# Empirical Study of the Universum SVM Learning for High-Dimensional Data

Vladimir Cherkassky and Wuyang Dai

Department of Electrical and Computer Engineering, University of Minnesota
Minneapolis MN 55455 USA
cherk001@umn.edu,daixx048@umn.edu

**Abstract.** Many applications of machine learning involve sparse high-dimensional data, where the number of input features is (much) larger than the number of data samples, $d \gg n$. Predictive modeling of such data is very ill-posed and prone to overfitting. Several recent studies for modeling high-dimensional data employ new learning methodology called Learning through Contradictions or Universum Learning due to Vapnik (1998,2006). This method incorporates a priori knowledge about application data, in the form of additional Universum samples, into the learning process. This paper investigates generalization properties of the Universum-SVM and how they are related to characteristics of the data. We describe practical conditions for evaluating the effectiveness of Random Averaging Universum.

## 1 Introduction and Background

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 using magnetic resonance imaging (MRI) and in image recognition studies, the dimensionality $d$ of the data vector is much larger than the sample size $n$. Such sparse high-dimensional training data sets represent new challenges for classification methods.

Most approaches to learning with high-dimensional data focus on improvements to existing *inductive methods* (Cherkassky and Mulier 2007, Schölkopf and Smola 2002) that try to incorporate a priori knowledge about the good models. Another approach to handling ill-posed high-dimensional classification problems adopts new *non-standard learning formulations* that incorporate a priori knowledge about application data and/or the goal of learning directly into the problem formulation (Cherkassky and Mulier, 2007). 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 relative advantages and limitations are still poorly understood.

The idea of 'inference through contradiction' was introduced by Vapnik (1998) in order to incorporate a priori knowledge into the learning process. This knowledge is introduced in the form of additional unlabeled data samples (called virtual examples or the *Universum*), that are used along with labeled training samples, to perform 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 digit 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 can not be assigned to any of the two classes (5 or 8).

Next we briefly review optimization formulation for the Universum SVM classifier (Vapnik, 2006). Let us consider inductive setting (for binary classification), where we have labeled training data $(\mathbf{x}_i, y_i)$, $(i = 1, ..., n)$, and a set of unlabeled examples from the Universum $(\mathbf{x}_j^*), (j = 1, ..., m)$. The Universum contains data that belongs to the same application domain as 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 hyperplanes $f(\mathbf{x}, \omega) = (\mathbf{w} \cdot \mathbf{x}) + b$. 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 have non-zero slack variables for either class label. So the Universum learning implements a trade-off between explaining training samples (using large-margin hyperplanes) and maximizing the number of contradictions (on the Universum).

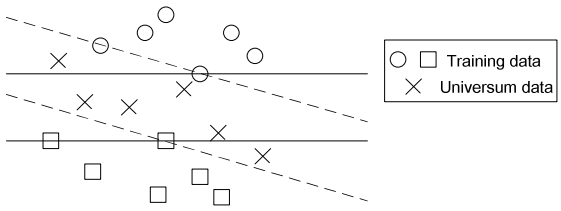



**Fig. 1.** Two large-margin separating hyperplanes explain training data equally well, but have different number of contradictions on the Universum. The model with a larger number of contradictions should be favored.

The quadratic optimization formulation for implementing SVM-style inference through contradictions is shown next following (Vapnik, 2006). For labeled training data, we use standard SVM soft-margin loss with slack variables $\xi_i$. For the Universum samples $(\mathbf{x}_j^*)$, we need to penalize the real-valued outputs

of our classifier that are 'large' (far away than zero). So we adopt $\varepsilon$-insensitive loss (as in standard support vector regression). Let $\xi_j^*$ denote slack variables for Universum samples. Then the Universum SVM formulation can be stated as:

$$\begin{aligned} minimize\ \ & R(\mathbf{w},b) = \tfrac{1}{2}(\mathbf{w}\cdot\mathbf{w}) + C\sum_{i=1}^{n}\xi_i + C^*\sum_{j=1}^{m}\xi_j^* \ \text{where} C, C^* \geq 0 \quad (1)\\ & \text{subject to constraints}\\ & y_i[(\mathbf{w}\cdot\mathbf{x}_i)+b] \geq 1-\xi_i,\ \xi_i \geq 0,\ i=1,\ldots,n \ \text{(for labeled data)}\\ & \text{¦}(\mathbf{w}\cdot\mathbf{x}_j)+b\text{¦} \leq \varepsilon+\xi_j^*,\ \xi_j^* \geq 0,\ j=1,\ldots,m \ \text{(for the Universum)} \end{aligned}$$

