EPFL

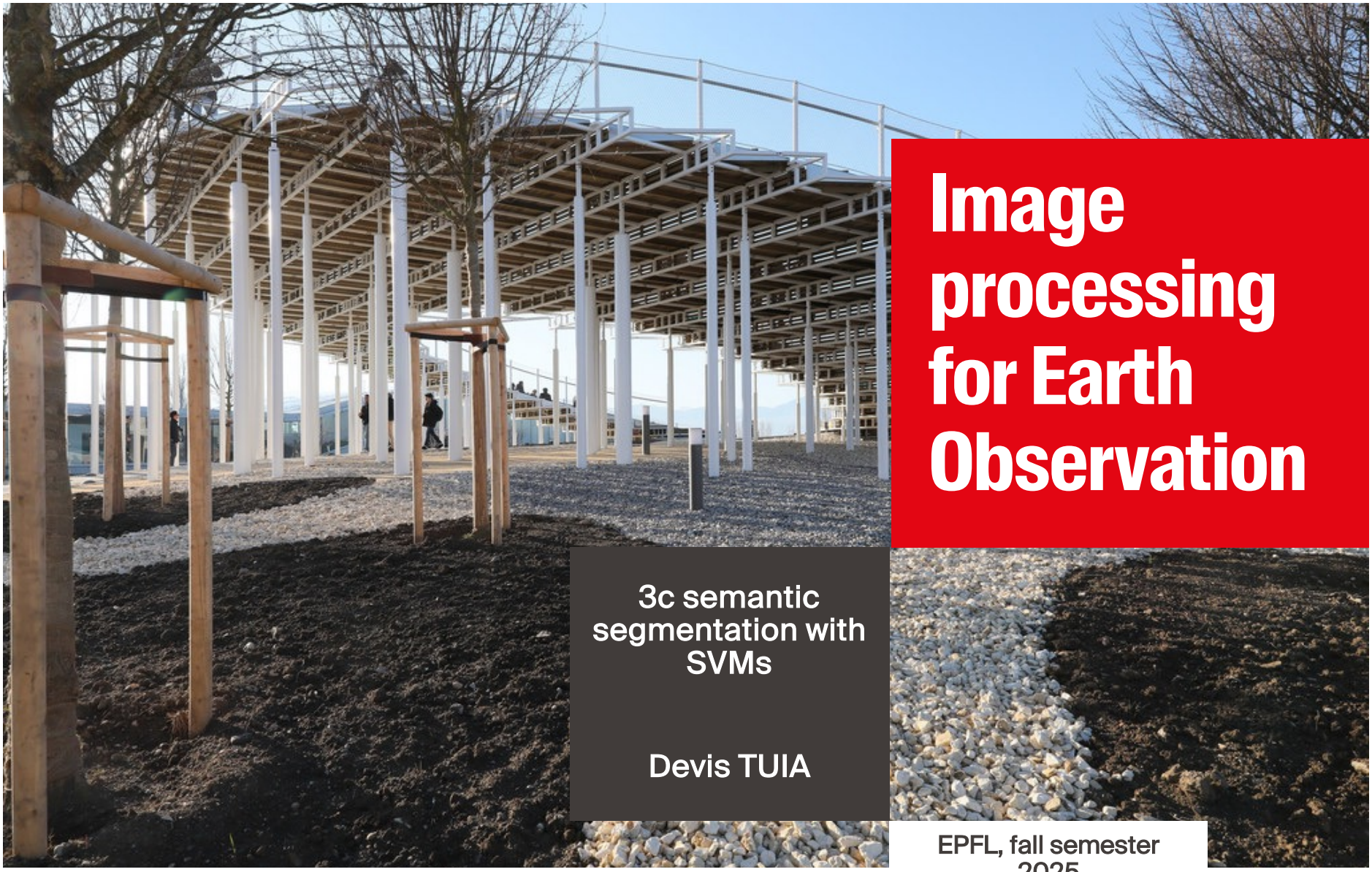


Image processing for Earth Observation

3c semantic segmentation with SVMs

Devis TUIA

EPFL, fall semester 2025

École polytechnique fédérale de Lausanne

Content (6 weeks)

- W1 General concepts of image classification / segmentation
  Traditional supervised classification methods (RF)
- W2 Traditional supervised classification methods (SVM)
  Best practices

- W3 Elements of neural networks
- W4 Convolutional neural networks
- W5 Convolutional neural networks for semantic segmentation
- W6 Sequence modeling, change detection

Support vector machines (SVM) is originally a linear classifier

- SVM builds a linear function between two classes using similarity between samples only.
- It finds a separating line of type $y = \langle w,x \rangle + b$
- If $y \geq 0$, response is +1, otherwise -1.
- All the training samples must be classified correctly
- Many linear equations satisfy this criterion

D. Tuia. ECEO

Support vector machines (SVM) is originally a linear classifier

The decision is taken by the sign wrt the current decision function

$$y_i \begin{cases} 1 & \text{if } \langle \mathbf{w}, \mathbf{x}_i \rangle - b \geq 0 \\ -1 & \text{if } \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq 0 \end{cases}$$

In other words, when testing a new point, if its coordinate is on one side of the decision function it will be of class 1, if on the other -1

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) > 0$$

Example (from James et al., an Introduction to statistical learning)

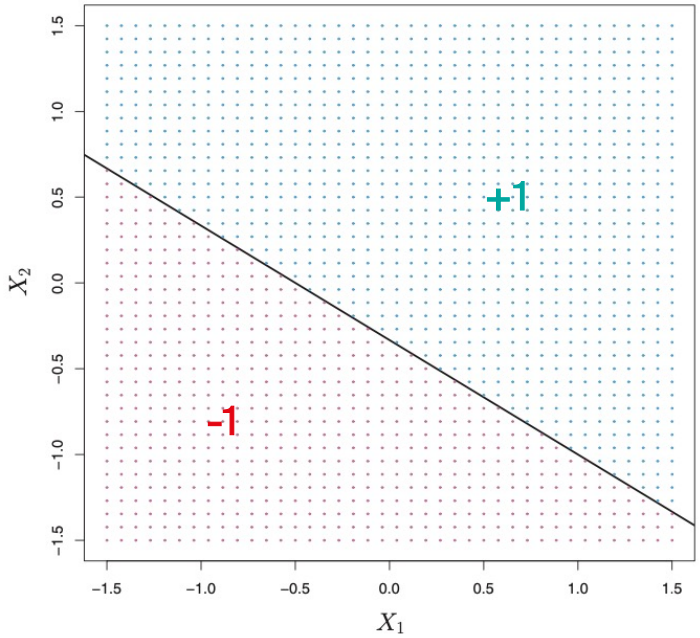
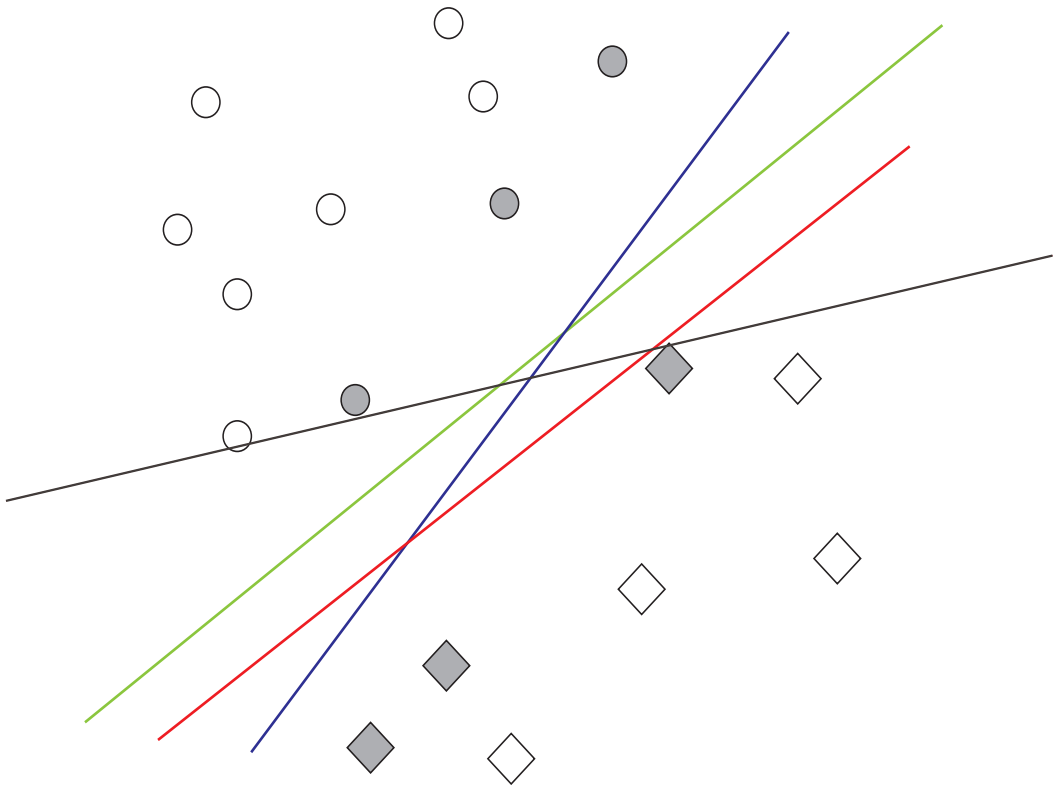


