
Data Mining and Knowledge Discovery, 13, 119–136, 2006

DOI: 10.1007/s10618-005-0031-5

VizRank: Data Visualization Guided by Machine Learning

GREGOR LEBAN gregor.leban@fri.uni-lj.si
Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, Ljubljana, Slovenia

BLAŽ ZUPAN blaz.zupan@fri.uni-lj.si
Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, Ljubljana, Slovenia; Department of Molecular and Human Genetics, Baylor College of Medicine, Houston, TX

GAJ VIDMAR gaj.vidmar@mf.uni-lj.si
Institute of Biomedical Informatics, University of Ljubljana, Vrazov trg 2, Ljubljana, Slovenia

IVAN BRATKO ivan.bratko@fri.uni-lj.si
Faculty of Computer and Information Science, University of Ljubljana, Tržaška 25, Ljubljana, Slovenia; Jozef Stefan Institute, Ljubljana, Slovenia

Received April 30, 2004; Accepted November 7, 2005

Published online: 16 May 2006

Abstract. Data visualization plays a crucial role in identifying interesting patterns in exploratory data analysis. Its use is, however, made difficult by the large number of possible data projections showing different attribute subsets that must be evaluated by the data analyst. In this paper, we introduce a method called VizRank, which is applied on classified data to automatically select the most useful data projections. VizRank can be used with any visualization method that maps attribute values to points in a two-dimensional visualization space. It assesses possible data projections and ranks them by their ability to visually discriminate between classes. The quality of class separation is estimated by computing the predictive accuracy of k-nearest neighbor classifier on the data set consisting of x and y positions of the projected data points and their class information. The paper introduces the method and presents experimental results which show that VizRank's ranking of projections highly agrees with subjective rankings by data analysts. The practical use of VizRank is also demonstrated by an application in the field of functional genomics.

Keywords: data visualization, data mining, visual data mining, machine learning, exploratory data analysis

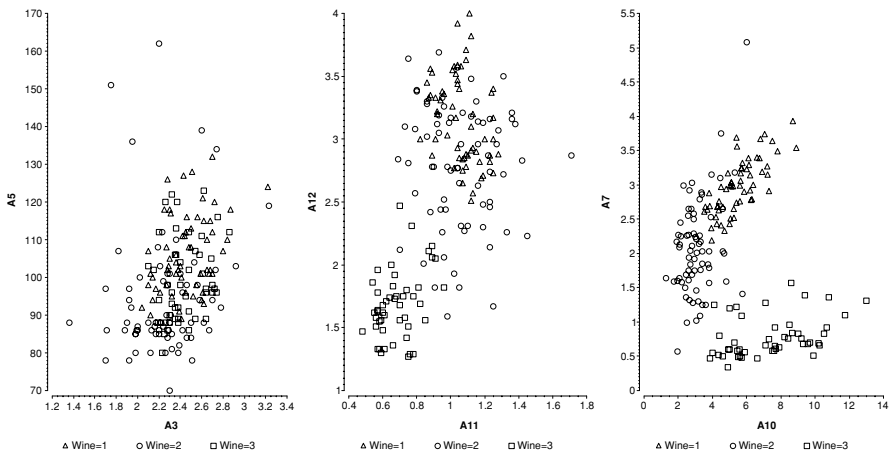
1. Introduction

Data visualization is an essential tool in data analysis since it enables us to visually detect complex structures and patterns in the data. In the words of Cleveland (1993): "Visualization is critical to data analysis. It provides a front line of attack, revealing intricate structure in data that cannot be absorbed in any other way." However, not all data projections provide equal degree of insight and the task of the data analyst in case of exploratory data analysis is to find most informative data projections. Of course, the patterns that we are looking for depend on the type of the data. In the case of unclassified data, interesting projections are those that reveal data structures such as trends, outliers or clusters of points. Projections of classified data, on the other hand, are informative

when they enable us to spot, i.e. visually induce, a rule that is successful in separating different classes. Besides finding important attributes in the data set, such displays also help building an understanding of the class structure.

Because real-life data sets contain several attributes, finding interesting projections can be a difficult and time-consuming task for the analyst, since the number of possible projections increases exponentially with the number of concurrently visualized attributes. For example, when analysing a data set with a simple scatterplot, the number of different scatterplots one has to inspect is $m(m-1)/2$, where m is the number of attributes in the data set.

For this reason, we have developed a method called VizRank, which is able to automatically rank visual projections of classified data by their success in showing different class values well separated. VizRank can be applied in combination with any visualization method that maps attribute values to the position of a plotted symbol. A new data set is constructed from the projection, containing the class value and just two attributes: x and y positions of data points. Projection usefulness is estimated by inducing a classifier—we opted for the k-nearest neighbor (k-NN) classifier—and evaluating its accuracy on this data set. This way, each projection is numerically evaluated with a value between 0 and 100. Projections which provide perfect class separation (there is no overlap between classes) receive value 100, while less informative projections receive correspondingly lower values. Figure 1 illustrates three scatterplot visualizations of the *wine* data set from the UCI repository (Blake and Merz, 1998). The data set comprises 13 continuous attributes which describe the results of chemical analysis used to determine the type of wine. The three chosen projections are ordered from the least informative (left) to the most informative (right). The values returned by our method were 47.76, 67.92 and 91.40, respectively. They were obtained using k-NN classifier with $k = 15$ (for the choice of k we always chose $\sqrt{N}$, where N is the number of examples in the data



(a) 47.76 (b) 67.92 (c) 91.40

Figure 1. Three scatterplots of the *wine* data set ($N = 178$, three types of wine), sorted from the least (*left*) to the most informative (*right*). Bellow each diagram, the score of the projection usefulness computed by the VizRank method using k-NN classifier is reported.

set, as proposed by Dasarathy (1991)). This example shows how VizRank's evaluations correspond to the human intuition about the usefulness of the selected projections.

In the rest of the paper, we first relate our work to other approaches to finding interesting-data projections. We then describe two visualization methods that we used in combination with VizRank to test it. The details of the VizRank method are presented next, followed by experimental results, where we investigate how well the rankings generated by VizRank match the rankings of human data analysts. We also present a case study where we have applied VizRank to the data set from functional genomics. We conclude with a summary.

