

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
- Conclusions

^{*} 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 is related to statistical learning theory.^[2]
- SVM became popular because of its success in handwritten digit recognition.^[3]
- SVM is now regarded as an important example of "kernel methods",^[4] 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] V. Vapnik. The Nature of Statistical Learning Theory. 2nd edition, Springer, 1999.

[3] 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. [4] 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
- Conclusions

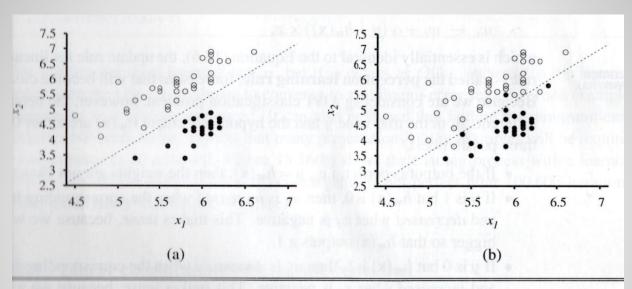
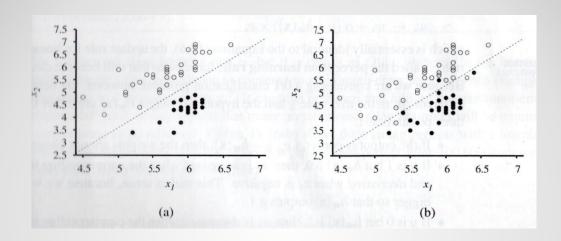


Figure 18.15 (a) Plot of two seismic data parameters, body wave magnitude x_1 and surface wave magnitude x_2 , for earthquakes (white circles) and nuclear explosions (black circles) occurring between 1982 and 1990 in Asia and the Middle East (Kebeasy *et al.*, 1998). Also shown is a decision boundary between the classes. (b) The same domain with more data points. The earthquakes and explosions are no longer linearly separable. [5]

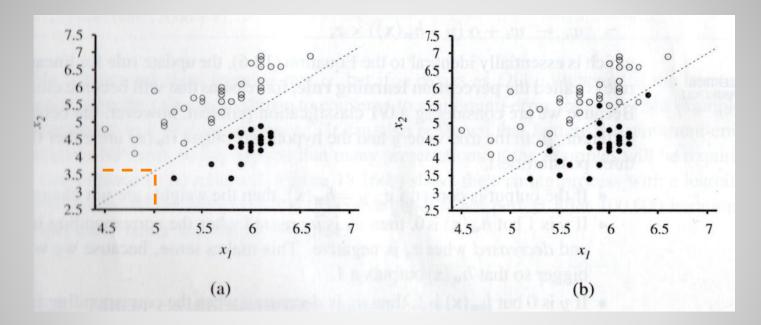
 A decision boundary is a line (or a surface, in higher dimensions) that separates the two classes.

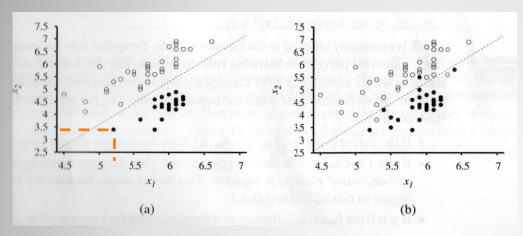
[5] Russell, S.J. and Norvig, P. (2010) Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall.



 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:

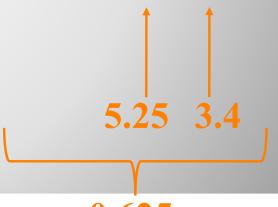
$$-4.9+1.7x_1-x_2=0$$



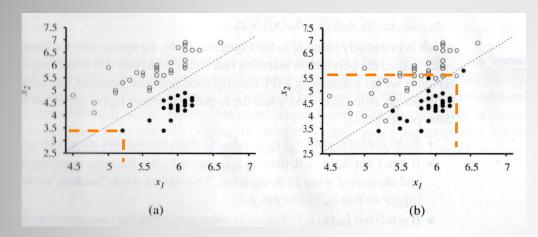


$$-4.9+1.7x_1-x_2=0$$

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



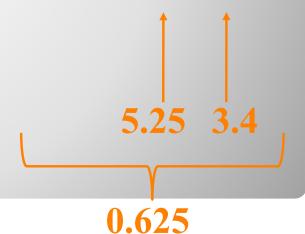
0.625

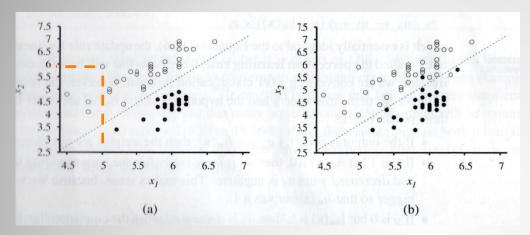


$$-4.9+1.7x_1-x_2=0$$

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

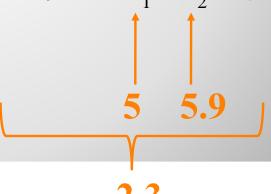
$$-4.9 + 1.7 (6.35) - 5.6 = 0.295$$
 $x_1 \quad x_2$

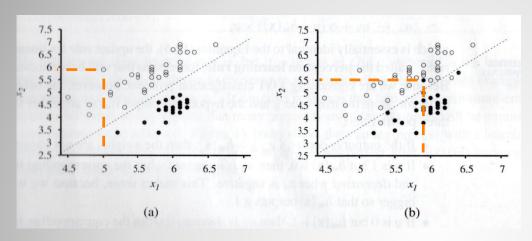




$$-4.9+1.7x_1-x_2=0$$

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



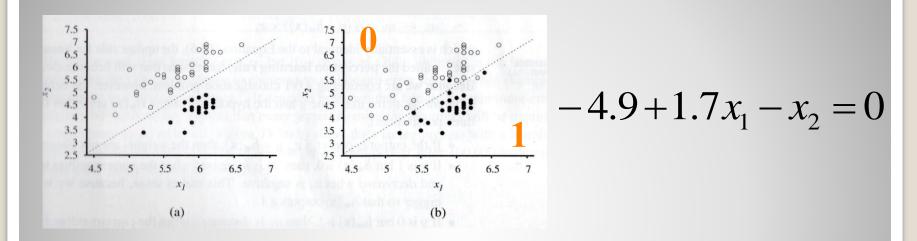


$$-4.9+1.7x_1-x_2=0$$

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

$$-4.9 + 1.7 (5.85) - 5.5 = -0.455$$

$$x_1 \quad x_2$$



- Nuclear explosions classification: $-4.9+1.7x_1-x_2>0$
- Earthquakes classification: $-4.9+1.7x_1-x_2<0$
- Given these training data, the classification task consists of learning a hypothesis that will take new (x_1,x_2) points and return either 0 for earthquakes or 1 for explosions.

• Using the convention of a dummy input $X_0=1$, we can write the classification hypothesis as:

$$h_{\mathbf{w}}(\mathbf{x}) = 1$$
 if $\mathbf{w} \cdot \mathbf{x} \ge 0$ and 0 otherwise

$$\mathbf{w} \cdot \mathbf{x} = (w_1, w_2, \dots, w_n) \cdot (x_1, x_2, \dots, x_n) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n$$

• Using the convention of a dummy input $X_0=1$, we can write the classification hypothesis as:

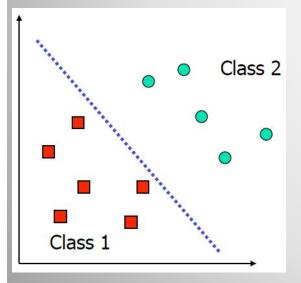
$$h_{\mathbf{w}}(\mathbf{x}) = 1$$
 if $\mathbf{w} \cdot \mathbf{x} \ge 0$ and 0 otherwise

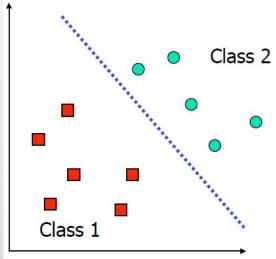
- Traditionally SVMs use the convention that class labels are 1 and -1, instead of the 1 and 0 we have been using so far.
- The linear separator is thus defined as:

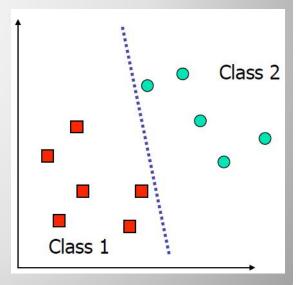
$$\{\mathbf{x}:\mathbf{w}\cdot\mathbf{x}+b=0\}$$

What is a good Decision Boundary?

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

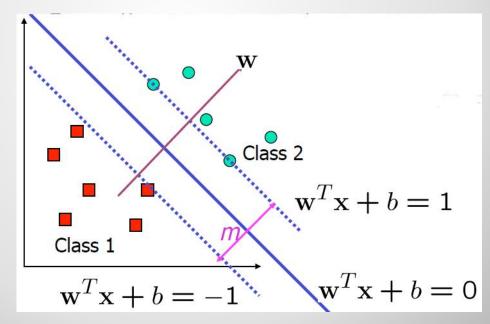






Large-margin Decision Boundary

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

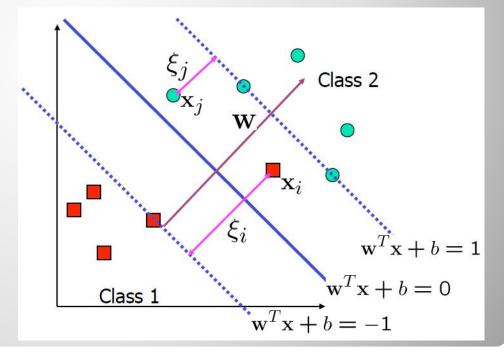


Outline

- 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
- Conclusions

Non-linearly Separable Problems

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



How do we find the decision boundary?

The optimization problem becomes [6] $\text{Minimize } \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^n \xi_i \\ \text{subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$

 C: tradeoff parameter between error and margin.

The Optimization Problem

The dual of this new constrained optimization problem is

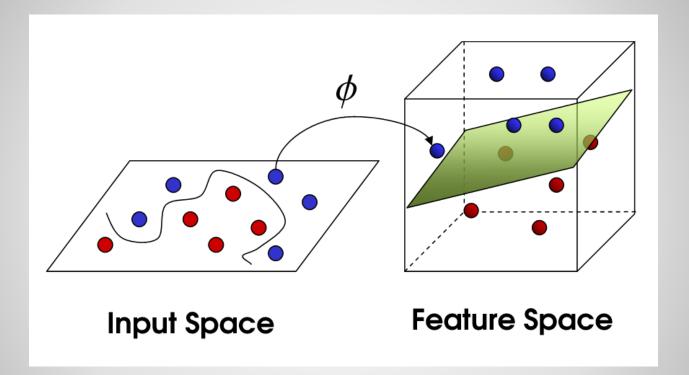
max.
$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$
 subject to $C \ge \alpha_i \ge 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$

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

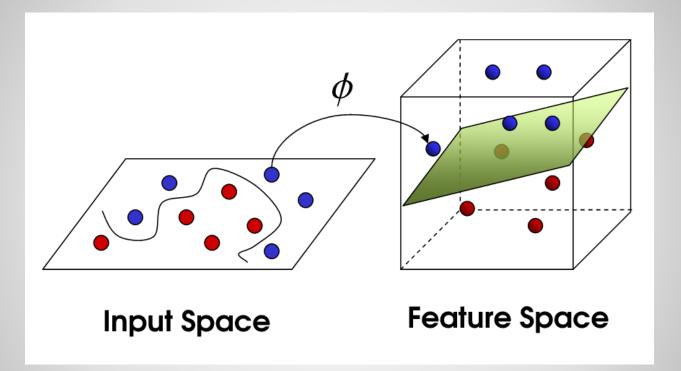
- So far, we have only considered largemargin classifier with a linear decision boundary.
- How to generalize it to become nonlinear?
- Key idea: transform x_i to a higher dimensional space to "make life easier".
 - Input space: the space the points x_i are located
 - Feature space: the space of $\phi(\mathbf{x}_i)$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