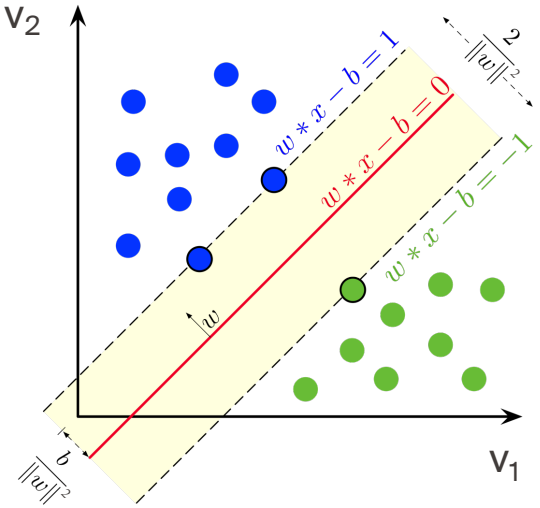
FIGURE 9.1. *The hyperplane* $1 + 2X_1 + 3X_2 = 0$ *is shown. The blue region is the set of points for which* $1 + 2X_1 + 3X_2 > 0$*, and the purple region is the set of points for which* $1 + 2X_1 + 3X_2 < 0$.

Which solution is the best?



Which solution is the best?

- The one that has the smallest risk of making errors on new data!
- If we believe our training data
  - We can stay in the middle
  - Equally far from both classes
- The plane that achieves that is the SVM plane
- Let's call this equally spaced, symmetrical plane, the margin



(why did I say plane?)

- In 2D, the separation between two classes is a line
- In 3D, the separation is a plane
- In more D, the separation is a hyperplane

→ So actually, in a D-dimensional space, we are looking for a hyperplane of dimension D-1

Rivisiting our initial equations

Since the margin is symmetric on both sides of the plane, we can re-write:

$$y_i \begin{cases} 1 & \text{if } \langle \mathbf{w}, \mathbf{x}_i \rangle - b \geq 1 \\ -1 & \text{if } \langle \mathbf{w}, \mathbf{x}_i \rangle - b \leq 1 \end{cases}$$

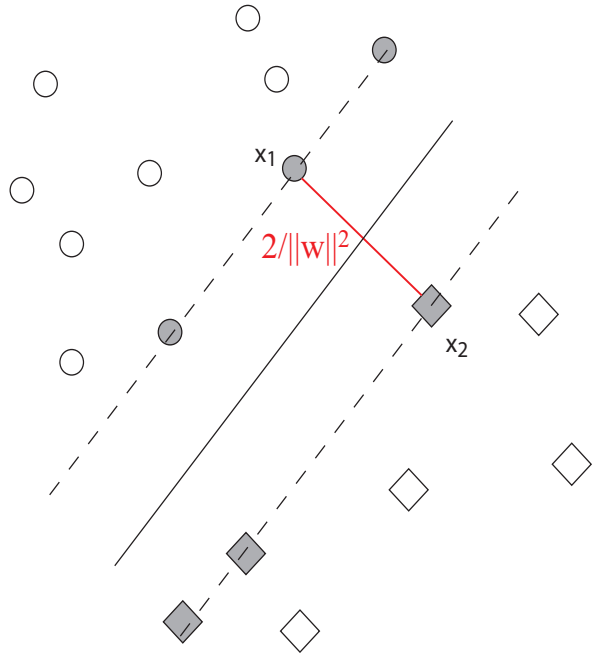
As before, this leads to the decision function:

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1$$

SVM

- We want the margin to be as wide as possible
- The aim of SVM is to find the hyperplane which is the further away from the closest samples of both classes.

  = maximize the margin width
- The plane has width $2/||w||^2$



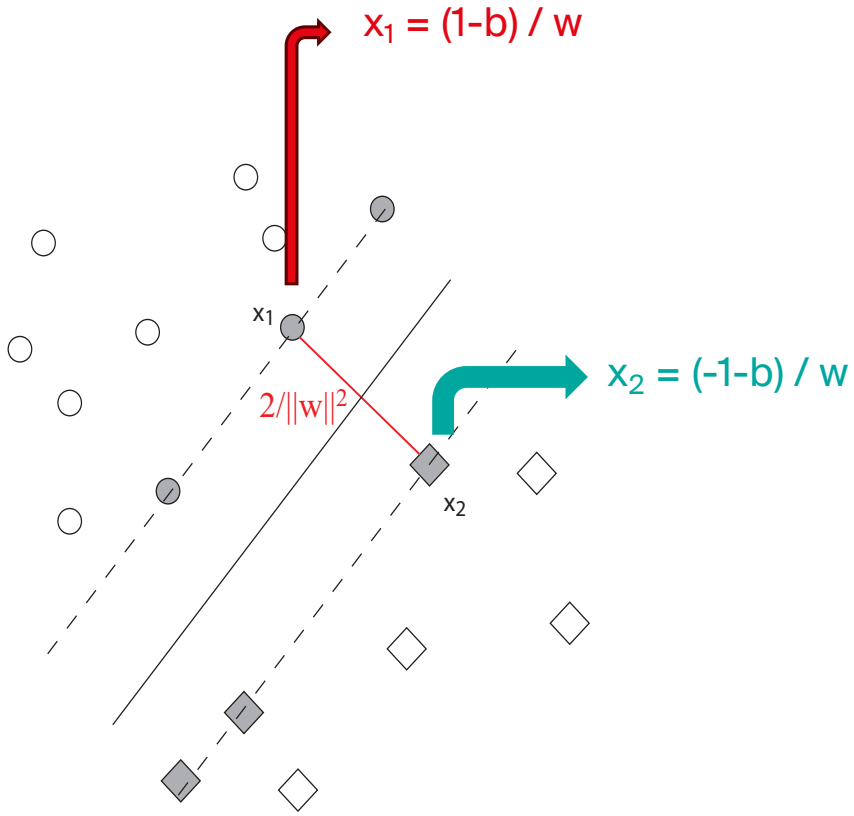
D. Tuia. ECEO

Why 2 / ||w||² ?

