Practical Conditions for Effectiveness of the Universum Learning

Vladimir Cherkassky, Fellow, IEEE, Sauptik Dhar and Wuyang Dai

***Abstract*— Many applications of machine learning involve analysis of sparse high-dimensional data, in which the number of input features is larger than the number of data samples. Standard inductive learning methods may not be sufficient for such data, and this provides motivation for non-standard learning settings. This paper investigates a new learning methodology called Learning through Contradictions or Universum support vector machine (U-SVM) [1], [2]. U-SVM incorporates a priori knowledge about application data, in the form of additional Universum samples, into the learning process. This paper investigates possible advantages of U-SVM versus standard support vector machine (SVM), and describes the practical conditions necessary for the effectiveness of the U-SVM. These conditions are based on the analysis of the univariate histograms of projections of training samples onto the normal direction vector of (standard) SVM decision boundary. Several empirical comparisons are presented to illustrate the practical utility of the proposed approach.**

***Index Terms*— Learning through contradiction, model selection, support vector machines, Universum SVM.**

I. Introduction

Sparse high-dimensional data is common in modern machine learning applications. 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. Similarly, in brain imaging studies the dimensionality of the input data vector is larger than the sample size. Such sparse high-dimensional problems represent new challenges for classification methods.

Most approaches to learning with high-dimensional data focus on improving existing inductive methods that try to incorporate a priori knowledge about the optimal model [3-5]. 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 [6].

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. In order to illustrate several non-standard methodologies, consider the task of hand-written digit recognition [3]. 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. Under the transductive [1] and semi-supervised learning settings [7], the learning system uses both labeled (training) and unlabeled (test) samples, in order to predict class labels for future inputs. Under the setting called Learning with Structured Data [2], the training data originates from t different persons (groups), and this additional information (about group labels) is incorporated into learning. Here the goal of learning is to estimate a single predictive model, since the group labels are not provided for test inputs. Another possible scenario is to assume that both the training and test data are generated by t persons, and that the group label is known for both training and test data. This setting known as Multi-Task Learning (MTL) requires estimation of t related classifiers [8-10]. Yet another modification of standard inductive learning assumes that along with labeled training data (i.e., handwritten digits) one has additional a priori information in the form of other handwritten letters. These handwritten letters reflect the style of handwriting and can potentially improve generalization. This leads to the setting known as Learning through Contradiction, or learning in the Universum environment [2]. Such non-standard learning settings reflect properties of real-life applications, and can result in improved generalization, relative to standard inductive learning. However, these new methodologies are more complex, and their advantages and limitations are not well understood.

The idea of ‘inference through contradictions’ was introduced by Vapnik [1, 2] in order to incorporate a priori knowledge into the learning process. Recall that standard

Manuscript received February 24, 2010. This work was supported by NSF grant ECCS-0802056, and by the BICB grant from the University of Minnesota, Rochester.

Vladimir Cherkassky is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455 USA.(phone: 612 625-9597; fax: 612 625-4583; e-mail: cherk001@umn.edu).

Sauptik Dhar is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN 55455 USA. (e-mail: dharx007@umn.edu).

Wuyang Dai is with the Department of Electrical and Computer Engineering,Boston University,Boston,MA 02215. (e-mail: wydai@bu.edu).

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. For example, if the goal of learning is to discriminate between handwritten digits 5 and 8, one can introduce additional 'knowledge' in the form of other handwritten digits 0, 1, 2, 3, 4, 6, 7, 9. These examples from the Universum contain certain information about handwritten digits, but they cannot be assigned to any of the two classes (5 or 8). Also note that Universum samples do not have the same distribution as labeled training samples.

Next, we briefly review optimization formulation for the Universum SVM classifier [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. 1). 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:

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

subject to constraints

for labeled data:

$$y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b] \geq 1-\xi_i \qquad \xi_i\;,\geq 0, i=1,...,n$$

for the Universum:

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

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.

Universum SVM performs regularization that depends on an additional set of unlabeled data (i.e., Universum) available to the learning algorithm. In this respect, it is similar to another well-known example of data-dependent regularization, called semi-supervised learning (SSL) or transduction [1,2,7]. However, the Universum learning is conceptually different from SSL, because the Universum data is *not from the same distribution* as the labeled training data. It is possible to incorporate the Universum into semi-supervised learning or transduction. In fact, Universum learning was originally proposed under the transductive learning setting [1]. More recently, SSL with Universum has been discussed in [12,13], which uses squared loss, rather than hinge loss, in its optimization formulations. In this paper, however, we only address Vapnik's inductive SVM-style inference through contradictions [2].

A successful practical application of U-SVM depends on *two design factors*: implementation of model selection and selection (or generation) of Universum data. Note that model selection becomes rather difficult due to the fact that the kernelized U-SVM has 4 tunable parameters: C, C^*, kernel parameter and ε. In addition, we need to specify the number of Universum samples. Standard SVM, in contrast, has only two tuning parameters. So in practice, standard SVM may yield better results than U-SVM, simply because it has an inherently simpler model selection. Alternatively, a poor generalization performance of U-SVM may be caused by a bad choice of the Universum data. In practice, it may be difficult to separate these two factors, and all existing empirical studies ([11-14]) are performed by expert researchers. In the absence of effective strategies for parameter tuning and practical criteria for the good choice of a Universum, general users cannot easily apply the U-SVM to their problems. So the main objective of this paper 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. The idea that 'a good universum needs to be positioned 'in-between' the two classes' is implicit in Vapnik's original formulation and in the loss function which penalizes Universum samples that are close to either class. However, after introduction of U-SVM several researchers tried to quantify this notion explicitly, in terms of (analytic) properties of the Universum and/or labeled training data. First, 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. Later, 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.

Our 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. As illustrated in Fig. 2, such Universum samples are generated by (randomly) selecting positive and negative training samples, and computing their average. For the problem of handwritten digit recognition, where the goal is to discriminate between handwritten digits 5 and 8, Fig. 3 shows two randomly chosen labeled examples and the corresponding Universum example obtained via averaging.

The paper is organized as follows. The proposed methodology is motivated by analysis of random averaging (RA) Universum. RA Universum samples are generated directly from labeled training data, so we can expect to express conditions for the effectiveness of RA Universum in terms of the statistical properties of labeled data. These properties can be displayed using a novel representation of training data via univariate histograms of projections, introduced in Section II. Section III specifies practical conditions for the effectiveness of RA Universum. Section IV provides empirical examples of several real-life and synthetic data sets, illustrating the effectiveness of RA Universum. Section V extends the conditions to other types of Universa, and demonstrates their effectiveness via empirical comparisons. Section VI provides analytic interpretation of these conditions, by relating them to analytic conditions in Sinz et al [14]. Finally, conclusions are presented in Section VII.

II. Representation of High-Dimensional Data via Univariate Projections

Let us consider binary classification problems with sparse high-dimensional data, where the input dimensionality is much larger than training sample size ($d >> n$). Since n

points generate an n-dimensional subspace (in the input space), the projections of the data points onto any direction vector in the $d-n$ dimensional subspace are all zeros. Also, the projections of the data points onto any vectors orthogonal to the hyperplane generated by the data are non-zero constants. Ahn and Marron [16] analyzed asymptotic ($d>>n$) properties of high-dimensional data for the binary classification setting, under the assumption that input variables are 'nearly independent'. Their analysis suggests that there exists a direction vector such that the projections of data samples from each class onto this direction vector collapse onto a single point. This projection vector is called the Maximal Data Piling (MDP) direction vector [16].

