FreeViz - An Intelligent Visualization Approach for Class-Labeled Multidimensional Data Sets

Janez Demšar[1], **Gregor Leban**[1], **Blaž Zupan**[1,2]
Faculty of Computer and Information Science, University of Ljubljana, Slovenia
Department Molecular and Human Genetics, Baylor College of Medicine, Houston, TX
janez.demsar@fri.uni-lj.si

Abstract

Within biomedical data analysis, visualization can greatly improve data understanding and support various data mining tasks. The paper presents FreeViz, a visualization technique for analysis of class-labelled, multi-dimensional data. FreeViz visualizations can present data on many features in the same graph, but through optimization procedure choose a projection that best separates instances of different class. The paper gives mathematical foundations of FreeViz, and presents its utility on various biomedical data sets, including those with thousands of features from cancer gene expression studies.

1 Introduction

Medical data analysis may largely benefit from visualization. The *right* visualization may outline which factors govern the data and uncover their interactions. In the paper, we will be concerned with predictive data mining tasks, where each data instance (case) is described with a set of features (predictive variables) and labelled with a class (*e.g.* outcome, diagnosis). Despite many visualization techniques available, there are not too many of those that can visualize several features in the same graph, and, for instance, include scatterplot (two or three features, the later if plotted in 3D), parallel coordinates and RadViz (both for presentation of data using many features) [Keim, 2002].

When considering data sets with many features, which are typical in the domain of biomedicine, the principal problem to solve is which features to visualize and which projection to use, that is, how to order the selected features in the graph. With increasing number of features, any manual search for good projections becomes unfeasible. In principle, we would then prefer to use some automatic search for *good* projections, that would optimize some criteria for quality of interestingness. For a single-class (unsupervised) data, a well-known technique of projection pursuit is available for the task [Huber, 1985]. But interestingly, for class-labelled data, such *intelligent data analysis* approaches are at best rare, while the task is somehow better defined: interesting visualization is the one that well separates data instances of different class. We are aware of two approaches in this category, McCarthy *et al.*'s RadViz projections that place correlated features in RadViz close to each other and thus try to improve on class separation, and Leban *et al.*'s Vizrank [Leban *et al.*, 2005] that directly optimizes class separation and uses the heuristic search through projection space [McCarthy *et al.*, 2004].

In the paper, we propose an iterative algorithm that optimizes class separation in visualization of class-labelled data sets. The visualization it uses is based on RadViz [Brunsdon *et al.*, 1998], and is called FreeViz since it relaxes the constraints of placement of feature anchors; in RadViz, these are placed on the boundary of a circle. FreeViz is fast, can propose good visualizations even in the case of highly-dimensional data sets such as those from cancer genomics within seconds, and can be further used for feature subset selection and feature interaction discovery.

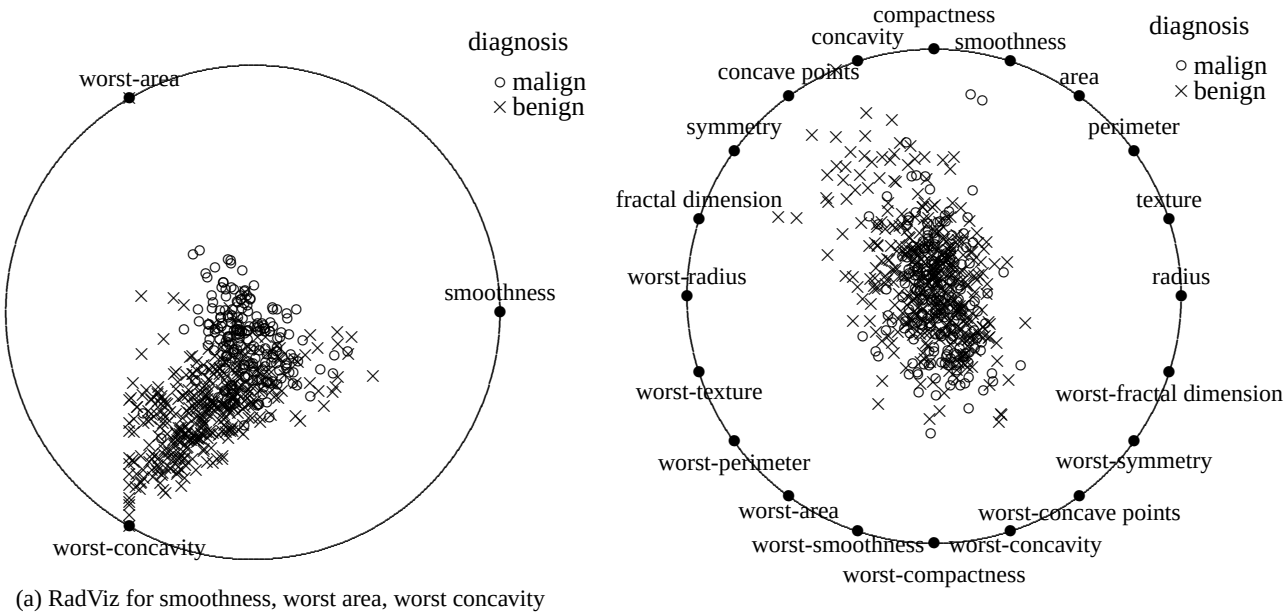
We first give the background on RadViz and its intelligent visualization counterpart VizRank. We formally describe FreeViz, present a mathematical derivation of its fitness (quality) function describe the corresponding implementation of the optimization algorithm. We then give several cases that show a utility of FreeViz in biomedical data analysis, also including examples that use large cancer gene expression data set. We conclude with discussion and ideas for further work.

