Kernel Methods and Visualization for Interval Data Mining

Thanh-Nghi Do[1] and François Poulet[2]

[1] College of Information Technology,
Can Tho University,
1 Ly Tu Trong Street, Can Tho, VietNam
(e-mail: dtnghi@cit.ctu.edu.vn)
[2] ESIEA Pôle ECD,
BP 0339, 53003 Laval-France
(e-mail: poulet@esiea-ouest.fr)

Abstract. We propose to use kernel methods and visualization tool for mining interval data. When large datasets are aggregated into smaller data sizes we need more complex data tables e.g interval type instead of standard ones. Our investigation aims at extending kernel methods to interval data analysis and using graphical tools to explain the obtained results. The user deeply understands the models' behaviour towards data. The numerical test results are obtained on real and artificial datasets.
Keywords: Kernel methods, Support vector machines, Visualization, Interval data, Data mining, Visual data mining.

1 Introduction

In recent years, real-world databases have increased rapidly, so that the need to extract knowledge from very large databases is increasing. Data mining can be defined as the particular pattern recognition task in the knowledge discovery in databases process. It uses different algorithms for classification, regression, clustering or association. The SVM algorithms proposed by [Vapnik, 1995] are a well-known class of algorithms using the idea of kernel substitution. They have shown practical relevance for classification, regression and novelty detection tasks. The successful applications of SVM and other kernel-based methods [Cristianini and Shawe-Taylor, 2000], [Shawe-Taylor and Cristianini, 2004] have been reported for various fields.

While SVM and kernel-based methods are a powerful paradigm, they are not favourable to deal with the challenge of large datasets. The learning task is accomplished through the quadratic program possessing a global solution. Therefore, the computational cost of an kernel approach is at least square of the number of training data points and the memory requirement makes them intractable. We propose to scale up their training tasks based on the interval data concept [Bock and Diday, 1999]. We summarize the massive datasets into the interval data. We adapt the kernel algorithms to deal with

this data. We construct a new RBF kernel of interval data used for classification, regression and novelty detection tasks. The numerical test results are obtained on real and artificial datasets.

Although SVM gives good results, the interpretation of these results is not so easy. The support vectors found by the algorithms provide limited information. Most of the time, the user only obtains information regarding support vectors and accuracy. He can not explain or understand why a model constructed by SVM makes a good prediction. Understanding the model obtained by the algorithm is as important as the accuracy because the user has a good comprehension of the knowledge discovered and more confidence in this knowledge. Our investigation aims at using visualization methods to try to explain the SVM results. We use interactive graphical decision tree algorithms and visualization techniques to give an insight into classification, regression and novelty detection tasks with SVM. We illustrate how to combine some strengths of different visualization methods to help the user to improve the comprehensibility of SVM results.

This paper is organized as follows. In section 2, we present a new Gaussian kernel construction to deal with interval data. In section 3, we briefly introduce classification, regression and novelty detection of interval data with SVM algorithms and other kernel-based methods. Section 4 presents a way to explain SVM results by using interactive decision tree algorithms. We propose to use an approach based on different visualization methods to try to interpret SVM results in section 5 before the conclusion and future works in section 6.

2 Non linear kernel function for interval data

Assume we have two data points x and $y \in R^n$. Here, we are interested in RBF kernel function because it is general and efficient. The RBF kernel formula in (1) of two data vectors x and y of continuous type is based on the Euclidean distance between these vectors, $d_E(x, y) =\| x - y \|$.

$$K\langle x, y\rangle = \exp\left(-\frac{\| x - y \|^2}{\gamma}\right) \tag{1}$$

For dealing with interval data, we only need to measure the distance between two vectors of interval type, then we substitute this distance measure for the Euclidean distance into RBF kernel formula. Thus the new RBF kernel can deal with interval data. The dissimilarity measure between two data vectors of interval type is the Hausdorff distance.
Suppose that we have two intervals represented by low and high values: $I_1 = [low_1, high_1]$ and $I_2 = [low_2, high_2]$, the Hausdorff distance between two intervals I_1 and I_2 is defined by (2):

$$d_H(I_1, I_2) = \max\left(|low_1 - low_2|, |high_1 - high_2|\right) \tag{2}$$

Let us consider two data vectors u, $v \in \Omega$ having n dimensions of interval type:

$u = ([u_{1,low}, u_{1,high}], [u_{2,low}, u_{2,high}], \ldots, [u_{n,low}, u_{n,high}])$