Various linear classifiers differ in approach for selecting the value of the vector $\mathbf{w}$, specifying the normal direction of a hyperplane $(\mathbf{w}\bullet\mathbf{x})+b=0$. For linear SVM classifiers under sparse high-dimensional settings, most data samples from one class lie on the margin border, and their projections onto the normal direction vector $\mathbf{w}$ of the SVM hyperplane tend to be the same (i.e., they project onto the same point). For high-dimensional settings, most linear classifiers (SVM, regularized linear discriminant analysis, etc.) yield the same direction vector $\mathbf{w}$ that coincides (asymptotically) with MDP direction vector (as shown in [17, 18]).

Asymptotic analysis also suggests a poor generalization for such high-dimensional settings, because all of the training data samples become support vectors (i.e., lie on the margin borders). In real-life applications, analytic assumptions in [16] may not hold, because:

- input features are often correlated,
- many application studies use nonlinear SVM.

So the data piling effect can be observed only approximately, in the sense that *most data samples lie near the margin borders*. Next, we illustrate this data piling effect using the WinMac text classification data set (UCI KDD 20 Newsgroups entry). This is a binary classification data set where each sample has 7511 binary features. The data is very sparse, and on average only a small portion (~7.3%) of features are non-zeros. We use 200 samples for training, and 200 independent validation samples for tuning regularization parameter C of a linear SVM. Fig. 4a shows the histogram of univariate projections of the training data onto the normal direction vector $\mathbf{w}$ of the SVM hyperplane. As expected, the training data is well separated and training samples from each class cluster near the margin borders, marked as +1 and -1. Also shown in Fig. 4b is the histogram of projections of the Universum generated from labeled training data via Random Averaging. As training samples cluster at the margin borders, Universum samples will cluster near the linear SVM decision boundary (marked 0 on the horizontal axis). In Fig. 4, the y axis of a histogram indicates the number of samples and the histogram of the projections are evaluated, separately for each class, by first calculating the range of projected values (i.e., *max_value – min_value*), and then dividing this range into 10 different bins. This same procedure is used for all other histograms of projections in this paper. Representation of high-dimensional SVM classifiers using univariate histograms of projections is quite useful for understanding properties of such classifiers [19]. In this paper, univariate histograms of projections are used for understanding conditions for the effectiveness of Universum learning.

For the WinMac data set, U-SVM is not likely to provide an improvement over linear SVM, because optimization formulation (1) enforces the Universum samples to lie near decision boundary. However, as shown in Fig. 4b, the Universum samples already lie near the optimal SVM hyperplane, so no additional improvement due to RA Universum can be expected for this data set.

Empirical comparisons between standard linear SVM and U-SVM for the WinMac data set confirm our intuitive interpretation of Fig. 4. These comparisons use,

- 200 training samples (100 samples per each class);
- 200 independent samples for validation, where validation data set is used for tuning parameters of SVM and U-SVM;
- 1,000 Universum samples generated from training data via random averaging;
- 1,000 independent test samples (used to estimate test error for each method).

All samples are randomly selected from the WinMac data set, and the experiments are repeated 10 times. During model selection, possible values for tuning parameters are as follows:

- parameter *C* ~ [0.01, 0.1, 1, 10, 100, 1000],
- *C**/C ~[0.01, 0.03, 0.1, 0.3, 1, 3, 10],
- ε ~ [0, 0.02, 0.05, 0.1, 0.2].

In all experiments presented in this paper, the regularization parameter C and the kernel parameter for the U-SVM are selected via training a standard SVM classifier (using only the labeled training data). So model selection for U-SVM involves tuning only two parameters, *C**/C and ε .

TABLE I
COMPARISON OF LINEAR SVM AND U-SVM ON WINMAC DATA SET

Training /validation set size	200
Average training error rate (SVM)	0
Average training error rate (U-SVM)	0
Average test error rate (SVM)	7.11% (0.92%)
Average test error rate (U-SVM)	7.14% (0.92%)
Ave. Number of Support Vectors (SVM)	195.60
Typical *C* values	1 or 0.01
Typical *C** values	0.01 or 0.001
Typical ε values	0

Performance results in Table I show average training and test errors for each method, where averages are calculated over 10 runs. The standard deviations of error rates are included in parenthesis. As expected, the U-SVM shows no improvement over standard linear SVM. Additional information in Table I show 'typical' values of tuning parameters selected by the model selection procedure. Note small values of parameter C^* suggesting that Universum samples have little effect on the final model. Even though Table I includes the typical value of ε, the effectiveness of Universum samples is mainly determined by the values of C and C* (or their ratio). Later in the paper we show only typical values of parameters C and C*.

III. Effectiveness of Random Averaging Universum

Comparisons for the WinMac data set suggest that it may be possible to judge the effectiveness of RA Universum by analyzing the histograms of projections of training samples onto the normal direction vector $\mathbf{w}$ of standard SVM model. In fact, for sparse high-dimensional training data, we can have 3 distinct types of projections:

- *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. 4).
- *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. 5a.
- *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. 5b.

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. 4b. (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. 4-5 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.

In summary, the RA Universum is expected to be effective if the training data is well-separable (in some optimally chosen kernel space). The same condition is also implemented by a standard SVM classifier, which seeks a decision boundary with high separability between the two classes. So the univariate histograms of projections for the standard SVM classifier, optimally tuned for a given data set, can be used to 'predict' the usefulness of the RA Universum.

The univariate histograms of projections of training data for nonlinear kernels are calculated using the dual representation of the SVM decision function $f(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$. That is, the projection of a training sample $\mathbf{x}_k$ onto the normal direction of nonlinear SVM decision boundary equals $f(\mathbf{x}_k) = \sum_{i=1}^{n} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_k) + b$. The predicted class label for sample $\mathbf{x}_k$ is the sign of $f(\mathbf{x}_k)$. All SVM software packages supply both the label values and real values of the decision function.

In practice, the condition of good separability holds only approximately, and some labeled samples may fall *inside* the margin borders (denoted as -1 or +1 in the projection space). This can be stated as the requirement that the fraction of training samples that project inside the margin borders is small. So the condition for the effectiveness of RA Universum can be quantified via the following index:

Separability Index *~ the fraction of (labeled) training data samples falling in the interval (-0.99, +0.99) in the univariate projection space.*

The smaller values of this index, say less than 5-6%, indicate high separability of the data, and will generally ensure improved generalization due to RA Universum. This condition for the effectiveness of U-SVM depends only on the properties of labeled training data, because this (RA) Universum is generated from labeled data. Other more general conditions will be presented later in Section V.

Next we illustrate the proposed index for the synthetic Noisy Hyperbolas data set, where the underlying distributions for two classes are given by functions: $x_1 = ((t-0.4)*3)^2+0.225$ and $x_2 = 1-((t-0.6)*3)^2-0.225$.Here, $t \in [0.2, 0.6]$ for class 1 and $t \in [0.4, 0.8]$ for class 2.Gaussian noise is added to both x_1 and x_2 coordinates. The degree of data separation is controlled by noise level. Two values of standard deviation of noise, 0.025 and 0.05, are used to represent low and high noise levels. Examples of 100 training samples for both noise levels are shown in Fig. 6.

For this data set, we apply a standard nonlinear RBF SVM and U-SVM using the following experimental protocol:

- 100 training samples (50 samples per class);
- 100 independent samples for validation, where validation data set is used for tuning parameters of SVM and U-SVM;
- 1,000 Universum samples generated from training data via random averaging;
- 2,000 independent test samples (used to estimate test error for each method).

The RBF kernel has the form $K(\mathbf{x}, \mathbf{x}') = \exp \; -\gamma \|\mathbf{x} - \mathbf{x}'\|^2$, where possible values of parameter γ are taken as [2^{-8}, 2^{-6}…, 2^{4}] during model selection. Each experiment is repeated 10 times using different random realizations of training, validation and test data, and the performance indices (test errors) are averaged over 10 runs. The empirical results are shown in Table II. Note that the low value of separability index correlates well with improved performance of the U-SVM relative to standard SVM. The histograms of projections for low and high noise levels in Fig.