- Take two points on the margin : x_1 and x_2
- They are at distance
  - $\langle w, x_1 \rangle + b = 1$
  - $\langle w, x_2 \rangle + b = -1$

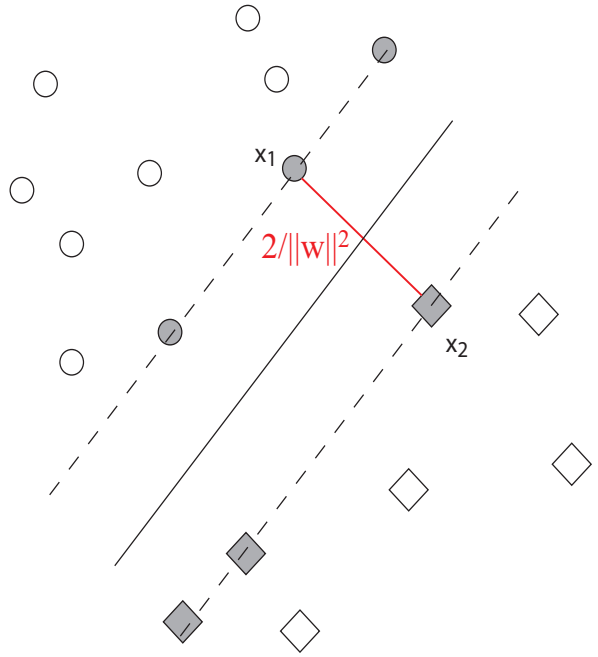
  from the decision plane
- Their respective distance is

$$d = \|\mathbf{x}_1 - \mathbf{x}_2\| = \left\| \frac{1 - b + b + 1}{\mathbf{w}} \right\| = \frac{2}{\|\mathbf{w}\|}$$



SVM

- The plane has width $2/||w||^2$
- These samples (in grey) are called the support vectors.
- They are the important ones, since they alone define the plane!
- All the other training samples do not matter to define the plane

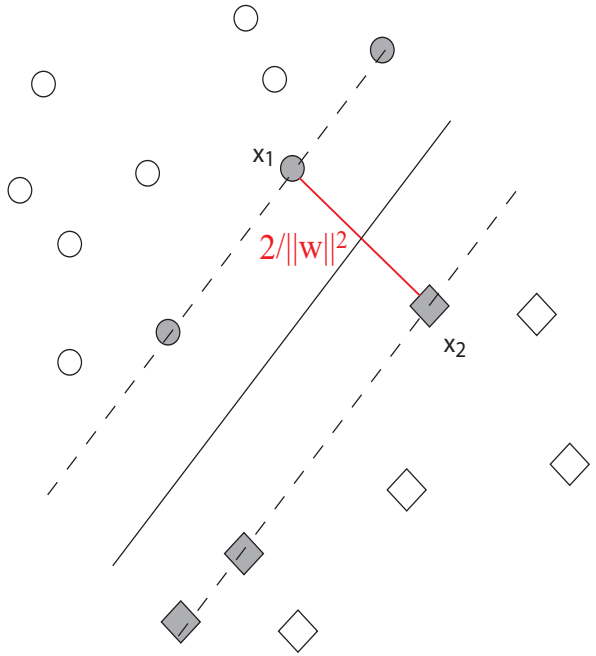


Optimize the SVM

- We want to maximize the width of the plane, $2/||w||^2$
- We do so by minimizing its inverse

$$\min_{\mathbf{w}} ||\mathbf{w}||^2$$

$$\text{s.t. } y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1, \forall i$$

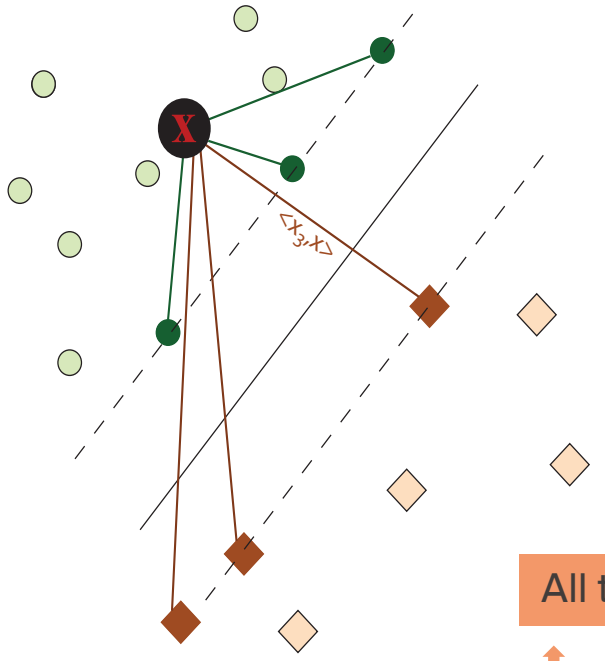


Decision function

- After optimization, each training samples has a Lagrangian multiplier α_i
- This parameter is nonzero only for the support vectors

- To predict a new sample x, the decision function is based on the distance between
  - The sample x and
  - All the training samples

- Similarity is assessed by a dot product <.,.>



$$y^* = sign\left(\sum_{i \in SV} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle \right)$$

All training samples

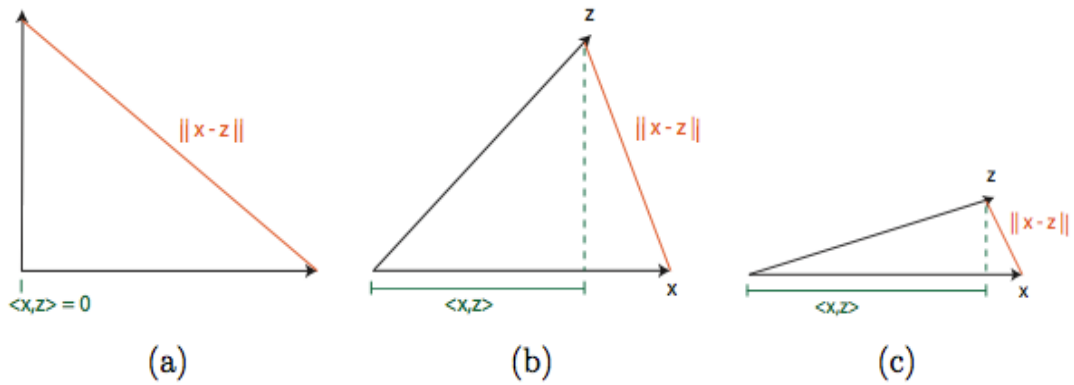
Nonzero only
For suppor vectors

New sample x

D. Tuia. ECEO

Why dot products?