Before we go on, notice that any modern visualization can largely benefit from colored display. Figures in the paper are printed in black and white, which at places significantly decreases their clarity. The reader is invited to visit a supplemental web page (www.ailab.si/supp/freeviz-idamap) for better images.

2 Background

RadViz [Brunsdon *et al.*, 1998] is a visualization that is suitable for data described with a set of continuous features scaled to the interval [0, 1]; discrete features can be visualized through first transforming them to continuous. The features are represented by anchors placed evenly on the unit circle. The data instances are plotted inside the circle; the position of each is determined by its features and the positions of the corresponding anchors. Informally, each anchor pulls the instance towards itself with a strength proportional to the value of the corresponding feature, so the position of an example depends upon the relative values of features (*e.g.* if all features have equal values, the instance is placed in the center).

Figure 1(a) shows a RadViz for three features (smoothness, worst area, worst concavity) of the Wisconsin Di-



(a) RadViz for smoothness, worst area, worst concavity

(b) RadViz using all twenty features

Figure 1: Two RadViz graphs for Wisconsin Breast Cancer Data

agnostic Breast Cancer data (WDBC) from the UCI ML repository [Blake and Merz, 1998]. The interpretation of such a graph is rather obvious: tissues with a large "worst area" tend to be malign and tissues with a large "worst concavity" are benign, while the role of smoothness is not clear. The problem arises when (or, better, because) the data instances are described by more than a few features. The actual WDBC data has 20 features and the corresponding RadViz looks as shown in Figure 1(b); the order of features is the same as in the data.

RadViz can be truly useful only when used with some methods for optimizing it. The features for Figure 1(a) were chosen using the algorithm VizRank developed by [Leban *et al.*, 2005], which exhaustively searches through all combinations of features within the specified parameters (usually we set the upper number of features to four or five) and evaluates the projection using a k nearest neighbors classifier. A projection is good if each instance is surrounded mostly by instances of its own class. To avoid overfitting, cross-validation is used instead of computing the quality of the graph directly. Since the number of combinations rises exponentially with the number of features, VizRank checks the projections ordered by the quality of the features they use, where the features are evaluated with a common measure such as ReliefF or information gain. Despite the huge number of combinations which can on microarray data easily reach 10^{20}, RadViz can most often find good projections within minutes of runtime.

By placing the anchors evenly around the circle and letting each pull in its own direction, Radviz assumes that the features are not correlated. Placing the anchors corresponding to strongly correlated features closer together would be potentially beneficial in conquering the noise and would, at the same time, offer a cleaner and more informative visualization. This idea is successfully exploited by McCarthy *et al.* [2004], but where a limitation with respect to RadViz is that visualization includes all available features.

The other limitation of RadViz is that it in principle assumes that all features are equally important. Since this is usually not the case, the quality of the projection is decreased since the pull of a less important feature(s) is as strong as those of the important ones.

