Visualization of Support Vector Machines with Unsupervised Learning

Lutz Hamel
Department of Computer Science and Statistics
University of Rhode Island, Kingston, RI 02881
lutz@inductive-reasoning.com

***Abstract* – The visualization of support vector machines in realistic settings is a difficult problem due to the high dimensionality of the typical datasets involved. However, such visualizations usually aid the understanding of the model and the underlying processes, especially in the biosciences. Here we propose a novel visualization technique of support vector machines based on unsupervised learning, specifically self-organizing maps. Conceptually, self-organizing maps can be thought of as neural networks that investigate a high-dimensional data space for clusters of data points and then project the clusters onto a two-dimensional map preserving the topologies of the original clusters as much as possible. This allows for the visualization of high-dimensional datasets together with their support vector models. With this technique we investigate a number of support vector machine visualization scenarios based on real world biomedical datasets.**

I. Introduction

Support vector machines represent a powerful new machine learning paradigm introduced in the early 1990's by Vapnik [1] based on kernel functions. Although these algorithms exhibit excellent machine learning performance it is often difficult to obtain an intuitive understanding of the induced model[1], since support vector machines do not share the same transparency that decision trees [2] or inductive logic programming models [3] possess. However, an intuitive understanding of the obtained classifier is important, particularly in the biosciences, not only for the validation of the model but also to deepen the insight into the underlying biological processes.

Support vector machine models consist of points in data space (the "support vectors") that identify a separating hyperplane between classes. The task of the support vector machine algorithm is to identify such a set of support vectors in a given dataset. Understanding where the support vectors are located with respect to the overall dataset and the kind of decision surface they induce provides substantial insight into the model.

Visualization of support vector models is a difficult problem due to the high-dimensionality of the typical dataset. Here we propose a visualization technique of support vector machines that makes use of unsupervised learning in order to compute an appropriate visualization of the given dataset together with the support vector machine model. More specifically, we use self-organizing maps [4] to visualize the data and the support vector model.

Conceptually, self-organizing maps can be thought of as neural networks that investigate a high-dimensional data space for clusters of data points and then project the clusters onto a two-dimensional map preserving the topologies of the original clusters as much as possible. In the simplest case, where we have two linearly separable clusters in high-dimensional space, each representing a different class, we would expect the self-organizing map to project the clusters onto the map with the support vectors and the discriminating hyperplane appropriately placed between them. Section IV A. describes this baseline experiment with a three-dimensional dataset.

It is interesting to observe that our visualization naturally guides further analysis of a given support vector model. For instance, when we observe simple clusters with smooth decision boundaries between them on the projected map for high-dimensional data, we can infer that a lower dimensional subspace exists that allows for the near perfect discrimination between the classes. In this case, a dimension reduction or feature selection on the original dataset would seem appropriate to simplify the support vector model. We show that the support vector model of the original, high-dimensional data can be used to suggest how to approach this dimension reduction [5]. The converse is true as well; if we observe an intricate cluster structure with complicated decision boundaries we can assume that a high-dimensional subspace is necessary in order to be able to discriminate between the various classes. This in turn implies that we need a complex support vector model to describe the induced decision surface.

The approach to the visualization of support vector machines proposed here exhibits two major advantages over existing techniques: (1) No preprocessing of the data is necessary for the visualization, that is, neither do we need to perform feature selection nor do we have to guess which features to choose for the display. (2) It seems that the projection of high dimensional data onto a two-dimensional map can provide a "big picture" overview of the support vector decision surface not possible with other visualization approaches. In some sense we can say that our approach is decision boundary oriented where as existing techniques are

[1] Here we only consider support vector machine classification.

1-4244-0623-4/06/$20.00 ©2006 IEEE.

feature oriented. We have implemented a prototype that produces PDF maps of the dataset projections and the visualization of the support vector models.[2]

The remainder of the paper is structured as follows: Section II introduces some basic notions on support vector machines. Section III briefly introduces self-organizing maps. Section IV describes various visualization experiments; most notably we describe three experiments with bio-medical data. Section V discusses related work in more detail. We conclude the paper with final remarks and notes on further research in Section VI.

II. Support Vector Machines

Support vector machines were introduced in the early 1990's as a new breed of classification algorithms based on optimal margins [1, 6, 7]. Conceptually, especially in the case of two linearly separable classes, support vector machines are fairly straight forward: find a hyperplane that separates the two classes in such a way that the hyperplane is equidistant from both clusters. In technical jargon: we aim to compute a hyperplane that maximizes the margin between the two classes.

