Practical Conditions for Effectiveness of the Universum Learning

UNIVERSITY OF MINNESOTA

This report is submitted to Professor Hui Zou in fulfillment of the requirements for the Course Project for STAT 5931.

PROJECT DONE BY: - SAUPTIK DHAR
STUDENT ID: - 3909558
EMAIL ID: dharx007@umn.edu

CONTENTS PAGE

1. INTRODUCTION 2
2. UNIVARIATE HISTOGRAM OF PROJECTIONS 3
3. UNIVERSUM LEARNING 9
   3.1. Practical conditions for the effectiveness Random Averaging (RA) U-SVM 12
   3.2. Practical conditions for the effectiveness U-SVM (in general) 16
4. CONCLUSION 20

CHAPTER 1. INTRODUCTION

Many applications in machine learning involve analysis of sparse high-dimensional data, in which the number of input features is larger than the number of data samples $(d \gg n)$. For example, in micro-array data analysis, technologies have been designed to measure the gene expression levels of tens of thousands of genes in a single experiment. However, the sample size in each data set is typically small ranging from tens to low hundreds due to the high cost of measurements. Such high-dimensional low sample size (HDLSS) problems represent new challenges for classification methods. Most approaches to learning with HDLSS data focus on improving existing inductive methods that try to incorporate a priori knowledge about the optimal model. Common examples include:

- clever preprocessing and feature extraction techniques that incorporate application-domain knowledge into the selection of a small number of informative features;
- selection of good kernels in SVM methods;
- specification of the prior distributions in Bayesian methods.

These techniques have been successfully used in many real-life applications [1]. Another approach to such ill-posed high-dimensional problems is to use non-standard learning settings that incorporate a priori knowledge about application data and/or the goal of learning directly into the problem formulation. One such non-standard learning methodology is the '*inference through contradictions*' or Universum Learning [2]. This methodology incorporates a priori knowledge about admissible data samples, which are used along with labeled training samples, for estimating a classifier. These admissible data samples are called Universum samples. Further, it is known that Universum samples do not belong to either binary class. Next we present two examples of Universum samples, in the context of real-life applications:

- consider the medical diagnosis of 'hypo-thyroid' disease [3]; where the goal is to estimate a binary decision rule for discriminating between 'hypo-thyroid' and 'normal' patients based on their demographic characteristics (age/sex), clinical test results etc. This decision rule is estimated using past medical records of correctly diagnosed patients, i.e., labeled training set. Suppose that we also have available data for the 'hyper-thyroid' patients. Although these 'hyper-thyroid' patients cannot

be assigned to any of the two classes ('hypo-thyroid' or 'normal'), they contain certain information about the thyroid disease. This additional information can be used to extract better decision rules for diagnosis (classification) of 'hypo-thyroid' disease.

– consider the task of hand-written digit recognition 5 vs.8 [1]. Under standard inductive learning setting, one has to estimate the class decision boundaries from labeled examples of handwritten digits. Then the prediction accuracy of a classifier is measured using an independent test set. However, along with labeled training data (i.e., handwritten digits) one has additional a priori information in the form of other handwritten digits '1', '2'…so on. These handwritten digits reflect the style of handwriting and can potentially improve generalization.

This leads to a setting where these unlabeled universum samples can be used to introduce additional apriori knowledge about the data samples into the learning methodology. Such a setting is called *Learning through Contradiction*, or learning in the Universum environment [2]. Such a non-standard learning setting reflects properties of real-life applications, and can result in improved generalization, relative to standard inductive learning as in standard SVM. However, such new methodologies are more complex, and their advantages and limitations are not well understood. This project aims to understand when Universum learning is preferred over the standard SVM learning based on some simple conditions using a graphical tool called the univariate histogram of projections.

CHAPTER 2. UNIVARIATE HISTOGRAM OF PROJECTIONS

One of the typical issues for HDLSS is that it is hard to have some intuition of the SVM based predictive models and the property of the underlying dataset. So there is a need for a simple method for visualizing the SVM models. Very few papers have addressed the problem of understanding the SVM classifiers based on graphical representation [4-6]. 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. [4] 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 [7]. 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. However, in most applications this is not the case, and such interactive methods can pose 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) [5]. 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 membership of the SOM units using the SVM model. However, such a method may complicate the task of the human modeler, as it requires selection of several SOM tuning parameters. Further, this method does not provide any visual representation of the soft margins.

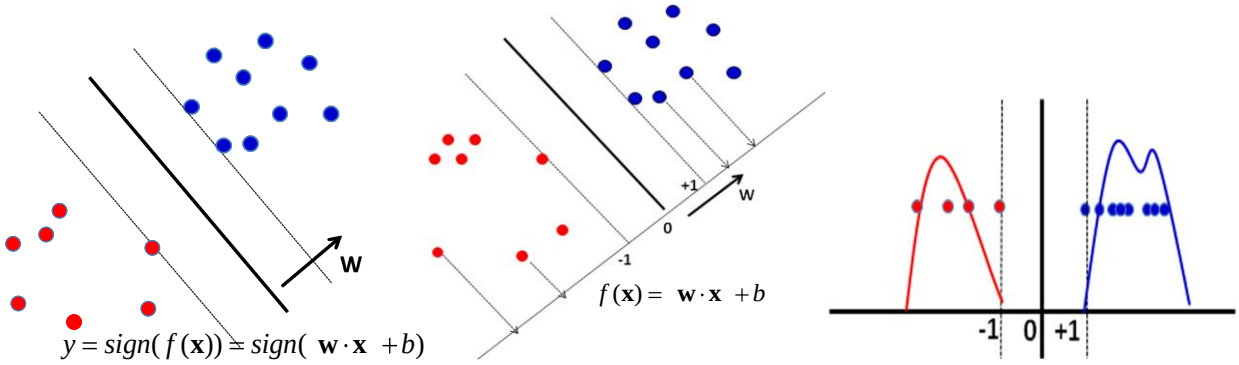
In this project I propose the idea of projecting the data onto the normal direction of the (SVM) classification decision boundary. The idea itself is very simple, and it has been widely used, under different contexts, in statistics and machine learning. Perhaps the original application of this idea dates 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 [1], [6], [8] and [9]. 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" in [1] and [8]. In [6], this method has been used mainly for feature selection and for SVM parameter tuning. In [9], this method is used to visualize the class conditional densities and hence determine a parametric form for the posterior density. This project advocates the method of projections for improved understanding of SVM models.

Histogram of Projections *~ is the histogram of the projection values of the data samples onto the normal weight vector of the SVM decision boundary.*

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, kernel).
b. Generate low-dimensional representation of training data by projecting it onto the normal direction vector of the SVM hyper plane estimated in (a).
c. Generate the histogram of the projected values obtained in (b).

These three steps are illustrated in Fig. 1.



