Lecture Notes in Computer Science, Vol. 5163, 346-356, 2008.

# Support Vector Machines for visualization and dimensionality reduction

Tomasz Maszczyk and Włodzisław Duch

Department of Informatics, Nicolaus Copernicus University, Toruń, Poland
tmaszczyk@is.umk.pl;Google:W.Duch
http://www.is.umk.pl

**Abstract.** Discriminant functions $g_{\mathbf{W}}(\mathbf{X})$ calculated by Support Vector Machines (SVMs) define in a computationally efficient way projections of high-dimensional data on a direction perpendicular to the discriminating hyperplane. These projections may be used to estimate and display posterior probability densities $p(C|g_{\mathbf{W}}(\mathbf{X}))$. Additional projection directions for visualization and dimensionality reduction are created by repeating the linear discrimination process in a space orthogonal to already defined projections. This process allows for an efficient reduction of dimensionality, visualization of data, at the same time improving classification accuracy of a single discriminant function. SVM-based sequential visualization shows that even if discrimination methods completely fail, nearest neighbor or rule-based methods in the reduced space may provide simple and accurate solutions.

**Key words:** Reduction of dimensionality, Data Visualization, SVM

## 1 Introduction

Many classifiers, such as neural networks or support vector machines, work as black-box predictors. Their quality is estimated in a global way, on the basis of some accuracy or cost measures. In practical applications it is important to be able to evaluate a specific case, showing the confidence in predictions in the region of space close to this case. Looking at the situation from the Bayesian perspective [1] it is clear that globally defined priors may be very different from local priors. Principal Components Analysis (PCA), Independent Component Analysis (ICA), Multidimensional Scaling (MDS) or other such methods commonly used for direct visualization of data [2] may be very useful for exploratory data analysis, but do not provide any information about reliability of the method used for classification of a specific case. Visualization methods already proved to be very helpful in understanding mappings provided by neural networks [3, 4].

This paper shows how to use any linear discriminant analysis (LDA), or SVM classifier in its linear or kernelized version, for dimensionality reduction and data visualization, providing interesting information and improving accuracy at the same time. The method presented here may be used for exploratory data visualization or for analyzing results of the LDA. There is no reason why linear

discriminants or nonlinear mappings provided by feedforward neural networks should be treated as black boxes.

In the next section a few linear and non-linear visualization methods are described, and visualization based on linear discrimination is introduced. For illustration visualization using linear SVM in one and two dimensions for several real and artificial datasets is presented in section 3. This type of visualization is especially interesting because it is fast, projections are easy to understand, and other methods do not seem to achieve significantly better projections. Conclusions are given in section four.

## 2 Visualization algorithms

Visualization methods are discussed in details in many books, for example [2, 5]. Below a short description of three popular methods, multidimensional scaling (MDS), principal component analysis (PCA), and Fisher discriminant anlaysis, is given, followed by description of our approach. In the next section empirical comparisons of these four methods are given. Although we have compared our method with many other non-linear and linear methods space limitation do not allow here to present more detailed comparisons.

Multidimensional scaling (MDS) is the only non-linear technique used here. The main idea, rediscovered several times [6–8], is to decrease dimensionality while preserving original distances in high-dimensional space. This is done either by minimization of specific cost functions [9] or by solving cubic system of equations [8]. MDS methods need only similarities between objects, so explicit vector representation of objects is not necessary. In metric scaling specific quantitative evaluation of similarity using numerical functions (Euclidean, cosine or any other measures) is used, while for non-metric scaling qualitative information about the pairwise similarities is sufficient. MDS methods also differ by their cost functions, optimization algorithms, the number of similarity matrices used, and the use of feature weighting. There are many measures of topographical distortions due to the reduction of dimensionality, most of them variants of the stress function:

$$S_T(\mathbf{d}) = \sum_{i>j}^{n} \left(D_{ij} - d_{ij}\right)^2 \tag{1}$$

or [8]

$$S_D(\mathbf{d}) = \frac{\sum_{i>j}^{n} \left(D_{ij} - d_{ij}\right)^2}{\sum_{i>j} \left(d_{ij}^2 + D_{ij}^2\right)} \tag{2}$$

