



WILEY

Interdisciplinary Reviews

DATA MINING AND
KNOWLEDGE DISCOVERY

Please select the appropriate article type (per your Contributor's Agreement) by double-clicking on the associated check box.

<input type="checkbox"/>	Overview	Overviews will provide a broad and relatively non-technical treatment of important topics at a level suitable for advanced students and for researchers without a strong background in the field.	~8,000 words 16 figures/tables 25-50 references
<input type="checkbox"/>	Advanced Review	Advanced Reviews, aimed at researchers and advanced students with a strong background in the subject, will review key areas of research in a citation-rich format similar to that of leading review journals.	~5,000 words 10 figures/tables 50-75 references
<input checked="" type="checkbox"/>	Focus Article	Focus articles will describe specific real-world issues, examples, implementations, etc. These articles will be technical in nature.	~3,000 words 6 figures/tables 15-25 references
<input type="checkbox"/>	Opinion	Opinions provide a forum for thought-leaders, hand-picked by the editors, to provide a more individual perspective on the field in question.	~3,000 words 6 figures/tables 15-25 references

Article title: Visualization and Interpretation of SVM Classifiers

First author: Full name and affiliation; plus email address if corresponding author Sauptik Dhar is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455 USA (e-mail: धार007@umn.edu).
Second author: Full name and affiliation; plus email address if corresponding author Vladimir Cherkassky is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455 USA (e-mail: cherk001@umn.edu).
Third author: Full name and affiliation; plus email address if corresponding author [Type information here]

Abstract

[Many machine learning applications involve modelling sparse high dimensional data. Examples include genomics, brain imaging, time series prediction etc. A common problem in such studies is the understanding of complex data-analytic models, especially nonlinear high-dimensional models such as Support Vector Machines (SVM). In this paper we provide a brief survey of the current techniques for the visualization and interpretation

of SVM based classification models and highlight on the issues with such techniques. Further, we present a simple graphical method that can be useful for understanding and visualization of SVM models.]

Introduction

[General acceptance of high-dimensional predictive data-analytic models by practitioners is hindered by their poor interpretation capability. This is especially evident in medical and biomedical applications, where practitioners and researchers are accustomed to classical biostatistical models using just a few explanatory (input) variables in a predefined (e.g., linear) parameterization. In contrast, modern machine learning methods may use hundreds or thousands of input variables and such methods adopt flexible parameterizations, such as multi-layer perceptrons (MLPs) or support vector machines (SVMs). These flexible nonlinear methods usually achieve better generalization (than classical statistical methods), but they are viewed by the users as ‘black-box’ models. Classification methods specify (nonlinear) decision boundary in the space of input variables. This decision boundary is estimated from available training data, but is intended for classifying future (or test) input samples. For high-dimensional data, understanding and interpretation of both the training data and the estimated decision boundary is challenging, because (a) human intuition fails in high dimensions, and (b) the geometric properties of finite high-dimensional data are very different from low-dimensional data¹.

Clearly, a simple graphical representation for such high-dimensional models is useful for their understanding by practitioners. However, we argue that meaningful interpretation of predictive black-box models should also reflect properties of an underlying modelling technique. For example, interpretation of SVM classifiers may not be possible without the understanding of its basic concepts such as margin. Yet most existing methods for interpretation and visualization of black-box models simply apply standard statistical data visualization techniques, like projecting the high-dimensional data onto a 2D plane manually selected by users; or some rule induction technique like a decision tree that tries to approximate the SVM model.

This paper provides a brief survey of the current techniques for the visualization and interpretation of SVM based classifiers and highlights the basic issues inherent to these methods. Further, we present a simple graphical method called the “univariate histogram of projections” which can be used for interpretation and visualization of SVM-based classifiers. The practical utility of this method is demonstrated for several high-dimensional data sets. Further, we show the utility of this method for understanding SVM models under unbalanced data settings.

The paper is organized as follows. Section I provides an overview of existing methods for visualization and interpretation of SVM classifiers. Section II provides a brief description of the univariate histogram of projections for both linear and non-linear SVM models. Section III describes how the proposed method can be used for SVM models with unbalanced data and unequal misclassification costs. Finally the conclusion section briefly describes relevant future work.]

II. METHODS FOR VISUALIZATION AND INTERPRETATION OF SVM MODELS

Two important aspects for the interpretation of SVM black-box models are Rule Extraction and Visualization.

Rule Extraction from Support Vector Machines

Rule Extraction from SVM models involve obtaining a set of 'if ... then ...else' rules that approximate the SVM decision boundary. There are two basic approaches adopted for such rule extraction techniques

Decompositional Approach

These methods involve local approximation of the SVM decision boundary using hyper-rectangles or hyper-ellipsoid regions in the input space. These regions are then interpreted as rules. Several representative methods include:

- SVM+ Prototype²
- RuleExSVM³
- HRE algorithm⁴
- Rule extraction from linear SVMs⁵
- SQReX-SVM⁶

Eclectic Approach

These methods typically approximate the trained SVM model using another (interpretable) learning method, which yields an interpretable model such as decision trees or fuzzy rules⁷. The assumption is that such an *eclectic* rule induction procedure closely approximates the original SVM model. The methods that follow this approach include:

- Learning-based rule-extraction from support vector machines⁸
- Eclectic Rule-Extraction from SVM⁹
- Active Learning Based Algorithm (ALBA)¹⁰.

Descriptions of these methods can be found in surveys^{10,11,12}. In this paper we highlight on the fundamental issues affecting all such approaches. For illustration we use the ALBA algorithm detailed next.

Algorithm1 Active Learning Based Algorithm (ALBA)

1. Estimate the SVM model with optimally tuned parameters (C and kernel parameters).
2. Change the class labels of the training data to labels predicted by the SVM classifier.