(a) (b) (c)

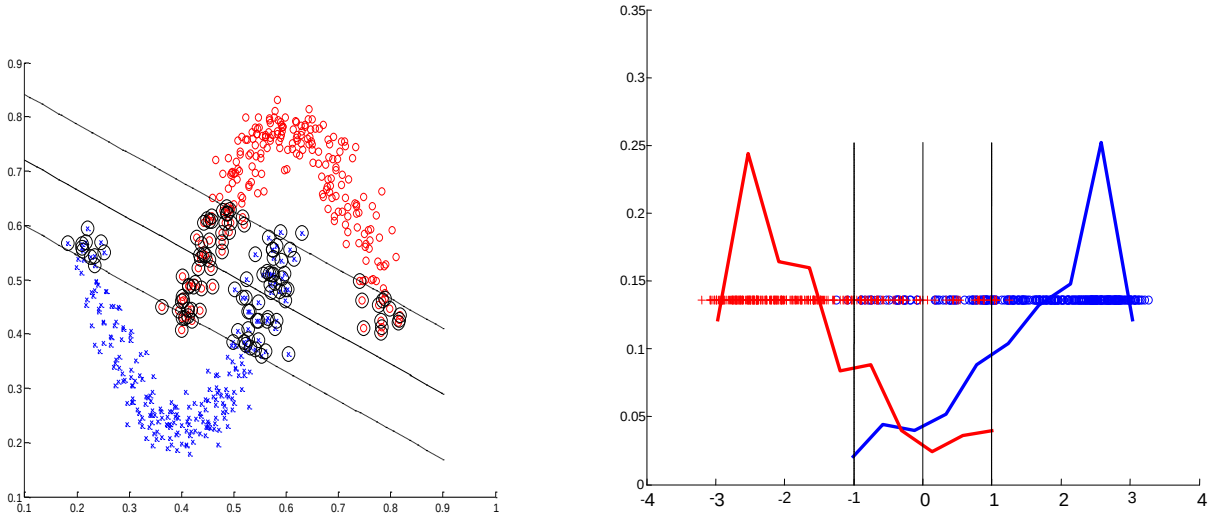
Fig.1. Illustration of the steps to generate the univariate histogram of projections.(a) The estimated SVM model and training data.(b) Projection of the training data onto the normal weight vector (**w**) of the SVM hyper plane.(c) Univariate histogram of projections. i.e. histogram of $f(\mathbf{x})$ values.

In Fig. 1c, 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 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,

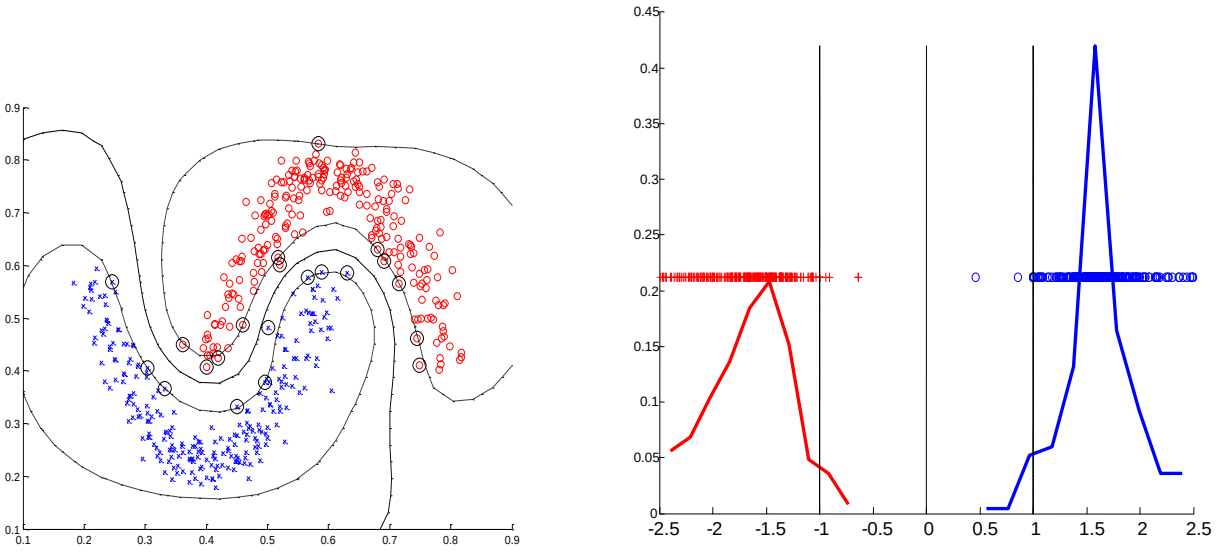
- Number of training samples= 100. (50 per class).
- Number of validation samples= 100. (This independent validation set is used for model selection).
- Number of test samples= 2000 (1000 per class).

The linear decision boundary for the synthetic *Noisy Hyperbolas* data set is shown in Fig. 2a. The projected values in Fig. 2b 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. 2b. 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 $f(\mathbf{x}_k) = \sum_i \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_k) + b$. For example, consider nonlinear decision boundary for the synthetic Noisy Hyperbolas data set in Fig. 3a.



(a) (b)

Fig.2. (a) Decision boundary for linear SVM model $y = sign(f(\mathbf{x})) = sign(\mathbf{w} \bullet \mathbf{x} + b)$. (b) Univariate histogram of projections for training samples $\mathbf{x}_k$ onto the normal vector of linear SVM decision boundary $f(\mathbf{x}_k) = (\mathbf{w} \bullet \mathbf{x}_k) + b$.



(a) (b)

Fig. 3. (a) Decision boundary for non-linear RBF SVM model $y = sign(f(\mathbf{x})) = sign(\sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b)$. (b) Univariate histogram of projections for training samples onto the normal vector of SVM decision boundary.

Using nonlinear RBF kernel of the form $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \left\| \mathbf{x} - \mathbf{x}' \right\|^2)$ with optimally tuned parameters yields SVM decision boundary shown in Fig. 3a, and the corresponding histogram of projections in Fig. 3b. Fig. 3b 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 two-dimensional data is clearly separable in the input space. However, for high-dimensional settings, visual representation of the data in the original input space, similar to Fig. 3a, 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 [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)

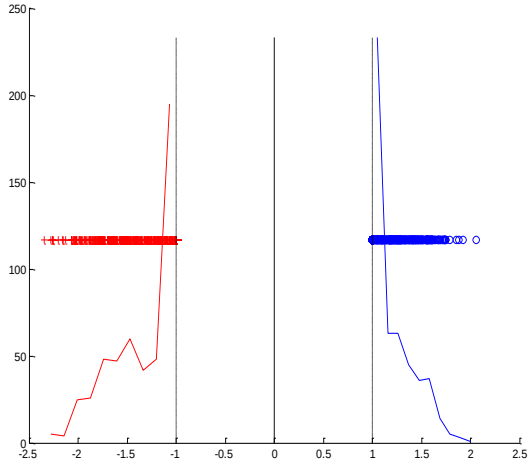
- Number of validation samples =1000. (This independent validation set is used for model selection).
- Number of test samples = 1866.

