

Visualization and Interpretation of SVM Classifiers

Sauptik Dhar and Vladimir Cherkassky, Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455 USA (e-mail: dhax007@umn.edu, cherk001@umn.edu).

Abstract— Many machine learning applications involve modeling 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). This paper provides a brief survey of the current techniques for the visualization and interpretation of SVM based classification models, and then highlights potential problems with such techniques. Finally, we present a simple graphical method for visualization and understanding of SVM models.

Keywords— Interpretation of black-box models, model selection, support vector machines, histogram of projections.

I. 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 data [1].

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 existing techniques for visualization and interpretation of SVM classifiers and points out several problems 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.

II. Methods for Visualization and Interpretation of SVM Models

SVM classifiers are well-known for their robustness and good prediction capabilities; however, they are usually viewed as black-box modeling methods. Two important aspects for the interpretation of SVM black-box models are Rule Extraction and Visualization.

A. 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,

- 1) *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,
 - a) SVM+ Prototype [2]
 - b) RuleExSVM [3]
 - c) HRE algorithm [4]
 - d) Rule extraction from linear SVMs [5]
 - e) SQReX-SVM [6]
- 2) *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 rules [7]. The assumption is that such an eclectic rule induction procedure closely approximates the original SVM model. The methods that follow this approach include,
 - a) Learning-based rule-extraction from support vector machines [8]
 - b) Eclectic Rule-Extraction from SVM [9]
 - c) Active Learning Based Algorithm (ALBA) [10]

Detailed descriptions of these methods can be found in surveys [10-12]. 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)

GIVEN: SVM model estimated from the training data.

GENERATE: additional 1,000 samples with input (\mathbf{x}) values near the support vectors and y -values (class labels) predicted by the SVM classifier.

APPLY: a rule induction algorithm to the combined data set containing both training samples and newly generated samples.

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

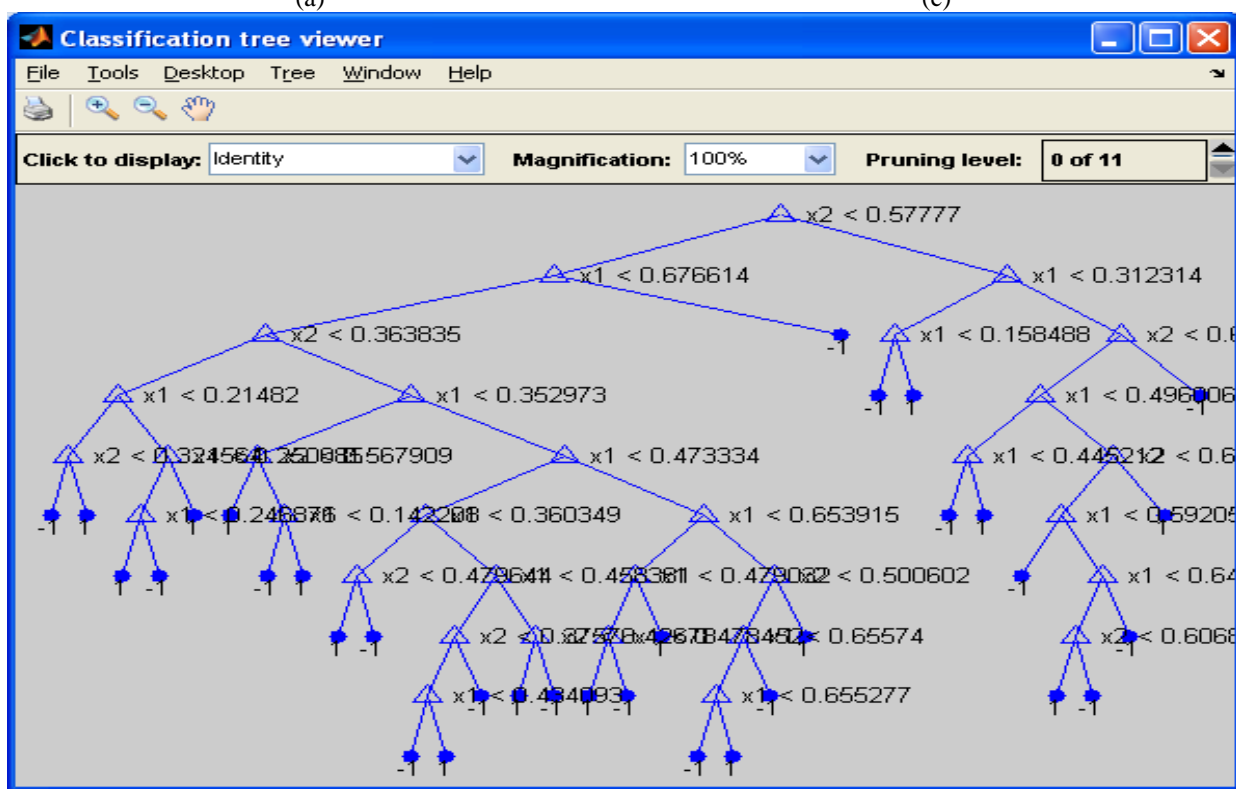
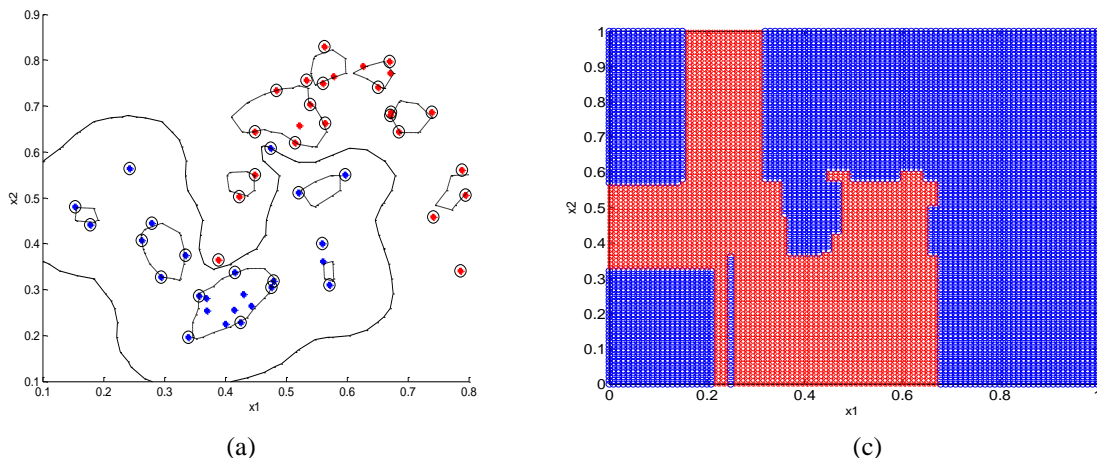
- How unique are the rules extracted by these methods from available data?
- Do these interpretable models provide the same generalization performance as the original SVM model?

