Mathematical and Quantative Methods

Using SVM for Classification

Dumitru Ciobanu[1]

Abstract: Support Vector Machines (SVMs)have found many applications in various fields. They have been introduced for classification problems and extended to regression. In this paperI review the utilization of SVM for classification problems and exemplify this with application on IRIS datasets. I used the Matlab programming language to implement linear and nonlinear classificators and apply this on the dataset.

Keywords: Support Vector Machines; optimal separating hyperplane; generalized separating hyperplane; nonlinear classifications; Iris dataset

JEL Classification: C02; C38; C45

1 Introduction

Support vector machines have a relatively short history being recently introduced, in the early 1990s. However, they are based on decades of research in computational learning theory done by Russian mathematicians Vladimir Vapnik and Alexey Chervonenkis. This theory, presented in the book of Vapnik from 1982 *Estimation of Dependences Based on Empirical Data*, was called Vapnik-Chervonenkis theory or simply VC theory (Vapnik, 2006). This book describes the implementation of support vector machines for linearly separable data (Cortes & Vapnik, 1995). A number of important extensions were made to the SVM. In 1992, Boser, Guyon and Vapnik proposed the use of kernel trick of Aizerman's to classify data separable using polynomial functions or radial basis functions. In 1995, Cortes and Vapnik extended the theory so that it can be applied for the training data inseparable, using a cost function. Later, in 1996 (Drucker, 1996), was developed a method for regression based on support vector.

It should be noted that there are many different algorithms for SVMs like SVM Lagrangian (LSVM), Lagrangian finite Newton SVM (NLSVM) or finite Newton SVM (NSVM), a comparison between different methods is shown in (Shu-Xia Lu, 2004).

[1] PhD Student, University of Craiova, Romania, Address: A. I. Cuza, no. 13, Craiova, 200585, Romania, Tel.: +40 251/411593, Fax: +40 251/411317, Corresponding author: ciobanubebedumitru@yahoo.com.

A Support Vector Machine (SVM) is a machine learning that can be used in classification problems (Cortes & Vapnik, 1995) and regression problems (Smola, 1996).

In order to perform classification, SVMs seek an optimal hyperplane that separates data into two classes. In Figure 1are presented some possibilities of linear separation of two sets of elements.

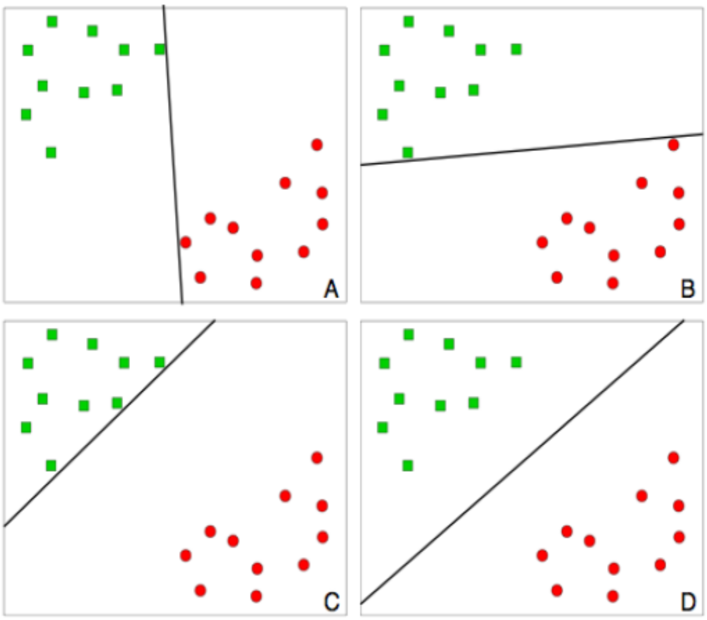

Figure 1. Different variants of linear separation of two sets

(Guggenberger, 2008)

Support vector machine are also called classifiers with maximum edge. This means that the resulted hyperplane maximizes the distance between the closest vectors from different classes taking into account the fact that a greater margin provides increased SVM generalization capability.