7 further illustrate the separability of the training data.

TABLE II
RELATIVE PERFORMANCE OF U-SVM ON THE HYPERBOLAS DATA SET FOR 100 TRAINING SAMPLES.

Noise level	σ = 0.025	σ = 0.05
Average test error (SVM)	0.69% (0.39%)	5.00% (0.71%)
Average test error (U-SVM)	0.39% (0.17%)	5.10% (1.29%)
Average index value	6.60% (5.40%)	10.60% (4.50%)
Typical C value	100 or 1000	100 or 10
Typical C* value	1 or 10	1 or 100

These empirical results suggest that:

- the Hyperbolas data set, under low noise, is well-separable (corresponding to *Case 3* as discussed in the beginning of this section);
- the Hyperbolas data set, under high noise, is not well-separable.

So for the low noise setting we can expect an improvement due to RA Universum, as confirmed by a lower test error in Table II. For high noise data, the separability index is larger and the U-SVM does not yield any improvement over standard SVM. This example also shows that the shape of the histogram of projections depends on the properties of the data, such as the noise level and sample size. Hence, the effectiveness of RA Universum is very much data-dependent. In particular, introducing Universum can be effective only if the labeled data is well-separable (using standard SVM).

IV. EMPIRICAL RESULTS FOR RA UNIVERSUM

This section presents additional empirical comparisons illustrating the effectiveness of RA Universum using the 'histogram of projections' method. We also illustrate the same approach to other types of Universa. It is important to keep in mind that high-dimensional data is very diverse, so three 'distinct types' of histograms (identified as Case 1, 2 and 3 in Section III) may only serve as approximations of real-life data. Empirical comparisons in this section use three high-dimensional data sets:

- *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.
- *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.
- *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.

For each data set, a classifier is estimated using the training data, its model complexity is optimally tuned using validation data, 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. Linear SVM parameterization is used for the synthetic data set, and both linear SVM and nonlinear RBF SVM are used for the MNIST data set. For the ABCDETC data set, a polynomial kernel (of optimal degree 3) is used in all experiments, following [11]. 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 $\varepsilon= [0, 0.02, 0.05, 0.1, 0.2, 0.3, 0.4]$. Generalization performance of standard SVM and U-SVM is shown in Table III, where the standard deviation of the estimated average test error is indicated in parentheses.

TABLE III
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 and MNIST data indicate that the training data is not well-separable, so the RA Universum should not yield any improvement. For MNIST data with RBF SVM, an average value of separability index is ~ 1.55%, and for MNIST data with linear SVM, the value of separability index is ~15%.

We further investigate the effectiveness of other types of Universum for MNIST data. In this experiment, the training set size is varied as 100, 200 and 1000; and validation set size is always taken to be the same as training set. For Universum data, 125 samples are randomly selected from each of the digits other than 5 or 8. So the total of 1000 Universum samples are used. The 'Other Digits' Universum has been used in many previous studies on U-SVM [11,13,14]. Table IV presents comparison results for this 'Other Digits' Universum. It shows that using 'Other Digits' Universum yields an improvement over standard SVM. Also, these results suggest that 'Other Digits' Universum is better than RA Universum, as evident from comparing the performance (for 1,000 training samples) with results reported earlier in Table III.

TABLE IV
TEST ERROR RATES AND TYPICAL PARAMETER VALUES OF 'OTHER DIGITS' UNIVERSUM SVM

Training set size	100	200	1000
SVM (RBF Kernel)	5.66% (1.89%)	3.69% (0.66%)	1.51% (0.20%)
U-SVM using 'Other Digits' Universum	4.86% (2.08%)	3.03% (0.67%)	1.09% (0.26%)
Typical C values selected for SVM	10 or 1 or 0.01	1	10 or 1
Typical C* values selected for U-SVM	0.1 or 0.01	0.1	3 or 0.3

In addition, two types of Universum, Random Averaging and Other Digits, are compared for low-sample size (100 training samples) in Fig. 10, showing the histograms of projections. As evident from Fig. 10a, the RA Universum is less effective because its projections are narrowly clustered near the SVM decision boundary. On the other hand, projections of the Other Digits Universum are distributed more uniformly between margin borders, suggesting its effectiveness.

V. CONDITIONS FOR EFFECTIVENESS OF THE UNIVERSUM

Even though our discussion in Sections II-IV focused on the RA Universum, the methodology of analyzing univariate histograms of projections can be extended to other types of Universa as well. That is, first we train a standard SVM classifier using labeled data, and analyze its histogram of projections. Then 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.

These two conditions are satisfied, for example, for 'Other Digits' Universum, as shown in Fig. 10b. We emphasize that conditions for the effectiveness of Universa depend on the properties of labeled data. In other words, a Universum can be evaluated only in the context of particular (labeled) training data. Condition (C1) can be related to the analysis performed in [14] suggesting that effectiveness of U-SVM depends on the difference between the means of the labeled training samples and of the Universum samples. Namely, this difference will be small for a symmetric histogram. Condition (C2) directly relates statistical properties of a Universum to the properties of labeled training data. This condition has not been specified in the prior research.

Note that conditions (C1)-(C2) are more general than earlier conditions for the RA Universum. This is because the RA Universum is completely specified by the labeled training data, so its conditions can be formulated in terms of the properties of this data (i.e., the separability index introduced in Section III).

Real-life high-dimensional data is usually well-separable by a standard SVM classifier, and this data may yield three 'typical' histograms of projections identified as Case 1, 2 and 3 in Section III.

This section presents several examples of 'good' and 'bad' Universum selections that illustrate conditions (C1) and (C2). The following is the general strategy used for analyzing the effectiveness of a Universum:

ALGORITHM 1: STRATEGY FOR ANALYZING THE EFFECTIVENESS OF A UNIVERSUM.

a) estimate standard SVM classifier for labeled training data;

b) generate low-dimensional representation of training data by projecting it onto the normal direction vector of the SVM decision boundary estimated in (a). The resulting histogram of projections can be used to analyze separability of the training data;

c) project the Universum data onto the normal direction vector of SVM hyperplane, and analyze projected Universum data in relation to projected training data. Specifically, the Universum is expected to yield an improvement (over standard SVM) only if both conditions (C1)-(C2) are satisfied.

The first set of experiments involves classification of handwritten digits '5' and '8' using the MNIST data. The goal is to investigate the effectiveness of three types of Universa: handwritten digits 1, 3 and 6, and to explain their effectiveness by analyzing histograms of projections of both labeled and Universum data sets. For this experiment:

- Training/validation set samples size is 100 (50 per class);
- Universum set sample size is 1,000;
- Test set sample size is 1,866.

Model selection for standard RBF SVM classifier and for U-SVM is performed using the validation data set. Each experiment is repeated 10 times with different random realizations of training/validation/Universum samples, and the average test error (and its standard deviation) is reported. The test error rates of SVM and U-SVM are shown in Table V, and the typical histograms of projections for training data and Universum data are shown in Fig. 11.

TABLE V
TEST ERROR RATES FOR MNIST DATA WITH DIFFERENT UNIVERSA. TRAINING SET SIZE IS 100 SAMPLES.

	SVM	U-SVM (digit 1)	U-SVM (digit 3)	U-SVM (digit 6)
Test error	4.78% (0.79%)	4.69% (0.70%)	4.54% (0.58%)	4.41% (0.69%)

Typical histograms of projections shown in Fig. 11 suggest that digit 1 Universum is less effective than digit 3 or 6 because it has more biased distribution between projections of labeled data, i.e., digits 5 and 8. The Universum samples for digits 3 and 6 are more widely and symmetrically distributed inside the margin borders, so they are expected to provide better performance (than digit 1 Universum). These findings are consistent with the empirical results in Table V, showing no statistically meaningful improvement for digit 1 Universum, and a small improvement for digits 3 and 6.

