

Clustering and Classification based on the L_1 Data Depth

Rebecka Jörnsten

*Department of Statistics, Rutgers University
501 Hill Center
PISCATAWAY, NJ, USA 08854*

Clustering and Classification based on the L_1 Data Depth

Rebecka Jörnsten

*Department of Statistics, Rutgers University
501 Hill Center
PISCATAWAY, NJ, USA 08854*

Abstract

Clustering and classification are important tasks for the analysis of microarray gene expression data. Classification of tissue samples can be a valuable diagnostic tool for diseases such as cancer. Clustering samples or experiments may lead to the discovery of subclasses of diseases. Clustering can also help identify groups of genes that respond similarly to a set of experimental conditions. In addition to these two tasks it is useful to have validation tools for clustering and classification. Here we focus on the identification of outliers - units that may have been misallocated, or mislabeled, or are not representative of the classes or clusters.

We present two new methods: DDclust and DDclass, for clustering and classification. These non-parametric methods are based on the intuitively simple concept of data depth. We apply the methods to several gene expression and simulated data sets. We also discuss a convenient visualization and validation tool - the Relative Data Depth (ReD) plot.

Key words: Clustering, Classification, Data Depth, Relative Data Depth

1 Introduction

Gene expression data are comprised of measured gene expression levels, under various experimental conditions. The “activities” of the genes are measured simultaneously for a large collection of genes. The expression levels of a gene across conditions is called the *gene profile*. Conditions, hereafter referred to

Email address: rebecka@stat.rutgers.edu, *phone* (732) 445-3145, *fax* (732) 445-3428 (Rebecka Jörnsten).

URL: <http://www.stat.rutgers.edu/rebecka> (Rebecka Jörnsten).

as *samples*, may correspond to for example a tissue type, or a type of cancer. In such experiments, class discovery (sample clustering) is an important task. Though pathologists may classify cancers in a certain way, genetic markers can identify subclasses of types of cancer. Hierarchical methods are frequently used for gene and sample clustering, though PAM ([10]) has recently gained much popularity. The PAM algorithm (partition around medoids) constrain the cluster representatives to belong to the set of observations, the medoids. Pairwise distances are thus sufficient input to PAM. The cluster assignments are given by a nearest-medoid partitioning of the data set. Though PAM is a more robust clustering algorithm than K-means, the medoid-constraint can be hazardous for noisy data. A K-median algorithm, where the cluster representatives are the multivariate medians (MVM), can improve the robustness of the clustering ([9]). For sample classification, we build predictive models for the sample classes based on all or selected gene expressions. Many classification methods have been applied to gene expression data, such as discriminant analyses (DA), k-nearest neighbors (kNN), support vector machines (SVM), boosting and bagging CART, and neural nets. On most data sets, the simple and complex methods perform near equally well, or poorly. In fact, the study of Dudoit *et al.* (2000) indicates that simple kNN, or diagonal linear DA (DLDA) often results in the best test error rate performance.

In this paper we present two new methods, for clustering and classification, based on the intuitively simple concept of data depth. Consider a cluster of observations. This cluster is identified by either known class labels, or cluster labels that result from a clustering algorithm. With each point z in the data space we identify a depth with respect to the cluster. The depth considered in this paper is the L_1 data depth of Vardi and Zhang ([17]). The L_1 depth of z is the amount of probability mass needed at z to make z the multivariate median of the data cluster - i.e. a robust representative of the cluster. Many other depth functions exist (see e.g. Liu et al, [12]). The L_1 depth is unusual in that it is non-zero outside the convex hull of the data cluster. The L_1 depths are therefore meaningful when comparing multiple clusters. The L_1 depth also has a closed form which makes it an efficient building block in complex algorithms. Data depth based clustering methods have to our knowledge not appeared in the literature before. Christmann ([2]) recently introduced a classifier related to support vector machines and based on the regression depth. However, this method is computationally intensive and the author states that it is at this point not feasible to apply the method to high-dimensional data.

Clustering and cluster validation are often treated as separate tasks. Here we refer to validation as the task of (i) selecting the number of clusters and (ii) identifying outliers. We mean by an outlier a unit that is not representative, by some selected measure, of the cluster it's been allocated to. Instead of treating clustering and validation separately we propose an algorithm that uses a validation criterion, the relative data depth ReD ([9]), as a clustering cost function. Recently, van der Laan et al ([11]) proposed a similar strategy. Their PAMsil algorithm clusters using the popular silhouette width validation

(*sil*) criterion ([10]). *ReD* is a statistic similar to the silhouette width. *ReD* is the difference between the depths with respect to the cluster an observation has been allocated to, and the nearest competing cluster. *sil* is the normalized difference of average distances with respect to the clusters. In contrast to *sil*, *ReD* is *independent of the scales* of individual clusters, and is thus not dominated by high variance clusters. In the present paper we exploit this property of *ReD* to do clustering. Our new clustering method, DDclust finds the partition which maximizes the average of $(1 - \lambda)sil + \lambda ReD$ over the observations, for a fixed $\lambda \in [0, 1]$. The multivariate medians (MVM) are the cluster representatives in DDclust. Thus, for $\lambda = 0$ DDclust is similar but not equivalent to PAMsil. We demonstrate on several simulated data sets that the inclusion of a data depth criterion in the clustering can improve clustering accuracy dramatically, compared to data generative labels. We also apply DDclust to gene expression data sets, for both sample and gene clustering and discuss the results. Different clustering algorithms produce different clusterings. On real data sets it is difficult to decide whether one method is better than another. The various clustering criteria simply focus on different aspects of the data. For example, PAMsil was found to more aggressively identify small clusters in the presence of large clusters compared with PAM. The DDclust method allows for the presence of clusters of different scales. PAM tends to produce clusters of similar scales.

We also propose a classifier based on the L_1 data depth, DDclass. We use real and simulated data to explore the properties of the method. DDclass is highly competitive on the gene expression data, and in our simulation study. We validate the classification results using the relative data depth *ReD*. *ReD* is found to be a good indicator of classification confidence.

The paper is organized as follows. In section 2 we review the L_1 and relative data depths. We present the DDclass algorithm in section 3, and DDclust in section 4. In section 5 we discuss results obtained on several gene expression data sets, and on simulated data. We conclude in section 6 with a discussion.

2 The L_1 Depth and Relative Data Depth

In ([9]) Jörnsten, Vardi and Zhang introduced the Relative Data Depth, *ReD*, as a cluster validation tool used in conjunction with an exact K-median algorithm. The PAM approximation restricts the cluster representatives to medoids. The K-median algorithm uses the multivariate medians (MVM) computed by the Weiszfeld algorithm ([17]). Let us denote by $x_i, i = 1, \dots, N$ the distinct observations in R^p which we wish to cluster. With each observation x_i we associate a multiplicity η_i . If the data set has no tie, $\eta_i = 1, \forall i$. A cluster assignment into K clusters is represented by a partition $I(1), \dots, I(K)$ of

$\{1, \dots, N\}$, where $I(k)$ is the set of labels of those x_i in cluster k . Given a cluster assignment, the representative of the k -th cluster, denoted by $y_0(k)$, is the multivariate L_1 -median defined by

$$y_0(k) = \arg \min C(y|k), \quad C(y|k) = C(y|I(k)) = \sum_{i \in I(k)} \eta_i \|x_i - y\|, \quad \forall k,$$

where $\|x - y\|$ is the Euclidean distance from y to x . Vardi and Zhang ([17]) introduced the L_1 -data depth induced from the multivariate L_1 -median. Given a cluster assignment and $z \in R^p$, let

$$\bar{e}(z|k) = \sum_{i \in I(k), x_i \neq z} \eta_i e_i(z) / \sum_{j \in I(k)} \eta_j, \quad e_i(z) = (x_i - z) / \|x_i - z\|,$$

i.e. e_i is the unit vector from z to observation x_i . Thus $\bar{e}(z|k)$ is the average of the unit vectors from z to all observations in the k -th cluster. The L_1 data depth of point z with respect to the k -th cluster is defined as