- SVM decision function is based on dot products as a measure of similarity
- Dot products are a measure of similarity in the input space (the bands)



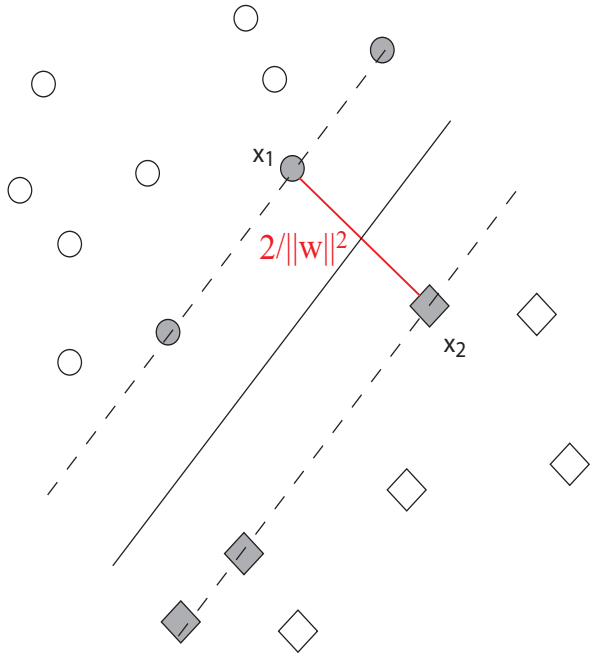
D. Tuia. ECEO

Tolerance to errors?

- Remember? We want to maximize the width of the plane, $2/||w||^2$
- We do so by minimizing its inverse

$$\min_{\mathbf{w}} ||\mathbf{w}||^2$$

- But we assume perfect classification in training!

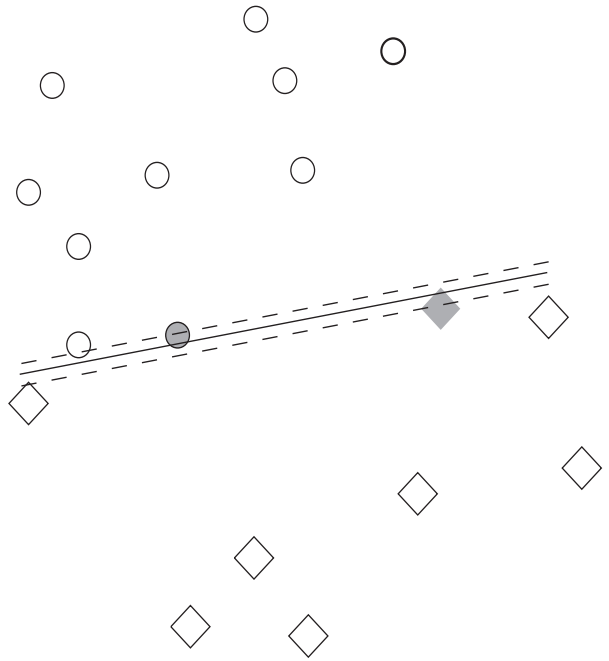


D. Tuia. ECEO

Tolerance to errors?

- One single missclassification can make the classifier overfit!

$$\min_{\mathbf{w}} ||\mathbf{w}||^2$$

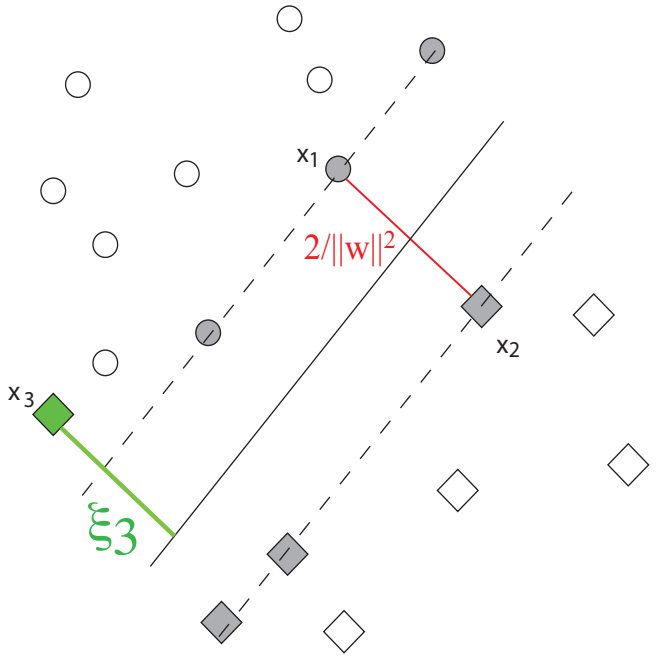


Tolerance to errors?

- One single missclassification can make the classifier overfit!

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + C \sum_i \xi_i$$

- So we relax the “perfect classification” rule and allow some mistakes

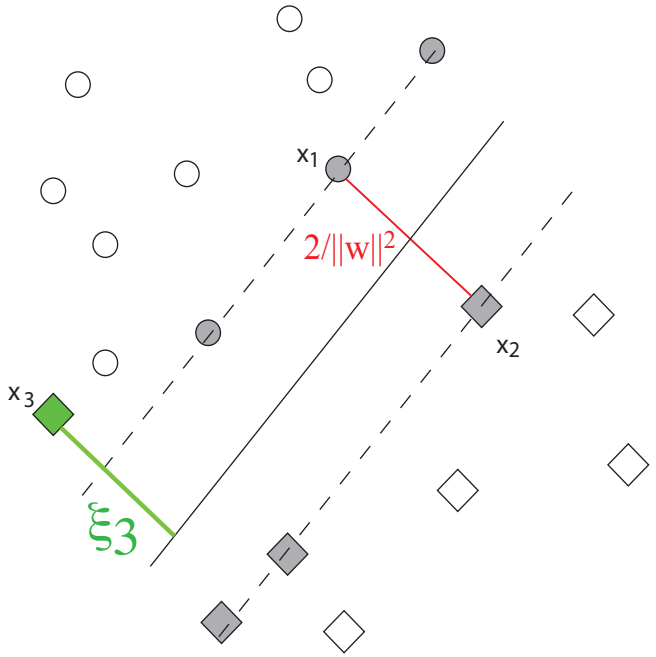


D. Tuia. ECEO

Tolerance to errors?

- One single missclassification can make the classifier overfit!

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + C \sum_i \xi_i$$



- Every time we make a mistake, we penalize by ξ_i (= the distance of the sample from the classification plane)
- If no mistake, $\xi_i = 0$
- Meaning : the bigger the mistake, the more we penalize

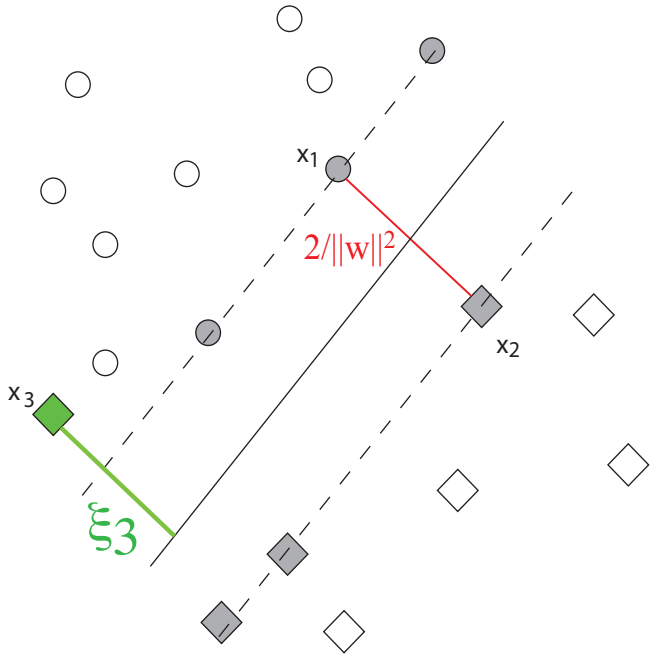
D. Tuia. ECEO

Tolerance to errors?

