Journal of Classification
https://doi.org/10.1007/s00357-019-09338-0

The δ-Machine: Classification Based on Distances Towards Prototypes

Beibei Yuan[1] · **Willem Heiser**[1] · **Mark de Rooij**[1]

Published online: 22 August 2019

Abstract

We introduce the δ-machine, a statistical learning tool for classification based on (dis)similarities between profiles of the observations to profiles of a representation set consisting of prototypes. In this article, we discuss the properties of the δ-machine, propose an automatic decision rule for deciding on the number of clusters for the K-means method on the predictive perspective, and derive variable importance measures and partial dependence plots for the machine. We performed five simulation studies to investigate the properties of the δ-machine. The first three simulation studies were conducted to investigate selection of prototypes, different (dis)similarity functions, and the definition of representation set. Results indicate that we best use the Lasso to select prototypes, that the Euclidean distance is a good dissimilarity function, and that finding a small representation set of prototypes gives sparse but competitive results. The remaining two simulation studies investigated the performance of the δ-machine with imbalanced classes and with unequal covariance matrices for the two classes. The results obtained show that the δ-machine is robust to class imbalances, and that the four (dis)similarity functions had the same performance regardless of the covariance matrices. We also showed the classification performance of the δ-machine compared with three other classification methods on ten real datasets from UCI database, and discuss two empirical examples in detail.

Keywords Dissimilarity space · Nonlinear classification · The Lasso

1 Introduction

Classification is an important statistical tool for various scientific fields. Examples of applications in which classification questions arise include early screening for Alzheimer's disease, detecting of spam in a mailbox, and determining whether someone is creditworthy for a mortgage. Many classification techniques are currently available, most of which base their classification rules on a set of predictor variables. Traditional classification techniques include logistic regression and discriminant analysis (see Agresti (2013)). The past two decades have seen an increase in the popularity of estimation and learning methods that

✉ Beibei Yuan
b.yuan@fsw.leidenuniv.nl

[1] Institute of Psychology, Leiden University, Leiden, The Netherlands

use basis expansion. Friedman et al. (2009) summarized a host of procedures dealing with basis expansion and/or ensemble methods for building a classification rule.

In daily life, human beings constantly receive input that needs to be categorized into distinct groups or classes. If, for example, we see something above our head, we look at it and instantly determine whether it is a plane, a bird, a fly, or a shooting star. The accuracy of these categorizations is very high, and we are seldom mistaken. There is a large body of literature investigating the way in which we (humans) categorize (for an overview, see Ashby 2014). Although there are different theories, most are defined on the concept of similarity or dissimilarity, i.e., the dissimilarity of the perceived observation from either *exemplars* or *prototypes*. In the theory about categorization, an exemplar is usually defined as an actually existing entity, whereas a prototype is defined as an abstract average of the observations in a category.

In line with the human potential for categorization, we propose the δ-machine, using dissimilarities as the basis for classification. From the available predictor variables we compute a dissimilarity (or similarity) matrix using a (dis)similarity function, which serves as the input for a classification tool. The classification is based in the dissimilarity space (Duin and Pekalska 2012), i.e., the dissimilarities take the role of predictors in classification techniques. By changing this basis, it is possible to achieve nonlinear classification in the original variable space. The classification rule we envision is based on the dissimilarity of a new case from these exemplars or prototypes. A classification rule of this kind is in line with the way we humans recognize objects in daily life and therefore might provide highly accurate classifications.

Scaling of predictor variables influences the dissimilarities; for this reason, scaling influences the classification performance. Many studies have been conducted to investigate different standardization methods (see e.g., Cooper and Milligan 1988; Steinley 2004; Vesanto 2001; Schaffer and Green 1996; Cormack 1971; Fleiss and Zubin 1969), but the conclusions were inconsistent. In sum, the performance of different standardization methods varies in relation to different types of data, different algorithms, standardization within or without clusters, and perhaps other factors. In all our applications, we use the z-score method, i.e., the test set was standardized by the corresponding mean and variance from the training set.

In this paper, we put forward the δ-machine approach to classification and prediction. We define the δ-function that computes dissimilarities between profiles of observations on the predictor variables, and classification is based on these dissimilarities. We propose variable importance measures and partial dependence plots for the δ-machine, and a new decision rule for deciding on the number of clusters for the K-means method. Furthermore, we show relationships with support vector machines (Cortes and Vapnik 1995) and kernel logistic regression (Zhu and Hastie 2012). In Section 4 and 5, we show a pilot study, five simulation studies, and application examples. In Section 6, finally, we discuss the results obtained and suggest some avenues for future development.

2 The δ-Machine

2.1 The Dissimilarity Space

Suppose we have data for a set of observations representing the training set, $i = 1, \ldots, I$. For each of these observations, we have measurements on P variables $X_1, \ldots, X_p, \ldots, X_P$ called a *row profile*. The measurements are denoted by lowercase letters, i.e., x_{i1} represents

the measurement for observation i on variable or feature 1. For every observation in the training set, a response Y is available which assigns observation i to one of $C + 1$ classes, i.e., $y \in \{0, 1, \ldots, C\}$. In the remainder of this paper, the focus will be on the case where $C = 1$.

Besides the training set, there is also a representation set indexed by $r = 1, \ldots, R$. This representation set consists of either exemplars or prototypes. As in the training set, the elements in the representation set are characterized by measurements or values on each of the P variables. For the representation set, no information concerning the response variable is necessary.

Finally, a test set is available indexed by $t = 1, \ldots, T$. For the test set, there are also measurements on the P variables and the response variable Y. The goal of the test set is to verify the accuracy of the classification rule. Instead of working with a training set and test set, we could use a cyclic algorithm, like 10-fold cross-validation, where parts of the data change roles.

Let us define a dissimilarity function $\delta(\cdot)$ that takes as input two profiles of equal length and returns a scalar d_{ir}, i.e.

$$d_{ir} = \delta(\mathbf{x}_i, \mathbf{x}_r), \tag{1}$$

where $\mathbf{x}_i$ is the vector which contains the row profile with the measurements for observation i on the P variables. The elements d_{ir} can be collected in the $I \times R$ matrix $\mathbf{D}$. Rows of this matrix are given by

$$\mathbf{d}_i = [d_{i1}, d_{i2}, \ldots, d_{iR}]^{\mathrm{T}} \tag{2}$$

collecting the dissimilarities of observation i towards the R exemplars/prototypes (i.e., an R vector).

Table 1 presents four examples of (dis)similarity function $\delta(\cdot)$. Unlike the Euclidean distance and the squared Euclidean distance, the Gaussian decay function and the Exponential decay function lead to similarity values. The Gaussian decay function and the Exponential decay function are commonly used in psychophysics (Nosofsky 1986). De Rooij (2001) used these functions to model probabilities in longitudinal data.

In multivariate analysis, the matrix $\mathbf{X}$ with elements x_{ip} defines a P-dimensional space, usually called the feature or predictor space. Similarly, the matrix $\mathbf{D}$ with elements d_{ir} defines an R-dimensional dissimilarity space (Pekalska and Duin 2005) or dissimilarity feature space. Note that the observations in this dissimilarity space occupy only the "positive" orthant of the R-dimensional space, since dissimilarities are positive by definition.

2.2 The Representation Set

In Section 2.1, we defined dissimilarities between the observations and a representation set. There are several choices for this representation set. One choice is simply to define the representation set as being equal to the training set. With this choice, matrix $\mathbf{D}$ is of size $I \times I$. However, this might be problematic for some classification tools to use in the next

Table 1 Four (dis)similarity functions

Reference	Function
The Euclidean distance	$\delta(\mathbf{x}_i, \mathbf{x}_r) = d_{ir} = \sqrt{\sum_{p=1}^{P}(x_{ip} - x_{rp})^2}$
The squared Euclidean distance	$\delta(\mathbf{x}_i, \mathbf{x}_r) = d_{ir}^2$
The Exponential decay function	$\delta(\mathbf{x}_i, \mathbf{x}_r) = \exp(-\frac{1}{P} \cdot d_{ir})$
The Gaussian decay function	$\delta(\mathbf{x}_i, \mathbf{x}_r) = \exp(-\frac{1}{P} \cdot d_{ir}^2)$

step, because **D** could be a singular matrix. Another choice is to randomly select a subset of the training set and use this random selection as the representation set. If R is the number of randomly selected observations, the size of matrix **D** is $I \times R$. Apart from the random selection procedure, systemic procedures can be applied to build a representation set. K-medoids clustering is a clustering method which breaks observations into clusters. The resulting medoids are chosen as the representation set. We will apply the partitioning around medoids (PAM) algorithm (Kaufman and Rousseeuw 1990) for K-medoids clustering. K-medoids clustering minimizes a sum of distances or dissimilarities, the unexplained part of the data scatter. In contrast, the evaluation criterion of clusters can be switched from the unexplained part to the explained part of the data scatter. Mirkin (2012) showed that the explained part can be written in terms of the summary inner product. The most contributing observation is the nearest in the inner product to the center. The representation set can be obtained from the clustering method based on these inner products (IP).

The above four choices, the entire training set, random selection of the training set, PAM, and the IP method, lead to dissimilarities between the observations and exemplars, i.e., actually existing entities. Another possible choice is to use prototypes. To find prototypes, clustering methods can be used. Two well-known clustering algorithms are K-means (MacQueen 1967) and probabilistic distance clustering (Ben-Israel and Iyigun 2008). Al-Yaseen et al. (2017) applied a modified K-means method to each outcome category to reduce the number of observations in the training set. The new observations are the averages of observations within clusters, called *high-quality observations*, which then are used for training the classifier.

Similarly, we implement a K-means clustering method to find a representation set defined on prototypes. The algorithm is applied on each outcome category of the response variable Y. For example, for a dataset with two classes, k_0 and k_1 clusters are obtained for the two classes, respectively. Then the representation set consists of $R = k_0 + k_1$ prototypes, so the size of the dissimilarity matrix **D** is $I \times R$.

For the K-means method, the number of clusters needs to be determined by users, while we propose to use an automatic decision rule, similar to the decision rule in Mirkin (1999) and Steinley and Brusco (2011). The number of clusters in each outcome category is determined by the percentage of explained variance compared with a user-specific threshold. The percentage of explained variance v_e is calculated by

$$v_e = \frac{\text{The total sum of squares} - \text{Total within-cluster sum of squares}}{\text{The total sum of squares}}. \tag{3}$$

In the algorithm, one needs to specify a threshold to apply the automatic decision rule. Afterwards, the number of clusters in each class is automatically determined. The steps of the K-means algorithm applying the automatic decision rule are shown in the below **Algorithm**.

Algorithm 1 K-means algorithm.

Input: the predictor variables matrix **X**, the response vector **y**, and a *threshold*
Output: the representation set defined on prototypes
Step 1: For $c = 0$ to C, set $K = 1$:
(a) For the observations belong to class c, do K-means clustering.
(b) Calculate the percentage of explained variance, v_e. Do
(b 1) If $v_e \geq$ *threshold*, K is the number of prototypes in class c. Else
(b 2) $K = K + 1$, go to (a)
Step 2: The representation set consists of the corresponding prototypes in each class.