The latter issue becomes especially critical for modeling High-Dimensional Low Sample Size (HDLSS) data.

The first issue is illustrated next using a simple example. Consider the synthetic 2 dimensional *Noisy Hyperbolas* data, where the underlying distributions for two classes are specified as follows: $x_2 = [(x_1-0.4)*3]^2+0.225$ 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 -1. Gaussian noise is added to the x_2 coordinates for both classes.

The degree of data separation is controlled by noise level $\sigma = 0.025$. For this experiment,

- Number of training samples = 50. (25 per class).
- Number of validation samples = 50. (This independent validation set is used for tuning SVM parameters).
- Number of test samples = 2000 (1000 per class).



(b)

Fig. 1. Illustration of the ALBA algorithm using the RBF SVM ($C = 4$ and $\gamma = 256$) 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.

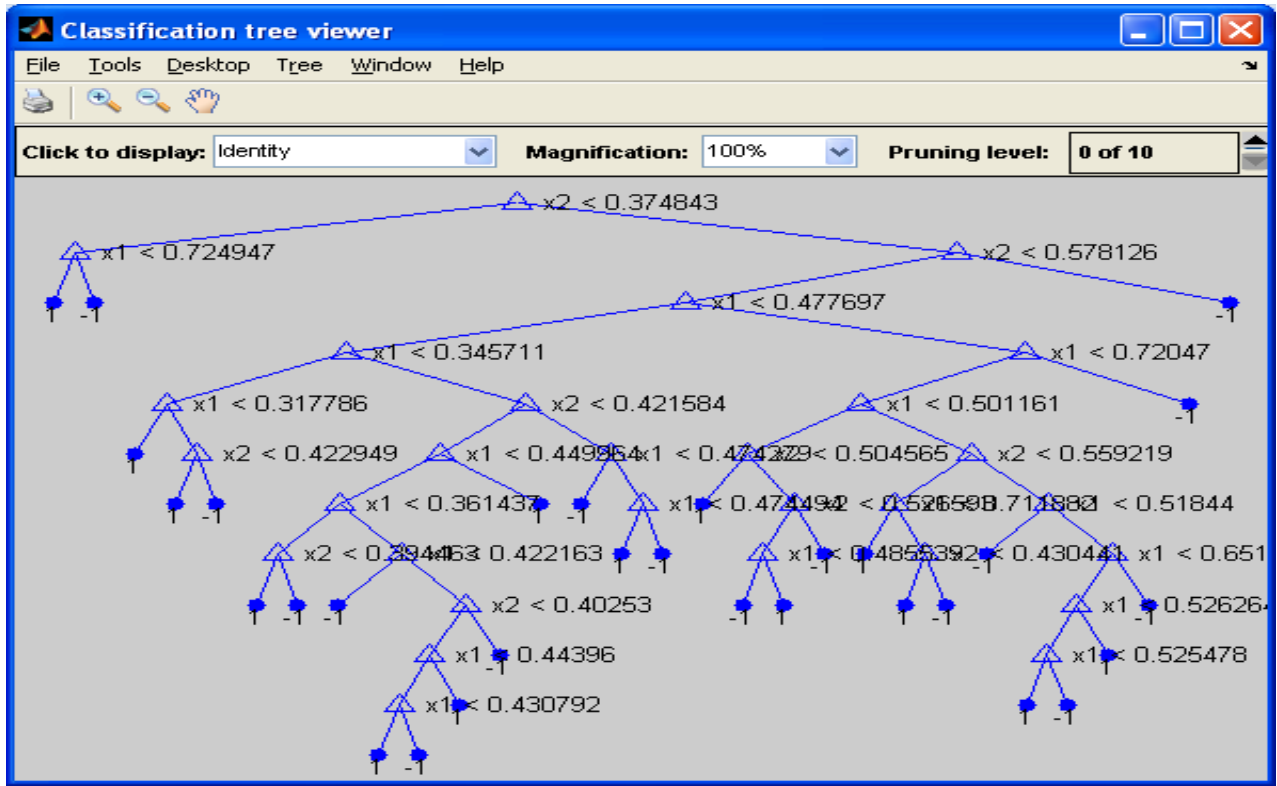
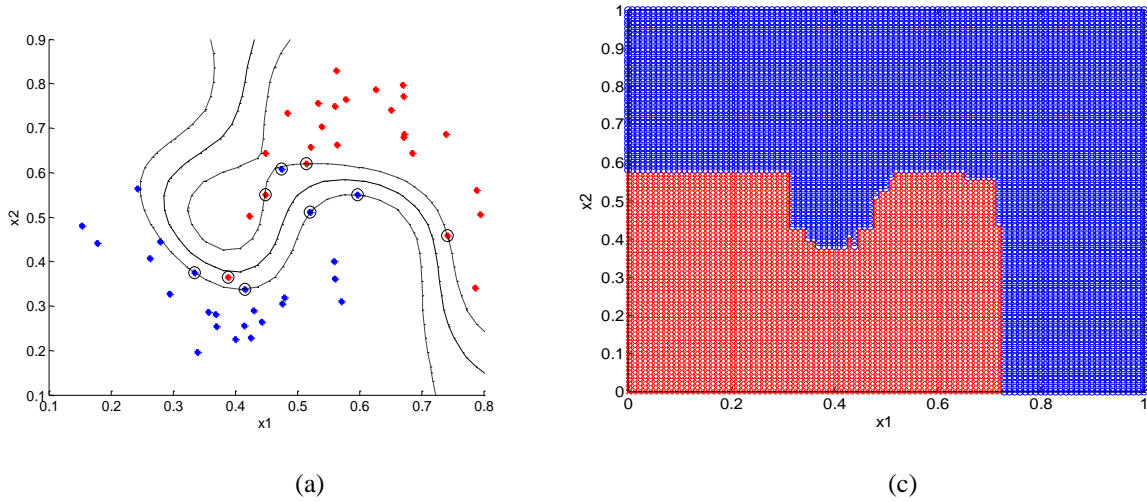
TABLE 1 Performance of ALBA and SVM for Hyperbolas data set.
Both SVM models have the same validation error 2%.