$$D(z|k) = 1 - \max [0, \|\bar{e}(z|k)\| - f(z|k)],$$

where $f(z|k) = \eta(z) / (\sum_{i \in I(k)} \eta_i)$ with $\eta(z) = \sum_{i=1}^N \eta_i I\{z = x_i\}$. The statistical interpretation of the cluster data depth is that $1 - D(z|k)$ is the minimum additional weight needed at z in order to make z the multivariate L_1 -median for the data set $\{z\} \cup \{x_i, i \in I(k)\}$ ([17]). In figure 1 (a) we illustrate how the L_1 depth is computed. For a central observation z_1 (deep) $\bar{e}(z_1) \simeq 0$ and $D(z_1)$ is close to 1. For a peripheral observation z_2 (not deep) $\bar{e}(z_2) \simeq 1$ and $D(z_2)$ is close to 0. ReD is a natural extension of the L_1 -data depth to multiple clusters. Each x_i is associated with cluster data depths $D(x_i|1), \dots, D(x_i|K)$. We define the within-cluster data depth of observations $x_i, i \in I(k)$ as $D_i^w = D(x_i|k)$. For $i \in I(k)$, we order the remaining $K - 1$ cluster data depths, $D(x_i|l), l \neq k$, by $D(x_i|l_1(i)) < \dots < D(x_i|l_{K-1}(i))$ according to $\|x_i - y_0(l_1)\| \leq \dots \leq \|x_i - y_0(l_{K-1})\|$. We define the between-cluster data depth of observation $x_i \in I(k)$ as $D_i^b = D(x_i|l_1(i))$, the depth with respect to the nearest competing cluster. ReD is the difference between the within- and between-cluster data depths. An observation x_i is well-clustered if $D_i^w \gg D_i^b$, i.e. ReD_i is close to 1. We select the number of clusters K that maximizes the average relative data depth $ReD(K) = \sum_{i=1}^N \eta_i ReD_i$, $ReD_i = D_i^w - D_i^b$. The ReD selection statistic is similar to the average silhouette width, where the silhouette width of observation i is defined as follows. For $z \in R^p$, let $\bar{d}(z|k) = \sum_{i \in I(k), x_i \neq z} \eta_i \|z - x_i\| / \sum_{i \in I(k), x_i \neq z} \eta_i$ be the average distance between z and observations not equal to z in the k -th cluster. For $x_i, i \in I(k)$, we compute the *silhouette width* $sil_i = (b_i - a_i) / \max(a_i, b_i)$, $a_i = \bar{d}(x_i|k)$, $b_i = \min_{l \neq k} \bar{d}(x_i|l)$. The silhouette method selects the number of clusters K that maximizes the average silhouette width $\sum_i \eta_i sil_i / \sum_i \eta_i$. The silhouette widths are affected by the scales of individual clusters. Thus, if the data is drawn from a distribution with different within-cluster variances, the silhouettes tend to be larger for the tight clusters. The silhouette width may fail to identify outliers in such

clusters and may lead to under-fitting ([9]). In contrast, ReD is independent of cluster scales and is not driven by high variance clusters (figure 1 (b,c)). The within- and between-cluster data depths can be used for cluster validation by visual inspection. An example with $K=3$ clusters is shown in the top panel of figure 4. Above the x -axis the sorted within-cluster data depths for each cluster are displayed, with corresponding between-cluster data depths below. Just below the x -axis are the first tier data depths, and stacked below are the second tier depths, colored by cluster. A lot of information is contained in these plots. Color patterns in the between-cluster data depths gives information about cluster boundaries. Observations that are have large L_1 -depths with respect to several clusters are potential outliers.

3 Classification via the L_1 Data Depth

The L_1 data depth of an observation with respect to a data cluster indicates how representative of the cluster that observations is. Thus, for data with class labels we expect the L_1 data depth to be maximized with respect to the data cluster corresponding to the correct class label. A simple classification rule is to classify observations with unknown labels by their maximum data depth with respect to labeled data. We refer to the labeled data as the training set, and the unlabeled as the test set. The rule applied to an observation x_i in either the training or the test set is thus

$$\hat{k}_i = \arg \max_l D(i|l), \quad l \in \{1, \dots, C\},$$

where C is the number of classes. For each observation x_i in the training set we can compute the relative data depth, ReD_i^{train} . If the correct label of observation x_i is k ,

$$ReD_i^{train} = D(i|k) - \max_{l \neq k} D(i|l), \in [-1, 1].$$

A training error corresponds to $ReD^{train} < 0$. For an observation x_i in the test set we do not know the correct label. Therefore the ReD value is computed according to the classification rule:

$$ReD_i^{test} = D(i|\hat{k}_i) - \max_{l \neq \hat{k}_i} D(i|l), \in [0, 1].$$

A small value of ReD^{test} suggests that we have little confidence in the classification of that test set observation. Mislabeled or noisy observations in the training set are a source for concern. Even if these observations are classified correctly on the training set, their presence may negatively affect the classification performance on a test set. We wish to identify such observations and

remove them from the training set before classifying the unlabeled data. We take a cross validation approach. We apply the data depth based classification rule to leave-one-out cross validation training data sets, and remove misclassified test observations. We refer to the basic classification rule as DDclass, and the rule with observations removed via cross validation as DDclass-CV. We outline the algorithms below.

DDclass

- (1) For observations j in the test set TE, predict label $\hat{k}_j = \arg \max_k D(j|TR^k)$, where TR^k is the set of observations in the training set labeled k .
- (2) For observations j in the test set TE, compute

$$D_j^{win} = \max_k D(j|TR^k), D_j^{next} = \max_{k \neq \hat{k}_j} D(j|TR^k),$$

and the Relative Data Depth, $\text{ReD}_j^{test} = D_j^{win} - D_j^{next}$.

DDclass-CV

- (1) On the training set TR , construct the leave-one-out training sets TR_b , $b = 1, \dots, N$, where N is the number of observations.
- (2) for $b = 1, \dots, N$
 - (i) For observation b , compute the L_1 data depths $k = 1, \dots, K$, $D(b|TR_b^k)$.
 - (ii) Predict label $\hat{k}_b = \arg \max_k D(b|TR_b^k)$
- (3) Identify set T such that $T = \{b : b \in TR, k(b) \neq \hat{k}_b\}$, where $k(b)$ is the correct label of observation b .
- (4) Form a reduced training set $TR^* = TR_{-T}$
- (5) For observations j in the test set TE, predict label $\hat{k}_j = \arg \max_k D(j|TR^{k,*})$
- (6) For observations j , compute the Relative Data Depth, ReD_j^{test} .

The ReD^{test} values are useful for visualization and validation. In the result section we show that low ReD values are associated with high test error rates. We also consider a classifier based on average distance, and a validation tool based on the silhouette width. We refer to this classifier as SILclass. The tuning algorithm SILclass-CV is similar to DDclass-CV, with the relative data depth replaced by the silhouette width.

SILclass

- (1) For observations j in the test set TE, predict label $\hat{k}_j = \arg \min_k \bar{d}(j|TR^k)$, where TR^k is the set of observations in the training set labeled k .
- (2) For observations j in the test set TE, compute

$$a_j = \bar{d}(j|TR^{\hat{k}}), b_j = \min_{k \neq \hat{k}_j} \bar{d}(j|TR^k).$$

The silhouette width, $\text{sil}_j^{test} = (b_j - a_j) / \max(b_j, a_j)$ is an indicator of classification confidence.

DDclass and SILclass could also be combined, i.e. classifying according to a weighted combination of depths and average distances. For simplicity we limit our study to pure depth based or distance based classifiers here. In the next section we explore the use of a combination of the criteria to do clustering.

4 Clustering via the Relative Data Depth

A simulation study in [9] showed that *ReD* is a good cluster validation tool. This suggests it may also be a useful criterion for clustering. A natural way of including a data depth criterion in clustering is through constrained vector quantization, usually implemented via the generalized Lloyd-Max algorithm (G-LM) ([14]). G-LM iterates between two conditions, the C-step and the N-step. The C-step computes the cluster representatives (multivariate medians, MVM) given the current partition. In standard LM the N-step is simply nearest-MVM allocation. The constraint is reflected in the N-step of the G-LM algorithm. An example is entropy constrained vector quantization (VQ) where the partition boundaries are shifted toward clusters with low cardinality. Here we wish to substitute the data depth for the entropy. The N-step now defines the boundary between clusters k and l , as points $z \in R^p$ where

$$\{z : (1 - \lambda)\|z - y_0(k)\| - \lambda D_z^w(k) = (1 - \lambda)\|z - y_0(l)\| - \lambda D_z^w(l)\}.$$

The tuning parameter $\lambda \in [0, 1]$ controls the amount of influence the data depth has over the clustering. In entropy constrained VQ we cluster for fixed values of λ , and search over the space of $\lambda \in [0, 1]$ until the entropy constraint is satisfied. Here however, there is no known a-priori lower bound for the data depth to satisfy. The relative range of the L_1 -distortions and the depths D^w are widely different, and will vary from problem to problem. This makes choosing an appropriate value for λ difficult. One possibility is to choose λ based on some measure of robustness and cluster reproducibility, e.g. via the cluster prediction strength of Tibshirani et al ([16]) or the resampling techniques of Dudoit et al ([4]). However, these computationally intensive methods are decidedly impractical for estimating λ since we have to generate multiple clusterings for each value of the tuning parameter.