We put the automatic decision rule or the criterion and the percentage of explained variance, under the context of the δ-machine. Mirkin (1999) used the similar idea as one possible stopping rule for algorithm separate-conquer clustering (SCC). Steinley and Brusco (2011) used the similar idea to construct the lower bound technique (LBT) to choose the number of clusters. Although the idea is similar, it is applied under different circumstances. For the δ-machine using the K-means algorithm, we use the K-means algorithm on each class of the data, and the algorithm finds prototypes simultaneously. Whereas algorithm SCC searches prototypes one by one, and the algorithm is applied on the entire data. For the method proposed by Steinley and Brusco (2011), the user still needs to specify the maximal $K_{\max}$. As the data are partitioned into $K = 2, \cdots, K_{\max}$, and LBT_K is computed correspondingly. The chosen K is the one corresponding to the lowest LBT_K.

A final option for deriving prototypes is to ask content-specific experts to name a few typical profiles called archetypes, and ask for values on the variables for these archetypes. When the expert comes up with R archetypes, the size of matrix $\mathbf{D}$ is $I \times R$.

2.3 Dissimilarity Variables in Classification

Let us denote by π the probability that Y is in one of the classes, i.e., $\pi = Pr(Y = 1)$. This probability for a specific observation will depend on the R dissimilarities from members of the representation set, which will be denoted by $\pi(\mathbf{d}_i) = Pr(Y = 1|\mathbf{d}_i)$. Although many different classification techniques can be used, our focus will be on logistic regression. That is, we define this probability to be

$$\pi(\mathbf{d}_i) = \frac{\exp\left(\alpha + \mathbf{d}_i^{\mathrm{T}}\boldsymbol{\beta}\right)}{1 + \exp\left(\alpha + \mathbf{d}_i^{\mathrm{T}}\boldsymbol{\beta}\right)}, \tag{4}$$

where $\boldsymbol{\beta}$ represents an R vector of regression coefficients. The model can be fitted by minimizing the binomial deviance:

$$\text{Deviance} = -2\sum_i \left[y_i \log \pi(\mathbf{d}_i) + (1 - y_i)\log(1 - \pi(\mathbf{d}_i))\right]. \tag{5}$$

If the representation set consists of the entire training set, it is necessary to select dissimilarity variables, because otherwise the logistic regression would be indeterminate. With some choices, i.e., random selection of training set or clustering methods, a standard logistic regression could be carried out. However, there is a danger of overfitting. To avoid overfitting, we implement two exemplar/prototype selection method: the Lasso (Tibshirani 1996; Friedman et al. 2010b) and forward-stepwise selection based on the lowest AIC. Selected exemplars (or prototypes) will be called *active* exemplars (or prototypes). A comparison of these two selection methods will be shown in Section 4.

We define the δ-machine as the method that uses the (dis)similarity function to compute a (dis)similarity matrix from the predictor matrix and then uses this dissimilarity matrix in logistic regression.

2.4 Variable Importance Measure

In a logistic regression with dissimilarity variables, it is impossible to see the value of the original variables, all we obtain are regression weights for dissimilarities from certain members of the representation set. Nevertheless, often the focus lies on the value of the original variables. Therefore, in this subsection, we define variable importance measures.

To verify the importance of a predictor variable X_p, a permutation test approach is used. One such example is the importance measure for random forests (Breiman 2001). In this approach, the values of variable X_p are randomly shuffled over observations in the training set. Once the variable is shuffled the dissimilarities are computed by applying the δ-function, and the dissimilarities are used in the logistic regression.

The variable importance measure is the deviance obtained using the original data minus the deviance of the model fitted used in the permuted variable. By repeating the whole procedure a large number of times, a mean decrease in deviance can be computed (together with a standard deviation or a 95% interval). The larger the average in decrease the more important the variable is for predictive performance. The procedure is repeated for each predictor variable, and the results are plotted in a variable importance plot.

2.5 Partial Dependence Plots

Although models are interpreted in terms of dissimilarity from one or more exemplars or prototypes, it might be of interest to see the relationship between predictor variables and the response. For this purpose, we use partial dependence plots (Friedman 2001; Berk 2008) representing a conditional relationship between a predictor variable and the response variable, keeping other variables at fixed values. To obtain the partial dependence plot for variable X_p, first, obtain the unique values of this variable and collect them in the vector $\mathbf{v}$ with length V_p. Second, define the V_p auxiliary matrices $\tilde{\mathbf{X}}$ containing the original variables and in the column of the pth variable, define a single unique value from the vector $\mathbf{v}$. Finally, for each of the auxiliary matrices, make a prediction of the response probability and average these over the observations to obtain a final predicted value. The variability in this prediction will be represented by the standard error.

3 Theoretical Comparisons with Support Vector Machines and Kernel Logistic Regression

Support vector machines have become popular methods for classification. Support vector machines work by trying to find a hyperplane that separates the two classes. The distance from the hyperplane to the observations is called the margin. The methodology seeks the hyperplane with the largest margin. The points closest to the hyperplane are called the support vectors. Support vector machines extend the idea of enlarging the original variable space by using kernels, where the dimension of the enlarged space can be very large (Friedman et al., 2009, pp.423–429; James et al. 2013, pp.350–354). The major improvement in support vector machines came through the kernel trick, which is to avoid working in the enlarged space, one only involves the predictor variables via inner products. It was recognized that much better separation was possible by using kernels rather than the original variables. Kernels can take different forms, but one important kernel is the radial basis kernel. The radial basis kernel is defined in terms of the squared Euclidean distance as

$$K(\mathbf{x}_i, \mathbf{x}_r) = \exp(-\gamma \times d_{ir}^2). \tag{6}$$

Although the hyperplanes are linear in the enlarged space, in the original variable space, this leads to nonlinear classification boundaries.

If the idea of seeking the largest margin is applied in the original variable space, it is called the support vector classifier or the linear support vector machine. Linear support vector machines are closely related to logistic regression procedures. It has been recognized

(Friedman et al. 2009) that the same kernels as used in SVMs may be used in logistic regression, i.e., a basis expansion is applied and then logistic regression is applied. A logistic regression based on the radial basis kernel is quite close to our logistic regression based on dissimilarities. In our approach, we use the dissimilarities directly, whereas kernel logistic regression (KLR) uses the exponent of minus the scaled (by γ) squared Euclidean distance.

The Gaussian decay function is quite similar to the radial basis function in support vector machines. The only thing that differentiates these two functions is that the radial basis function has a tuning parameter, while the Gaussian decay function has a fixed value $1/P$.

4 Simulation Studies

4.1 The New Decision Rule of the *K*-Means Method: a Pilot Study

In Section 2.2, we proposed an automatic decision rule of deciding on the number of clusters for the K-means method. An artificial problem is generated to show the usefulness of the automatic decision rule in terms of the predictive performance. The problem has two classes. Each of the classes consists of four 2-dimensional Gaussians. The centers are equally spaced on a circle around the origin with radius r (see Fig. 1). The radius r controls if clusters have overlap or not. With a larger radius, there is less overlap. In our study, we set the radius $r \in \{2\sqrt{2}, 8\}$, which stands for the data with or without overlap (left and right sides of Fig. 1).

In this pilot study, the representation set contains either prototypes or exemplars. The prototypes are selected using the K-means method with the automatic decision rule, when we choose the number of prototypes to be the number of clusters that can explain at least v_e of the variance in each outcome category. Two levels of the percentage of explained variance v_e are chosen: 0.5 and 0.9. To avoid a local minimum, we use 100 random starting points. With the resulting cluster statistics, Euclidean distances are computed between the observations in the training set and the estimated cluster means (prototypes). The exemplars are selected by three different methods. The first method is simply to treat the whole training set as the representation set, we calculate Euclidean distances between the training set itself resulting an $I \times I$ distance matrix, where I is 500 in this case. The second method is to use partitioning around medoids (PAM) (Kaufman and Rousseeuw 1990) to select exemplars on each class. We set the number of clusters from two to ten, the optimal number of medoids

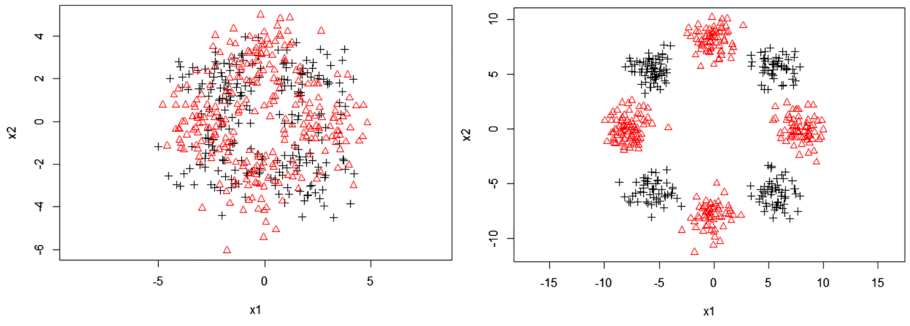


Fig. 1 Examples of the artificial problem in the pilot study. The left one is the radius $r = 2\sqrt{2}$, and the right one is $r = 8$. The triangle and the cross denote the observation classified as class 0 and class 1, respectively

for each class is chosen based on the optimum average silhouette width (Rousseeuw 1987). With the selected medoids from the two classes, Euclidean distances from the observations in the training set and the selected medoids are computed. Besides PAM, we also use the clustering method based on the inner product (IP) (Mirkin 2012). To reduce computational time, we use the partition from PAM, and set them as the initial setting for the IP method.

The Lasso is performed resulting in active prototypes or exemplars. For each condition, 100 replications will be simulated. With each dataset, a test set is generated with size 1000. The misclassification rate (MR) in the test set is used to evaluate the predictive performance. The misclassification rate is the percentage of misclassified observations. For this pilot study, the MR will be calculated for each condition, and the average MRs will be presented in the results table.

To perform this study, we use the open source statistical analysis software R (R Core Team 2015). PAM is implemented in the `cluster` package (Maechler et al. 2013). The Lasso is implemented in the `glmnet` package (Friedman et al. 2010a).

For both cases, $r \in \{2\sqrt{2}, 8\}$, the δ-machine using a higher level of the percentage of explained variance $v_e = 0.9$ had a lower MR than using the threshold $v_e = 0.5$, see Table 2. Although $v_e = 0.5$ resulted in less active prototypes than $v_e = 0.9$ in both cases, MR increased sharply from $v_e = 0.5$ to $v_e = 0.9$.

The δ-machine using the K-means method ($v_e = 0.9$) and using the whole training set had the same MR, which indicates that the K-means method using the automatic decision rule can successfully find high-quality observations. Moreover, the δ-machine using the K-means method leads to a smaller number of active prototypes than that of the one using the entire training set. Therefore, only a small number of high-quality prototypes is enough to train a good classifier. For the data without overlap, the δ-machine using the K-means method ($v_e = 0.9$) found one prototype for each cluster precisely. This result supports the usefulness of the automatic decision rule of the K-means method.