METHODS	RBF ($C = 4$ and $\gamma = 256$)		RBF ($C = 128$ and $\gamma = 16$)	
	SVM	ALBA	SVM	ALBA
Training Error (%)	0	0	4.0	4.0
Test Error (%)	5.20	5.25	5.05	5.10

For this dataset, we use the RBF kernel of the form $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$. We select two different SVM models which provide the same validation error 2%. The first model with $C = 4$ and $\gamma = 256$ yields the SVM decision boundary shown in Fig. 1a. The ALBA decision tree for this SVM model (shown in Fig. 1b) partitions the input space into regions (shown in Fig. 1c). The same steps are also performed for the second model with $C = 128$ and $\gamma = 16$ and the corresponding SVM and ALBA models are shown in Fig. 2. The training and test errors for both SVM and ALBA for the two models are provided in Table 1.

As evident from Table 1, the prediction accuracy of the SVM and its ALBA model is quite similar for both the models. 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 a significant 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 $28 \times 28 = 784$. On average, 22% of the input features are non-zero which makes this data sparse. For this example we use the following setting,



(b)

Fig. 2. Illustration of the ALBA algorithm using the RBF SVM ($C = 128$ and $\gamma = 16$) for the Noisy Hyperbolas data ($\sigma = 0.025$). (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.

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

- 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 $K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}')^d$ (with $d=3$) 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.

Results in Table 2 show that the SVM model and the corresponding ALBA tree have the same (zero) training error. 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*. Such approaches do not yield accurate predictive models for high-dimensional data.

The deficiencies of SVM interpretation techniques presented in this section are due to the fact that such methods have been developed for *large-sample* settings typically used in classical statistics. In particular, traditional statistical approach assumes that observed data has the structure:

$$\text{Observed data} \sim \text{Deterministic Model} + \text{Noise}$$

So the goal of data modeling is to estimate the deterministic part (or *true model*). This goal is usually pursued under various probabilistic assumptions about noise distribution. This view reflects the philosophy of modeling simple systems, where an estimated model is causal and unique. In this case, a true model, or its close approximation, can be accurately estimated given sufficient number of data samples. Since the true model is unique, it can be interpreted.

Under predictive learning approach, the goal is to estimate a model that can predict well the system's output [1, 14]. That is, the goal is to estimate certain properties of unknown statistical distribution, which are important for accurate prediction. However, this task does not require estimation of the *true probabilistic model*. Also, an estimated predictive model may not be unique, as demonstrated earlier in this section.

Further, classical model interpretation techniques use (rule-based) model parameterization, which assumes that a good model can be approximated using a small number of simple logic rules. This assumption holds true for many low-dimensional problems. However, rule-based parameterization may not be suitable for sparse high-dimensional data. SVM methodology advocates using a different parameterization using support vectors and kernel functions. For high-dimensional models, this SVM parameterization cannot be accurately represented by a small set of rules, as illustrated by the digit recognition example presented in this section.

B. Visualization of the SVM models

Very few papers have addressed the problem of understanding the SVM classifiers based on graphical representation [15-17]. 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. [15] 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 software [18]. 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. introduce another method called support vector machine visualization (SVMV) [16]. 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 described next 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 method.

More recent examples of using the idea of projecting the data onto the normal direction of SVM decision boundary are discussed in [1, 17, 19, 20]. 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 Mulier [1] and by Ahn and Marron [19]. Poulet used this method mainly for feature selection and for SVM parameter tuning [17]. In the paper by Platt [20], this method is used to visualize the class conditional densities, in order to determine a parametric form for the posterior density of the

class distributions $P(\text{class} | \text{SVM_model})$. This paper advocates the method of projections for improved understanding of SVM models. The main motivation is that such univariate histograms of projections reflect well-known properties of SVM-based classifiers. These properties are presented in a simple graphical form, which is quite useful for understanding *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 ~ 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.

