A multi prototype classification algorithm and its application to multi class diagnostics

Mario Ziller

This paper introduces a novel, universal distance-based classification procedure. It is based on a simple geometric model. Considering all objects as points in a metric space, a class is imagined as covered by potentially differentsized hyperspheres, the centres of which are referred to as prototypes. The radii of the hyperspheres are individually optimised by a generalised ROC-analysis. For the approximate solution of the entire discrete optimisation problem, a greedy algorithm was developed and implemented in R. It runs in $O(k^2 \bullet n^2 \bullet \log(n))$ time where k is the number of prototypes to be selected and n the number of training objects. For application to multi class problems, one against all approach is performed. The diagnostic decision is finalised for that class of maximum positive predictive value when in doubt. Objects not recognised as a member of any of the classes are assigned to an additional residual class. The performance of the classification system presented is demonstrated on various data examples, and in comparison with other methods.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

A multi prototype classification algorithm and its application to multi class diagnostics

Mario Ziller[1]

[1]**Biomathematics Working Group, Friedrich-Loeffler-Institut, Federal Research Institute for Animal Health, Greifswald - Insel Riems, Germany**

ABSTRACT

This paper introduces a novel, universal distance-based classification procedure. It is based on a simple geometric model. Considering all objects as points in a metric space, a class is imagined as an overlap of potentially different-sized hyperspheres, the centres of which are referred to as prototypes. The radii of the hyperspheres are individually optimised by a generalised ROC-analysis. For the approximate solution of the entire discrete optimisation problem, a greedy algorithm was developed and implemented in R. It runs in $\mathcal{O}(k^2 \cdot n^2 \cdot log(n))$ time where k is the number of prototypes to be selected and n the number of training objects.

For application to multi class problems, one against all approach is performed. The diagnostic decision is finalised for that class of maximum positive predictive value when in doubt. Objects not recognised as a member of any of the classes are assigned to an additional residual class. The performance of the classification system presented is demonstrated on various data examples, and in comparison with other methods.

Keywords: Prototype classifier, Global distance-based classification, ROC analysis, Greedy algorithm, Multi class diagnostics

1 INTRODUCTION

Many diagnostic decisions in medicine, biology, and far beyond are substantiated on large pattern data bases e.g. of DNA-sequences, mass spectra, or others. These data bases often include many predefined diagnostic classes (Wong et al., 2010; De Bruyne et al., 2011; Karger et al., 2011; Hong et al., 2008). The classification is usually done by comparison based on distances or similarity scores. Screening for similar proof-samples by local distance-based methods makes use of the nearest neighbour principle. This works well but may be computationally intensive (Tan et al., 2005). For classification assignment, the distances

of an unknown object to all of the data base objects must be assessed. Therefore, it may become ineligible for application in high-throughput diagnostics e.g. of emerging diseases in case of a pandemic situation.

For comparable problems, a global approach may be more useful. Its application to large data sets operates much faster. There are various global distance-based methods like centroid- or medoid-based classification, etc. (Breiman et al., 1984; Tan et al., 2005; Gordon, 1999; Hastie et al., 2009). They use one or more class representatives for comparison. One can imagine that in general class members should have lower distances to other class members than to non-members. Therefore, the choice of the distance measure is of great importance for the correctness of subsequent class-assignments. Simple k-means or k-medoids models consider only one representative of each class. They may be effective if all classes are spherically shaped (Tan et al., 2005; Hastie et al., 2009). Plain assignments to the nearest centroids or medoids, which other classifiers like e.g. neural gases (Martinetz and Schulten, 1991) make use of, however, would require either classes of nearly equally sized hyperspheres or the selection of possibly a great many representatives.

All this might be improper in real practical applications. In this context, a more general approach (Ziller et al., 2011) was developed. In this, a class is considered being covered by several different-sized hyperspheres. Hence, more than one centre of each class and different radii are inspected. For the optimum sizing of those hyperspheres receiver operating characteristic (ROC)-analysis is utilised, which is a well-established standard framework for evaluating diagnostic tests and assessing suitable cutoffs (Hanley and McNeil, 1982; Greiner et al., 2000). For the classification task under consideration, this fundamental idea was generalised for application to multi-core models.

In this paper, a novel multi prototype classifier is outlined. The basic classification algorithm for single class problems and its generalisation to multi class problems are expounded. Performance reliability of the classifier is demonstrated on various data sets in comparison with other well-established classification tools.