The Kernel Trick

Recall the SVM optimization problem

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

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

With kernel function

$$\begin{aligned} &\text{max. } W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{subject to } C \geq \alpha_i \geq \mathbf{0}, \sum_{i=1}^n \alpha_i y_i = \mathbf{0} \end{aligned}$$

Modification Due to Kernel Function

Original

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

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

With kernel function

max.
$$W(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^n \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$
 subject to $C \geq \alpha_i \geq 0, \sum_{i=1}^n \alpha_i y_i = 0$

Characteristics of the solution

- Many of the \(\alpha_i\) are zero
 - w is a linear combination of a small number of data points
- \mathbf{x}_{i} with non-zero α_{i} are called support vectors (SV)
 - The decision boundary is determined only by the SV
 - Let t_j (j=1, ..., s) be the indices of the s support vectors. We can write $\mathbf{w} = \sum_{j=1}^s \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$
- For testing with a new data z
 - Compute $\mathbf{w}^T\mathbf{z} + b = \sum_{j=1}^s \alpha_{t_j} y_{t_j}(\mathbf{x}_{t_j}^T\mathbf{z}) + b$ and classify \mathbf{z} as class 1 if the sum is positive, and class 2 otherwise

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
- Conclusions

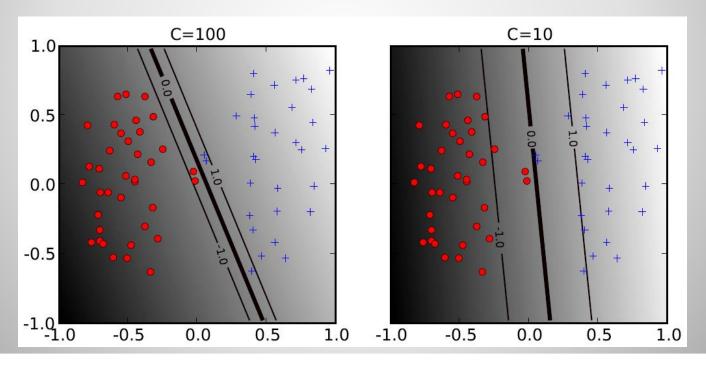
SVM four basic kernels

- 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, γ , r, and d are kernel parameters. [7]