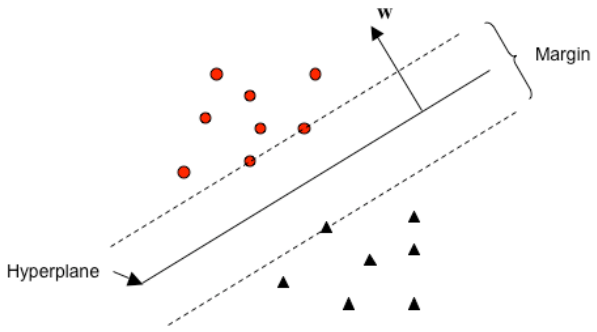

Fig. 1: Optimal margin hyperplane separating two classes.

Fig. 1 shows an optimal margin hyperplane separating two classes. We can construct a classifier f based on this hyperplane that can classify any unknown point $\mathbf{x}$,

$$f(\mathbf{x}) = label(\mathbf{w} \bullet \mathbf{x} + b), \tag{1}$$

where

$$label(n) = \begin{cases} \text{ball} & \text{if } n \geq 0, \\ \text{triangle} & \text{if } n < 0, \end{cases} \tag{2}$$

$\mathbf{w}$ is the normal vector of the given hyperplane, $\mathbf{w} \bullet \mathbf{x}$ represents the dot product between the normal $\mathbf{w}$ and some point $\mathbf{x}$, and b is the intercept. Instead of assigning symbolic labels to the classes it is usual to assign the numeric labels +1 and -1 in such a way that the +1 label is assigned to the class being pointed to by $\mathbf{w}$,

$$label(n) = \begin{cases} +1 & \text{if } n \geq 0, \\ -1 & \text{if } n < 0, \end{cases} \tag{3}$$

[2] Available at http://homepage.cs.uri.edu/faculty/hamel/cibcb2006

simplifying the underlying mathematics.

Notice that in Figure 1 the margin is limited by three data points. These three points fully define the margin and with it the orientation of the decision surface. These three points are called the *support vectors*. The goal of training a support vector machine with a given dataset is to identify such support vectors and with them the optimal decision surface. The training algorithm can be stated as a quadratic programming problem,

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m} \alpha_i, \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i \alpha_i = 0, \\ & \alpha_i \geq 0,\ i = 1,\ldots,m, \end{aligned} \tag{4}$$

where y_l represents the numeric class label for data point $\mathbf{x}_l$, and α_l is the support vector coefficient for data point $\mathbf{x}_l$. The indices i and j are indices over all the points m of a dataset. In general, the value of α_l is 0 for any point $\mathbf{x}_l$ not considered a support vector. A number of well established algorithms exist to optimize (4), e.g. [8, 9].

Perhaps the most interesting feature of (4) is the use of the kernel function (or just kernel) k. When applying support vector machines to particular datasets the user typically has a choice of which kernel to use. Choices for kernel functions include the *linear kernel* which is just the dot product in data space,

$$k(\mathbf{x}, \mathbf{z}) = \mathbf{x} \bullet \mathbf{z}, \tag{5}$$

the *polynomial kernel* of degree d,

$$k(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \bullet \mathbf{z} + 1)^d, \tag{6}$$

and the *gaussian kernel*,

$$k(\mathbf{x}, \mathbf{z}) = \exp\ -\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\sigma}, \tag{7}$$

where σ is a free parameter. Here, the construction $\|\mathbf{x} - \mathbf{z}\|$ typically represents the Euclidean distance between vectors $\mathbf{x}$ and $\mathbf{z}$. The polynomial and gaussian kernels allow for the construction of non-linear hypersurfaces for the separation of the classes in data space.

In the case that none of the induced models can separate the classes perfectly, support vector machines allow for models to admit certain modeling exceptions. At the theoretical level this is accomplished via *slack variables*. At the algorithmic level the only thing that changes is that the support vector coefficients are further constrained compared to (4),

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) - \sum_{i=1}^{m} \alpha_i, \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i \alpha_i = 0, \\ & C \geq \alpha_i \geq 0,\ i = 1,\ldots,m, \end{aligned} \tag{8}$$

to reflect the fact that some data points have to be modeled by making them explicit support vectors that do not necessarily support the decision surface directly. Without this further constraint value C the support vector coefficients for these modeling exceptions would grow to infinity. The value of C is a user defined constant at model construction time and is called the "cost" value.

We can construct a binary classifier f in this general setting given the support vector coefficients and the kernel function,

$$f(\mathbf{x}) = label(\sum_{i=1}^{m} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) - b), \quad (9)$$

the offset b can also be computed from the support vector coefficients. This is easily extended to the multi-class setting by using schemes such as *one-against-the-rest* [7].

A. Feature Selection with Linear Kernels

In general, it is difficult to ascertain how a particular support vector machine uses dataset attributes or features. That is especially true for the non-linear models produces by the polynomial and gaussian kernels. Here we outline an approach developed in [5] based on linear models. If we have a linear support vector model, then we can construct the normal **w** to the decision surface using the support vectors α_i,

$$\mathbf{w} = \sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i. \quad (10)$$

Now, the orientation of the normal vector holds substantial information on the relative importance of the features in the dataset, that is, the larger the projected component of the normal onto a particular dimension, the higher the importance of that particular feature.