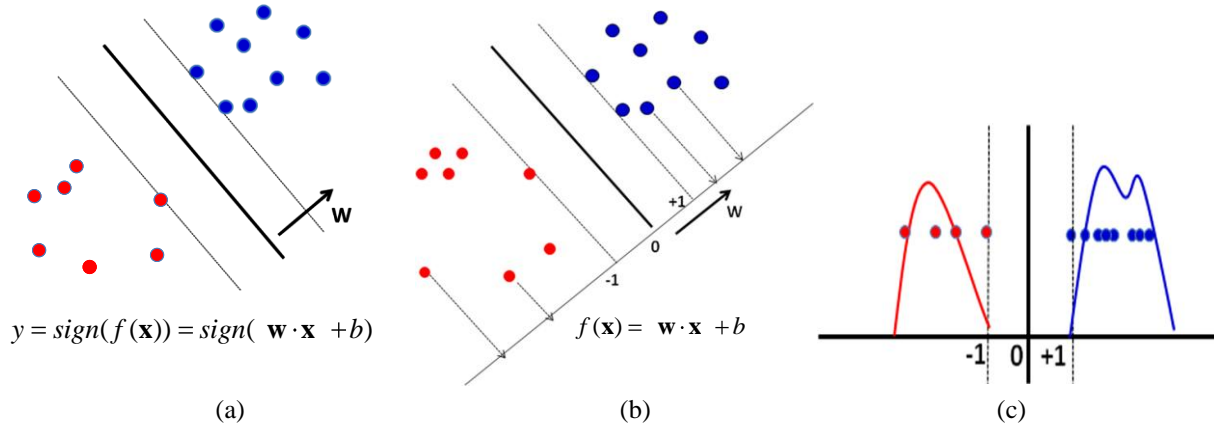


Fig. 3. Illustration of the steps to generate the univariate histogram of projections. (a) The estimated SVM model and training data.(b) Projection of the training data onto the normal weight vector (\mathbf{w}) of the SVM hyperplane.(c) Univariate histogram of projections. i.e. histogram of $f(\mathbf{x})$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

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.

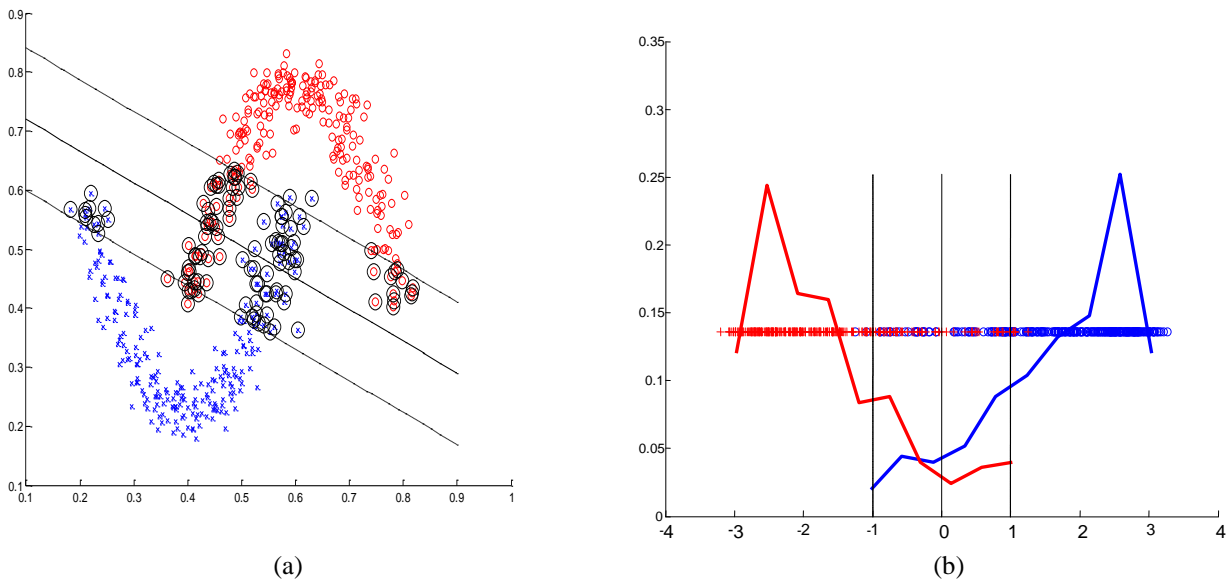


Fig. 4. (a) Decision boundary for linear SVM model $y = \text{sign}(f(\mathbf{x})) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$. (b) Univariate histogram of projections for training samples \mathbf{x}_k onto the normal vector of linear SVM decision boundary $f(\mathbf{x}_k) = (\mathbf{w} \cdot \mathbf{x}_k)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