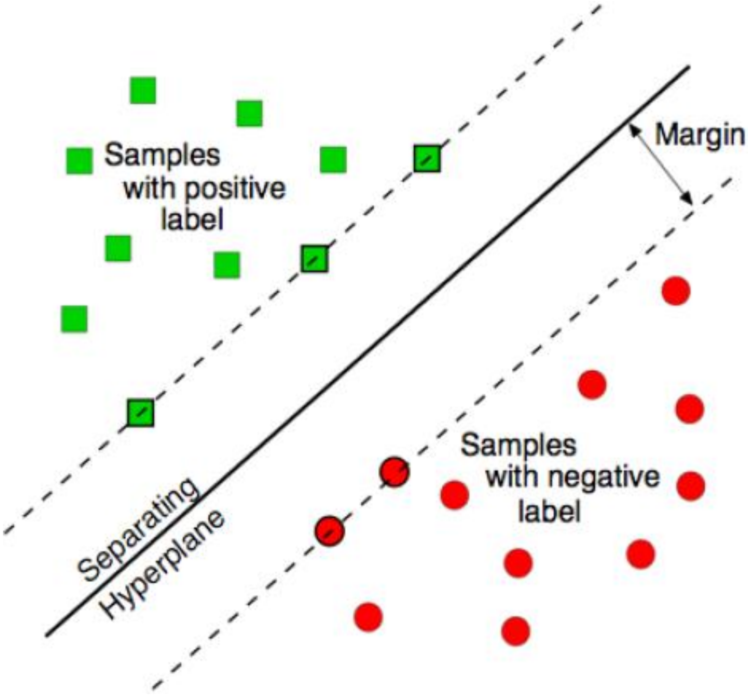

Figure 2. Optimal separating hyperplane. The vectors on dotted lines are support vectors

(Guggenberger, 2008)

The elements closest to the optimal separating hyperplane are called support vectors and only they are considered by the SVMs for the classification task. All other vectors are ignored.

2. Optimal Separating Hyperplane

The basic problem that SVM learns and solves is that of classification in two categories of a data set.

Classification problem implies a set of observations represented as pairs (x_i, y_i), i = 1, …, r, where $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$. Each observation contains an n-dimensional vector and an associated class. The aim is to determine the optimal separation hyperplane, that is the hypersurface (n-1)-dimensional, which best separates the two classes Figure 3.

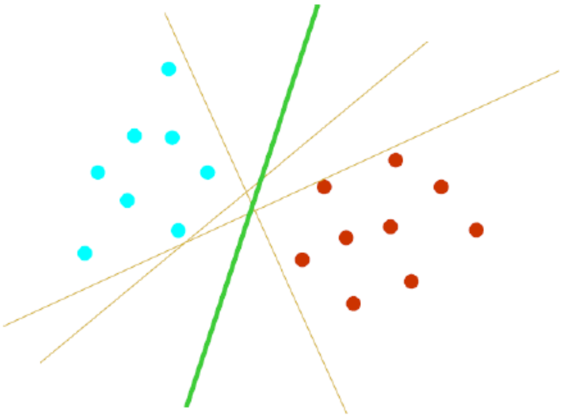

Figure 3. Optimal Separating Hyperplane

(Gunn, 1998)

The simplest situation is that there exist a hyperplane defined by a normal vector *w*, which separates the classes,

$$\langle w, x \rangle + b = 0 \tag{1}$$

Because this hyperplane is invariant to scalar multiplication, we can choose *w* and *b* so as to meet the requirement

$$\min_i \left| \langle w, x_i \rangle + b \right| = 1 \tag{2}$$

Constraint in equation (2) tells us that the norm of weight vector *w* must be equal to the inverse distance from the nearest point of the dataset to hyperplane.

Also, the equation (2) leads to a breakdown of points in two categories.

$$\langle w, x_i \rangle + b \geq 1 \tag{3}$$

$$\langle w, x_i \rangle + b \leq -1 \tag{4}$$

Assuming that the first category corresponds to points labeled 1 and the second category to points labeled -1, the two inequalities are rewritten as

$$y_i \left[\langle w, x_i \rangle + b \right] \geq 1,\ i = 1, \ldots, r\,. \tag{5}$$

$\langle w, x_i \rangle + b = 1and\langle w, x_i \rangle + b = -1$ are two hyperplans parallel with separating hyperplane. This is represented in Fig. 2., where the separating hyperplane is represented by a solid line and those two parallel hyperplans by dotted lines. Dotted lines contain some of the training points. These points are called support vectors and completely determines the solution of classification problem. The distance between the dotted lines is called margin and is to be maximized.