The next set of results also involves classification of handwritten digits ‘5’ and ‘8’ using MNIST data. The setting is identical to the first experiment, except that now 1,000 training/validation samples are used (500 per class). The test error rates of SVM and U-SVM are shown in Table VI, and typical histograms of projections for training data and Universum data are shown in Fig. 12.

TABLE VI
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. 12 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. This can be intuitively expected, by noting the similarity between the first principal component of digits 1 and 8. Further, histograms shown in Fig. 12b and Fig. 12c 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 VI. As evident from these experiments, the effectiveness of a Universum is always related to a particular training data set. For instance, digits 3 and 6 Universa are very effective for training sample of size 1,000, but are less effective for training sample of size 100.

The second experiment also involves the classification of handwritten digits ‘5’ and ‘8’ using MNIST data. However, the goal now is to show how the poor performance of ‘bad’ Universum can be predicted using the proposed methodology. We use the same experimental set-up with 1,000 training/validation labeled samples, but also include artificial Universum samples formed as follows:- Each component (pixel) of a 28x28=784 –dimensional sample follows a binomial distribution with probability p(x=1) = 0.1395. This probability value 0.1395 is selected such that the average intensity of Universum samples is the same as that of the training data (averaged for both digit 5 and 8). Fig. 13 shows an example of such a Universum sample.

Experimental results comparing the test error rates for standard RBF SVM classifier and U-SVM using 1,000 Universum samples are shown in Table VII. Typical histograms of projections for training and Universum data are shown in Fig. 14. As expected, this ‘bad’ Universum does not yield any meaningful improvement (over standard SVM), and this can be anticipated by analyzing the histogram of projections in Fig. 14. Note that the histogram for the Universum data in Fig. 14 does not satisfy either condition (i.e., symmetric *and* wide distribution).

TABLE VII
TEST ERROR RATES FOR BINOMIALLY DISTRIBUTED UNIVERSUM.

	SVM	U-SVM (noise)
Test error	1.56% (0.27%)	1.55% (0.25%)

The third set of experiments involves classification of handwritten characters ‘a’ and ‘b’ using the ABCDETC data. The goal is to investigate the effectiveness of three types of Universa: ‘All upper case letters from A to Z’, ‘All digits from 0 to 9’ and ‘Random Averaging’ of training data.

For this experiment:

- Training/validation set samples size is 150 (75 per class);
- Universum set sample size is 1,500;
- Test set sample size is 209, i.e., 105 samples from class ‘a’ and 104 from class ‘b’.

For this data set, we use a 3rd degree Polynomial Kernel. Model selection for the standard Polynomial SVM classifier and for U-SVM is performed using the validation data set. Each experiment is repeated 10 times with different random realizations of training/validation/Universum samples, and the average test error (and its standard deviation) is reported. The test error rates of SVM and U-SVM are shown in Table VIII, and typical histograms of projections for training and Universum data are shown in Fig. 15.

TABLE VIII
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. 15 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 VIII.

In summary, these results suggest that the Universum distribution should be wide enough, relative to the margin borders of standard SVM model estimated from labeled training data. The ‘good’ Universum helps to stabilize SVM decision boundary, and makes it less sensitive to random variability of training samples.

VI. Analytic Interpretation

This section establishes the connection between our practical conditions for the effectiveness of Universum learning and recent analytic results [14]. Sinz et al [14] analyzed the geometric relations of the decision hyperplane learnt with the U-SVM to the Universum data set, and showed that the optimal solutions tend to make the normal vector orthogonal to the principal directions of the Universum. That is, under optimization formulation (1), the U-SVM algorithm 'tries to find a direction $\mathbf{w}^*$ such that the variance of the projections of the Universum samples on that direction is small' [14]. As argued earlier in Section 1, this insight is not very practical, because it does not explicitly describe the properties of Universa *in relation to the labeled training data*. In fact, according to the U-SVM formulation (1) an optimal direction $\mathbf{w}^*$ tries to achieve two goals:

1. Separate labeled samples with large margin (as in standard SVM);
2. Minimize the variance of Universum samples.

Under high-dimensional settings, labeled training data tends to be separable (in some optimally chosen kernel space), so the first goal can be achieved by a standard SVM. This motivates a two-step strategy in Section 5 (shown in Section V), where the standard SVM is estimated first, and then the conditions for the effectiveness of a Universum (i.e., for goal 2) are stated (in C1-C2)**.** This incremental strategy also alleviates the problem of model selection, because parameters of standard SVM are tuned separately.

Further, our conditions (C1)-(C2) for the effectiveness of a Universum implement the above cited analytic property that the optimal direction vector $\mathbf{w}^*$ minimizes the variance of the projections of Universum samples [14]. Namely, our conditions apply to projections of Universum samples onto the vector $\mathbf{w}$ of the standard SVM model. Condition (C1) ensures that the mean of projected Universum samples falls close to SVM decision boundary, or equivalently that the mean of Universum is (approximately) the same as the mean of training samples. This is clearly necessary for minimizing the variance of projections of Universum according to [14]. Condition (C2) ensures that Universum data can indeed provide an improvement relative to standard SVM. That is, if the Universum is narrowly distributed near SVM decision boundary, then the solution vector $\mathbf{w}$ of standard SVM would provide small variance of projections of the Universum, so that no additional improvement (due to this Universum) can be expected.

For the least-squares U-SVM, closed-form analytic interpretation becomes possible. Sinz et al [14] showed an equivalency between the (least-squares) U-SVM learning and the maximization of a hybrid Rayleigh's coefficient due to the kernel oriented Principal Component Analysis (kPCA) and kernel Fisher discriminant analysis (kFDA):

$$\max_{\mathbf{w}} \frac{\overbrace{\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{b}}\mathbf{w}}^{\text{from kFDA}}}{\underbrace{C(\mathbf{w}^{\mathrm{T}}\mathbf{S}_{\mathrm{w}}\mathbf{w})}_{\text{from kFDA}} + \underbrace{C^*\sum_{j=1}^{m}\mathbf{w}^{\mathrm{T}}(\mathbf{x}_j^*-\boldsymbol{\mu})(\mathbf{x}_j^*-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{w}}_{\text{from kPCA}}} \qquad (2)$$

where,

$\mathbf{w} \equiv$ The normal weight vector of decision hyper plane.

$\boldsymbol{\mu}$ ≡The empirical mean of the training samples $\boldsymbol{\mu} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i$.

$\boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \equiv$ The empirical class means given by, $\boldsymbol{\mu}_c = \frac{1}{n_c}\sum_{i\in c}\mathbf{x}_i$

c=Class -1,+1.

$\mathbf{S}_{\mathrm{b}} \equiv$ The between class scatter matrix; $\mathbf{S}_{\mathrm{b}} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^{\mathrm{T}}$.

$\mathbf{S}_{\mathrm{w}} \equiv$ The within class scatter matrix given by,

$$\mathbf{S}_{\mathrm{w}} = \sum_{c=-1,+1}\sum_{i\in c}(\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^{\mathrm{T}} .$$

$\mathbf{x}_j^* \equiv$ The universum samples, where $j = 1...m$.

$C, C^* \geq 0$, control for the tradeoff between minimization of errors and maximization of the number of contradictions.

Our conditions (C1)-(C2) can be only approximately related to the analytic criterion (2), because we use original U-SVM formulation (with hinge loss). Under our approach, the effectiveness of the Universum is evaluated relative to standard SVM model estimated from labeled data (shown in Algorithm 1). This approach can be interpreted using the analytic formulation (2) as follows:

- Minimize the term marked 'from kPCA', since the other two terms in (2) correspond to the solution provided by standard SVM, and they are fixed.