2 MULTI PROTOTYPE CLASSIFICATION

2.1 Theoretical framework

Mathematical model

For a short mathematical outline, let all objects be considered as points in a given metric space. Any class to be investigated is modelled as a union of various, potentially different-sized hyperspheres which may overlap. The centres of the hyperspheres represent reference objects of the class, henceforth referred to as 'prototypes'. The radii of the hyperspheres are used as thresholds, hereafter referred to as 'cutoffs', for the diagnostic decision. They are individually optimised by a generalised ROC-analysis which all

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

other hyperspheres were fixed in. An object is assigned to the class under consideration if its distance to at least one of the prototypes is not greater than the respective cutoff. In this approach only distances between the objects are given. Feature vectors or other information are not available. No additional model assumptions, e.g. about the distribution of point distances or about cluster morphologies, i.e. a radially decreasing density of objects within the class, were made.

Let O be the set of n objects considered, $C \subset O$ the subset of m class members, and d a metric distance function on O. The class should be characterised by the prototypes $x_i \in C, i = 1,...,k$ and the related cutoffs $f_i \in R \geq 0$. The number k of prototypes is arbitrary but fixed and must have been selected before the analysis starts. The membership of an object $y \in O$ to the class C was then modelled by the decision rule

$$y \in \hat{C} \quad \Leftrightarrow \quad \exists i \in 1,...,k : d(y,x_i) \leq f_i \,.$$

The set of all predicted class members is therfore

$$\hat{C} = \bigcup_{i=1}^{k} \hat{C}_i = \bigcup_{i=1}^{k} \{y \in O : d(x_i,y) \leq f_i\} \,.$$

ROC-Analysis

ROC-analysis is a well-established and for this purpose generalised method of estimating thresholds for effective dichotomisation. This procedure represents an essential part of the investigated algorithm and will be massively applied in it. Cutoff thresholds for diagnostic decisions can be determined by classical ROC-analyses (Hanley and McNeil, 1982; Greiner et al., 2000). For a given feature function f, a threshold f^* should be determined which makes $f(x) \leq f^*$ a good classifier, i.e. according to the declared notation, an object $x \in O$ will be assigned to class C if and only if $f(x) \leq f^*$. In the described classification algorithm, the distance to a fixed prototype is used as feature-function. The best cutoff f^* can be found by optimising a criterion function g.

As usual, sensitivity Se is defined as the proportion of correctly predicted class members with respect to all class members, whereas specificity Sp is the proportion of correctly predicted non-class members with respect to all non-class members.

$$Se = \left|\hat{C} \cap C\right| / \left|C\right| ,$$
$$Sp = \left|\overline{\hat{C}} \cap \overline{C}\right| / \left|\overline{C}\right| ,$$

where $\overline{\ }$ denotes the set complement with respect to O. For a good classification result, sensitivity and

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

specificity both should be high. The function

$$g = 1 - (1 - Se)^2 - (1 - Sp)^2$$

was therefore chosen as optimisation criterion.

Generalised ROC-Analysis

For the optimisation of cutoffs in case of more than one prototype, the idea of ROC-analysis was generalised. All but one of the prototypes with their corresponding cutoffs have been fixed, and the remaining prototype and its cutoff are optimised in each step, only. Without loss of generality, let the prototypes x_i and their cutoffs f_i be fixed for $i = 1,...,k-1$. For a prototype x_k, the cutoff f_k can be optimised by generalising the ROC principle.

According to the model

$$\hat{C} = \bigcup_{i=1}^{k} \hat{C}_i = \bigcup_{i=1}^{k-1} \hat{C}_i \cup \hat{C}_k,$$

for this particular case, sensitivity and specificity can be written as

$$Se = \left|\hat{C} \cap C\right| / |C| = \left(\left| \bigcup_{i=1}^{k-1} \hat{C}_i \cap C \right| + \left| \hat{C}_k \cap \overline{\bigcup_{i=1}^{k-1} \hat{C}_i} \cap C \right| \right) / |C|, \text{and}$$

$$Sp = \left|\overline{\hat{C}} \cap \overline{C}\right| / \left|\overline{C}\right| = \left(\left| \overline{\bigcup_{i=1}^{k-1} \hat{C}_i} \cap \overline{C} \right| - \left| \hat{C}_k \cap \overline{\bigcup_{i=1}^{k-1} \hat{C}_i} \cap \overline{C} \right| \right) / \left|\overline{C}\right|.$$

From this partition it becomes obvious that the analysis needs to be run over all remaining points, not predicted as class members according to the first $k-1$ prototypes, only.