The δ-machine using PAM and using the K-means method ($v_e = 0.9$) had the same classification performance in terms of MR for both the data with and without overlap. Using PAM, eight active exemplars were successfully found in both data cases, while using the K-means method, we found eight active prototypes only for the data without overlap. Thus, in this study, the δ-machine using PAM resulted in sparser models than using the K-means method. The IP method did not perform well for the data with overlap $\left(r = 2\sqrt{2}\right)$, similar to the δ-machine using prototypes ($v_e = 0.5$). But for the data without overlap ($r = 8$), the IP method had satisfactory results.

Table 2 The average misclassification rate (MR) and the average number of active prototypes or exemplars

Method	$r = 2\sqrt{2}$		$r = 8$	
	MR	Active prototypes/exemplars	MR	Active prototypes/exemplars
$v_e = 0.5$	0.46 (0.03)	5.01 (1.53)	0.31 (0.07)	5.00 (0.90)
$v_e = 0.9$	0.29 (0.02)	15.96 (1.81)	0.00 (0.00)	8.00 (0.00)
The training set	0.29 (0.02)	66.14 (21.33)	0.00 (0.00)	33.97 (16.61)
PAM	0.29 (0.02)	8.00 (0.00)	0.00 (0.00)	8.00 (0.00)
Inner product	0.45 (0.04)	4.30 (2.43)	0.02 (0.04)	7.97 (0.17)

Standard deviations are shown within brackets

Table 3 Summary of independent factors and levels for simulation study 1–3. The last row is the extra factor for simulation study 3

Factor	#	Level
Artificial problem	3	Two norm, four blocks, Gaussian ordination
The training size	2	50, 500
The number of predictor variables	3	2, 5, 10
The number of irrelevant variables	3	0, 2, 5
The representation set (only for simulation study 3)	4	The training set, $v_e = 0.5$, $v_e = 0.9$, PAM

4.2 Five Simulation Studies

In simulation study 1, we discuss two techniques for selecting active exemplars: the Lasso, and forward-stepwise selection based on AIC. The comparison of the Lasso and forward selection based on AIC, within the δ-machine is made. Three types of artificial datasets are generated, each varying in the following factors: the training size, the number of predictor variables, and the number of irrelevant variables (Table 3). The sizes of a training set are 50 and 500, corresponding to a small and large dataset. To explore the effects of the numbers of predictor variables, we choose 2, 5, and 10. These are the relevant variables which have a relation with the response variable. The irrelevant variables, on the contrary, are sampled from the standard Gaussian distribution and are included in the data as noise.

Simulation study 2 investigates the difference between the δ-machine with the Euclidean distance and three (dis)similarity functions mentioned in Table 1. The same factors are used as in Simulation study 1.

Simulation study 3 is to investigate the selection of the representation set. The research question now arises, how a representation set should be selected out of the whole training set to guarantee a good gradeoff between the accuracy and the complexity of the model. In Section 4.1, a pilot study showed that the K-means method using the automatic decision rule and partitioning around medoids (PAM) (Kaufman and Rousseeuw 1990) had a good performance, but the inner product clustering method (IP) (Mirkin 2012) did not perform well. Because of the poor performance of the IP method, in simulation study 3, we do not include it. In this study, we go further, which considers more factors as shown in Table 3 than the pilot study.

The data generated from the above three simulation studies all have balanced classes. Therefore, in simulation study 4, we investigate the performance of the δ-machine for imbalanced data. Japkowicz and Stephen (2002) considered the class imbalance problem on three different dimensions: the degree of concept complexity, the size of the training set, and the level of imbalance between the two classes. In this simulation study, we only consider the two norm problem, keep the number of predictors as two, the size of the training set is kept fixed at 500. The levels of class imbalance are defined as the ratio of the size of the majority class to that of the minority class. Six levels of class imbalance are considered: {1, 2, 4, 8, 16, 32}, which were inspired by Japkowicz and Stephen (2002). For example, if the level of imbalance is 2, the data have one class of size 333 and the other class of size 167.

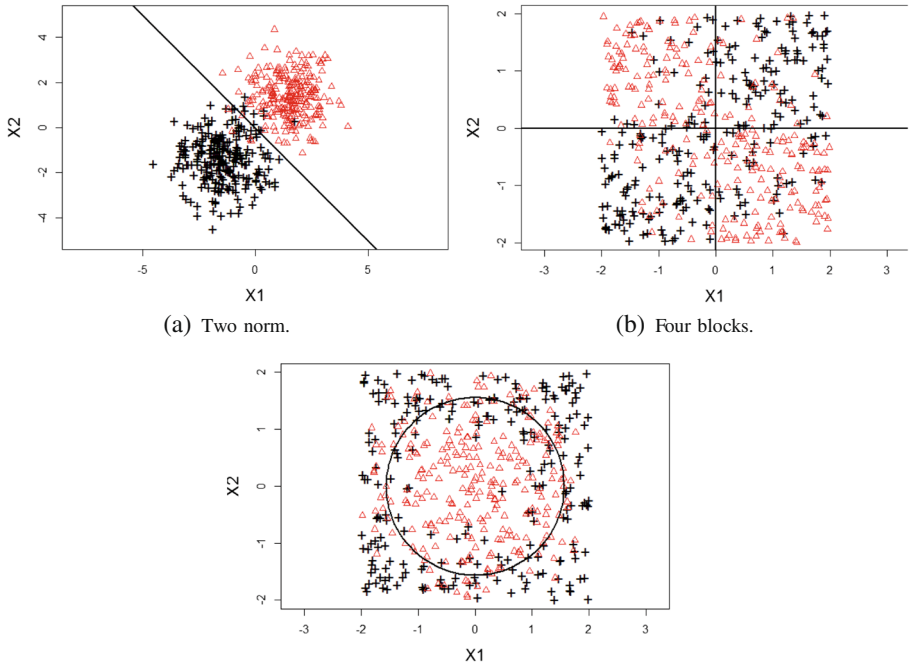
Support vector machines (SVMs) have been shown to be robust to the class imbalance problem, because they base the decision line on a small number of support vectors (Japkowicz and Stephen 2002). Therefore, we consider support vector machines with radial basis kernel (SVM(RBF)) as a reference and compare the performance of the δ-machine to SVM(RBF). The performance of different methods is evaluated by the misclassification

rate (MR). Because the MR is sensitive to the distribution over classes, in this simulation study, two extra criteria will be used: the area under the receiver operating characteristic (ROC) curve (AUC) and the F-measure. The ROC curve is created by plotting the sensitivity against (1−specificity) at various threshold settings (Fawcett 2006). The F-measure is a harmonic mean of precision and sensitivity (Van Rijsbergen 1979).

In simulation study 4, the data from the two norm problem are the two classes drawn from two multivariate Gaussian distribution with different means but the same covariance matrix, the identity matrix (denoted by Σ). In simulation study 5, we consider the situation where the two classes have different means and covariance matrices. Similar to the way of generating imbalanced data, we consider the ratios of standard deviation of the two predictors from the first class to the second class. Let us keep the covariance matrix of the second class as the identity matrix. The covariance matrices for the first class are Σ and 4Σ. As in simulation study 4, we fix the number of predictor variable to 2 and the size of the training set to 500.

In all simulation studies, for each condition, 100 replications will be simulated. With each dataset, a test set of size 1000 is generated using the same characteristics. The misclassification rate (MR) in the test set is used to evaluate the predictive performance, except for the two extra criteria in simulation study 4.

The forward selection based on the AIC logistic regression is implemented in the stepAIC function in the `MASS` package (Venables and Ripley 2002). SVM(RBF) is implemented in the svm function in the `e1071` package (Meyer et al. 2014).



(a) Two norm. (b) Four blocks.

(c) Gaussian ordination.

Fig. 2 Examples of three artificial problems in two dimensions. The triangle and the cross denote the observation classified as class 0 and class 1, respectively. The lines are the decision boundaries in each problem

4.3 Data Descriptions

The three types of artificial problems are given as follows: two norm, four blocks, and Gaussian ordination (Fig. 2).

- `Two norm`: This problem has two classes. The points are drawn from two multivariate Gaussian distributions, and their covariance matrices are equal to the identity matrix. Class 0 is multivariate Gaussian distribution with mean $(a, a, \ldots, a)$ and class 1 with mean $(-a, -a, \ldots, -a)$, $a = 2/\sqrt{P}$. The two classes are linearly separable.
- `Four blocks`: The data consist of two classes. The predictor variables are independent and identically generated from the uniform distribution in the range $(-2, 2)$. For observation i, the product of predictor variables is calculated, which is $\prod_{p=1}^{P} x_{ip}$. According to the product, the rank of observation i in all generated observations is obtained. Then the rank is transformed to a scale between 0 and 1, i.e.,

$$\text{Z.rank} = \frac{\text{rank} - 1}{I - 1}. \tag{7}$$

  After that, Z.ranks are used as the probabilities to draw observed classes from a Bernoulli distribution. Therefore, if an observation has a higher product, it has a higher probability of becoming class 1, and vice versa. Figure 2b illustrates that an observation's class is closely related to whether the product is positive or negative.

  Since the probabilities are used to assign observations to class 0 or class 1 by drawing from a Bernoulli distribution, the noise is generated systematically. Observations with probability around 0.5 have a higher chance of being assigned to the class with the lowest probability.
- `Gaussian ordination`: The variables are independent and identically generated from the uniform distribution in the range $(-2, 2)$. For observation i, the sum of the squares of the predictor variables is calculated, which is $\sum_{p=1}^{P} x_{ip}^2$, and the rank of the value is generated. Again, the ranks are transformed to a scale between 0 and 1. The rescaled ranks, Z.ranks, are taken as the probabilities, and the corresponding outcomes are generated from the Bernoulli distribution using the probabilities. Therefore, a point that is close to the origin of the coordinate plane has a high probability of being assigned to the class 0 (see Fig. 2c).

  The transformed ranks are used as the probabilities rather than using the sum of squares variables, because the sum of squares measure, which is defined by many coordinates, has little difference in the distances between pairs of observations in space (Beyer et al. 1999). In other words, the sum of squared variables of observations do not indiscriminate in a high-dimensional space. When the dimensionality increases, the proportional difference between the farthest observation and the closest observation diminishes.

The baseline error for the two norm problem, the four blocks problem, and the Gaussian ordination problem is 0.02, 0.25, and 0.25 respectively. The term "the baseline error" denotes the percentage of the generated response variable that is inconsistent with the true class, i.e., the class that has the highest probability. Different artificial problems have different ways to determine the true class of an observation. For the two norm problem, according to the generated data, we calculate the densities from the two multivariate Gaussian distributions. Then, the observation is assigned to the class which fits a multivariate Gaussian distribution that has the higher density. This is the true class of the observation in question. If we say an observation is assigned wrongly, then the true class is inconsistent with the

generated y. For the four blocks problem, the true class of an observation is determined by the sign of the product of this observation's predictor variables. For the Gaussian ordination problem, the true class of an observation is determined by whether the Z.rank of this observation is larger than 0.5.

4.4 Result Analysis

As the increase of replications can always achieve significant results, emphasis of the simulation studies is on the effect sizes. To assess the effect size of the factors and their interactions, partial η squared $\left(\eta_p^2\right)$ (Cohen 1973; Richardson 2011) is used. Suppose that there are two independent factors, A and B. The formula of partial η squared for factor A is

$$\eta_p^2 = \frac{SS(A)}{SS(A) + SS(\text{Withingroups})},$$

where $SS(A)$ and $SS(\text{Withingroups})$ are the sum of squares for the effect of factor A and $SS(\text{Withingroups})$ is the sum of squares within the groups. A common rule of thumb is that η_p^2 values of 0.01, 0.06, and 0.14 represent a small, a medium, and a large effect size, respectively (Rovai et al. 2013).