Then the universum samples contribute to the maximization of the hybrid Rayleigh's coefficient through the minimization of the term $\sum_{j=1}^{m}\mathbf{w}^{\mathrm{T}}(\mathbf{x}_j^*-\boldsymbol{\mu})(\mathbf{x}_j^*-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{w}$. Further, this term can be rewritten as the sum of two terms:

$$\sum_{j=1}^{m}\mathbf{w}^{\mathrm{T}}(\mathbf{x}_j^*-\boldsymbol{\mu})(\mathbf{x}_j^*-\boldsymbol{\mu})^{\mathrm{T}}\mathbf{w}$$

$$= \mathbf{w}^{\mathrm{T}}[m(\boldsymbol{\mu}_{\mathrm{U}}-\boldsymbol{\mu})(\boldsymbol{\mu}_{\mathrm{U}}-\boldsymbol{\mu})^{\mathrm{T}}]\mathbf{w} + \sum_{j=1}^{m}[\mathbf{w}^{\mathrm{T}}(\mathbf{x}_j^*-\boldsymbol{\mu}_{\mathrm{U}})(\mathbf{x}_j^*-\boldsymbol{\mu}_{\mathrm{U}})^{\mathrm{T}}\mathbf{w}] \qquad (3)$$

where $\boldsymbol{\mu}_{\mathrm{U}} = \frac{1}{m}\sum_{j=1}^{m}\mathbf{x}_j^*$ is the mean of universum samples.

The first term in (3) is the squared distance (d^2) between the means of the Universum samples and training samples projected onto the normal weight vector (**w**) of the standard SVM model. 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. Hence, our condition (C1) is equivalent to the first term in (3), i.e. minimization of the projected distance (d) between the mean of the universum samples $\boldsymbol{\mu}_U$ and the mean of the training samples $\boldsymbol{\mu}$. Because in the first term of (3), the distance between the means is very small, due to our condition (C1), maximization of the Rayleigh's coefficient (2) depends mainly on minimization of the second term, i.e. the variance of the universum samples projected onto the normal weight vector. Thus, for a case where we have a wide distribution (larger variance) of the universum samples projected onto the normal weight vector, as stated in our condition (C2); we may expect to maximize the Rayleigh's coefficient in (2) by minimizing this large variance. On the other hand, if this variance is small, we expect no or little improvement from the Universum.

VII. Summary

This paper investigates the effectiveness of the U-SVM for finite-sample data. 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.

Empirical results using several real-life and synthetic data sets illustrate the usefulness of the proposed approach, for several types of Universa, and several real-life and synthetic data sets. Proposed practical conditions are also shown to be closely related to analytic conditions independently derived in [14]. However, our 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);

Further, the proposed approach significantly simplifies model selection for U-SVM. That is, the regularization parameter C and the kernel parameter for the U-SVM formulation (1) are selected via training a standard SVM classifier (using only the labeled training data). Then model selection for U-SVM involves tuning only two remaining parameters, C*/C and ε .

In conclusion, we point out that most studies of the U-SVM use balanced data sets with equal misclassification costs. That is, the number of positive and negative labeled samples is (approximately) the same, and the relative cost of false positive and false negative errors is assumed to be the same. This paper also assumes such a balanced setting, where false positive and false negative errors are assigned equal cost in the optimization formulation (1). Many practical applications involve unbalanced data and unequal costs. So there is a need for future research on the properties and conditions for the effectiveness of Universum under such unbalanced settings.

Acknowledgment

The authors thank anonymous reviewers whose comments helped to improve this paper.

References

[1] V.N.Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

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

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

[4] T. Hastie, R. Tibshirani and J. Friedman, *The Elements of Statistical Learning. Data Mining, Inference and Prediction*. New York: Springer, 2001.

[5] B. Schölkopf, and A. Smola, *Learning with Kernels*. MIT Press, 2002.

[6] G. Camps-Valls, J. L. Rojo -Alvarez, and M. Martinez-Ramon, *Kernel Methods in Bioengineering, Signal and Image Processing*. London: Idea Group Publishing, 2007.

[7] O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, Cambridge. MA:The MIT Press,2006.

[8] R. Caruana, "Multi-task learning," *Machine Learning.*, vol. 28, pp. 41-75,July 1997.

[9] T. Evgeniou, and M. Pontil, "Regularized multi-task learning," in *Proc. 17th SIGKDD Conf. on Knowledge Discovery and Data Mining*,2004, pp. 109-117.

[10] L. Liang, and V. Cherkassky, "Connection between SVM+ and Multi-Task Learning," *IJCNN*, 2008.

[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, and C. Zhang. "Semi-Supervised Classification with Universum." *Proceedings of the 8th SIAM Conference on Data Mining (SDM)*, 2008, pp. 323-333.

[13] S. Chen, and 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.

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

[16] J. Ahn, and J.S. Marron, "The direction of maximal data piling in high dimensional space," Technical Report, University of North Carolina at Chapel Hill, 2005.

[17] J. Ye, and T. Wang, "Regularized (Quadratic) Discriminant Analysis for high dimensional, low sample size data," Proc. *SIGKDD*, 2006, pp 454—463.

[18] J. Ye, and T. Xiong, "Computational and theoretical analysis of null space and orthogonal linear discriminant analysis," *Journal of Machine Learning Research*, vol. 7, pp. 1183—1204, 2006.

[19] V.Cherkassky, and S.Dhar "Simple Method for Interpretation of High-Dimensional Nonlinear SVM Classification Models," DMIN, July 2010, pp. 267-272.

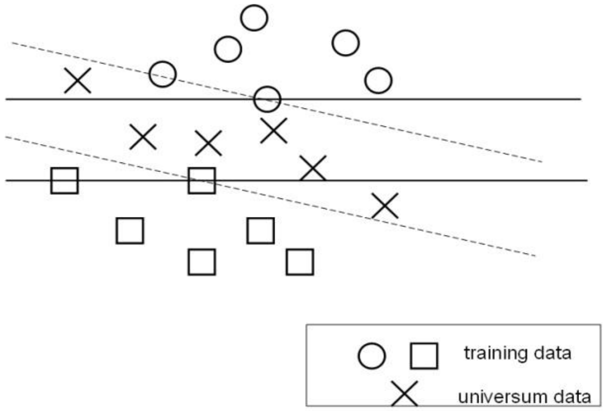


Fig. 1. Two large-margin separating hyperplanes explain training data equally well, but have different number of contradictions on the Universum. The model with a larger number of contradictions should be favored.

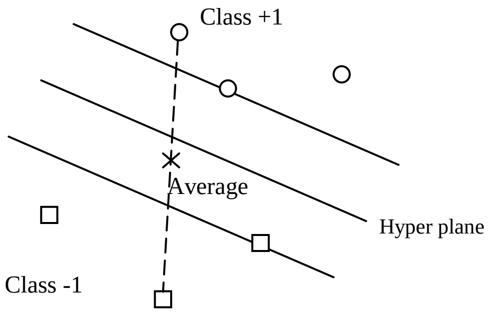


Fig. 2. Generation of the Universum data by averaging.

