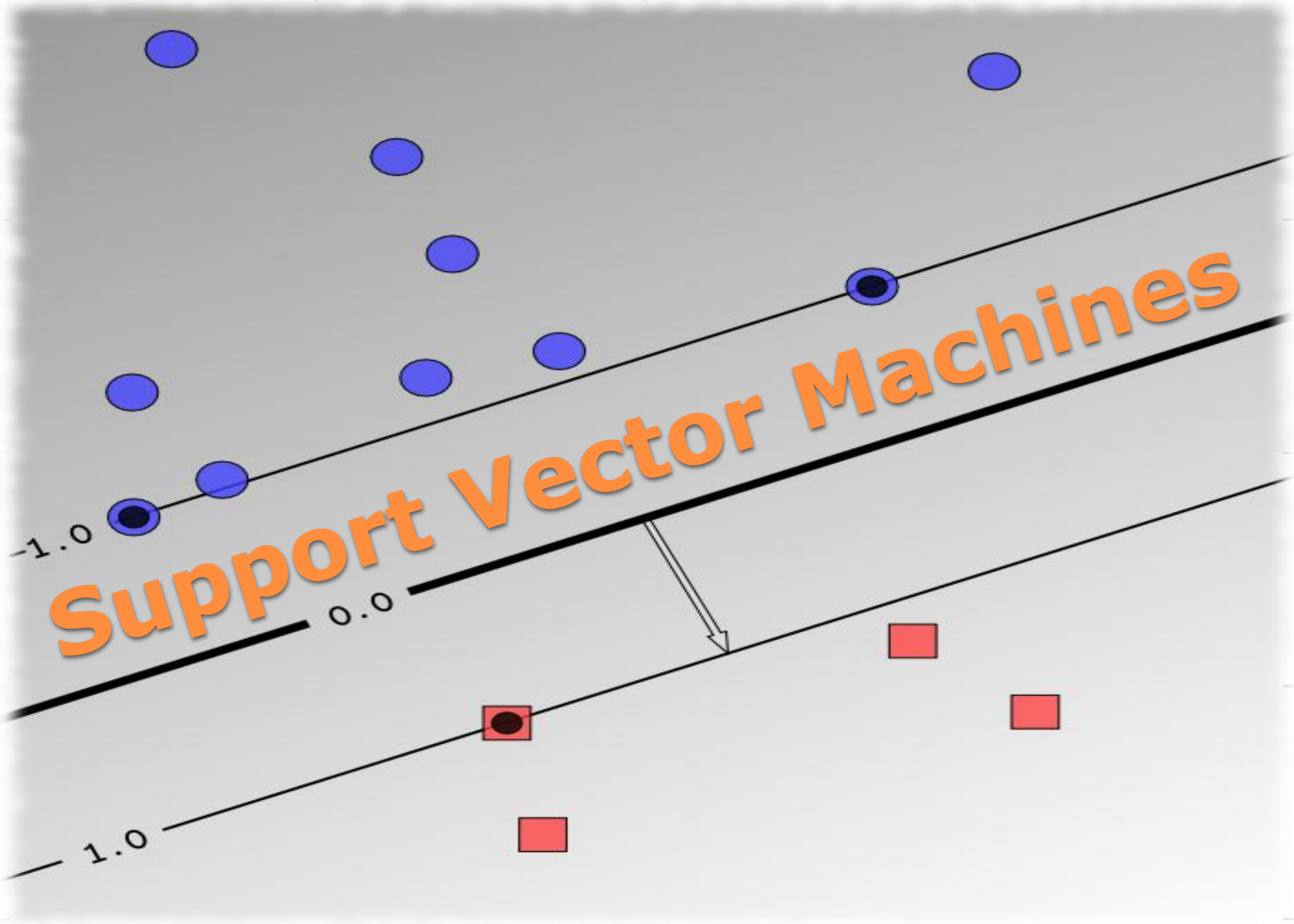
# Support Vector Machines



# Outline*

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

* Credits: These slides are a modified version of "A Simple Introduction to Support Vector Machines" by Martin Law. Lecture for CSE 802, Department of Computer Science and Engineering, Michigan State University.

# History of SVM

- SVM was first introduced in 1992.[1]
- SVM became popular because of its success in handwritten digit recognition.[2]
- SVM is now regarded as an important example of "kernel methods",[3] a key area in machine learning.


[1] B.E. Boser et al. A Training Algorithm for Optimal Margin Classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory 5 144-152, Pittsburgh, 1992.

[2] L. Bottou et al. Comparison of classifier methods: a case study in handwritten digit recognition. Proceedings of the 12th IAPR International Conference on Pattern Recognition, vol. 2, pp. 77-82, 1994.

[3] http://www.kernel-machines.org/publications/pdfs/0701907.pdf

# Outline

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

# Linear classifiers in a nutshell

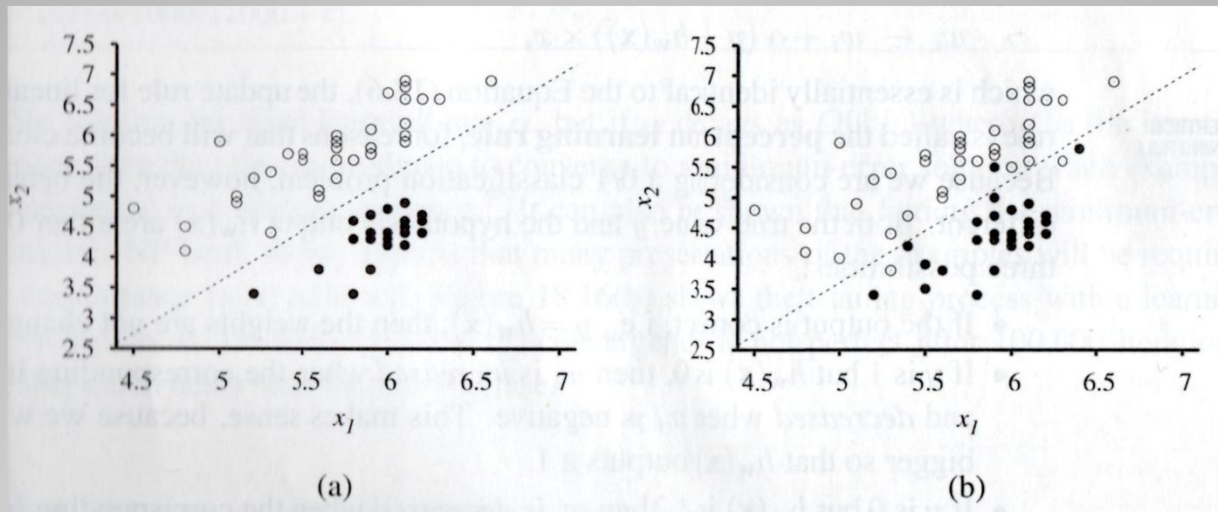
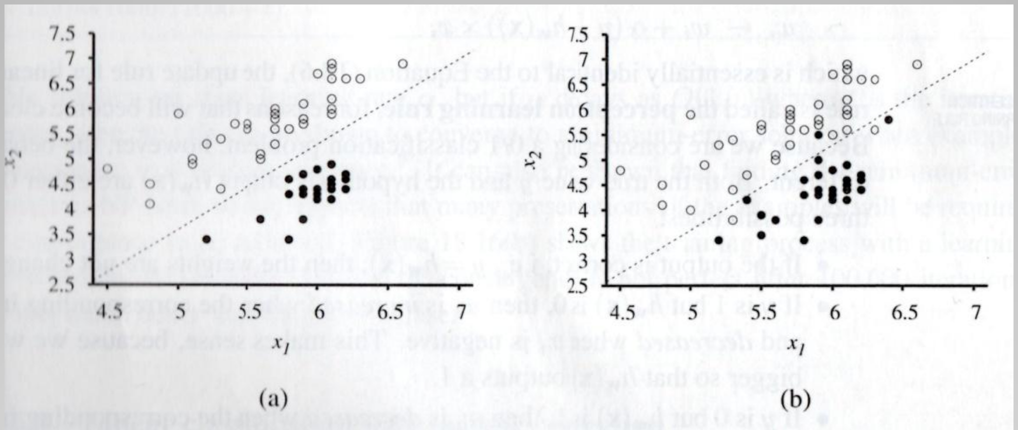



**Figure 18.15** (a) Plot of two seismic data parameters, body wave magnitude $x_1$ and surface wave magnitude $x_2$, for earthquakes (white circles) and nuclear explosions (black circles) occurring between 1982 and 1990 in Asia and the Middle East (Kebeasy *et al.*, 1998). Also shown is a decision boundary between the classes. (b) The same domain with more data points. The earthquakes and explosions are no longer linearly separable. [4]

- A **decision boundary** is a line (or a surface, in higher dimensions) that separates the two classes.

[4] Russell, S.J. and Norvig, P. (2010) Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall.

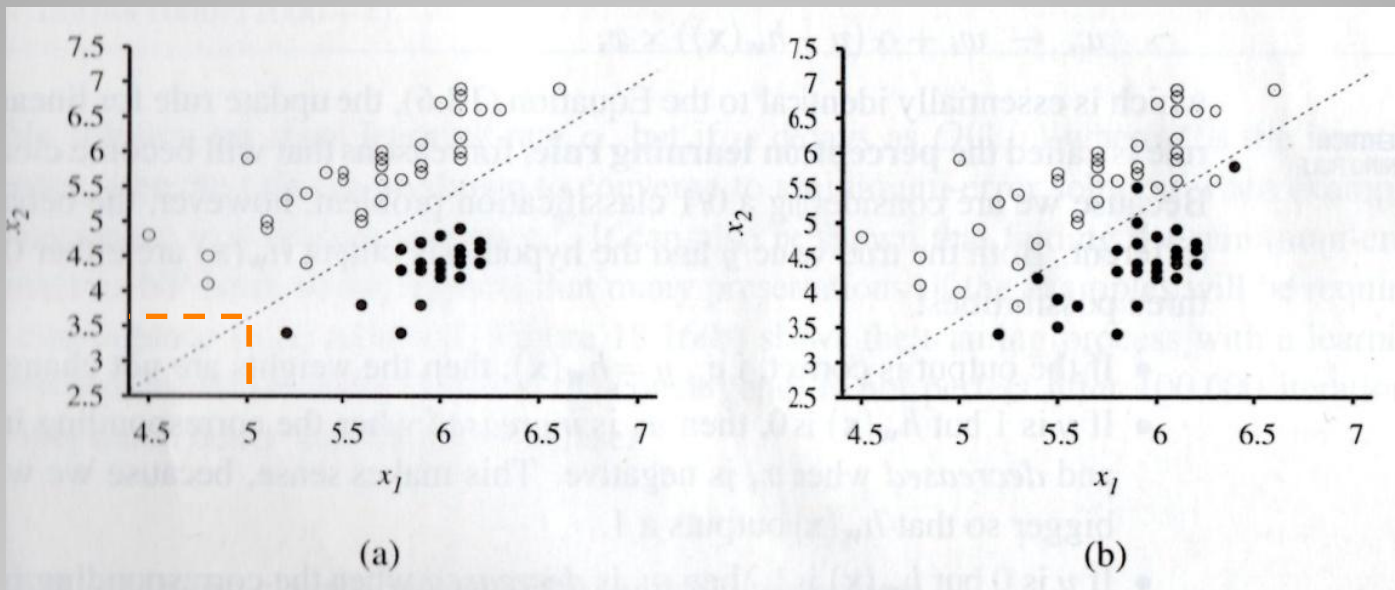
# Linear classifiers in a nutshell




- A linear decision boundary is called a **linear separator** and data that admit such a separator are called **linearly separable.** The linear separator in this case is:

$$-4.9+1.7x_1-x_2=0$$

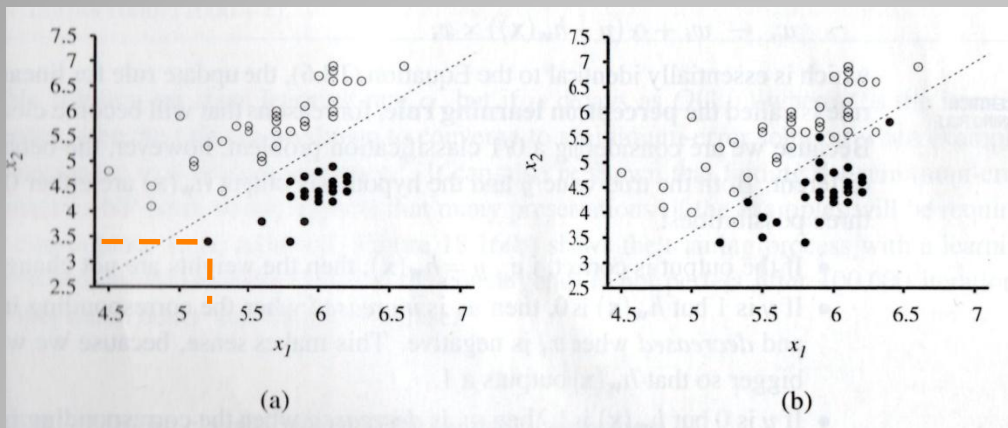
# Linear classifiers in a nutshell




5 3.6

$$-4.9 + 1.7x_1 - x_2 = 0$$

# Linear classifiers in a nutshell



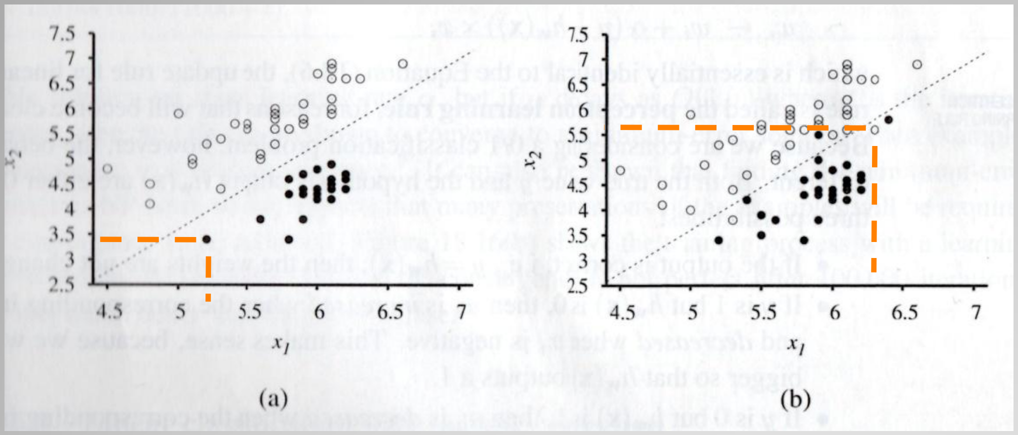

$$-4.9 + 1.7x_1 - x_2 = 0$$

- Nuclear explosions classification: $-4.9 + 1.7x_1 - x_2 > 0$

5.25 3.4

0.625

# Linear classifiers in a nutshell



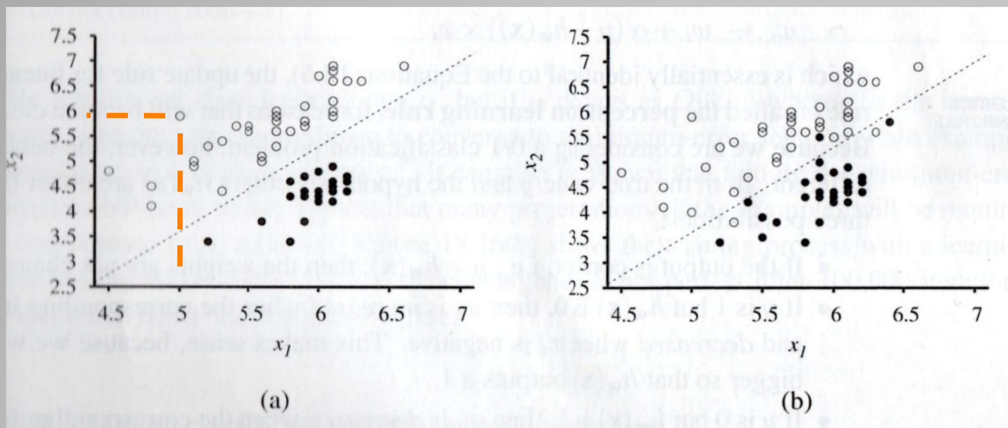

$$-4.9 + 1.7x_1 - x_2 = 0$$

- Nuclear explosions classification: $-4.9 + 1.7x_1 - x_2 > 0$

$$-4.9 + 1.7\,\underbrace{(6.35)}_{x_1} - \underbrace{5.6}_{x_2} = 0.295$$

$$\underbrace{5.25 \quad 3.4}_{0.625}$$

# Linear classifiers in a nutshell



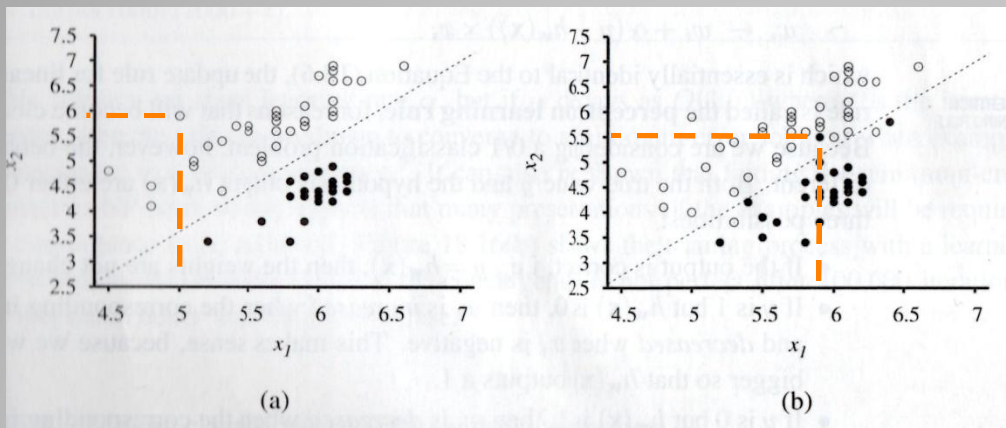

$$-4.9 + 1.7x_1 - x_2 = 0$$

- Nuclear explosions classification: $-4.9 + 1.7x_1 - x_2 > 0$
- Earthquakes classification: $-4.9 + 1.7x_1 - x_2 < 0$

5 5.9

-2.3

# Linear classifiers in a nutshell




$$-4.9 + 1.7x_1 - x_2 = 0$$

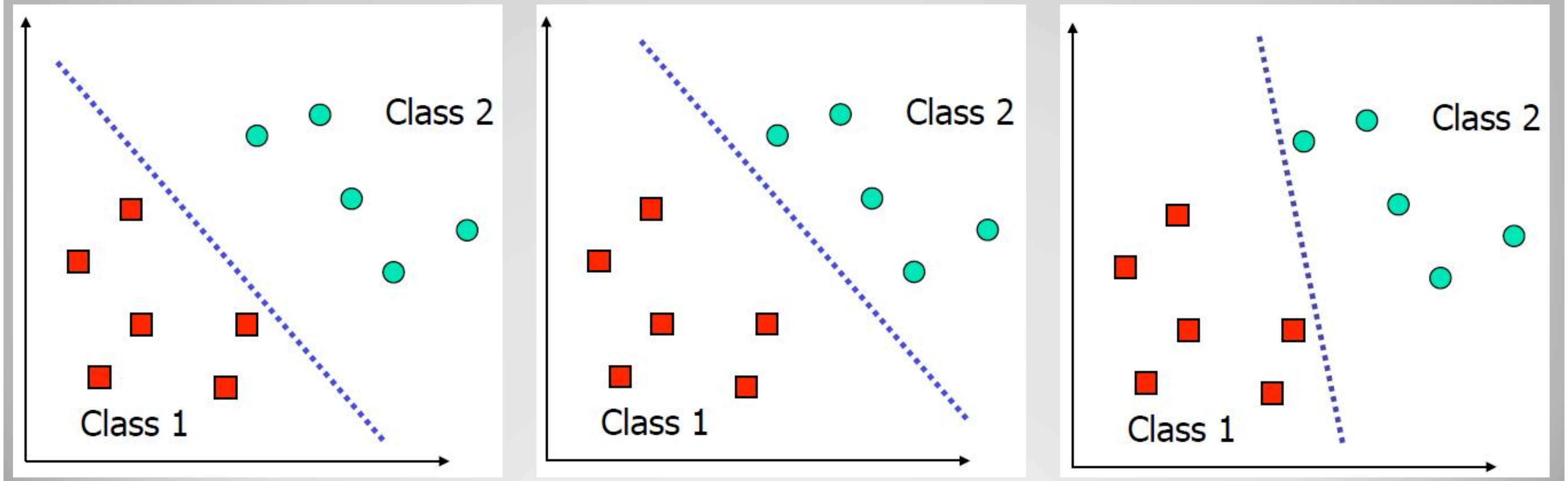
- Nuclear explosions classification: $-4.9 + 1.7x_1 - x_2 > 0$
- Earthquakes classification: $-4.9 + 1.7x_1 - x_2 < 0$

$$-4.9 + 1.7\underbrace{(5.85)}_{x_1} - \underbrace{5.5}_{x_2} = -0.455$$

$$\underbrace{5 \qquad 5.9}_{-2.3}$$

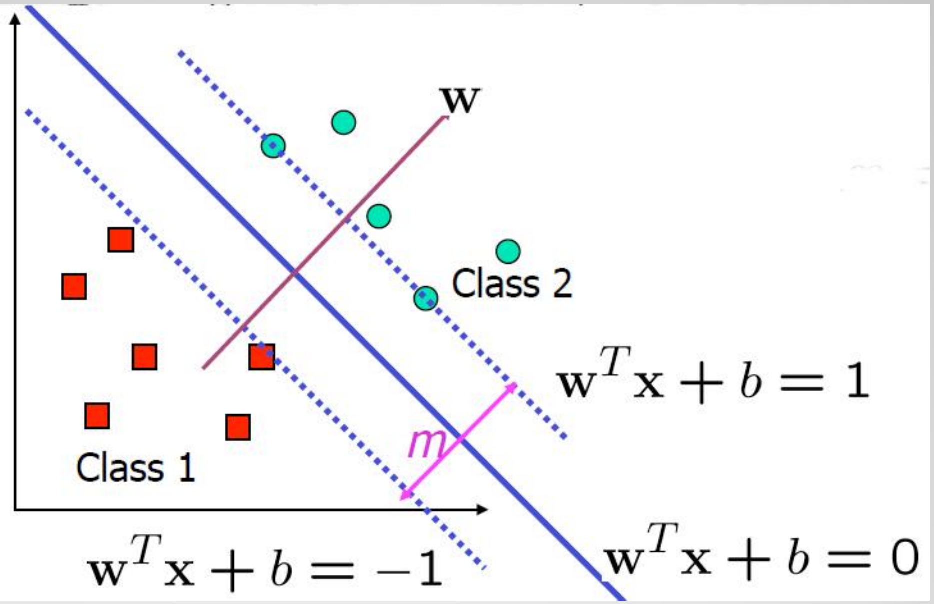
# What is a good Decision Boundary?