where $d_{ij}$ are distances (dissimilarities) in the target (low-dimensional) space, and $D_{ij}$ are distances in the input space, pre-processed or calculated directly using some metric functions. These measures are minimized over positions of all target points, with large distances dominating in the $S_T(\mathbf{d})$. $S_D(\mathbf{d})$ is zero for perfect reproduction of distances and 1 for complete loss of information (all $d_{ij} = 0$), weighting the errors $|D_{ij} - d_{ij}|$ by squared summed distances. The sum runs

over all pairs of objects and thus contributes $O(n^2)$ terms. In the $k$-dimensional target space there are $kn$ parameters for minimization. For visualization purposes the dimension of the target space is $k$ =1-3, but the number of objects $n$ may be quite large, making the approximation to the minimization process necessary [10].

MDS cost functions are not easy to minimize, with multiple local minima for quite different mappings. Initial configuration is either selected randomly or based on projection of data to the space spanned by principal components. Dissimilar objects are represented by points that are far apart, and similar objects are represented by points that should be close, showing clusters in the data. Orientation of axes in the MDS mapping is arbitrary and the values of coordinates do not have any simple interpretation, as only relative distances are important.

PCA is a linear projection method that finds orthogonal combinations of input features $\mathbf{X} = \{x_1, x_2, ..., x_N\}$ accounting for most variation in the data. Principal components $\mathbf{P}_i$ result from diagonalization of data covariance matrix [11], and are sequentially ordered according to the size of the eigenvalues. They provide directions of maximum variability of data points, thus guaranteeing minimal loss of information when position of points are recreated from their low-dimensional projections. Taking 1, 2 or 3 principle components and projecting the data to the space defined by these components $y_{ij} = \mathbf{P}_i \cdot \mathbf{X}_j$ provides for each input vector its representative $(y_{1j}, y_{2j}, ...y_{kj})$ in the target space. For many data distributions such projections will not show interesting structures.

Kernel PCA [12] finds directions of maximum variance for training vectors mapped to an extended space. This space is not constructed in an explicit way, the only condition is that the kernel mapping $K(\mathbf{X}, \mathbf{X}')$ of the original vectors should be a scalar product $\Phi(\mathbf{X}) \cdot \Phi(\mathbf{X}')$ in the extended space. This enables interesting visualization of data, although interpretation of resulting graphs may be rather difficult.

Supervised methods that use information about classes determine more interesting directions. Fisher Discriminant Analysis (FDA) is a popular algorithm that finds a linear combination of variables separating various classes as much as possible. FDA maximizes the ratio of between-class to within-class scatter, seeking a direction $\mathbf{W}$ such that

$$\max_{\mathbf{W}} J_{\mathbf{W}} = \frac{\mathbf{W}^T \mathbf{S}_B \mathbf{W}}{\mathbf{W}^T \mathbf{S}_I \mathbf{W}} \tag{3}$$

where $\mathbf{S}_B$ and $\mathbf{S}_I$ are given by

$$\mathbf{S}_B = \sum_{i=1}^{C} \frac{n_i}{n} (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T; \qquad S_I = \sum_{i=1}^{C} \frac{n_i}{n} \hat{\mathbf{\Sigma}}_i \tag{4}$$

Here $\mathbf{m}_i$ and $\hat{\mathbf{\Sigma}}_i$ are the sample means and covariance matrices of each class and $\mathbf{m}$ is the sample mean [5].

FDA is frequently used for classification and projecting data on a line. For visualization generating the second FDA vector in a two-class problem is not so

trivial. This is due to the fact that the rank of the $\mathbf{S}_B$ matrix for the $C$-class problems is $C-1$. Cheng *et al.* [13] proposed several solutions to this problem:

- stabilize the $\mathbf{S}_I$ matrix by adding a small perturbation matrix;
- use pseudoinverse, replacing $S_I^{-1}$ by the pseudoinverse matrix $S_I^{\dagger}$;
- use rank decomposition method.

In our implementation pseudoinverse matrix has been used to generate higher FDA directions.