We here choose to work in a simplified setting, using the silhouette widths and relative data depths, both with range $[-1, 1]$. The role of L_1 -distortion is played by the silhouette width, the relative data depth replaces the data depth. The clustering criterion function we maximize with respect to a partition $I_1^K = \{I(1), \dots, I(K)\}$ is thus given by

$$\arg \min_{I_1^K} C(I_1^K), \quad C(I_1^K) = \frac{1}{N} \sum_{k=1}^K \sum_{i \in I(k)} (1 - \lambda) sil_i + \lambda ReD_i, \quad \lambda \in [0, 1].$$

We refer to the clustering method that maximizes $C(I_1^K)$ as DDclust. The choice of λ is now easy to interpret. For $\lambda = 0$ the clustering criterion is equivalent to the average silhouette width. For $\lambda = 1$ it is given by the average relative data depth. Recently, van der Laan et al ([11]) proposed the PAMsil algorithm. This algorithm searches for the medoids that maximize the average silhouette width. PAMsil is similar, but not equivalent to DDclust with $\lambda = 0$ since we do not restrict the cluster representatives to medoids. By varying λ we find alternative clusterings of the data. Small λ encourages equal scale clusters, whereas larger λ allows for the presence of unequal scale clusters.

For practical implementation of an algorithm that maximizes $C(I_1^K)$ we take an iterative approach. A starting point is readily available from PAM. Another approach is to seed the algorithm with randomly selected observations, sufficiently distant, as cluster representatives. At each iteration we move observations x_i to new locations and accept a new partition \tilde{I}_1^K if $C(\tilde{I}_1^K) > C(I_1^K)$. We use simulated annealing to avoid getting trapped in local maxima. That is, a new partition $I_1^K \rightarrow \tilde{I}_1^K$ is accepted if $C(\tilde{I}_1^K) > C(I_1^K)$. If $C(\tilde{I}_1^K) \leq C(I_1^K)$ we still accept the new partition \tilde{I}_1^K with probability $P(\beta, \Delta(C))$, where $\Delta(C) = C(I_1^K) - C(\tilde{I}_1^K)$, and β is a tuning parameter. $P(\beta, \cdot)$ is decreasing with increasing β , and $P(\cdot, \Delta(C))$ is increasing with decreasing $\Delta(C)$. We increase β every iteration such that the probability of accepting a non-improving partition approaches zero. The relocation of one observation affects most other observations and thus $C(I_1^K)$. The computational burden of DDclust is thus much greater than that of PAM. PAMsil also requires the update of all the silhouette widths in the medoid-swap. In order to speed up the computation we restrict the set of observations that can be relocated. We select a threshold T and only consider observations x_i such that $c_i = (1 - \lambda) sil_i + \lambda ReD_i \leq T$. In addition, we limit the possible destination of x_i to the nearest competing cluster as defined by x_i 's distance to the corresponding multivariate median (MVM). That is $x_i, i \in I(k)$ can only be moved to $I(l) : l = \arg \min_{j \neq k} \|x_i - y_0(j)\|$.

DDclust

- (1) Start with an initial partition I_1^K , e.g. the PAM clustering. Set $\beta = \beta_{init}$.
- (2) Compute the multivariate medians $y_0(1), \dots, y_0(K)$, the silhouette widths sil_i and relative data depths ReD_i for all observations $x_i, i = 1, \dots, N$. Compute the total value of the partition $C(I_1^K)$.
- (3) For all observations compute $c_i = (1 - \lambda) sil_i + \lambda ReD_i$. Identify a set of observations $S = \{i : c_i \leq T\}$, where T is a prefixed threshold.
- (4) For a random subset $E \subset S$, identify the nearest competing clusters. Define the partition with E relocated as \tilde{I}_1^K .
- (5) Compute the value of the new partition, $C(\tilde{I}_1^K)$. If $C(\tilde{I}_1^K) > C(I_1^K)$ set $I_1^K \leftarrow \tilde{I}_1^K$. If $C(\tilde{I}_1^K) \leq C(I_1^K)$ set $I_1^K \leftarrow \tilde{I}_1^K$ with probability $Pr(\beta, \Delta(C))$. Otherwise keep I_1^K . Set $S = S_{-E}$, removing the subset E from S .
- (6) Iterate 4-5 until set S is empty.
- (7) If no moves were accepted for the last M iterations and $\beta < \infty$, set $\beta = \infty$

and iterate 3-6. If no moves were accepted for the last M iterations and $\beta = \infty$, terminate the algorithm. Otherwise set $\beta = 2\beta$ and iterate 3-6.

PAM clustering is in most cases a good starting point for DDclust. We compared results obtained using random starting medoids/MVMs. If a random starting point was used it was necessary to include simulated annealing ($\beta_{init} < \infty$). We converged to the same clustering as the PAM seeded algorithm. If PAM was the starting point, and $\beta_{init} = \infty$ the algorithm terminated quickly. In the simulation study we set $\beta_{init} = \infty$, but we recommend $\beta_{init} < \infty$ if single data sets are analyzed, or that DDclust with $\beta = \infty$ is run several times to check for convergence to a local optimum. The random sets $E \subset S$ were chosen by first drawing the size of the set $|E| \sim U[1, MAX]$, and then randomly $|E|$ elements from S . We used $MAX = 5$. We set the iteration factor $M = 5$ in our implementation. The algorithm terminates faster if T is small since fewer observations are considered for relocation. Small T is acceptable if the initial partition is close to the optimum. However, if T is too small we are less likely to find the optimum clustering. $T = 0$ is a natural choice and worked well in practice. We also compared clusterings generated when T was selected to allow for a fixed fraction of observations to be considered for relocation.

The computational burden of DDclust is substantial. The number of units to be clustered is the most constraining factor since for each observation a depth and silhouette value has to be computed, and updated for every partition. Consider the following example. Data was generated according to a simulation model presented in the next section. If we run DDclust ($\beta = \infty$) on a data set of size $N = 150$ the algorithm terminates in 20 seconds. If we increase N to 300 DDclust terminates in 2 minutes, and for $N = 450$ it takes a little over 6 minutes. The algorithm scales well to high-dimensional data since all depth and distance computations are vectorized. On the leukemia data set in the next section we examine the cases of dimension 100, 200 and 1000. DDclust terminates in 10 seconds, 1 minute and 2 minutes respectively. These results were obtained running R on a 1GHz PC.

5 Results

In this section we apply our data depth clustering algorithm, DDclust, and the classification method, DDclass, to several data sets. We study three different gene expression data sets. On two publicly available data sets we use DDclust and DDclass for sample clustering and classification. On these particular data sets sample clusterings agree quite well with the known sample labels, though this need not be the case with an unsupervised approach. On a third gene expression data set we use DDclust for gene clustering.

To study the performance of the algorithms in a more controlled setting we also generate simulated data from several models. Our simulation setup is

multivariate normal, with three classes, three informative variables and ten noise variables. The class means are $(0, 0, 0)$, $(\delta, -\delta, \delta)$, $(-\delta, -\delta, -\delta)$ for the informative variables and mean zero for the noise variables. We use parameter values $\delta = 2, 3$. The class covariance structure is diagonal, with different scale for the informative variables in each class parametrized by a parameter γ , $\sigma^2 = 1, 1/\gamma^2$ and γ^2 respectively, $\gamma = 1, 2$ or 3 . The noise variables have unit variance. To examine the performance of the DDclust algorithm we generate 50 observations from each class for training. We study the learning error rates compared with the generative class labels. We also generate independent test sets, 500 observations from each class and classify according to the partitions generated by DDclust. For classification we generate a smaller data set for training, 25 observations in each class. We apply DDclass to these data sets and independent test sets consisting of 500 observations from each class.

On the gene expression data sets we use leave-one-out cross validation to examine the performance of the DDclust algorithm. To estimate the test error rate of classifiers we use random tenfold cross validation. Each cross validation training set is formed by randomly selecting nine tenths of the samples for training, and keeping one tenth for testing. Using the training sets we prefilter the genes. For clustering we select the top 100, 200 and 1000 most variable genes. For classification, we identify the top 200 genes that discriminate between the classes via the between-to-within (B/W) sum of squares. We compare the DDclass cross validation test error results to those of kNN, where the number of neighbors are selected through cross validation. We also include various discriminant analysis methods.