The results obtained from the first three simulation studies will be analyzed using fixed effect models, and statistics for factors and their interactions will be tested. In this paper, for ease of interpretation, we focus on two-way interaction effects. The three artificial problems will be modeled separately. Therefore, for each problem, the model is as follows:

$$\text{MR} = \tau + m + v + iv + s + m \times v + m \times iv + m \times s + v \times iv + v \times s + iv \times s + \epsilon, \quad (8)$$

where "$\times$" represents the interaction between two factors, τ is the intercept, ϵ is the random error, and m, v, iv, s are the method, the number of predictor variables, the number of irrelevant variables, and the training size, respectively. For simulation study 1, the method factor indicates the two variable selection methods. For simulation study 2, the method factor contains the δ-machine with the Euclidean distance and the other three (dis)similarity functions. For simulation study 3, the method factor stands for the four representation set: the training set, the representation set selected by the K-means method with the automatic decision rule ($v_e = 0.5$, $v_e = 0.9$), and the representation set selected by PAM.

For simulation 4 and 5, we do not consider many factors as the first three studies. The ANOVA model is as follows:

$$\text{outcome} = \tau + \text{method} + A + \text{method} \times A + \epsilon, \quad (9)$$

where, in simulation study 4, the outcomes are MR, AUC, and the F-measure, the factor A stands for the level of imbalance factor and the method factor contains the δ-machine using the four (dis)similarity functions and SVM(RBF); and in simulation study 5 the outcome is MR, the method factor contains the δ-machine with the Euclidean distance and the other three (dis)similarity functions, and the factor A stands for the covariance factor.

4.5 Results

4.5.1 Comparison of Two Selection Methods in the δ-Machine

Table 4 shows the misclassification rates (MRs) averaged over all design factors except the method factor. The standard deviations are shown within brackets. For the three artificial problems, the δ-machine using the Lasso outperformed the one using the forward stepwise

Table 4 The average misclassification rates of the δ-machine with two types of variable selection methods

Problem	Lasso	Forward stepwise
Two norm	*0.04* (0.02)	0.05 (0.03)
Four blocks	*0.44* (0.09)	0.46 (0.08)
Gaussian ordination	*0.32* (0.06)	0.34 (0.06)

Best results are set in italics. Standard deviations are shown in brackets

method in terms of MRs. For the three problems, the method factor had a medium effect size, which indicates that the MR decreases if we use the Lasso rather than the forward stepwise method as the selection method in the δ-machine. The other method-related factors had either no effect size or only small effect size (results not shown). In sum, the δ-machine with the Lasso outperformed the forward stepwise method in all three artificial problems. Moreover, the Lasso was also superior to the forward stepwise method in terms of computational time.

4.5.2 Comparison of the Four (dis)Similarity Functions in the δ-Machine

The misclassification rates and their standard deviations (shown within brackets) are given in Table 5. Table 6 gives the estimated η_p^2 for each model component for the three problems. The method-related effect sizes which are larger than or equal to 0.06 are set in italics, indicating at least medium effect sizes.

Generally speaking, for the two norm problem, the δ-machine had quite good results in terms of MRs, which were very low. Although the δ-machine with the Euclidean distance had the lowest MR overall, the effect size of the method (m) was small. Thus, for the linearly separable problem, the δ-machine had a satisfactory performance, and this performance can be marginally changed by using different dissimilarity functions. More specifically, the δ-machine using the Euclidean distance slightly outperformed the other dissimilarity functions.

For the four blocks problem, the δ-machine had very high MRs, especially with the squared Euclidean distance which failed to predict the four blocks problem, as the mean MR was around 0.5. The large effect size of the method factor m shows that changing a dissimilarity function had a large influence on the performance of the δ-machine. The interaction of the method with the number of predictor variables ($m : v$) also had a large effect size. Figure 3a illustrates that, when the data had two predictor variables, the δ-machine with the other three dissimilarity functions performed very similar, whereas the δ-machine

Table 5 The average misclassification rate of the δ-machine with four (dis)similarity functions

Problem	Euc.	Exp.	Gau.	sq.Euc.
Two norm	0.03 (0.02)	0.04 (0.02)	0.04 (0.03)	0.04 (0.02)
Four blocks	0.44 (0.09)	0.44 (0.09)	0.44 (0.09)	0.50 (0.02)
Gaussian ordination	0.32 (0.06)	0.32 (0.06)	0.32 (0.06)	0.36 (0.07)

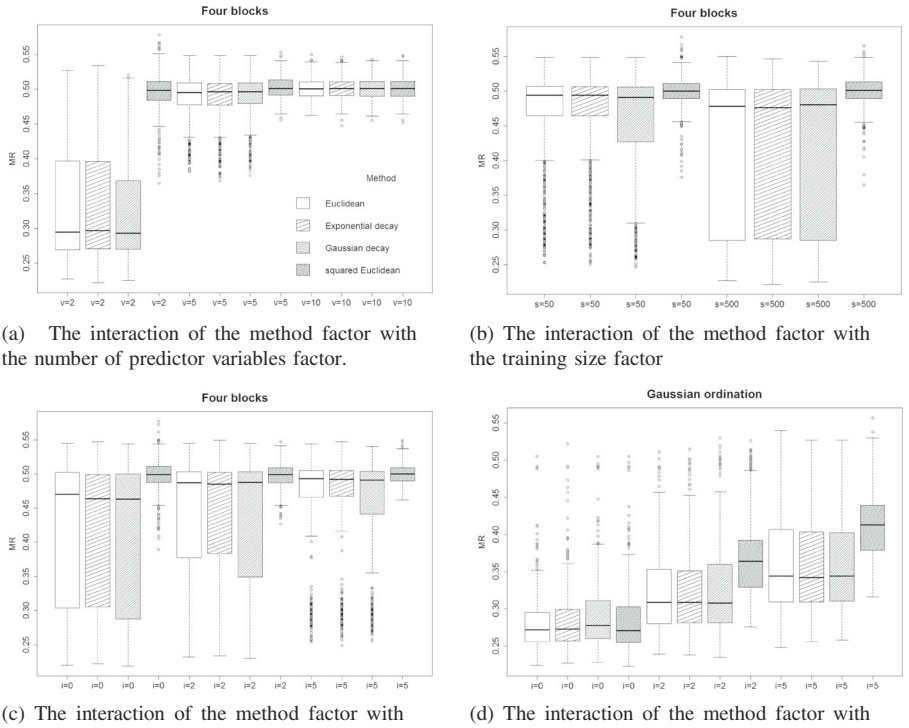
Standard deviations are shown within brackets. *Euc*., *Exp*., *Gau*., and *sq.Euc*., are abbreviations of the Euclidean distance, the Exponential decay, the Gaussian decay, the squared Euclidean distance, respectively

Table 6 Estimated effect size η_p^2 for each model component for the three problems

Problem	m	v	s	iv	$m:v$	$m:s$	$m:iv$	$v:s$	$v:iv$	$s:iv$
TN	0.02	0.00	0.32	0.04	0.00	0.02	0.00	0.00	0.01	0.03
FB	*0.44*	0.79	0.26	0.18	*0.54*	*0.12*	*0.06*	0.27	0.15	0.01
GOM	*0.15*	0.00	0.53	0.55	0.04	0.02	*0.09*	0.00	0.04	0.06

The method-related effect sizes which are larger than or equal to 0.06 are set in italics. *TN*, *FB*, and *GOM* are abbreviations of the two norm, the four blocks, and the Gaussian ordination problems

using the squared Euclidean distance had very high MRs. As the number of predictor variables increases, the differences among the δ-machine with four dissimilarity functions did not exist, for MRs of the four dissimilarity functions were all around 0.5. The large effect of the number of predictor variables v also indicates that the number of predictor variables had a very strong influence on MRs. The interaction between the method and the training size ($m:s$) had a medium effect size. The difference in MRs of the δ-machine with different dissimilarity functions became larger when the training size was larger (see Fig. 3b), but the performance of the δ-machine with the Euclidean distance, the Exponential decay, and the Gaussian decay were very similar, regardless of the size of the training set. Figure 3c illustrates the medium effect size of the interaction between the method and the number of



(a) The interaction of the method factor with the number of predictor variables factor.

(b) The interaction of the method factor with the training size factor

(c) The interaction of the method factor with the number of irrelevant variables factor

(d) The interaction of the method factor with the number of irrelevant variables factor

Fig. 3 The box plots for MRs of the δ-machine with the Euclidean distance and the three (dis)similarity functions. The shading patterns are explained in graph (**a**)

irrelevant variables ($m : iv$). When the data had no irrelevant variables, the δ-machine with the Euclidean distance, the Exponential decay, and the Gaussian decay had lower MRs than the δ-machine with the squared Euclidean distance. As the number of irrelevant variables increases, the difference of MRs for the δ-machine with different (dis)similarity functions gets smaller.

In sum, the squared Euclidean function is not recommended in this problem. The δ-machine with the other three (dis)similarity functions have the similar predictive performance. Therefore, in the case of a problem with two predictor variables, where there is significant interaction between two variables, the δ-machine with the squared Euclidean distance is the last to recommend. In the case of data with more than two predictor variables, the δ-machine fails to predict on this problem.

For the Gaussian ordination problem, the δ-machine with the squared Euclidean distance again had the highest MR. The method factor m had a large effect size, indicating that using different dissimilarity functions had large influence on the MR. The interaction between the method and the number of irrelevant variables ($m : iv$) had a medium effect. Figure 3d reveals the interaction between these two factors. If the data have no irrelevant variables, four dissimilarity functions perform similarly. But if the data have irrelevant variables, the squared Euclidean distance is the last one to choose.

4.5.3 Selection of the Representation Set

From simulation study 2, we found, when the number of predictor variables was larger than two, the δ-machine failed to predict the four blocks problem. Therefore, in this study for the four blocks problem, we made a comparison on the data with only two predictor variables. For the remaining two problems: the two norm problem, the Gaussian ordination problem, we considered all factors mentioned on Table 3.