**Linear SVM** algorithm searches for a hyperplane that provides a large margin of classification, using regularization term and quadratic programming [14]. Non-linear versions are based on a kernel trick [12] that implicitly maps data vectors to a high-dimensional feature space where a best separating hyperplane (the maximum margin hyperplane) is constructed. Linear discriminant function is defined by:

$$g_{\mathbf{W}}(\mathbf{X}) = \mathbf{W}^T \cdot \mathbf{X} + w_0 \tag{5}$$

The best discriminating hyperplane should maximize the distance between decision hyperplane defined by $g_{\mathbf{W}}(\mathbf{X}) = 0$ and the vectors that are nearest to it, $\max_{\mathbf{W}} D(\mathbf{W}, \mathbf{X}^{(i)})$. The largest classification margin is obtained from minimization of the norm $\|\mathbf{W}\|^2$ with constraints:

$$Y^{(i)} g_{\mathbf{W}}(\mathbf{X}^{(i)}) \geq 1 \tag{6}$$

for all training vectors $\mathbf{X}^{(i)}$ that belong to class $Y^{(i)}$. Vector $\mathbf{W}$, orthogonal to the discriminant hyperplane, defines direction on which data vectors are projected, and thus may be used for one-dimensional projections. The same may be done using non-linear SVM based on kernel discriminant:

$$g_{\mathbf{W}}(\mathbf{X}) = \sum_{i=1}^{N_{sv}} \alpha_i K(\mathbf{X}^{(i)}, \mathbf{X}) + w_0 \tag{7}$$

where the summation is over support vectors $\mathbf{X}^{(i)}$ that are selected from the training set. The $x = g_{\mathbf{W}}(\mathbf{X})$ values for different classes may be smoothed and displayed as a histogram, estimating $p(x|C)$ class-conditionals and calculating posterior probabilities $p(C|x) = p(x|C)p(C)/p(x)$. Displaying $p(C|x)$ shows distance of vectors from decision borders, overlaps between classes, on this basis allowing for immediate estimation of reliability of classification.

SVM visualization in more than one dimension requires generation of more discriminating directions. The first direction should give $g_{\mathbf{W}_1}(\mathbf{X}) < 0$ for vectors from the first class, and $> 0$ for the second class. This is obviously possible only for data that are linearly separable. If this is not the case, a subset of all vectors $\mathcal{D}(\mathbf{W}_1)$ will give projections on the wrong side of the zero point, inside $[a(\mathbf{W}_1), b(\mathbf{W}_1)]$ interval that contains the zero point. Visualization may help to separate the remaining $\mathcal{D}(\mathbf{W})$ vectors. In case of linear SVM the best additional directions may be obtained by repeating SVM calculations in the space orthogonalized to the already obtained $\mathbf{W}$ directions. One may also use

only the subset of $\mathcal{D}(\mathbf{W})$ vectors, as the remaining vectors are already separated in the first dimension. SVM training in its final phase is using anyway mainly vectors from this subset. However, vectors in the $[a(\mathbf{W}_1), b(\mathbf{W}_1)]$ interval do not include some outliers and therefore may lead to significantly different direction.

In two dimensions the classification rule is:

- If $g_{\mathbf{W}_1}(\mathbf{X}) < a(\mathbf{W}_1)$ Then Class 1
- If $g_{\mathbf{W}_1}(\mathbf{X}) > b(\mathbf{W}_1)$ Then Class 2
- If $g_{\mathbf{W}_2}(\mathbf{X}) < 0$ Then Class 1
- If $g_{\mathbf{W}_2}(\mathbf{X}) > 0$ Then Class 2

where the $[a(\mathbf{W}_1), b(\mathbf{W}_1)]$ interval is determined using estimates of posterior probabilities $p(C|x)$ from smoothed histograms, with a user-determined confidence parameter (for example $p(C|x) > 0.9$ for each class). One could also introduce such confidence intervals for the $\mathbf{W}_2$ direction and reject vectors that are inside this interval. An alternative is to use the nearest neighbor rule after dimensionality reduction.

This process may be repeated to obtain more dimensions. Each additional dimension should help to decrease errors, and the optimal dimensionality is obtained when new dimensions stop decreasing the number of errors in crossvalidation tests. If more dimensions is generated rules will be applied in sequential manner with appropriate intervals, and only for the last dimension zero is used as a threshold. In this way hierarchical system of rules with decreasing reliability is created. Of course it is possible to use other models on the $\mathcal{D}(\mathbf{W}_1)$ data, for example Naive Bayes approach, but we shall not explore this possibility concentrating mainly on visualization.