- One single missclassification can make the classifier overfit!

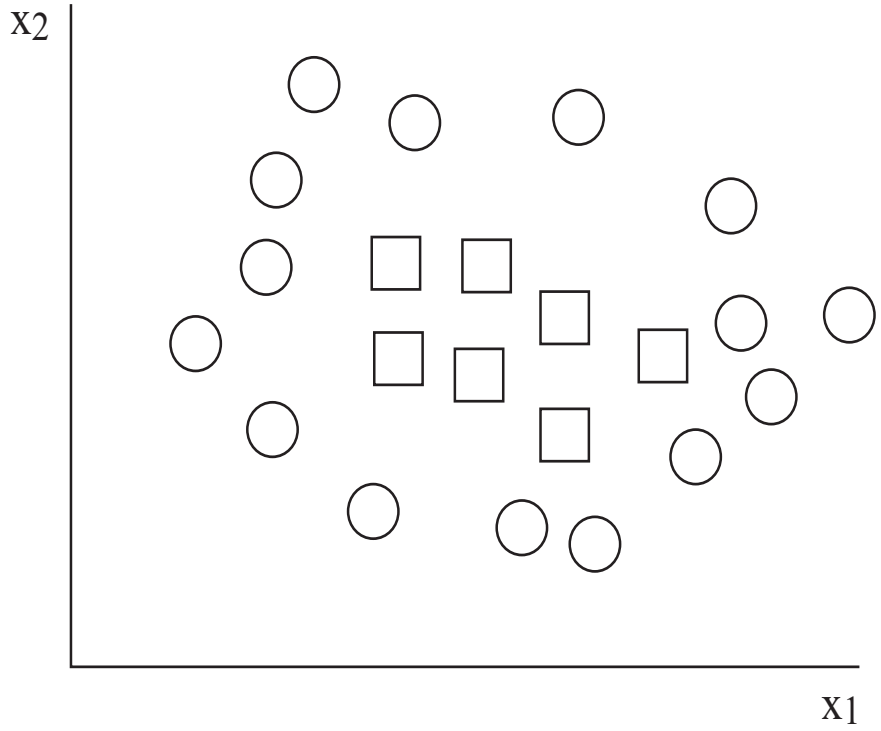
$$\min_{\mathbf{w}} ||\mathbf{w}||^2 + C \sum_i \xi_i$$

- We want to minimize this functional, so each error makes a solution less appealing.
- The bigger C, the least we allow mistakes.



D. Tuia. ECEO

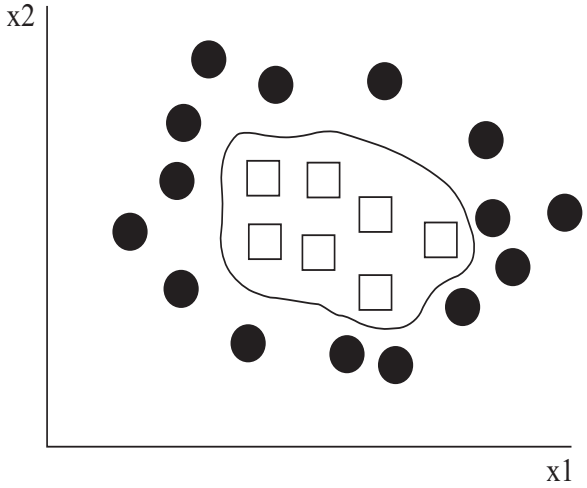
Need nonlinear response?



D. Tuia. ECEO

2 solutions

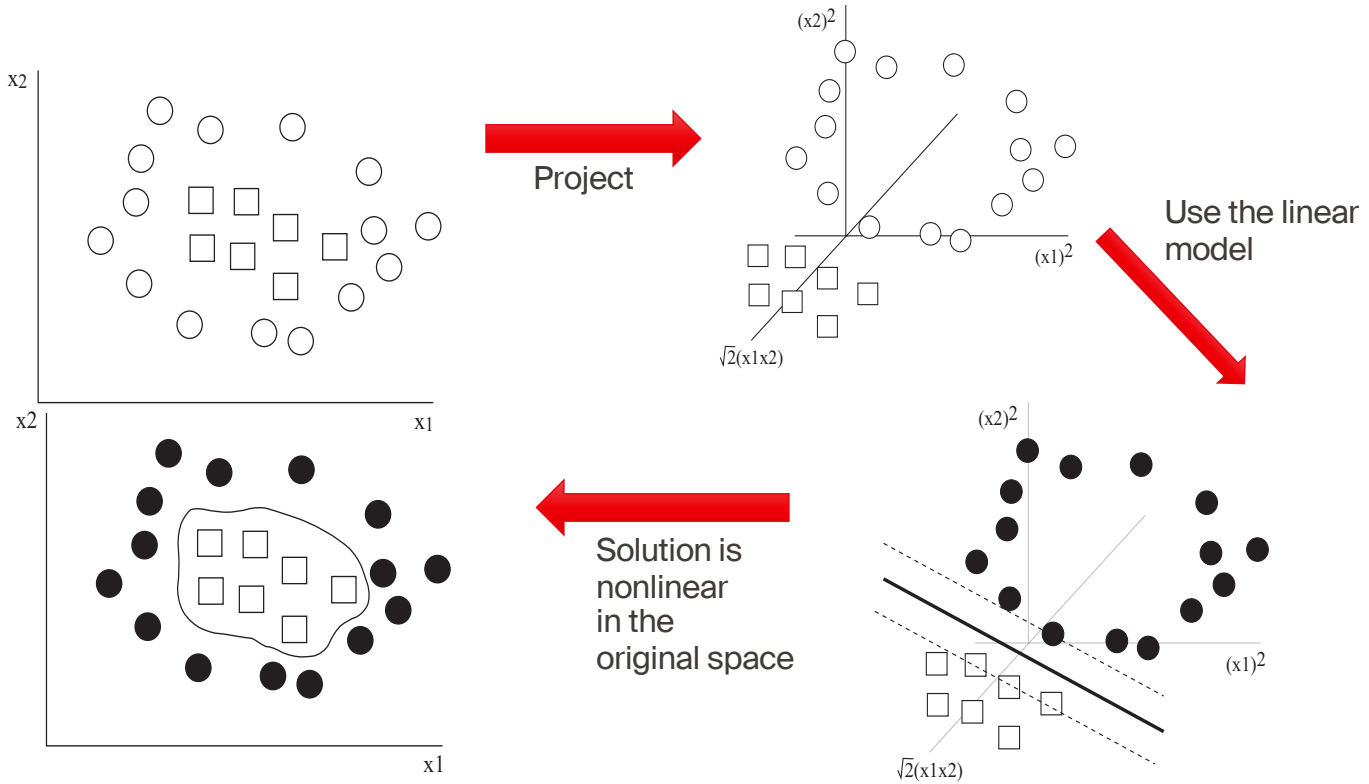
- Build a complex nonlinear model



- Project the data, where linear separation is achievable!