For the two norm and the Gaussian ordination problems, the δ-machine based on either the training set or the reduced representation set had similar MRs (see Table 7 (a)). In the analysis of variance (see Table 8), for the two norm problem, the method factor and the two-way interactions including the method factor had either no effect or a small effect. The small effect size of the method factor suggests that using different representation sets hardly influenced the performance of the δ-machine. For the Gaussian ordination problem, the effect size of the method (m) was medium. The δ-machine using the whole training set

Table 7 The average results of the three problems for each type of the representation set

Problem	$v_e = 0.5$	$v_e = 0.9$	The whole training set	PAM
(a) The average misclassification rate				
Two norm	0.04 (0.02)	0.04 (0.02)	0.03 (0.02)	0.04 (0.02)
Four blocks	0.32 (0.07)	0.33 (0.07)	0.33 (0.08)	0.46 (0.08)
Gaussian ordination	0.34 (0.06)	0.32 (0.05)	0.32 (0.06)	0.35 (0.07)
(b) The average numbers of active prototypes/exemplars				
Two norm	6.37 (3.43)	7.58 (5.80)	6.62 (3.29)	4.74 (1.93)
Four blocks	8.98 (5.77)	44.73 (55.61)	31.80 (41.28)	7.54 (6.07)
Gaussian ordination	18.28 (16.09)	75.72 (85.59)	46.72 (60.47)	7.84 (4.78)

Standard deviations are shown within brackets

Table 8 Estimated effect size η_p^2 for each model component for the three problems

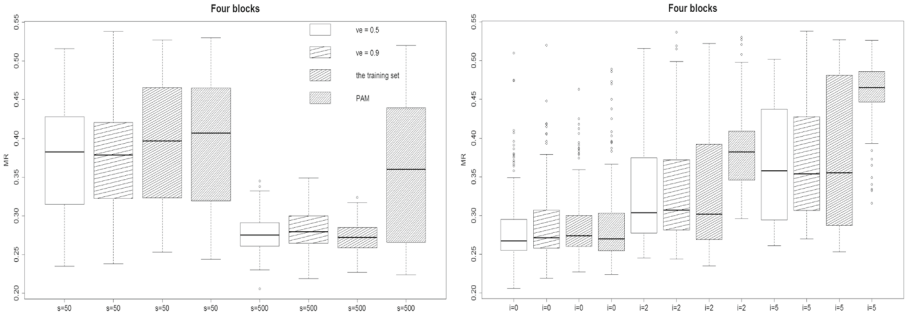
Problem	m	v	s	iv	$m:v$	$m:s$	$m:iv$	$v:s$	$v:iv$	$s:iv$
TN	0.01	0.01	0.29	0.05	0.00	0.02	0.00	0.00	0.01	0.02
FB	*0.25*		0.58	0.59		*0.15*	*0.16*			0.10
GOM	*0.10*	0.02	0.54	0.58	0.02	0.02	*0.06*	0.01	0.04	*0.07*

The method-related effect sizes which are larger than or equal to 0.06 are set in italics. *TN*, *FB*, and *GOM* are abbreviations of the two norm, the four blocks, and the Gaussian ordination problems

or using the K-means method (v_e = 0.9) had the lowest MRs. The δ-machine using PAM had the highest MR but also had the sparsest solutions.

For the four blocks problem, we only kept the data with two relevant variables. Thus, in Table 8, the spaces under the number of features (v) are blank. The method factor and the two-way interactions including the method factor had large effect. The large effect of the method factor was shown in Table 7(a), which the δ-machine using PAM had much higher MR than using other methods. Figure 4a illustrates the large effect size of the interaction between the method and the training size factor ($m : s$). When the size of the training set was 50 ($s = 50$), the δ-machine using the four representation sets had similar results. After the sample size increased to 500, PAM had the worst performance in terms of MRs. The other three methods performed similarly. Figure 4b showed the large effect of the interaction between the method and the number of irrelevant variables ($m : i$). If the data have no irrelevant variable, the four representation sets performed similarly. If the data have irrelevant variables, PAM is the worst representation set selection method to choose.

Table 7 (b) gives the average numbers of active prototypes/exemplars for each problem and each type of the reduced representation set. PAM and the K-means method ($v_e = 0.5$) had the smallest number of active exemplars/prototypes among all the three problems. Except for the two norm problem, the effect size of the method on the number of active prototypes was large (results not shown). This large effect size indicates that using the K-means method ($v_e = 0.5$) and PAM instead of the other two representation selection methods had fewer active prototypes/exemplars.



(a) The interaction of the method factor with the training size factor

(b) The interaction of the method factor with the number of irrelevant variables factor

Fig. 4 The box plots for MRs of the δ-machine with different representation selection methods. The shading patterns are explained in graph (**a**)

Table 9 Estimated effect size η_p^2 for each model component for the two norm problem

The method-related effect sizes which are larger than or equal to 0.06 are underlined

Outcome	Method	Ratio	Method:ratio
MR	0.03	0.63	0.02
AUC	0.05	0.58	0.05
F-measure	0.05	0.65	<u>0.07</u>

In sum, for the three artificial problems, the δ-machine using the K-means method ($v_e = 0.5$) had comparable misclassification rate with the δ-machine using the whole training set. Moreover, the δ-machine using the K-means method ($v_e = 0.5$) offered sparser solutions, resulting in more interpretive models. PAM performed well on the two norm and Gaussian ordination problem in terms of MRs and the sparseness of the solutions. For the four blocks problem, PAM had the worst performance.

4.5.4 Study with Imbalanced Classes

Table 9 displays the effect sizes for the method factor (*method*), the level of imbalance factor (*ratio*), and the interaction between the two (*method:ratio*). The method factor and the interaction (*method:ratio*) had small effects, except that the influence on the F-measure for the interaction has a medium effect ($\eta_p^2 = 0.07$). Therefore, we only show the results of the F-measure in Fig. 5. Furthermore, because SVM(RBF) and the δ-machine using the four dissimilarity functions performed the same when the levels of imbalance were from 1 to 4, we only show the results for levels 8 to 32. Figure 5 illustrates that with increasing class imbalance, the δ-machine using the Euclidean distance and the squared Euclidean distance function achieved higher classification performance than using the other two dissimilarity functions. Moreover, compared with SVM(RBF), the δ-machine had competitive

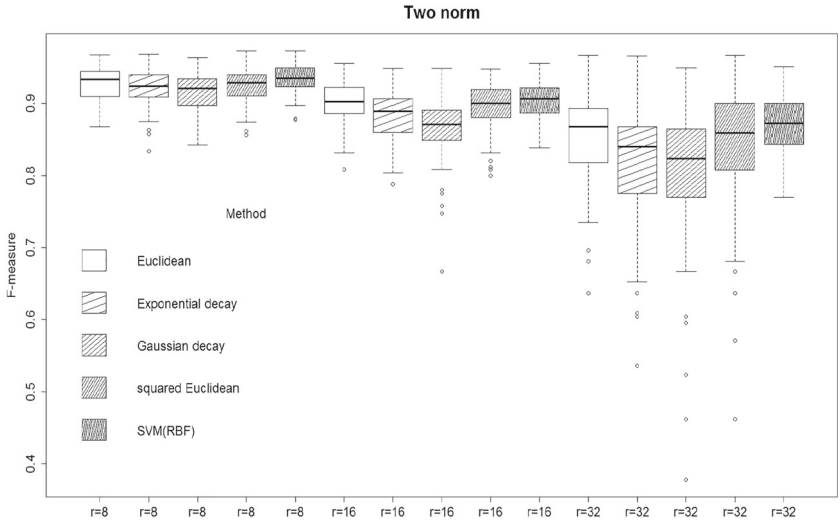


Fig. 5 The box plots for the F-measure of the δ-machine with different dissimilarity functions compared with SVM(RBF)

Table 10 The short descriptions of the 10 UCI datasets

Dataset	Problem	Variable	Size	Class distribution (%)
1	Haberman's Survival	3	306	73.53/26.47
2	Blood Transfusion Service Centre	4	748	23.80/76.20
3	Banknote Authentication	4	1372	44.46/55.54
4	Liver	5	345	44.93/55.07
5	Pima Indian Diabetes	8	768	34.90/65.10
6	Ionosphere	34	351	64.10/35.90
7	Spambase	57	4601	39.40/60.60
8	Sonar	60	208	53.37/46.63
9	Musk version 1	166	476	43.49/56.51
10	Musk version 2	166	6598	15.41/84.59

performance. The high average F-measure value (above 0.9, results not shown) suggests that the δ-machine is not sensitive to the class imbalance problem.

4.5.5 Study with Unequal Covariance Matrices

The covariance factor does influence the MR of the δ-machine, as $\eta_p^2 = 0.94$. With unequal covariance matrices the performance of the δ-machine became worse, which the average MR has increased from 0.02 to 0.08. The interaction between the method factor and the covariance factor had an effect size close to zero ($\eta_p^2 = 0.002$, results not shown), which means the δ-machine using the four dissimilarity functions had the same performance regardless of the two covariance matrices.

5 Applications

5.1 UCI Datasets

Ten empirical datasets (Table 10) from the UCI machine learning database (Newman et al. 1998) were used to evaluate the performance of the δ-machine and to compare the δ-machine with other methods. For these ten datasets, the number of predictor variables ranges from 3 to 166. The sizes of the datasets vary from 208 to 6598. For each dataset, half of the observations were assigned to the training set, the other half constituted the test set. As the distribution of the response variable is skewed in some datasets (e.g., Musk dataset version 2), stratified sampling was conducted when the training set and test set were sampled. Dataset descriptions are shown in Appendix.

The following methods were compared: the δ-machine using four (dis)similarity functions, classification trees, support vector machines with radial basis kernel (SVM (RBF)), and the Lasso. We decided to use SVM(RBF) here, because it is related to the δ-machine using the Gaussian decay function. The comparison of the δ-machine and the Lasso is a comparison of a linear classifier in the dissimilarity space versus the predictor space. We also used classification trees. Tree models are easily interpreted. But they are often not competitive with other supervised learning methods (James et al. 2013). Also classification trees suffer from high variance. Three outcome measures were chosen: the misclassification rate

(MR), the area under the receiver operating characteristic (ROC) curve (AUC), and the F-measure. The last two criteria were introduced because of the imbalanced classes in some datasets.

We mainly focused on the results of AUC shown in Table 11 (a) and took the results of MR and the F-measure as the references (see Table 11 (b) and (c)). In order to compare the four methods, we ordered ten datasets into three parts. Part I shows the datasets that had substantial AUC differences among the δ-machine with four (dis)similarity functions and had substantial differences between the δ-machine, classification trees, SVM(RBF), and the Lasso, where we define a substantial difference as a difference between the highest AUC and the lowest AUC larger than or equal to 0.05. Part II shows the datasets that had no substantial differences among the δ-machine with four functions, but had substantial difference between the δ-machine methods and the other methods; Part III shows the datasets that were insensitive to the choice of classification methods. Hence, Part I is used to evaluate the difference among the four dissimilarity functions in the δ-machine. Part I and Part II are important for the comparison of the performance of the different classification methods.

5.1.1 Results

For both datasets from Part I (see Table 11 (a)), the Gaussian decay function achieved the best prediction in terms of AUC, followed by the Euclidean distance and the exponential decay function. The squared Euclidean distance had the lowest AUC.

The comparison of the δ-machine with the other three classification methods were made based on the datasets from Part I and Part II in Table 11 (a). The δ-machine and the SVM(RBF) had the best predictive performance among all the methods in terms of AUC. Among the nine datasets from Part I and Part II, the δ-machine had five times lower, two times equal, and two times higher values than those of the SVM(RBF). From the five times, two datasets had a substantial decrease of AUC if the δ-machine was applied instead of the SVM(RBF). The performance of the δ-machine is better than the SVM(RBF) if we use the criteria of the F-measure and MR (see Table 11 (b) and (c)), which is shown by more bold values than the SVM(RBF) in these two tables. The similar results of the δ-machine and the SVM(RBF) using the three criteria show that the δ-machine is very competitive and sometimes superior to the support vector machine.