In case of non-linear kernel, $g_{\mathbf{W}}(\mathbf{X})$ provides the first direction, while the second direction may be generated in several ways. The simplest approach is to repeat training on $\mathcal{D}(\mathbf{W})$ subset of vectors that are close to the hyperplane in the extended space using some other kernel, for example a linear kernel.

## 3 Illustrative examples

The usefulness of the SVM-based sequential visualization method has been evaluated on a large number of datasets. Here only two artificial binary datasets, and three medical datasets downloaded from the UCI Machine Learning Repository [15] and from [16], are presented as an illustration. A summary of these data sets is presented in Tab. 1. Short description of these datasets follows:

1. **Parity_8:** 8-bit parity dataset (8 binary features and 256 vectors).
2. **Heart** disease dataset consists of 270 samples, each described by 13 attributes, 150 cases belongs to group "absence" and 120 to "presence of heart disease".
3. **Wisconsin** breast cancer data [17] contains 699 samples collected from patients. Among them, 458 biopsies are from patients labeled as "benign", and 241 are labeled as "malignant". Feature six has 16 missing values, removing these vectors leaves 683 examples.

4. **Leukemia:** microarray gene expressions for two types of leukemia (ALL and AML), with a total of 47 ALL and 25 AML samples measured with 7129 probes [16]. Visualization is based on 100 best features from simple feature ranking using FDA index.

| Title | #Features | #Samples | #Samples per class | | Source |
|---|---|---|---|---|---|
| Parity_8 | 8 | 256 | 128 $C_0$ | 128 $C_1$ | artificial |
| Heart | 13 | 270 | 150 "absence" | 120 "presence" | [15] |
| Wisconsin | 10 | 683 | 444 "benign" | 239 "malignant" | [17] |
| Leukemia | 100 | 72 | 47 "ALL" | 25 "AML" | [16] |

**Table 1.** Summary of datasets used for illustrations

For each dataset four two-dimensional mappings have been created using MDS, PCA, FDA and SVM-based algorithms described in Sec. 2. Results are presented in Figs. 1-5.

High-dimensional parity problem is very difficult for most classification methods. Many papers have been published on special neural models for parity functions, and the reason is quite obvious, as Fig. 1 illustrates: linear separation cannot be easily achieved because this is a $k$-separable problem that should be separated into $n+1$ intervals for $n$ bits [18, 19]. PCA and SVM find a very useful projection direction $[1, 1..1]$, but the second direction does not help at all. MDS is completely lost, as it is based on preservations of Euclidean distances that in this case do not carry useful information for clustering. FDA shows significant overlaps for projection on the first direction. This is a very interesting example showing that visualization may help to solve a difficult problem in a perfect way even though almost all classifiers will fail.

Variations on this data include random assignment of classes to bit strings with fixed number of 1 bits, creating $k$-separable ($k \leq n$) data that most methods invented for the parity problem cannot handle [18]. All three linear projections show for such data correct cluster structure along the first direction. However, linear projection has to be followed by a decision tree or the nearest neighbor method, as the data is nonseparable.

For Cleveland Heart data linear SVM gives about $83 \pm 5\%$ accuracy, with the base rate of 54%. Fig. 2 shows nice separation of a significant portion of the data, with little improvement due to the second dimension. MDS and PCA are somewhat less useful than FDA and SVM projections.

Displaying class-conditional probability for Parity and Cleveland Heart in the first SVM direction (Fig. 3) may also help to estimate the character of overlaps and the resulting errors, and help to decide what type of transformation should follow initial projection.

Wisconsin breast cancer dataset can be classified with much higher accuracy, around $97 \pm 2\%$, and therefore shows strong separation (Fig. 4), with benign cancer cases clusterized in one area, and a few outliers that appear far from