Further we use the nonlinear RBF kernel and tune the SVM parameters on an independent validation set according to the following model selection (Algorithm 1). For this digits data set typical tuning parameter values obtained are $C \sim 1$ or 2 and $\gamma \sim 2^{-6}$

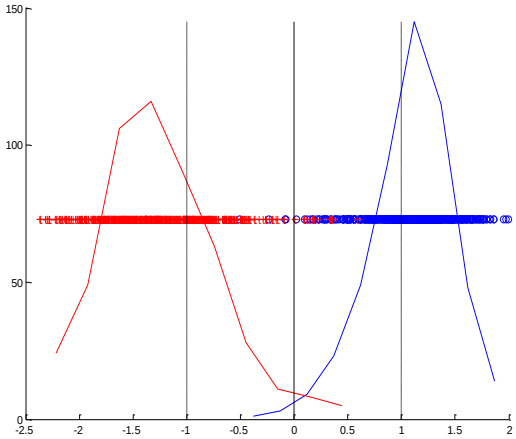
Algorithm 1 Model Selection (with independent validation set)

- 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.

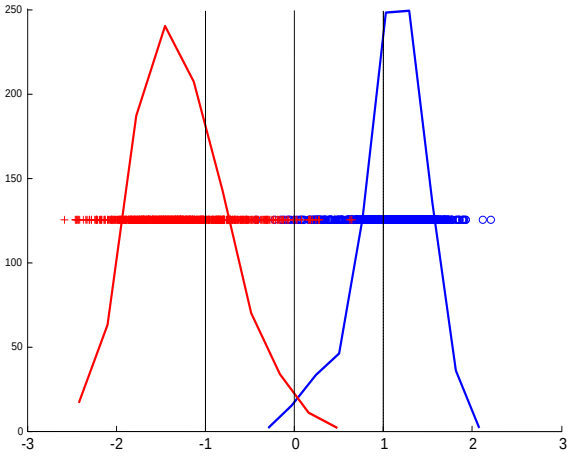
The univariate histogram of projections for the training data for the optimally tuned SVM model is shown in Fig. 4a. As evident from Fig. 4a, the training samples are well separable in this optimally chosen RBF kernel space. This is typically the case for HDLSS setting, where the training samples are 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, 8]. 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. 4b and Fig. 4c. 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. 4a where many training samples are outside the soft margin is typical. However, during model selection (in Algorithm 1), 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 result in the overlapping histograms for validation and test data, as shown in Fig. 4b and Fig. 4c. So the histogram of projections technique enables better understanding of the model selection strategy for tuning SVM parameters, for this data set. Thus this histogram technique may be quite helpful for simple yet effective understanding of the SVM based predictive models.



(a)



(b)



(c)

Fig. 4. 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%).

CHAPTER 3. UNIVERSUM LEARNING

The idea of 'inference through contradictions' was introduced by Vapnik [2] in order to incorporate a priori knowledge into the learning process. Recall that standard inductive learning methods introduce a priori knowledge about the space of admissible models. It may be argued that in real applications (especially with sparse high-dimensional data) such 'good' parameterizations are hard to come by. However, it may be feasible to introduce a priori knowledge about admissible data samples. These additional unlabeled data samples (called virtual examples or the Universum) are used along with labeled training samples, to perform an inductive inference. Examples from the Universum are not real training samples. However, they reflect a priori knowledge about application domain.

Next, we briefly review optimization formulation for the Universum SVM classifier [1,2]. Let us consider an inductive setting (for binary classification), where we have labeled training data and a set of unlabeled examples from the Universum. The Universum contains data that belongs to the same application domain as the training data, but these samples are known not to belong to either class. These Universum samples are incorporated into inductive learning as explained next. Let us assume that labeled training data is linearly separable using large margin. Then the Universum samples can either fall inside the margin or outside the margin borders (see Fig.5).

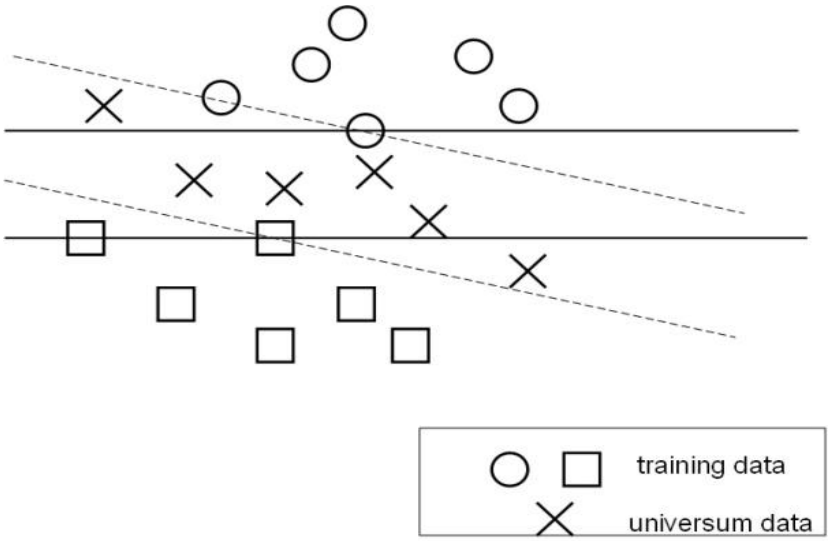


Fig. 5. Two large-margin separating hyperplanes explain training data equally well, but have different number of contradictions on the Universum. The model with a larger number of contradictions should be favored.

Note that we should favor hyperplane models where the Universum samples lie inside the margin, because these samples do not belong to either class. Such Universum samples (inside the margin) are called contradictions, because they are falsified by the model (i.e., have non-zero slack variables for either class label). The Universum learning implements a trade-off between explaining training samples (using large-margin hyperplanes) and maximizing the number of contradictions (on the Universum).

The quadratic optimization formulation for implementing an SVM-style inference through contradictions is shown next following [2]. For labeled training data, we use standard SVM soft-margin loss with slack variables ξ_i. For improved readability, we show only linear parameterization for the Universum SVM; however it can be generalized to the nonlinear case using kernels. For the Universum samples $\mathbf{x}_j^*$, we need to penalize the real-valued outputs of our classifier that are 'large'. This is accomplished using ε – insensitive loss (as in standard support vector regression). Let ξ_j^* denote slack variables for samples from the Universum. Then the Universum SVM formulation can be stated as:

$$\min_{\mathbf{w},b} R(\mathbf{w},b) = \frac{1}{2}(\mathbf{w}\cdot\mathbf{w}) + C\sum_{i=1}^{n}\xi_i + C^*\sum_{j=1}^{m}\xi_j^* \quad (1)$$

subject to constraints

for labeled data: $y_i[(\mathbf{w}\bullet\mathbf{x}_i)+b] \geq 1-\xi_i \qquad \xi_i, \geq 0, i=1,...,n$

for Universum data: $\left|(\mathbf{w}\cdot\mathbf{x}_j^*)+b\right| \leq \varepsilon+\xi_j^* \qquad \xi_j^* \geq 0,\ j=1,...,m \qquad$ where $C, C^* \geq 0; \varepsilon \geq 0$

Parameters C and C^* control the trade-off between minimization of errors and the maximization of the number of contradictions. Selecting 'good' values for these parameters constitutes model selection (usually performed via resampling). When C^*=0, this U-SVM formulation is reduced to standard soft-margin SVM.