2. Related work

There are several approaches that enable us to view multidimensional data in the most interesting way. Since we are focused on projections of classified data, we will not discuss the wide array of well-known statistical dimension reduction methods, such as principal component analysis or factor analysis, which are primarily used to analyze the attribute space as such.

Fisher's linear discriminant, better known as linear discriminant analysis (LDA), is a classification method that can be thought of as projecting high-dimensional data onto a line and performing classification in this one-dimensional space. The projection is chosen so that it maximizes the distance between the means of the two classes while minimizing the variance within each class. For c-class problem, the generalization of this method involves $c-1$ discriminant functions and it is called multiple discriminant analysis. These methods are primarily used for classification purposes, but have also been used for data visualization. Dillon et al. (1998) created *class-preserving projections* by using discriminant functions as axes for projecting data in a scatterplot. Since discriminant functions are optimized for class separation, such projections can show different classes well separated. The visualization of LDA was further elaborated by Cook and Yin (2001). The main advantage of this method compared to VizRank is that there exists a closed form solution for efficient computation of discriminant functions. Its drawback is that LDA is optimal only for data where each class has unimodal Gaussian density with well-separated means. LDA also creates only $c-1$ new features which can in most cases produce only a few scatterplots. An additional problem is the ability to gain insight from such projections. Each axis is a linear combination of original attributes, which makes projection interpretation relatively demanding. For extensions of LDA regarding distributional assumptions, see e.g. Kaski and Peltonen (2003) and Torkkola (2003).

Projection pursuit (Friedman and Tukey, 1974; Huber, 1985; Nason, 1992) is an algorithm for creating scatterplot projections of high dimensional data where each axis is a linear combination of attributes. Although it is mainly used in unsupervised learning to search for clusters of points, it does, like VizRank, numerically evaluate projections. The criterion of "interestingness" is different, though. Diaconis and Friedman (1984) gave theorems which show that under suitable conditions, most projections show approximately Gaussian distribution of data points. This suggests that it is useful to search for projections where the data is most non-normally distributed. Projection pursuit evaluates projections using a criterion function called projection pursuit index. Most of projection

pursuit indices are based on entropy measures and are derivable so that gradient based approach can be used to find more interesting projections.

3. Visualization methods

There is a huge number of techniques that one can use to visualize multidimensional data. According to Keim and Kriegel (1996), we can classify them into five different groups: geometric, pixel-oriented, icon-based, hierarhical and graph based techniques.

VizRank method can be successfully applied with any geometric visualization method where examples are visualized as points in a two-dimensional space and the values of attributes only influence the position of the point and not its size, shape or color (symbol properties can, however, be used to represent class value). Examples of such methods are scatterplot, radviz, polyviz, and gridviz (Grinstein et al., 2001; Hoffman and Grinstein, 1999; Hoffman et al., 1997). An example of a geometric visualization method where VizRank can not be used is the parallel coordinates technique (Inselberg, 1981), where the visualized attributes are shown by parallel vertical axes and each m-dimensional example is not represented by a point, but by $m - 1$ connected line segments.

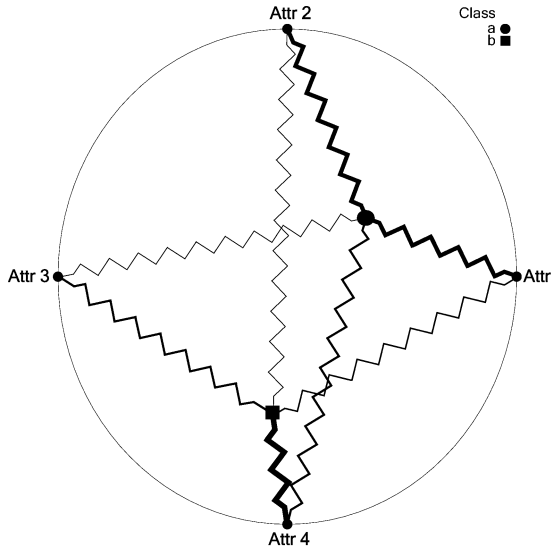
The two visualization methods with which we applied and evaluated the VizRank method are scatterplot and radviz. Scatterplot is the basic and a very popular visualization method and was selected because it projects two selected attributes in a very clear and comprehensible form. Radviz, in contrast to the scatterplot, is able to concurrently visualize a larger number of attributes, but the projection is more difficult to interpret.

3.1. Scatterplot

Scatterplot with all its variants (Harris, 1999) is one of the oldest and most utilized visualization methods. In its basic form, it depicts the relation between two continuous attributes. Attributes are represented with a pair of perpendicular coordinate axes. Each data example is shown as a point in the plane whose position is determined by the values of the two selected attributes. The number of visualized attributes can be increased by mapping them to color, size and shape of the visualized point (Cleveland and McGill, 1984). We must, however, be aware that when visualizing a larger data set, the points can substantially overlap and the additional attributes may not be successfully perceived. In our experiment, we used additional point information only to represent the class value. Examples of scatterplots are presented in Figure 1.

3.2. Radviz

Radviz (Hoffman et al., 1997), which stands for radial visualization, is a method where the examples are represented by points inside a circle. The visualized attributes correspond to points equidistantly distributed along the circumference of the circle. The easiest way to understand the method is by using a physical analogy with multiple springs (Figure 2). For visualizing each data example represented by m attributes, m springs are used, one spring for each attribute. One end of each spring is attached to the attribute's position on the circumference, and the other to the position of the data point inside the circle. The stiffness of each spring in terms of Hooke's law is determined by the



Attr1	Attr2	Attr3	Attr4	Class
0.75	0.75	0.1	0.3	a
0.2	0.1	0.3	0.95	b

Figure 2. Radviz plot with two data examples from the table. Springs are drawn to illustrate how positions of points are calculated. Stiffness of each spring corresponds to attribute value and is depicted with line thickness. Data point is drawn where the sum of all spring forces equals 0.

corresponding attribute value—the greater the attribute value, the greater the stiffness. The data point is then placed at the position where the sum of all spring forces equals 0. Prior to visualizing, the values of each attribute are usually standardized to the interval between 0 and 1 to make all the attributes equally important in "pulling" the data point. Some properties of the radviz method are:

- All the points that have approximately equal values of all the attributes after standardization, lie close to the center of the circle.
- Points that have approximately equal values at the attributes which lie on the opposite sides of the circle, will also lie close to the center.
- If one attribute value is much larger than the values of the other attributes, then the point will lie close to the point on the circumference of the circle which corresponds to this attribute.