The visualization we propose, FreeViz, overcomes both limitations by allowing the anchors to be placed anywhere in the circle. The correlated features can thus be placed together and the less important features can be put nearer to the circle's center to lower their impact. Even more than with RadViz, the usefulness of FreeViz depends upon the methods for optimizing it.

3 Formal Description and Optimization

Let $A^i = [A^i_x, A^i_y]$ be the i-th anchor, and $\mathbf{A}$ be a matrix of anchors. Each instance is described by a vector of feature values, $\mathbf{e} = [e^1, e^2, \ldots, e^n]$. The position of instance e in the circle is computed as $e_x = \sum_i e^i A^i_x$, $e_y = \sum_i e^i A^i_y$ or, in matrix notation, $\mathbf{e}' = \mathbf{eA}$. $\mathbf{A}$ thus represents a linear transformation that projects from the original feature space to a two-dimensional FreeViz.

Instead of using k-nearest neighbours, as VizRank does, we will optimize the projection by minimizing its potential energy, vaguely following the real-world physics of gravitational/electric fields [Halliday and Resnick, 1978]. Let $\mathbf{F}_{f \to e}$ be the force acting on instance e due to instance f. The force will depend on the distance between the two instances, their charges (weights of instances) and the type of their charges (instances' class – instances of the same class will attract and instances of different classes

will repel each other). When a particle e is moved by $\mathbf{de}'$ the work and the change of the potential energy equals $dE = A = -\mathbf{F}_{f \to e}\mathbf{de}'$.

In a system of multiple particles, the force acting on a particle equals the sum of forces exerted by all other particles,

$$\mathbf{F}_e = \sum_{f \neq e} \mathbf{F}_{f \to e}$$

and the change of potential energy when moving e is $dE_e = \mathbf{F}_e\mathbf{de}'$. When multiple particles are moved at once (as they will be in our case), the change of energy equals the sum of changes,

$$dE = -\sum_e \mathbf{F}_e \, \mathbf{de}'$$

We shall use the gradient method to optimize the system, *i.e.* to minimize its potential energy by moving the anchors. For this, we need to compute the gradient of the energy as a function of the anchors' position. Consider that $\mathbf{e}' = \mathbf{eA}$ and so $\mathbf{e}' = \mathbf{e}\,\mathbf{dA}$. When anchors are moved, the change in energy equals

$$dE = -\sum_e \mathbf{F}_e \, (\mathbf{e}\,\mathbf{dA})$$

For moving the x-coordinate of the i-th anchor, the related change in energy is $dE = \sum_e \mathbf{F}_{e,x} e^i dA^i_x$, where $F_{e,x}$ is the x-component of the force F_e, therefore

$$\frac{dE}{dA^i_x} = -\sum_e \mathbf{F}_{e,x} e^i$$

The computation of the y-coordinate is analogous. The formula is consistent with our intuition and with the nature which (at least on grand scale) minimizes the potential energy by accelerating the objects in the direction opposite to the energy gradient (that is, in the direction of the force). Instances are attracted or repelled from each other, but since they are held in place by the anchors, the forces between them are transmitted to the anchors. The force acting on each particle is distributed between the anchors proportionally with the values of corresponding features, e^i.

The formula is independent of the definition of the force. Its sign should depend upon whether the two instances are from the same class or not, so the force is attractive in the former and repulsive in the latter case. If instances are weighted, the force should rise linearly with the instance's weight. As for the distance, in our three dimensional space the usual large scale forces decrease by the inverse-square law, $F \sim 1/r^2$. In the two-dimensional world of FreeViz, the density of the field lines decreases linearly with the distance, so the force should be proportional to $1/r$. On the other hand, we can borrow the idea of Gaussian kernels from the statistics and let the force be proportional to e^{-r^2}. After some testing we found that the inverse-square law works best, while with linear or Gaussian kernels the force decrease with distance seems too slow.