- Consider a two-class, linearly separable classification problem
- Many decision boundaries!
- Are all decision boundaries equally good?



# Large-margin Decision Boundary

- The decision boundary should be as far away as possible from the data of both classes.

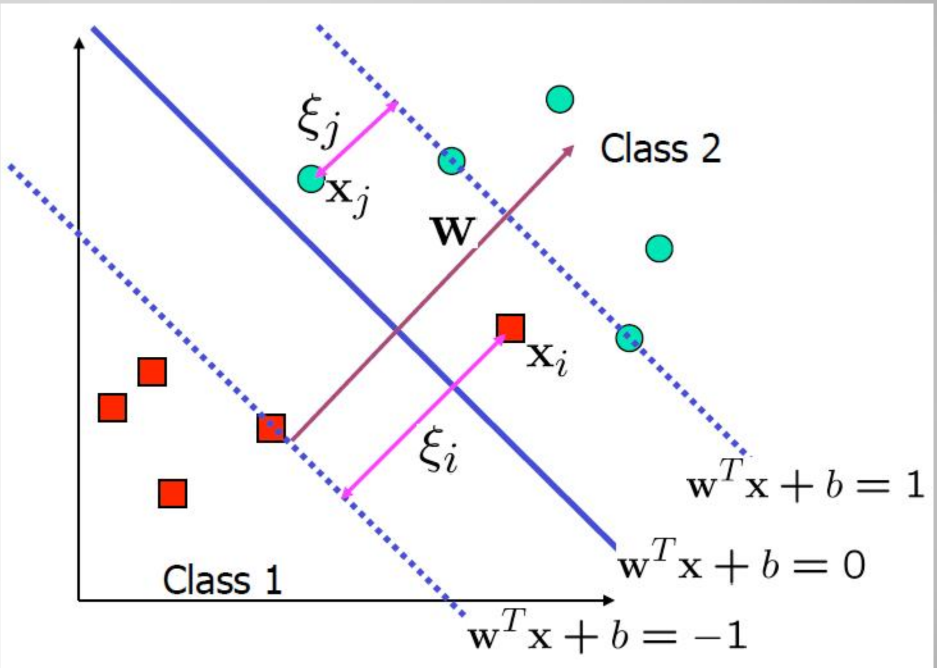


# Outline

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

# Non-linearly Separable Problems

- We allow "error" $\xi_i$ in classification; it is based on the output of the discriminant function $\mathbf{w}^T\mathbf{x}+b$



# How do we find the decision boundary?

The optimization problem becomes [5]

$$\text{Minimize } \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i$$

$$\text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

- C : tradeoff parameter between error and margin.

[5] Cortes, C. and Vapnik, V.N. (1995) Support vector networks. Machine Learning 20, 273–297.

# The Optimization Problem

The dual of this new constrained optimization problem is

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

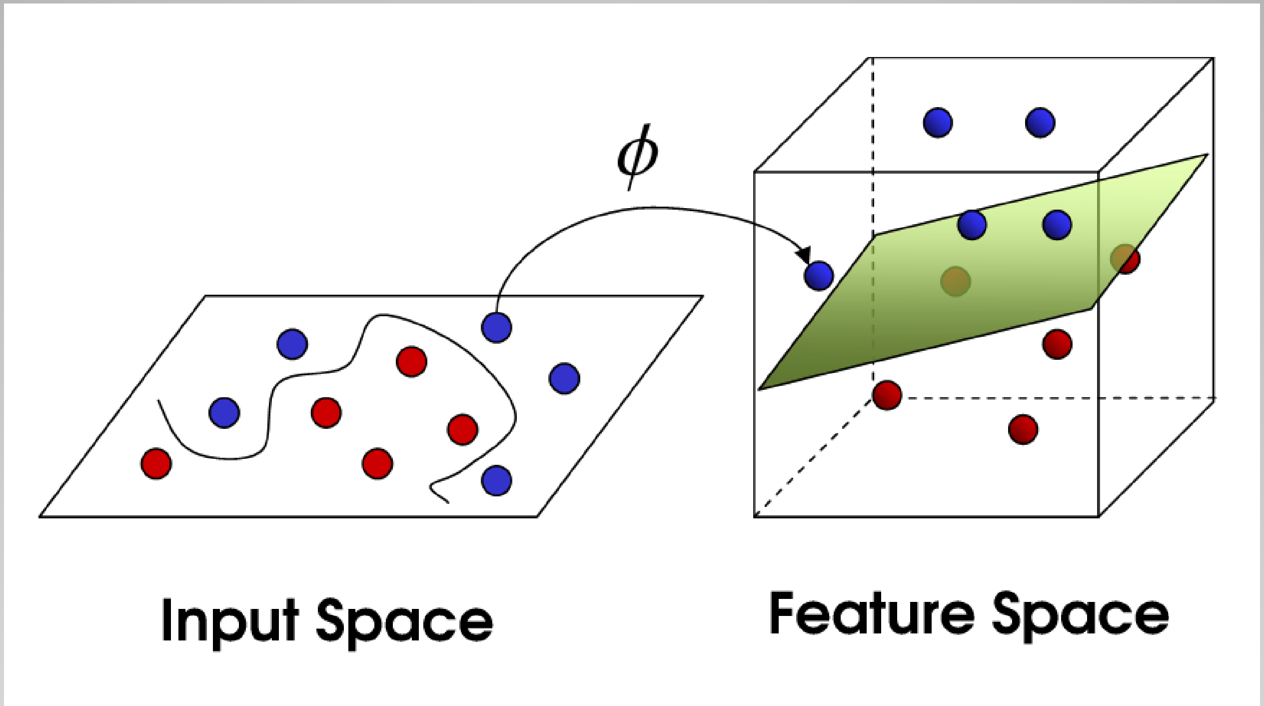
$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

**w** is recovered as $\mathbf{w} = \sum_{j=1}^{n} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$

# Extension to Non-linear Decision Boundary

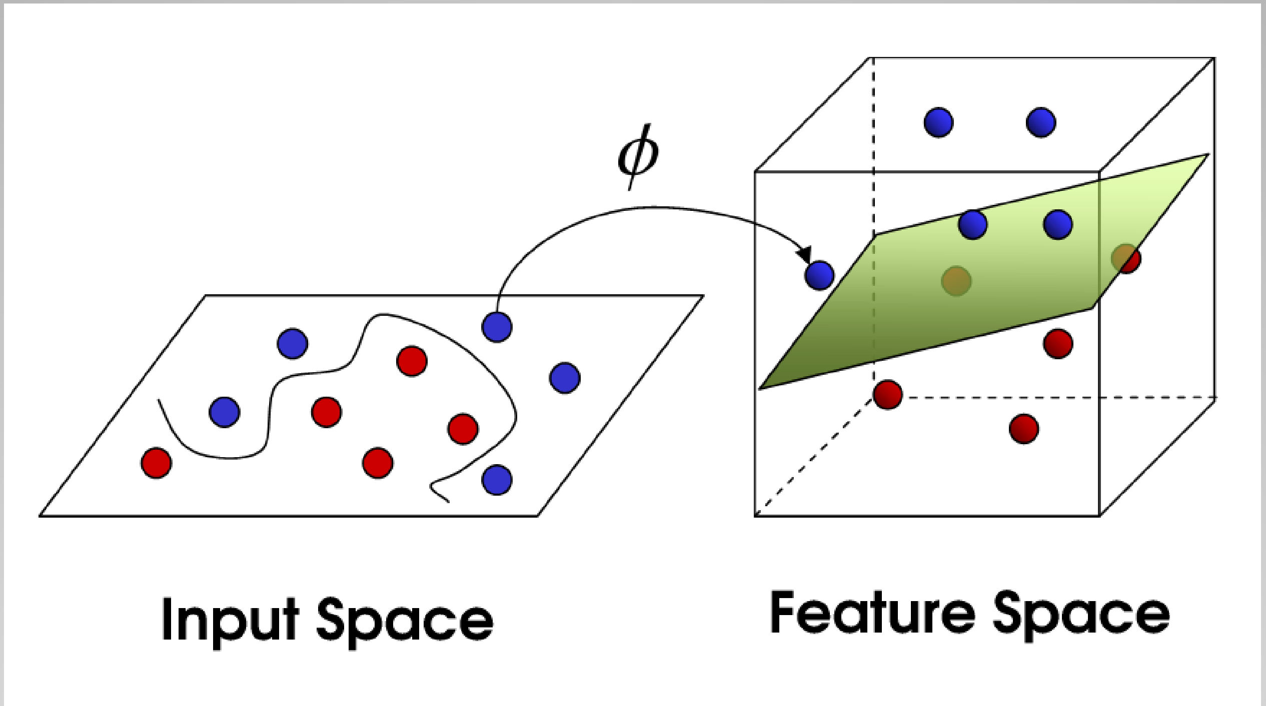
- So far, we have only considered large-margin classifier with a linear decision boundary.
- How to generalize it to become nonlinear?
- Key idea: transform $\mathbf{x}_i$ to a higher dimensional space to "make life easier".
  - Input space: the space the points $\mathbf{x}_i$ are located
  - Feature space: the space of $\phi(\mathbf{x}_i)$ after transformation.

# Transforming the Data




- Why transform?
  - Linear operation in the feature space is equivalent to nonlinear operation in input space.

# Transforming the Data




- Computation in the feature space can be costly because it is high dimensional.
    - The feature space is typically infinite-dimensional!
- The kernel trick comes to rescue.