The margin is $\rho(w,b)=\frac{2}{\|w\|}$ and the maximization of margin is equivalent with maximization of the function

$$L(w)=\frac{1}{2}\|w\|^2 \tag{6}$$

with constraints (5).

The solution to optimization problem (6) with constraints (5) is given by the saddle point of Lagrange functional (Minoux, 1986),

$$L(w,b,\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{r}\alpha_i\left(y_i\left[\left\langle w,x_i\right\rangle+b\right]-1\right) \tag{7}$$

were α is the vector of the Lagrange multipliers.

The Lagrangian must be minimized in rapport with *w*, *b* and maximized in function of $\alpha \geq 0$. Classic theory of Lagrange duality allow us to transform the primal problem (7) in the dual problem, which is easier to solve. The dual problem has the form,

$$\max_{\alpha} W(\alpha)=\max_{\alpha}\left(\min_{w,b} L(w,b,\alpha)\right) \tag{8}$$

That is

$$\max_{\alpha} W(\alpha)=\max_{\alpha}\left(-\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j y_i y_j\left\langle x_i,x_j\right\rangle+\sum_{k=1}^{r}\alpha_k\right) \tag{9}$$

and the solution is given by

$$\alpha^*=\arg\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j y_i y_j\left\langle x_i,x_j\right\rangle-\sum_{k=1}^{r}\alpha_k\right) \tag{10}$$

with constraints

$$\alpha_i \geq 0,\ i = 1,\ldots,r$$
$$\sum_{i=1}^{r} \alpha_i y_i = 0 \tag{11}$$

Solving equation (10) with constraints (11) is determined Lagrange multipliers and optimal separating hyperplane, given by

$$w^* = \sum_{i=1}^{r} \alpha_i y_i x_i$$
$$b^* = -\frac{1}{2}\left\langle w^*, x_k + x_s \right\rangle, \tag{12}$$

where x_k and x_s are any of support vectors coming from the two classes, that satisfy relations $\alpha_k, \alpha_s > 0$ and $y_k = -1, y_s = 1$.

Then, the hard classifier (inflexible edges)

$$f(x) = \operatorname{sgn}\left(\left\langle w^*, x\right\rangle + b^*\right). \tag{13}$$

From Kuhn-Tucker conditions,

$$\alpha_i\left(y_i\left[\left\langle w, x_i\right\rangle + b\right] - 1\right) = 0,\ i = 1,\ldots,r\,, \tag{14}$$

result that only points x_i that satisfy

$$y_i\left[\left\langle w, x_i\right\rangle + b\right] = 1, \tag{15}$$

will have nonzero Lagrange multipliers. These points are called support vectors (SV). If the data is linearly separable all support vectors will be on edge and their number can be very small. Consequently, the hyperplane is determined by a small subset of the training set. Eliminating from the training set points that are not support vectors and recalculate the optimal separating hyperplane will achieve the same result. Thus, support vector machines (SVM) are used to summarize information contained in the training data using support vector.

3. Generalised Optimal Separating Hyperplane

Most times the data provided for classification are not linearly separable. One way to perform classification in such cases is generalized optimal separating hyperplane. It separates linear data supporting classification errors. In Fig. 4. we have an intuitive graphical representation of generalized optimal separating hyperplane.

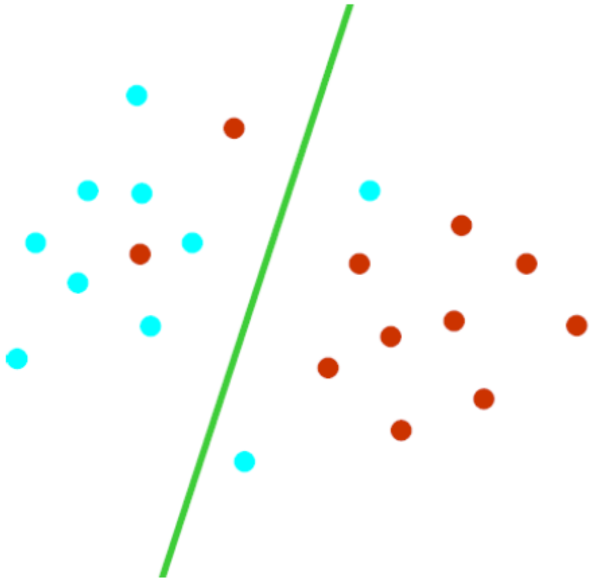

Figure 4. Generalized Optimal Separating Hyperplane

(Gunn, 1998)

Cortes and Vapnik introduced variables $\xi_i \geq 0$ that mesures the classification errors (Cortes & Vapnik, 1995).