$v = ([v_{1,low}, v_{1,high}], [v_{2,low}, v_{2,high}], \ldots, [v_{n,low}, v_{n,high}])$

The Hausdorff distance between two vectors u and v is defined by (3):

$$d_H(u,v) = \sqrt{\sum_{i=1}^{n} \max\left(|u_{i,low} - v_{i,low}|^2, |u_{i,high} - v_{i,high}|^2\right)} \tag{3}$$

By substituting the Hausdorff distance measure d_H into RBF kernel formula, we obtain a new RBF kernel for dealing with interval data. This modification tremendously changes kernel algorithms for mining interval data. No algorithmic changes are required from the habitual case of continuous data other than the modification of the RBF kernel evaluation. All the benefits of the original kernel methods are kept. The kernel-based learning algorithms like Support Vector Machines (SVM [Vapnik, 1995]), Kernel Fisher's Discriminant Analysis (KFDA [Mika *et al.*, 1999]), Kernel Principal Component Analysis (KPCA [Schölkopf *et al.*, 1998]), Kernel Partial Least Squares (KPLS [Rosipal and Trejo, 2001]) can use the RBF function to build interval data models in classification, regression and novelty detection.

3 Interval data analysis with kernel methods

3.1 Support vector machines

$$\begin{gathered} \min\,(1/2)\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j K\langle x_i, x_j\rangle - \sum_{i=1}^{m}\alpha_i \\ s.t. \sum_{i=1}^{m} y_i \alpha_i = 0 \\ C \geq \alpha_i \geq 0 \end{gathered} \tag{4}$$

where C is a positive constant used to tune the margin and the error.

Let us consider a binary linear classification task with m data points in a n-dimensional input $x_1, x_2, \ldots, x_m$ having corresponding labels $y_i = \pm 1$. SVM classification algorithm aims to find the best separating surface as being furthest from both classes. It is simultaneously to maximize the margin between the support planes for each class and minimize the error. This can be accomplished through the quadratic program (4).

From the α_i obtained by the solution of (4), we can recover the separating surface and the scalar b determined by the support vectors (for which $\alpha_i > 0$). By changing the kernel function K as a linear inner product, a polynomial,

a radial basis function or a sigmoid neural network, we can get different classification model. The classification of a new data point x is based on:

$$sign(\sum_{i=1}^{\sharp SV} y_i \alpha_i K\langle x, x_i\rangle - b)$$

For one-class (novelty detection), the SVM algorithm is to find a hypersphere with a minimal radius R and center c which contains most of the data. And then novel test points lie outside the boundary of the hypersphere.

SVM can also be applied to regression problem by the introduction of an alternative loss function. By using an ϵ-insensitive loss function proposed by Vapnik, Support vector regression (SVR) aims to find a predictive function *f(x)* that has at most ϵ deviation from the actual value y_i.

These tasks can be also accomplished through the quadratic program. [Bennett and Campbell, 2000] and [Cristianini and Shawe-Taylor, 2000] provide more details about SVM and others kernel-based learning methods.

We have added a new construction kernel code to the publicly available toolkit, LibSVM (ref. http://www.csie.ntu.edu.tw/∼cjlin/libsvm). Thus, the software program is able to deal with interval data in classification, regression and novelty detection tasks. To apply the SVM algorithms to the multi-class classification problem (more than 2 classes), LibSVM uses one-against-one strategy. Assume that we have k classes, LibSVM construct *k*(k-1)/2* models. A model separates i^{th} class against j^{th} class. Then to predict the class for a new data point, LibSVM just predicts with each model and finds out which one separates the furthest into the positive region. We have used datasets from Statlog, the UCI Machine Learning Repository (ref. http://www.ics.uci.edu/∼mlearn/MLRepository.html), Regression Datasets (ref. http://www.liacc.up.pt/∼ltorgo/Regression/DataSets.html) and Delve (ref. http://www.cs.toronto.edu/∼delve). By using K-means algorithm [MacQueen, 1967], the large datasets are aggregated into smaller ones. A data point in interval datasets corresponds to a cluster, the low and high values of an interval are computed by the cluster data points. Some other methods for creating interval data can be found in [Bock and Diday, 1999]. The interval version of datasets is shown in table 1 and 2. We report the cross validation accuracy on classification results and mean squared error on regression results presented in table 1.

The results on novelty detection task are presented in table 2 with the number of outliers and significant outliers (furthest from other data points in the dataset). To the best of our knowledge, there is no other available algorithm being able to deal with interval data in both non linear classification, regression and novelty detection tasks. There is not experimental results on interval data mining provided by the others algorithms. Therefore, we only report results obtained by our approach. It is difficult to compare with the others ones.