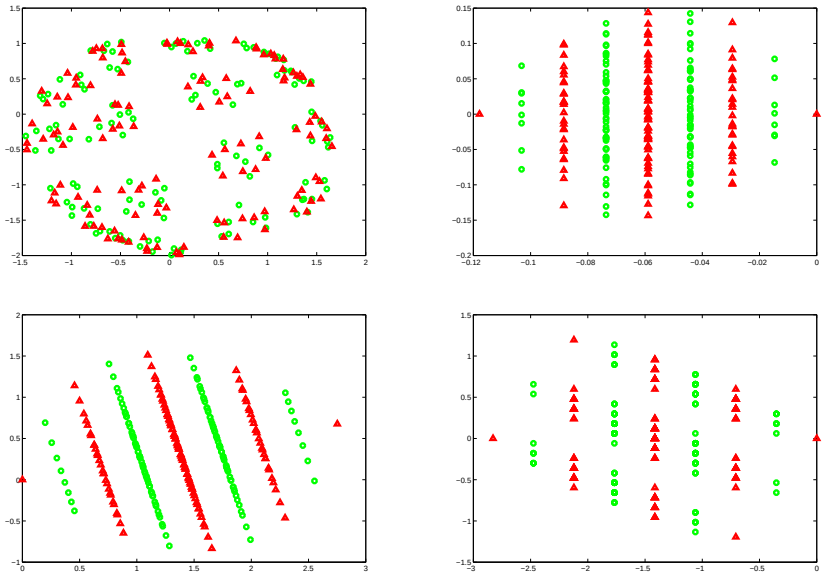

**Fig. 1.** 8-bit parity dataset, top row: MDS and PCA, bottom row: FDA and SVM.

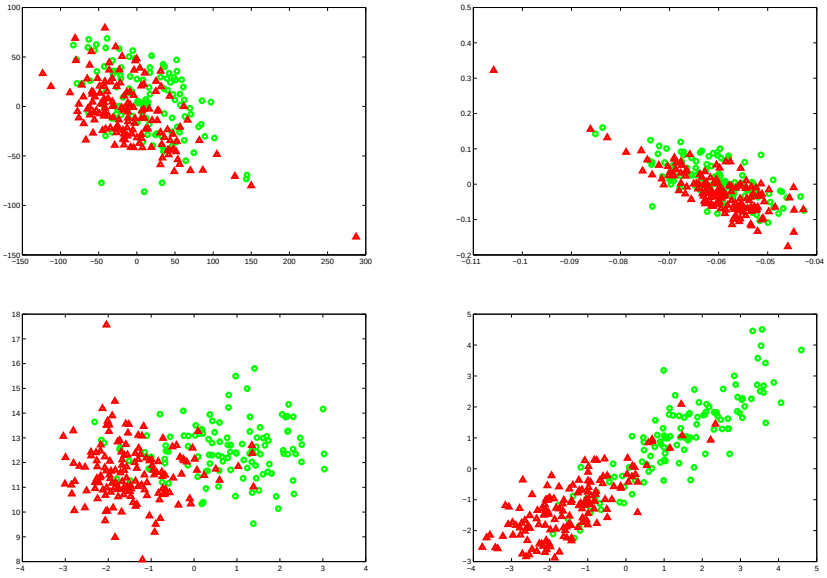

**Fig. 2.** Heart data set, top row: MDS and PCA, bottom row: FDA and SVM.

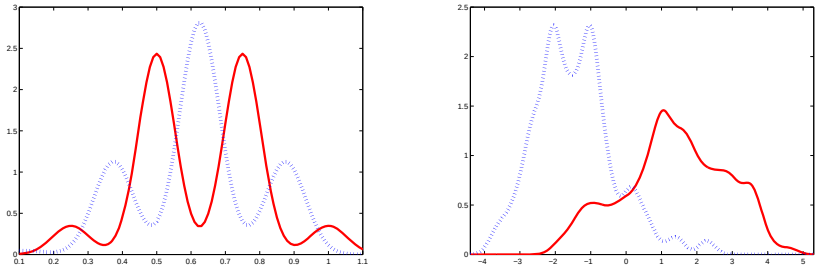

**Fig. 3.** Estimation of class-conditional probability for Parity and Cleveland Heart in the first SVM direction.

the main benign cluster, mixing with malignant cases. Most likely these are real misdiagnosed outliers that should in fact be malignant. Only in case of SVM the second direction shows some additional improvement.

Leukemia shows remarkable separation using two-dimensional SVM projection (Fig. 5), thanks to maximization of margin, providing much more interesting projection than other methods. The first direction shows some overlap but in crossvalidation tests it yields significantly better results than the second direction.