In these conditions, the optimization problem will minimize classification errors. Constraints for the inseparable case will be of the form

$$y_i \left[\left\langle w, x_i \right\rangle + b \right] \geq 1 - \xi_i,\ i = 1, \ldots, r\ . \qquad (16)$$

where $\xi_i \geq 0$.

Generalized optimal separating hyperplane is determined by the vector w that minimize the functional

$$L(w,\ \xi) = \frac{1}{2}\left\| w \right\|^2 + C\sum_{i=1}^{r} \xi_i \qquad (17)$$

with constraints (16), where C is a given constant.

The solution of minimization of the functional (17) with constraints (16) is given by the saddle point of the following Lagrangian (Minoux, 1986),

$$L(w,b,\alpha,\xi,\beta)=\frac{1}{2}\|w\|^2+C\sum_{i=1}^{r}\xi_i-\sum_{i=1}^{r}\alpha_i\left(y_i\left[\left\langle w,x_i\right\rangle+b\right]-1+\xi_i\right)-\sum_{i=1}^{r}\beta_i\xi_i\,, \quad (18)$$

where α and β are the Lagrange multipliers. The Lagrangian must minimized about *w, b, x* and maximized about α, β. To solve this optimization problem is recalled, as in the classical case at the dual problem

$$\max_{\alpha,\beta} W(\alpha,\beta)=\max_{\alpha,\beta}\left(\min_{w,b,\xi} L(w,b,\alpha,\xi,\beta)\right) \quad (19)$$

Explicitly, the dual problem is written

$$\max_{\alpha} W(\alpha)=\max_{\alpha}\left(-\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j y_i y_j\left\langle x_i,x_j\right\rangle+\sum_{k=1}^{r}\alpha_k\right) \quad (20)$$

and the solution is

$$\alpha^*=\arg\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j y_i y_j\left\langle x_i,x_j\right\rangle-\sum_{k=1}^{r}\alpha_k\right) \quad (21)$$

with constraints

$$\begin{aligned}&0\le\alpha_i\le C,\ i=1,\ldots,r\\&\sum_{i=1}^{r}\alpha_i y_i=0\end{aligned} \quad (22)$$

The solution of minimization problem in the case of linearly inseparable data is identical to those from data linearly separable case except the bounds of Lagrange multipliers. Yet, there was an additional problem, namely determining the coefficient *C*. This parameter offers new possibilities to control over the classifier. Blanz and collaborators have used the value C = 5 (Blanz et al, 1996), other researchers regard *C* as directly related to a regularization parameter (Smola & Scholkopf, 1998), but eventually *C* must be chosen so that to reflect the knowledge of noise from data (Gunn, 1998).

4. Generalization in Multidimensional Feature Space

Another approach to separate two classes is to transfer, using a nonlinear applications, the input space into a feature space with higher dimension in which data can be separated using optimal separating hyperplane Fig. 5.

The idea is based on the method introduced by Aizerman and colleagues (Aizerman, Braverman & Rozonoer, 1964) which eliminates problems arising from increasing the dimension (Bellman, 1961).

Nonlinear functions that can be used must meet certain conditions, known as Mercer conditions. Among the most used functions that satisfy these requirements we mention the polynomial, the base radial and sigmoidal functions.