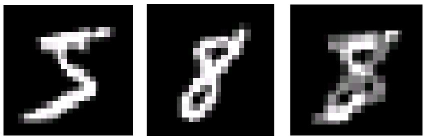
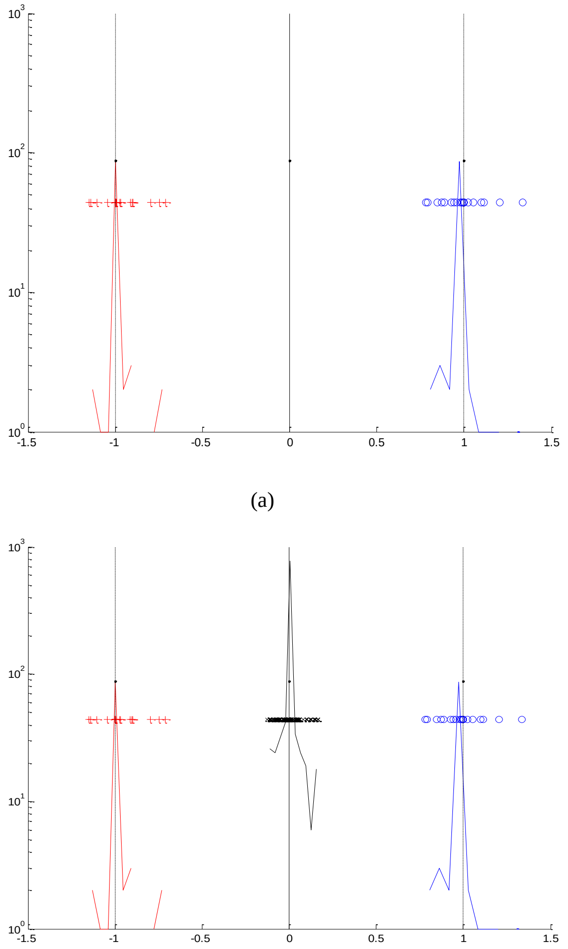


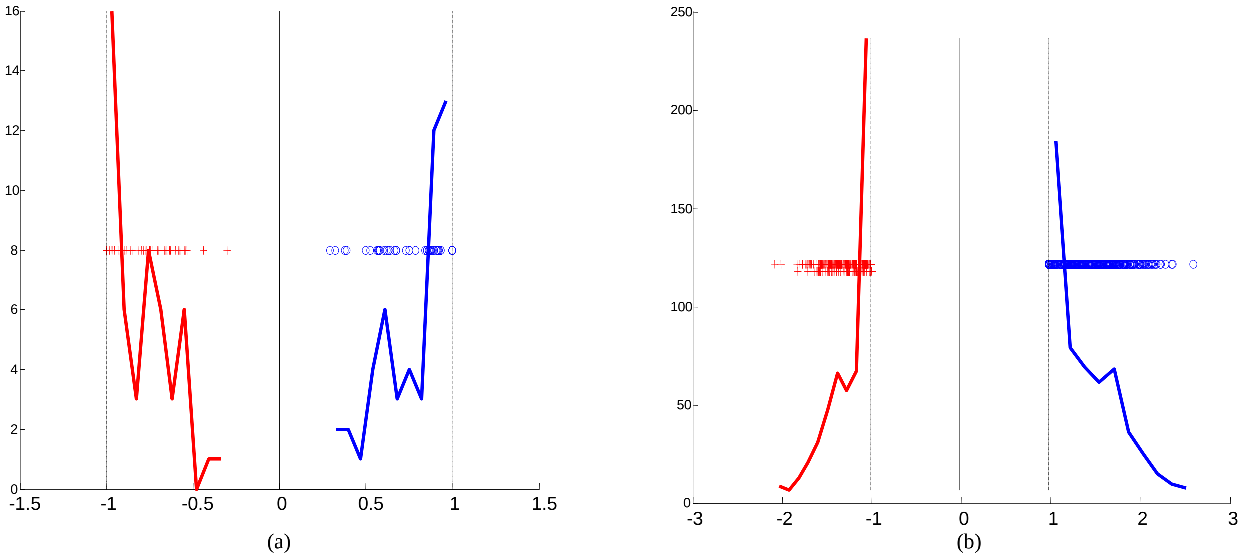
Fig. 3. Example of randomly chosen handwritten digits 5 and 8 and the corresponding Universum sample obtained by averaging.



(a)

(b)

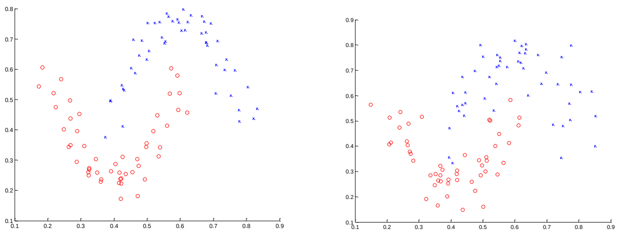
Fig. 4. Histogram of projections of the training data and the universum samples onto the normal direction vector of the SVM hyper plane. (a)Training samples of the two classes in red and blue.(b) Training samples of the two classes in red and blue and the universum samples in black.



(a)

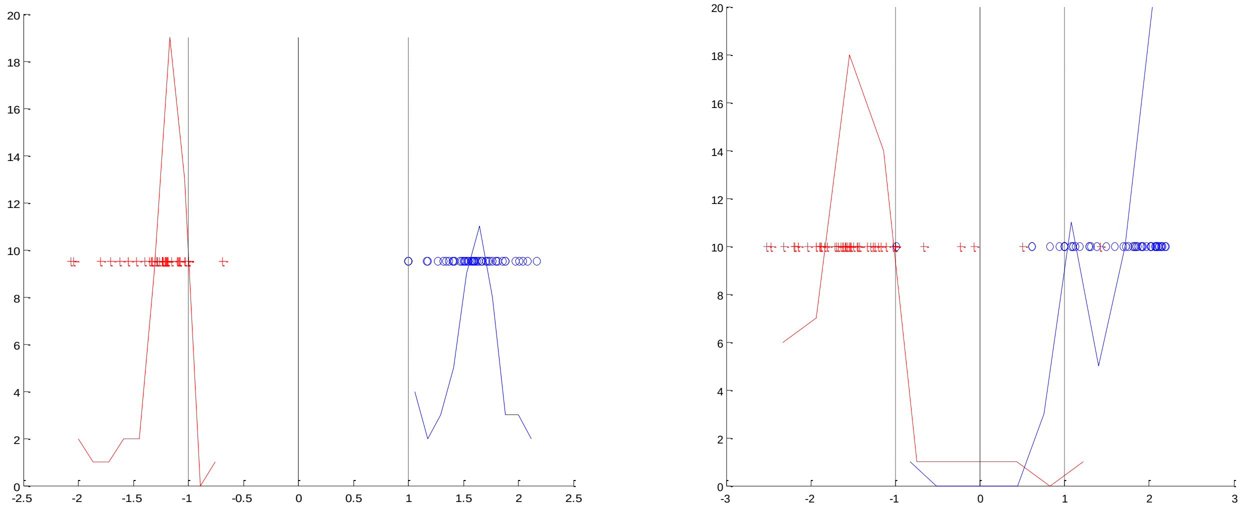
(b)

Fig. 5. Typical histogram. (a) Case 2: training data is separable, and its projections cluster inside margin borders. (b) Case 3: training data is separable, and its projections cluster outside margin borders.