For the two other methods, classification trees and the Lasso, the Lasso had intermediate performance. The results showed that for the nine datasets from Part I and Part II, the δ-machine had three times equal and five times higher AUC than the Lasso. For three out of the five datasets, the difference of AUC between the δ-machine machine and the Lasso was substantial. The comparison of the δ-machine and the Lasso reveals that linear classifiers build in the dissimilarity space can achieve a better performance than the ones build in the original predictor space. Classification trees had the worst performance among all methods.

As the ten datasets are well known and are frequently used for evaluation of classification methods (Ghazvini et al. 2014; Duch et al. 2012; Tao et al. 2004), we compare the performance of the δ-machine against the previous results. In general, there was not a single classification method that had significantly best results for all these datasets in the past, because each classification method relies on assumptions, estimators, or approximations. Any classification method can achieve the best performance if there is an ideal dataset that fulfills these (Duin et al. 2010). The three studies carried out by Duch et al. (2012), Tao et al. (2004), and Ghazvini et al. (2014) had seven, two, and one out of ten datasets overlapping with our ten datasets, respectively. From these studies, we conclude that in

Table 11 Results of AUC, MR and the F-measure in the test sets of the 10 UCI datasets

Part	Dataset	The δ-machine				CART	SVM (RBF)	The Lasso	ref.
		Euclidean	sq.Euclidean	Exp.decay	Gau.decay				
(a) AUC results									
I	9	0.93	0.88	0.93	*0.95*	0.68	**0.96**	0.89	
	10	0.97	0.93	0.97	*0.98*	0.94	**0.99**	0.97	
II	1	0.67	0.67	0.67	0.66	no tree	**0.73**	0.71	
	2	**0.77**	0.76	0.75	0.75	0.71	0.72	**0.77**	
	4	0.68	0.69	0.67	0.67	0.65	**0.70**	0.67	
	5	**0.80**	0.79	**0.80**	**0.80**	0.70	0.79	**0.80**	
	6	**0.99**	0.98	**0.99**	**0.99**	0.85	**0.99**	0.89	
	7	**0.97**	**0.97**	**0.97**	**0.97**	0.89	**0.97**	**0.97**	
	8	0.83	0.81	0.83	0.82	0.65	**0.90**	0.78	
III	3	1.00	1.00	1.00	1.00	0.98	1.00	1.00	
(b) the F-measure results									
	9	0.82	0.76	0.83	***0.86***	0.60	**0.86**	0.79	
	10	0.80	0.74	0.81	***0.87***	0.79	0.82	0.83	
	1	0.84	0.84	***0.85***	***0.85***	no tree	**0.85**	0.83	
	2	0.32	0.27	0.42	***0.47***	0.45	0.25	0.11	
	4	0.55	*0.58*	0.52	0.50	**0.60**	0.50	0.42	
	5	0.57	0.58	*0.59*	0.55	0.58	0.59	**0.60**	
	6	0.97	0.96	0.97	**0.99**	0.89	0.96	0.92	
	7	0.90	0.90	***0.91***	***0.91***	0.86	0.90	0.90	
	8	0.77	*0.78*	0.76	0.76	0.67	**0.80**	0.77	
	3	1.00	1.00	1.00	1.00	0.97	0.99	0.99	
(c) MR results									
	9	0.16	0.20	0.15	*0.12*	0.29	**0.11**	0.18	0.08[3]
	10	0.05	0.07	0.05	***0.04***	0.06	0.05	0.05	0.10[3]
	1	0.26	0.27	0.27	***0.25***	no tree	**0.25**	0.28	0.25[1]
	2	0.21	0.22	***0.20***	0.21	0.23	0.22	0.23	0.21[1]
	4	0.35	***0.31***	0.35	0.36	0.36	0.35	0.39	0.30[1]
	5	0.26	***0.25***	***0.25***	0.26	0.27	0.26	**0.25**	0.23[1]
	6	0.04	0.06	0.03	***0.01***	0.14	0.06	0.10	0.05[1]
	7	***0.07***	0.08	***0.07***	***0.07***	0.11	0.08	0.08	0.06[1]
	8	0.27	*0.25*	0.27	0.30	0.35	**0.23**	0.26	0.14[1]
	3	0.00	0.00	0.00	0.00	0.02	0.01	0.01	0.03[2]

The best value among the four dissimilarity functions and among the four classification methods per dataset is shown in italic and in bold, respectively. Results of references [1], [2], and [3] were from Duch et al. (2012), Ghazvini et al. (2014), and Tao et al. (2004)

five out of the ten datasets, SVM(RBF) had achieved the lowest MR. SVM with a linear kernel, multilayer perceptron networks (Hornik et al. 1989), the iterative axis-parallel rectangle algorithm (Dietterich et al. 1997) and kernel-based multiple-instance learning model

(Tao et al. 2004) had one best result each. We compare the results of the δ-machine against these best results in the last column of Table 11 (c). Only the result of dataset 8 had a substantial lower MR value (0.14) than the δ-machine. Therefore, the δ-machine had a convincing performance in general.

5.2 Two Empirical Examples

In this section, we give two empirical examples. The first example shows that how the δ-machine is applied and interpreted, and uses the variable importance plot and partial dependence plots to show the relationship between the original predictor variables and the outcome variable. The aim of the second example is to show the difference of using the whole training set as the representation set and a smaller representation set selected by the K-means clustering and to illustrate the non-linear boundaries generated by the δ-machine in the original feature space.

5.2.1 Kyphosis Data

The first dataset concerns about the results of "laminectomy," a spinal operation carried out on children, to correct for a condition called "kyphosis" (see Hastie and Tibshirani 1990) for details. Data are available for 81 children in the R-package `gam` (Hastie 2015). The response variable is dichotomous, indicating whether a kyphosis was present after the operation. There are three numeric predictor variables:

- `Age`: in months
- `Number`: the number of vertebrae involved
- `Start`: the number of the first (topmost) vertebra operated on.

The predictor variables were standardized, before we computed the Euclidean distances. We then used these distances as variables in the Lasso logistic regression. The regression estimated function is

$$\begin{aligned}\log\left(\frac{\pi}{1-\pi}\right) &= 29.93 + 0.09d_1 - 0.19d_3 - 1.11d_{10} - 3.03d_{11} - 1.20d_{23} \\ &-3.22d_{25} + 1.27d_{27} + 1.87d_{35} + 0.22d_{43} + 0.54d_{44} - 1.95d_{46} + 1.71d_{48} - 1.91d_{49} \\ &+1.39d_{51} - 4.05d_{53} + 1.16d_{59} - 1.11d_{61} + 0.72d_{71} + 1.2d_{72} - 3.06d_{77}. \end{aligned} \quad (10)$$

With every unit increase in the distance towards observation 1 the log odds of presenting kyphosis after the operation go up by 0.09, with every unit increase in the distance towards observation 3 the log odds go down by 0.19. In order to see which predictor variables are important for the prediction and the conditional influence of each predictor on the response, we made a variable importance graph and partial dependence graphs. These are shown in Fig. 6, where it can be seen that the most important predictor is Start, this is followed by Age, while Number is of minor importance. The relationship between the three predictor variables and the probability is single-peaked, first going up, later declining.

5.2.2 Mroz Data

The Mroz data consist of 753 observations. The observations, from the Panel Study of Income Dynamics (PSID), relate to married women ($N = 753$). The data are available in the `car` package in R (Fox and Weisberg 2011). The outcome variable is whether or not the woman is participating in the labor force. Two predictor variables are available:

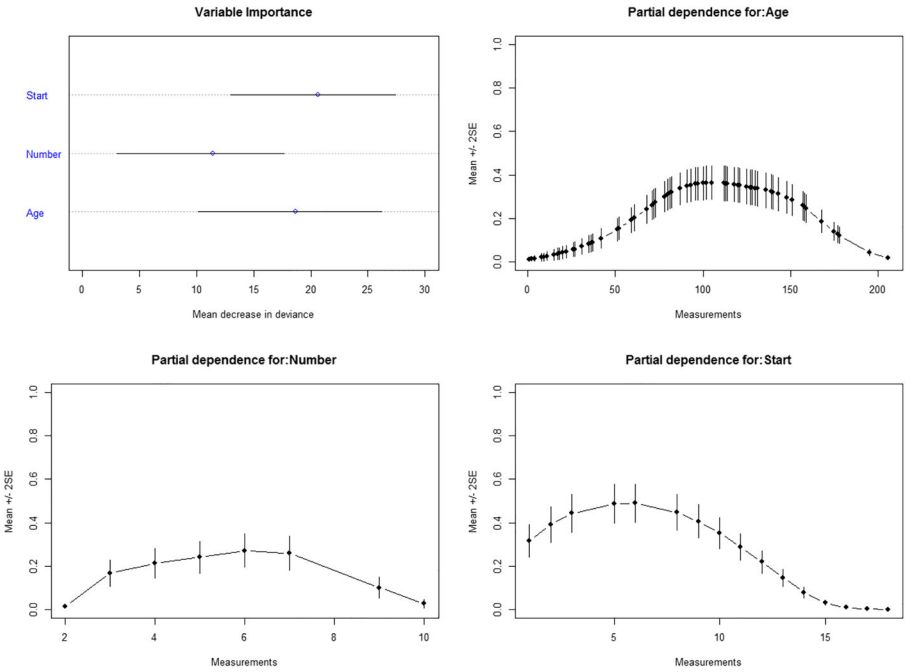


Fig. 6 Variable importance and partial dependence plots for the kyphosis data

- `lwg`: log expected wage rate; for women in the labor force, the actual wage rate; for women not in the labor force, an imputed value based on the regression of lwg on the other variables.
- `inc`: family income excluding wife's income.

The data were split randomly into a training set of 400 observations and a test set of the remaining 353 observations. In both the training and the test set, the percentage of women participating in the labor force (lfp) was about 57%.

We fitted the models on the training set and the representation set based on prototypes that explain 0.5 of the variance ($v_e = 0.5$) and subsequently made predictions. The percentage correct, the sensitivity, the specificity, the positive predictive value, and the negative predictive value were computed on the test set and were given in Table 12.

Table 12 shows that although the δ-machine using the training set had a higher percentage correct than that of the one using a smaller representation set, the difference was marginal, and the other results were competitive. Moreover, the number of active prototypes of the δ-machine using the smaller representation set was five, which was smaller than the 19 active exemplars of the one using the training set, resulting in a more interpretable model. The prediction regions derived by the δ-machine are presented in Fig. 7 which also includes the active exemplars/prototypes. It can be seen that the decision lines were nonlinear in the original two dimensional space.