A more important consideration regarding the force is whether it needs to decrease or increase with the distance. When separating instances of different classes, we are most concerned with those that are close together, while we do not need to push the groups that are already well separated even further apart. The repulsive force must therefore fall with the distance. On the other hand, the attractive force would try to squeeze the well-defined groups of instances from the same class into a point, and this unneeded effect would rise as the instances come closer together. For a contrast, if an instance is far from other instances of its own class and surrounded by instances of another, the former will not attract it, due to a large distance, while the latter will push it around and, in the best case, throw it out in a random direction. The attractive force should therefore increase with the distance.

```
Input:  number of instances N
        number of features A
        instance projections P
        a table of instances E
        classes of instances C
Output: a vector of gradients G

initialize F to 0
for e := 1 to N
    for f := e+1 to N
        dx := P[e].x - P[f].x
        dy := P[e].y - P[f].y
        r := sqrt(sqr(dx) + sqr(dy))

        if C(e) = C(f)
            then F_ef := -r^2
            else F_ef := 1/r^2

        Fefx := F_ef * dx/r
        F[e].x += Fefx
        F[f].x -= Fefx

        Fefy := F_ef * dy/r
        F[e].y += Fefy
        F[f].y -= Fefy

initialize G to 0
for e := 1 to N
    for i := 1 to A
        G[e].x += F[e].x * E[e][i]
        G[e].y += F[e].y * E[e][i]
```

Figure 2: Computation of gradients for FreeViz optimization

In a sense, the repulsive forces act like the electromagnetic or gravitational forces which decrease by the distance, while the attractive forces resemble the strong force that binds quarks and which increases by the distance, like a rubber band.

In the algorithm for computation of gradients (Figure 2) we make use of the action-reaction symmetry: the force between each pair of instances is computed only once and added to the sum of forces for both instances, but with different directions ($F_{f \to e} = -F_{e \to f}$). The force (`F_ef`) is separated into its x and y components (`Fefx` and `Fefy`) by multiplying it by projections to x and y axis, dx/r and dy/r, respectively.

The algorithm is rather simple and relatively fast: its

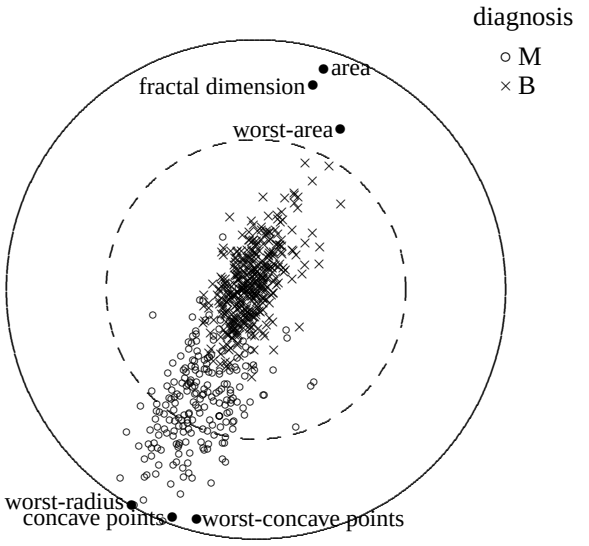


Figure 3: FreeViz for Wisconsin Breast Cancer data

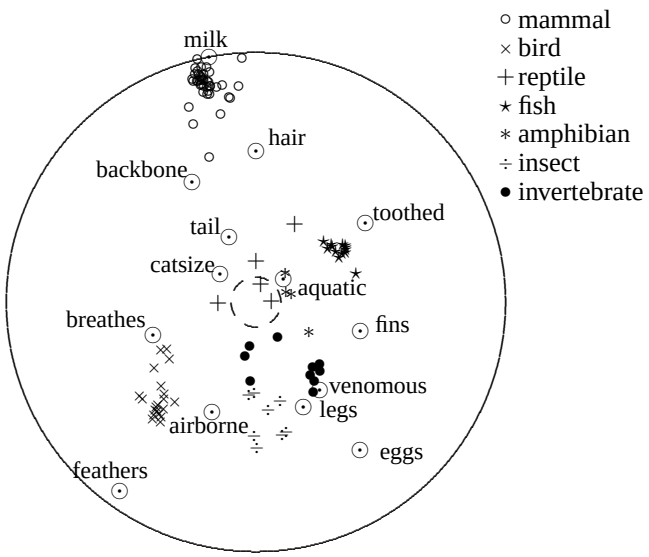


Figure 4: FreeViz for zoology database

time complexity is $O(N^2 + NA)$, where N is the number of instances and A the number of features; the first term comes from computation of forces between particles and the second from the loop that distributes the forces acting on each instance between the anchors. Although the operations performed by the algorithms are rather elementary, the squared number of instances suggests that the algorithm may be less useful when the number of instances is large.