The visualization of a given data set, and also its usefulness, largely depends on the selection of visualized attributes and their ordering around the circle perimeter. The total number of possible orderings of m attributes is $m!$, but some of them are equivalent, since they represent the same visualization. For each ordering of m attributes, there are $m - 1$ equivalent orderings that can be created by shifting the position of each attribute anchor for up to $m - 1$ times—each of these orderings will only represent a rotated version of the original visualization. For each ordering we can also find an equivalent

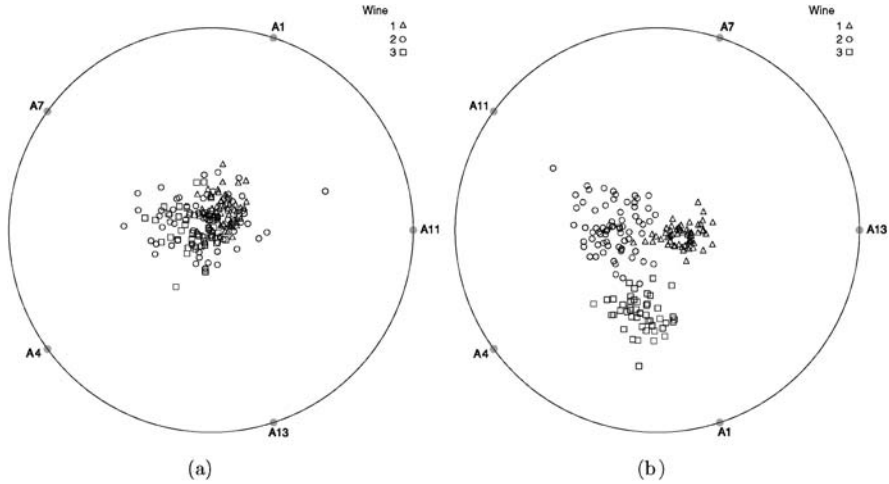


Figure 3. Two radviz plots of five attributes in different orders from the *wine* data set.

visualization if we reverse the order of the attributes. Therefore, it can be shown that the total number of different visualizations with m attributes is $(m - 1)!/2$.

Figure 3 shows two radviz visualizations of five attributes from the *wine* data set. The order of the attributes in Figure 3(a) makes a completely uninformative visualization, since it shows all classes overlapping. On the other hand, the order of the same attributes in Figure 3(b) creates a visualization which shows all three classes almost perfectly separated. From the position of the points in this figure, we are able to make some conclusions about the values of the shown attributes. For example, examples of class *Wine* = 3 lie close to the positions of attributes $A4$ and $A1$, which means that they have large value at attributes $A4$ and $A1$ and small value at attributes $A11$, $A7$ and $A13$. We can infer similar conclusions for the other two class values.

3.3. Visualization of discrete attributes

Although scatterplot and radviz method are primarily intended for visualizing continuous attributes, they can also be applied to discrete attributes. Since discrete attributes have only a limited number of possible values, many points can overlap and thus much information remains hidden. To avoid this, a simple and often used solution is altering the positions of points by a small random noise ("jitter") (Chambers et al., 1983), which improves the perception of data distribution. It is also very important to note that when visualizing nominal discrete attributes, any impression of order of attribute values is only a consequence of visualization and not a true property of the attribute.

4. Details of VizRank method

Given a data set and a visualization method, VizRank evaluates possible projections and provides the data analyst with an ordered list of currently evaluated projections along with the assessment of their "usefulness". The list is updated each time a new projection

is evaluated. This way, the data analyst is relieved of the unguided search through numerous possible projections, and can focus only on those top-ranked by VizRank that are likely to provide the best insight into the data. To construct a ranking of possible projections, the VizRank method computes numerical estimates of the usefulness of each data projection in the following two steps:

1. Generate the graphical representation of data using the selected set of attributes.
2. Automatically assess the "visual usefulness" of the representation.

As already mentioned, we chose two geometric visualization methods for step 1: scatterplot and radviz. Step 2 is the more difficult part of the VizRank method. The task here is to measure the potential value of a particular graphical data representation with respect to enabling a human to get an insight into the data. In other words, our aim is to measure how likely it is that a user will spot a visual pattern in the data that corresponds to a regularity in the domain.

The solution of VizRank is to apply a machine learning method to the graphically represented data, and estimate the accuracy (in the sense of quality, appropriateness, performance) of the induced classifier. The graphical representation of the data means that the available attributes for learning are only the x and y positions of the examples in the plot and their class value, i.e. the features that the user can see in the representation. The performance of the classifier (measured using some scoring function like classification accuracy or Brier score) is considered to be indicative of visual usefulness because it will be high if examples with different classes are well separated in the projection, and low otherwise. Since we can easily define projections with pure, well separated classes as more interesting and preferable to projections where classes overlap, it is reasonable to expect that the measure of classification performance gives us a plausible estimate of usefulness of a given projection.

The choice of the learning algorithm is a key element in evaluating a projection from the visualization point of view. The result of learning can be seen as a decision boundary between examples that belong to different classes. Although there are many classification algorithms available, they are not equally suitable for our purpose. The reason for that is that each of them has a different bias in forming decision boundaries, and different boundary shapes are not equally suitable for visualization. For example, in learning decision trees, decision boundaries can only be straight horizontal and vertical lines that partition the whole projection into a structure of rectangles. Although decision trees often perform well as classifiers, we can easily imagine projections with well separated classes where decision trees would fail to discover visually simple decision boundaries due to the restriction in boundary shape.

Since we want to measure the potential usefulness of a visualization for human to gain insight, we want to select a learning method that will produce "visually obvious" classification rules. Such rules are patterns that we can expect the user would be able to spot in the graphical representation. One such learning method that we chose to use in VizRank is the k-NN method where the attributes available to k-NN are only the x and y positions of the visualized examples.