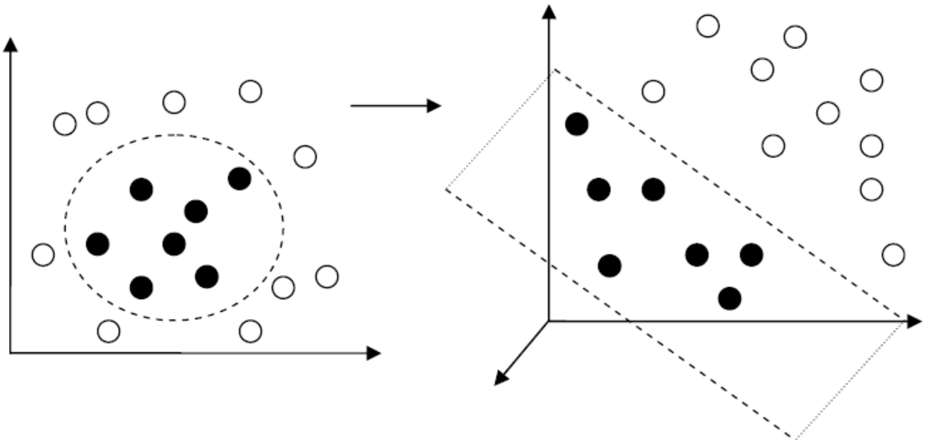

Figure 5. Using a higher dimension space for the linear separation of data

(Lovell & Walder, 2006)

The optimization problem in this case, can be written

$$\alpha^* = \arg\min_{\alpha}\left(\frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{r}\alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right) - \sum_{k=1}^{r}\alpha_k\right), \tag{23}$$

where $K(\cdot,\cdot)$ is the kernel function that performs nonlinear translation of input space to feature space and the constraints are the same as for generalized linear case

$$\begin{aligned} &0 \le \alpha_i \le C,\ i = 1,\dots,r \\ &\sum_{i=1}^{r} \alpha_i y_i = 0. \end{aligned} \tag{24}$$

It solves the optimization problem (23) with the restrictions (24) and determine the Lagrange multipliers. With this is build a hard classifier in feature space

$$f(x) = \operatorname{sgn}\left(\sum_{x_i \in SV} \alpha_i^* y_i K(x_i, x) + b^* \right), \tag{25}$$

where

$$\begin{aligned} &\langle w^*, x \rangle = \sum_{x_i \in SV} \alpha_i^* y_i K(x_i, x) \\ &b^* = -\frac{1}{2} \sum_{x_i \in SV} \alpha_i^* y_i \left[K(x_i, x_k) + K(x_i, x_s) \right] \end{aligned} \tag{26}$$

with x_k and x_s any of the support vectors coming from the two classes.

5. Case Study Iris Dataset

For exemplification of SVM classification we use Iris data set (Fig. 6.). It consists of 150 observations, 50 Iris setosa, 50 Iris versicolor and Iris virginica 50, with 4 characteristics: length of sepals, sepals width, length of petals and petals width.