The solution to the optimization problem (1) defines the large margin hyper plane $f(\mathbf{x}) = (\mathbf{w}^*\cdot\mathbf{x}) + b^*$ that incorporates a priori knowledge (i.e., Universum samples) into the final model. The dual formulation for inductive SVM in the Universum environment, and its nonlinear (kernelized) version can be obtained using optimization theory and standard SVM techniques, where the decision function in the dual space is constructed by using a kernel matrix of both the labeled samples and the Universum samples [2]. This quadratic optimization problem is convex due to convexity of the constraints for labeled data and for the Universum. Efficient computational algorithms for solving this problem involve modifications of standard SVM software [11]. The U-SVM software is available at http: //www.kyb.tuebingen.mpg. de/bs/people/fabee/universvm.html. For my project I have prepared the MATLAB interface to the universum executable file and this is available at http://www.ece.umn.edu/ users/cherkass/predictive_learning/SOFTWARES.html

The main objective of this project is to derive practical conditions for the effectiveness of Universum learning. Initial prior work [11], [15] focused on the algorithmic implementation of the U-SVM, and its empirical validation. These studies confirmed that Universum learning can improve generalization performance, especially for sparse high-dimensional data. However, the obtained performance strongly depends on a good choice of the Universum. More recent studies have proposed and analyzed criteria for a good choice of a Universum [12-14]. These studies provide different characterizations related to

the intuitive notion that a good Universum set should be positioned 'in between' the two classes, as illustrated in Fig. 1. Sinz et al [14] showed that the optimal decision boundary of the U-SVM tends to make the normal vector orthogonal to the principal direction of the Universum data set. This condition holds for both the original Vapnik's Universum formulation (1) and for the least-squares U-SVM, where the squared loss function is adopted for both labeled and Universum samples. Further, they show the connection (equivalency) between the least-squares U-SVM and the maximization of an explicit analytic criterion. Chen and Zhang [13], proposed a graph-theoretic index for measuring the 'in-betweenness' of Universum samples. However, they assume SSL framework, and use squared loss in their SVM-style optimization formulation. Their approach aims at selecting a portion of the Universum data set that is 'useful' for boosting generalization performance.

This work pursues the same general objective as [14], i.e. the characterization of a good Universum for Vapnik's original formulation (1). However, we take a more practical and specific approach. That is, we ask the following questions:

i. Can a given Universum data set improve generalization performance of standard SVM classifier trained using only labeled data?
ii. Can we provide practical conditions for (i), based on the geometric properties of the Universum data and labeled training data?

This approach is more suitable for non-expert users, because:

- practitioners are interested in using U-SVM only if it provides an improvement over standard SVM;
- the problem of (full-blown) model selection for the U-SVM is alleviated, because its two parameters (kernel parameter and C) are tuned separately, during training standard SVM classifier.

The proposed strategy for analyzing practical conditions for the effectiveness of the Universum is outlined below:

a. estimate standard SVM classifier for a given (labeled) training data set. Note that this step includes optimal model selection, i.e. optimal tuning of the regularization parameter C and kernel;
b. generate low-dimensional representation of training data by projecting it onto the normal direction vector of SVM hyperplane estimated in (a);
c. project the Universum data onto the normal direction vector of SVM hyperplane, and analyze projected Universum data in relation to projected training data.

Then statistical properties of the projected Universum data relative to labeled training data, in (c), may suggest whether using this Universum will improve the prediction accuracy of standard SVM estimated in step (a).

Selection of the Universum is usually application-dependent [2], [11]. However, there is a possibility of generating Universum data directly from labeled training data. This approach is called *random averaging* and it does not rely on a priori knowledge about application domain. For example for the problem of handwritten digit recognition, where the goal is to discriminate between handwritten digits 5 and 8, Fig.

6 shows two randomly chosen labeled examples and the corresponding Universum example obtained via averaging.

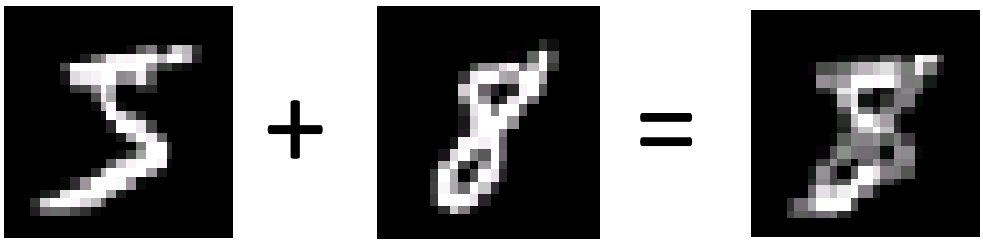


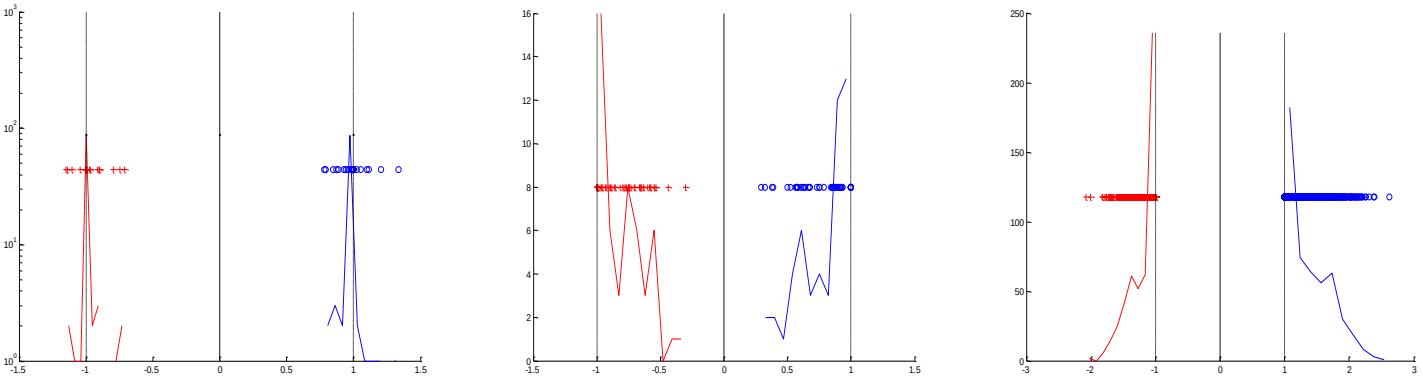
Fig. 6. Example of randomly chosen handwritten digits 5 and 8 and the corresponding Universum sample obtained by averaging.