3. Calculate the average distance of training samples from the support vectors as follows, for all dimensions j do

```
distancej = 0
for all support vectors svi do
    for all training data instance x do
        distancej = distancej + |dj - svij|
    end for
end for
```

$$\text{distance}_j = \frac{\text{distance}_j}{(\#sv) \times (\#Training_samples)}$$

end for

4. Create 1,000 extra data instances close to the support vectors.

for $i = 1$ to 1,000 do

Randomly choose one of the support vectors sv_k and generate an extra data instance x_i close to it.

for all dimensions j do

$$x_{i,j} = sv_{k,j} + \left[(rand(0,1) - 0.5) \times \frac{\text{distance}_j}{2} \right]$$

end for

Provide a class label y_i using the trained SVM

end for

5. Run *rule induction* algorithm on the data set containing both the training data, and newly created data samples

There are two critical issues for both the *decompositional* or *eclectic* approaches.

- How unique are the rules extracted by these methods from available data?
- How such methods scale for High-Dimensional Low Sample Size (HDLSS) data?

The first issue can be explained using a simple example. Consider the synthetic 2 dimensional *Noisy Hyperbolas* data, where the underlying distributions for two classes are given by functions: $x_2 = ((x_1-0.4)*3)^2+0.225$ with $x_1 \in [0.2, 0.6]$ for class 1 and $x_2 = 1-((x_1-0.6)*3)^2-0.225$ with $x_1 \in [0.4, 0.8]$ for class 2. Gaussian noise is added to the x_2 coordinates for both the classes. The degree of data separation is controlled by noise level $\sigma = 0.025$. For this experiment,

- Number of training samples = 100. (50 per class).
- Number of validation samples = 100. (This independent validation set is used for model selection).
- Number of test samples = 2000 (1000 per class).

For this dataset, the nonlinear RBF kernel of the form $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$ with optimally tuned parameters yields the SVM decision boundary shown in Fig. 1a. We further construct the ALBA decision tree for this SVM model (as shown in Fig. 1b) which partitions the input space into regions (as shown in Fig. 1c). The same steps are also performed for the Polynomial kernel $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d$ and the corresponding figures are shown in Fig. 2 for the 5th order polynomial kernel. The training and test errors for both SVM and ALBA for the different kernels are provided in Table 1.

As evident from Table 1, the prediction accuracy of the SVM and its ALBA model is quite similar for different kernels. However, from Fig. 1b and Fig. 2b we can see that the rules obtained (using ALBA) from these two SVM models are very different. This difference is also highlighted in Figs. 1c and 2c showing the partitioning of the input space by the two different ALBA decision tree models. Hence we conclude that the rules extracted by these two decision trees are very different. This leads to the dilemma; '*Which of these two rule sets should be used?*' This question may be critical in medical applications, when a clinician tries to understand the importance

of a patient's attributes for the medical diagnosis. Thus, the non-uniqueness of the rule sets, especially at the upper part of the decision tree is an important drawback for such techniques.

Another important issue is the scalability of such approaches for HDLSS data. To illustrate this point, consider the sparse high dimensional *MNIST handwritten digit data set*¹³, where training data samples represent handwritten digits 5 and 8, and the goal is to estimate binary classifier for discriminating these two digits. Each sample is represented as a real-valued vector of size $28*28=784$. On average, 22% of the input features are non-zero which makes this data sparse. For this example we use the following setting:

- Number of training samples= 1000. (500 per class)
- Number of validation samples =1000. (This independent validation set is used for model selection).
- Number of test samples = 1866.

For this dataset, we apply both the RBF and Polynomial SVM ($d=3$) and obtain the ALBA decision trees for the two SVM models. The training and test errors for both SVM and ALBA for the different kernels are provided in Table 2.

From Table 2 we can clearly see that the SVM model and the corresponding ALBA tree have the same training errors. However, their test errors are quite different, and the ALBA decision tree performs significantly worse (than SVM) on an independent test set. Hence, we conclude that the ALBA approach, along with other rule-extraction methods, suffer from the *curse of dimensionality*. So these approaches do not provide useful predictive models for high-dimensional data.

Visualization of the SVM models

Very few papers have addressed the problem of understanding the SVM classifiers based on graphical representation^{14,15,16}. These papers typically apply standard statistical data exploration/visualization techniques, such as manual selection of low-dimensional projections and identification of a few 'important' input features, to analyze the SVM models. Carega et al.¹⁴ proposed the method of projecting the data samples onto a 2-D plane. This 2-D plane is manually selected by the user using the R interface of the ggobi software¹⁷. Based on the location of the support vectors and the separability of the data in the projected 2-D space, this approach tries to identify the importance of the 2-D plane and hence the importance of the variables forming this 2-D plane. Such an interactive technique may be helpful when we have significant domain knowledge (about the 'true' model), or when the data is relatively simple and low-dimensional. However, in most applications this is not the case, and such interactive methods can place a heavy burden on a human modeler trying to examine large number of possible projections. Wang et al. introduces another method called support vector machine visualization (SVMV)¹⁵. In their method, the input samples are projected nonlinearly onto a 2-D Self-Organizing Map (SOM). Further, they embed the SVM decision boundary in the SOM map by predicting the class label of the SOM units using the SVM model. However, such a method may complicate the task of the human modeler, as it requires optimal selection of several SOM tuning parameters. Further, this method does not provide any visual representation of the soft margins.

So there is a need for a simple method for visualizing the SVM models. The method proposed in this paper is based on the idea of projecting the data onto the normal direction of the (SVM) classification decision boundary. The idea itself is very simple, and it has been widely used, under different contexts, in statistics and machine learning. Perhaps the original application of this idea dates back to interpretation of high-dimensional data onto the direction given by Fisher's Linear Discriminant Analysis (LDA) method.