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 separ –

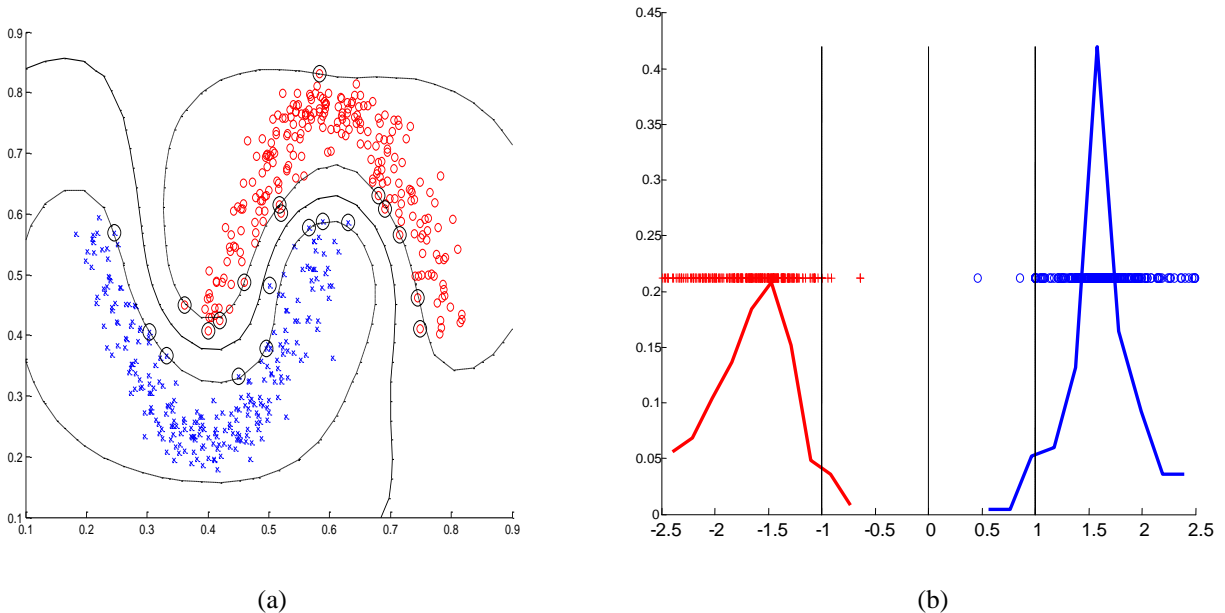


Fig.5. (a) Decision boundary for non-linear RBF SVM model $y = \text{sign}(f(\mathbf{x})) = \text{sign}(\sum_{i \in S.v} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$.

(b) Univariate histogram of projections for training samples onto the normal vector of SVM decision boundary.

able 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 SVM classifiers.

For example, consider the sparse high dimensional *MNIST handwritten digit data set* described in Section II. For this data set, we use the RBF kernel and tune the SVM parameters on an independent validation set according to the following model selection algorithm.

For this data set, typical *optimal* tuning parameter values are $C \sim 1$ or 2 and $\gamma \sim 2^{-6}$. 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 usually the case for HDLSS setting, where the training samples are 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], [19]. However, this separability of the training data does not imply separability of 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