In the following subsections we provide the practical conditions for effectiveness of U-SVM learning along with the results.

3.1 Practical conditions for the effectiveness Random Averaging (RA) U-SVM

As discussed before the RA universum samples are generated from the training samples. Hence, it makes sense to find practical conditions for the effectiveness of RA U-SVM based on the univariate histograms of the training samples. Typically for HDLSS settings we have mainly 3 cases,

- *Case 1:* univariate projections of the training data onto the normal direction vector of standard SVM model cluster strongly on margin borders (as in Fig. 7a).
- *Case 2:* univariate projections of the training data onto the normal direction vector of standard SVM model cluster inside margin borders, as shown in Fig. 7b.
- *Case 3:* univariate projections of the training data onto the normal direction vector of standard SVM model cluster outside margin borders, as shown in Fig. 7c.



(a) (b) (c)

Fig. 7. Typical histogram. (a) Case 1 (b) Case 2 (c) Case 3

Based, on our research findings RA-USVM is likely to out-perform the SVM only for the Case 3. From the nature of the U-SVM optimization formulation (1), it can be expected that RA Universum would not be effective for *Case 1*, because Universum samples will be narrowly distributed near SVM decision boundary, as shown in Fig. 7a. (However, other types of Universum, with a wider distribution, may be effective). For *Case 2*, the labeled training data cannot be separated with large margin, so the original motivation for Universum learning (to stabilize selection of a large-margin hyperplane) does not apply. So in this case, the Universum should not provide any improvement. However, U-SVM is expected to provide an improvement in *Case 3*, where random averaging would produce Universum samples widely distributed around SVM decision boundary in the projection space, and even possibly outside the margin borders of standard SVM. Note that histograms in Figs. 7 assume that the training data is linearly separable, which usually holds true for high-dimensional data. For lower-dimensional data, the separability can be often achieved using nonlinear kernels. We confirm our claim with the help of the following experiment.

EXPERIMENT 1 COMPARISON OF SVM with RA-USVM for 3 different datasets.

DATASETS USED

DATA 1.*Synthetic 1000-dimensional hypercube data set*, where each input is uniformly distributed in [0, 1] interval and only 200 out of 1000 dimensions are relevant for classification. An output class label is generated as $y = \text{sign}(x_1+x_2+\ldots+x_{200} - 100)$. For this data set, only linear SVM is used because the optimal decision boundary is known to be linear. The training set size is 1,000, validation set size is 1,000, and test set size is 5,000. For U-SVM, 1,000 Universum samples are generated via random averaging.

DATA 2.*Real-life MNIST handwritten digit data set*, where data samples represent the handwritten digits 5 and 8. Each sample is represented as a real-valued vector of size 28*28=784. On average, approximately 22% of the input features are non-zero which makes this data very sparse. The training set size is 1,000, validation set size is 1,000, and test set size is 1,866 samples. For U-SVM, 1,000 Universum samples are generated via random averaging.

DATA 3.*Real-life ABCDETC data set*, where data samples represent the handwritten lower case letters ‘a’ and ‘b’. Each sample is represented as a real-valued vector of size 100*100=10000. The training set size is 150 (75 per class), validation set size is 150 (75 per class). The remaining 209 samples are used as test samples (105 from class ‘a’ and 104 from class ‘b’). For U-SVM, 1,500 Universum samples are generated via random averaging.

METHODS DESCRIPTION

For each data set, a classifier is estimated using the training data, its model complexity is optimally tuned using validation data as shown in Algorithm 2, and finally the test error is estimated using test data. The results of such an experiment depend on a random realization of training and validation data. So each experiment is repeated 10 times, using different random realizations, and the average test error rates are reported for comparison.

- For Data 1 we use only the linear SVM as the underlying model is linear.
- For Data 2 we use both the linear as well as the RBF SVM.
- For Data 3 we use the polynomial SVM with degree d=3 (following [11]). A preliminary experiment has shown degree d=3 to be optimal.

Also, the following range of parameters is used during model selection: $C\sim[10^{-11},10^{-9},10^{-7},10^{-5},10^{-3},10^{-1},1,100]$, $C/C^*\sim[10^{-4},3x10^{-4},10^{-3},3x10^{-3},10^{-2},3x10^{-2},10^{-1},3x10^{-1},1,3]$ and ε= [0, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4]. Note that, we perform the model selection by first tuning the model parameters for SVM (i.e. C and kernel parameter) and then tuning the model parameters for the U-SVM by keeping the C and kernel parameter fixed.

Algorithm 2 Model Selection for U-SVM(with independent validation set)

(TUNE SVM PARAMETERS FIRST)

- 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.

(NOW TUNE U-SVM PARAMETERS)

- Keep (C, γ) fixed as obtained from tuning the SVM and then tune the U-SVM parameters (C*,ϵ), similarly as before.

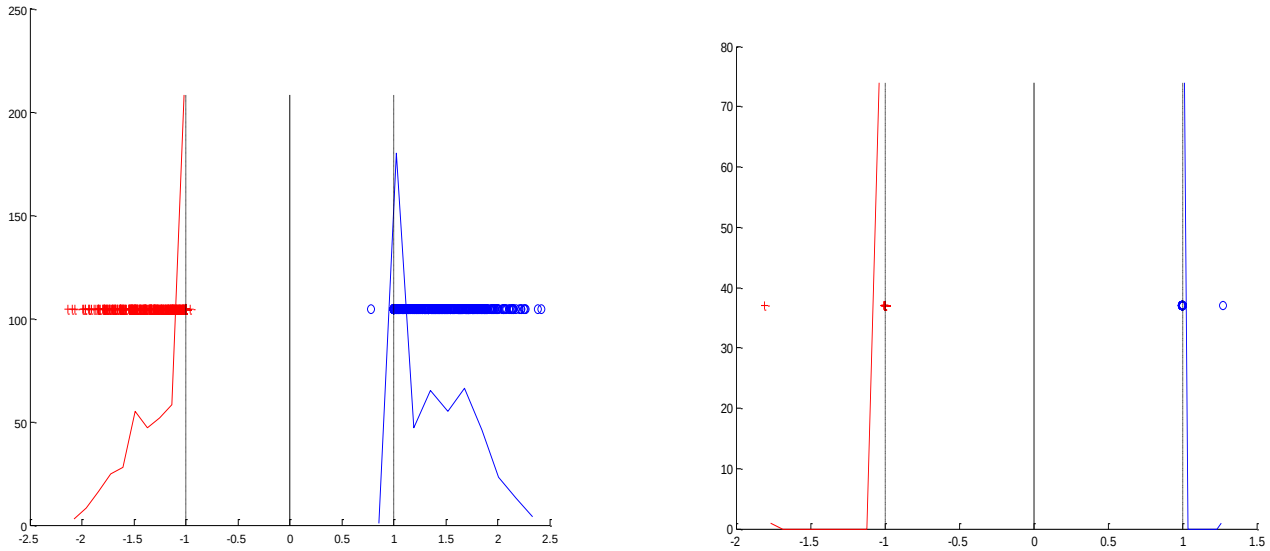
RESULTS

Generalization performance of standard SVM and U-SVM is shown in Table I, where the standard deviation of the estimated average test error is indicated in parentheses.