If we are dealing with a non-linear model we can approximate the non-linear model with a linear model making use of slack variables and the associated modeling exceptions in order to obtain an approximate importance ranking on the features.

III. Self-Organizing Maps

Self-organizing maps [4] were introduced by Kohonen in 1982 and can be viewed as tools to visualize structure in high-dimensional data. Self-organizing maps are considered members of the class of unsupervised machine learning algorithms, since they do not require a predefined concept but will learn the structure of a target domain without supervision.

Typically, a self-organizing map consists of a rectangular grid of processing units. Multidimensional observations are represented as feature vectors. Each processing unit in the self-organizing map also consists of a feature vector called a reference vector or reference model. The goal of the map is to assign values to the reference models on the map in such a way that all observations can be represented on the map with the smallest possible error. However, the map is constructed under constraints similar to regression surfaces in multiple-regression analysis in the sense that the reference models cannot take on arbitrary values but are subject to a smoothing function called the neighborhood function. During training the values of the reference models on the map become ordered so that similar reference models are close to each other on the map and dissimilar ones are further apart from each other.

The training of the map is carried out by a sequential regression process, where t = 1, 2,... is the step index. For each observation $\mathbf{x}(t)$, we first identify the index c of some reference model which represents the best match in terms of Euclidean distance by the condition,

$$c = \underset{i}{\operatorname{argmin}} \left\| \mathbf{x}(t) - \mathbf{m}_i(t) \right\|, \ \forall i. \quad (11)$$

Here, the index i ranges over all reference models on the map. Next, all reference models on the map are updated with the following regression rule where model index c is the reference model index as computed in (11),

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}[\mathbf{x}(t) - \mathbf{m}_i(t)], \ \forall i. \quad (12)$$

Here h_{ci} is the neighborhood function that is defined as follows,

$$h_{ci} = \begin{cases} 0 & \text{if } |c - i| > \beta, \\ \eta & \text{if } |c - i| \le \beta. \end{cases} \quad (13)$$

$|c - i|$ represents the distance between the best matching reference model c and some other reference model i on the map, β is the neighborhood distance and η is the learning rate. It is customary to express η and β also as functions of time. This regression is usually repeated over the available observations many times during the training phase of the map.

An advantage of self-organizing maps is that they have an appealing visual representation. Fig. 2 shows animals mapped onto a self-organizing map. Each animal is described by a set of 13 features [4] such as how many legs, does it possess feathers, does it hunt, *etc*. In effect, each animal can be considered a point in thirteen-dimensional space and the map in Fig. 2 is a two-dimensional projection of this thirteen-dimensional space.

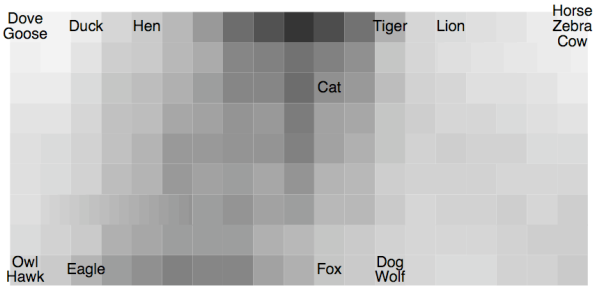

Fig. 2: Mapping animals on to a self-organizing map.

Each square in the map represents a reference model. The shading of the map represents the level of quantization or mapping error for the map: light shading represents a small quantization error; dark shading represents a large quantization error. Contiguous areas of low quantization error

represent clusters of similar entities. Besides low quantization error, proximity is also a clustering indicator: the further apart, the more dissimilar two objects on the map. For example, in Fig. 2 we find two major clusters: mammals and birds. Within these major clusters we can find sub-clusters such as large hooved mammals in the top right corner of the map and predatory birds in the bottom left corner.

In this paper we make use of this ability of self-organizing maps to visualize high-dimensional spaces in order to visualize high-dimensional biomedical datasets together with their support vector models.

IV. Visualization Experiments

We discuss four experiments. The first experiment is a baseline experiment with a synthetic dataset illustrating the basic functionality of our visual approach. The remaining three datasets are biomedical datasets.

A. Baseline Visualization

The first visualization is intended to provide an overview of the functionality of our technique and can also be thought of as a baseline – we test whether the support vectors are mapped to reasonable locations on the self-organizing map. The actual dataset is given in Table 1.

TABLE 1
A Linearly Separable Synthetic Dataset.