Spinal Cord Injury data - Gene clustering. We applied the DDclust algorithm to a gene expression data set provided by Jonathan Z. Pan and Ronald P. Hart, at the W.M. Keck Center for Collaborative Neuroscience, and the Department of Cell Biology & Neuroscience, Rutgers University. The data set consists of collected gene expression data from 22 NGEL 2.0 rat oligonucleotide arrays with 4,803 genes and control spots. 22 Long-Evans Hooded rats were treated with 5 different anti-inflammatory drugs 1 hour prior to axotomy and explantion into culture. Inflammation likely contributes to secondary damage after spinal cord injury (SCI). It is therefore of great interest to study the effect of the administration of anti-inflammatory drugs in general, and some drugs in particular, on injured spinal cord. The gene expression profiles were compared for the following treatments: MP, acetaminophen, indomethacin, NS-398, a combination of IL-1ra and soluble TNFR:Fc (5 drugs), as well as for uninjured and injured but untreated samples. The spinal cord was dissected into 1mm segments and incubated with or without drugs in serum-free medium for 4 hours, then total cellular RNA was prepared. Three replicate experiments for each drug (four for the fifth drug combination) were conducted. Each microarray data set was normalized using loess and control spots. An ANOVA filter with Benjamini-Hochberg multiple testing adjustment at level 5% was used

to remove genes that were not significantly differently expressed between the 7 conditions. This left 313 genes for further study. (These preprocessing steps were conducted using the GeneSpring software, [5]).

We applied DDclust and PAM to the 313×22 size data set with varying numbers of clusters, and $\lambda = 0, 0.25, 0.5, 0.9, 1$. We also clustered the reduced dimension data set 313×7 consisting of the median gene expression measurement of each drug class. The average silhouette width was maximized for six clusters using the full data set, and five clusters using the median data set. The same was true using *ReD* to select the number of clusters. The sixth cluster contained one single gene, with one outlying measurement in one of the drugs. We focus here on the reduced dimension data set, the median values.

Among the clusters, the fifth stands out from the rest. It is a small cluster clearly separated from the other clusters. The average *ReD* and *sil* values for Cluster 5 are high (table 1). This cluster contains genes that are upregulated for one drug only - NS-398. A partial list of the genes in this cluster be found in table 2. These genes are related to reduced scarring, detoxification and enhanced macrophage accumulation - all proposed to be related to reduced secondary damage. In vivo results confirmed that NS-398 does reduce lesion volume compared to untreated tissue. Based on previous experiments (not side-by-side comparisons), neither MP nor acetaminophen significantly reduced lesion volumes under these conditions.

The other gene clusters contain more genes and are not as well separated as the fifth cluster. Cluster 1 contains genes that are downregulated in the drug samples compared with uninjured and untreated samples. Cluster 4 contains genes that are upregulated in all the drug samples. Clusters 2 and 3 are clusters with less apparent structure, and contain genes with differential expression close to 1 for most of the samples. The data depth plot (figure 2) can be used to assess the quality of the gene clusters (depicted is the PAM clustering). Cluster 5 is clearly a well-clustered set of genes. The color patterns in the lower panel indicate that Clusters 1 and 2 are neighbors and are poorly separated. The average silhouette of Cluster 1 is relatively high (table 1). This is because both Cluster 1 and 2 are small scale clusters (low variance), and this inflates the silhouette width. Cluster 2 is also a neighbor of Clusters 3 and 4, and contains a few outliers that are deeper with respect to those clusters. Since Clusters 3 and 4 are larger scale clusters the silhouette widths of Cluster 2 are not as large on average as those of Cluster 1. However, the observations in Cluster 2 neighboring Cluster 1 have large silhouette widths despite the lesser separation of these clusters. In figure 3, the lower panel shows the silhouette widths of the PAM clustering. As discussed above the silhouette widths can be misleading when the cluster variances (scales) differ. Another example is the well separated Cluster 4, neighboring the small variance Cluster 2. The silhouette width thus identifies Cluster 4 as a very poor cluster, but the relative data depths for this cluster is quite high (figure 3 top).

Applying the DDclust algorithm with varying λ did not affect the fifth cluster. Increasing λ from 0 to .5 and above moved 13 genes from Clusters 1 and 2

to Cluster 4. This cluster contains genes that are upregulated in all the drug treatment classes. Among the genes that were relocated are Metallothionein, a standard stress response which protects cells from oxidative damage, and an enzyme - a prostaglandin degrader. Prostaglandins production is inhibited by aspirin. A follow-up study is now underway and future work may help shed more light on the roles of these genes in the treatment of spinal cord injury.

Leukemia. The data set comes from a study of gene expressions in two types of leukemia: acute lymphoblastic leukemia (ALL), and acute myeloid leukemia (AML). The study was made on 25 cases of AML, and 47 cases of ALL, i.e. $n = 72$ samples total. The ALL cases were comprised by 38 B-cell type (ALL-B) and 9 T-cell type (ALL-T) samples. Gene expressions were measured for $p = 6,817$ genes simultaneously. We follow the approach of Golub et al and remove genes with low signal-to-noise ratio ($\max/\min < 5$ or $\max - \min < 500$, over 72 samples). This reduces the number of genes to 3,571. We also standardize *samples*, to mean 0 and variance 1. Most genes exhibit little variation across samples. These near constantly expressed genes are not very useful for sample cluster or class separation, though may of course be biologically relevant. Here we select only the 100, 200 or 1000 most variable genes (across samples) for further analysis. The reduced data set thus consists of $K = 3$ classes of standardized samples, in $p = 100, 200$ or 1000 dimensions.

We apply PAM and the DDclust algorithms to these data set with $K = 3$ clusters. A single application of PAM ($p = 100$) produces the silhouette width plot in the bottom panel of figure 4. The PAM cluster labels agree closely with the sample labels. The ALL-B sample 17 is misallocated to the ALL-T cluster. Still, sil_{17} is relatively high (index 47 in the plot). The top panel of figure 4 shows the data depths of the PAM clustering. We apply the DDclust algorithm with $\lambda = 0, 0.25, 0.5, 0.9, 1$. For $\lambda = 0, 0.25$ and 0.5, observation 66 (index 70) is moved from its correct location in the ALL-B class to the AML class. This observation had a negative *sil* value, but a positive data depth value. A data depth plot for the new partition identifies this observation with a negative *sil* and *ReD* value. This is an observation that is clearly difficult to cluster. Increasing λ above 0.5 moves observation 66 back to its original location, and moves observation 67 (index 46) from the ALL-T to the ALL-B class. This observation had a negative *ReD* value in the original partition. A new data depth plot shows a near 0 *ReD* value for this observations in the updated partition. Both observations 66 and 67 were observations that were difficult to classify in a supervised fashion (e.g. [3]). Looking at the data depth plot (top panel of figure 4) observation 66 and 67 (index 46 and 70) both fall between the three clusters. The above results were obtained with threshold $T = 0$ without simulated annealing. We now set $T = .1$ and use simulated annealing. This did not change the clustering outcome for $\lambda > 0.5$, but for $\lambda = 0.5$ we now get the same solution as for $\lambda > 0.5$. We find that the partition $C(I_1^K)$ with $\lambda = 0.5$ and $T = 0$ corresponds to a local maximum.

We use leave-one-out cross validation to further evaluate the results. On each

validation training set we prefilter the genes, cluster and then predict the label for the omitted observations. In table 5 the number of misclassifications using 100, 200 or 1000 genes are reported. On this data set PAM, K-median and DDclust perform similarly. With 1000 genes PAM and DDclust ($\lambda = 0$ or 1) clusterings start to deviate from the sample labels, while DDclust results with midrange λ remain stable.

We now cluster the top 200 genes using DDclust and PAM. The average silhouette width and *ReD* statistic were both maximized with six gene clusters. Examination of the six gene clusters show they are clearly related to the sample classes. Though the gene clustering is unsupervised, we check whether the cluster representatives can be used for sample classification. We repeat the clustering on 10-fold cross validation sets and use the generated gene clusters to classify the omitted samples via kNN. The PAM medoids result in 7 test errors, and the K-median MVMs in 6 errors. DDclust ($\lambda = 0, .25, .5, .9, 1$) results in 4, 4, 6, 4 and 4 errors respectively. With more weight put on the silhouette width (small λ) observation 66 is misclassified, whereas observation 67 is misclassified when more weight is put on the data depth (large λ).