Table I Test error rates for MNIST, ABCDETC and synthetic data sets.

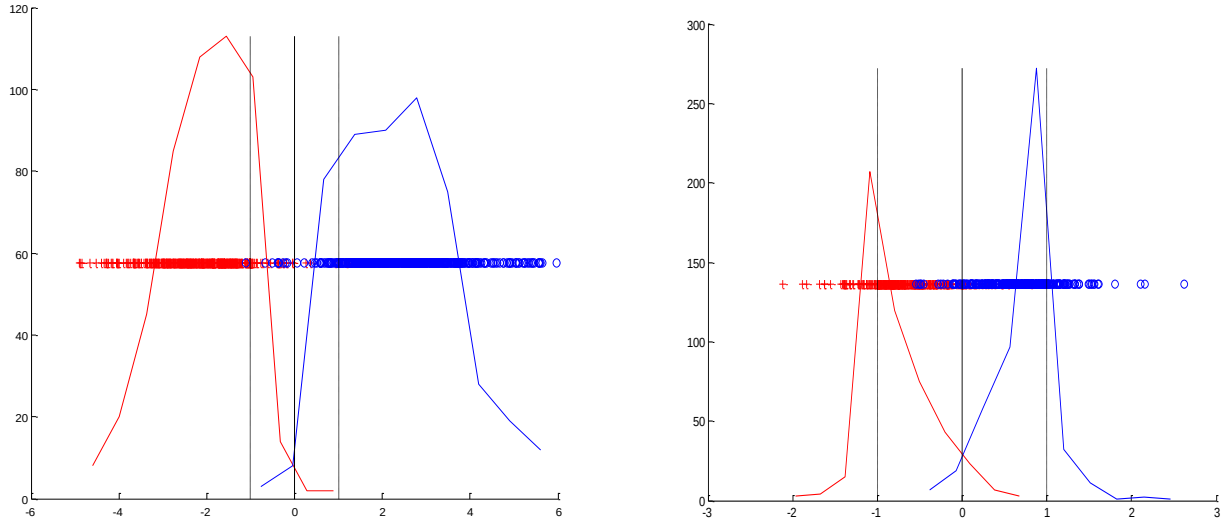
	SVM	U-SVM(RA)
Synthetic data (Linear)	26.63% (1.54%)	26.89% (1.55%)
MNIST(Linear)	4.58 %(0.34%)	4.62 %(0.37%)
MNIST (RBF Kernel)	1.37% (0.22%)	1.20% (0.19%)
ABCDETC (Poly)	20.48 %(2.60%)	18.85 %(2.81%)

Comparison results indicate that U-SVM yields an improvement over standard SVM for MNIST digits (when using RBF kernel) and for the ABCDETC data. These results can be explained by examining the histograms of projections. Fig. 8a shows the histogram of projections of training data onto the normal direction of the RBF SVM decision boundary for the MNIST data, suggesting that this data is well-separable. Similarly, Fig. 8b shows the histogram of projections onto the normal direction of the Polynomial SVM decision boundary for the ABCDETC data, suggesting that this data is also well-separable. On the other hand, the histogram of projections for linear SVM in Fig. 9 for both synthetic



(a) (b)

Fig. 8. (a). Histogram of projections of MNIST training data onto normal direction of RBF SVM decision boundary. Training set size ~ 1,000 samples. (b) Histogram of projections of ABCDETC training data onto normal direction of Polynomial SVM decision boundary. Training set size ~ 150 samples.



(a) (b)

Fig. 9. Histogram of projections onto normal direction of linear SVM. (a) MNIST data set.(b) synthetic data set.

and MNIST data indicate that the training data is not well-separable, so the RA Universum should not yield any improvement.

3.2 Practical conditions for the effectiveness U-SVM (in general)

As our next goal we investigate the effectiveness of other types of Universum. We found that a Universum data set will be effective if its histogram of projections satisfies two conditions:

(C1) It is symmetric relative to the (standard) SVM decision boundary, and

(C2) It has wide distribution between margin borders denoted as points -1/+1 in the projection space.

The mathematical intuition behind such conditions is provided in APPENDIX A. To confirm our claim we further provide our experimental results,

EXPERIMENT 2 COMPARISON OF SVM with USVM for MNIST DATA using digits '1', '3' and '6' as unlabeled Universum samples.

DATASETS USED

Real-life MNIST handwritten digit data set, where data samples represent the handwritten digits 5 and 8. Each sample is represented as a real-valued vector of size 28*28=784.

- 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.
- Number of Universum samples= 1000.

METHODS DESCRIPTION

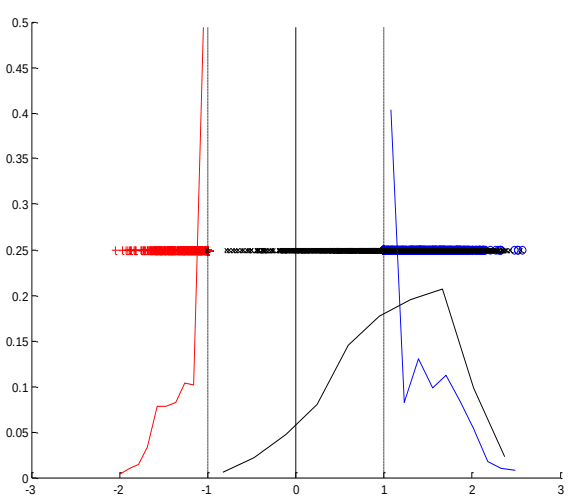
For both SVM and U-SVM the classifier is estimated using the training data, its model complexity is optimally tuned using validation data as previously shown in Algorithm 2, and finally the test error is estimated using test data. The experiment is repeated 10 times, using different random realizations, and the average test error rates are reported for comparison. We use the RBF kernel of the form $K(\mathbf{x},\mathbf{x}') = \exp(-\gamma\|\mathbf{x}-\mathbf{x}'\|^2)$. The range of parameters used during model selection: C~[10^{-11},10^{-9},10^{-7},10^{-5},10^{-3},10^{-1},1,100], C/C*~[10^{-4} ,3x10^{-4},10^{-3},3x10^{-3},10^{-2}, 3x10^{-2},10^{-1},3x10^{-1},1,3] ,γ=[2^{-8} ,…,2^{2}] and ε= [0, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4].