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

Solution to the above optimization problem defines the large margin hyperplane $f(\mathbf{x},\omega^*) = (\mathbf{x}\cdot\mathbf{w}^*) + b^*$ that incorporates a priori knowledge (i.e. Universum samples) into the final SVM model. The dual formulation for inductive SVM in the Universum environment, and its nonlinear kernelized version can be readily obtained using standard SVM techniques (Vapnik, 2006). The above quadratic optimization problem is convex due to convexity of constraints for labeled data and for the Universum. Efficient computational algorithms for solving this optimization problem involve modifications of standard SVM software (Weston et al, 2006). Universum SVM software is available at `http://www.kyb.tuebingen.mpg.de/bs/people/fabee/universvm.html`.

Successful application of U-SVM depends on implementation of model selection and selection of Universum data. Note that model selection becomes more difficult as kernelized U-SVM has 4 tunable parameters: $C$, $C^*$, kernel and $\varepsilon$. In addition, we need to specify the number of Universum samples. In contrast, standard SVM has only two tuning parameters. So in practise, standard SVM may yield better performance than U-SVM, simply because it has inherently more robust model selection.

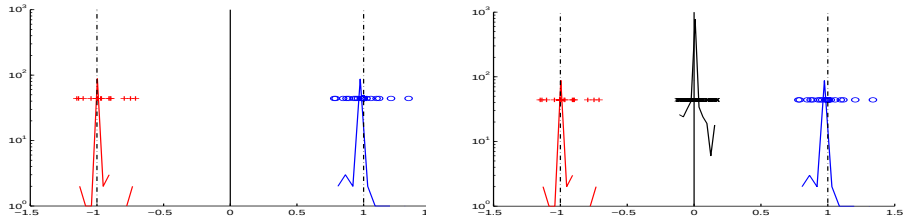
Selection of Universum samples is usually application-dependent (Vapnik 2006, Weston et al 2006). However, there is a possibility of generating Universum data directly from labeled training data. This approach is called *random averaging* (RA) and it does not rely on a priori knowledge about application domain. Such RA Universum samples are generated by randomly selecting a pair of positive and negative training samples, and computing their average. This paper investigates practical conditions for the effectiveness of U-SVM using random averaging (RA). As Universum samples are generated directly from labeled training data, we expect to express these conditions via the properties of training data. These properties can be conveniently presented using novel representation of high-dimensional training data via univariate histograms introduced in section 2. Section 3 specifies practical conditions for the effectiveness of RA Universum. Section 4 provides empirical examples illustrating the effectiveness of U-SVM learning. Conclusions are presented in Section 5.

## 2 Representation of High-Dimensional Data via Univariate Projections

Let us consider binary classification problems with sparse high-dimensional data, where the input dimensionality is larger than training sample size ($d \gg n$). Since $n$ points generate $n$-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 (2005) analyzed asymptotic ($d \gg n$) properties of high-dimensional data for the binary classification setting, under the assumption that input variables are 'nearly independent'. Their analysis suggests that asymptotically there is 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 direction vector.

Various linear classifiers differ in approach for selecting the value of the vector $\mathbf{w}$, specifying the normal direction of a hyperplane $(\mathbf{x} \cdot \mathbf{w}) + b$. 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 SVM hyperplane normal direction vector $\mathbf{w}$ tend to be the same (i.e., they project onto the same point). In real-life applications, analytic assumptions in (Ahn and Marron 2005) do not hold, so the data piling effect can be observed only approximately, in the sense that many data samples lie near the margin borders. Next we illustrate the 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 ($\sim$7.3%) of features are non-zeros. We use 200 samples for training, and 200 independent validation samples for tuning linear SVM model parameter $C$. Fig. 2(a) shows the histogram of univariate projections of the training data onto the normal direction vector $\mathbf{w}$ of the SVM hyperplane. As expected, 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. 2(b) is the histogram of projections of the Universum samples generated from training data via Random Averaging. As training samples cluster at the margin borders, Universum samples will cluster near linear SVM decision boundary (marked 0 on the horizontal axis). In Fig. 2, the $y$ axis of a histogram indicates the number of samples and the histogram of 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 procedure is used for other histograms of projections shown later in this paper.