Table 12 Percentage correct, sensitivity, specificity, positive predictive value, and negative predictive value in the test set for the δ-machine using the entire training set (exemplars) and using prototypes ($v_e = 0.5$)

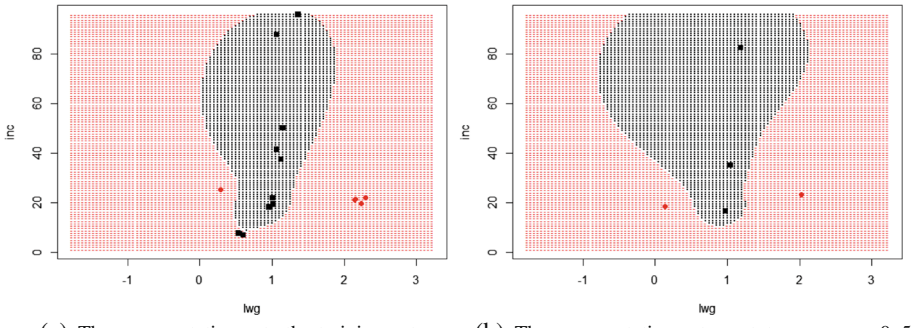
Method	Perc.Correct	Sensitivity	Specificity	Pos.Pred.Value	Neg.Pred.Value
The δ-machine (exemplars)	0.70	0.73	0.66	0.74	0.64
The δ-machine ($v_e = 0.5$)	0.69	0.75	0.61	0.71	0.65

6 Discussion

The psychological theory of categorization inspired the development of a method using dissimilarities as the basis for classification, which we called the δ-machine. From a set of variables, a similarity or dissimilarity matrix is computed using a (dis)similarity function. This matrix, in turn, is used as a predictor matrix in penalized logistic regression. It was shown that changing basis creates nonlinear classification rules in the original predictor space. The results on the empirical data sets in Section 5 showed that the δ-machine is competitive and often superior to other classification techniques such as support vector machines, Lasso logistic regression, and classification trees.

Five simulation studies have been conducted to investigate the properties of the δ-machine. The results showed that it is better to use the Lasso to select prototypes than forward stepwise based on AIC, that the Euclidean distance is a good dissimilarity measure, and that finding a smaller representation set based on prototypes gives sparse but competitive results. We found that the δ-machine is not sensitive to class imbalances, because boundaries between classes are built only on a few high-quality prototypes or exemplars. The four (dis)similarity functions had the same performance regardless of the covariance matrices.

In the pilot study and simulation study 3, we compared different methods to select the representation set. In the pilot study, three methods have been considered, the K-means method with the automatic decision rule, PAM, and the inner product clustering method. The first two methods had good performance, but the inner product clustering method did not perform well for the data with overlap. Because of this poor performance, we left out the inner product method in simulation study 3. The results obtained from both studies showed the performance of the δ-machine barely changed when one used a smaller representation set. In addition, using a smaller representation set offers sparser solutions and leads



(a) The representation set: the training set (b) The representation set: prototypes $v_e = 0.5$

Fig. 7 Predictions of labor force participation as a function of log wage rate and income for the Mroz data. Included in the graph with larger points are the active exemplars/prototypes

to more interpretable models. The results obtained from the pilot study showed that PAM had the same MR but sparser solutions than the K-means method. The K-means method using a higher level of the percentage of explained variance ($v_e = 0.9$) had lower MRs than the lower one ($v_e = 0.5$). The results obtained from simulation study 3 showed that the δ-machine using the K-means method with the automatic decision rule ($v_e = 0.5$) had the best performance among all the artificial problems regarding the good balance of MRs and the sparseness of the solutions. We chose two levels of the percentage of explained variance: $v_e = 0.5$ and $v_e = 0.9$. In some sense, these are arbitrary but both of them showed the great predictive performance in different types of data. Therefore, the choice of the percentage is related to the purpose of the study and data. The sparseness of PAM in our simulations might be caused by pre-setting the number of medoids for each class to be between two and ten, which limits the maximum number of exemplars to 20. To reduce the computational time, we have used the clustering results obtained from PAM as the initial setting for the inner product method. Even though the initial setting contained the prior information of the good performing PAM in most cases, the classification results deteriorated for the inner product method.

In our approach, the question in which respect two observations are similar is defined by the predictor variables. Of course, using a different set of variables might lead to completely different measures of dissimilarity, which would change the classification. The predictive performance of the Euclidean distance and the other three (dis)similarity functions in the δ-machine has been studied in simulation study 2, and it can be concluded that the Euclidean distance is a good dissimilarity function and we suggest to use it as the default dissimilarity measure. The squared Euclidean distance is sensitive to outliers, in this sense, that it creates large distances. These large distances have a negative influence on the classification performance.

For many dissimilarity measures, properties are well known (Gower 1966, 1971). However, the properties of dissimilarity measures in terms of classification performance are unknown. Such properties can be developed for various dissimilarity measures. Furthermore, it might be interesting to look at the added value of asymmetric dissimilarity measures. In standard dissimilarity measures $\delta_{ab} = \delta_{ba}$, i.e., the measure is symmetric. It has been pointed out that some dissimilarities are asymmetric; the Kullback-Leibler distance for two distributions for example is an asymmetric distance function. An asymmetric dissimilarity measure can always be broken down into a symmetric part and a skew symmetric part (Gower 1971), and further research might investigate how both parts perform in terms of classification.

We mainly investigated the performance of the δ-machine on the three artificial problems, that is, the three types of relationships between the outcome variable and the predictor variables. The two norm problem is a linearly separable problem, whereas the other two problems are not linearly separable. The four blocks problem creates data containing high-order interaction terms. The Gaussian ordination problem creates data containing quadratic terms. In real life, the relationship between the two is always unknown, and there are unlimited possibilities and far more complex relationships. Therefore, our simulation studies only shed some lights to the performance of the δ-machine. We have included the ten UCI datasets to evaluate the performance of the δ-machine. The results obtained showed that the δ-machine had convincing results in general, and that overall it is competitive to other classification methods. The comparison of the Lasso and the δ-machine suggested that building a classifier in the dissimilarity space achieves better predictive performance than building a classifier in the predictor space.

Bergman and Magnusson (1997) contrasted two approaches in the field of developmental psychopathology: the variable-oriented approach and the person-oriented approach. In a variable-oriented approach, the analytical unit is the variable, whereas in a person-oriented approach the analytical unit is the pattern. The approach we propose, however, the δ-machine, can bridge these two approaches. Since the δ-machine focuses on dissimilarities between profiles and prototypes, it is a person-oriented approach. The proposed variable importance measures and partial dependence plots, on the other hand, can be considered variable-oriented, and they enable us to interpret the importance of original variables and the conditional relationship between each predictor variable and the response variable. The partial dependence plots are similar to plots obtained using generalized additive models (Hastie and Tibshirani 1990) or from categorical regression using optimal scaling (Van der Kooij 2007).

Besides the methods mentioned in Section 3, other distance-based classification methods have been proposed before, i.e., the methods proposed by Boj et al. (2015) and Commandeur et al. (1999). Boj et al. (2015) argued that in marketing or psychology the data available often consists of dissimilarity data or distance information rather than of a set of variables. With the dissimilarity matrix ($\mathbf{D}$) and a response variable Y, the authors propose first to find a set of latent dimensions ($\mathbf{Z}$) representing the dissimilarity matrix, and then to use these latent dimensions as predictor variables in a generalized linear model (i.e., a logistic regression). This procedure is different from our procedure in the sense that we use the dissimilarity matrix directly as the predictors in the logistic regression procedure, whereas Boj et al. (2015) use the underlying configuration. In fact, if one started with a matrix of variables $\mathbf{X}$ and defined the distances on the basis of these variables, distance-based logistic regression would give the same result as a logistic regression based on $\mathbf{X}$ in terms of deviance and predicted probabilities.

Commandeur et al. (1999) provided a whole family of techniques for distance-based multivariate analysis (DBMA). For the data having numerical predictor variables ($\mathbf{X}$) and a categorical response (Y), firstly, Commandeur et al. (1999) defined the indicator matrix $\mathbf{Y}$, with a size of $I \times (C + 1)$, where $y_{ic} = 1$ if observation i belongs to class c, and otherwise zero. The Euclidean distance is used to obtain a distance matrix $d(\mathbf{XB_x})$ and a distance matrix $d(\mathbf{YB_y})$, where $\mathbf{B_x}$ and $\mathbf{B_y}$ are general matrices, and $d(\cdot)$ is the Euclidean distance function. DBMA finds a common configuration $\mathbf{Z}$ that simultaneously minimizes the STRESS loss function. Details of the algorithm can be found in Meulman (1992) or Commandeur et al. (1999). DBMA emphasizes the representation of objects and variables in a lower dimensional space, usually two-dimensional space. A disadvantage of DBMA is that it does not give a clear prediction rule to a new observation. The relationship between predictor variables and the response variable is indirect through the common configuration $\mathbf{Z}$. While the δ-machine is a classification and prediction approach, moreover, the proposed variable importance measures and partial dependence plots can illustrate the relationship between predictor variables and the response variable.

Although we focused on logistic regression as classification tool, other classification tools could be used. Discriminant analysis or classification trees could be used with the matrix $\mathbf{D}$ as the predictor matrix. Both will lead to interpretable models. Finally, more advanced classification techniques such as random forests (Breiman 2001) or boosting (Freund and Schapire 1997) could be applied, with the dissimilarity variables as input.

In the current paper, we focused on classification problems. We are confident that the results can be adapted for linear regression problems, i.e., the prediction of a continuous outcome variable. Other type of outcomes can also be modeled using the same ideas, i.e., counts using Poisson regression, ordinal outcomes using the proportional odds model, or

another multinomial model for ordinal outcomes (Agresti 2013). For a nominal outcome, variable discriminant analysis or the multinomial logit model can be used. In this manner, a versatile program can be built using dissimilarities as input.

For categorical variables, other dissimilarity measures can be defined, an overview of which can be found in Cox and Cox (2000). In some research settings, numeric and categorical predictor variables go side by side. In that case, a dissimilarity measure for mixed variables is needed. Gower (1971) defined such a general dissimilarity measure. These distances usually satisfy the distance axioms of minimality, symmetry, and the triangle inequality. The (dis)similarity coefficient has sufficient flexibility to be used under many circumstances. Therefore, the δ-machine can be extended to deal with the mixed types of predictor variables problem.

It is also important to investigate the performance of the δ-machine in a high-dimensional space. In the current paper, the number of variables included was up to 10. Increasingly, however, datasets are collected including very many variables, say 10,000 or more. In such a high-dimensional classification setting there are often many irrelevant variables. In addition, the relevant variables will be highly related. Therefore, the relevance of relevant variables is masked by the irrelevant variables. A second issue is that the distance caused by a difference in the relevant variables relative to the distance caused by the irrelevant variables becomes smaller. It means that concepts such as proximity and distance become less meaningful if we do not select variables. This issue is known to be especially detrimental for learners based on distances. Friedman and Meulman (2004) proposed a new approach for clustering observations on subsets of predictor variables. This approach allows for the subsets of predictor variables for different clusters to be the same, partially overlapping, or different, in contrast to a conventional feature selection approach which seeks clusters on the same predictor variables. It might be used in conjunction with the δ-machine; this will also be a topic of further investigation.

All methods discussed and shown were programmed in `R`. The code plus syntax for the analysis presented can be obtained upon request from the first author.

Appendix: Real datasets

All real datasets come from UCI repository.

Dataset 1. Haberman's Survival Dataset (Yeh et al.,2009): The dataset contains 306 patients who had undergone surgery for breast cancer at the University of Chicago's Billings Hospital between 1958 and 1970. The data have four variables.