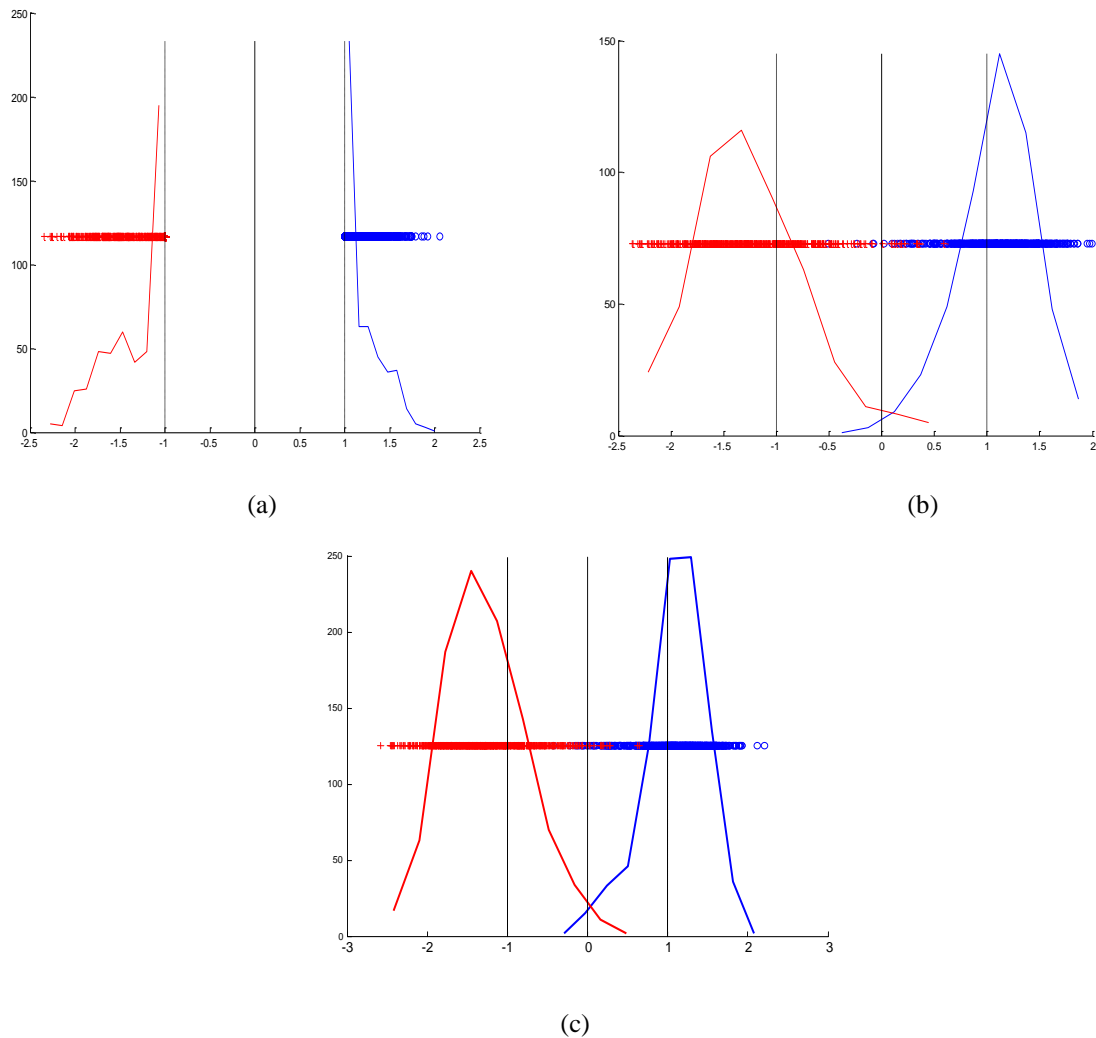


Fig. 6. Univariate histogram of projections for MNIST data set (a) training data; (b) validation data (validation error 1.7%); (c) test data (test error 1.23%).

Algorithm 2 Model Selection (using 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 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.

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} . In general, the misclassification costs are specified based on application-domain requirements [1].

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.

$$\text{weighted test error} = C_{fp}P_{fp} + C_{fn}P_{fn} \quad (1)$$

where P_{fp} and P_{fn} denote the probability (or fraction) of false positive and false negative errors in the test set respectively [1]. 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 patients [21]. 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 following 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 TRM [21]. 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

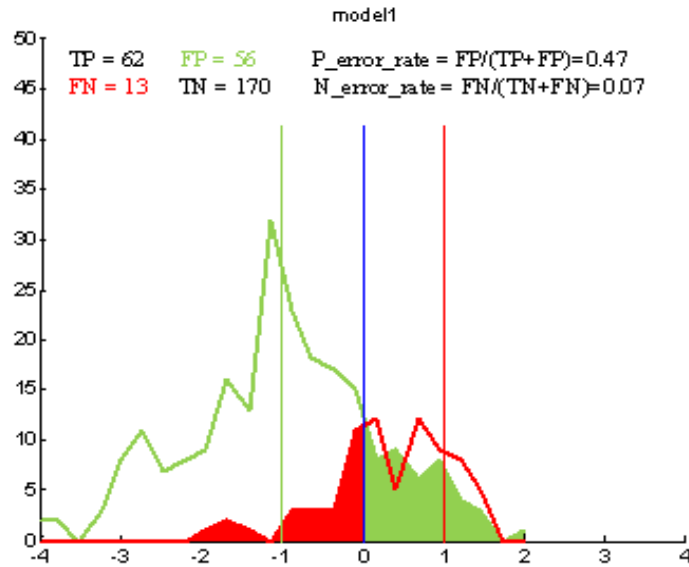


Fig. 7. Univariate histogram of projections of high-dimensional data onto normal vector w . Negative samples (\sim Alive) are shown in green, positive (\sim Dead) in red. False positives and false negative errors are indicated as green-shaded and red-shaded areas, respectively.

positive (\sim 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.

V. 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, 19], and improved understanding of SVM parameter tuning and SVM modeling of unbalanced data sets. Recently, this method has been also used [22, 23] to explain the conditions for the effectiveness of a new learning methodology called Universum Learning [14]. 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.