We compare classification of the Leukemia data with DDclass, SILclass, kNN, and DLDA. On this data set all methods perform about equally well. In table 3 the 500 random tenfold cross validation error rates are shown. We also state the percent of cross validation sets where a given method achieved the best test error rate. In most cases the methods performed the same, so this percentage is relatively high across the board. DDclass is the best, followed closely by kNN and DLDA. In figure 5 the testerror rates of the leukemia samples across the 500 tenfold cross validation sets are shown together with the relative data depths ReD^{test} . High testerror rates are associated with low ReD^{test} values. The exception is observation 67, a sample that most methods misclassify. In figure 6 we compare testerror rates and sil^{test} values. The low sil values also correspond to high testerror rates. Clearly, ReD^{test} and sil^{test} are good tools for identifying observations that we classify with little confidence. In figure 7 we show the proportion of times DDclass-CV remove an observation from the training set prior to classifying the test set. Observations 2, 12, 17, 66 and 67 stand out as observations that are frequently removed. Some of these observations also have a high testerror rate. On the full data set DDclass and DLDA give 0 training errors, SILclass 1 error. The tuning methods remove the same 2 observations, samples 17 and 67.

Colon. Alon *et al* (1999) used high-density oligonucleotide arrays to study gene expression patterns in tumor and normal colon tissues. The expression levels of $p = 6500$ genes were measured in $n = 62$ samples, 40 tumor and 22 normal colon tissue samples. In the publicly available data set, the 2000 genes with the highest minimal intensity across samples were kept for further study. As for the Leukemia data we find the 100, 200 and 1000 genes that exhibit maximum variation across the 62 samples and cluster the samples using these genes. The

results with 200 genes are shown in table 5. With 100 genes and 1000 genes all algorithms generate results that deviate substantially from the sample labels. To simplify the discussion here we focus on the case (200 genes) where the sample labels and cluster labels are close. We compare the leave-one-out misclassification errors using PAM, K-median and DDclust with $\lambda = 0, 0.25, 0.5, 0.9, 1$ and $T = 0$, for $K = 2$ clusters. K-median and DDclust ($\lambda = .25, .5$) predictions agree closely with the sample labels. DDclust ($\lambda = 0, .9, 1$) then follows, and lastly PAM. The classification performance is comparable for all the methods (table 4). DLDA is followed by DDclass. SILclass and kNN perform a little worse. On the full data set DLDA misclassifies 8 observations, whereas we obtain only 2 training errors with all the other methods. 8 observations were removed from the data set by DDclass-CV and SILclass-CV

Simulation study. We generate training and test data according to the model described in the beginning of the section. 50 data sets were generated from each sub-model, and training and test error rates compared for PAM and DDclust with $\lambda = 0, 0.25, 0.5, 0.9$ and 1. We set $T = 0$ in the simulation study. However, to make a fair comparison of different λ we find the maximum fraction of observations in the set $\{c_i < T\}$ across λ . We then readjust T for each λ to allow for the same fraction of observations to be considered for relocation. In figure 8 the test errors of DDclust are shown for the simulation model with parameters $\delta = 2, 3$ and $\gamma = 1, 2, 3$. In each row the right most figure corresponds to a higher degree of separation of the classes. In each column, the lower panels correspond to a higher degree of scale difference between the classes. In all figures it is clear the PAM clustering results in poor classification performance on the test set. The K-median shows a drastic improvement. As we increase the separation all methods perform better. As we increase scale difference, the performance of DDclust with small λ starts to deteriorate. DDclust with $\lambda = 1$ is the best for any $\gamma \neq 1$, whereas DDclust with moderate $\lambda \leq .5$ is better for $\gamma = 1$. A λ in the midrange is a safe choice in that it allows for equal or unequal scale clusters. This is also suggested by the cross validation studies in the previous subsection.

We compare DDclass, SILclass, kNN and the Bayes methods on the simulated data sets. In figure 9 we study the same simulation scenarios as above. kNN performs rather poorly. DDclass and SILclass show similar performance when $\gamma = 1$. For $\gamma \neq 1$ SILclass performance deteriorates. However, DDclass continues to be competitive with the Bayes rules (here DLDA/DQDA), especially for $\delta = 3$. DDclass-CV improves slightly over DDclass. SILclass-CV results are similar to those of SILclass.

6 Conclusion

We presented two new methods, DDclust and DDclass, for clustering and classification. DDclust finds clusterings which maximize a combination of average silhouette width and average relative data depth. The degree to which the data depth determines the clustering is controlled by a tuning parameter λ . For $\lambda = 1$ the relative data depth is the sole criterion, for $\lambda = 0$ we maximize the average silhouette width. We compared clustering results to data generative labels in a simulation study. We found that small λ are appropriate for finding equal scale clusters, and large λ for finding unequal scale clusters. Varying λ thus allows the clustering method to focus on different aspects of the data. We did not address the issue of selecting values for λ . Our simulation study suggests that moderate values for $\lambda \simeq 0.5$ works well in multiple settings - i.e. finds equal scale or unequal scale clusters in accordance with the data generative distributions. A cross validation study on real gene expression data showed that DDclust was robust and generated clusters could be used to predict sample labels better than PAM. Gene clustering with DDclust performed better than PAM in terms of the gene clusters' sample predictive properties. DDclass (and SILclass) are competitive classifiers on the gene expression data sets we analyzed and performed well on the simulated data. The visualization and validation tools (ReD and *sil* test values) were shown to agree well with actual classification performance.

R code for DDclust, DDclass and the *ReD* visualization tool is available at <http://www.stat.rutgers.edu/~rebecka/DDcl>.

At present neither method can handle missing data. Imputation of missing values has to be done prior to applying the methods. We are currently working on the further development of the methods for the missing data scenario. Most other data depths cannot be used for clustering since they attain the same value for any distance from a cluster. We are interested in examining whether some depth measures are preferable to others in clustering and classifying data of different structure.

7 Acknowledgments

The Spinal Cord injury data was provided by Jonathan Z. Pan and Ronald P. Hart, at the W.M. Keck Center for Collaborative Neuroscience, and the Department of Cell Biology & Neuroscience, Rutgers University.

This work was in part supported by a grant from the NSF, DMS 0306360.

The author wishes to thank two anonymous referees for their many helpful comments.

Tables

cluster	avg. dist	separation	avg. <i>sil</i>	avg. DD
1	.40	.19	.53	.24
2	.61	.19	.29	.23
3	.82	.33	.24	.23
4	1.30	.383	.03	.28
5	.98	.84	.46	.35

Table 1

Cluster information: separation, average silhouette width and data depth for the 5 clusters in the SCI data

Genes in cluster 5	Function
Heme oxygenase	Heat shock protein, defense mechanism against free radicals, detoxification
small inducible gene JE	increased accumulation of macrophages
MCP-1/MIP-1	"
decorin	reduced scarring in CNS

Table 2

4 genes in cluster 5, upregulated in NS398 group

Leukemia	10-fold CV (200) fivenumber summ.	% of CV sets w. best error rate
DDclass-CV	(0,0,0,12.5,25)	92.6
DDclass	"	"
kNN	(0,0,0,12.5,25)	88.8
SILclass	(0,0,0,12.5,37.5)	86.8
SILclass-CV	"	"
DLDA	(0,0,0,12.5,25)	87.6

Table 3

Leukemia - five number summaries (min, lower quartile, median, upper quartile, max) of cross validation test error rates. % of CV sets where the test error rate is the best across methods.

Colon	10-fold CV (200) fivenumber summ.	% of CV sets w. best error rate
DDclass-CV	(0,0,16.7,16.7,66.7)	85.8
DDclass	“	“
kNN	(0,0,16.7,16.7,66.7)	76.2
SILclass	(0,0,16.7,16.7,66.7)	81.2
SILclass-CV	“	79.2
DLDA	(0,0,16.7,16.7,50)	92.8

Table 4

Colon - five number summaries (min, lower quartile, median, upper quartile, max) of cross validation test error rates. % of CV sets where the test error rate is the best across methods.

Leukemia	100 genes	200	1000	Colon	200 genes
PAM	1	2	14	PAM	18
KMEDIAN	1	2	2	KMEDIAN	11
$\lambda = 0$	4	4	10	$\lambda = 0$	13
$\lambda = .25$	2	3	2	$\lambda = 0.25$	9
$\lambda = .5$	3	3	2	$\lambda = 0.5$	9
$\lambda = .9$	2	3	8	$\lambda = 0.9$	15
$\lambda = 1$	2	4	9	$\lambda = 1$	15

Table 5

Leukemia and Colon - clustering results. 10 fold cross validation misclassification errors. Partitions were generated on training sets, and then used to classify the test sets.