D. Tuia. ECEO

In higher dimensional spaces, linear separation is more likely to be possible (Cover theorem, 1965)



D. Tuia. ECEO

Cover theorem in Mario land (super paper Mario, Wii)

Need nonlinear response?

- The trick is to use the right projection
- BUT
- The expression of the projecting function f(x) can become quite complicated (more than the nonlinear complex model)!
- There are infinitely many projections, impossible to go by trial and error!

D. Tuia. ECEO

Need nonlinear response?

- Solution:
  - Use samples in the original space
  - Get the projected, nonlinear solution
- HOW?

D. Tuia. ECEO

Kernels

- There are mathematical functions that use input samples, but correspond to a dot product in projected higher dimensional spaces.
- They are called kernels.

$$K : x \rightarrow \phi(x)$$

$$K(\mathbf{x}_1, \mathbf{x}_2) = \langle \phi(\mathbf{x}_1), \phi(\mathbf{x}_2) \rangle$$

- In higher dimensional spaces, linear separation is more likely to be possible (Cover theorem, 1965)

D. Tuia. ECEO

Kernels

- The beauty of it, is that we never need to compute the mappings f(x)!
- The single samples always appear in dot products (their respective distance), never as such!

$$y^* = sign\Big(\sum_{i \in SV} \alpha_i y_i \langle \mathbf{x}_i, \mathbf{x} \rangle\Big)$$

- Ex: squared polynomial kernel on a 2D space [x_{d1},x_{d2}]

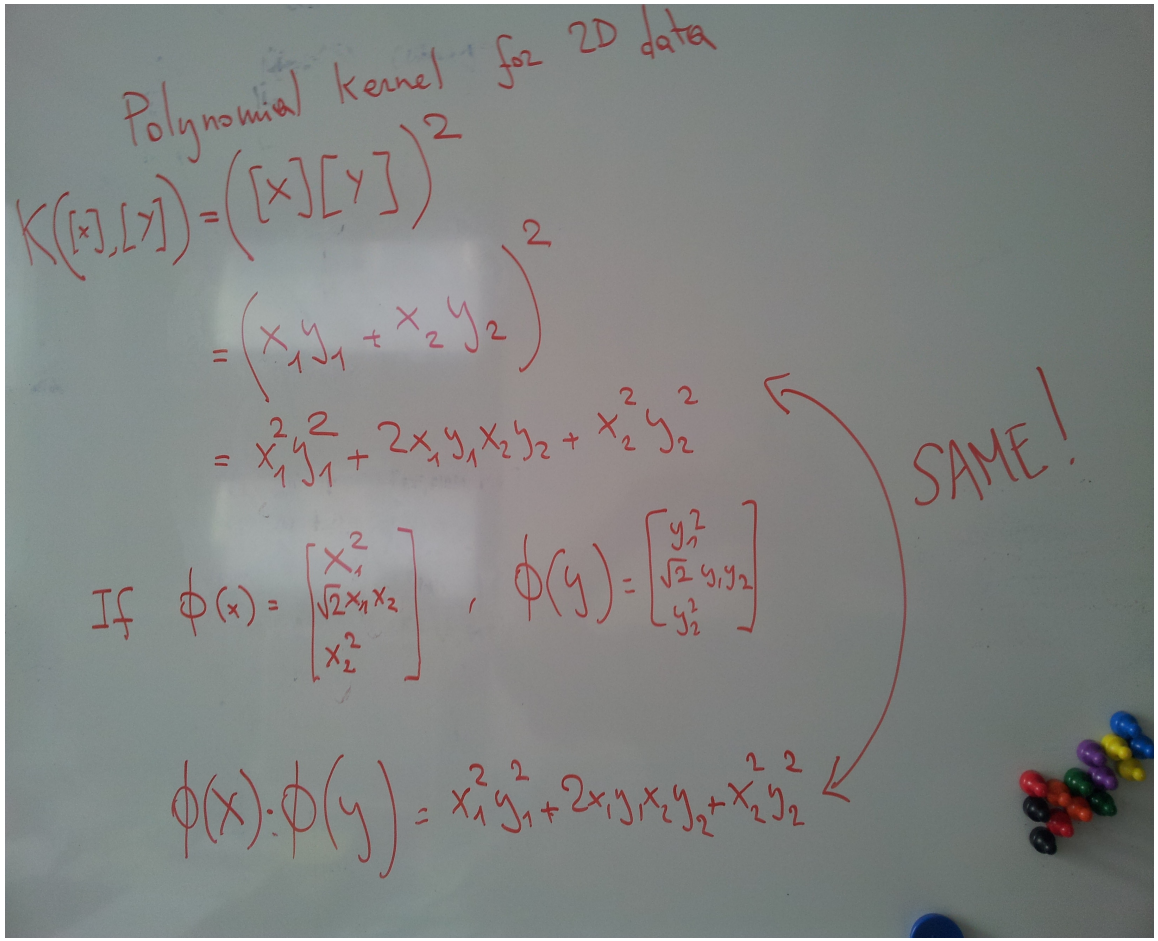
$$K(\mathbf{x}_1, \mathbf{x}_2) = \left(\langle \mathbf{x}_1, \mathbf{x}_2 \rangle\right)^2$$

corresponds to a dot product between the two samples in the 3D space with coordinates [$(x_{d1})^2$, sqrt(2)$x_{d1}x_{d2}$, $(x_{d2})^2$]

D. Tuia. ECEO

Proof

- Careful. With respect to the notation of the last slide:
  - $x_1 = [x]$ (the first sample)
  - $x_2 = [y]$ (the 2nd sample)
  - $x_{d1} = x_1$ (first dimension of x)
  - $x_{d2} = x_2$ (2nd dimension of x)
  - $y_{d1} = y_1$ (first dimension of y)
  - $y_{d2} = y_2$ (2nd dimension of x)
  - $\phi(x).\phi(y) = <\phi(x).\phi(y)>$ (the dot product)



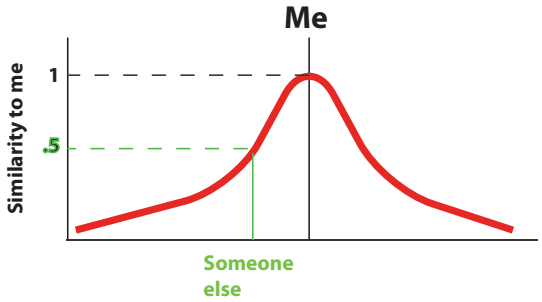
D. Tuia. ECEO

Similarity (recall)

- It is a function that scores high if two objects look alike and low if they don't

vs.