# Outline

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

# The Kernel Trick

Recall the SVM optimization problem

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

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

The data points only appear as inner product

As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly

Many common geometric operations (angles, distances) can be expressed by inner products

Define the kernel function $K$ by

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$$

# Modification Due to Kernel Function

Original

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

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

With kernel function

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

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

# Modification Due to Kernel Function

Original

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

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

With kernel function

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

$$\text{subject to } C \geq \alpha_i \geq 0, \sum_{i=1}^{n} \alpha_i y_i = 0$$

# Introducing a kernel function

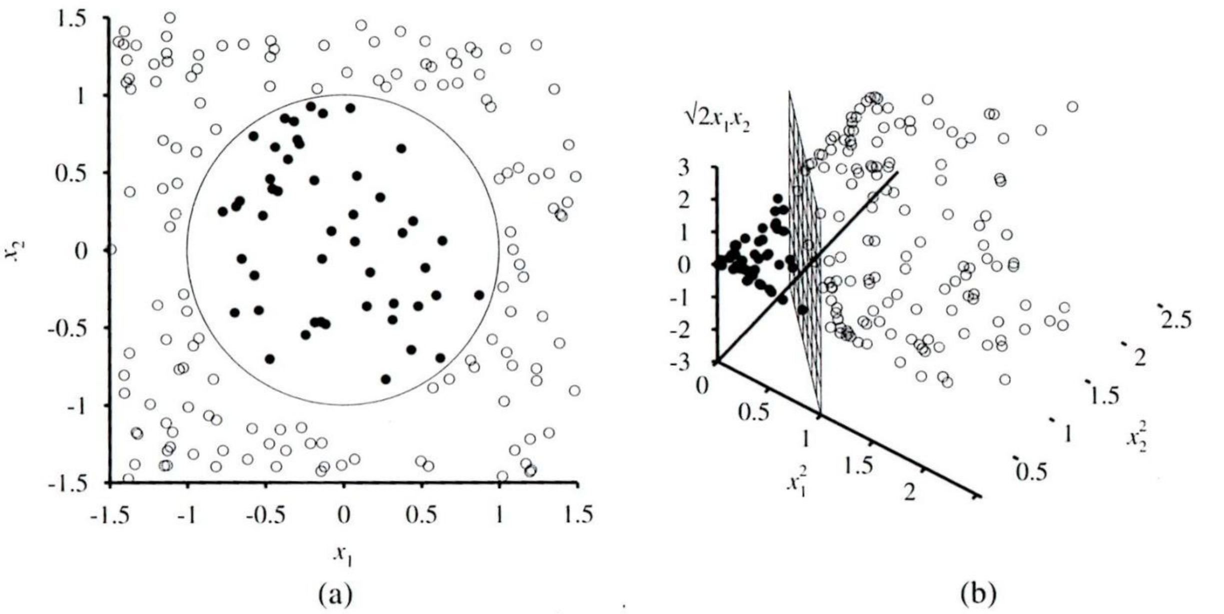



**Figure 18.31** (a) A two-dimensional training set with positive examples as black circles and negative examples as white circles. The true decision boundary, $x_1^2 + x_2^2 \leq 1$, is also shown. (b) The same data after mapping into a three-dimensional input space $(x_1^2, x_2^2, \sqrt{2}x_1x_2)$. The circular decision boundary in (a) becomes a linear decision boundary in three dimensions. Figure 18.30(b) gives a closeup of the separator in (b).

# Outline

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

# SVM four basic kernels

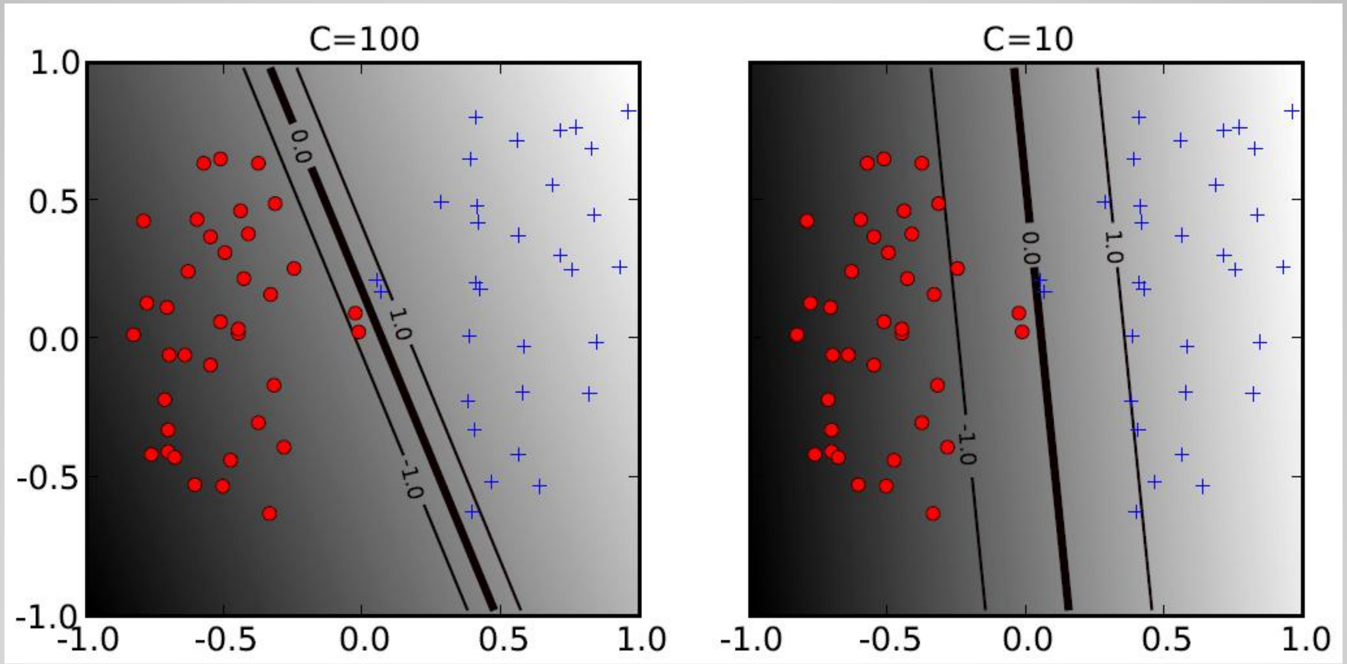
- linear: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$.
- polynomial: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i{}^T \mathbf{x}_j + r)^d$, $\gamma > 0$.
- radial basis function (RBF): $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$, $\gamma > 0$.
- sigmoid: $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\gamma \mathbf{x}_i{}^T \mathbf{x}_j + r)$.

Here, $\gamma$, $r$, and $d$ are kernel parameters. [6]

[6] A Practical Guide to Support Vector Classification. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin.
http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

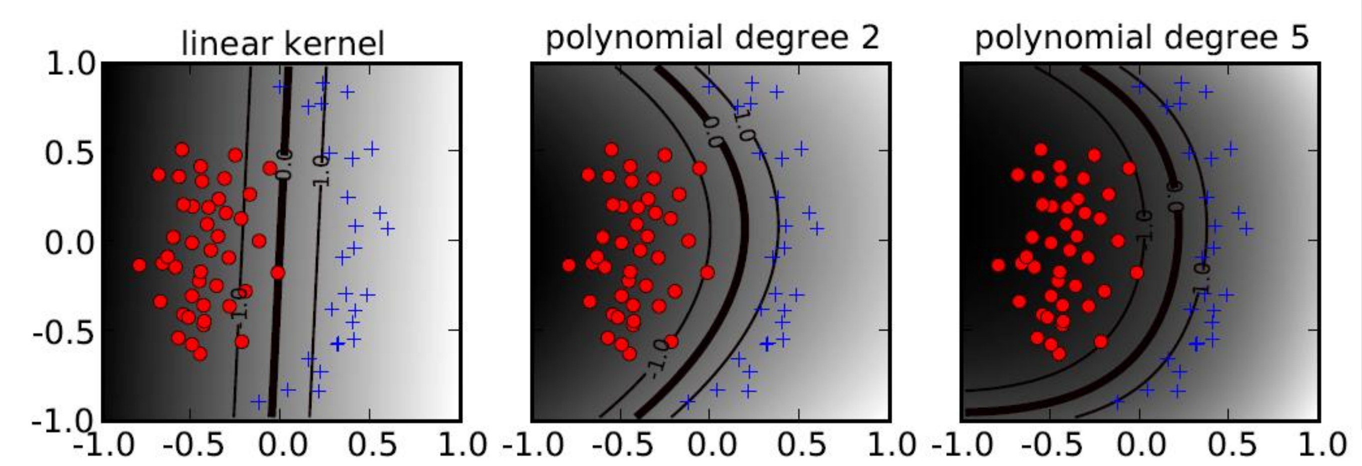
# Linear kernel

- C is not a parameter of this kernel, but its values are usually tune up to improve the classifier performance.

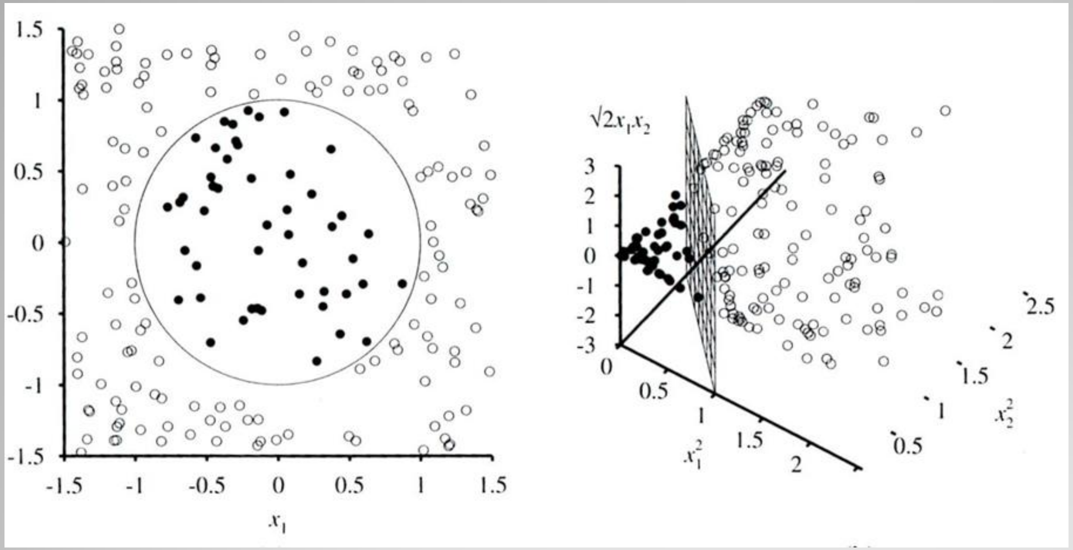


# Polynomial kernel

- Parameter *d* increases the features space:
  - Number of monomials with sum of exponents lower or equal to *d*.
- Higher degree polynomial kernels allow a more flexible decision boundary, but may lead to overfitting.