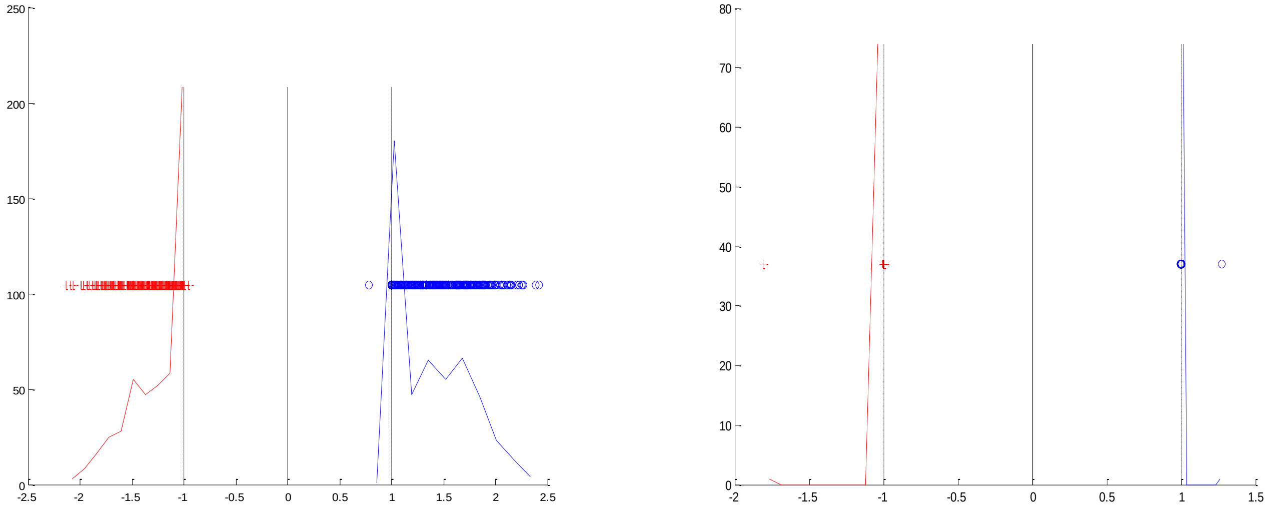
(a) (b)

Fig. 6. Noisy Hyperbolas data sets. (a) Standard deviation of noise is 0.025 (b) Standard deviation of noise is 0.05.



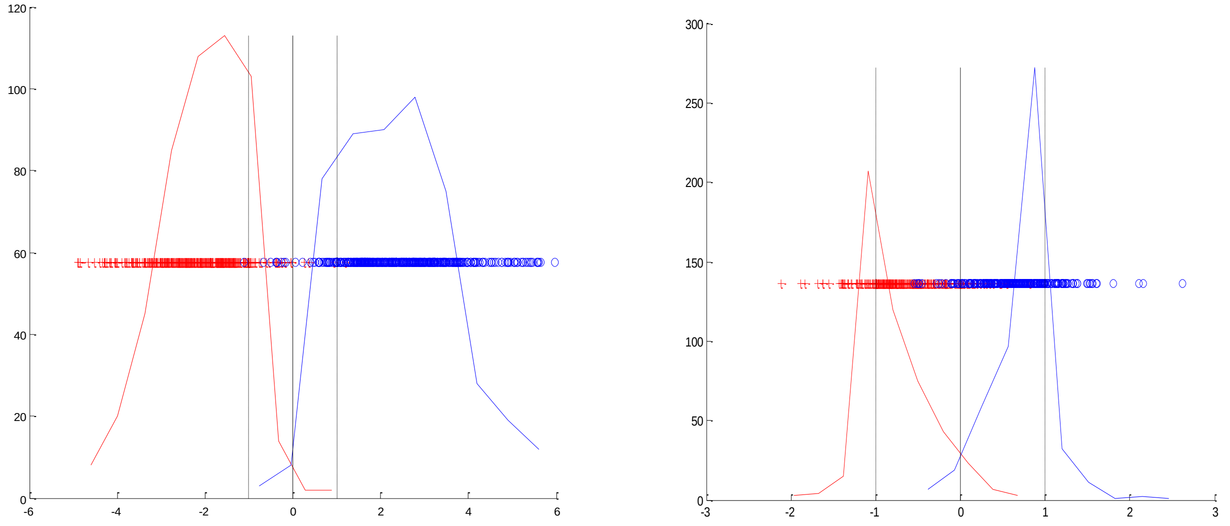
(a) (b)

Fig. 7. Histogram of projections of training data of Hyperbolas data set onto the normal direction of RBF SVM decision boundary. (a) Low Noise Hyperbolas data. (b) High Noise Hyperbolas data.



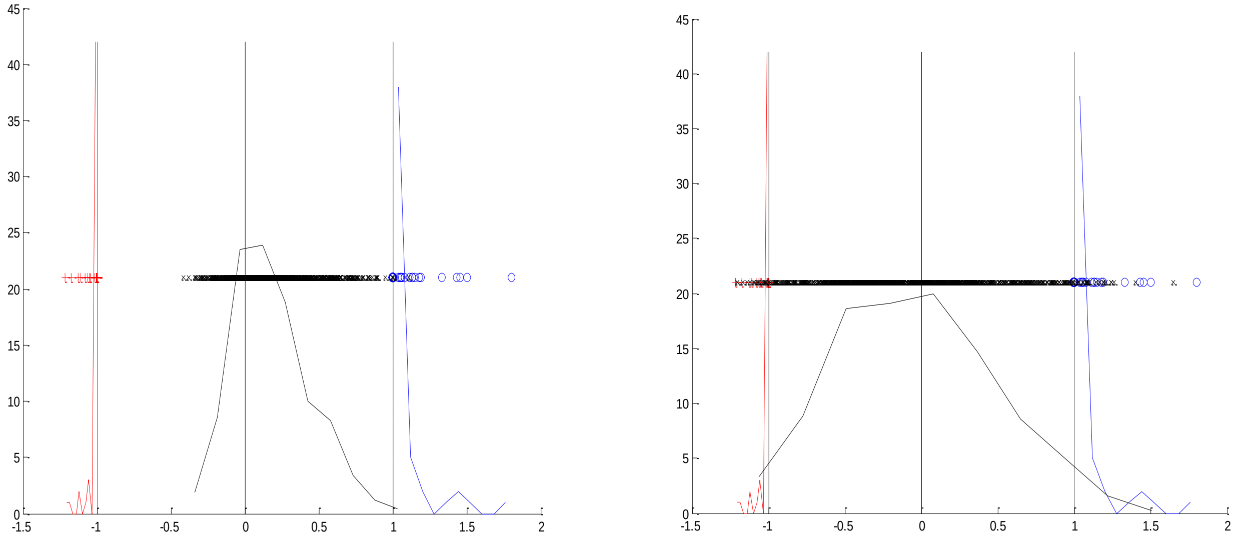
(a) (b)

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.



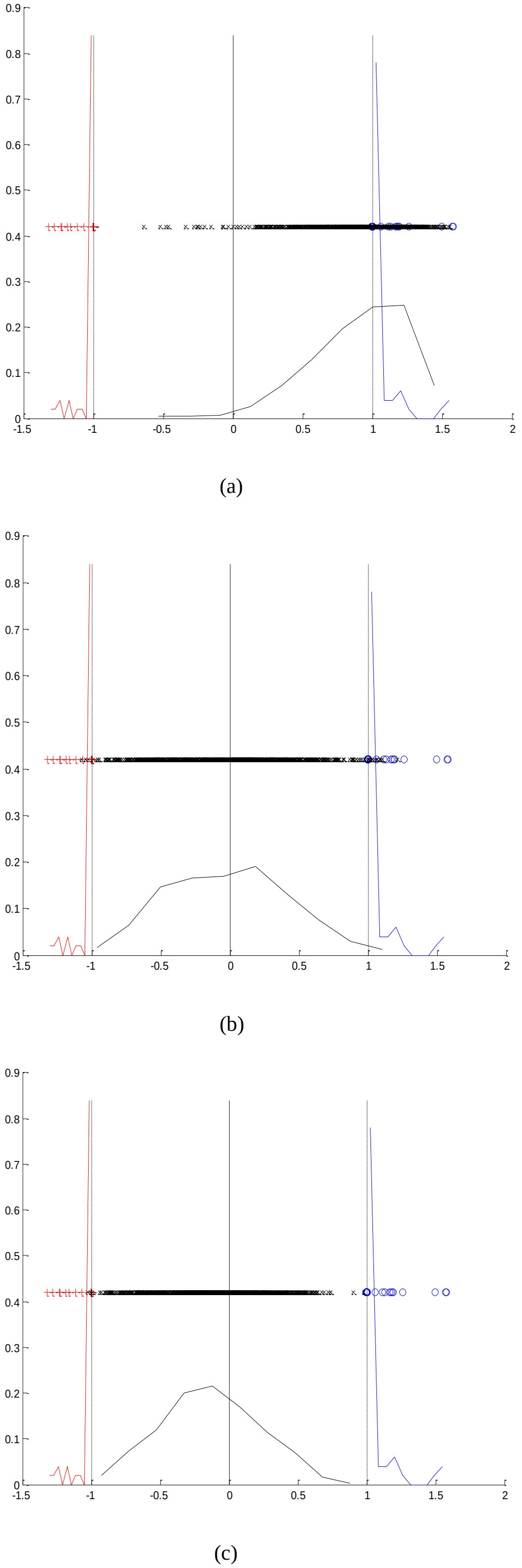
(a) (b)

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



(a) (b)

Fig. 10. The histogram of projections of Universum data onto normal direction of RBF SVM decision boundary. Training set size ~ 100 samples. (a) Random Averaging Universum.(b) Other Digits Universum.