Figures

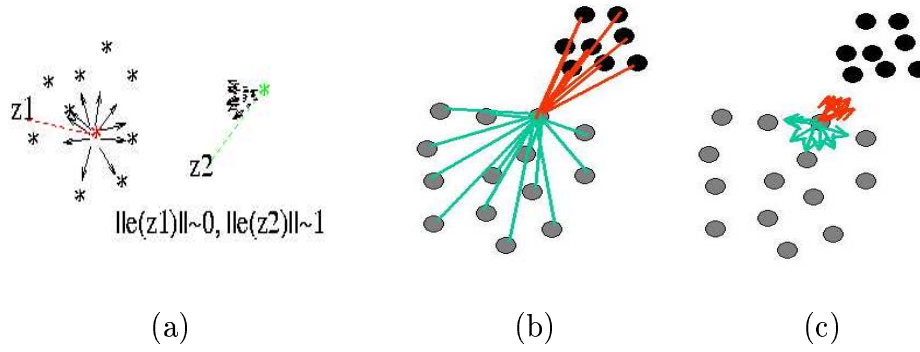


Fig. 1. (a) Computing the L_1 data depth of observations z_1 (deep) and z_2 (not deep), (b) The Silhouette width - the marked observation has sil close to 0 because it belongs to the large scale cluster, (c) The Relative Data Depth - the marked observation has ReD greater than 0 since ReD is not affected by the different scales of the clusters.

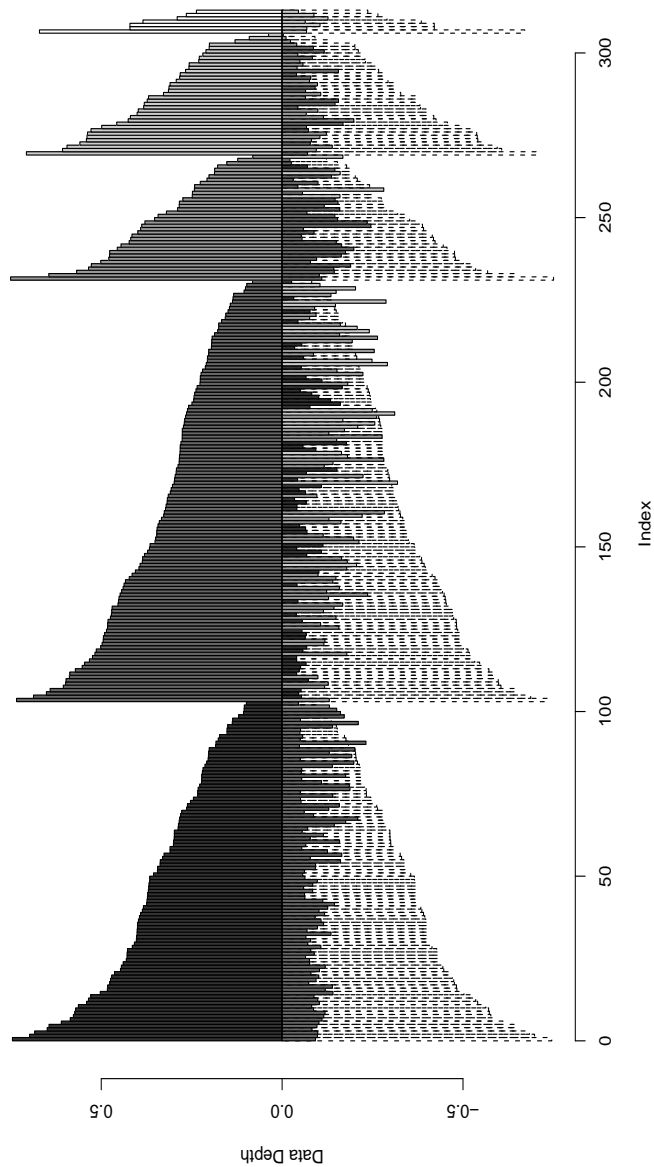


Fig. 2. Data Depth plot of the spinal cord injury data. There are 5 clusters. The top panel depicts the within-cluster data depth, below the depths with respect to the nearest competing cluster. Dashed lines are the mirrored within-cluster data depths.

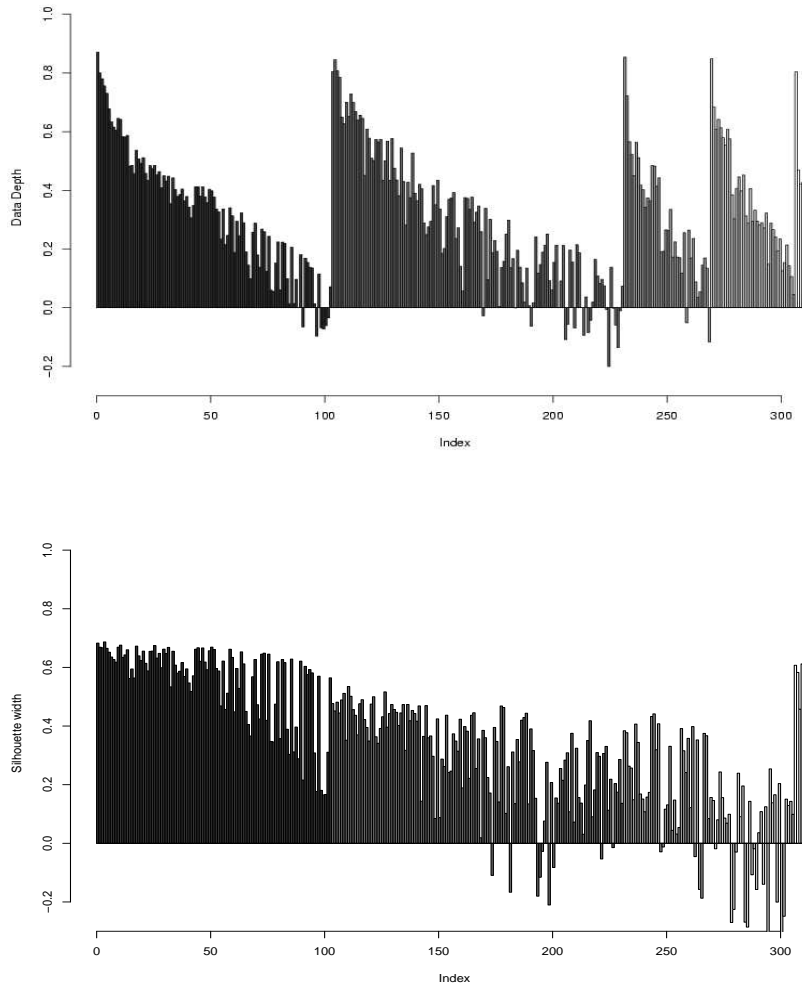


Fig. 3. Top: *ReD* - Relative Data Depth of the SCI data. Bottom: Silhouette widths of the SCI data - note cluster 4 with almost all negative silhouette widths despite it being clearly separated from the other clusters (cmp table 1.)