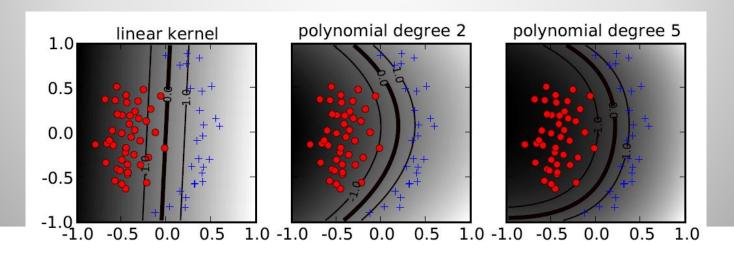
Linear kernel

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

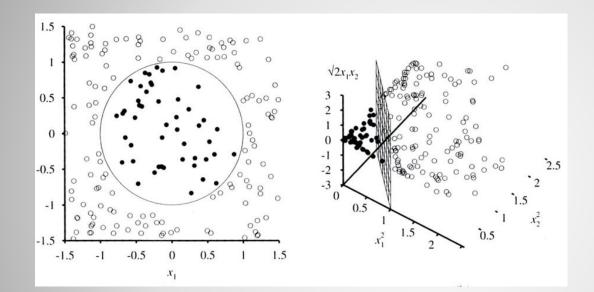


Polynomial kernel

- 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



$$f_1 = x_1^2$$

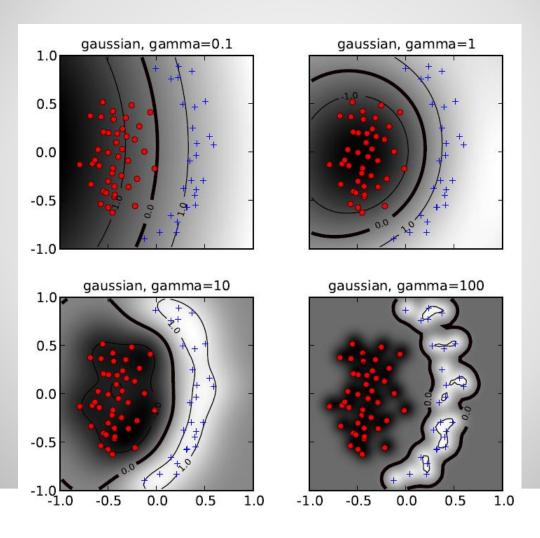
$$f_2 = x_2^2$$

$$f_3 = \sqrt{2}x_1 x_2$$

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



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.^[8]

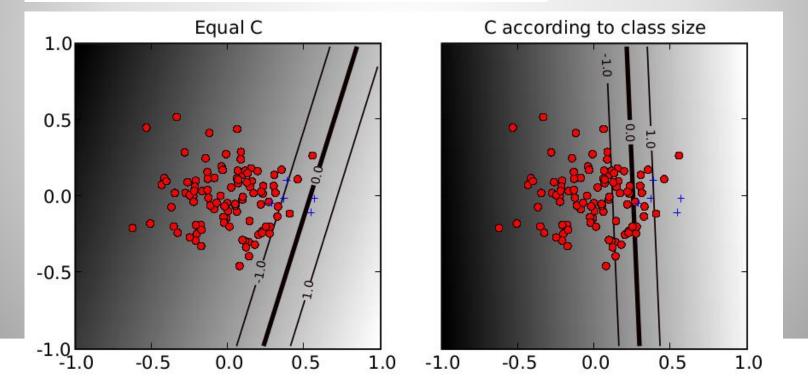
Outline

- 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
- Conclusions

SVMs for Unbalanced Data

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

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



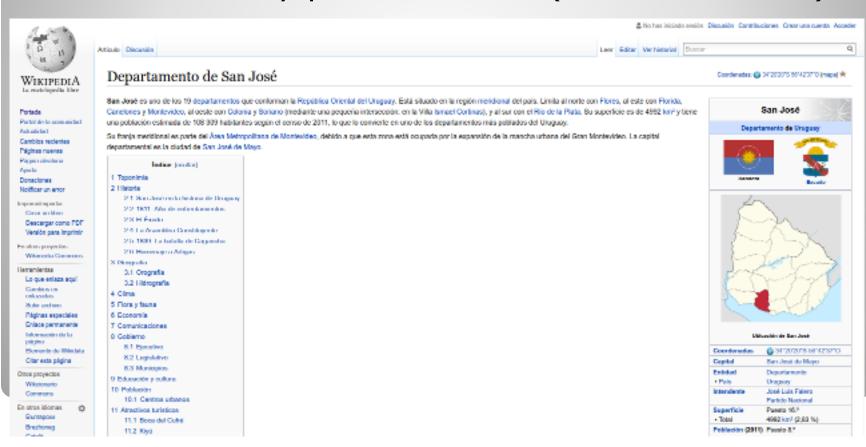
- One can change the QP formulation to become multi-class.^[9,10]
- More often, building binary classifiers which distinguish between:
 - one of the labels and the rest (one-versus-all).
 - between every pair of classes (one-versus-one).