To summarize, projection evaluation in VizRank method is based on the following. First, we project the high-dimensional data to a plane, using the selected visualization method. The projection can be viewed as a feature construction process—we can treat

the x and y position of the points as two new features. In the next step, VizRank creates a new data set with the two newly constructed features and the class information. This data set is then used by a machine learning method, whose task is to learn to predict the class value for the examples in the data set. The success of learning is then evaluated and is used as an estimate of usefulness of the projection—the prediction performance on the data set will be high if different classes are well separated and will be low if the classes overlap.

4.1. k-nearest neighbor method

k-NN is a machine learning method that predicts class value for an unlabeled example by analyzing its k neighboring examples. Each of the k neighbors votes for its class value and their vote is usually weighted according to the distance from the example. In our implementation, we weighted the votes using function e^{-t^2/s^2}, where t is the distance to the example and s is chosen so that the impact of the farthest of k examples is 0.001. The result of the voting is a probabilistic class prediction and the example can be labeled with the most probable class value.

To be able to define the neighborhood, we must first define a metric for measuring the distance between examples. Our implementation uses Euclidean metric, which has several desirable mathematical properties (like invariance to projection rotation). Despite of its limitations for emulating human judgement of dissimilarity (Santini and Jain, 1996, 1999), it is also a useful approximation for the human criterion (Cutting and Vishton, 1995). As for parameter k, the number of neighbors used in class prediction, we want to use a large value to obtain a reliable prediction, while at the same time we want to keep it small enough so that we only consider nearby examples in prediction. The method is less sensitive to the choice of k if the votes of the neighbors are weighted according to their distance, so that near neighbors have greater influence on the prediction than far neighbors. In any case, Dasarathy (1991) showed that we can use $k = \sqrt{N}$ as a generally useful rule of thumb, where N is the number of training examples (we followed this rule in all our experiments).

4.2. Evaluating the usefulness of a projection

There are many scoring functions measure the performance of a classifier. A measure that is often used in machine learning is classification accuracy. It is defined as the proportion of cases when the classifier correctly predicted the class value. However, classification accuracy has a 0/1 loss function and is therefore not very sensitive in the case of probabilistic classification: for classification accuracy, it does not matter if the classifier predicted the correct class with the probability of 1 or with the probability of, say, 0.51. Since in our case the main goal is not to evaluate the classifier but to evaluate the projection, it makes more sense to work with the predicted probabilities. For a probabilistic classifier, such as k-NN, a more useful measure is the average probability $\bar{P}$ that the classifier assigns to the correct class value:

$$\bar{P} = E(P_f(\mathrm{y} \mid x)) = \frac{1}{N} \sum_{i=1}^{N} P_f(y_i \mid x_i) \tag{1}$$

where N is the number of examples, y_i, is the correct class value for data example x_i and $P_f(y_i \mid x_i)$ is the probability assigned to the correct class value y_i, for example x_i by the classifier f. This measure was used in both experiments described in the next section. Another measure that would give even greater emphasis to the predicted probabilities is Brier score (Brier, 1950). Given two probability distributions for a given example, the predicted class probability distribution p', and the actual class probability distribution p, where the class can take c values, the Brier score of the prediction is:

$$b(p; p') = \frac{1}{c} \sum_{i=1}^{c} (p'_i - p_i)^2 \tag{2}$$

The larger the Brier score, the worse the prediction performance. In practical evaluation of a testing example, the actual class probability distribution p is assigned a probability of 1 to the true class and 0 to other classes. Brier score on a test data set is computed as in Eq. (2) for each test example, and then an average value across all examples is reported.

To estimate the scores for the classification models, we used the leave-one-out evaluation schema. The prediction of the k-NN model was tested on all examples, where for each test the example being classified was not included in the neighborhood used for prediction of the class.

To compare how the choice of a scoring function influences the ranking of projections in practice, we performed an experiment. We evaluated all 3081 scatterplot projections of the *yeast* data set (see Section 5.2) using classification accuracy, average probability assigned to the correct class ($\bar{P}$) and Brier score. Correlations between different measures are reported in Table 1 and are all statistically significant at $p < 0.01$. The largest discrepancies were observed between classification accuracy and $\bar{P}$, while there was almost perfect agreement in ranking between $\bar{P}$ and Brier score. As expected, the biggest difference in ranking was for projections that have a large number of critical examples that lie on the edge of the clusters. Such examples have lower probability of correct prediction, but still high enough so that they are not misclassified. Brier score and $\bar{P}$ measure are more conservative in such cases and take prediction uncertainty into account, thus lowering classification success. This is illustrated in Figure 4, which shows a scatterplot projection of an artificial data set with 100 examples described with two continuous attributes and a binary class attribute. Examples in each class lie within elongated clusters that are very close to each other, but they are perfectly separable using $y = x$ as the discriminant function. Classification accuracy of k-NN classifier with $k = 10\,(= \sqrt{100})$ evaluated on this projection is 100.0%, while $\bar{P}$ is only 75.1%. Since k-NN predictions on this data set are highly uncertain (due to the distance between clusters) we find the value, returned by $\bar{P}$ score function, as a more reasonable estimate of the projection usefulness.

We can conclude that classification accuracy is the least appropriate measure for evaluating projections because it discards valuable information about prediction (un)certainty. Since $\bar{P}$ and Brier score are highly correlated it does not matter which one we choose. Nevertheless, we advise using $\bar{P}$, because its predicted value of projection usefulness is easier to interpret. This is also the measure that we used to compute the projection scores in all our experiments.

Table 1. Correlations between different scoring functions for evaluation of classifiers on *yeast* data set. All correlations are significant at the 0.01 level

	$\bar{P}$[b]	Brier score
CA[a]	0.857	−0.872
$\bar{P}$		−0.961

[a]Classification accuracy.
[b]Average probability assigned to the correct class.