# Polynomial kernel




$$f_1 = x_1^2$$

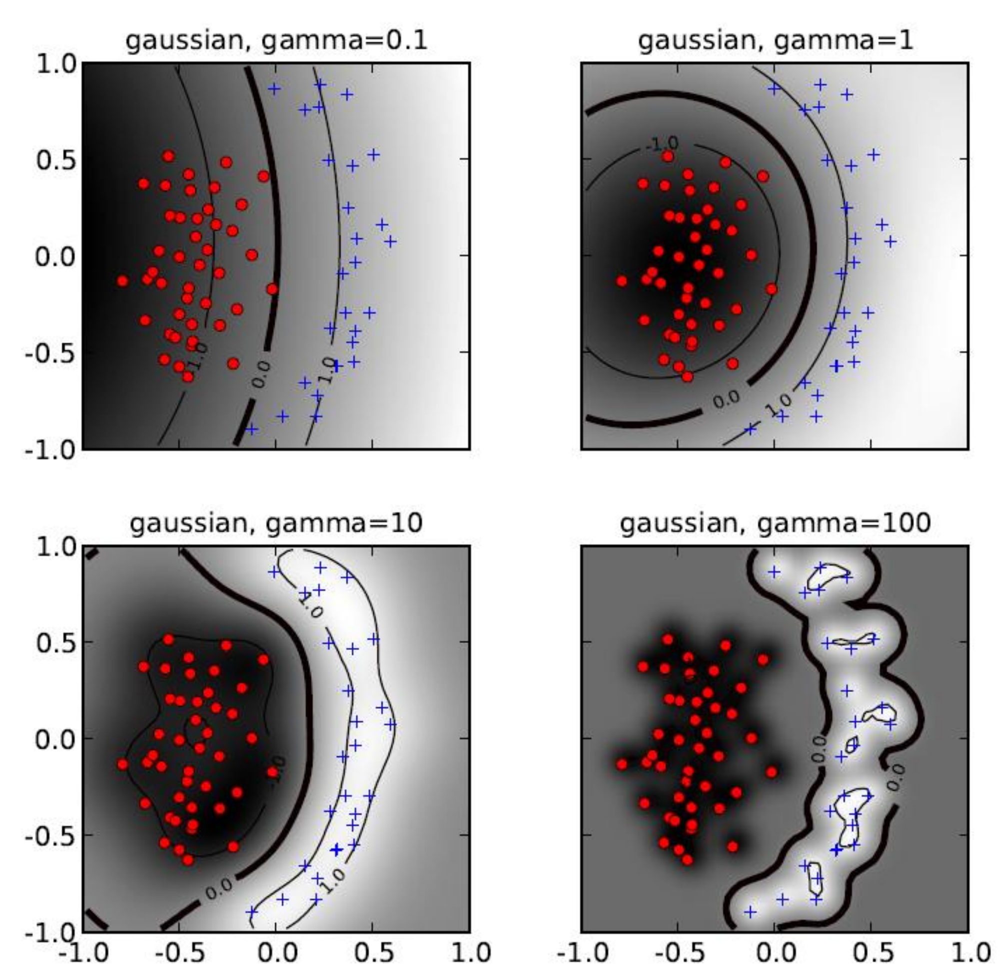
$$f_2 = x_2^2$$

$$f_3 = \sqrt{2}x_1x_2$$

- The numbers of monomials is calculated as:

$$\binom{d+m-1}{d} = \frac{(d+m-1)!}{d!(m-1)!} = \frac{(2+2-1)!}{2!(2-1)!} = 3$$

# RBF kernel



gaussian, gamma=0.1
gaussian, gamma=1
gaussian, gamma=10
gaussian, gamma=100
1.0
0.5
0.0
-0.5
-1.0
-1.0
-0.5
0.0
0.5
1.0

# Sigmoid Kernel

- This kernel was quite popular for support vector machines due to its origin from neural network theory.
- An SVM model using a sigmoid kernel function is equivalent to a two-layer, perceptron neural network.
- The sigmoid kernel is not better than the RBF kernel in general.[7]

[7] Hsuan-Tien Lin and Chih-Jen Lin. A Study on Sigmoid Kernels for SVM and the Training of non-PSD Kernels by SMO-type Methods. http://home.caltech.edu/~htlin/publication/doc/tanh.pdf

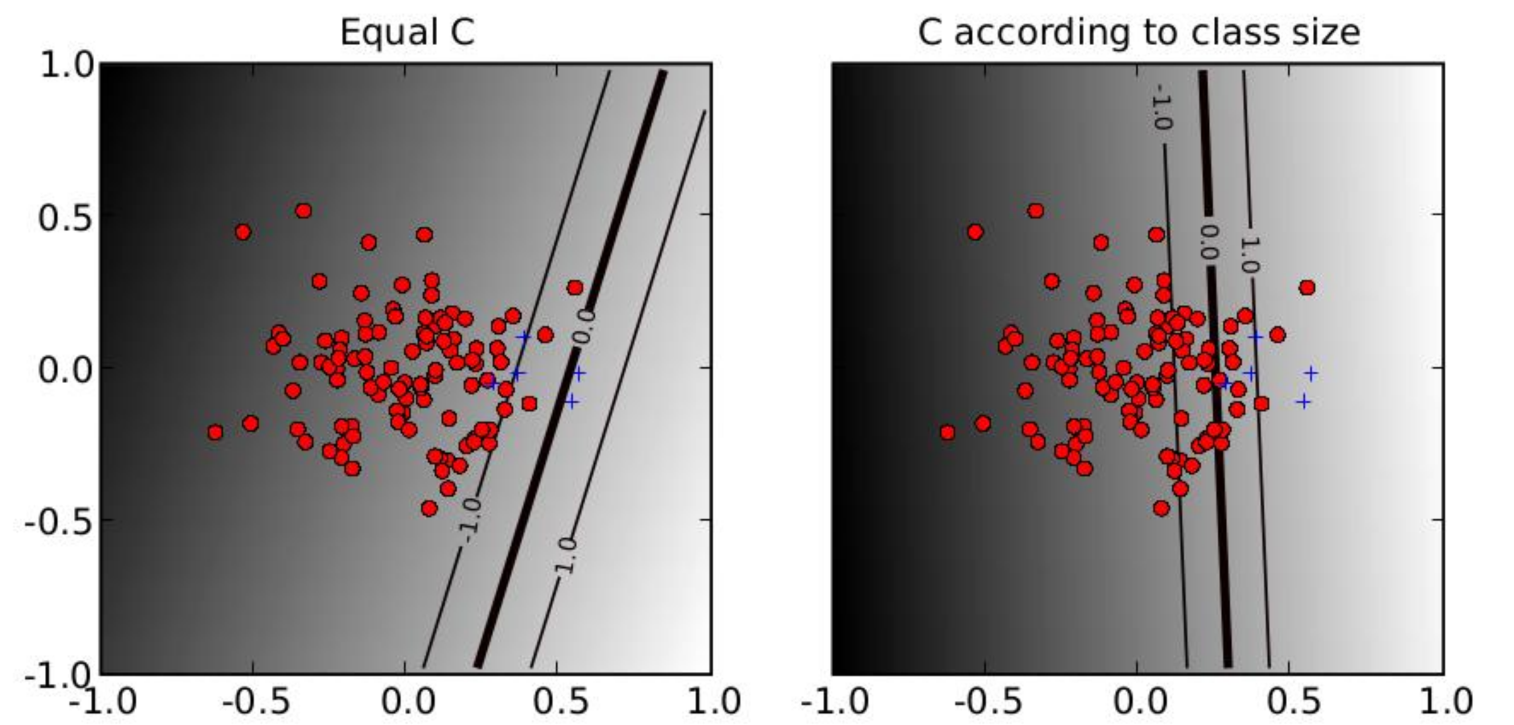
# Outline

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

# SVMs for Unbalanced Data

$$C \sum_{i=1}^{n} \xi_i \longrightarrow C_+ \sum_{i \in I_+} \xi_i + C_- \sum_{i \in I_-} \xi_i$$

$$C_+ n_+ = C_- n_-$$



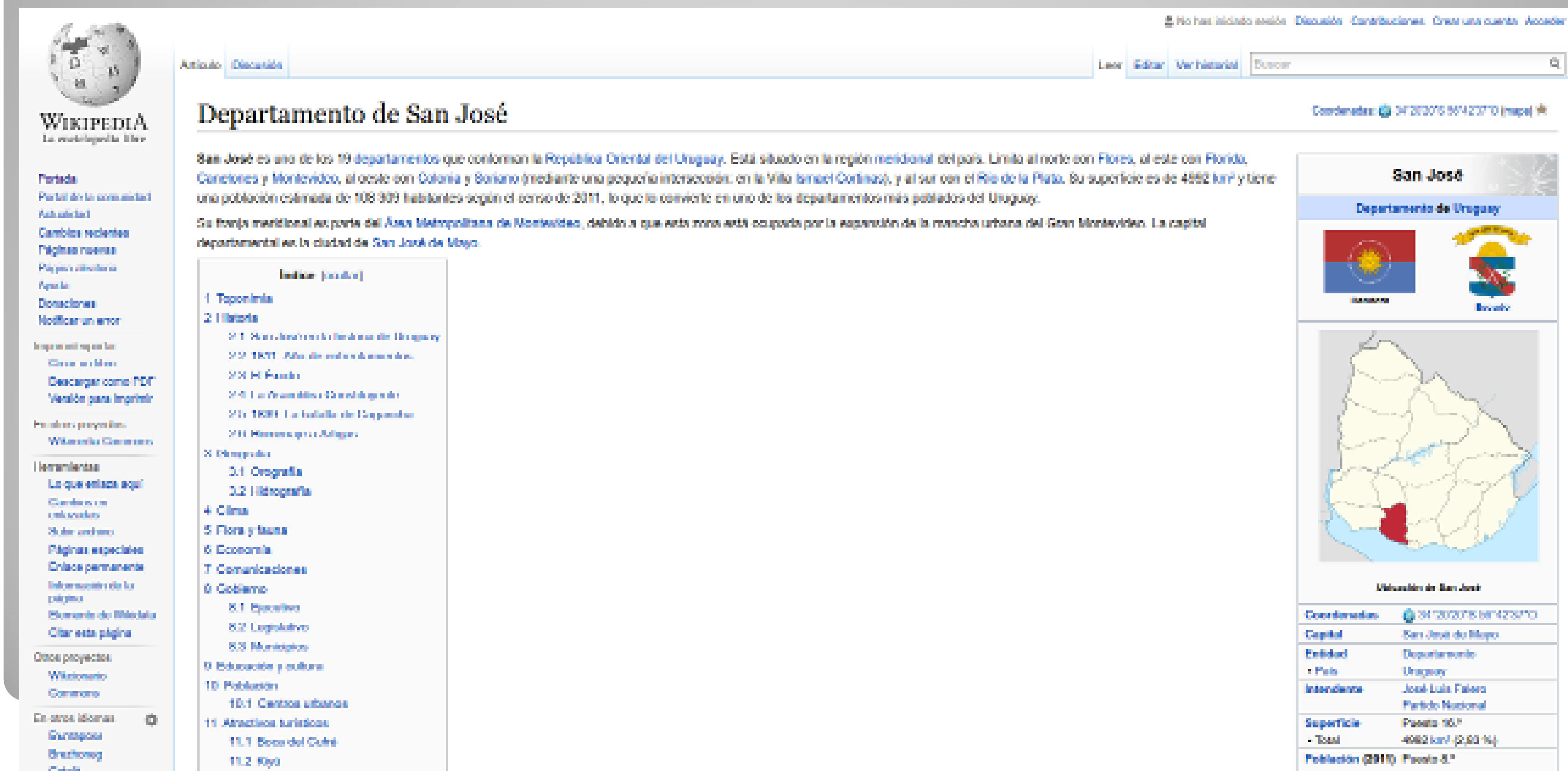
# SVM for multi-class classification?

- One can change the QP formulation to become multi-class.
- More often, building binary classifiers which distinguish between:[8]
  - one of the labels and the rest (*one-versus-all*).
  - between every pair of classes (*one-versus-one*).