To compare the influence of dimensionality reduction on accuracy of classification for each dataset classification using SVM with linear kernel has been performed in the original and in the reduced two-dimensional space. 10-fold crossvalidation tests have been repeated 10 times and average results collected in Table 2, with accuracies and standard deviations for each dataset. These calculations intend to illustrate the efficiency of dimensionality reduction only; in case of Leukemia starting from pre-selected 100 features from the microarray data does not guarantee correct evaluation of generalization error (feature selection should be done within crossvalidation in order to do it). With such a large number of features and a very few samples, SVM with Gaussian kernel will show nicely separated class-conditional probabilities, but in crossvalidation will perform poorly, showing that strong overfitting occurs.

For Heart and Wisconsin data both FDA and SVM give significantly better results than other methods used in this comparison, with SVM achieving much better results than FDA for Leukemia, as should also be obvious from data visualization. Adding the second SVM direction to the first one has obviously negligible influence on the SVM results. However, visualizations show (Fig. 1) that for highly non-separable types of data the linear SVM projection may still be useful for dimensionality reduction and should be used for preprocessing of data for other classification or regression algorithms. The Gaussian kernel SVM fails as badly as linear SVM on some types of data, but it may work perfectly well on the reduced data. This is true not only for Boolean problems with complex logic, but also for microarray data such as Leukemia, where crossvalidation results with