ID	X	Y	Z	Class	alpha
1	0	0	0	A	0.00
2	1	0	0	A	0.81
3	0	1	0	A	1.00
4	0	0	1	A	1.00
5	4	0	0	B	0.90
6	0	4	0	B	0.95
7	0	0	4	B	0.95
8	5	5	5	B	0.00

It is a three-dimensional dataset and a quick inspection reveals that the classes A and B are linearly separable in an optimal way by a plane that intersects the axes at (2.5, 0, 0), (0, 2.5, 0), and (0, 0, 2.5) (see Fig. 3, the balls represent class A and the squares class B). The columns of the table should be self-explanatory with the exception of the right most one. This column lists the α-values for each data point computed by a linear support vector machine separating the two classes A and B. Given the position of the separating hyperplane, it should be no surprise that all six data points close to the decision surface are chosen as support vector, i.e., have an α-value > 0. In Fig. 3 the support vectors are indicated with the asterisks. Conversely, the points at (0, 0, 0) and (5, 5, 5) are not chosen as support vectors; their α-values are 0.

Fig. 4 shows the projection of the dataset in Table 1 onto a two-dimensional self-organizing map. The classes are indicated with their corresponding letters. The numbers in parentheses are the ID numbers of the individual points given in Table 1. Finally, support vectors are again indicated via asterisks.

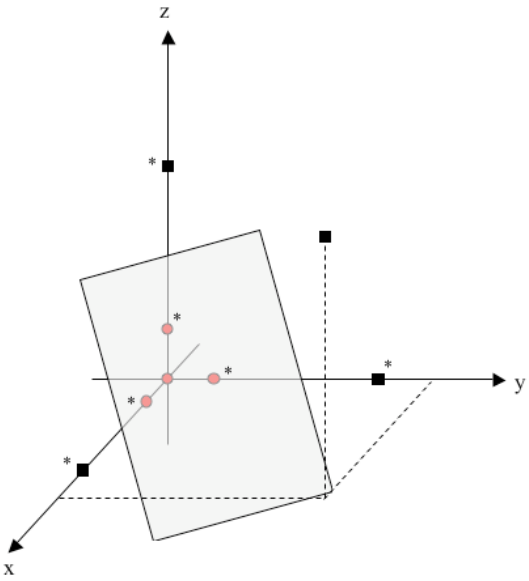

Fig. 3: Two classes with their separating hyperplane and the corresponding support vectors (asterisks).

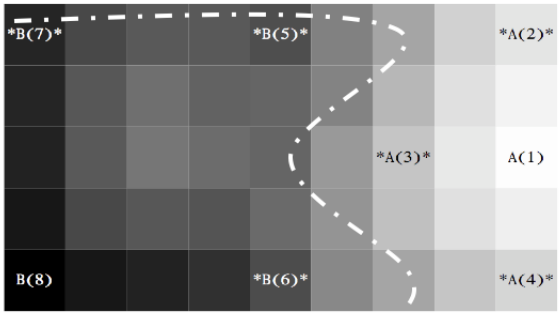

Fig. 4: Visualization of data points, support vectors, and decision surface on a self-organizing map.

A number of interesting observations can be made on the projected data. First, as one would expect, the linear decision surface in three-dimensional space is projected as higher-order polynomial surface in two-dimensional space. The projected decision surface is indicated on the map with a dashed white line. Second, we can observe that the support vectors are mapped along the decision surface. Third, the data points classified by the decision surface but which themselves are not support vectors lie in appropriate regions of the map suitably demarked by support vectors. Finally, class A forms a much tighter cluster than class B, that is, the map displays a much smaller quantization error for class A than class B. This seems self-evident given the data structure shown in Fig. 3. Here, class A forms a tight cluster around the origin and class B forms a loose cluster around the point (5, 5, 5).

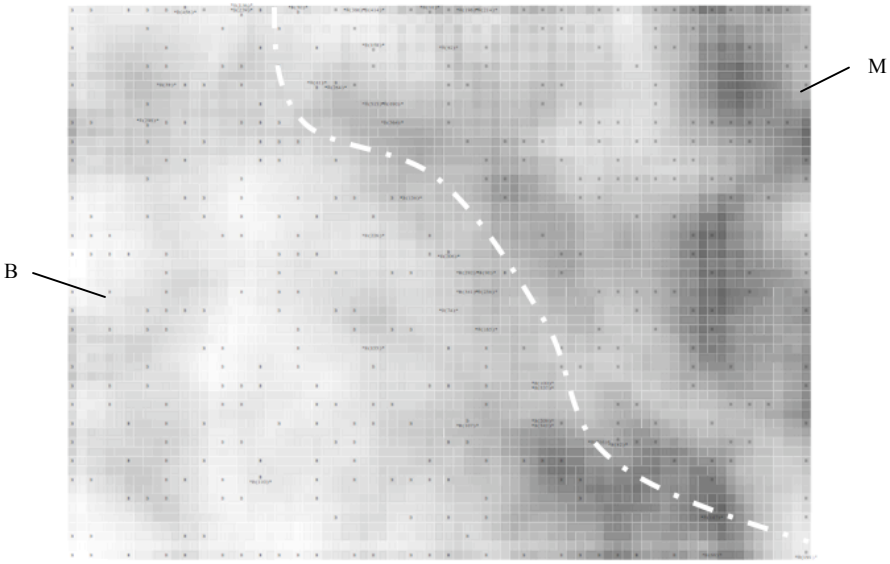

Fig. 5: Visualization of the Wisconsin breast cancer data with the support vector model and the decision surface.

