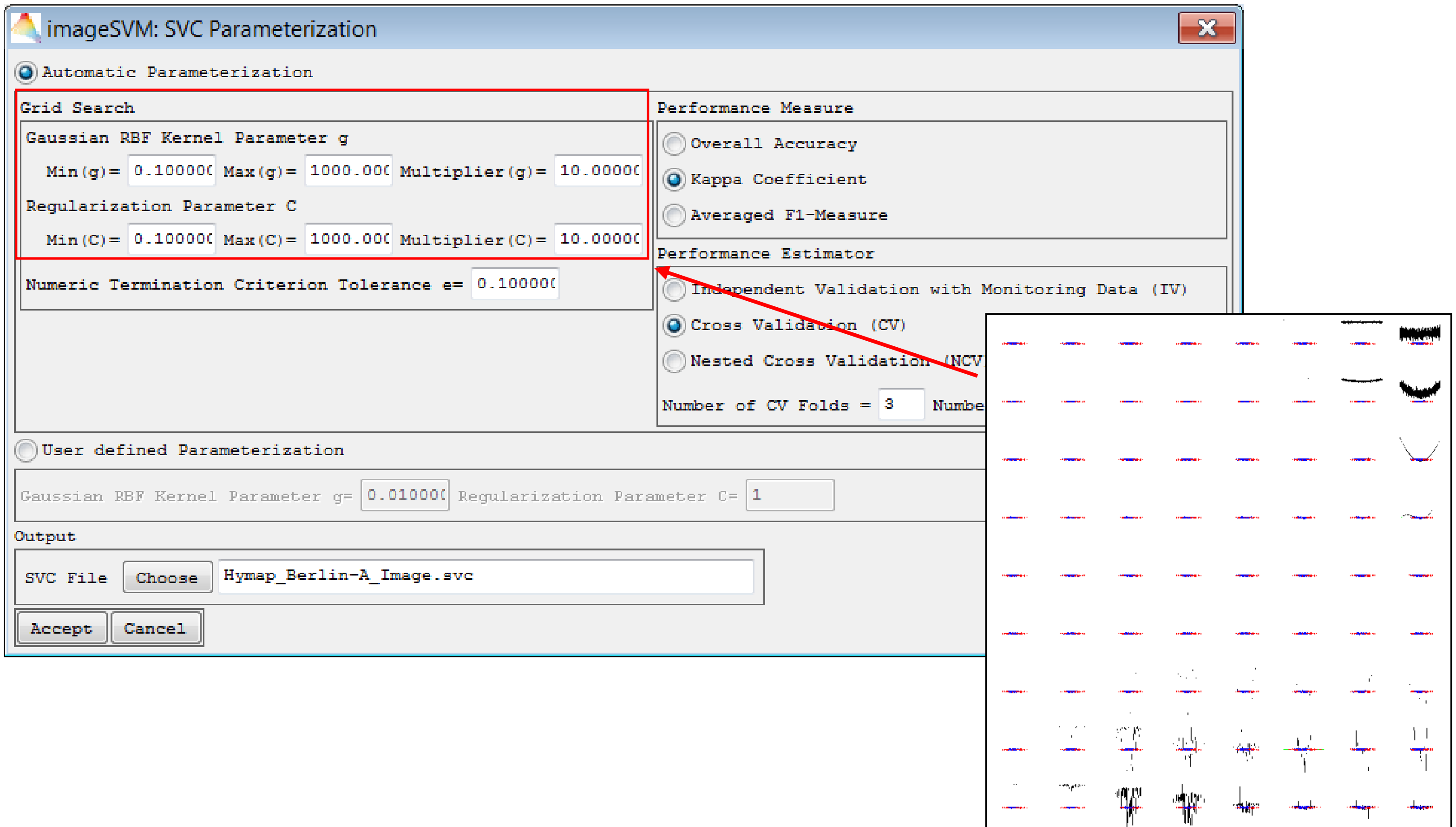
kernel parameter *g* and penalty parameter *c*



imageSVM inside EnMAP-Box software (remote sensing software)