[8] M. Galar, A. Fernández, E. Barrenechea, H. Bustince and F. Herrera, An Overview of Ensemble Methods for Binary Classifiers in Multi-class Problems: Experimental Study on One-vs-One and One-vs-All Schemes. Pattern Recognition 44:8 (2011) 1761-1776. http://sci2s.ugr.es/ovo-ova

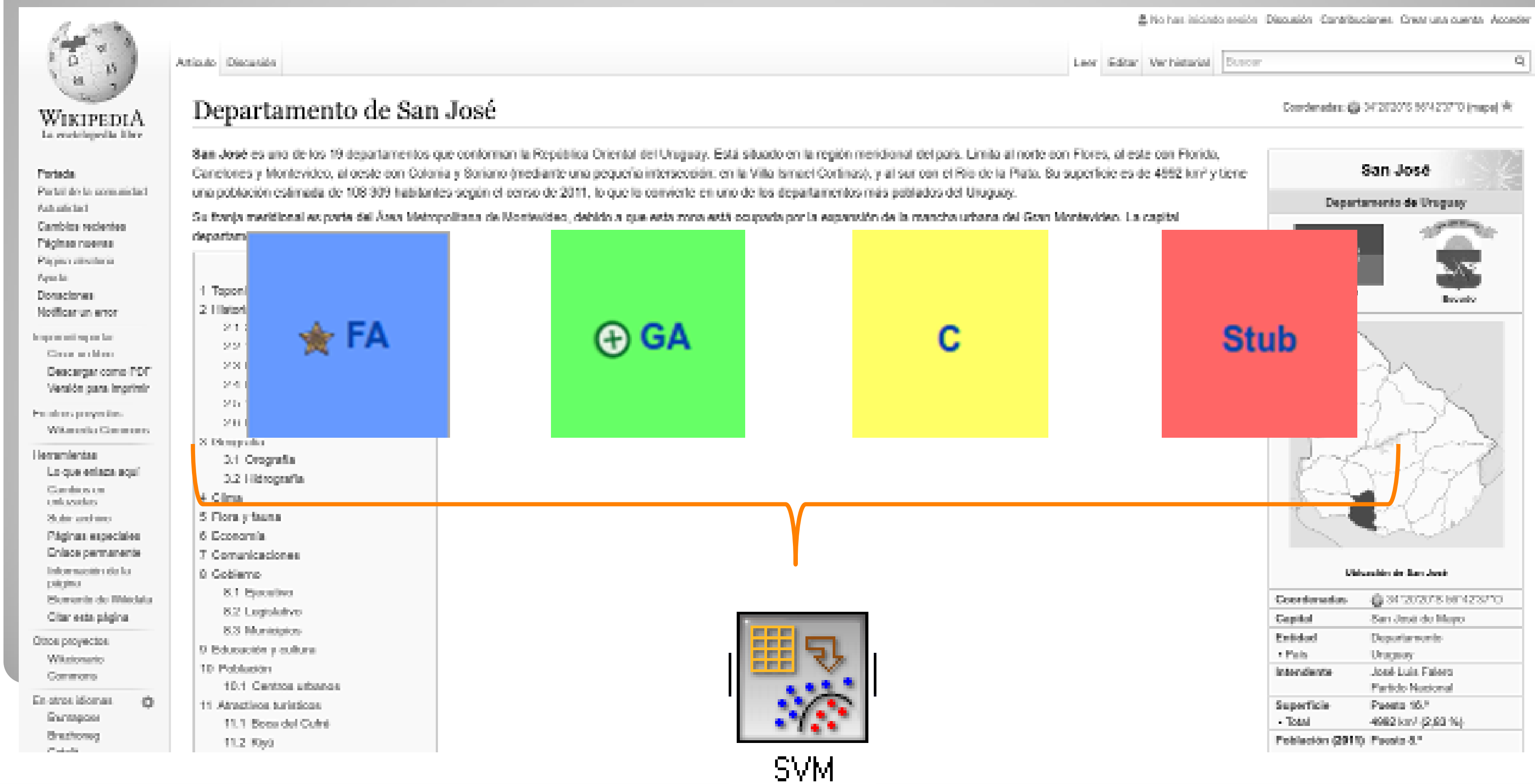
# SVM for multi-class classification?

- One of the labels and the rest (*one-versus-all*)
- Between every pair of classes (*one-versus-one*)



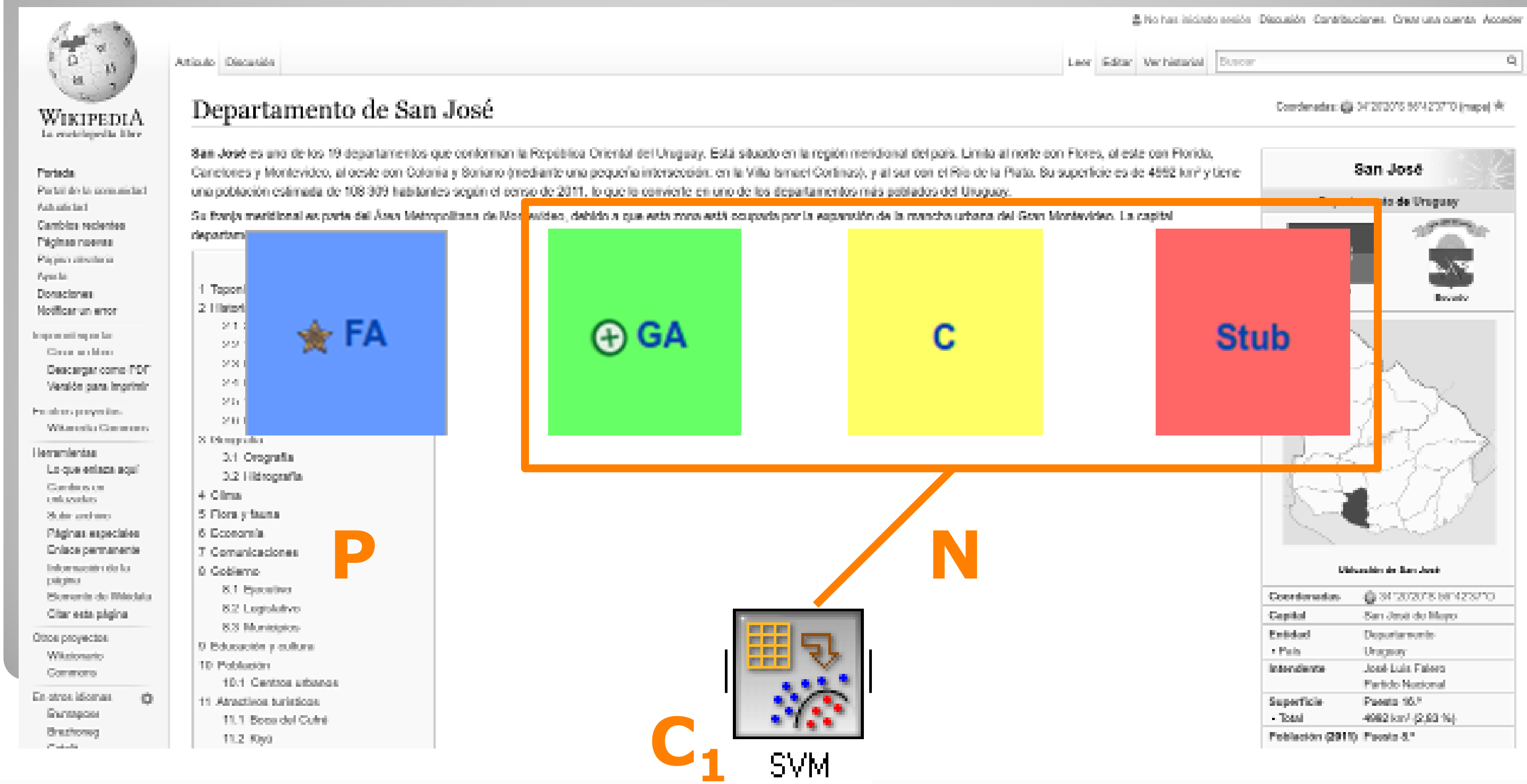
# SVM for multi-class classification?

- One of the labels and the rest (*one-versus-all*)
- Between every pair of classes (*one-versus-one*)



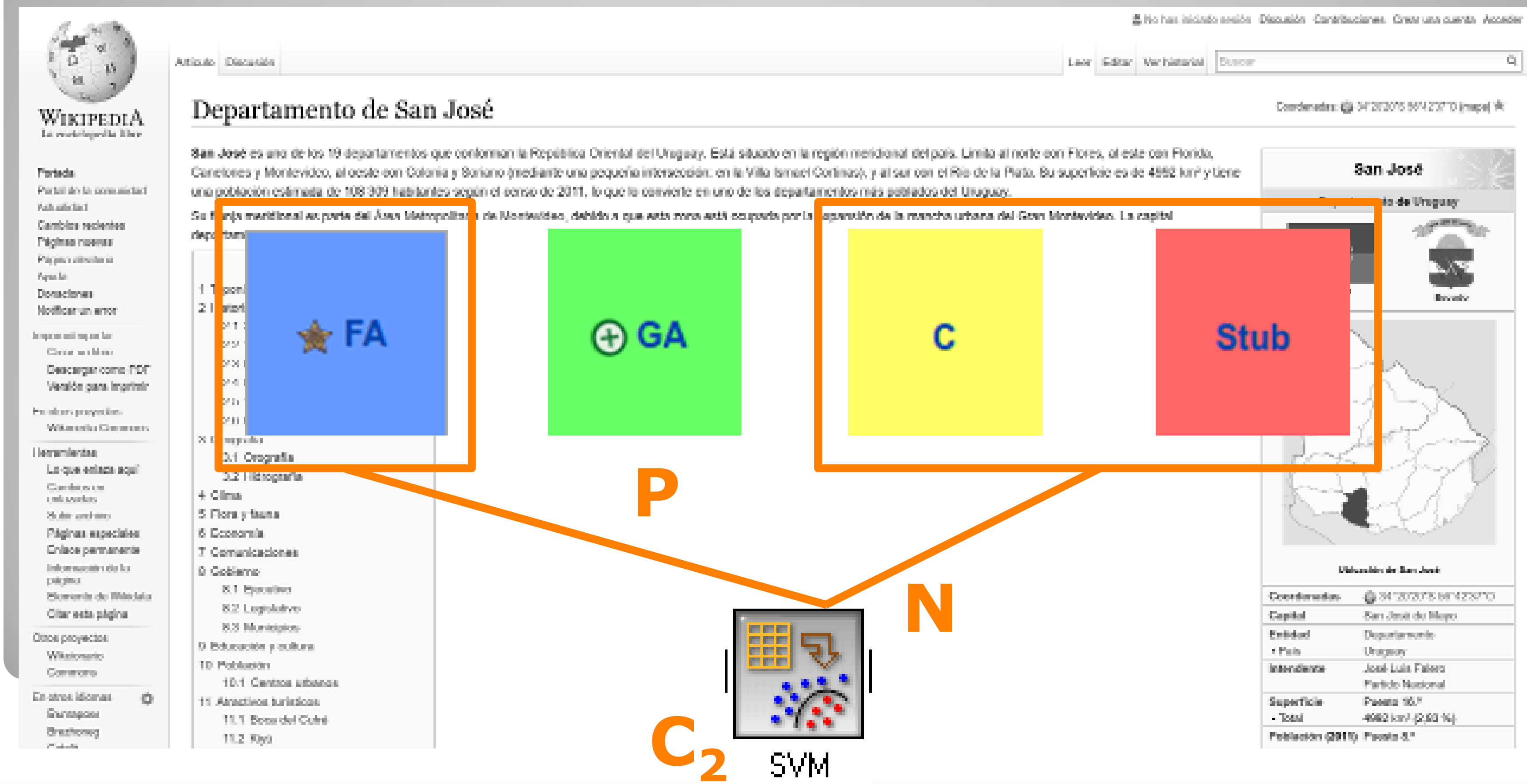
# SVM for multi-class classification?