The computed gradients can be used in optimization with the ordinary gradient method; at each step, the gradient vector is subtracted from the vector of anchors, the anchors are centered and renormalized (the farthest anchor should lie on the unit circle), and the projections are recomputed. The procedure is repeated until there is no considerable decrease (*e.g. 1 %*) of the potential energy for few consecutive steps.

Gradient method of optimization could be replaced with more advanced methods, but we found it fit for our purpose: it is fast and does not seem to stop in local minima.

Figure 3 shows a FreeViz for WDBC optimized by the proposed algorithm. For a clearer picture, we did not plot the features whose anchors are less than $0.5r$ from the center (marked with a dashed circle). The "area", "fractal-dimension" and "worst-area" listed in order of importance, seem to be correlated evidences for benignity, while the other three features speak for malignity of the tumor.

An important note about the algorithm is that it should not be used when the number of features exceeds the number of instances. Formally, if **E** is a matrix of instances and its rank equals the number of instances, the system $\mathbf{EA} = \mathbf{P}$ can be solved for any matrix of instances' positions **P**. In other words, if we have more features than instances, there exists a matrix of anchor positions for any prescribed positions of instances. The described algorithm is in this case able to overfit the data, resulting in meaningless projections.

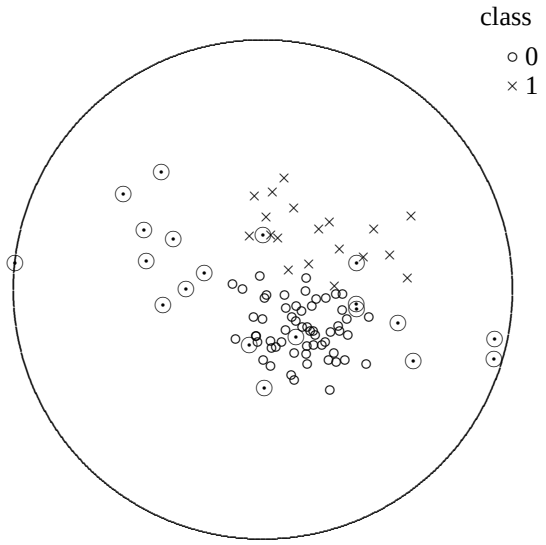
4 Case Studies and Discussion

We start with an example on a zoology data set which contains 101 animals described by their properties (lay eggs, breath, have hair...) and classified into seven groups (mammals, birds, reptiles...). As Figure 4 shows, the animals can be separated using the FreeViz projection and the corresponding positions of features make sense. For instance, mammals (○) have hair, backbone, and as the most important feature, milk. Being airborn is typical of birds (×) and insects (÷); the former are distinguished by feathers, and the latter have more legs (this feature can have values 0, 2 and 4). Amphibians are put between fish and reptiles.

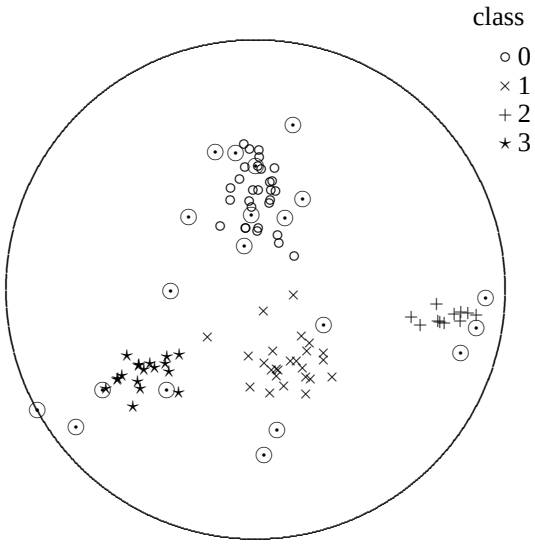
To test the visualization on a more complex data, we have tried FreeViz on several microarray cancer data sets. The resulting visualizations are shown in Figure 5. The feature names are intentionally uninformative (paper focuses on the study of class-separability, and while biomedical interpretation would be useful, it is beyond the scope of our reported study) and we have hidden them for the sake of clarity. The legend is omitted for the same reasons. To limit the number of features well below the number of instances, we have used ReliefF [Kononenko *et al.*, 1997] to select 20 most important genes for each data set (except for Lung cancer which has somewhat larger number of instances, where we have chosen a subset of 40 features).