More recent examples of using the idea of projecting the data onto the normal direction of SVM decision boundary are discussed in "References 1, 16, 18 and 19." However, the motivation behind using this technique

has been different in all these cases. It has been used to investigate the properties of sparse high-dimensional data, and to show the effect of *data piling* by Cherkassky and Mulier¹ and by Ahn and Marron¹⁸. Poulet used this method mainly for feature selection and for SVM parameter tuning¹⁶. In the paper by Platt¹⁹ this method is used to visualize the class conditional densities and hence determine a parametric form for the posterior density $P(\text{class} | \text{SVM_model})$. Our paper advocates the method of projections for improved understanding of SVM models. The main technical contribution is that the analysis of univariate projections of the data incorporates well-known properties of SVM-based classifiers. These properties are, in fact, quite useful for understanding of ‘black-box’ SVM models by the practitioners and general users.

III. UNIVARIATE HISTOGRAM OF PROJECTIONS

This section describes simple graphical representation of the training data for binary classification and the SVM decision boundary (estimated from this data) via the *univariate histogram of projections*.

Univariate Histogram of Projections \sim is the histogram of the projection values of the data samples onto the normal direction of the SVM decision boundary.

For linear SVM model $D(\mathbf{x}) = \text{sign}[(\mathbf{w} \cdot \mathbf{x}) + b]$ this normal direction is given by the ‘weight’ vector \mathbf{w} .

Such a histogram is obtained via the following three steps:-

- a. Estimate standard SVM classifier for a given (labeled) training data set. Note that this step includes optimal model selection, i.e. tuning of SVM parameters (regularization parameter and kernel).
- b. Generate low-dimensional representation of training data by projecting it onto the normal direction vector of the SVM hyperplane estimated in (a).
- c. Generate the histogram of the projected values obtained in (b).

These three steps are illustrated in Fig. 3, for linear SVM model.

In Fig. 3c, SVM decision boundary is marked as zero, and the margin borders for positive/negative classes are marked, respectively as +1/-1. Visual analysis of this univariate histogram of projections can be helpful for understanding high-dimensional data. For example, consider a linear decision boundary for the synthetic *Noisy Hyperbolas* data set in Fig. 4a. The projected values in Fig. 4b are calculated analytically using linear SVM model $f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b$, and then the histogram of the projection of training samples is generated as shown in Fig. 4b. The projected values for the two classes overlap, and this is consistent with the fact that the training samples are not linearly separable.

For non-linear SVM kernels, the projected values $f(\mathbf{x})$ are calculated by using the kernel representation in the dual space. In this case, the projection of training sample \mathbf{x}_k onto the normal direction of the nonlinear SVM

decision boundary equals $f(\mathbf{x}_k) = \sum_{i \in S, v} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_k) + b$, where \mathbf{x}_i is a support vector. For example, consider

nonlinear decision boundary for the synthetic Noisy Hyperbolas data set in Fig. 5a. Using nonlinear RBF kernel of the form $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$ with optimally tuned parameters yields SVM decision boundary shown in Fig. 5a, and the corresponding histogram of projections in Fig. 5b. Figure 5b clearly shows that the training samples are well separable using nonlinear decision boundary. In this case, the histogram does not add much to our understanding, because this data is clearly separable in the two dimensional input space. However, for high-dimensional settings, visual representation of the data in the original input space, similar to Fig. 5a, is impossible, so the histogram representation becomes indeed quite useful. For instance, for most applications

with high-dimensional data, the training data is typically linearly separable. However, this does not usually imply separability (i.e., good generalization) for future (test) samples. Visual representation of such high-dimensional data, using histograms of projections of training and test samples onto the normal direction of an optimal (nonlinear) SVM decision boundary can help to explain and understand generalization properties of SVMs.

To illustrate this point, consider the sparse high dimensional *MNIST handwritten digit data set* described in section II. For this experiment we use the nonlinear RBF kernel and tune the SVM parameters on an independent validation set according to the following model selection algorithm. For this digits data set typical tuning parameter values are $C \sim 1$ or 2 and $\gamma \sim 2^{-6}$

Algorithm 2 Model Selection (with independent validation set)

- Estimate the SVM model from training data, for each set of (C, γ) parameter values.
 - Select the SVM model parameter (C^*, γ^*) providing the smallest classification error for the validation set.
-

The univariate histogram of projections for the training data for the optimally tuned SVM model is shown in Fig. 6a. As evident from Fig. 6a, the training samples are well separable in this optimally chosen RBF kernel space. This is typically the case for HDLSS setting, where the training samples are generally well separable in some optimally chosen kernel space. Also, the histogram of projections clearly illustrates the clustering of data samples at the margin borders. This effect, called data piling, is typical for high-dimensional data^{1, 18}. However, separability of the training data does not imply separability for the validation or test samples. This can be seen from the projections of validation and test samples in Fig. 6b and Fig. 6c. The SVM optimization algorithm tries to achieve high separability of the training data by penalizing the samples that are inside the soft margin. Hence the histogram in Fig. 6a where many training samples are outside the soft margin is typical. However, during model selection (in Algorithm 2), we are only concerned that the validation samples are correctly classified by the model. Hence we may select a model that allows the validation/test samples to be within the soft-margin as long as it provides small validation error. This results in the overlapping histograms for validation and test data, as shown in Fig. 6b and Fig. 6c. So the histogram of projections technique enables better understanding of the model selection strategy for tuning SVM parameters, for this data set.

IV. HISTOGRAMS OF PROJECTIONS FOR UNBALANCED DATA

Many practical applications use unbalanced data and different misclassification costs. This is common in most biomedical applications, fraud detection etc. Here ‘unbalanced data’ refers to the fact that the number of positive samples is much smaller than negative ones. Also, for such data sets, the cost of false negative errors C_{fn} is typically set higher than that of false positives C_{fp} . However, generally, the misclassification costs should be specified based on application-domain requirements¹.

In such a setting, the quality of a classifier is estimated by its weighted classification error for test samples, with weights given by the misclassification costs, i.e.

$$weighted_test_error = C_{fp}P_{fp} + C_{fn}P_{fn} \quad (1)$$

where P_{fp} and P_{fn} denote the probability (or fraction) of false positive and false negative errors in the test set respectively¹. In this case, the univariate histogram of projection can be quite helpful for interpreting the SVM

models. In particular, this simple graphical representation can contribute to wider acceptance of the black-box SVM predictive models in the medical community.