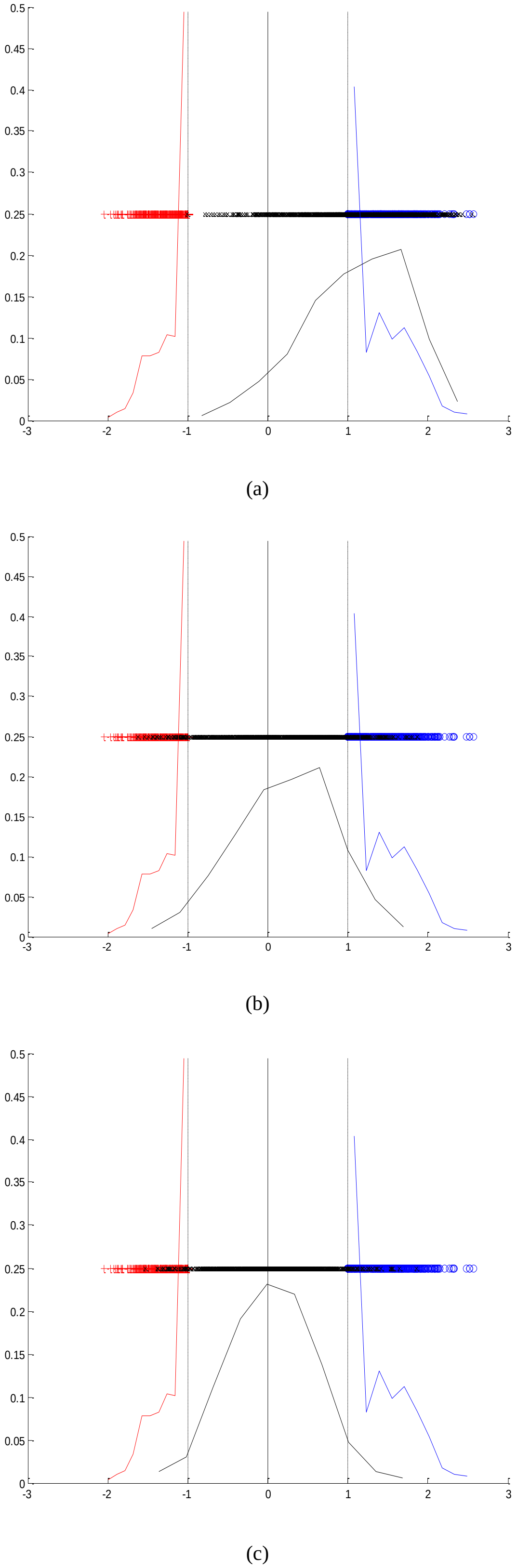


(a)

(b)

(c)

Fig. 11. Univariate histogram of projections for 3 different types of Universa. Training set size ~ 100 samples, Universum set size ~ 1,000 samples. (a) digit 1 Universum(b)digit 3 Universum. (c) digit 6 Universum.



(a)

(b)

(c)

Fig. 12. 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.

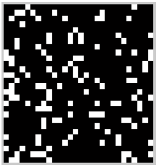


Fig. 13. Universum sample via binomial noise distribution.

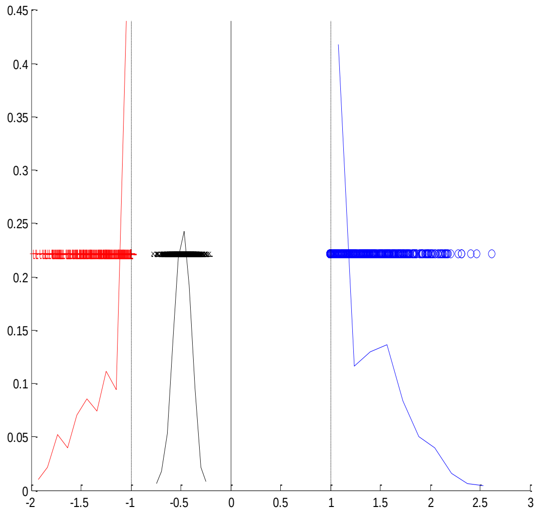
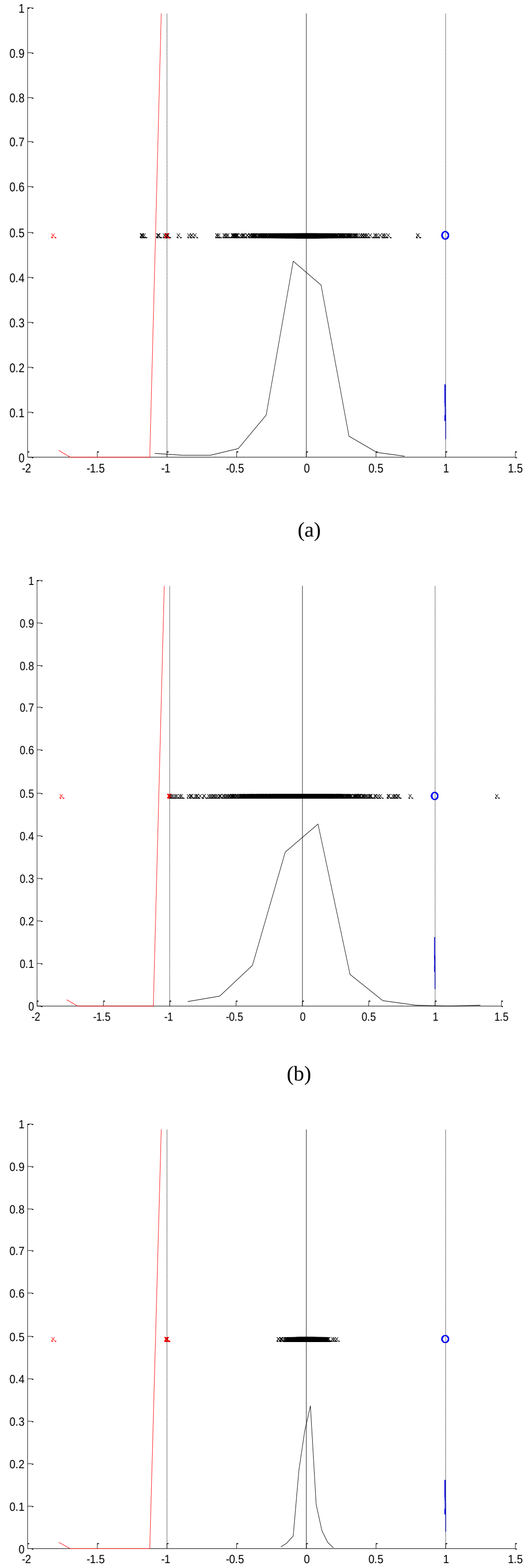


Fig. 14. Histogram of projections for binomially distributed Universum.



(a)

(b)

(c)

Fig. 15. 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.