As the last thing we investigate how the support vector machine uses the features X, Y, and Z. Using (10) from above allows us to compute the normal **w** of the induced decision surface given the α-values of the support vectors. The normal here computes to **w** = (-2.81, -2.81, -2.81), that is, all three features are of equal importance and class A is assigned the numeric label +1.

B. The Wisconsin Breast Cancer Dataset

Our next experiment is the visualization of a support vector model of the Wisconsin breast cancer dataset [10] publicly available from the UCI dataset repository [11]. The dataset contains 569 instances of fine needle cell aspirates of which 212 are malignant and 357 are benign. Each instance is described by thirty features such as cell radius, smoothness of the cell surface, concavity among others. It is known that in thirty-dimensional space the two classes are linearly separable and a number of techniques have been used to construct linear classifiers, e.g. [12]. Here we construct a linear support vector machine in thirty-dimensional space and then visualize this model with a self-organizing map. Fig. 5 shows this map. Unfortunately, due to space constraints we cannot show the map in its original size and therefore the labels are not very readable, but the map is divided into two regions: a malignant region (right side) and a benign region (left side).[3] With the exception of minor irregularities right at the boundary between the two classes the decision surface is smooth considering that we are projecting a thirty-dimensional space onto two dimensions. Also, note that the support vectors (long labels) are placed along the decision surface. There are a few exceptions and we could consider "drilling through" to the instances representing these support vectors to find out why these instances are considered outliers by the self-organizing map and if there is any biological significance to that (to be studied in a follow-on paper).

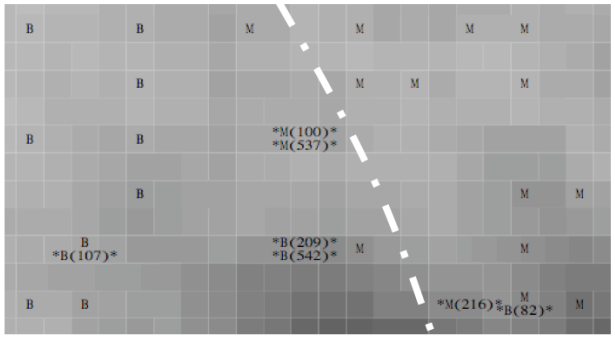

Fig. 6**:** Detailed view of a section of Wisconsin breast cancer data self-organizing map.

In order to provide a more concrete sense of the kind of information shown on the map in Fig. 5, Fig. 6 displays a detailed view of a section of the map right at the border between the two classes. The decision surface is indicated with the dashed line. The labels B and M indicate that benign and malignant instances were mapped to these locations, respectively. Support vectors are indicated with asterisks. The number in parentheses is the instance identifier of the support vector data points.

Given that the two classes can be projected so smoothly from a thirty-dimensional space onto a two-dimensional surface begs the question whether a much lower dimensional subspace exists that allows for the near perfect classification of the dataset. Plotting the features using the feature selection as outlined above we obtain the graph given in Fig. 7. From this graph it seems that a four-dimensional feature space might be sufficient for the classification of the dataset. Additional experiments confirm this; reducing the number of features does not significantly alter the map.

[3] Maps available at http://homepage.cs.uri.edu/faculty/hamel/cibcb2006

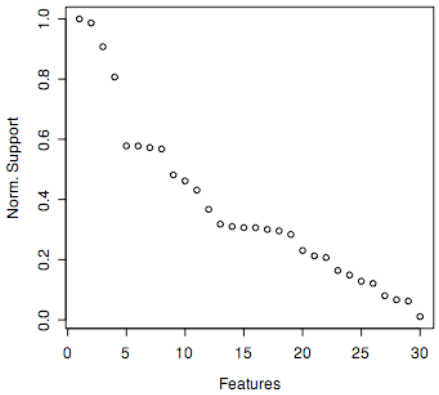

Fig. 7**:** Feature ranking of the Wisconsin breast cancer data.

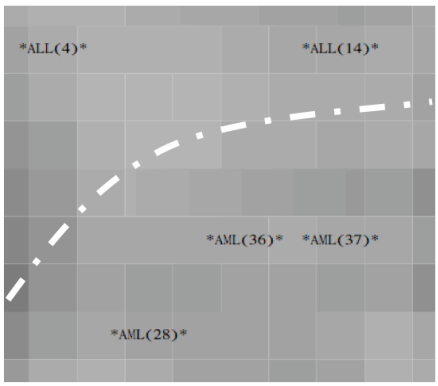

Fig. 9**:** Detail of the AML-ALL dataset visualization.