[9] Ma, Y; Guo G. (Eds.) Support Vector Machines Applications. Chapter 2: "Multi-Class Support Vector Machine" Springer-Verlag (2014)

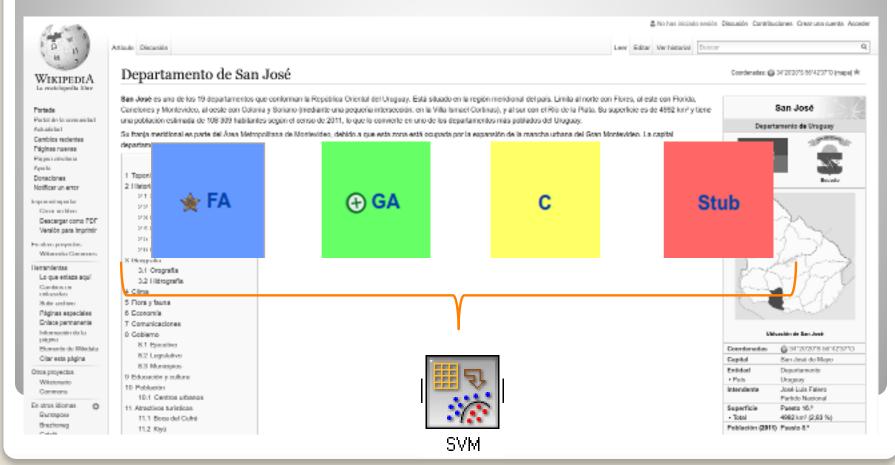
[10] Gerrit J.J. van den Burg, Patrick J.F. Groenen, GenSVM: A Generalized Multiclass Support Vector Machine, Journal of Machine Learning Research 17 (2016) 1-42

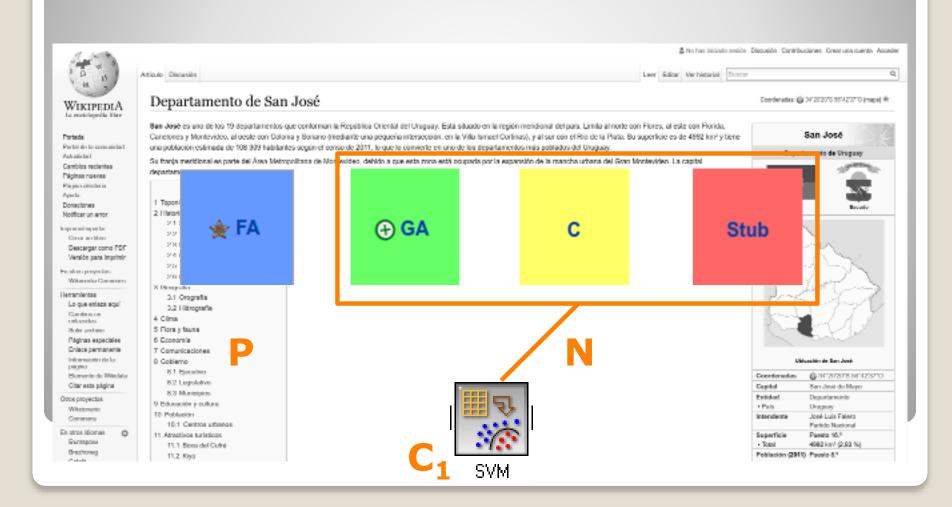
[11] M. Galar, A. Fernández, E. Barrenechea, H. Bustince and F. Herrera, An Overview of Ensemble Methods for Binary Classifiers in Multi-class Problems: Experimental Study on One-vs-One and One-vs-All Schemes. Pattern Recognition 44:8 (2011) 1761-1776. http://sci2s.ugr.es/ovo-ova

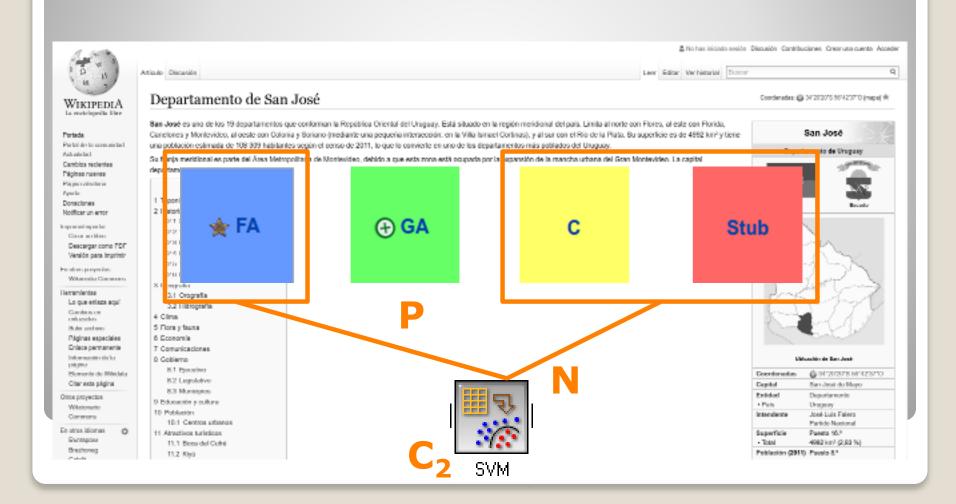
- One of the labels and the rest (one-versus-all)
- Between every pair of classes (one-versus-one)