- In the Gaussian case:

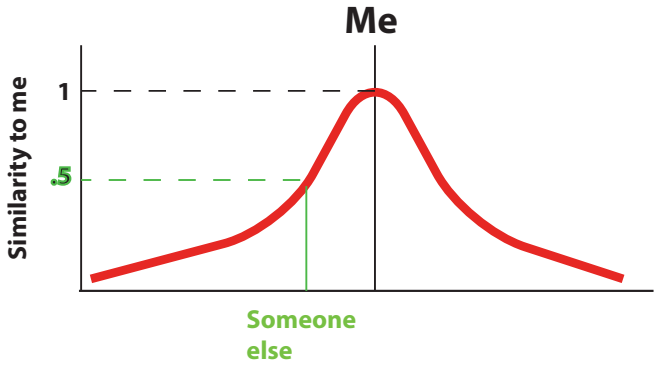


D. Tuia. ECEO

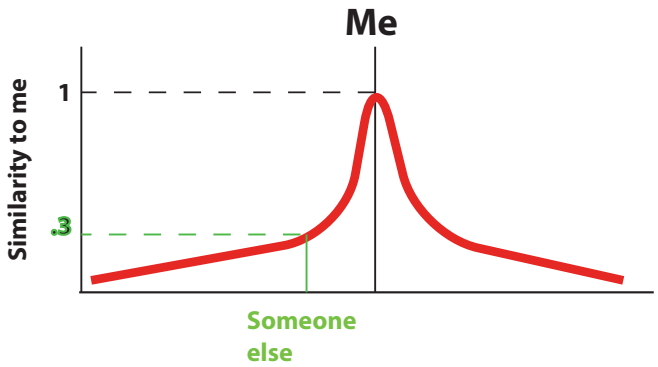
RBF (or Gaussian) kernel

- The g parameter will decide how much similarity decreases with feature distance.

$$K(me, se) = \exp(-2\gamma^2(me - se)^2)$$



Large bandwidth (large gamma)

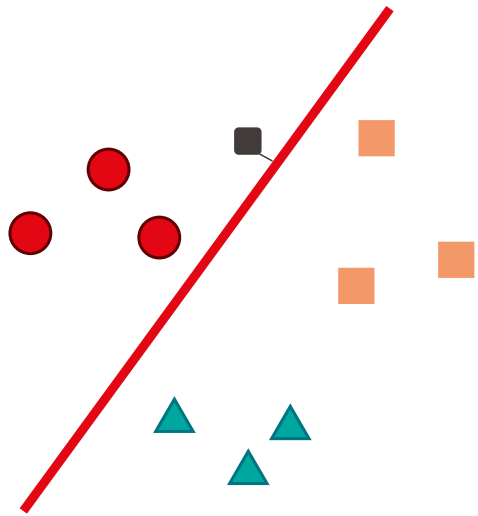


Small bandwidth (small gamma)

D. Tuia. ECEO

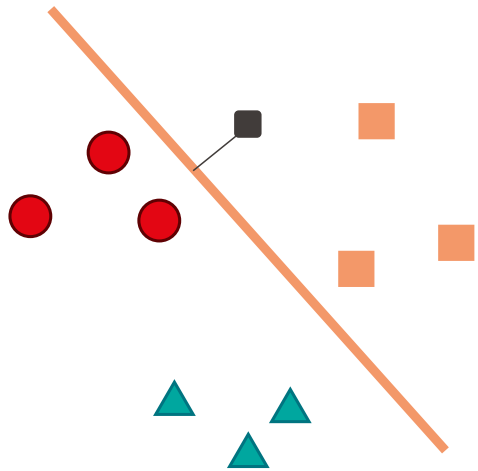
Ho to achieve multiclass response?

- SVM is a binary classifier
- We have 2 ways (all softs implement one or another):
  - OAA: one against all



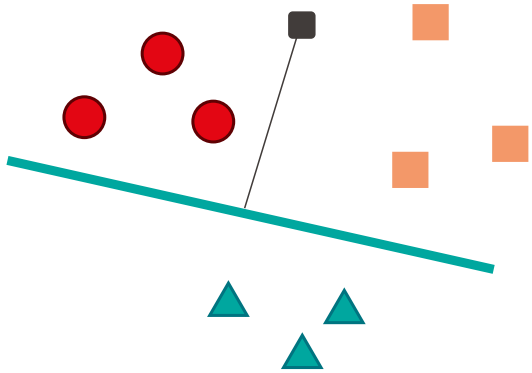
Ho to achieve multiclass response?

- SVM is a binary classifier
- We have 2 ways (all softs implement one or another):
  - OAA: one against all



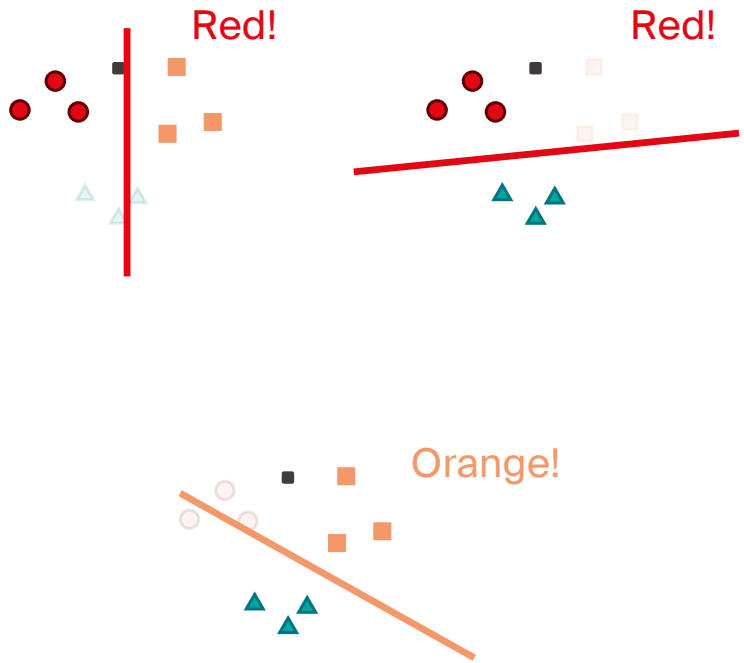
How to achieve multiclass response?

- SVM is a binary classifier
- We have 2 ways (all softs implement one or another):
  - OAA: one against all



Ho to achieve multiclass response?

- SVM is a binary classifier
- We have 2 ways (all softs implement one or another):
  - OAA: one against all
  - OAO: one against one



Other cool things about kernels: they are commutative (you can sum and multiply them)

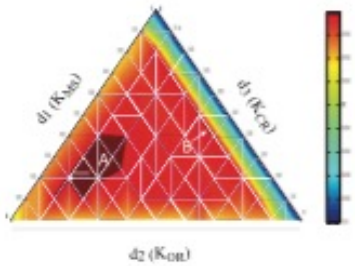
- color : very discriminative
- height : discriminative
- width : not discriminative

$$K = \underbrace{0.6}_{d_C} * K_C + \underbrace{0.3}_{d_H} * K_H + \underbrace{0.1}_{d_W} * K_W$$

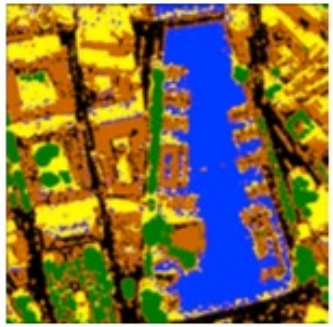
Combining spectral and spatial features

D. Tuia, F. Ratle, A. Pozdnoukhov, and G. Camps-Valls. Multi-source composite kernels for urban image classification. *IEEE Geosci. Remote Sens. Lett.*, 7(1):88–92, 2010.