Datasets	Points	Dims	Protocol	Accuracy	Mean squared error
Wave(3 classes)	30	21	leave-1-out	80.00%	0.462389
Iris(3 classes)	30	4	leave-1-out	100.00%	0.078389
Wine(3 classes)	36	13	leave-1-out	97.22%	0.075182
Pima(2 classes)	77	8	leave-1-out	79.22%	0.212736
Segment(7 classes)	319	19	10-fold	91.22%	1.696050
Shuttle(7 classes)	594	9	10-fold	94.78%	1.096640

Table 1. SVM classification and regression results

Datasets	Points	Dims	Nb. oulliers	Significant outliers
Shuttle	594	9	31	9
Bank8FM	450	8	12	6

Table 2. One-class SVM results

3.2 Other kernel-based methods

Many multivariate statics algorithms based on generalized eigenproblems can be also kernelized [Shawe-Taylor and Cristianini, 2004], e.g Kernel Fisher's Discriminant Analysis (KFDA), Kernel Principal Component Analysis (KPCA), Kernel Partial Least Squares (KPLS), etc. These kernel-based methods can also use the RBF function to build interval data models. We use KPCA and KFDA to visualize datasets in the embedding space where the user can intuitively see the separating boundary between the classes based on the human pattern recognition capabilities. The eigenvectors of the data

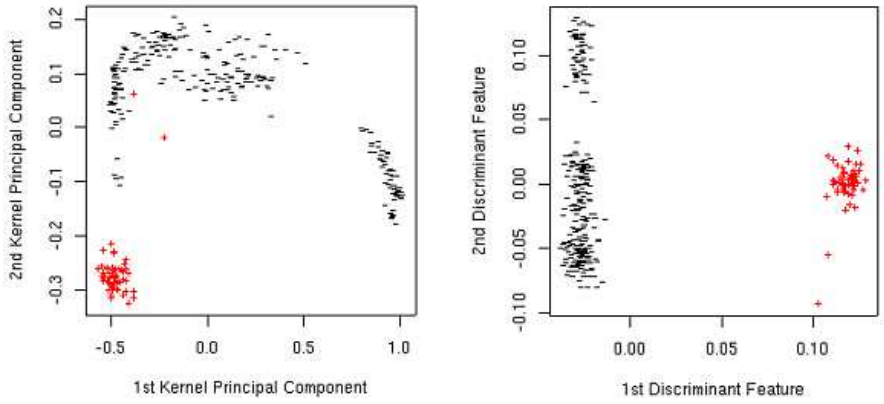

Fig. 1. Visualization of Kernel Principal Component Analysis (left) and Kernel Fisher's Discriminant Analysis (right) on the Segment dataset.

can be used to detect directions of maximum variance, and thus, linear PCA is to project data onto principal components by solving a eigenproblem. By

using a kernel function instead of the linear inner product in the formula, we obtain non linear PCA (KPCA). An example of the visualization of the Segment interval dataset (the class 7 against all) with KPCA using the RBF kernel function is shown in figure 1 (left).

In linear FDA, we consider projecting all the multi-dimensional data onto a generic direction w, and then separately observing the mean and the variance of the projections of the two classes. By substituting the kernel function for a linear inner product into the linear FDA formula, we have non linear FDA (KFDA). An example of the visualization of the Segment interval dataset (the class 7 against all) with KFDA using the RBF kernel function is shown in figure 1 (right).

And thus, the separating boundary between two classes is clearly represented in the embedding space.

4 Inductive rules extraction for explaining SVM results

Although SVM algorithms have shown to build accurate models, their results are very difficult to understand. Most of the time, the user only obtains information regarding the support vectors being used as "black box" to classify the data with a good accuracy. The user does not know how SVM models can work. For many data mining applications, understanding the model obtained by the algorithm is as important as the accuracy.
We propose here to use interactive decision tree algorithms [Poulet, 2003] to try to explain the SVM results. The SVM performance in classification task is deeply understood in the way of IF-THEN rules extracted intuitively from the graphical representation of the decision trees that can be easily interpreted by humans.