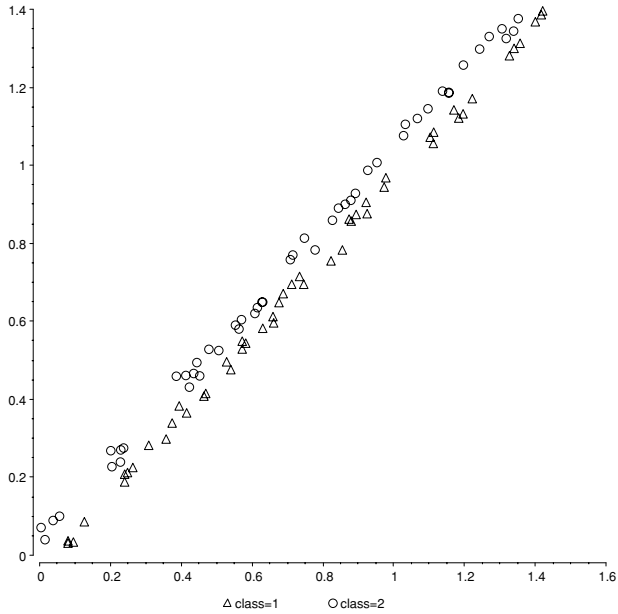


Figure 4. Scatterplot projection of an artificial data set. Purpose of this figure is to illustrate the difference in projection evaluation when using different measures of classification success. Accuracy of k-NN classifier on this projection evaluated using classification accuracy is 100.0%, using average probability assigned to the correct class ($\bar{P}$) it is 75.1% while the Brier score is 0.163

4.3. Computational complexity and a search heuristic

To evaluate a projection we used an implementation of a k-NN algorithm, which uses $O(N^2)$ time, where N is the number of examples. Despite its high complexity we found this implementation of k-NN sufficient for our experiments. For example, automatic generation and ranking of 2300 radviz projections of a data set with 200 examples was computed in 2 minutes on a Pentium 4 PC with 2.4 GHz processor. Analysis of large data sets would nevertheless have to use a more efficient implementation of k-NN. A popular implementation that could be used is the *k-D* tree nearest neighbor algorithm by Friedman et al. (1977) which would evaluate a projection in $O(N \log(N))$ time.

Computational time used to evaluate a projection is very important, since number of different projections may be high. In the case of the radviz method displaying l

attributes of the total m, the l attributes can be selected in $\binom{m}{l}$ different ways and each selection of l attributes can produce $(l-1)!/2$ different radviz projections. Since the number of projections increases exponentially with l, we are often limited to evaluate only projections with small number of attributes. Our experience with the radviz method shows that for classified data this is not a serious drawback, since projections with a large number of attributes (>8) are very difficult to interpret.

Despite using a fast k-NN algorithm and evaluating only projections with a small number of attributes, we are still unable to evaluate all possible projections when analyzing very high-dimensional data sets. For this reason we developed a simple and fast search heuristic that enables VizRank to find top-ranked projections by evaluating only a small subset of possible projections. Our heuristic first numerically evaluates the "quality" of each attribute in the data set using the ReliefF measure (Kononenko and Simec, 1995). Other measures of attribute quality, like Gini index or Gain ratio, could also be used, but we selected ReliefF since it can equally handle discrete and continuous attributes. The heuristic then estimates the usefulness of each projection as the sum of ReliefF values for the attributes that participate in the projection. Possible projections can be ranked by this estimate and VizRank can use this ranking to determine the order in which it will assess the projections. The rationale behind using the quality of attributes as a heuristic for faster identification of interesting projections is that attributes that are by themselves better at class separation are more likely to produce informative projections than attributes that are worse.

To evaluate this heuristic, we performed an experiment using the *yeast* data set used in Section 5.2. We first assessed all scatterplot projections and radviz projections with 3 attributes using VizRank. Then we computed how many projections as ranked by the heuristic we have to evaluate to find the actual best 10, 20 and 50 projections as ranked by VizRank. The results are shown in Figure 5(a) for scatterplot and Figure 5(b) for radviz, respectively. The vertical axis represents the percentage of best projections

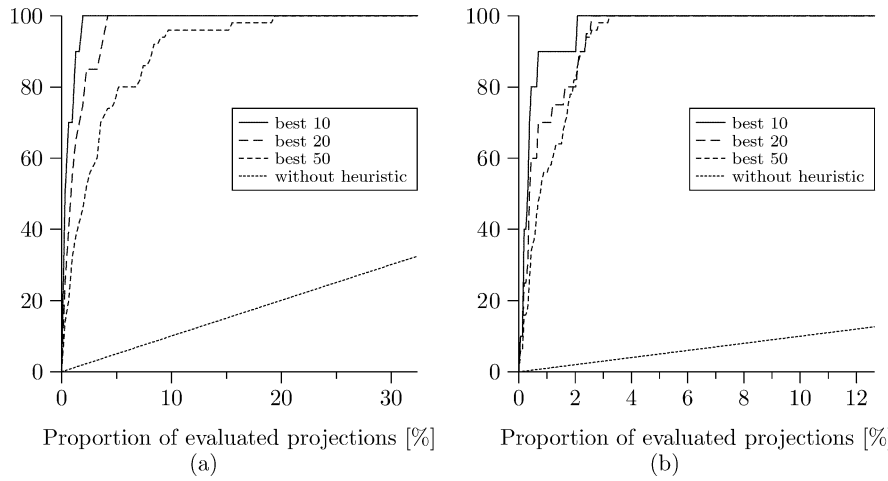


Figure 5. Lift curves for projection search heuristic for scatterplot (a) and radviz (b) method. While x axis represents the percentage of evaluated projections, y axis shows the percentage of the actual best 10, 20 and 50 projections that have been found.

found, while horizontal axis represents the percentage of assessed projections. Since the curves are very steep we shortened the original size of the x axis to one third for scatterplot and to one eighth for radviz. To illustrate the effectiveness of the heuristic we have also drawn a curve that represents the search progress if no heuristic is used. It is evident that the heuristic is very successful. To find the best 50 projections we have to assess about 20% of all scatterplot projections (600 projection) and only about 3% of possible radviz projections (2500 projections). Based on these results and on the results on several other data sets not reported in this paper, we can conclude that the proposed heuristic can lead to a significant reduction of the number of projections VizRank needs to consider to find the most interesting ones.

5. Experimental analysis

In this section, we report the results from two experiments that study the usefulness of the VizRank method. First, we present results from a psychological experiment testing the agreement between ranking of projections done by VizRank and rankings done by human subjects. The second experiment is a case study on a data set from functional genomics, where VizRank was used to find data projections that may reveal interesting biological phenomena.