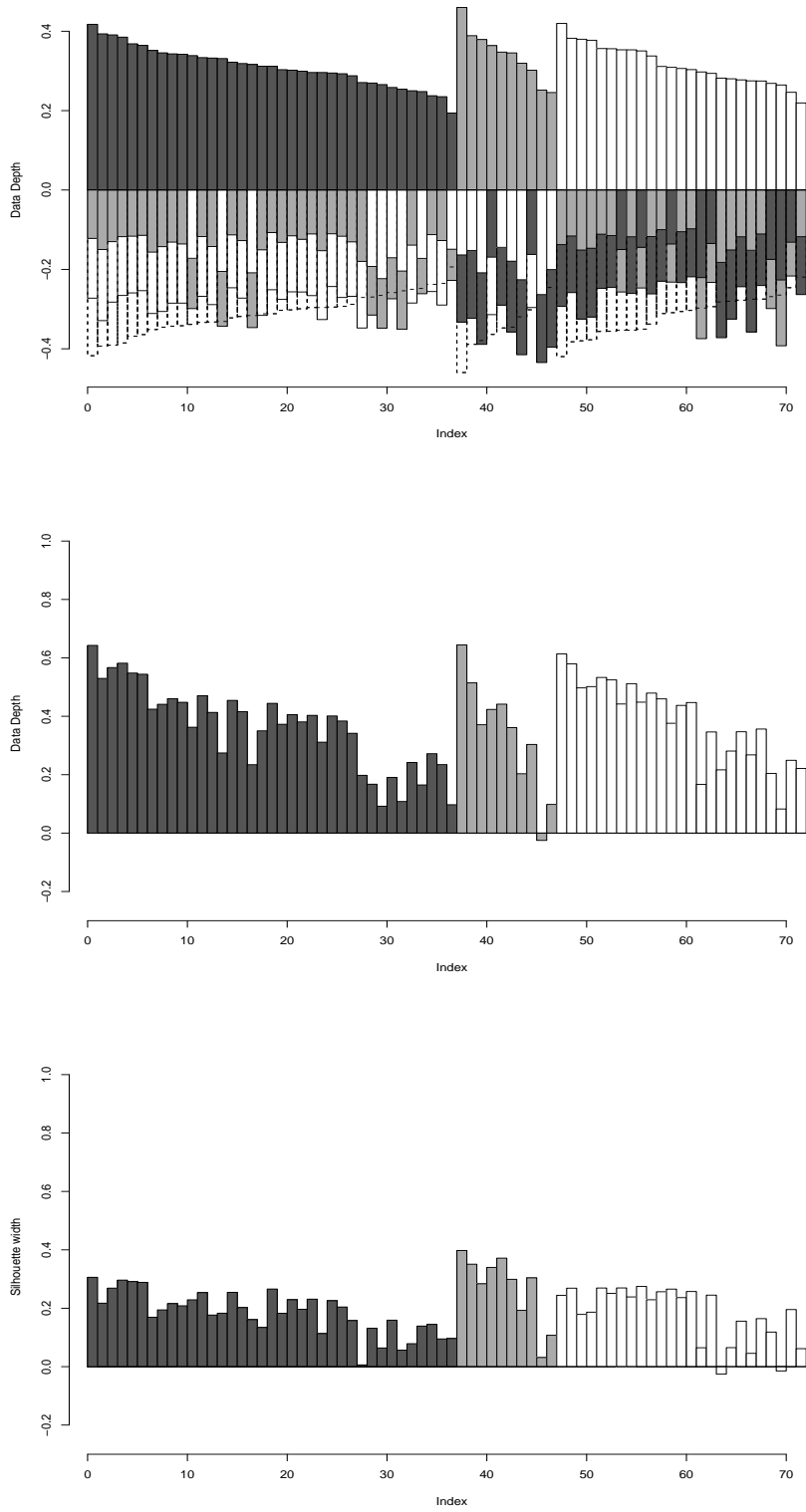


Fig. 4. Top: Data Depth Plot - Leukemia data (AML,ALL-T,ALL-B) Middle: *ReD* - Relative Data Depth. Bottom: Silhouette widths.

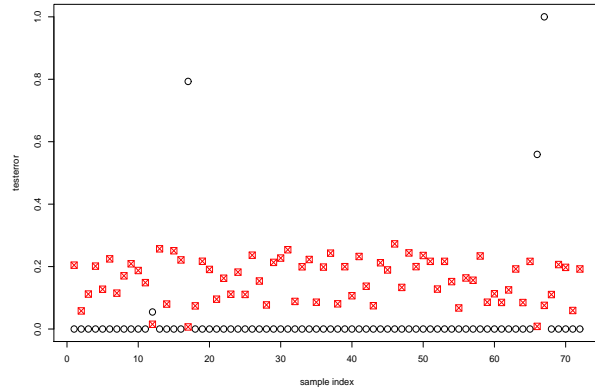


Fig. 5. Testerror rates (open black circles) and ReD (red boxes) of the 72 leukemia samples, across 500 cross validation sets. Observations that are frequently misclassified have near 0 ReD values.

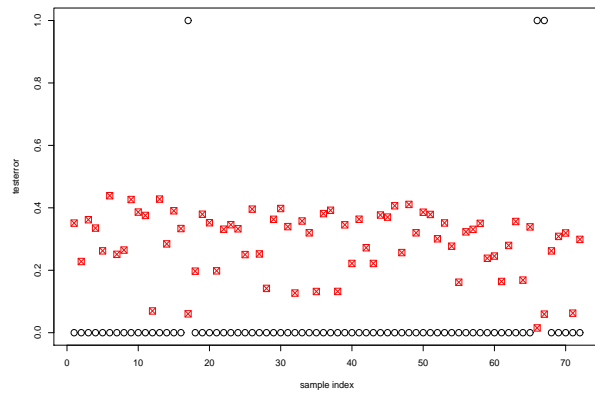


Fig. 6. Testerror rates (open black circles) and Sil (red boxes) of the 72 leukemia samples, across 500 cross validation sets. Observations that are frequently misclassified have near 0 Sil values.

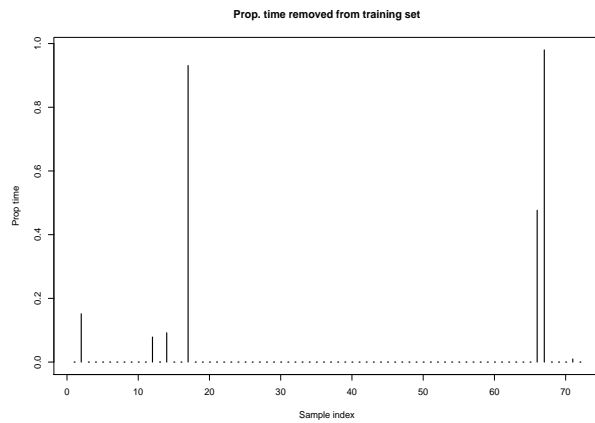


Fig. 7. Proportion of times a leukemia sample in the training set was removed by the DDclass-CV method. Observations 2, 12, 17, 66 and 67 were frequently removed.

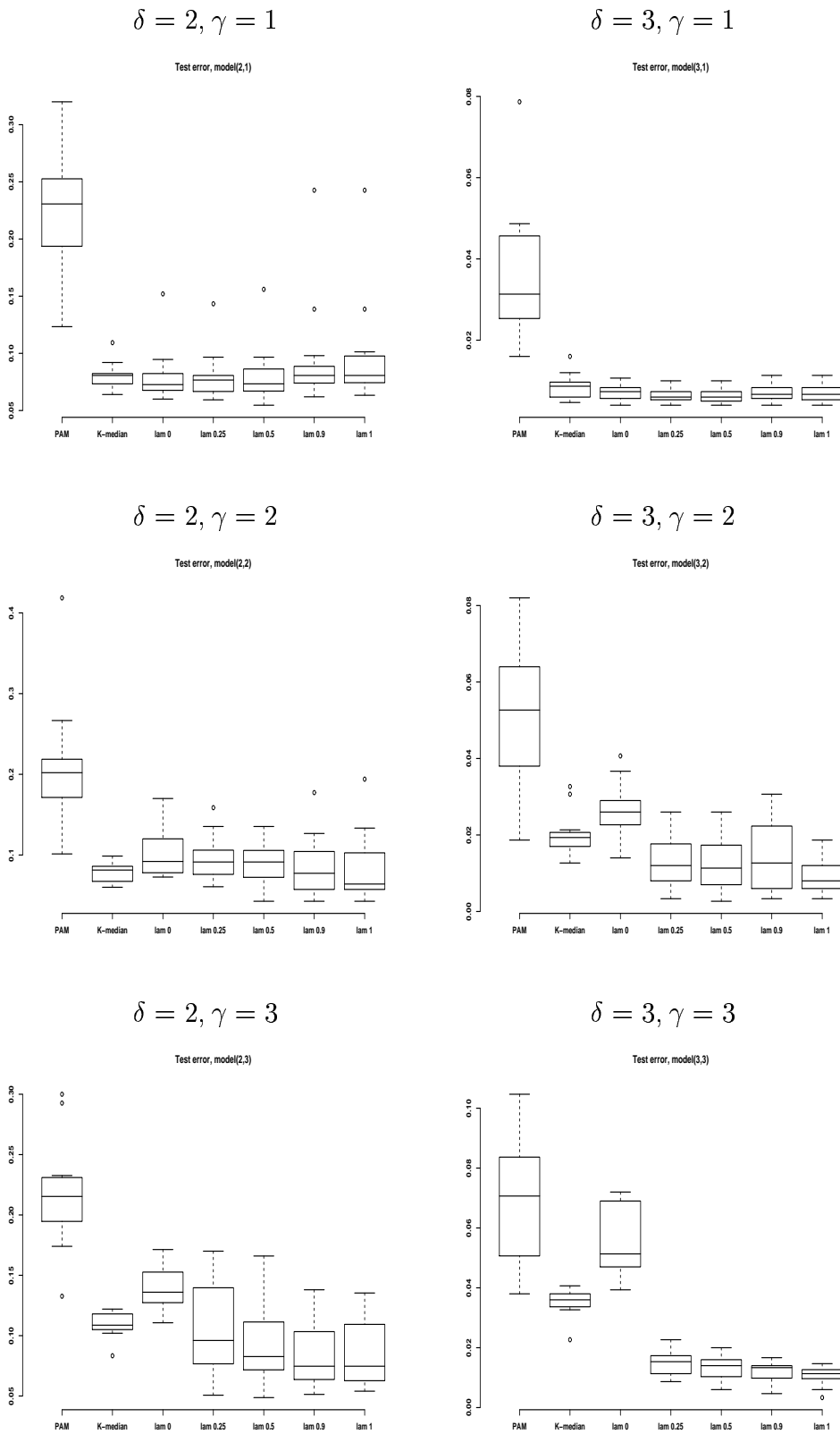


Fig. 8. Clustering results - Test error rates. Left Column: $\delta = 2$. Right column: $\delta = 3$. Row 1-3: $\gamma = 1 - 3$. In each figure, boxplots in the following order: PAM, K-median, DDclust ($\lambda = 0, .25, .5, .9, 1$).