Dataset 2. The database of Blood Transfusion Service Center in Hsin-Chu City in Taiwan. Seven hundred forty-eight observations were randomly selected from the the donor database. The data have four variables and a binary variable representing whether he/she donated blood before.

Dataset 3. Banknote authentication Dataset: The data have four variables which are extracted from images through the wavelet transform tool. The images were taken from genuine and forged banknote samples.

Dataset 4: Liver Disorders Dataset: The data contain of seven variables. The first five variables, $X_1 - X_5$, are all blood tests, and are sensitive to liver disorders that might arise from excessive alcohol consumption. The last variable, X_7, was widely misunderstood as the response variable (McDermott and Forsyth 2016). X_7 was not originally collected and is a selector of assigning the data into train and test set in a particular experiment. Therefore, in the study, we discard X_7. The variable X_6 is a numerical variable, which indicates the number of alcoholic drinks. Based on the suggestion of McDermott and Forsyth (2016), we dichotomized X_6 as $X_6 > 3$ and used it as the response variable.

Dataset 5: Pima Indians Diabetes Dataset: The data were selected from a larger database by some constraints. All patients here are females at least 21 years old of Pima Indian heritage. The data have eight variables.

Dataset 6: Ionosphere Dataset: The data were collected by a system which has a phased array of 16 high-frequency antennas with a total transmitted power of 6.4 kW in Goose Bay, Labrador. The targets were free electrons in the ionosphere. If there are some type of structure in the ionosphere, it returns as "Good" radar. On the contrary, it returns as "Bad" radar. Their signals pass through the ionosphere.

By using an autocorrelation function, the received signals were transformed to the arguments which are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Each instance in this dataset is described by two variables per pulse number. Thus, the data have 34 variables. There are 351 objects in the dataset, that is 126 "Bad" and 225 "Good."

Dataset 7: Spambase Dataset: The spam e-mails came from the postmaster and individuals who had filed spam. The non-spam e-mails came from filed work and personal e-mails. There are 57 variables which indicate whether a particular word or character was frequently occurring in the e-mail.

Dataset 8: Sonar dataset: The data contain 208 patterns which were obtained by bouncing sonar signals. Each pattern is a set of 60 numbers which ranges from 0.0 to 1.0. The goal is to discriminate the sonar signals between a metal cylinder and a roughly cylindrical rock.

Dataset 9: Musk dataset version 1: 166 variables are used to describe the exact shape, or conformation, of the molecule. The goal is to predict whether the new molecule is musk or not.

Dataset 10: Musk dataset version 2: The description is the same as dataset 9, except version 1 has a smaller sample size.

References

Agresti, A. (2013). *Categorical data analysis*, 3rd edn. New Jersey: Wiley.

Al-Yaseen, W.L., Othman, Z.A., Nazri, M.Z.A. (2017). Multi-level hybrid support vector machine and extreme learning machine based on modified K-means for intrusion detection system. *Expert Systems with Applications*, *67*, 296–303.

Ashby, F.G. (2014). *Multidimensional models of perception and cognition*, 1st edn. New York: Psychology Press.

Ben-Israel, A., & Iyigun, C. (2008). Probabilistic D-clustering. *Journal of Classification*, *25*(1), 5–26.

Bergman, L.R., & Magnusson, D. (1997). A person-oriented approach in research on developmental psychopathology. *Development and Psychopathology*, *9*(02), 291–319.

Berk, R.A. (2008). *Statistical learning from a regression perspective*, 1st edn. New York: Springer.

Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U. (1999). When is "nearest neighbor" meaningful. In Beeri, C., & Buneman, P. (Eds.) *Database theory - ICDT 99* (pp. 217–235). Springer: Berlin.

Boj, E., Caballé, A., Delicado, P., Esteve, A., Fortiana, J. (2015). Global and local distance-based generalized linear models. *TEST*, *25*(1), 170–195.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32.

Cohen, J. (1973). Eta-squared and partial Eta-squared in fixed factor ANOVA designs. *Educational and Psychological Measurement*, *33*(1), 107–112.

Commandeur, J.J., Groenen, P.J., Meulman, J. (1999). A distance-based variety of nonlinear multivariate data analysis, including weights for objects and variables. *Psychometrika*, *64*(2), 169–186.

Cooper, M.C., & Milligan, G.W. (1988). The effect of measurement error on determining the number of clusters in cluster analysis. In Gaul, W., & Schader, M. (Eds.) Data, expert knowledge and decisions (pp. 319–328). Berlin: Springer.

Cormack, R.M. (1971). A review of classification, Journal of the Royal Statistical Society. Series A (General), 321–367.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Cox, T.F., & Cox, M.A. (2000). *Multidimensional scaling*, 2nd edn. Boca Raton: CRC press.

De Rooij, M. (2001). *Distance models for transition frequency data: Ph.D dissertation*. Leiden University: Department of Psychology.

Dietterich, T.G., Lathrop, R.H., Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, *89*(1-2), 31–71.

Duch, W., Jankowski, N., Maszczyk, T. (2012). Make it cheap: learning with O (nd) complexity. In: The 2012 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–4.

Duin, R.P., Loog, M., Pekalska, E., Tax, D.M. (2010). Feature-based dissimilarity space classification. In: Recognizing patterns in signals, speech, images and videos. Springer, pp. 46–55.

Duin, R.P., & Pekalska, E. (2012). The dissimilarity space: bridging structural and statistical pattern recognition. *Pattern Recognition Letters*, *33*(7), 826–832.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, *27*(8), 861–874.

Fleiss, J.L., & Zubin, J. (1969). On the methods and theory of clustering. *Multivariate Behavioral Research*, *4*(2), 235–250.

Fox, J., & Weisberg, S. (2011). *An R companion to applied regression*, 2nd edn. Thousand Oaks: Sage.

Freund, Y., & Schapire, R.E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, *55*(1), 119–139.

Friedman, J. (2001). Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, *29*(5), 1189–1232.

Friedman, J., Hastie, T., Tibshirani, R. (2009). *The elements of statistical learning*, 2nd edn. New York: Springer.

Friedman, J., Hastie, T., Tibshirani, R. (2010a). glmnet: regularization paths for generalized linear models via coordinate descent, R package version 1.6-4, Available at http://www.jstatsoft.org/v33/i01/.

Friedman, J., Hastie, T., Tibshirani, R. (2010b). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, *33*(1), 1–22.

Friedman, J., & Meulman, J. (2004). Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 66*(4), 815–849.

Ghazvini, A., Awwalu, J., Bakar, A.A. (2014). Comparative analysis of algorithms in supervised classification: a case study of bank notes dataset. *International Journal of Computer Trends and Technology*, *17*(1), 39–43.

Gower, J.C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, *53*(3-4), 325–338.

Gower, J.C. (1971). A general coefficient of similarity and some of its properties. *Biometrics*, *27*(4), 857–871.

Hastie, T. (2015). gam: generalized additive models, R package version 1.12.

Hastie, T., & Tibshirani, R. (1990). Generalized additive models, 1st, Vol. 43, CRC Press, Boca Raton.

Hornik, K., Stinchcombe, M., White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359–366.

James, G., Witten, D., Hastie, T., Tibshirani, R. (2013). *An introduction to statistical learning*, 1st edn. New York: Springer.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: a systematic study. *Intelligent data analysis*, *6*(5), 429–449.

Kaufman, L., & Rousseeuw, P.J. (1990). *Finding groups in data: an introduction to cluster analysis*. New York: Wiley.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In Cam, L.M.L., & Neyman, J. (Eds.) *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability* (pp. 281–297). Berkeley: Calif.: University of California Press.

Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K. (2013). Cluster: cluster analysis basics and extensions, R package version 1.14.4.
McDermott, J., & Forsyth, R.S. (2016). Diagnosing a disorder in a classification benchmark. *Pattern Recognition Letters*, *73*, 41–43.
Meulman, J. (1992). The integration of multidimensional scaling and multivariate analysis with optimal transformations. *Psychometrika*, *57*(4), 539–565.
Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F. (2014). e1071: Misc Functions of the Department of Statistics (e1071), TU Wien, R package version 1.6-4, Available at http://CRAN.R-project.org/package=e1071.
Mirkin, B. (1999). Concept learning and feature selection based on square-error clustering. *Machine Learning*, *35*(1), 25–39.
Mirkin, B. (2012). *Clustering: a data recovery approach*, (pp. 230–233). Boca Raton: Chapman & Hall.
Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J. (1998). UCI repository of machine learning databases, Available at http://www.ics.uci.edu/mlearn/MLRepository.html.
Nosofsky, R.M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, *115*(1), 39–57.
Pekalska, E., & Duin, R.P. (2005). *The dissimilarity representation for pattern recognition: foundations and applications*. Singapore: World Scientific.
R Core Team (2015). R: a language and environment for statistical computing, Vienna, Austria: R Foundation for Statistical Computing, Available at http://www.R-project.org/.
Richardson, J.T. (2011). Eta squared and partial eta squared as measures of effect size in educational research. *Educational Research Review*, *6*(2), 135–147.
Rousseeuw, P.J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, *20*, 53–65.
Rovai, A.P., Baker, J.D., Ponton, M.K. (2013). *Social science research design and statistics: a practitioner's guide to research methods and IBM SPSS* Vol. 2. Chesapeake: Watertree Press.
Schaffer, C.M., & Green, P.E. (1996). An empirical comparison of variable standardization methods in cluster analysis. *Multivariate Behavioral Research*, *31*(2), 149–167.
Steinley, D. (2004). Standardizing variables in K-means clustering. In Banks, D., McMorris, F.R., Arabie, P., Gaul, W. (Eds.) Classification, clustering, and data mining applications (pp. 53–60). Berlin: Springer.
Steinley, D., & Brusco, M.J. (2011). Choosing the number of clusters in K-means clustering. *Psychological Methods*, *16*(3), 285.
Tao, Q., Scott, S., Vinodchandran, N.V., Osugi, T.T. (2004). SVM-based generalized multiple-instance learning via approximate box counting. In: Proceedings of the 21st international conference on machine learning. ACM, pp. 101.
Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, *58*(1), 267–288.
Van der Kooij, A.J. (2007). *Prediction accuracy and stability of regression with optimal scaling transformations: Ph.D dissertation*. Leiden University: Department of Education and Child Studies.
Van Rijsbergen, C.J. (1979). Information retrieval (2nd ed.): Butterworths.
Venables, W.N., & Ripley, B.D. (2002). *Modern applied statistics with S*, 4th edn. New York: Springer. Available at http://www.stats.ox.ac.uk/pub/MASS4.
Vesanto, J. (2001). Importance of individual variables in the K-means algorithm. In Cheung, D., Williams, G.J., Li, Q. (Eds.) *Advances in knowledge discovery and data mining* (pp. 513–518). Berlin: Springer.
Yeh, I.-C., Yang, K.-J., Ting, T.-M. (2009). Knowledge discovery on RFM model using bernoulli sequence. *Expert Systems with Applications*, *36*(3), 5866–5871.
Zhu, J., & Hastie, T. (2012). Kernel logistic regression and the import vector machine. *Journal of Computational and Graphical Statistics*, *14*(1), 185–205.