5.1. Psychological experiment

We performed an experiment in which we evaluated how the ranking of projections proposed by VizRank agrees with rankings assigned by analysts. For this purpose, we chose six data sets; five from the UCI repository (*housing, imports, credit, voting* and *wine*) and one medical data set from our research practice (*circlet*—data set describing upper limb motion using haptic interface (Bardorfer et al., 2001)). All data sets had at least 5 attributes (discrete and/or continuous) and a class attribute with no more than 5 possible values. Two of the selected data sets had a continuous class but we categorized it into a binary class with an approximately uniform distribution. The reason for the limit on the number of class values is that for actual visualization, each value must be represented with a different color. If a greater number of colors were used, some of the colors might not be distinguishable enough, which could influence the subjects' ranking.

Selected data sets were visualized using scatterplot and radviz method. We evaluated scatterplots for all different pairs of attributes and all radviz projections with up to 5 visualized attributes. Projections were generated and evaluated automatically, without any additional human intervention. For each data set, the list of possible projections was computed and sorted by their usefulness. From this list we then chose 6 projections: the highest and the lowest ranked projection and four intermediate projections so that numerical estimates of usefulness were approximately equidistant. The six chosen projections for all data sets were color printed on A5 sized paper.

Twelve post-graduate students of computer science and cognitive psychology were involved in the experiment. They were selected since they were knowledgeable in both machine learning and fundamentals of perception. Each of them had twelve

tasks; six with scatterplots and six with radviz plots. Each task consisted of sorting six projections by their usefulness. Since initial order of projections could influence the final order as determined by the raters, we used incomplete Latin square experimental design, so that each rater had a different initial projection order and different order of data sets. The order of visualization methods was also varied; half of the raters first rated the scatterplots and the other half first rated the radviz plots. No time limits were imposed. Typical ranking time for all twelve tasks was 30 minutes.

For each visualization method, average ranks across raters were calculated for each projection for each data set. Agreement among raters for a given data set and a given visualization method was assessed using Kendall's coefficient of concordance W (Siegel and Castellan, 1988). Agreement of the group of raters with the theoretical ranking was assessed with the coefficient of correlation of a group of judges with a criterion (T_c, i.e., average Kendall's τ; (Siegel and Castellan, 1988)). To compare concordance between groups, the seldom used but long known L statistic was used (Schucany and Frawley, 1973).

Merely the average ranks (Tables 2 and 3) clearly demonstrate that the expert raters were almost in perfect agreement with the theoretical ranking of projections. Further support for such observation comes from the fact that for each data set, as well as for the pooled data, there is statistically significant agreement within and between visualization methods regarding the ranking of projections ($p < 0,001$ for all L-tests). Since there is also statistically significant agreement within and between the raters ranking scatterplots first and the raters ranking the radviz plots first ($p < 0,001$ for all L-tests), considering either each data set separately or the pooled data, we can conclude that the order of presentation did not influence the rankings.

5.2. *A case study on a problem from functional genomics*

To further evaluate VizRank, we have used a data set from functional genomics and analyzed annotated gene expression data set on budding yeast *Saccharomyces cerevisiae*. We used the data from 79 different DNA microarray hybridization

Table 2. Mean data analysts' ranks of selected scatterplot projections

		VizRank ranking				
Data set	(best) #1	#2	#3	#4	#5	(worst) #6
Wine	1.07	1.93	3.00	4.00	5.14	5.86
Voting	1.00	2.21	3.14	3.64	5.00	6.00
Imports	1.00	2.36	3.14	3.50	5.00	6.00
Housing	1.07	2.29	3.00	4.07	4.71	5.86
Credit	1.50	1.50	3.14	3.86	5.00	6.00
Circlet	1.17	2.25	2.58	4.00	5.08	5.92
Total	1.15	2.07	3.01	3.85	4.99	5.93

Table 3. Mean data analysts' ranks of selected radviz projections

Data set	(best) #1	VizRank ranking #2	#3	#4	#5	(worst) #6
Wine	1.00	2.00	3.29	3.71	5.00	6.00
Voting	1.00	2.43	3.07	3.64	4.86	6.00
Imports	1.00	2.43	2.86	4.07	4.64	6.00
Housing	1.00	2.79	3.29	3.43	4.50	6.00
Credit	1.21	1.86	3.36	3.57	5.29	5.71
Circlet	1.42	1.58	3.00	4.17	4.92	5.92
Total	1.11	2.17	3.19	3.69	4.90	5.93

Table 4. Measures of concordance (see Section 5.1 for explanation)

	Scatterplot		Radviz	
	W	T_c	W	T_c
Wine	0,975	0,936	0,984	0,967
Voting	0,950	0.956	0,925	0,889
Imports	0,934	0,856	0,896	0,856
Housing	0,834	0,856	0,796	0,789
Credit	0,956	0,900	0,909	0,878
Circlet	0,940	0.933	0,946	0,867

Note: All the W values are significantly different from zero at $p < 10^{-6}$.

measurements experiments from Eisen et al. (1998). The data were drawn from time courses during the following eight processes: cell division cycle after synchronization by alpha factor arrest (ALPH; 18 time points), centrifugal elutriation (ELU; 14 time points), temperature-sensitive cdc15 mutant (CDC15; 15 time points), sporulation (SPO, 7 time points plus four additional samples), shock by high temperature (HT, 6 time points), reducing agents (D, 4 time points), low temperature (C; 4 time points), diauxic shift (DX, 7 time points). Our analysis was inspired by the utility of this data in a study of various machine learning approaches by Brown et al. (2000), who used gene function annotation from Munich Information Center for Protein Sequences Yeast Genome Data Base (MYGD) (http://mips.gsf.de/proj/yeast). In particular, they were interesting for characterization of five functional classes, of which we have considered the three which were represented with the highest number of genes (cytoplasmic ribosomes, 121 genes; proteasome, 35 genes; respiration, 30 genes). We then used VizRank to see how these three groups of genes can be differentiated based on their expression data.