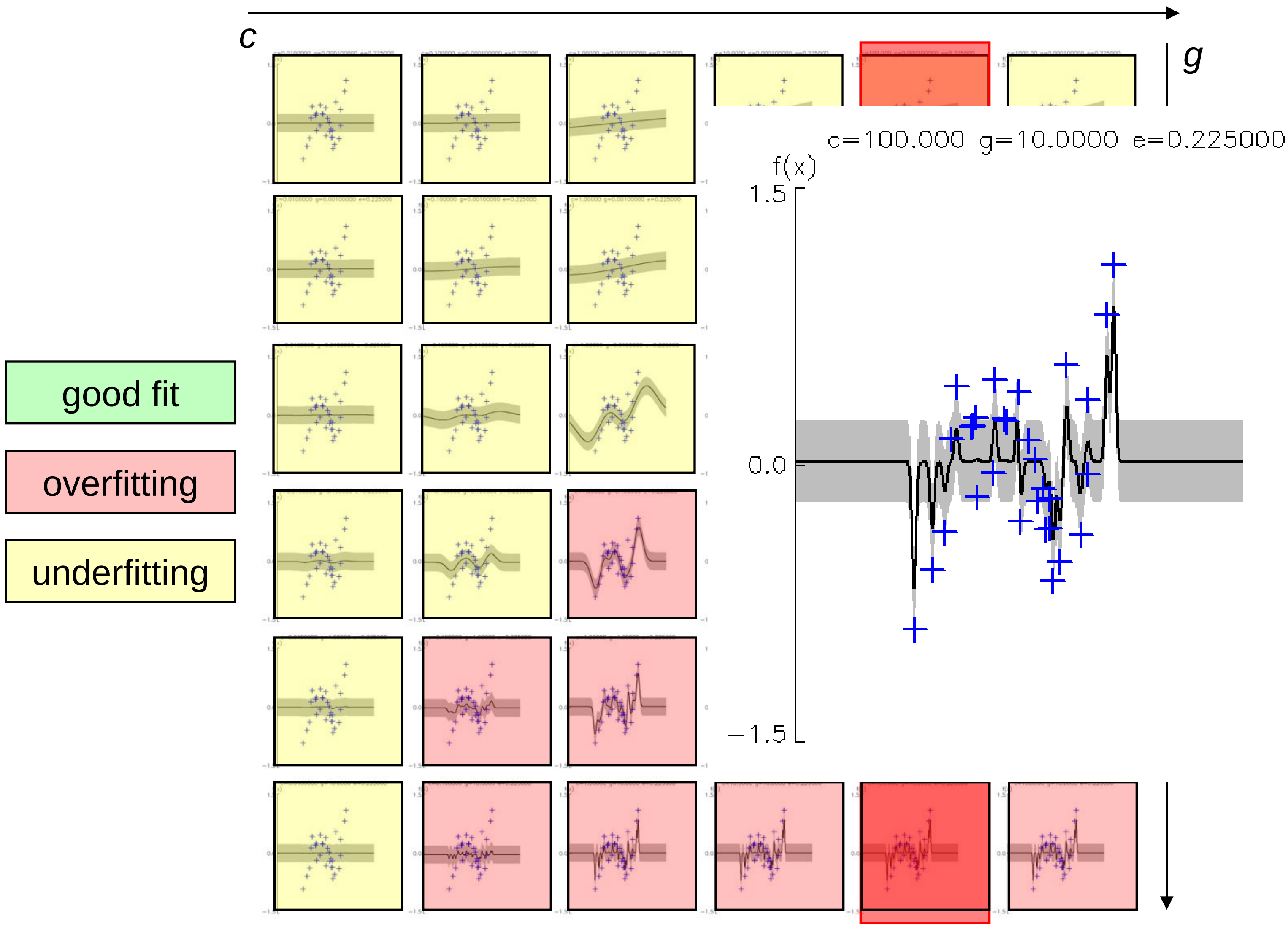
A SVM implementation for classification and regression ***imageSVM*** is freely available inside ***EnMAP-Box*** software (contact *andreas.rabe@geo.hu-berlin.de*).

Suitable parameters are estimated via grid search and cross-validation.



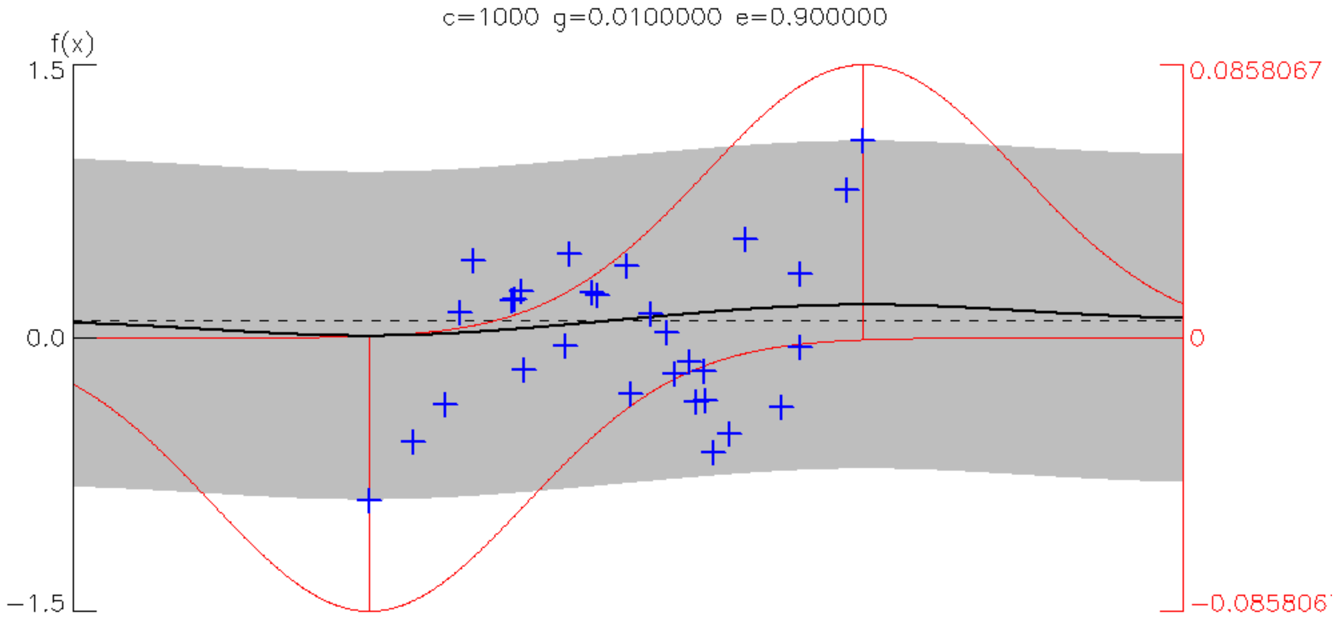
outlook - SVM regression

SVM regression - kernel parameter *g* and penalty parameter *c*



outlook - SVM regression

SVM regression - epsilon-loss function



Thank you very much for your attention.

Any questions?

References

Burges, C. J. C. (1998). "A Tutorial on Support Vector Machines for Pattern Recognition." Data Mining and Knowledge Discovery **2**(2): 121-167.

Chang, C.-C. and C.-J. Lin (2001). LIBSVM: a Library for Support Vector Machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Vapnik, V. (1999). The Nature of Statistical Learning Theory, Springer-Verlag.