For this data set, U-SVM is not likely to provide an improvement over linear SVM, because optimization formulation (1) tries to force the Universum samples to lie near decision boundary. However, as shown in Fig. 2(b), Universum samples already lie near the optimal hyperplane (of standard SVM model), so no additional improvement due to U-SVM can be expected for this data set.



(a) histogram of projections (b) histogram of projections of Universum

**Fig. 2.** Histogram of projections of training data onto the normal direction vector **w** of the SVM hyperplane

Next we show empirical comparisons between standard linear SVM and U-SVM for the WinMac data set, in order to confirm our intuitive interpretation of Fig. 2. Our 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 testing error for each method).
All samples are randomly selected from the WinMac data set, and experiments are repeated 10 times. During model selection, possible values for tuning parameters of U-SVM are given below:
$C \sim$[0.01,0.1,1,10,100,1000], $C^*/C \sim$[0.01,0.03,0.1,0.3,1,3,10] and $\varepsilon \sim$[0,0.02, 0.05,0.1,0.2]. These parameter values for U-SVM are also used for modeling other data sets presented in this paper.
Performance results in Table 1 show average training and testing error for each method, where averages are calculated over 10 runs. As expected, Universum SVM shows no improvement over standard linear SVM. Additional information in Table 1 shows 'typical' values of tuning parameters selected by the model selection procedure. Note small values of parameter $C^*$ suggesting that Universum data samples have little effect on the final model. In Table 1, the typical value of $\varepsilon$ is also shown, but the effectiveness of Universum is mainly determined by the values of $C$ and $C^*$ (or their ratio). So only typical values of parameter $C$ and $C^*$ will be shown later in this paper.

## 3 Conditions for Effectiveness of Random Averaging Universum

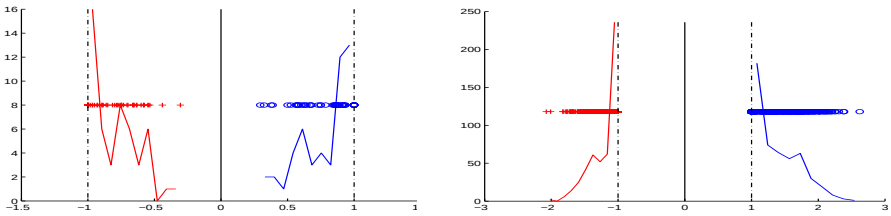
Comparisons for the WinMac data set suggest that it may be possible to judge the effectiveness of RA Universum by analyzing the histograms of projections of training samples onto the normal direction vector **w** of standard SVM model. In fact, for sparse high-dimensional training data sets, we can have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

**Table 1.** Comparison of Linear SVM and U-SVM on WinMac data set. Standard deviation of error rate is shown in parenthesis.

| | |
|---|---|
| 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 $\varepsilon$ values | 0 |

- *Case 1:* univariate projections of training data onto SVM decision boundary cluster strongly on margin borders (as in Fig. 2).
- *Case 2:* univariate projections of training data onto SVM decision boundary cluster inside margin borders, as shown in Fig. 3(a).
- *Case 3:* univariate projections of training data onto SVM decision boundary cluster outside margin borders, as shown in Fig. 3(b).

From the nature of the U-SVM optimization formulation (1), it can be expected that Universum would not provide any improvement for cases (1) and (2), because Universum samples generated by random averaging are distributed narrowly near SVM decision boundary in the projection space, as shown in Fig. 2(b). However, U-SVM is expected to provide an improvement in case (3), where random averaging would produce Universum samples scattered far away from SVM decision boundary in the projection space, and possibly outside the margin borders of standard SVM.



(a) *Case 2* (b) *Case 3*

**Fig. 3.** *Typical histograms*: training data is separable and its projections cluster *inside* or *outside* the margin borders

Note that histograms in Figs. 2-3 assume that training data is separable. This is generally true for sparse high-dimensional data. For lower-dimensional data, we assume that separability can be achieved using some nonlinear kernel. So the conditions for the effectiveness of RA Universum can be stated as follows:

1. Training data is well-separable (in some optimally chosen kernel space).
2. The fraction of training data samples inside SVM margin borders is small.
The same conditions are also sought by standard SVM classifiers during model selection. That is, optimally selected SVM parameters (kernel and $C$ value) aim at achieving high degree of separation between training samples from two classes. So analysis of univariate histograms for standard SVM model, optimally tuned for a given data set, can be used to 'predict' the usefulness of RA Universum.