C. The AML-ALL Leukemia Dataset

Our third visualization experiment uses the AML-ALL leukemia dataset published by the Broad Institute at MIT [13]. The goal is to distinguish between acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The dataset consists of 38 bone marrow samples (27 ALL and 11 AML; we are using only the training set, a test set is also available from the website), over 7129 probes from 6817 human genes. The two classes can be separated perfectly with a linear support vector machine with cost C=1. Fig. 8 depicts the dataset together with the support vector model and the decision surface. The AML class is at the bottom right of the map and the remainder of the map is dedicated to the ALL class. Due to space limitations the map is not very readable, an excerpt of the map is given in Fig. 9.

Considering that we are projecting a 7000-dimensional space onto a two-dimensional map the decision surface is remarkably smooth suggesting that there exists a much lower dimensional space that allows for the classification of the instances. On the other hand, due to the fact that the separating hyperplane is induced in 7000-dimensional space we have support vectors that support the hyperplane in 7000-dimensional space but are projected onto locations very far away from the decision surface on the map.

A glance at the feature ranking (Fig. 10) confirms our suspicion, only a handful of features are ranked highly, the majority of the features has less than half the support than the highly ranked features. Performing feature selection on the original dataset with the top five features given in Fig. 10 (gene accession numbers: M96326_rna1_at, M25079_s_at, Z19554_s_at, M27891_at, and Y00433_at) and recomputing the support vector model and the self-organizing map gives us Fig. 11. Here the top right of the map contains the AML class and the remainder of the map is dedicated to the ALL class. Notice that less support vectors are necessary for this model and also notice that the majority of the support vectors are now close to the decision surface.

D. The Pap Smear Dataset

Our final experiment is the visualization of a Pap smear dataset [14]. The dataset consists of 52 instances where each

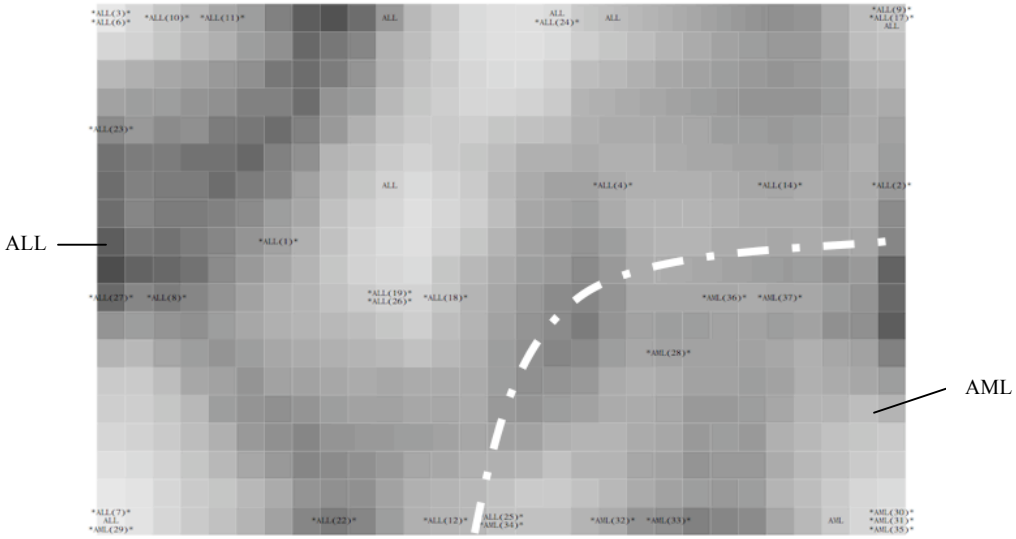

Fig. 8**:** The AML-ALL dataset visualization with model and decision surface.

instance consist of 251 wavenumbers and is classified as either precancerous (Y) or not (N). The wavenumbers represent near-infrared spectroscopy measurements ranging from 4,000 to 10,000 cm^{-1} of cervical cells. Of the 52 instances there were 32 normal, 9 precancerous, and 11 cancerous cells. The precancerous instances are particularly difficult to classify as they possess traits of both the cancerous as well as the normal cells. Here we use a polynomial model with degree seven and cost equal to 1 in order to classify the instances. Fig. 12 depicts the visualization of this dataset together with its support vector model and decision surface. Note that the decision surface is much more complicated than in any of the other experiments; in fact, it is broken into two regions. It is important to realize that the self-organizing map algorithm uses only data point information to compute the two-dimensional projection; no classification or model information flows into that computation. Given this, it is remarkable how well the independently computed support vector model follows the SOM projection. This also illustrates that our visualization technique works for non-linear models. Perhaps another interesting observation is that the bottom left region contains no generalizations. All data points in this region are support vectors. It would be interesting to investigate this further and determine a biomedical reason why this particular region is difficult to classify. Finally, in the top left corner we have two support vectors of opposite classes mapped to the same map element. This might indicate another region of the decision surface.