With 79 attributes, there are 3081 different scatterplots to consider. For these, VizRank projection scores varied from 99.45 (the best projection) to the 50.86 (the worst one). Interestingly, the top ten projections all included an attribute coming from measurements on sporulation, with a second attribute representing a measurement from either heat shock or diauxic shift experiments. The best two projections are shown in Figure 6(a)

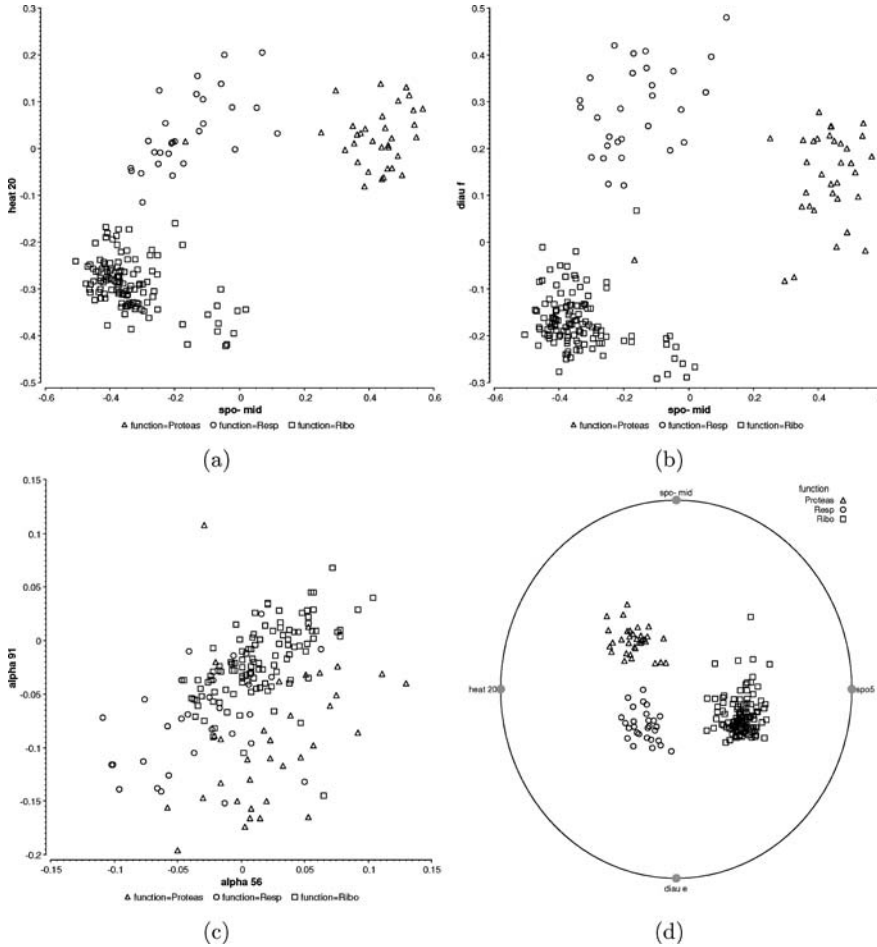


Figure 6. Two best projections found by VizRank, with scores 99.45 (a) and 98.91 (b), showing the scatterplots of 186 genes from three functional groups (*yeast* data set). Attribute "*spo-mid*" corresponds to a gene expression measurement during sporulation of budding yeast, "*heat-20*" to a measurement during the heat shock, and "*diau f*" to a measurement during diauxic shift. For comparison, the scatter-plot (c) uses a middle-ranked projection with a score of 74.73. Figure (d) shows a radviz projection with four attributes that scored 100.0 and offers a perfect separation of functional groups.

and (b). The two scatterplots indicate that a single gene expression measurement during sporulation can clearly separate genes from the proteasome functional group from those from the cytoplasmic ribosomes or respiration. To further separate the latter two functional groups, an additional attribute is required from either heat shock experiment Figure 6(a) or diauxic shift Figure 6(b). The utility of the gene expression measurements during the diauxic shift in characterization of the two of our three functional groups—cytoplasmic ribosomes and respiration—has previously been reported in the study by DeRisi et al. (1997). Both projections from Figure 6(a) and (b) also depict an outlier which is in both cases a gene called *YDR069C (Ubiquitin isopeptidase)*. Interestingly,

YDR069C is one of the genes in the list of consistently misclassified genes by Brown et al. (2000) and reported to be loosely associated with its functional group and regulated differently from the rest of proteasome.

We also investigated the same data set using radviz visualizations with four attributes. Since the overall number of such projections for our data set was large (4,507,503), VizRank was run with search heuristic (Section 4.3) and evaluated only 10,000 most promising projections. Of these, we first found that most projections that well separated genes of different functional groups (score higher than 95) used attributes from at least two different types of experiments (for instance sporulation and diauxic shift). There was no suitable projection where separation would be achieved with all the measurement coming from the same type of experiment. Such result is biologically relevant as it speaks about the minimal number of experiments to define the gene function in this domain. The best projection by VizRank is shown in Figure 6(d). It offers a perfect separation of classes, and an easy interpretation of the influence of attributes: "*heat 20*" and "*spo5 11*" separate genes from cytoplasmic ribosomes functional group from other two groups, while attributes "*diau* e" and "*spo-mid*" enable us to clearly distinguish the proteasome group from the cytoplasmic respiration group.

The three functional classes we have used in our example were also among those studied by Brown et al. (2000). They used support vector machines and report on reliable classification performance. They do not, however, report on particular rules that would, based on 79 measured attributes, characterize the functional groups. Our experiments demonstrate that where such rules exist, VizRank can provide means to find them and identify corresponding visual representation. For a higher number of attributes, a combination of VizRank and radviz proved effective. VizRank, together with associated visualizations, however, is not a replacement to classification induction methods, but should instead be used in the early stage of exploratory data analysis to identify interesting attributes and relations. If, however, VizRank can find simple classification models—like in the case of our functional genomics study—these should probably be used in place of more complex and less intuitive ones. Further motivation and details on utility of VizRank as a tool for functional genomics are reported in Leban et al. (2005).