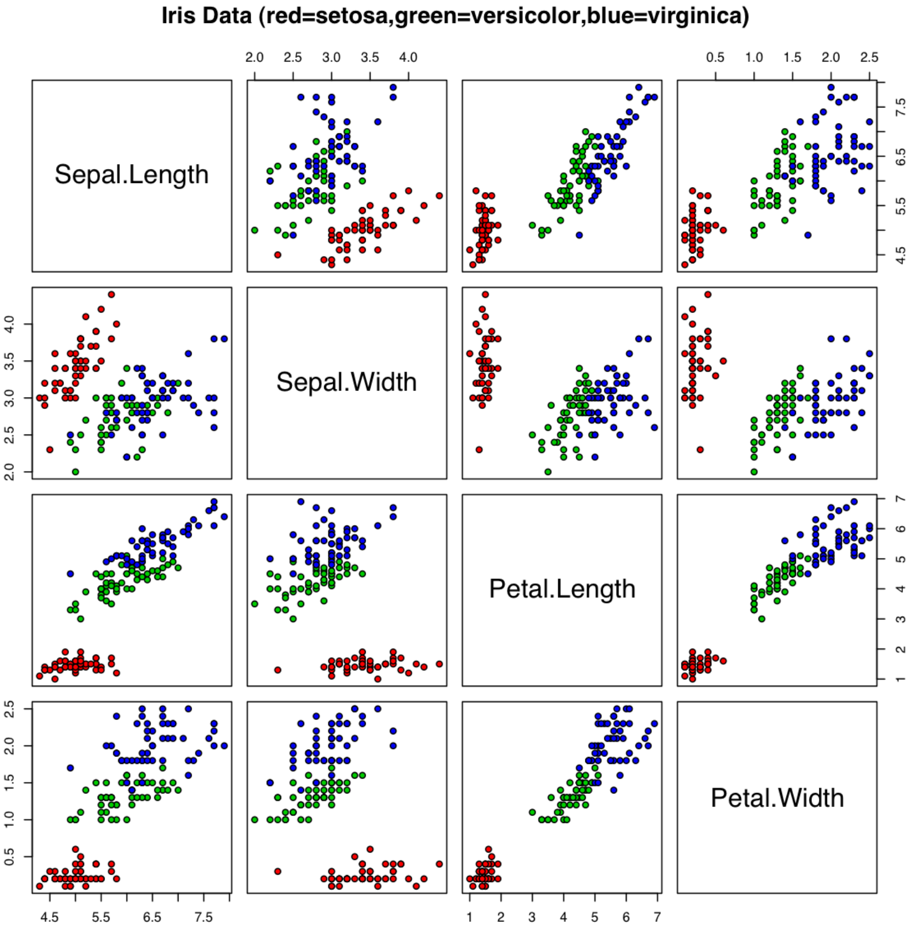

Figure 6. Representations of Iris data set based on pairs of two features
(http://en. wikipedia. org/wiki/File:Anderson%27s_Iris_data_set. png, accessed in 2012)

Iris data set has been extensively used for exemplification of classification and grouping methods because in binary representations have both linearly separable classes (iris setosa - iris versicolor and iris setosa - iris virginica) and classes that are not linearly separable (iris virginica - iris versicolor).

For exemplification of different methods of classification we use the graphic representation sepals length versus petals length.

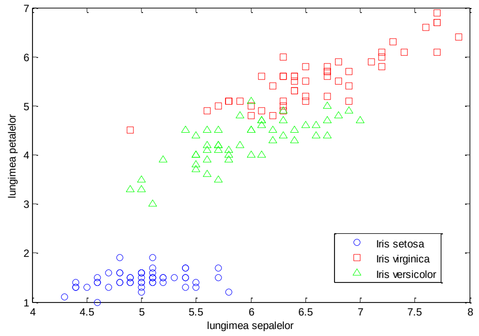

Figure 7. The representation sepals length versus petals length of iris dataset

For linear separation we use classes Iris setosa and Iris versicolor (Fig. 8.) And for nonlinear classification we exemplify using classes iris virginica and iris versicolor (Fig. 10.).

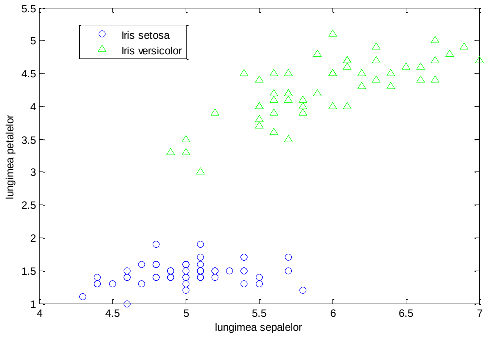

Figure 8. Iris versicolor and iris setosa according to the length of sepals and the length of petals

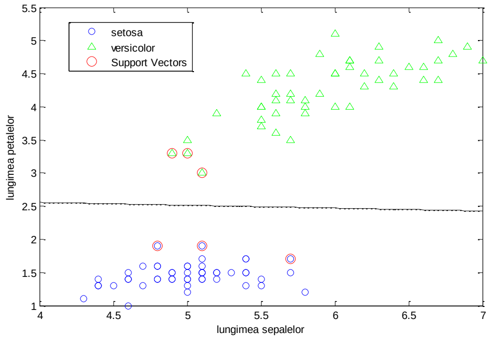

Figure 9. Linear separation of classes iris setosa and iris versicolor with highlighting of support vectors

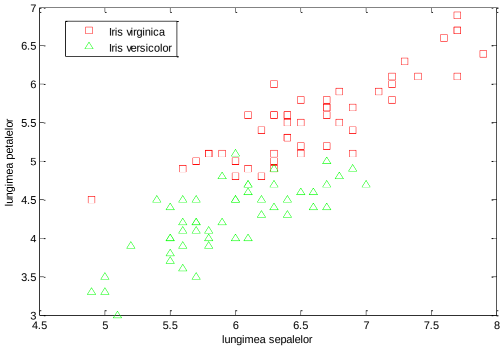

Figure 10. Iris versicolor and iris virginica according to the length of sepals and the length of petals

In this case we note that data are not linearly separable, so we use linear classifier with flexible edges and nonlinear classifiers.