As an example, consider a recent study of SVM predictive modeling of Transplant-Related Mortality (TRM) for Blood-and-Marrow Transplant patients²⁰. This paper describes application of predictive data-analytic modeling for diagnosis of Graft-versus-Host Disease (GVHD), which is a common side effect of an allogenic bone marrow or cord blood transplant. The disease affects about 50% of transplant recipients and is the leading cause of early (acute GVHD) and late (chronic GVHD) mortality post transplant. GVHD occurs because some of donor's immune cells (known as T cells) attack cells in recipient's body. Hence, it may be interesting to analyze the association of genetic variants of donor and transplant recipient genes with acute and chronic graft versus host disease. Additional clinical and demographic factors that may impact the outcome (of successful transplants) include: recipient and donor age, type of donor (related/unrelated), disease status prior to transplant, stem cell source (peripheral blood vs. marrow), etc. In this study, the goal of modeling was to predict patient's survival (alive or dead) one year post-transplant using the patients' clinical, genomic and demographic data. The data has records of 301 patient samples each with 136 SNPs. This data set has 221 samples labeled 'alive' and 75 samples labeled

'dead', i.e. the ratio of alive-to-dead samples is 3:1. Further, the ratio of misclassification costs used $\frac{C_{fp}}{C_{fn}}$ in the study was set as 1:3. The prediction accuracy of the classifier is measured using the weighted test error (1). This study followed a standard experimental procedure, where initially a small set of 14 informative features was selected using standard feature selection methods, and then an SVM classifier was estimated for this 14 dimensional dataset.

Here we reproduce (in Fig. 7) the univariate histogram of projections for 14-dimensional SVM model for predicting TRM²⁰. This figure illustrates unbalanced class distributions, and the effect of unequal misclassification costs on the performance of SVM classifier. In particular, as shown in Fig. 7, the misclassification error rates for patients classified as negative (Alive) is small (~7%), whereas the error rate for patients classified as positive (Dead) is large (~47%). Such a dichotomy is common for many practical problems with unbalanced data. In practical terms, it means that prediction of test samples classified as negative (~Alive) is very reliable and can be made with high confidence. On the contrary, prediction of test samples classified as positive (~Dead) is very inaccurate. Thus the proposed technique of univariate projections can also enable improved understanding of the inherent 'bias' of estimated predictive models for unbalanced data with unequal costs. The graphical representation, such as shown in Fig. 7, is very intuitive and useful for medical practitioners, who may be reluctant to make decisions based on complex mathematical models.

Conclusion

[This paper presented a simple method for interpretation of high dimensional nonlinear SVM models using univariate histograms of projections. The resulting graphical representation can be used

- a) by practitioners who need to make predictive/diagnostic decisions but are reluctant to do so based solely on black-box SVM data-analytic models.
- b) by researchers concerned with in-depth analysis and improved understanding of SVM-based classifiers.

Overall, we found this method quite useful for understanding of the multivariate SVM models under different standard and non-standard learning settings. This paper shows several such examples, including the data piling phenomenon for high-dimensional data^{1, 18}, improved understanding of SVM parameter tuning and SVM modeling of unbalanced data sets. Recently, this method has been used^{21, 22} to explain the conditions for the effectiveness of a new learning methodology called Universum Learning²³. Future related work may involve application of this method to multi-class problems, and to improving SVM model selection for unbalanced classification problems.

In conclusion, we point out that all additional insights about interpretation and visualization of high dimensional SVM-based classifiers, presented in this paper, have resulted from incorporating important properties of SVM models into a simple univariate graphical representation. This is a critical point, because many earlier SVM interpretation techniques simply apply classical statistical visualization approaches developed for low-dimensional settings.]

Notes

[Please add any notes here]

References

1. Cherkassky V, Mulier F. *Learning from Data Concepts: Theory and Methods*. 2nd ed. New York: Wiley; 2007.
2. Nunez H, Angulo C, Catala A. Rule-Extraction from Support Vector Machines. Proc. of European Symposium on Artificial Neural Networks, Burges; 2002 107-112.
3. Fu X, Ongt C, Keerthit S, Hung G, Goh L. Extracting the Knowledge Embedded in Support Vector Machines. Proc. IEEE Int'l Conf. Neural Networks; 2004 291-296.
4. Zhang Y, Su H, Jia T, Chu J. Rule Extraction from Trained Support Vector Machines. Proc. Ninth Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining; 2005 61-70.
5. Fung G, Sandilya S, Rao R. Rule Extraction from Linear Support Vector Machines. Proc. 11th Int'l Conf. Knowledge Discovery and Data Mining; 2005 32-40.
6. Barakat N, Bradley A. Rule Extraction from Support Vector Machines: A Sequential Covering Approach. IEEE Trans. Knowledge and Data Eng.; 2007, 19:729-741.
7. Cherkassky V, Muller F. *Learning from Data: Concepts, Theory, and Methods*. New York: Wiley; 1998.
8. Barakat N, Diederich J. Learning-Based Rule-Extraction from Support Vector Machines: Performance on Benchmark Data Sets. Proc. Conf. Neuro-Computing and Evolving Intelligence; 2004.
9. Barakat N, Diederich J. Eclectic Rule-Extraction from Support Vector Machines. Int'l J. Computational Intelligence; 2005, 2:59-62.
10. Martens D, Baesens B, Gestel T. Decompositional Rule Extraction from Support Vector Machines by Active Learning. IEEE Transactions on Knowledge and Data Engineering; 2009, 21(2):178-191.
11. Barakat N, Bradley A. Rule-extraction from support vector machines: A review. Neurocomputing; March 2010.doi:10.1016/j.neucom.2010.02.016.
12. Diederich J. *Rule Extraction from Support Vector Machines*. Springer 2008.
13. <http://www.cs.nyu.edu/~roweis/data.html> (accessed: Dec 4, 2010).
14. Caragea D, Cook D, Honavar VG. Gaining insights into support vector machine pattern classifiers using projection-based tour methods. SIGKDD; 2001 251-256.