RESULTS

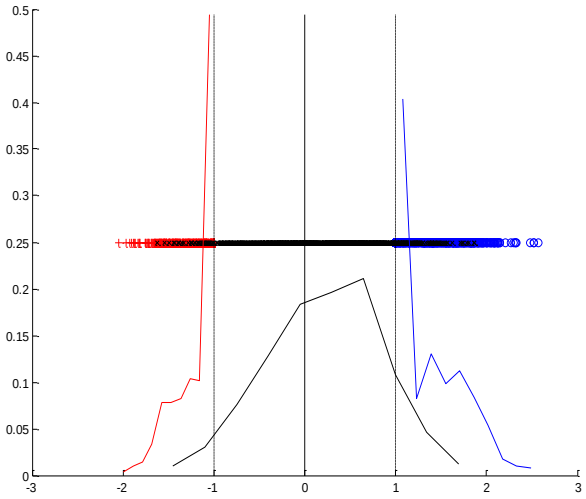
Table II Test error rates for MNIST data with different Universa. Training set size is 1,000 samples.

	SVM	U-SVM (digit 1)	U-SVM(digit 3)	U-SVM(digit 6)
Test error	1.47% (0.32%)	1.31% (0.31%)	1.01% (0.28%)	1.12% (0.27%)

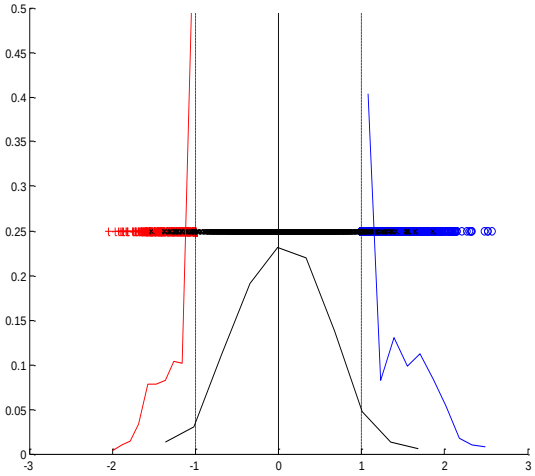
Analysis of the histograms in Fig. 10 confirms an earlier finding that digit 1 is not a good choice for the Universum, because its projections are more biased towards the distribution of digit 8.



(a)



(b)



(c)

Fig. 10. Univariate histogram of projections for 3 different types of Universa. Training set size ~ 1,000 samples. Universum set size ~ 1,000 samples. (a) digit 1 Universum. (b) digit 3 Universum. (c) digit 6 Universum.

Further, histograms shown in Fig. 10b and Fig. 10c satisfy conditions (C1)-(C2) for the effectiveness of Universum. Hence, we can expect both digits 3 and 6 Universa to yield an improved prediction accuracy over standard SVM, which is confirmed by the empirical results in Table II.

EXPERIMENT 3 COMPARISON OF SVM with USVM for ABCDETC DATA using 'All upper case letters from A to Z', 'All digits from 0 to 9' and 'Random Averaging' of training data as unlabeled Universum samples.

DATASETS USED

Real-life ABCDETC data set, where data samples represent the handwritten letters. Each sample is represented as a real-valued vector of size 100*100=10000.

- Number of training samples= 150. (75 per class)
- Number of validation samples =150. (This independent validation set is used for model selection).
- Number of test samples = 209.(105 samples from class 'a' and 104 from class 'b').
- Number of Universum samples= 1500.

METHODS DESCRIPTION

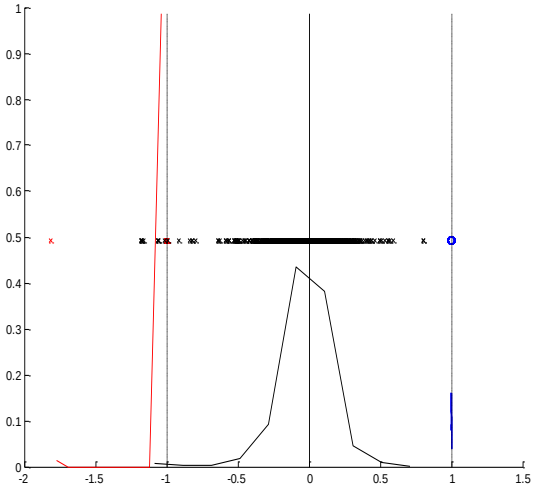
For both SVM and U-SVM the classifier is estimated using the training data, its model complexity is optimally tuned using validation data as previously shown in Algorithm 2, and finally the test error is estimated using test data. The experiment is repeated 10 times, using different random realizations, and the average test error rates are reported for comparison. We use the polynomial kernel of degree d=3. The range of parameters used during model selection: $C\sim[10^{-11},10^{-9},10^{-7},10^{-5},10^{-3},10^{-1},1,100]$, $C/C^*\sim[10^{-4},3x10^{-4},10^{-3},3x10^{-3},10^{-2},3x10^{-2},10^{-1},3x10^{-1},1,3]$ and ε= [0, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4].

RESULTS

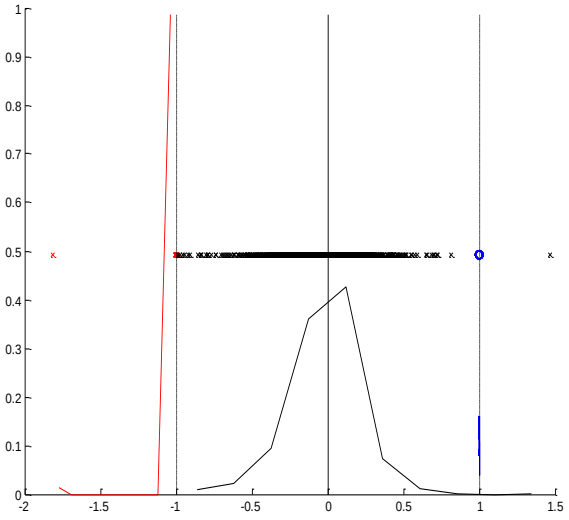
Table III Test error rates for ABCDETC data with different Universa. Training set size is 150 samples.

	SVM	U-SVM (upper case)	U-SVM (all digits)	U-SVM (RA)
Test error	20.47%(2.60%)	18.42 % (2.97%)	18.37 % (3.47%)	18.85 % (2.81%)

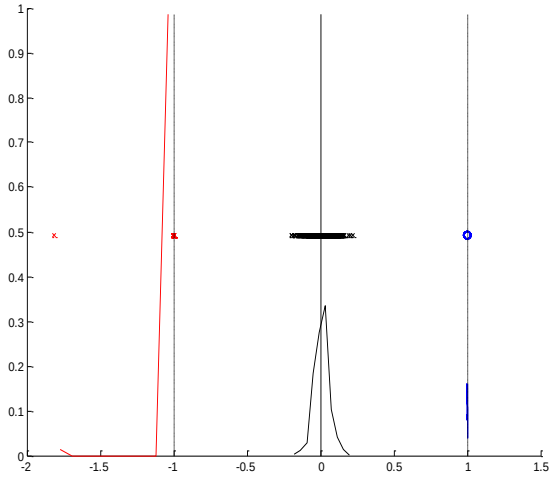
The histograms in Fig. 11 show that for both the 'Upper case letters A-Z' and 'digits 0-9' the Universum samples have a wider distribution than the Universum samples obtained via Random Averaging. Hence, we can expect both 'Upper case letters A-Z' and 'digits 0-9' to be more effective than RA. This is confirmed by the empirical results in Table III.