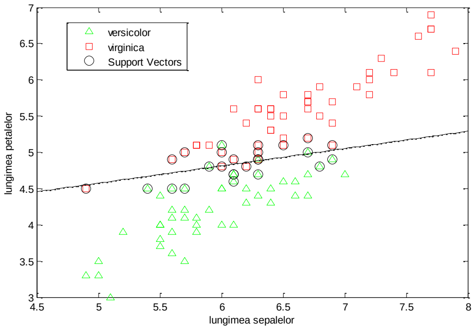

Figure 11. Linear separation with flexible edges of classes iris virginica and iris versicolor with highlighting of support vectors

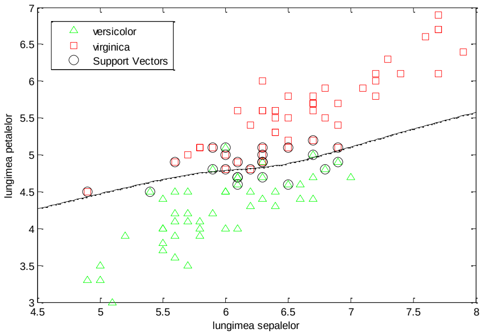

Figure 12. Nonlinear separation using a polynomial kernel of classes iris virginica and iris versicolor with highlighting of support vectors

Figure 12. represents a case of using a polynomial kernel of order 3 and Fig 13. a situation encountered for a kernel of type radial basis function.

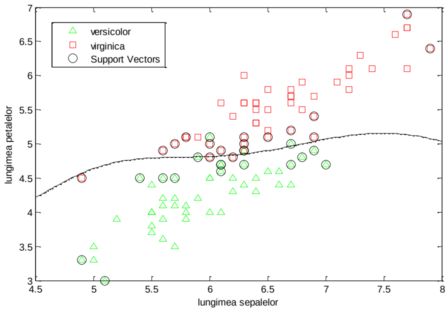

Figure 13. Nonlinear separation using a kernel of type radial basis function of classes iris virginica and iris versicolor with highlighting of support vectors

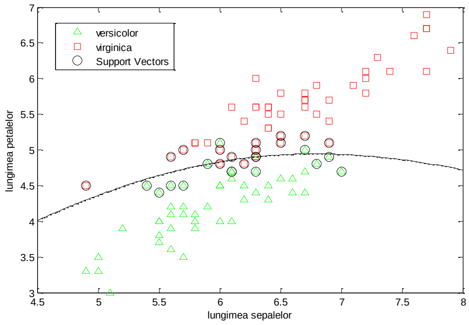

Figure 14. Nonlinear separation using a kernel of type quadratic function of classes iris virginica and iris versicolor with highlighting of support vectors

For a quadratic kernel In the case of a kernel of multilinear perceptron type.

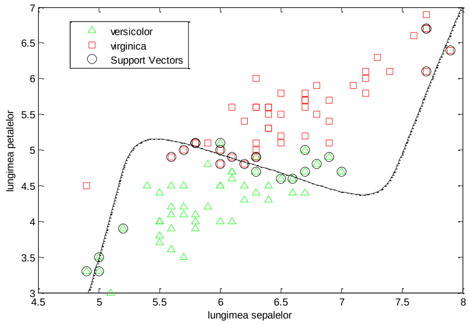

Figure 15. Nonlinear separation using a kernel of type multilinear perceptron of classes iris virginica and iris versicolor with highlighting of support vectors

5. Conclusion

SVM is one of the most promising algorithms in machine learning field and there are many examples in which SVMs are successfully used, for example, text classification, face recognition, character recognition (OCR - Optical Character Recognition), Bioinformatics. On these datasets SVMs apply very well and often exceeds the performance of other traditional techniques. Of course, this is not a magic solution as set forth in (Bennett & Campbell, 2000), there are still some open issues, such as incorporation of domain knowledge, a new model selection and interpretation of results produced by SVMs.

SVMs have been used in several real-world problems:

- classification of text (and hypertext);
- image classification;
- in bioinformatics (protein classification, classification of types of cancer);
- classification of music;
- handwritten character recognition.