- One of the labels and the rest (*one-versus-all*)



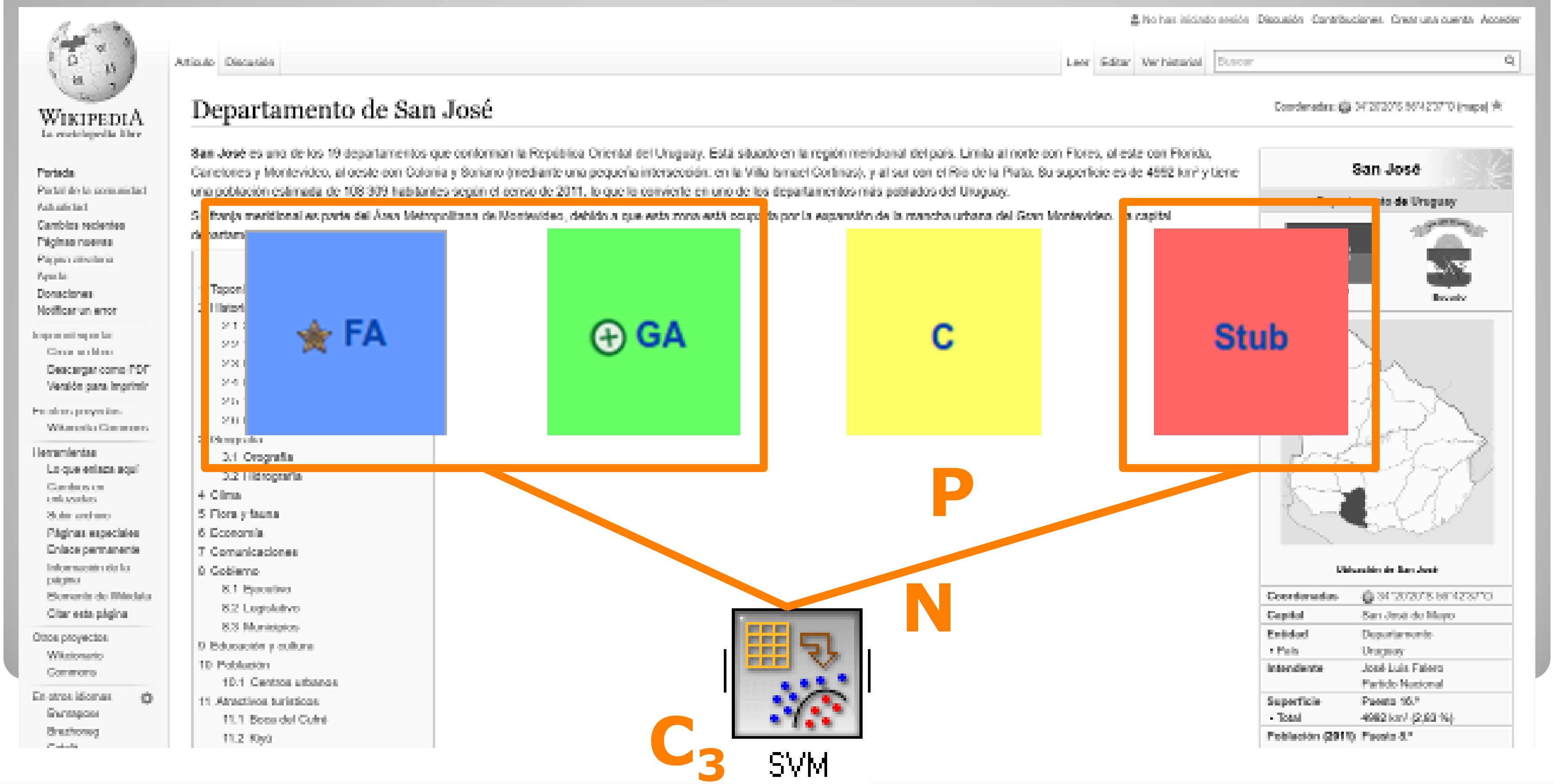
# SVM for multi-class classification?

- One of the labels and the rest (*one-versus-all*)



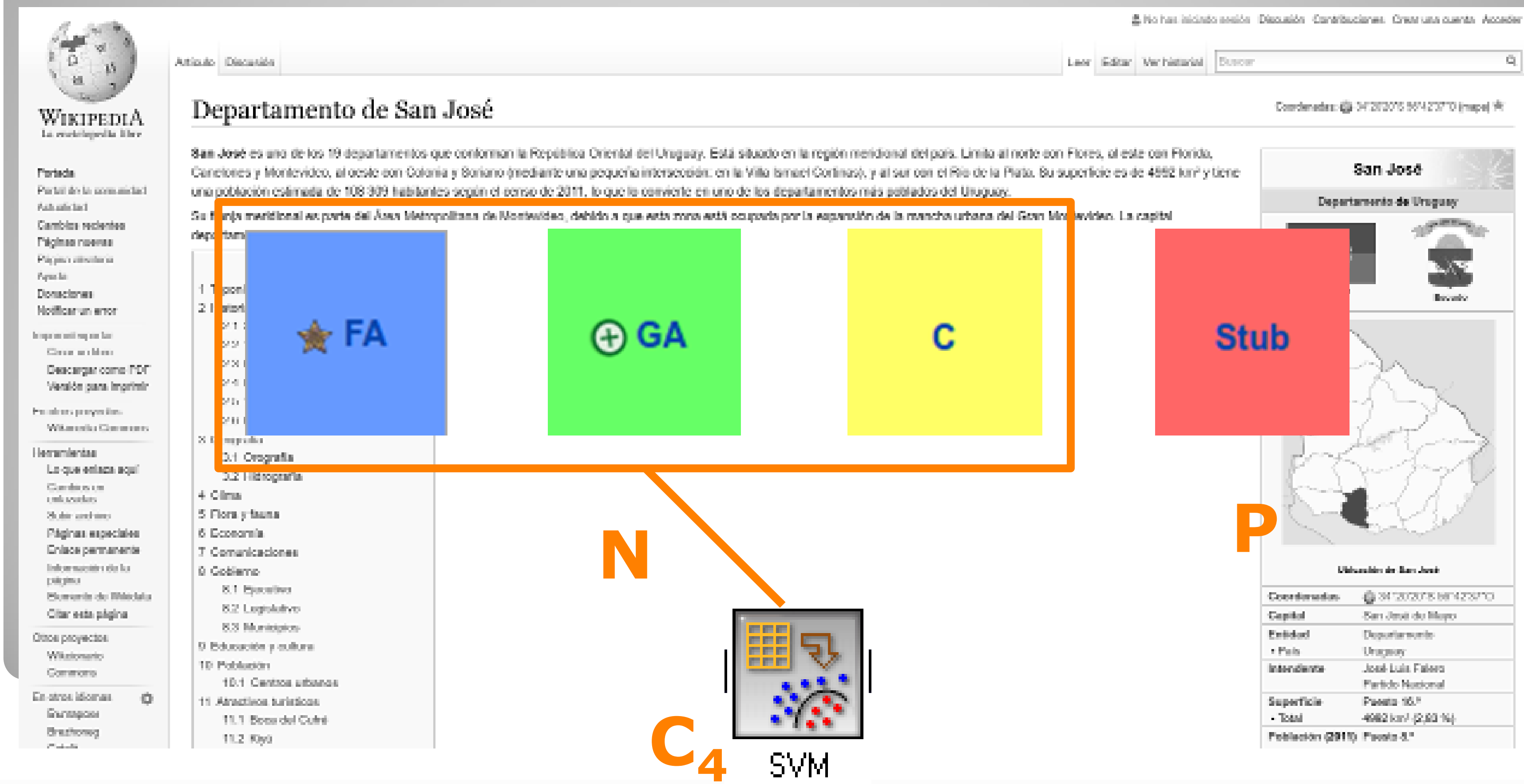
# SVM for multi-class classification?

- One of the labels and the rest (*one-versus-all*)



# SVM for multi-class classification?

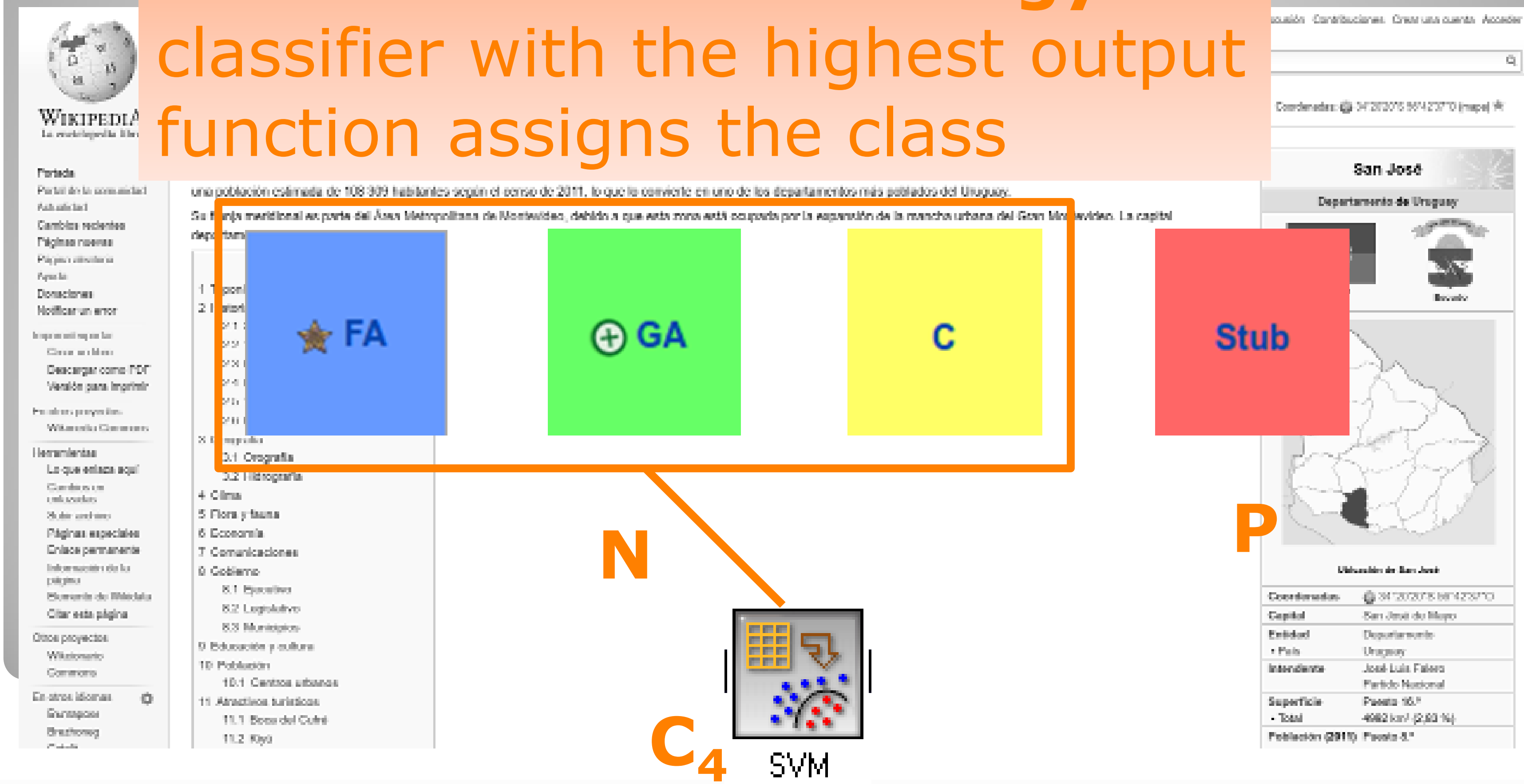
- One of the labels and the rest (*one-versus-all*)