Given our hypothesis that complicated projections are due to non-trivial subspaces that contain the classification information, we should be able to observe this in the feature ranking for this dataset. In order to use our feature selection approach we approximate the polynomial model with a linear model and then rank the components of the hyperplane normal as above. Fig. 13 shows this feature ranking. Note that the curve in the ranking declines only gradually indicating that substantial information is contained within the lesser ranked features. Experiments indicate that one needs at least a

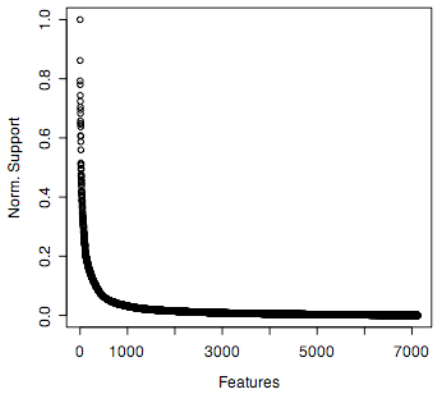

Fig. 10: Feature ranking of the AML-ALL dataset.

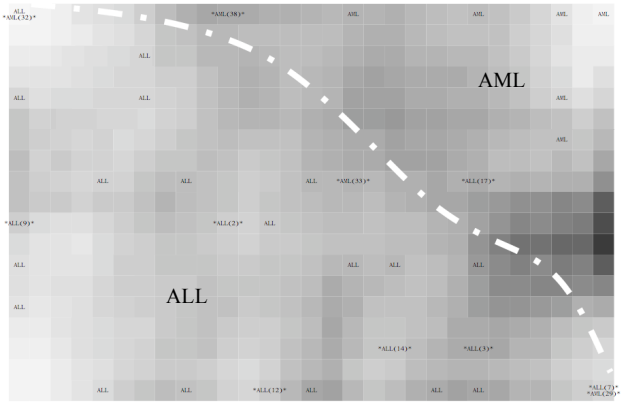

Fig. 11: Visualization of the low-dimensional AML-ALL dataset.

twenty-dimensional subspace in order to achieve reasonable classification accuracy. Further experimentation is necessary to see if the dimension reduction aligns the support vectors, especially from the N class, better with the decision surface.

V. Related Work

There has been relatively little work on visualizing support vector machine classifiers in realistic settings with the notable exceptions [15, 16, 17]. These approaches distinguish themselves from our approach in that they require considerable insight and preprocessing by the user with respect to the data features (which can be daunting in the case of the AML-ALL data set). In the typical setting with these approaches the user needs to pick a feature or a collection of features to be displayed. An advantage here is that many details on how a particular feature relates to the induced decision surface are visible. In particular, Poulet [15] allows the user to accept or reject support vectors thus in effect hand tuning the classifier. This is in contrast to our approach where we aim to provide an overall impression on how the decision surface relates to all data points and support vectors.

VI. Conclusions And Further Research

It seems that the approach to visualizing support vector machines in realistic settings put forth here is promising. It does convey a sense of the "big picture" of what the classifier looks like in high-dimensional spaces.

In our experiments we have touched upon ways that the visualization guides further analysis of the induced decision surfaces but we have not yet explored these possibilities in any depth. One interesting extension would be the ability to "drill through" from the map to the underlying data instances enabling an immediate inspection of outliers or support vectors much in the sense as advocated by [15]. We also need to further investigate the biological ramifications of these visualizations.