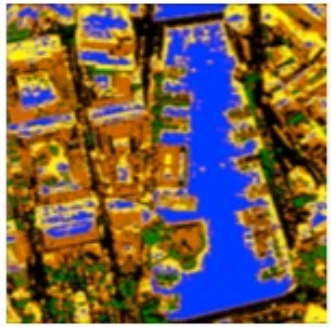
Image



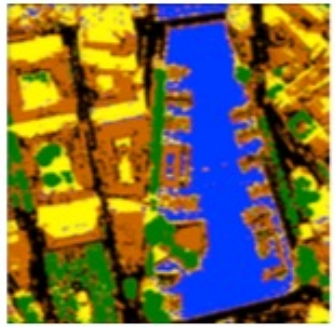
Kappa surface



Spectral bands K_{MS}



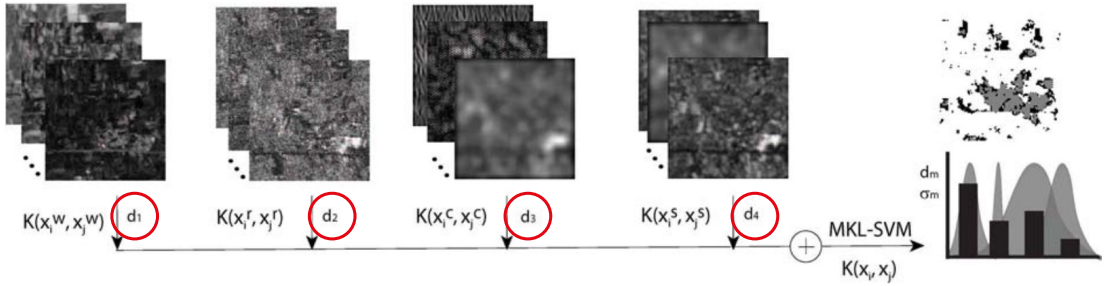
Opening K_{Or}



$d_1 K_{MS} + d_2 K_{Or} + d_3 K_{Cr}$

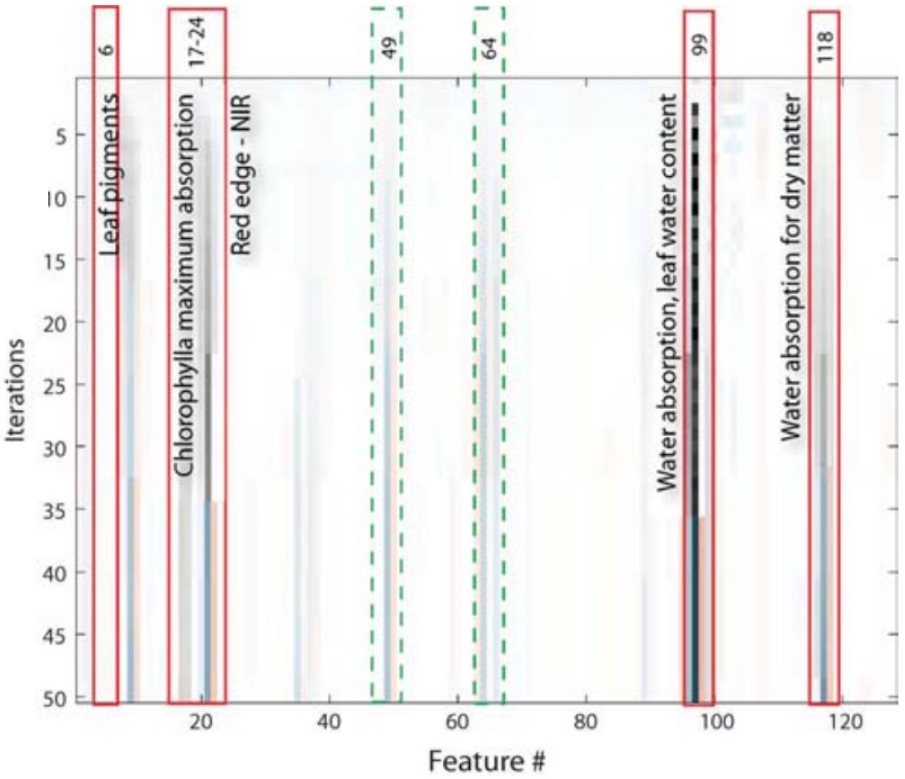
Learning more complex combinations

D. Tuia, G. Camps-Valls, G. Matasci, and M. Kanevski. Learning relevant image features with multiple kernel classification. *IEEE Trans. Geosci. Remote Sens.*, 48(10):3780 – 3791, 2010.



Single features (or groups) kernel weights learned by optimization (gradient descent)

This allows you to discover which features are important!



Ranking features generated on the fly

D. Tuia, M. Volpi, M. dalla Mura, A. Rakotomamonjy, and R. Flamary. Automatic feature learning for spatio-spectral image classification with sparse SVM. *IEEE Trans. Geosci. Remote Sens.*, 52(10):6062–6074, 2014.

- Here we used a linear SVM
- Rank new features according to “how well they align with current errors of the model”
- If they don’t align, add!
- Then iterate

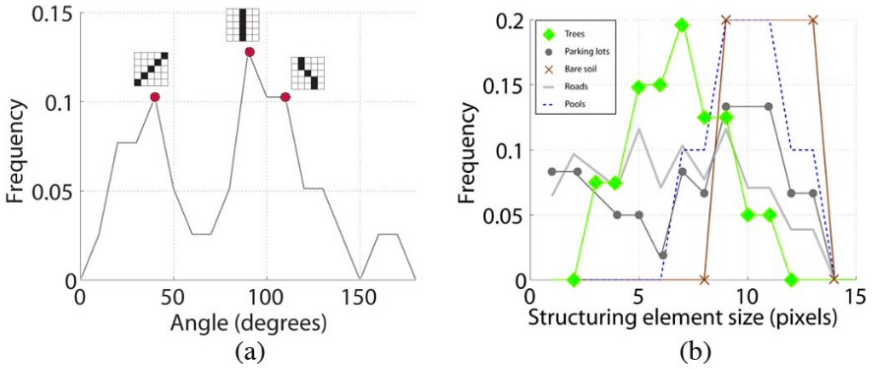


Fig. 5. (a) Orientation of linear structuring elements for the class “roads.” (b) Structuring element size within the morphological filters selected for five classes (for color legend, refer to Table IV).

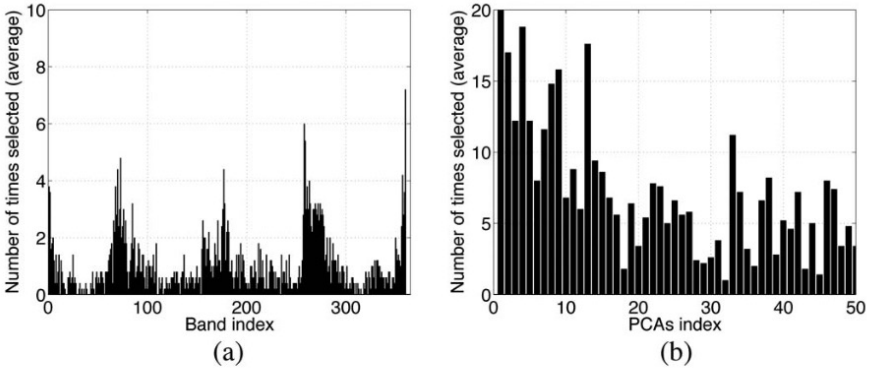


Fig. 6. Variables selected for filtering in one run of the (a) AS − Bands and (b) AS − PCA experiments, respectively, for the Indian Pines data set. The plots report the average of the bands selected by five runs of the algorithm with different initializations.

SVM

- Pros
  - Performances
  - Solid maths behind
  - Works well in high dimensionalities
  - Is sparse (only uses a subset of the data, the support vectors)

- Cons
  - More complex that the others
  - Free parameters to tune
  - Longer runtimes (scales at least quadratically with number of samples)

D. Tuia. ECEO