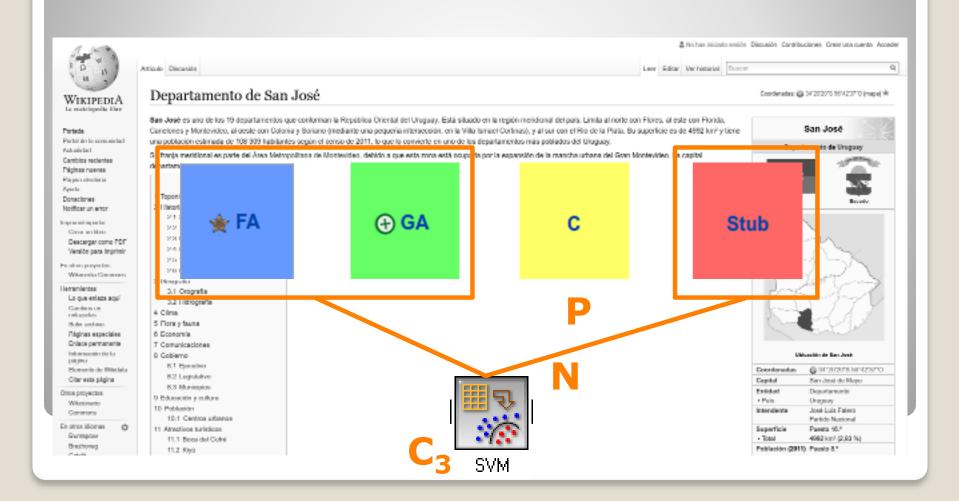


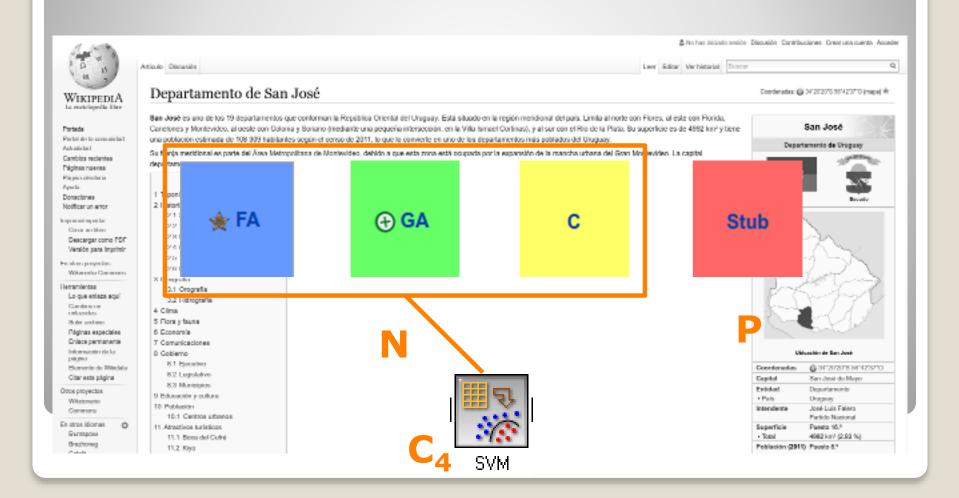
- One of the labels and the rest (one-versus-all)
- Between every pair of classes (one-versus-one)