Figure 2 is an example of the inductive rule extraction explaining support vector classification results on the Segment interval dataset. The SVM algorithm using the RBF kernel function classifies the class 7 (considered as +1 class) against all other classes (considered as -1 class) with 100.00 % accuracy. CIAD uses 2D scatter plot matrices [Carr *et al.*, 1987] for visualizing interval data [Poulet, 2003]: the data points are displayed in all possible pair-wise combinations of dimensions in 2D scatter plot matrices. For n-dimensional data, this method visualizes *n(n-1)/2* matrices. A data point in two interval dimensions is represented by a two dimensions primitive cross and color corresponds to the class. The user interactively chooses the best separating split (parallel to an axis) to interactively construct the decision tree (based on the human pattern recognition capabilities) or with the help of automatic algorithms. The obtained decision tree having 4 leaves (corresponding to 4 rules) can explain the SVM model. One rule is created for each path from the root to a leaf, each dimension value along a path forms a conjunction and the leaf node holds the class prediction. And thus, the non linear SVM is

Input: non label dataset SP et a SVM classification function f
Output: inductive rule set $IND\text{-}RULE$ explaining the SVM model

1. Classify non label dataset SP using SVM classification function f, we obtain label set L assigned to SP:

$$\{SP,\ L\} = \text{SVM\_classify}(SP,\ f)$$

2. Interactively constructing decision tree model DT on dataset $\{SP,\ L\}$ using visual data mining decision tree algorithms, e.g CIAD [Poulet, 2003].

3. User extracts inductive rules $IND\text{-}RULE$ from graphical representation of decision tree model DT:

$$IND\text{-}RULE = \text{HumanExtract}(\text{graphical } DT)$$

Table 3. Inductive rules extraction from SVM models

interpreted in the way of the 4 inductive rules (IF-THEN) that will be easy to understand.

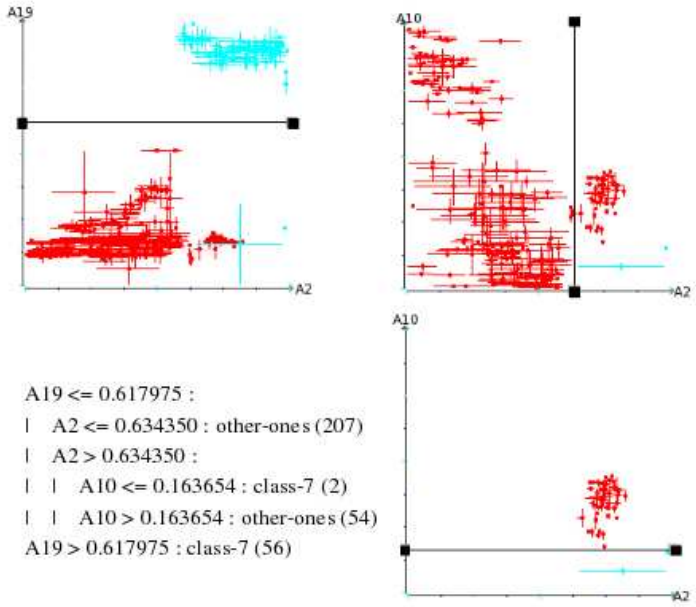

Fig. 2. Visualization of the decision tree explaining the SVM result on the Segment dataset.

5 Visualization tool for explaining SVM results

We have studied some ways to try to explain SVM results by using the graphical representation of high dimensional data. The information visualization methods guide the user towards the most appropriate visualizations for viewing mining results (post-processing step). There are many possibilities to visualize data by using different visualization methods, but all of them have some strengths and some weaknesses. We use the linking technique to combine different visualization methods to overcome the single one. The same information is displayed in different views with different visualization techniques providing useful information to the user. The interactive brushing technique allows the user to focus on a region (brush) in the data displayed to highlight groups of data points. And thus, the linked multiple views provide more information than the single one. We use the interactive brushing and linking techniques and the different visualization methods to try to explain SVM results. For classification tasks with SVM algorithms,

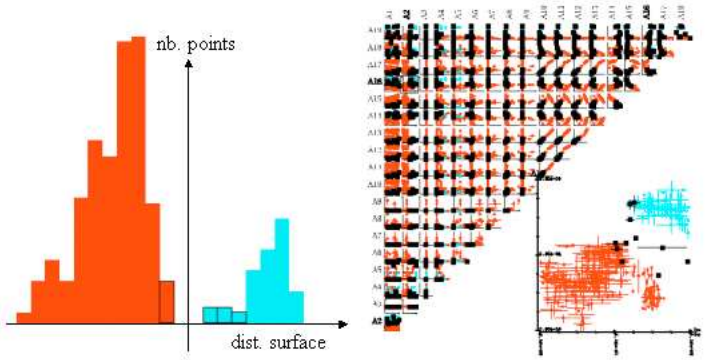

Fig. 3. Visualization of the classification result on the Segment dataset.

understanding the margin (furthest distance between +1 class and -1 class) is one of the most important key of the support vector classification. For this, it is necessary to see the points near the separating boundary between the two classes.