References

- [1] V. Cherkassky, and F. Mulier, *Learning from Data Concepts: Theory and Methods*, 2nd ed. NY: Wiley, 2007.
- [2] H. Nunez, C. Angulo, and A. Catala, "Rule-Extraction from Support Vector Machines," *Proc. European Symp. Artificial Neural Networks*, pp. 107-112, 2002.
- [3] X. Fu, C. Ongt, S. Keerthi, G. Hung, and L. Goh, "Extracting the Knowledge Embedded in Support Vector Machines," *Proc. IEEE Int'l Conf. Neural Networks*, pp. 291-296, 2004.
- [4] Y. Zhang, H. Su, T. Jia, and J. Chu, "Rule Extraction from Trained Support Vector Machines," *Proc. Ninth Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining*, pp. 61-70, 2005.
- [5] G. Fung, S. Sandilya, and R. Rao, "Rule Extraction from Linear Support Vector Machines," *Proc. 11th Int'l Conf. Knowledge Discovery and Data Mining*, 2005.
- [6] N. Barakat, and A. Bradley, "Rule Extraction from Support Vector Machines: A Sequential Covering Approach," *IEEE Trans. Knowledge and Data Eng.*, vol. 19, no. 6, pp. 729-741, June 2007.
- [7] V. Cherkassky, and F. Muller, *Learning from Data: Concepts, Theory, and Methods*. Wiley, 1998.
- [8] N. Barakat, and J. Diederich, "Learning-Based Rule-Extraction from Support Vector Machines: Performance on Benchmark Data Sets," *Proc. Conf. Neuro-Computing and Evolving Intelligence*, 2004.
- [9] N. Barakat, and J. Diederich, "Eclectic Rule-Extraction from Support Vector Machines," *Int'l J. Computational Intelligence*, vol. 2, no. 1, pp. 59-62, 2005.
- [10] D. Martens, B. Baesens, and T. Gestel, "Decompositional Rule Extraction from Support Vector Machines by Active Learning," *IEEE Trans. on Knowledge and Data Engineering*, vol. 21, no. 2, 2009.
- [11] N. Barakat, and A. Bradley, "Rule-extraction from support vector machines: A review," *Neurocomputing*, March 2010.
- [12] J. Diederich, *Rule Extraction from Support Vector Machines*, Springer 2008.
- [13] Sam Roweis, "sam roweis:data,"[Online].Available: <http://www.cs.nyu.edu/~roweis/data.html>[Accessed: May 5, 2010] .
- [14] V. N. Vapnik, *Estimation of Dependencies Based on Empirical Data. Empirical Inference Science: Afterword of 2006*. New York: Springer, 2006.
- [15] D. Caragea, D. Cook, and V. G. Honavar, "Gaining insights into support vector machine pattern classifiers using projection-based tour methods," *SIGKDD*, 2001, 251-256.

- [16] X. Wang, S. Wu, and Q. Li, "SVMV- a novel algorithm for the visualization of SVM classification results," *Advances in Neural Networks - ISNN 2006*, Berlin Heidelberg: Springer-Verlag, 2006, pp.968-973.
- [17] F. Poulet, "SVM and graphical algorithms: a cooperative approach," *Proceedings. Fourth IEEE International Conference on Data Mining*, 2004, pp. 499-502.
- [18] D. Cook, and D. F. Swayne, *Interactive and Dynamic Graphics for Data Analysis: With Examples Using R and GGobi*, NY: Springer,2007.
- [19] J. Ahn, and J.S. Marron, "The Maximal Data Piling Direction for Discrimination", *Biometrika.*, vol. 97, 254–259, 2010.
- [20] J. C. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods," *In Advances in Large Margin Classifiers*, 1999, pp. 61-74.
- [21] C. Feng, V. Cherkassky, D. Weisdorf, M. Arora, and B. Van Ness, "Predictive modeling of Transplant-Related Mortality," *Proc. of the 2010 Design of Medical Devices Conf.*, Minneapolis, April 2010.
- [22] V. Cherkassky, and W. Dai, "Empirical Study of the Universum SVM Learning for High-Dimensional Data," in *Proc. ICANN 2009*.
- [23] V. Cherkassky, S. Dhar, and W. Dai, "Practical Conditions for Effectiveness of the Universum Learning," *IEEE Trans. on Neural Networks*, 2011, to appear.