In (Chen, Jeong & Hardie, 2008), the authors propose a method GARCH (Generalized AutoRegressive Conditional Heteroscedasticity) based on recurrent SVR whose performance exceeds other approaches such as moving average, recurrent neural networks and parameterized GARCH in terms of their ability to predict the financial market volatility. Important aspect that recommend the use of SVM we mention the absence of local minima, control solution capacity (Christiani & Shawe-Taylor, 2000) and the ability to effectively use multidimensional data (Cortes & Vapnik, 1995).

Strengths of SVM:

- Training is relatively easy to achieve;
- No local optimal, unlike neural networks;
- Suitable for multidimensional data relatively well;
- Non-traditional data such as strings and trees can be used as input to SVM, instead of feature vectors;
- The compromise between complexity and classification error can be controlled explicitly;
- By performing logistic regression (sigmoidal) with SVM on a set of output data, SVM can be interpreted in terms of probability.

Weaknesses of SVM:

- It needs a good choice for kernel function;
- Training takes a long time.

In graphic representations can see the small number of support vectors, basically those who are using the classifier. Due to the small number of support vector classification of new cases require scarce resources of time and computing power.

The best classification for linearly inseparable case, were obtained for polynomial and radial basis kernels which underlines once again the importance of a correct choice for the kernel function used.

6. References

Aizerman, M. A.; Braverman, E. M. & Rozonoer, L.I. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, Vol. 25, pp. 821–837.

Bellman, R. E. (1961). *Adaptive Control Processes*. Princeton, NJ: Princeton University Press.

Bennett, K. P.; Campbell, C. (2000). Support vector machines: hype or hallelujah? *SIGKDD Explorations Newsl.*, Vol 2, No. 2, pp. 1–13.

Boser, B.; Guyon, I.; Vapnik, V. (1992). A training algorithm for optimal margin classifiers. *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pp. 144–52.

Blanz, V.; Schölkopf, B.; Bülthoff, H.; Burges, C.; Vapnik, V. & Vetter, T. (1996). *Comparison of view-based object recognition algorithms using realistic 3D models*, In: C. von der Malsburg, W. von Seelen, J. C. Vorbrüggen, and B. Sendhoff (eds.): *Artificial Neural Networks - ICANN'96*. Springer Lecture Notes in Computer Science Vol. 1112, Berlin, pp. 251-256.

Chen, S.; Jeong, K.; Härdle, W. (2008). *Support Vector Regression Based GARCH Model with Application to Forecasting Volatility of Financial Returns*, SFB 649 "Economic Risk", Humboldt-Universität zu Berlin, Berlin. Available online at http://edoc.hu-berlin.de/series/sfb-649-papers/2008-14/PDF/14.pdf.

Christiani, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*. Cambridge: Cambridge University Press.

Cortes, C. & Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20(3), pp. 273-297.

Drucker, H.; Burges, C.; Kaufman, L.; Smola, A. & Vapnik, V. (1996). *Support vector regression machine*. Advances in Neural Information Processing Systems, Cambridge: MIT Press 9(9): 155–61.

Guggenberger, A. (2008). *Another Introduction to Support Vector Machines*, Available online at http://mindthegap.googlecode.com/files/AnotherIntroductionSVM.pdf, accessed May 2012.

Gunn, S. R. (1998). *Support Vector Machines for Classification and Regression*, University of Southampton, Available online at http://www.svms.org/tutorials/Gunn1998.pdf, accessed April 2012.

Lovell, B. C.; Walder, C. J. (2006). Support Vector Machines for Business Applications. *Business Applications and Computational Intelligence*. Hershey, U.S.A: Idea Group, pp. 267-290.

Minoux, M. (1986). *Mathematical Programming: Theory and Algorithms*. John Wiley and Sons.

Smola, J. (1996). Regression estimation with support vector learning machine. *Master's thesis*. Munchen: Technische Universitat Munchen.

Smola, A.; Schölkopf, B. (1998). On a Kernel-based Method for Pattern Recognition, Regression, Approximation and Operator Inversion. *GMD Technical Report* No. 1064.

Shu-Xia Lu, X. -Z. W. (2004). A comparison among four SVM classification methods: Lsvm, nlsvm, ssvm and nsvm. *Proceedings of 2004 International Conference on Machine Learning and Cybernetics*, vol. 7, pp. 4277–4282, Shanghai, China.

Vapnik, V. (2006). *Empirical Inference Science*. Afterword in 1982 reprint of Estimation of Dependences Based on Empirical Data.