(a)



(b)



(c)

Fig. 11. Univariate histogram of projections for 3 different types of Universa for ABCDETC data Training set size ~ 150 samples. Universum set size ~ 1,500 samples. (a) 'Upper case letters A to Z' Universum. (b) 'digits 0-9' Universum. (c) RA Universum.

4. CONCLUSION

In general, performance of learning methods is always affected by the properties of application data at hand. New learning settings, such as U-SVM, are inherently more complex than standard SVM and they have more tuning parameters. So it is important to have practical criteria that ensure potential advantages of using U-SVM for a given data set. This is a difficult problem, because the effectiveness of U-SVM depends on the properties of labeled data as well as Universum samples. Meaningful analytic characterization of such data sets is quite difficult. So we propose a novel representation of training data using projections of this data onto the normal direction of SVM decision boundary. Analysis of the univariate histograms of projections, presented in this paper, leads to practical conditions for the effectiveness of Universum learning. That is, a Universum data set is effective, if its univariate histogram of projections is symmetric *and* widely distributed, relative to (standard) SVM decision boundary. The conditions are more useful for practitioners than analytic criteria in [14], because our approach:

- Provides an explicit characterization of the properties of the Universum and the properties of labeled training data. These properties are conveniently represented in the form of univariate histograms;
- Directly relates prediction performance of U-SVM to that of standard SVM (using only labeled data);
- Significantly simplifies model selection for U-SVM.

APPENDIX

In this document we provide the mathematical intuition behind our practical conditions for the Effectiveness of U-SVM learning. We derive an equivalent form for the U-SVM optimization formulation and provide connection between the optimization formulation and the practical conditions.

The U-SVM formulation can be given as (For simplicity we set $\varepsilon = 0$)

$$\min.\ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} H(y_i f(\mathbf{x}_i)) + C^* \sum_{j=1}^{m} \left| f(\mathbf{x}_j) \right| \quad \text{..... (1)}$$

where $i = 1...n$ (Total # of Training samples) and $j = 1...m$ (Total # of Universum samples)

Now let us consider, $f(\mathbf{x}_j) = f(\mathbf{x}_j) + \bar{u} - \bar{u} = \bar{u} + d_j$, where the mean of the projected values for the universum samples, $\bar{u} = \frac{1}{m}\sum_{j=1}^{m} f(\mathbf{x}_j)$ and the deviation of the projected value for the j[th] universum sample from the mean, $d_j = f(\mathbf{x}_j) - \bar{u}$.

Now, assuming that the universum samples are independent of each other we get,

$$(\sum_{j=1}^{m} \left| f(\mathbf{x}_j) \right|)^2 = \sum_{j=1}^{m} d_j^2 + m^2\bar{u}^2 = m\sigma_U^2 + m^2\bar{u}^2$$

where $\sigma_U^2 = \frac{1}{m}\sum_{j=1}^{m} d_j^2$

Hence, the problem (3) can be reformulated as,

$$\min.\ \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} H(y_i f(\mathbf{x}_i)) + C^* (m\sigma_U^2 + m^2\bar{u}^2)^{\frac{1}{2}} \quad \text{..... (2)}$$

Thus, (1) and (2) are equivalent problems. From (2) we can easily see that for an HDLSS setting we may expect $\sum_{i=1}^{n} H(y_i f(\mathbf{x}_i)) \approx 0$ and hence the U-SVM tries to find a direction vector **w** for which,

- The mean of the projection values of the universum samples is close to the decision boundary $f(\mathbf{x}) = 0$ (i.e the universum samples are symmetric w.r.t the decision boundary).
- The variance of the projection values of the universum samples is small.

For high-dimensional data, most training samples cluster at/near the margins. So, for balanced data sets, the mean of the training samples is likely to be the standard SVM decision boundary. So our conditions say,

- The histogram of projection of the Universum data needs to be symmetric relative to the (standard) SVM decision boundary, i.e. we need the mean of the projected values of Universum samples $\bar{u}$ to be close to the SVM decision boundary i.e $\bar{u} \approx 0$.
- When $\bar{u} \approx 0$ for minimizing (2) we need to minimize the variance of the universum samples. Hence for the case we have a wider distribution of the universum samples about the SVM margins we can minimize (4) by minimizing this wide variance of the universum samples.

REFERENCES

[1] V. Cherkassky, and F. Mulier, Learning from Data Concepts: Theory and Methods, 2nd ed. NY: Wiley, 2007.

[2] V. N. Vapnik, Estimation of Dependencies Based on Empirical Data. Empirical Inference Science: Afterword of 2006. New York: Springer, 2006.

[3] http://www.cs.washington.edu/research/jair/volume2/turney95a-html/appendixA.5.html#top.

[4] D. Caragea, D. Cook, V. G. Honavar, "Gaining insights into support vector machine pattern classifiers using projection-based tour methods," SIGKDD 2001, 251-256.

[5] 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.

[6] F. Poulet, "SVM and graphical algorithms: a cooperative approach," ICDM 2004. Proceedings. Fourth IEEE International Conference on Data Mining, 2004, pp. 499-502.

[7] Dianne Cook and Deborah F. Swayne, Interactive and Dynamic Graphics for Data Analysis: With Examples Using R and GGobi, NY: Springer,2007.

[8] Ahn, J., and Marron, J. S., "The Maximal Data Piling Direction for Discrimination", Biometrika., vol. 97, 254–259, 2010.

[9] 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.

[10] Sam Roweis, "sam roweis:data,"[Online].Available: http://www.cs.nyu.edu/~roweis/data.html[Accessed: May 5, 2010] .

[11]J. Weston, R. Collobert, F. Sinz, L. Bottou, and V. Vapnik, "Inference with the Universum," Proc. ICML, 2006, pp. 1009-1016.

[12]D. Zhang, J. Wang, F. Wang, C. Zhang. "Semi-Supervised Classification with Universum." Proceedings of the 8th SIAM Conference on Data Mining (SDM). 2008. pp. 323-333.

[13]S. Chen, C. Zhang, "Selecting Informative Universum Sample for Semi-Supervised Learning."IJCAI, 2009.

[14]F. Sinz, O. Chapelle, A. Agarwal, and B. Schölkopf, "An Analysis of Inference with the Universum," In Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems(NIPS), pp 1–8, 2008.X. Bai and V. Cherkassky, "Gender classification of human faces using Inference through Contradictions", Proc. IJCNN-2008.

[15]X. Bai and V. Cherkassky, "Gender classification of human faces using Inference through Contradictions", Proc. IJCNN-2008.