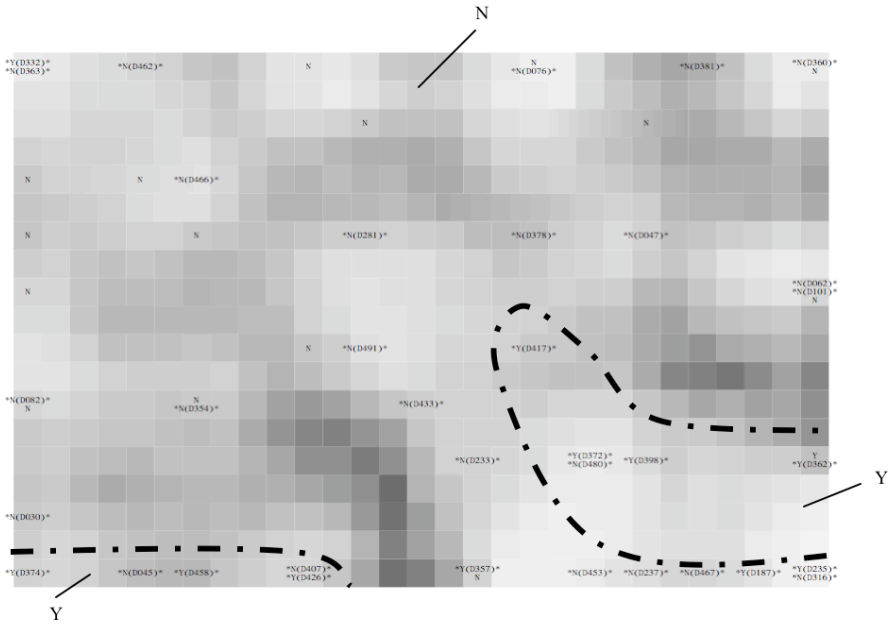

Fig. 12**:** The Pap smear dataset visualization with model and decision surface.

Another interesting research direction to pursue is to use manifold learning, such as locally linear embedding (e.g. [18]), instead of self-organizing maps. It would be interesting to see if in this setting the decision surfaces will be detected as distinguished structures perhaps reducing support vector outliers we have observed in the visualizations above.

Finally, in all of the maps above the decision surfaces are interpolated manually given the locations of the support vectors. It would greatly enhance the visualization if these interpolations could be done by machine on the projection maps.

ACKNOWLEDGEMENTS

The author would like to thank Professor Chris Brown from the Chemistry Department at the University of Rhode Island for providing access to the Pap smear dataset.

REFERENCES

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory,* 2nd ed., New York: Springer, 2000, pp. 314.

[2] L. Breiman, *Classification and Regression Trees.* Pacific Grove, Calif.: Wadsworth & Brooks/Cole, 1984, pp. 358.

[3] S. Muggleton and L. D. Raedt, "Inductive Logic Programming: Theory and Methods," *JLP,* vol. 19, pp. 629-679, 1994.

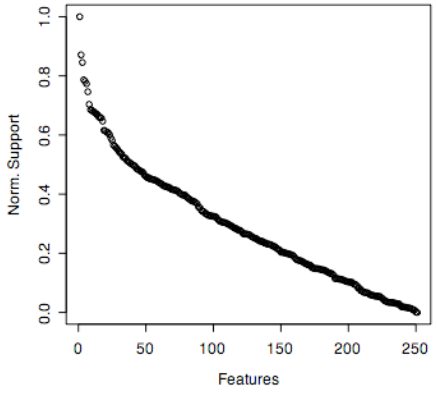

Fig. 13**:** Feature ranking of the Pap smear dataset.

[4] T. Kohonen, *Self-Organizing Maps,* 3rd ed., Berlin ; New York: Springer, 2001, pp. 501.

[5] J. Brank, M. Grobelnik, N. Milic-Frayling and D. Mladenic, "Feature selection using linear support vector machines," MSR-TR-2002-63, Microsoft Research 2002, 2002.

[6] K. P. Bennett and C. Campbell, "Support vector machines: hype or hallelujah?" *ACM SIGKDD Explorations Newsletter,* vol. 2, pp. 1-13, 2000.

[7] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2002.

[8] C. C. Chang and C. J. Lin. (2006, LIBSVM: A library for support vector machines. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm

[9] J. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," *Advances in Kernel Methods-Support Vector Learning,* pp. 185-208, 1999.

[10] W. Street, W. Wolberg and O. Mangasarian, "Nuclear feature extraction for breast tumor diagnosis," *Proc.SPIE,* vol. 1905, pp. 861-870, 1993.

[11] D.J. Newman, S. Hettich, C.L.Blake and C.J. Merz, 1998, UCI repository of machine learning databases. Available: http://www.ics.uci.edu/mlearn/MLRepository.html

[12] O. L. Mangasarian, W. N. Street and W. H. Wolberg, "Breast Cancer Diagnosis and Prognosis via Linear Programming," *Oper. Res.,* vol. 43, pp. 570-577, 1995.

[13] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing and M. Caligiuri, "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science,* vol. 286, pp. 531-537, 1999.

[14] Z. Ge, C. W. Brown and H. J. Kisner, "Screening Pap Smears with Near-Infrared Spectroscopy," *Appl. Spectrosc.,* vol. 49, pp. 432-436, 1995.

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

[16] D. Caragea, D. Cook and V. G. Honavar, "Gaining insights into support vector machine pattern classifiers using projection-based tour methods," *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* pp. 251-256, 2001.

[17] A. Jakulin, M. Možina, J. Demšar, I. Bratko and B. Zupan, "Nomograms for visualizing support vector machines," *Conference on Knowledge Discovery in Data,* pp. 108-117, 2005.

[18] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *Journal of Machine Learning Research,* vol. 4, pp. 119–155, 2003.