15. Wang X, Wu S, Li Q. SVMV- a novel algorithm for the visualization of SVM classification results. *Advances in Neural Networks - ISNN 2006, Berlin Heidelberg: Springer-Verlag; 2006 968-973.*
16. Poulet F. SVM and graphical algorithms: a cooperative approach. *Proceedings Fourth IEEE International Conference on Data Mining; 2004 499-502.*
17. Cook D, Swayne DF. *Interactive and Dynamic Graphics for Data Analysis: With Examples Using R and GGobi.* New York: Springer, 2007.
18. Ahn J, Marron, JS. The Maximal Data Piling Direction for Discrimination. *Biometrika; 2010, 97:254–259.*
19. Platt JC. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In *Advances in Large Margin Classifiers; 1999 61-74.*
20. Feng C, Cherkassky V, Weisdorf D, Arora M, Van Ness B. Predictive modeling of Transplant-Related Mortality. *Proc. of the 2010 Design of Medical Devices Conf. Minneapolis, April 2010.*
21. Cherkassky V, Dai W. Empirical Study of the Universum SVM Learning for High-Dimensional Data. *Proc. ICANN; 2009.doi:10.1007/978-3-642-04274-4_96*
22. Cherkassky V, Dhar S, Dai W. Practical Conditions for Effectiveness of the Universum Learning. *IEEE Trans. on Neural Networks.* Feb 2010. (submitted for publication)
23. Vapnik VN. *Estimation of Dependencies Based on Empirical Data. Empirical Inference Science: Afterword of 2006.* New York: Springer; 2006.

]

Figure captions

[FIGURE 1] Illustration of the ALBA algorithm using the RBF SVM for the Noisy Hyperbolas data (with $\sigma = 0.025$). (a) The RBF SVM decision boundary. (b) The ALBA decision tree for the RBF SVM model. (c) Partitioning of the input space into decision regions by the ALBA decision tree.

FIGURE 2| Illustration of the ALBA algorithm using the Polynomial SVM for the Noisy Hyperbolas data (with $\sigma = 0.025$). (a) The Polynomial SVM decision boundary. (b) The ALBA decision tree for the Polynomial SVM model. (c) Partitioning of the input space into decision regions by the ALBA decision tree.

FIGURE 3| Illustration of the steps to generate the univariate histogram of projections. (a) The estimated SVM model and training data.(b) Projection of the training data onto the normal weight vector (\mathbf{w}) of the SVM hyperplane.(c) Univariate histogram of projections. i.e. histogram of $f(\mathbf{x})$ values.

FIGURE 4| (a) Decision boundary for linear SVM model $y = \text{sign}(f(\mathbf{x})) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$. (b) Univariate histogram of projections for training samples \mathbf{x}_k onto the normal vector of linear SVM decision boundary $f(\mathbf{x}_k) = (\mathbf{w} \cdot \mathbf{x}_k) + b$.

FIGURE 5| (a) Decision boundary for non-linear RBF SVM model $y = \text{sign}(f(\mathbf{x})) = \text{sign}(\sum_{i \in S, v} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$. (b) Univariate histogram of projections for training samples onto the normal vector of SVM decision boundary.

FIGURE 6| Univariate histogram of projections for MNIST data set (a) training data; (b) validation data (validation error 1.7%); (c) test data (test error 1.23%).

FIGURE 7| Univariate histogram of projections of high-dimensional data onto normal vector \mathbf{w} . Negative samples (\sim Alive) are shown in green, positive (\sim Dead) in red. False positives and false negative errors are indicated as green-shaded and red-shaded areas, respectively.

]

Tables

[TABLE 1| Performance of ALBA and SVM for Hyperbolas data set

METHODS	RBF		Polynomial (d=5)	
	SVM	ALBA	SVM	ALBA
Training Error (%)	1.0	1.0	5.0	5.0
Test Error (%)	2.5	3.0	4.5	4.95

TABLE 2| Performance of ALBA and SVM for the MNIST data set

METHODS	RBF		Polynomial (d=3)	
	SVM	ALBA	SVM	ALBA
Training Error (%)	0	0	0	0
Test Error (%)	1.23	6.48	1.98	8.47

]

Further Reading/Resources

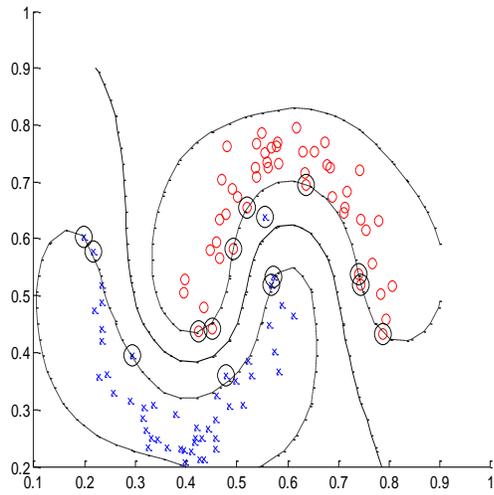
[Please insert any further reading/resources here]

Related Articles

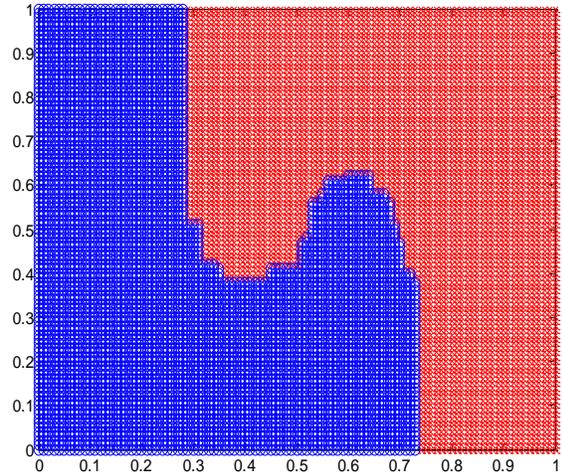
Article ID	Article title
203	Visual data mining