Figure 5(a) shows the visualization of the data set that studies the outcome for the diffuse large B-cell lymphoma (DLBCL) [Shipp *et al.*, 2002], where the selected 20 features are well able to separate between the two classes. In another example, the data on four types of tumors in childhood (SRBCT) [Wang *et al.*, 2003], see Figure 5(b), the optimization yielded an even clearer separation.

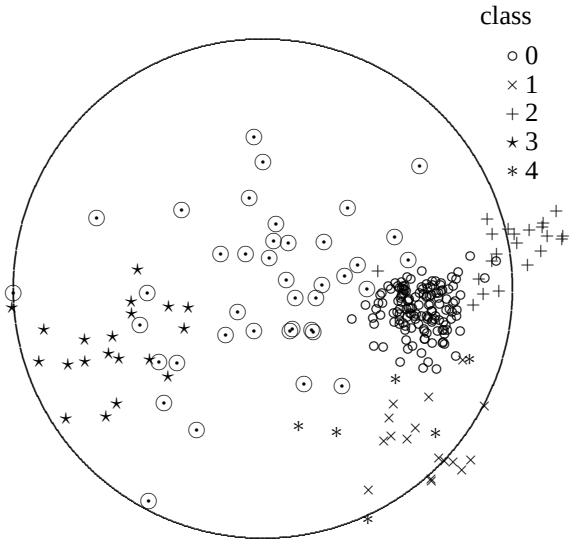
The largest data set we tackled is that on a lung cancer [Bhattacharjee *et al.*, 2001] with 203 instances, 12600 genes and five classes (Figure 5(c)). The separation is generally good, except for the class $*$, which is apparently too small, so the total force that its instances exert on anchors



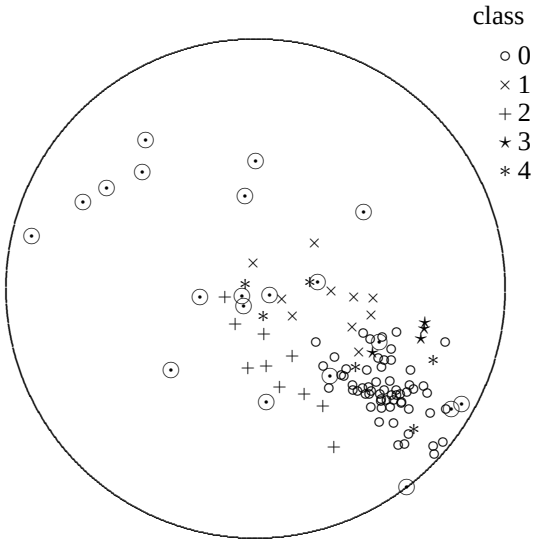
(a) DLBCL: 77 instances, 20 (out of 5469) genes



(b) SRBCT: 82 instances, 20 (out of 2308) genes



(c) Lung cancer: 203 instances, 40 (out of 12600) genes



(d) Brain tumor: 90 instances, 20 (out of 5920) genes

Figure 5: FreeViz on cancer microarray data

is incomparable to the forces by instances of the larger classes. In such cases, the algorithm could be augmented by adjusting the strength of forces according to the size of classes.

For the brain tumor data with 5920 genes and 90 instances (Figure 5(d)), separation was somewhat worse. Again, the instances belonging to the smaller classes ($\star$ and $*$) are lost between those of the large classes.

In all cases, running ReliefF took up to half a minute, while FreeViz optimization took a few seconds on a mediocre PC (Pentium IV, 1800 MHz).

5 Conclusion and Future Work

The paper presents a new method for intelligent visualization of class-labelled, multi-dimensional data sets. We have presented its utility on a number of biomedical data sets. Results of these preliminary studies are very encouraging: FreeViz is very fast and in all presented cases found visualizations of high quality with clear class separation.