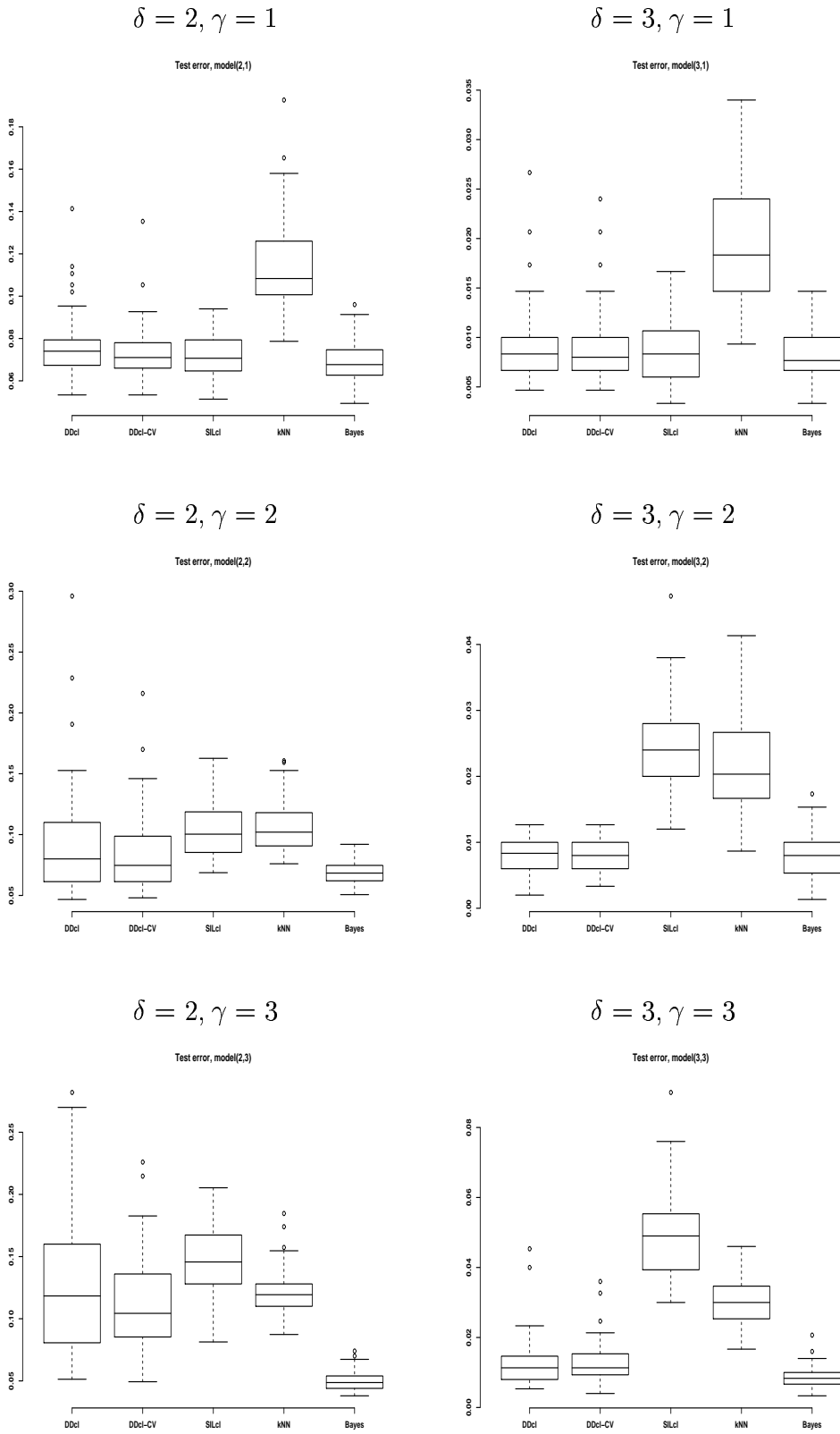


Fig. 9. Classification - Test error rates. Left Column: $\delta = 2$. Right column: $\delta = 3$. Row 1-3: $\gamma = 1 - 3$. In each figure, boxplots in the following order: DDclass, DDclass-CV, SILclass, kNN, Bayes rule (DLDA or DQDA).

References

- [1] U. Alon, N., Barkai, D.A., Notterdam, K., Gish, S., Ybarra, D., Mack, A.J., Levine, *Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays*, Proc. Natl. Acad. Sci, (1999) **96**:6745-6750
- [2] Christmann, A. (2002) *Classification Based on the SVM and on Regression Depth*. Statistical data analysis based on the L1norm and related methods. Birkhauser, Statistics for industry and technology. Y. Dodge editor.
- [3] S. Dudoit, J. Fridlyand, T. Speed. *Comparison of discrimination methods for the classification of tumors using gene expression data*. Journal of the American Statistical Association, **97** (2002), 77-87.
- [4] S. Dudoit, J. Fridlyand. *Application of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method*. Technical report 600 (2001), Department of Statistics, UC Berkeley.
- [5] GeneSpring - Silicon Genetics.
<http://www.sigenetics.com/Products/GeneSpring/>
- [6] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. *Molecular classification of cancer: class discovery and class prediction by gene expression monitoring*. Science, **286** (1999), 531-537
- [7] T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Straudt, W. C. Chang, D. Botstein, P. Brown. *Gene shaving as a method for identifying distinct sets of genes with similar expression patterns*. Genome Biology, **1(2)** (2000), 1-21
- [8] T. Hastie, R. Tibshirani, D. Botstein, P. Brown. *Supervised Harvesting of Expression Trees*. Technical report (2000), Department of Statistics, Stanford University.
- [9] R. Jörnsten, Y., Vardi, and C-H. Zhang *A Robust Clustering Method and Visualization Tool Based on Data Depth*, Statistical data analysis based on the L1norm and related methods. Birkhauser 2002, Statistics for industry and technology. Y. Dodge editor.
- [10] L. Kaufman, and P. J. Rousseeuw. *Finding Groups in Data: An introduction to cluster analysis*. (1990) Wiley, New York.
- [11] M. van der Laan, K. Pollard, J. Bryan. *A New Partitioning Around Medoids Algorithm*. Technical report (2002), Division of Biostatistics, UC Berkeley.
- [12] R. Liu, J. Parelius, K. Singh, *Multivariate analysis by data depth: descriptive statistics, graphics and inference (with discussion)*. Annals of Stat., **27**, 783-858, 1999.

- [13] J. Luo, D. Duggan, Y. Chen, J. Sauvageot, C.M. Ewing, M. Bittner, J.M. Trent, W.B. Isaacs, *Human prostate cancer and benign prostatic hyperplasia: molecular dissection by gene expression profiling*, Cancer research (2001) **61**:4683-4688
- [14] G. J. Sullivan. *Efficient Scalar Quantization of Exponential and Laplacian Random Variables*, IEEE trans. on information theory (1996), **42(5)**, p. 1365-1374.
- [15] R. Tibshirani, G. Walther, and T. Hastie. *Estimating the number of clusters in a dataset via the gap statistic*. JRSSb (Statistical Methodology) (2001) **63**, 2, 411
- [16] R. Tibshirani, G. Walther, D. Botstein, and P. Brown. *Cluster validation by prediction strength* Technical report (2001), Stanford University, Department of Biostatistics.
- [17] Y. Vardi, and C-H. Zhang. *The multivariate L_1 -median and associated data depth*. Proceedings of the National Academy of Sciences, **97** (2000), 1423-1426.
- [18] M. West, J. R. Nevins, J. R. Marks, R. Spang, C. Blanchette, H. Zuzan. *DNA microarray data analysis and regression modeling for genetic expression profiling*. Preprint (2001), Department of Statistics (Duke Univ).