For achieving this goal, we propose to use the data distribution according to the distance to the separating surface. While the classification task is processed (based on the support vectors), we also compute the data distribution according to the distance to the separating surface. For each class, the positive distribution is the set of correctly classified data points and the negative distribution is the set of misclassified data points. The data points being near the frontier correspond to the bar charts near the origin. When the bar charts corresponding to the points near the frontier are selected, the data points are also selected in the other views (visualization methods) by

using the brushing and linking technique. We use 2D scatter plot matrices for visualizing interval data. The user can see approximately the boundary between classes and the margin width. This helps the user to evaluate the robustness of the model obtained by support vector classification. He can also know the interesting dimensions (corresponding to the projections providing a clear boundary between the two classes) in the obtained model. Figure 3 is an example of visualizing support vector classification results on the Segment interval dataset (the class 7 against all). From data distribution according to the distance to the separating surface, the 4 bar charts near the origin are brushed, and then the corresponding points are linked and displayed in 2D scatter plot matrices. The dimensions 2 and 16 corresponding to the projection provides a clear boundary between the two classes and are interesting in the model obtained.

We have extended this idea for visualizing support vector regression results. We have also computed the data distribution according to the distance to the regression function. After that, we combine the histogram with 2D scatter plot matrices for visualization. When the user selects the data points far from the regression function, he can know how the function fits data. If the function well predicts the data points of high density region then the model obtained is interesting.

For a novelty detection task, we visualize the outliers allowing the user to valid them. The approach is based on the interactive linking and brushing of the histogram and 2D scatter plot views. The histogram displays the data distribution according to the distance to the hypersphere obtained by one class SVM. The data points far from the hypersphere are brushed in the histogram view, thus they are automatically selected in 2D scatter plot view. The user can validate the outliers. And then, the dimensions corresponding to the projection presents clearly the outliers and are interesting in the obtained model.

6 Conclusion

We have presented in this paper the interval data mining approach using kernel-based and visualization methods.

We have proposed to construct a new RBF kernel on interval data. This modification tremendously changes kernel-based algorithms. No algorithmic changes are required from the usual case of continuous data other than the modification of the RBF kernel evaluation. Thus, kernel-based algorithms can deal with interval data in classification, regression and novelty detection. It is extremely rare algorithms being able to construct non linear models on interval data for the three problems: classification, regression and novelty detection.

We have also proposed two ways to try to explain SVM results that are a well-known "black box". The first one is to use interactive decision tree

algorithms for explaining the SVM results. The user can interpret the SVM performance in the way of IF-THEN rules extracted intuitively from the graphical representation of the decision trees that can be easily interpreted by the user. The second one is based on a set of different visualization techniques combined with linking and brushing techniques gives an insight into classification, regression and novelty detection tasks with SVM. The graphical representation shows the interesting dimensions in the obtained model.

A forthcoming improvement will be to extend our approach to data of taxonomic or mixture types.

References

[Bennett and Campbell, 2000]K. Bennett and C. Campbell. Support vector machines: Hype or hallelujah ?. *SIGKDD Explorations*, pages 1–13, 2000.

[Bock and Diday, 1999]H-H. Bock and E. Diday. *Analysis of Symbolic Data*. Springer-Verlag, 1999.

[Carr *et al.*, 1987]D-B. Carr, R-J. Littlefield, W-L. Nicholson, and J-S. Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, pages 424–436, 1987.

[Cristianini and Shawe-Taylor, 2000]N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[MacQueen, 1967]J. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[Mika *et al.*, 1999]S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K-R. Müller. Fisher discriminant analysis with kernels. *Neural Networks for Signal Processing IX*, pages 41–48, 1999.

[Poulet, 2003]F. Poulet. Interactive decision tree construction for interval and taxonomical data. In *Proceedings of VDM@ICDM'03, 3nd Workshop on Visual Data Mining*, pages 183–194, 2003.

[Rosipal and Trejo, 2001]R. Rosipal and L-J. Trejo. Kernel partial least squares regression in reproducing kernel hilbert space. *Journal of Machine Learning Research*, pages 97–123, 2001.

[Schölkopf *et al.*, 1998]B. Schölkopf, A. Smola, and K-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, pages 1299–1319, 1998.

[Shawe-Taylor and Cristianini, 2004]J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[Vapnik, 1995]V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.