The univariate histograms of projections of training data for nonlinear kernels are calculated using representation of SVM decision function in the dual space, $f(\mathbf{x}) = \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b$. That is, the value of projection of training sample $\mathbf{x}_k$ onto the normal direction of nonlinear SVM decision boundary is expressed as $f(\mathbf{x}_k) = \sum \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_k) + b$. The predicted class label for sample $\mathbf{x}_k$ is the sign of $f(\mathbf{x}_k)$. Most existing SVM software packages supply both the label values and real values of the decision function.

In practice, data samples will not always fall *precisely* on the margin borders (denoted as -1 or +1 in the projection space), so the condition for the effectiveness of RA Universum can be quantified via simple separation index:

***Separation Index*** $\sim$ *the fraction of training data samples falling in the interval (-0.99,+0.99) in the univariate projection space.*

Smaller values of this index, say less than 5-6%, indicate higher separability of the training data, and will generally ensure improved prediction accuracy due to Universum generated via random averaging. This index can be used in practise as 'rule-of-thumb' rather than precise necessary condition guaranteeing improved performance of U-SVM vs standard SVM.

## 4 Empirical Results

This section presents additional empirical comparisons between standard SVM and U-SVM classifiers, using two 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 significant. An output class label is generated as $y = sign(x_1 + x_2 + \ldots + x_{200} - 100)$. For this data set, only linear SVM is used because optimal decision boundary is known to be linear. 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 from training data.
- *Real-life MNIST handwritten digit data set*, where data samples represent handwritten digit 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. Training set size is 1,000, validation set size is 1,000. For U-SVM, 1,000 Universum samples are generated via random averaging from training data. Separate test set of size 1,866 is used for all experiments.

For each data set, a classifier is applied to training data, its model complexity is optimally tuned using independent validation data set, and then the test error of an optimal model is estimated using test data. The results of such an

experiment would depend on random realization of training and validation data. So each experiment is repeated 10 times, using different random realizations, and average error rates are reported for comparison. Linear SVM parameterization is used for synthetic data set, and both linear SVM and nonlinear RBF SVM are used for MNIST data set. Comparison of generalization performance of standard SVM and U-SVM is shown in Table 2, where standard deviation of estimated average test error is indicated in parenthesis.

**Table 2.** Test error rates for MNIST and synthetic data sets.

| | SVM | U-SVM |
|---|---|---|
| MNIST (RBF Kernel) | 1.37%(0.22%) | 1.20%(0.19%) |
| MNIST (Linear Kernel) | 4.58%(0.34%) | 4.62%(0.37%) |
| Synthetic data (Linear Kernel) | 26.63%(1.54%) | 26.89%(1.55%) |

These results indicate that U-SVM yields improvement over SVM only for digits data when using RBF kernel. These results can be explained by examining the histograms of projections of training data. Fig. 4 shows the histogram of training data projections onto the normal direction of the RBF SVM decision boundary, suggesting that this data is well-separable. On the other hand, histogram of projections for linear SVM shown in Fig. 5 for both data sets, indicate that training data is not well-separable, so the Universum SVM should not provide any improvement. For MNIST data with RBF SVM, (average) value of separability index is 1.55%, whereas for MNIST data with linear SVM, (average) value of separability index is 15%.

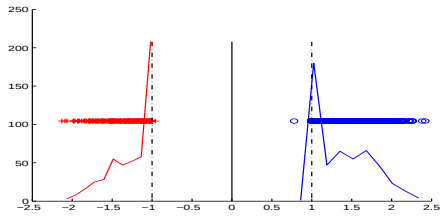
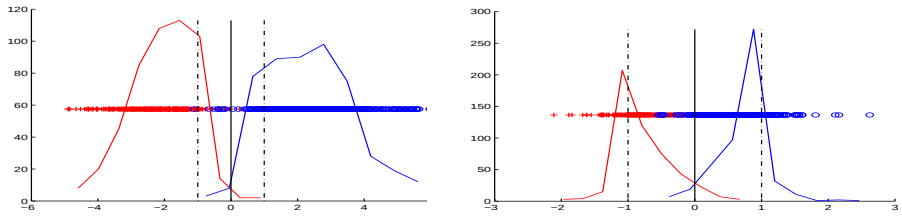



**Fig. 4.** Histogram of projections of MNIST training data onto normal direction of RBF SVM decision boundary. Training set size∼1,000 samples.