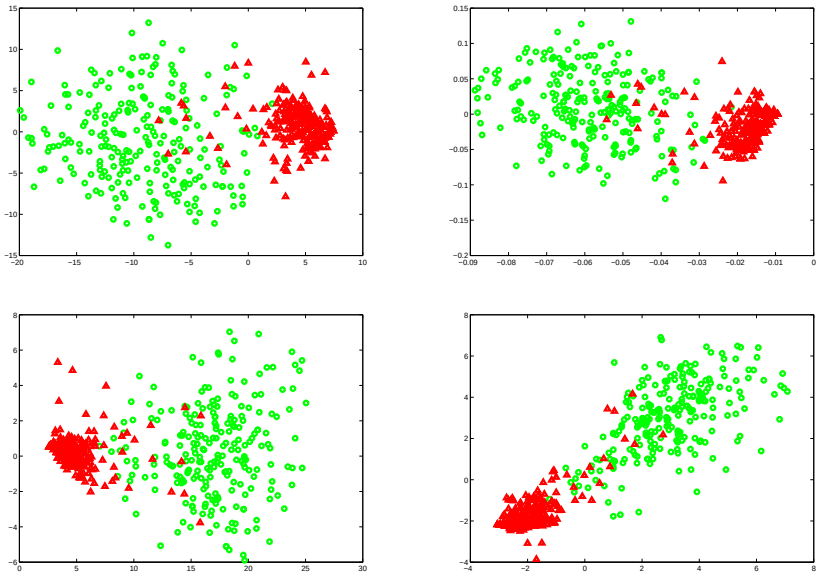

**Fig. 4.** Wisconsin data set, top row: MDS and PCA, bottom row: FDA and SVM.

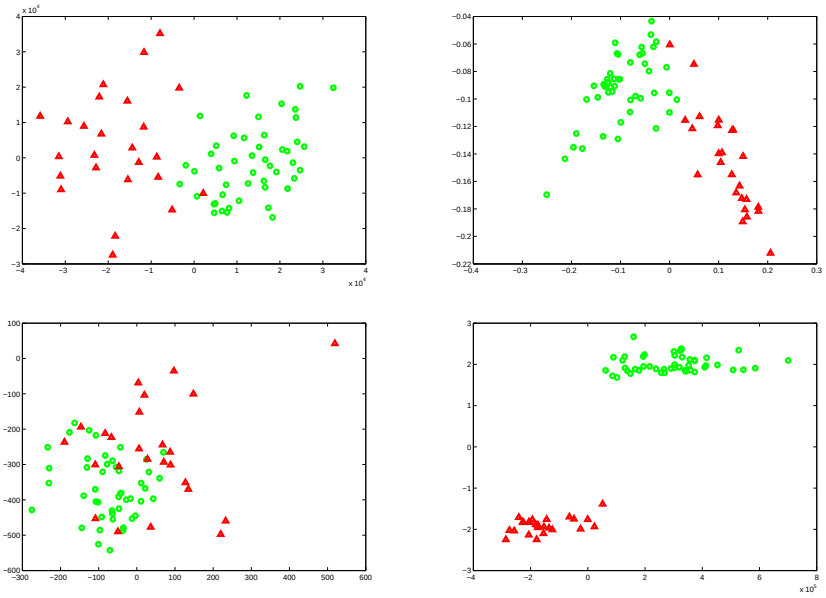

**Fig. 5.** Leukemia data set, top row: MDS and PCA, bottom row: FDA and SVM.

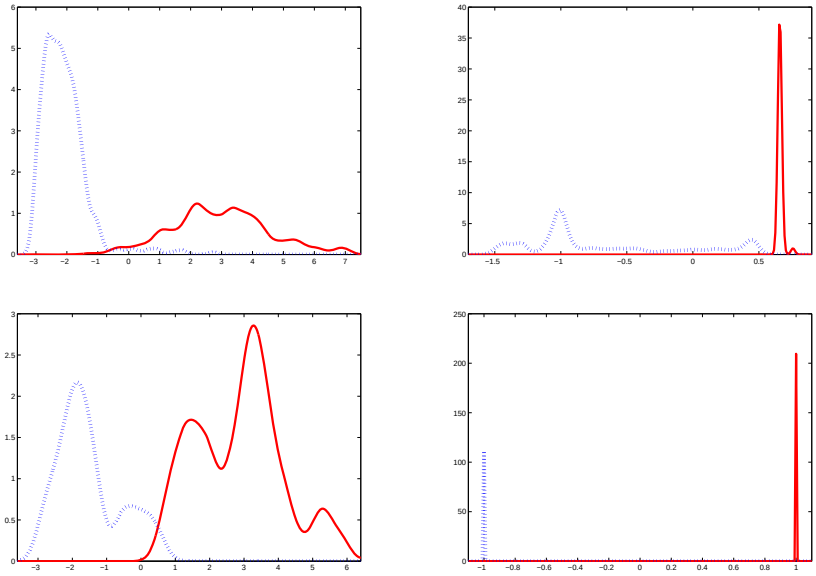

**Fig. 6.** Estimation of class-conditional probability for Wisconsin using linear and Gaussian kernel SVM (top row); the same for Leukemia (bottom row).

Gaussian kernel on the original data shows some error ($98.6 \pm 4.5\%$), while the same crossvalidation on the two-dimensional data consistently gives 100%.

## 4 Conclusions

There are many methods for data visualization, some of them quite sophisticated [20], with PCA and MDS among the most common. Visualization allows for exploratory data analysis, giving much more information than just global information about expected accuracy or probability of individual cases. In real applications visualization is sometimes used for initial data exploration, but rarely to the evaluation of the mapping implemented by predictors. Visualization can certainly help to understand what black box classifiers really do [3, 4]. In industrial, medical or other applications where safety is important evaluation of confidence in predictions, that may be done using visualization methods, is critical.

Sequential dimensionality reduction based on SVM has several advantages: it enables visualization, guarantees dimensionality reduction without loss of accuracy, increases accuracy of the linear discrimination model, is very fast and preserves simple interpretation. Information obtained from unsupervised methods, such as PCA or kernel PCA, provide directions of highest variance, but no information about reliability of classification. There is no reason why SVM

| | # Features | Parity_8 | Heart | Wisconsin | Leukemia |
|---|---|---|---|---|---|
| PCA | 1 | 41.76±6.24 | 55.56±8.27 | 65.00±5.98 | 65.23±15.62 |
| PCA | 2 | 41.69±5.30 | 55.56±8.27 | 65.00±5.98 | 77.55±19.10 |
| MDS | 1 | 39.66±5.76 | 60.26±9.31 | 97.00±2.00 | 60.18±18.05 |
| MDS | 2 | 38.22±5.40 | 68.63±9.00 | 96.65±2.10 | 94.46± 8.39 |
| FDA | 1 | 40.25±6.54 | 85.00±6.58 | 97.17±1.91 | 75.57±15.37 |
| FDA | 2 | 38.72±7.13 | 85.19±6.32 | 97.13±2.03 | 81.79±14.10 |
| SVM | 1 | 41.91±6.51 | 84.81±6.52 | 97.26±1.81 | 97.18± 5.68 |
| SVM | 2 | 41.84±6.16 | 84.81±6.52 | 97.26±1.81 | 97.18± 5.68 |
| | All | 31.41±4.80 | 83.89±6.30 | 96.60±1.97 | 95.36± 7.80 |