# SVM for multi-class classification?

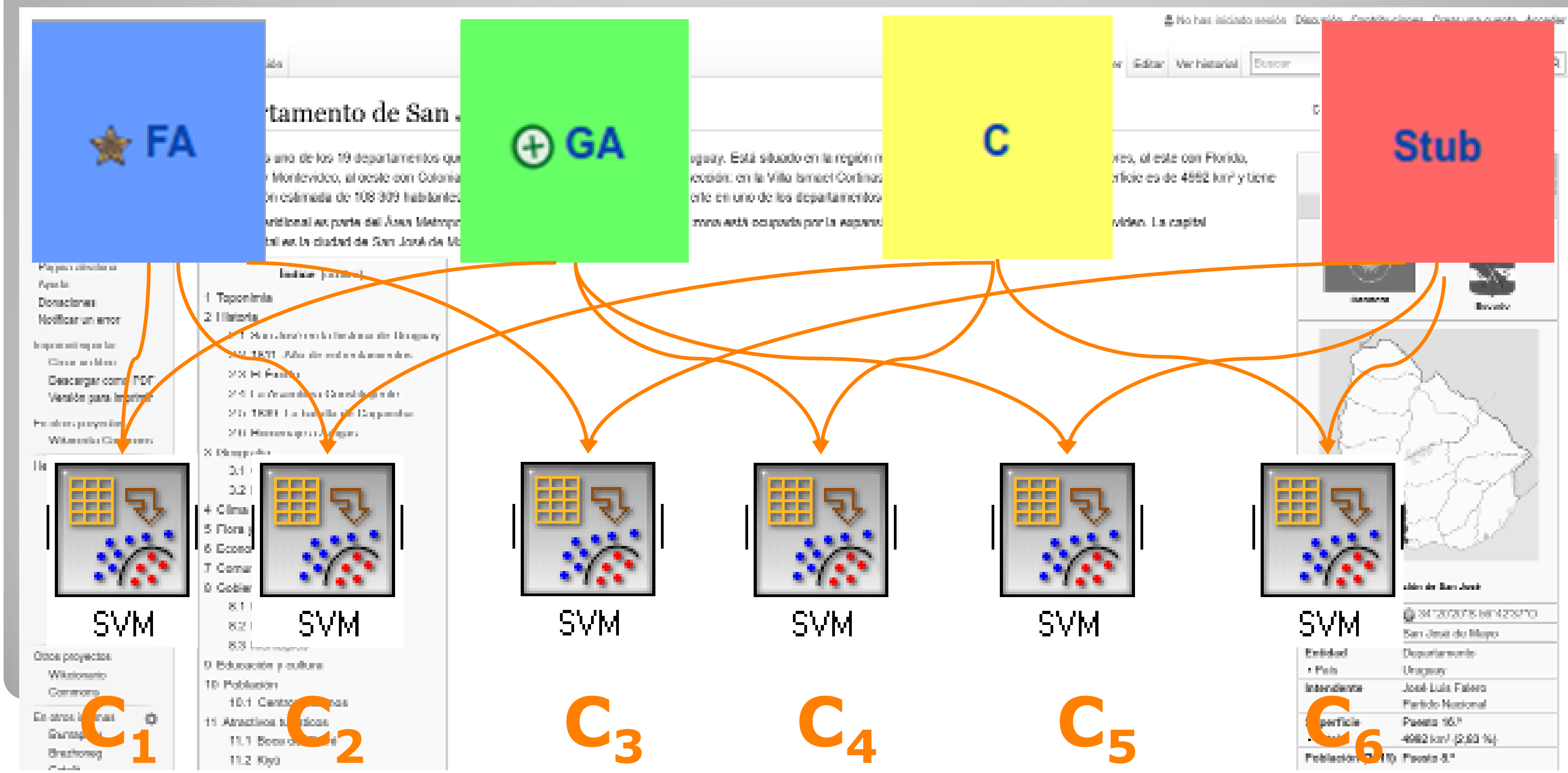
- One of the labels and the rest (*one-versus-all*)

**winner-takes-all strategy:**
classifier with the highest output function assigns the class



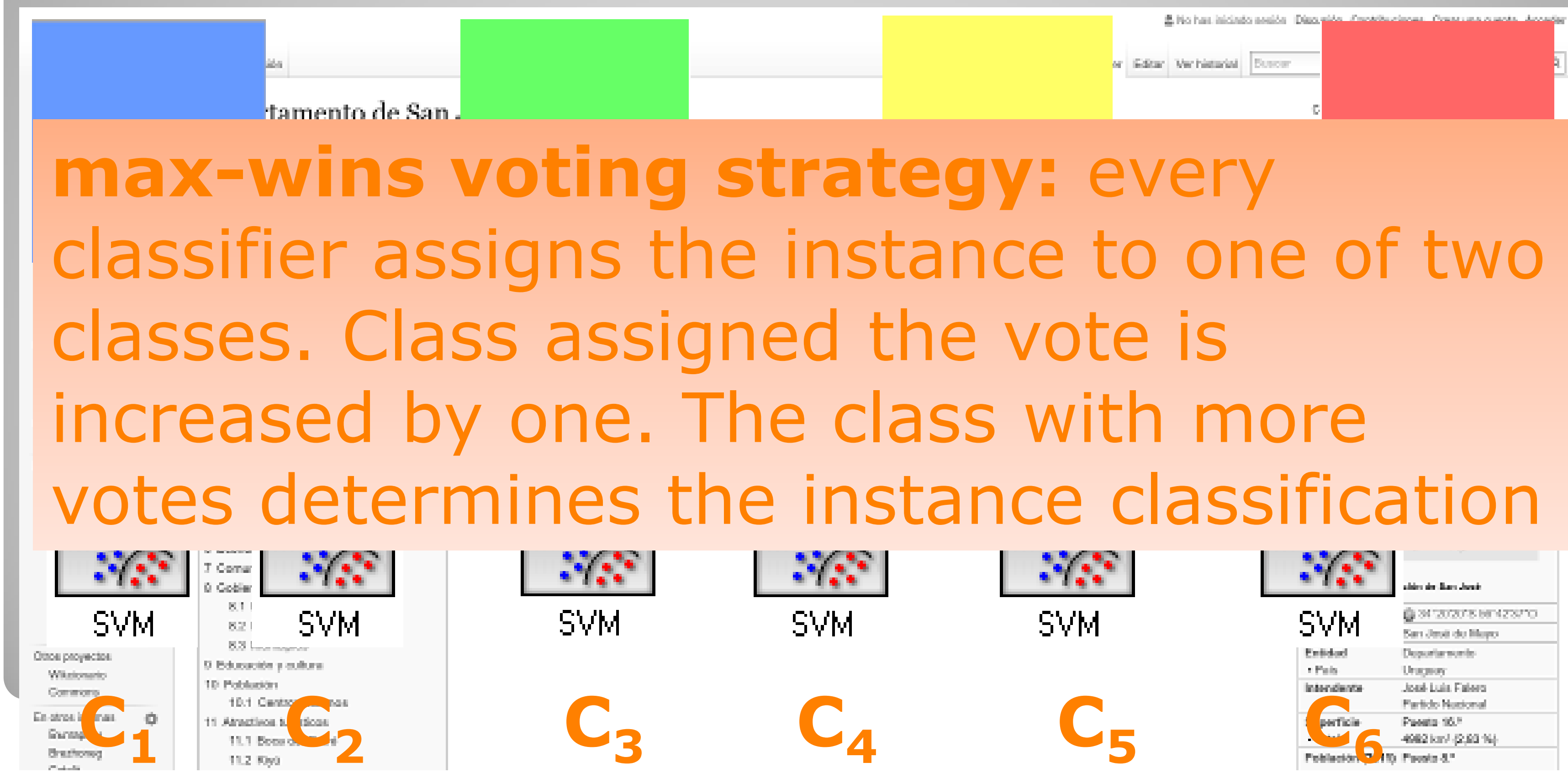
# SVM for multi-class classification?

- Between every pair of classes (*one-versus-one*)



# SVM for multi-class classification?

- Between every pair of classes (*one-versus-one*)




**max-wins voting strategy:** every classifier assigns the instance to one of two classes. Class assigned the vote is increased by one. The class with more votes determines the instance classification

# Outline

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

# Software

- A list of SVM implementation can be found at http://www.kernel-machines.org/software.html
- Some implementation (such as LIBSVM) can handle multi-class classification.
- SVMLight is among one of the earliest implementation of SVM.
- Several Matlab toolboxes for SVM are also available.
- PyML - Machine Learning in Python http://pyml.sourceforge.net/svm_howto.html
- http://scikit-learn.org/stable/modules/svm.html

# Outline

- A brief history of SVM
- Linear classifiers in a nutshell
  - Linear separable
  - Nonlinear separable
- Creating nonlinear classifiers: the kernel trick
- SVM kernel functions
- SVM extensions
- Software
- Concluding examples and discussion

# Conclusions

- Kernel methods owe their name to the use of **kernel functions**, which enable them to operate in a high-dimensional, **implicit** feature space without ever computing the coordinates of the data in that space, but rather by simply computing the **inner products** between the images of all pairs of data in the feature space.
- This operation is often computationally cheaper than the explicit computation of the coordinates. This approach is called the **kernel trick**.
- SVM is one of the most popular kernel-based methods.

# Conclusions

- Choosing the kernel function: probably the most tricky part of using SVM.
  - In practice, a low degree polynomial kernel or RBF kernel with a reasonable width is a good initial try. Grid search to perform parameters tuning. See e.g. [9].
- How to use SVM for multi-class classification?
  - One can change the QP formulation to become multi-class.
  - More often, building binary classifiers which distinguish between:
    - one of the labels and the rest (*one-versus-all*).
    - between every pair of classes (*one-versus-one*).

[9] A Practical Guide to Support Vector Classification. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

# References


- A Simple Introduction to Support Vector Machines by Martin Law. Lecture for CSE 802, Department of Computer Science and Engineering, Michigan State University.
- Artificial Intelligence: A Modern Approach. Stuart Russell and Peter Norvig. 3rd Edition. Section 18.9, pp.744-748
- Data Mining: Practical Machine Learning Tools and Techniques. Ian H. Witten and Eibe Frank. 2nd Edition. Section 6.3, pp. 214-222
- A Practical Guide to Support Vector Classification. Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf
- A User's Guide to Support Vector Machines. Asa Ben-Hur and Jason Weston. In Data Mining Techniques for the Life Sciences, 2010, Springer.