6. Conclusion

We presented a method called VizRank, that is able to rank data projections by their expected usefulness. The method works with visualization methods where visualized attributes only influence the position of the plotted symbol. Usefulness of a projection can then be defined as a property that describes how well clusters with different class values are geometrically separated. To evaluate usefulness of a projection, we used k-nearest neighbor algorithm and measured its predictive accuracy on a data set constructed from the projection. Such a data set consists of x and y positions of points and their class information. Prediction performance of the algorithm is then used as a numerical measure of projection usefulness. This method allows us to automatically construct and evaluate projections and to present a short list of the most informative projections to data analyst. We also performed an experiment showing that the rankings of projections computed by VizRank method agree almost perfectly with the rankings assigned by data analysts. These experimental results,

together with a successful case study from the field of functional genomics, indicate that our method can be successfully used as an aid to data analyst in exploratory data analysis.

In general, VizRank, which is based on the *k*-NN classifier, lacks the statistical inference apparatus available in LDA or projection pursuit. It is also primarily aimed for original attributes (untransformed dimensions), but in statistical terms it is highly robust since it makes no assumptions about the probability distributions either of the original data, or in the projection space. Furthermore, it has proven to perform well on relatively large and complex data sets, particularly in combination with the proposed search heuristic.

VizRank is implemented within the Orange data mining software (Demšar and Zupan, 2004) and is freely available at http://www.ailab.si/orange. Supplemental pages are available at http://www.ailab.si/supp/DMAKD-05.

Acknowledgments

The authors wish to thank Uros Petrovic for the help on analysis of yeast gene expression data set and twelve post-graduate students of University of Ljubljana who for participating in the experiments. We would also like to acknowledge the support from a Program Grant (P2-0209) from Slovenian Research Agency.

References

Bardorfer, A., Munih, M., and Zupan, A. 2001. Upper limb motion analysis using haptic interface. IEEE/ASME Transactions on Mechatronics, 6(3):253–260.

Blake, C. and Merz, C. 1998. UCI repository of machine learning databases.

Brier, G.W. 1950. Verification of forecasts expressed in terms of probabilities. Monthly Weather Review, 78:1–3.

Broder, A.J. 1990. Strategies for efficient incremental nearest neighbor search. Pattern Recognition, 23(1–2):171–178.

Brown, M.P., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C., Furey, T.S., Ares, M.J., and Haussler, D. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proceedings of the National Academy of Sciences, 97(1):262–267.

Chambers, J.M., Cleveland, W.S., Kleiner, B., and Tukey, P.A. 1983. Graphical Methods for Data Analysis, Chapman and Hall.

Cleveland, W.S. 1993. Visualizing data, New Jersey: Hobart Press (Summit).

Cleveland, W.S. and McGill, R. 1984. The many faces of a scatter plot. Journal of the American Statistical Association, 79(388):807–822.

Cook, R.D. and Yin, X. 2001. Dimension reduction and visualization in discriminant analysis. Australian and New Zealand Journal of Statistics, 43(2):147–199.

Cutting, J.E. and Vishton, P.M. 1995. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. Handbook of perception and cognition, San Diego, CA: Academic Press, pp. 69–117.

Dasarathy, B.W. 1991. Nearest neighbor (NN) norms: NN pattern classification techniques, IEEE Computer Society Press.

Demšar, J. and Zupan, B. 2004. From experimental machine learning to interactive data mining, a white paper. AI Lab, Faculty of Computer and Information Science, Ljubljana.

DeRisi, J.L., Iyer, V.R., and Brown, P.O. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science, 278:680–686.

Diaconis, P. and Friedman, D. 1984. Asymptotics of graphical projection pursuit. Annals of Statistics, 1(12):793–815.

Dillon, I., Modha, D., and Spangler, W. 1998. Visualizing class structure of multidimensional data. Proceedings of the 30th Symposium on the Interface: Computing Science and Statistics, Minneapolis, MN.

Duda, R.O., Hart, P.E., and Stork, D.G. 2001. Pattern Classification, John Wiley and Sons, Inc.

Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. 1998. Cluster analysis and display of genome-wide expression patterns. PNAS, 95(25):14863–14868.

Friedman, J.H., Bentley, J.L., and Finkel, R. 1977. An algorithm for finding best matches in logarithmic expected time. ACM Transactions on Mathematical Software, 3(3):209–222.

Friedman, J.H. and Tukey, J.W. 1974. A projection pursuit algorithm for exploratory data analysis. IEEE Transactions on Computers, 23:881–890.

Grinstein, G., Trutschl, M. and Cvek, U. 2001. High-dimensional visualizations. Proceedings of the Visual Data Mining Workshop, KDD.

Harris, R.L. 1999. Information graphics: A comprehensive illustrated reference, New York: Oxford Press, pp. 290–297.

Hastie, T., Tibshirani, R., and Friedman, J. 2001. The Elements of Statistical Learning, Springer.

Hoffman, P.E. and Grinstein, G.G. 1999. Dimensional anchors: A graphic primitive for multidimensional multivariate information visualizations. Proc. of the NPIV 99.

Hoffman, P.E., Grinstein, G.G., Marx, K., Grosse, I., and Stanley, E. 1997. DNA visual and analytic data mining. IEEE Visualization, 1:437–441.

Huber, P. 1985. Projection pursuit (with discussion). Annals of Statistics, 13:435–525.

Inselberg, A. 1981. *n*-dimensional graphics, part *i*-lines and hyperplanes, Technical Report G320-2711, IBM Los Angeles Scientific Center.

Kaski, S. and Peltonen, J. 2003. Informative discriminant analysis. Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), 1:329–336.

Keim, D.A. and Kriegel, H. 1996. Visualization techniques for mining large databases: A comparison. Transactions on Knowledge and Data Engineering, Special Issue on Data Mining, 8(6):923–938.

Kononenko, I. and Simec, E. 1995. Induction of decision trees using relieff. Mathematical and statistical methods in artificial intelligence, Springer Verlag.

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

Nason, G. 1992. Design and Choice of Projection Indices, PhD thesis, University of Bath.

Santini, S. and Jain, R. 1996. The use of psychological similarity measure for queries in image databases.

Santini, S. and Jain, R. 1999. Similarity measures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 21(9):871–883.

Schucany, W. and Frawley, W. 1973. A rank test for two group concordance. Psychometrika, 2(38):249–258.

Siegel, S. and Castellan, J. 1988. Nonparametric statistics for the behavioral sciences, 2nd edn. McGraw-Hill.

Torkkola, K. 2003. Feature extraction by non-parametric mutual information maximization. Journal of Machine Learning Research, 3:1415–1438.