**Table 2.** 10-fold crossvalidation accuracy in % for four datasets with reduced features.

decision borders should not be visualized using estimations of class-dependent probabilities, or posterior probabilities $p(C|x)$ in one or two dimensions. This process gives insight into the character of data, helping to construct appropriate predictors by combining linear or non-linear projections with other data models, such as decision trees or nearest neighbor models. For highly non-separable data (with inherent complex logic, symbolic data) $k$-separability approach may be the most appropriate, for very sparse high-dimensional data linear projection on one or two directions may be followed by kernel methods [12, 14], prototype-based rules [21] or the nearest neighbor methods. Such observations allow for implementation of meta-learning as composition of transformations [22], for automatic discovery of the simplest and most reliable models. Visualization will also help to evaluate the reliability of predictions for individual cases, showing them in context of the known cases, and providing information about decision borders of classifiers. We plan to add visualization of probabilities and scatterograms to a few popular SVM packages soon.

## References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer Verlag (2006)
2. Pękalska, E., Duin, R.: The dissimilarity representation for pattern recognition: foundations and applications. World Scientific (2005)
3. Duch, W.: Visualization of hidden node activity in neural networks: I. visualization methods. In Rutkowski, L., Siekemann, J., Tadeusiewicz, R., Zadeh, L., eds.: Lecture Notes in Artificial Intelligence. Volume 3070. Physica Verlag, Springer, Berlin, Heidelberg, New York (2004) 38–43
4. Duch, W.: Coloring black boxes: visualization of neural network decisions. In: Int. Joint Conf. on Neural Networks, Portland, Oregon. Volume I. IEEE Press (2003) 1735–1740
5. Webb, A.: Statistical Pattern Recognition. J. Wiley & Sons (2002)
6. Torgerson, W.: Multidimensional scaling. i. theory and method. Psychometrika **17** (1952) 401–419
7. Sammon, J.: A nonlinear mapping for data structure analysis. IEEE Transactions on Computers **C18** (1969) 401–409

8. Duch, W.: Quantitative measures for the self-organized topographical mapping. Open Systems and Information Dynamics **2** (1995) 295–302
9. Cox, T., Cox, M.: Multidimensional Scaling, 2nd Ed. Chapman and Hall (2001)
10. Naud, A.: An Accurate MDS-Based Algorithm for the Visualization of Large Multidimensional Datasets. Lecture Notes in Computer Science **4029** (2006) 643–652
11. Jolliffe, I.: Principal Component Analysis. Springer-Verlag, Berlin; New York (1986)
12. Schölkopf, B., Smola, A.: Learning with Kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA (2001)
13. Cheng, Y.Q., Zhuang, Y.M., Yang, J.Y.: Optimal Fisher discriminant analysis using the rank decomposition. Pattern Recognition **25**(1) (1992) 101–111
14. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other Kernel-Based Learning Methods. Cambridge University Press (2000)
15. Merz, C., Murphy, P.: UCI repository of machine learning databases (1998-2004) http://www.ics.uci.edu/∼mlearn/MLRepository.html.
16. Golub, T.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science **286** (1999) 531–537
17. Wolberg, W.H., Mangasarian, O.: Multisurface method of pattern separation for medical diagnosis applied to breast cytology. In: Proceedings of the National Academy of Sciences. Volume 87., U.S.A. (1990) 9193–9196
18. Grochowski, M., Duch, W.: Learning highly non-separable Boolean functions using Constructive Feedforward Neural Network. Lecture Notes in Computer Science **4668** (2007) 180–189
19. Duch, W.: $k$-separability. Lecture Notes in Computer Science **4131** (2006) 188–197
20. van der Maaten, L., Postma, E., van den Herik, H.: Dimensionality reduction: A comparative review. in print (2008)
21. Duch, W., Blachnik, M.: Fuzzy rule-based systems derived from similarity to prototypes. In Pal, N., Kasabov, N., Mudi, R., Pal, S., Parui, S., eds.: Lecture Notes in Computer Science. Volume 3316. Physica Verlag, Springer, New York (2004) 912–917
22. Duch, W., Grudziński, K.: Meta-learning via search combined with parameter optimization. In Rutkowski, L., Kacprzyk, J., eds.: Advances in Soft Computing. Physica Verlag, Springer, New York (2002) 13–22