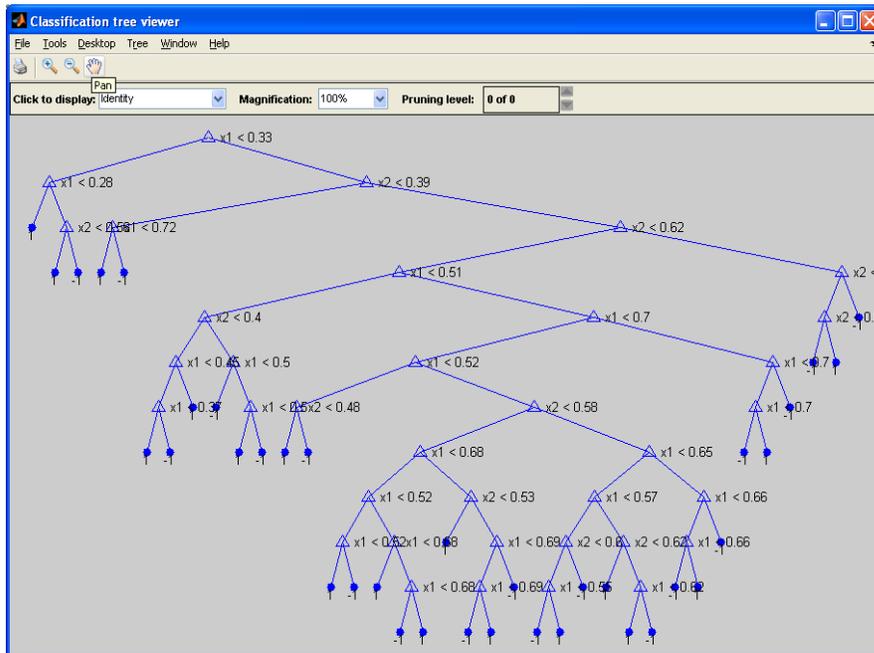
WIREs DMKD article
list.doc



(a)

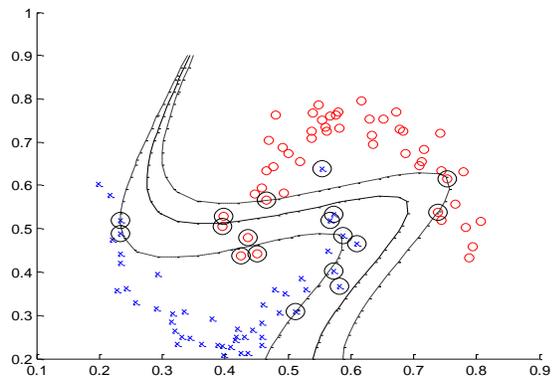


(c)

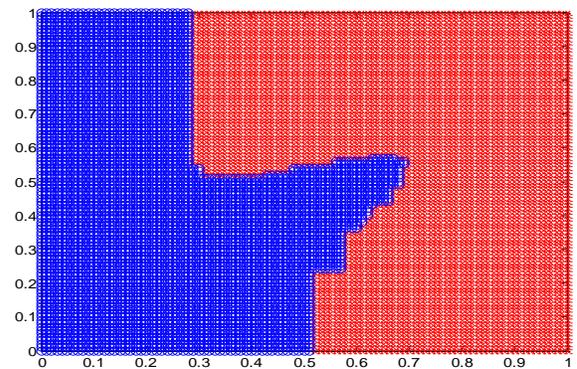


(b)

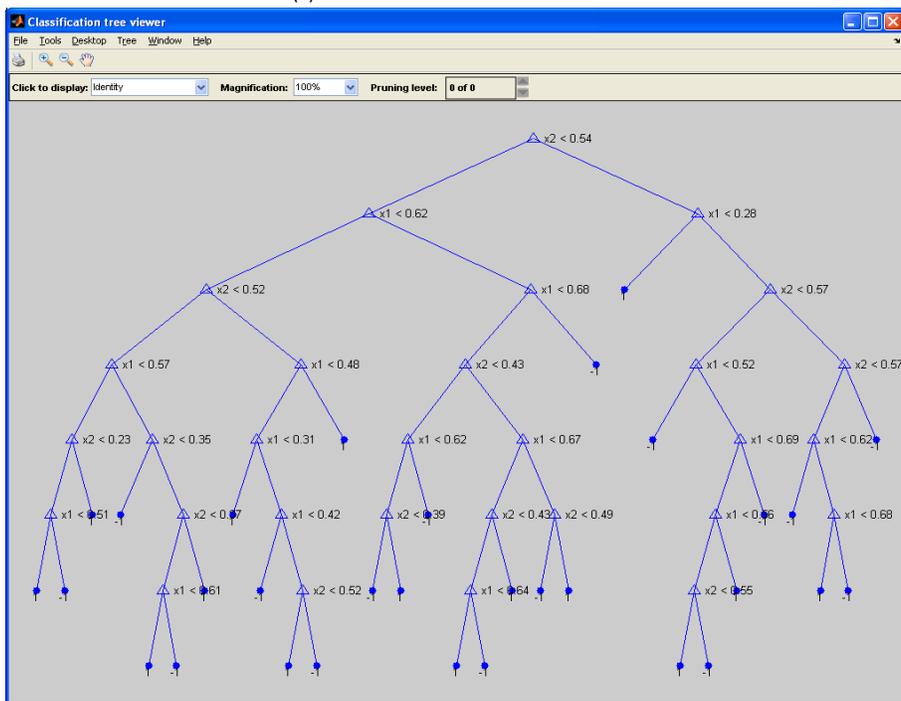
Figure 1 | Illustration of the ALBA algorithm using the RBF SVM for the Noisy Hyperbolas data (with $\sigma = 0.025$). (a) The RBF SVM decision boundary. (b) The ALBA decision tree for the RBF SVM model. (c) Partitioning of the input space into decision regions by the ALBA decision tree.