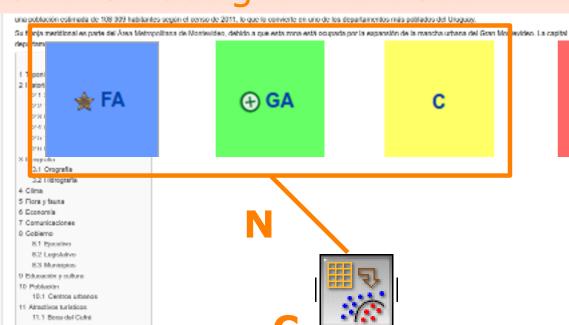






One of the labels and the rest (one-versus-all)

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



WIKIPEDL^A

Partal de la somunidad

Descenger come PDF Verside para Imprintr Facilities proyection Withmonto Germania Universidade

Lo-que enlaga soul

Páginas especiales Enlace permanente

Información de la

Oltar esta página

Otros proyectos

Wikelemento

Commission

En otros idiomas.

Suntagood

Brezhonea

Photography do 1000 labor

Combiners

Made and see

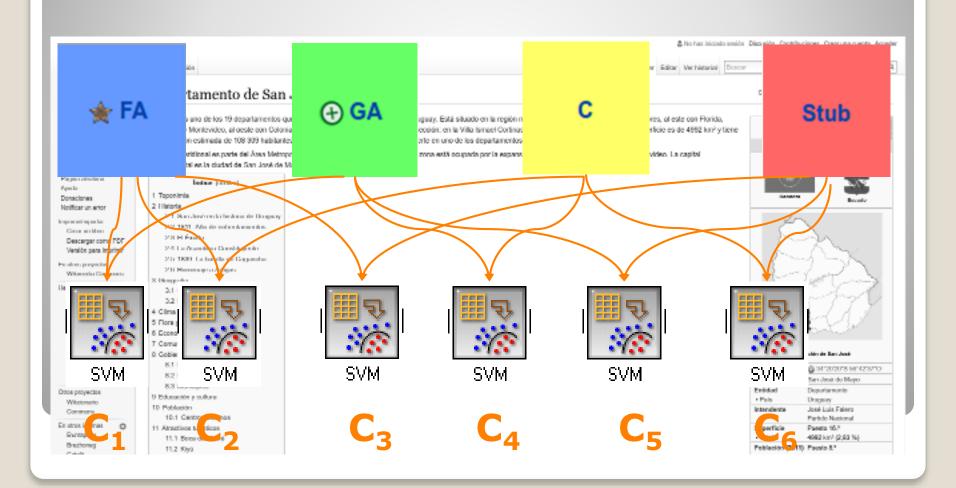
amb systems

Cambles reclentes Págines nuevas Págines struteria Aprela Donaciones Notificar un error

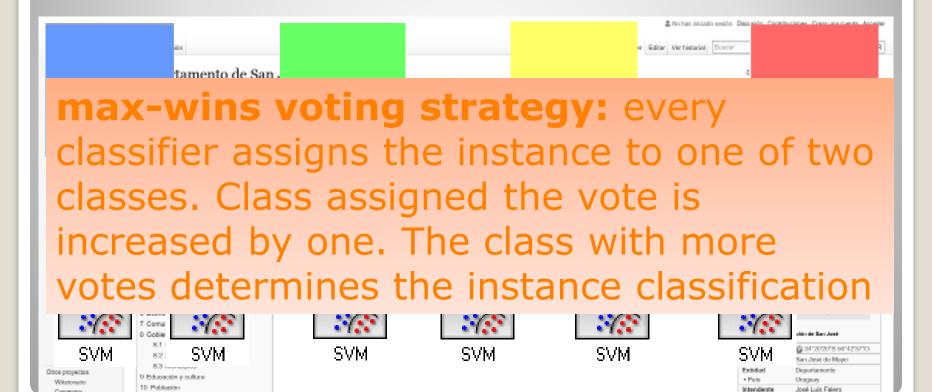


sousión Contribuciones Crear una quenta Acceder

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



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



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
- Conclusions

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
- https://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
- Conclusions

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. [7].
- 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).

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.