Finally, we investigate the effectiveness of other types of Universum for MNIST data. In this experiment, the training set size is varied as 100, 200 and 1,000; and the 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 1,000 Universum samples are used. Table 3 presents comparison between standard SVM and Universum SVM using



(a) MNIST data set (b) Synthetic data set

**Fig. 5.** Histogram of projections onto normal direction of linear SVM hyperplane.

'other digits' as Universum. Table 3 shows that using other digits for Universum always results in improvement over standard SVM. These results (for 1,000 training samples) should be compared with results obtained using random averaging U-SVM reported in Table 2.

**Table 3.** Test error rates and 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 |

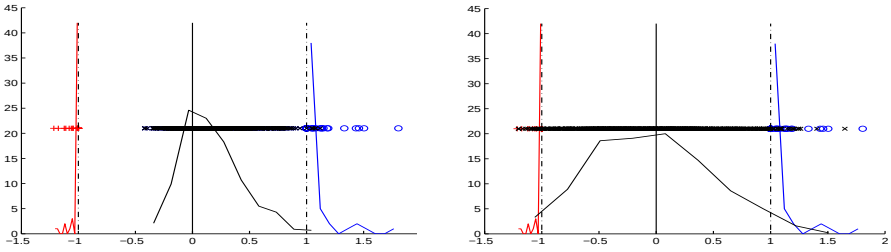
In addition, two types of Universum, Random Averaging and Other Digits, are compared for low sample size (100 training samples) in Fig. 6, showing the histograms of projections. As evident from Fig. 6(a), the RA Universum is less effective because its projections are narrowly clustered near SVM decision boundary. On the other hand, projections of the Other Digits Universum are distributed more uniformly between margin borders, suggesting its effectiveness.

Comparison of histograms of projection in Fig. 4 (for 1,000 training samples) and Fig. 6(a) (for 100 samples) shows that the effectiveness of the RA Universum depends on the training sample size.

## 5 Summary

This paper investigates the effectiveness of the Random-Averaging Universum SVM for high-dimensional data. Our analysis suggests that relative advantages of using U-SVM depend on the properties of training data, such as sample size and noise level. In many situations, using RA Universum SVM does not offer any improvement over standard SVM.

In general, relative performance of learning methods is always affected by the properties of application data at hand. New learning settings, such as Universum-



(a) Random Averaging Universum (b) Other Digits Universum

**Fig. 6.** The histogram of projections of Universum data onto normal direction of RBF SVM decision boundary. Training set size∼100 samples.

SVM, are inherently more complex than standard SVM, and they have more tuning parameters. So it is important to have simple practical criteria that guarantee potential advantages of using U-SVM, for a given data set. To this end, the paper describes novel representation of high-dimensional training data using projections of this data onto the normal direction of SVM decision boundary. Analysis of the univariate histograms of projected training data, presented in this paper, leads to practical conditions for the effectiveness of RA Universum. Empirical results using several real-life and synthetic data sets illustrate the usefulness of the proposed histogram representation and analysis, for random averaging Universum. The same approach can be used for analyzing effectiveness of other types of Universum, such as 'other digits' used for MNIST data set.

**Acknowledgments.** This work was supported by NSF grant ECCS-0802056, by A. Richard Newton Breakthrough Research Award from Microsoft Corporation, and by BICB grant from the University of Minnesota, Rochester.

## References

1. Ahn,J., & Marron, J.S.(2005). The direction of maximal data piling in high dimensional space. Technical Report, University of North Carolina at Chapel Hill.
2. Cherkassky,V., and Mulier, F.(2007). Learning from Data Concepts: Theory and Methods, Second Edition, NY: Wiley.
3. Schölkopf,B. and A.Smola, Learning with Kernels, MIT Press, 2002
4. Vapnik,V.N., Statistical Learning Theory, Wiley, NY 1998
5. Vapnik,V.N., Estimation of Dependencies Based on Empirical Data. Empirical Inference Science: Afterword of 2006, Springer, 2006.
6. Weston,J., Collobert,R., Sinz,F., Bottou,L. and V.Vapnik, Inference with the Universum, Proc. ICML, 2006