(a)



(c)



(b)

FIGURE 2 | Illustration of the ALBA algorithm using the Polynomial SVM for the Noisy Hyperbolas data (with $\sigma = 0.025$). (a) The Polynomial SVM decision boundary. (b) The ALBA decision tree for the Polynomial SVM model. (c) Partitioning of the input space into decision regions by the ALBA decision tree.

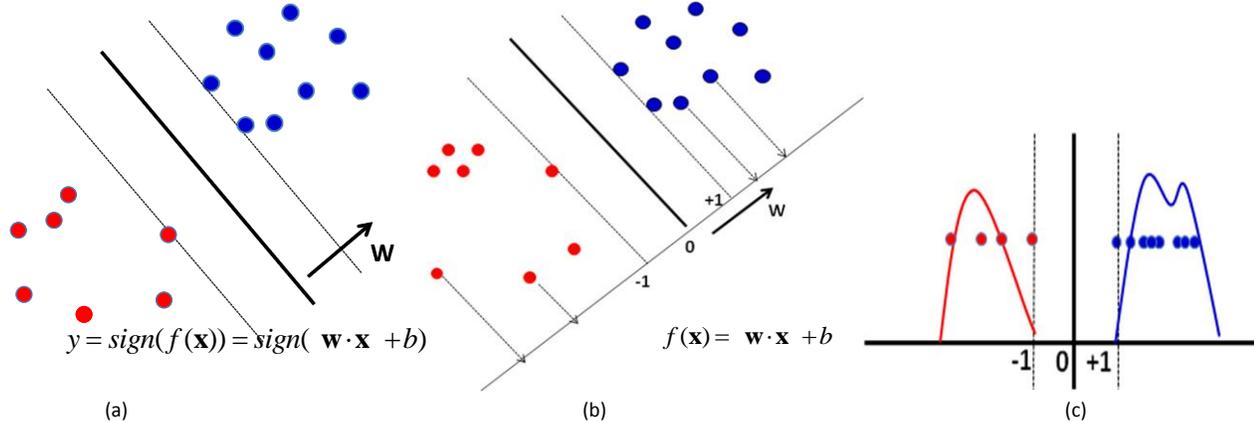


FIGURE 3 | Illustration of the steps to generate the univariate histogram of projections. (a) The estimated SVM model and training data. (b) Projection of the training data onto the normal weight vector (w) of the SVM hyperplane. (c) Univariate histogram of projections. i.e. histogram of $f(\mathbf{x})$ values.

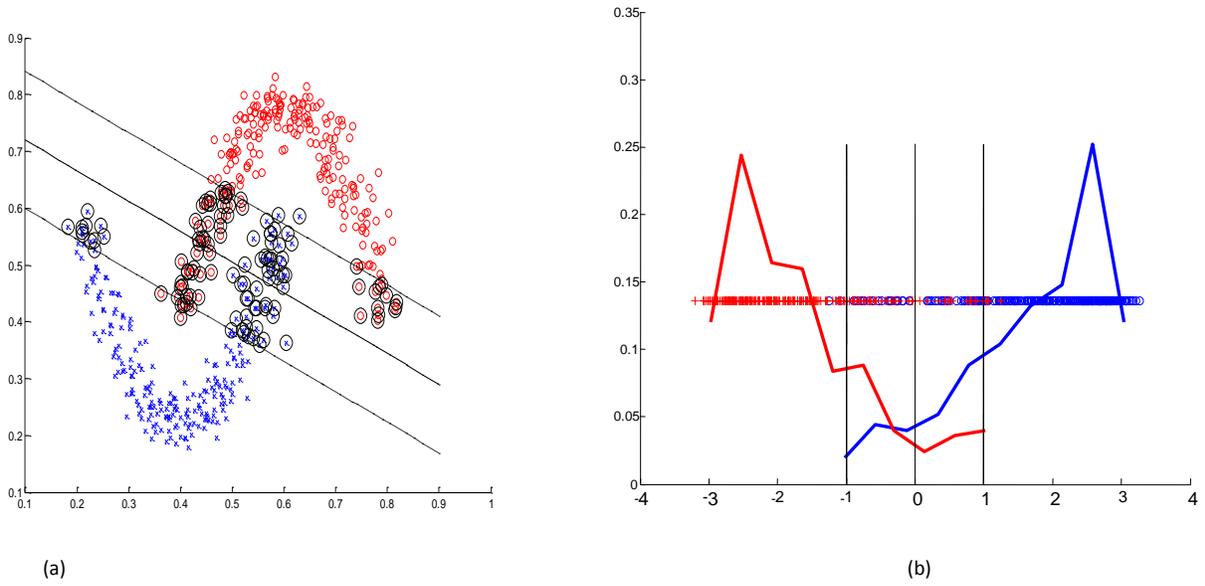


FIGURE 4]. (a) Decision boundary for linear SVM model $y = \text{sign}(f(\mathbf{x})) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$. (b) Univariate histogram of projections for training samples \mathbf{X}_k onto the normal vector of linear SVM decision boundary $f(\mathbf{x}_k) = (\mathbf{w} \cdot \mathbf{x}_k) + b$.