There are many ideas that we have on how FreeViz can be exploited further. Some most important include:

- Visualization of probabilities. By computing the potential fields for a grid of points in the circle, it is possible to color the inside of the circle so that the color corresponds to the most probable class for an instance projected to that point and the color's saturation to the probability. We have implemented this functionality, but presenting it in the proceedings would require a color print, so we show it only on supplemental web pages (www.ailab.si/supp/freeviz-idamap).
- Classification. FreeViz visualization can be employed in classification of new cases. The simplest method, for instance, to produce a classifier from those pictures is to project the instance which is to be classified into the FreeViz space and observe its k nearest neighbors. Our experiments (not published here) with this are very encouraging and show that obtained classification accuracy, AUC and Brier scores are in the same range as those from logistic regression, naive Bayesian classifier and SVM.
- Misclassification costs. With the current implementation of the algorithm, the strength of repulsive forces depends upon the distance between the instances but not on their classes. By modifying it so that different combinations of classes would repel with different strengths, misclassification costs could easly be incorporated within analysis.

FreeViz is a available as a part of RadViz visualization widget in open-source data mining suite Orange (www.ailab.si/orange, [Demšar and Zupan, 2004; Zupan *et al.*, 2004]. As such it also offers other functionality, such as manual placement of anchors, selection of subsets of examples and similar, which is not described in this paper. See also supplemental web page (www.ailab.si/supp/freeviz-idamap) for additional material and figures from the paper in color.

References

[Bhattacharjee *et al.*, 2001] A. Bhattacharjee, W. G. Richards, and J. Staunton *et al*. Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma sub-classes. *Proc. Natl. Acad. Sci. USA*, 98 (24), 2001.

[Blake and Merz, 1998] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[Brunsdon *et al.*, 1998] C. Brunsdon, A. S. Fotheringham, and M. E. Charlton. An investigation of methods for visualising highly multivariate datasets. In *Case Studies of Visualization in the Social Sciences*, pages 55–80. Joint Information Systems Committee / ESRC, 1998.

[Demšar and Zupan, 2004] J. Demšar and B. Zupan. *Orange: From Experimental Machine Learning to Interactive Data Mining, A White Paper*. Faculty of Computer and Information Science, Ljubljana, Slovenia, 2004.

[Halliday and Resnick, 1978] D. Halliday and R. Resnick. *Physics*. John Wiley and Sons, New York, 3rd edition, 1978.

[Huber, 1985] P. J. Huber. Projection pursuit. *The Annals of Statistics*, 13(2):435–474, 1985.

[Keim, 2002] D. A. Keim. Information visualization and visual data mining. *Transactions on Visualization and Computer Graphics*, 6(1):59–78, 2002.

[Kononenko *et al.*, 1997] I. Kononenko, E. Šimec, and M. Robnik Šikonja. Overcoming the myopia of inductive learning algorithms with ReliefF. *Applied Intelligence Journal*, 7(1):39–56, 1997.

[Leban *et al.*, 2005] G. Leban, I. Bratko, U. Petrovic, T. Curk, and B. Zupan. Vizrank: finding informative data projections in functional genomics by machine learning. *Bioinformatics*, 21(3):413–414, 2005.

[McCarthy *et al.*, 2004] J. F. McCarthy, K. A. Marx, P. E. Hoffman, A. G. Gee, P. O'Neil, M L. Ujwal, and J. Hotchkiss. Applications of machine learning and high-dimensional visualization in cancer detection, diagnosis, and management. *Annals of New York Academy of Sciences*, 1020:239–262, 2004.

[Shipp *et al.*, 2002] M. A. Shipp, K. N. Ross, and P. Tamayo *et al*. Diffuse large b-cell lymphoma outcome prediction by gene expression profiling and supervised machine learning. *Nature Medicine*, 8:68–74, 2002.

[Wang *et al.*, 2003] Zuyi Wang, Yue Wang, and Jianping Lu *et al*. Discriminatory mining of gene expression microarray data. *J. VLSI Signal Process. Syst.*, 35(3):255–272, 2003.

[Zupan *et al.*, 2004] B. Zupan, G. Leban, and J. Demšar. *Orange: Widgets and visual programming, A White Paper*. Faculty of Computer and Information Science, Ljubljana, Slovenia, 2004.