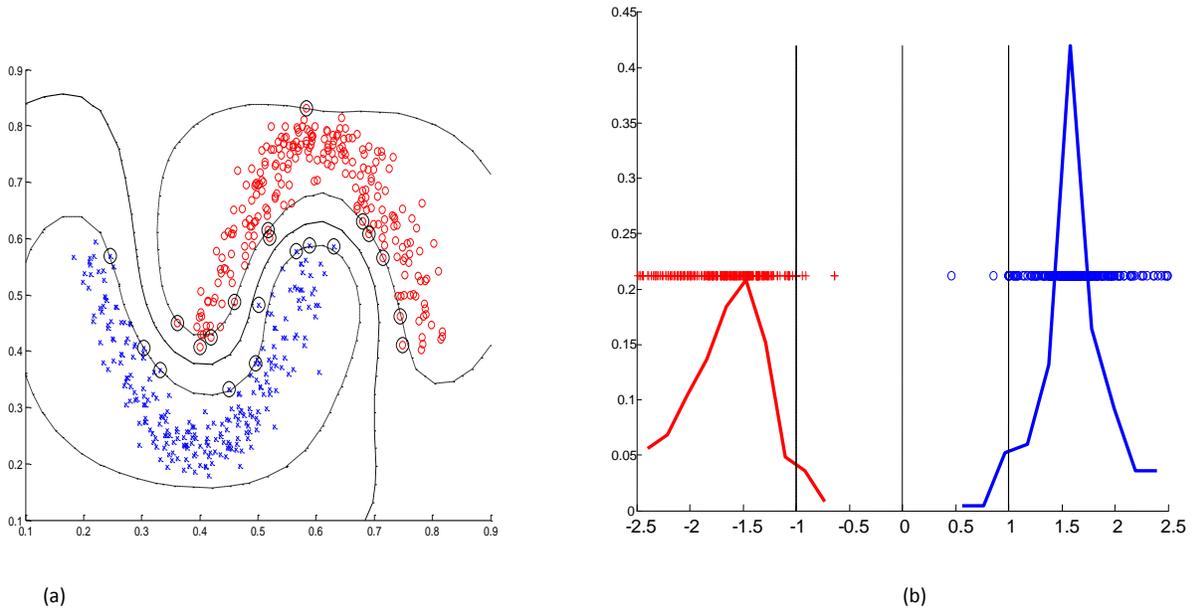
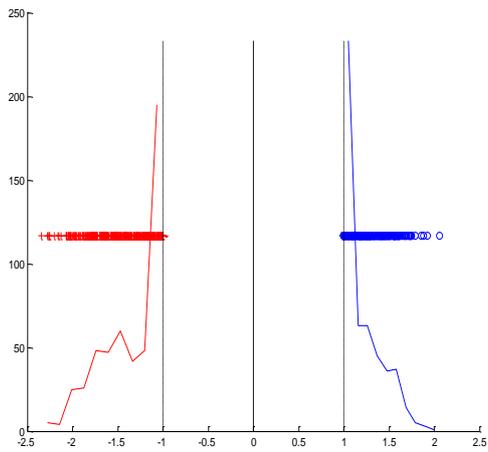
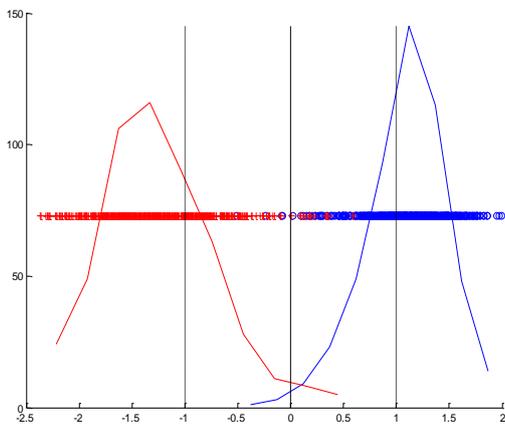


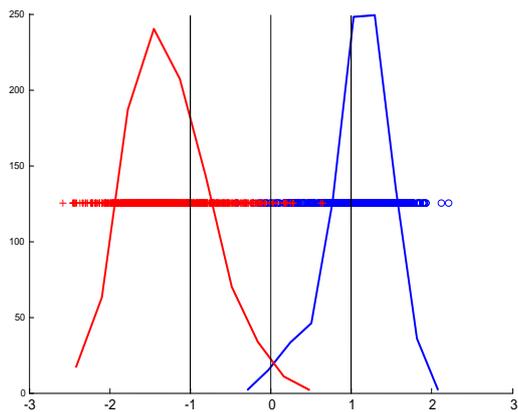
FIGURE 5]. (a) Decision boundary for non-linear RBF SVM model $y = \text{sign}(f(\mathbf{x})) = \text{sign}(\sum_{i \in S.V} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$. (b) Univariate histogram of projections for training samples onto the normal vector of SVM decision boundary.



(a)



(b)



(c)

FIGURE 6]. Univariate histogram of projections for MNIST data set (a) training data; (b) validation data (validation error 1.7%); (c) test data (test error 1.23%).

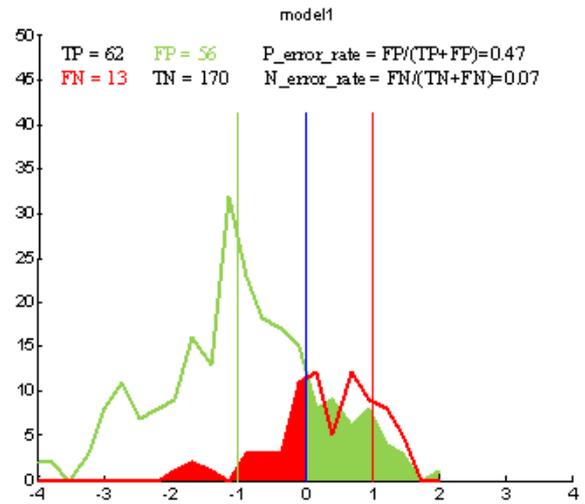


FIGURE 7]. Univariate histogram of projections of high-dimensional data onto normal vector w . Negative samples (\sim Alive) are shown in green, positive (\sim Dead) in red. False positives and false negative errors are indicated as green-shaded and red-shaded areas, respectively.