Discrete optimisation

The estimation of the parameters $\{(x_i \in C, f_i \in \mathbb{R} \geq 0), i = 1,...,k\}$ of the given model based on training data $O = \{o_j \in O, j = 1,...,n\}$ with $|C| = m$ leads to a discrete optimisation problem: Choose a subset of k objects out of the m training class objects as prototypes, and k corresponding cutoffs out of the actual distances between training objects and prototypes which jointly optimise the objective function g:

$$\max_{(x_i, f_i)} \ (1 - (1 - Se)^2 - (1 - Sp)^2).$$

However, the direct solution of the entire optimisation task would take a prohibitive amount of computation time and memory consumption. There are $\binom{m}{k}$ possible choices of prototypes and n potential cutoffs for every prototype. For an effective assessment of the best cutoff, prior sorting of all distances

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

between the objects and the prototype considered is necessary. This can be done in $\mathcal{O}(n \cdot log(n))$ time. Furthermore, the number of prototypes is intended to be small, and for a well-posed problem, the numbers of class members and of non-members both should be of the same order. Presuming $k \ll n$ and $m, n\text{-}m \in \mathcal{O}(n)$ therefore, results in a time consumption of

$$\binom{m}{k} \cdot n \cdot log(n) \cdot k \quad \in \quad \mathcal{O}(n^{k+1} \cdot log(n)).$$

The exact solution of this problem needs exponential time in k and polynomial time with a high exponent with respect to n. The below presented greedy algorithm, however, reduces time consumption in both aspects.

2.2 Algorithm and implementation

Greedy algorithm

The complete solution of the multi-dimensional discrete optimisation problem would soon exceed available computer resources. For a reasonable determination of sufficiently good k prototypes and their corresponding cutoffs, a greedy algorithm (Cormen et al., 2009) was developed. The problem was reduced, therefore, to repeatedly choosing only one of the k prototypes anew. An approximate solution of the entire problem can be found this way. The main principles of the algorithm are here described in the first instance. Several improvements and practical simplifications will be incorporated in a subsequent paragraph. Starting with a random choice of k objects of the training set as preliminary prototypes, and with cutoffs all initialised with -1, an iteration process is started. A negative radius of a hypersphere corresponds to the empty set. At the start, there are no predicted class members.

In each cycle, it is investigated for all prototypes whether a better cutoff can be found, or there is one of the residual objects which improves g after substitution and appropriate cutoff-estimation by the generalised ROC-procedure. The substitution is manifested for the prototype for which the maximum improvement of the ROC-criterion g is reached, if possible. Thus, the criterion-value g is actually improved in each step.

If no better g-value is found the algorithm stops. The estimation of the classification parameters has finished therewith. The main ideas of the greedy algorithm are summarised in the pseudo code (see Tab. 1).

Runtime analysis

The algorithm considers substituting one out of k prototypes and its associated cutoff in each iteration-cycle. All $m-k$ remaining class members and the one, old respective prototype are potential new prototypes, and the distances from the current prototype to all n objects of the training data might be cutoffs. Again,

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

Table 1. Pseudocode of the greedy algorithm.

Start	random choice of preliminary prototypes with negative cutoffs

Iteration	repeat
	for i=1 to k do
	{ fix all other prototypes and cutoffs, consider prototype i
	for all remaining class members and prototype i itself
	{ { substitute prototype i
	optimise cutoff by generalised ROC-Analysis
	calculate criterion
	if any improvement then replace by best new prototype and cutoff
	else stop
	end

cutoff assessment needs $\mathcal{O}(n \cdot log(n))$ time because of prior sorting. Furthermore, numerous applications of the algorithm to example data support the assumption that the number of iteration steps j is of the same order as the number of prototypes. This will also be exemplified by data in Sect. 4.

Presuming $m, n\text{-}m \in \mathcal{O}(n)$ as before, the time consumption of the entire algorithm is thus

$$k \cdot (m-k+1) \cdot n \cdot log(n) \cdot j \quad \in \quad \mathcal{O}(k^2 \cdot n^2 \cdot log(n)).$$

Finally, the greedy algorithm needs polynomial time in both parameters. In view of $k \ll n$, the practical overall order is $\mathcal{O}(n^2 \cdot log(n))$.

Practical implementation

The depicted algorithm enables a quick estimation of good prototypes and respective cutoffs. However, some steps of the general algorithm are redundant and should be skipped to make it faster. Other segments may be simplified or rearranged. The following four aspects below were treated. Thus, the final algorithm works more effectively. The computational effort could be reduced in a practically pronounced extent this way, although it does not change its asymptotic order.

(1) The first idea of enhancing the performance of the algorithm is a triangle-like successive extension of the set of prototypes at the start. In the first k iteration cycles, only one new prototype is added. The treatment of more than one dummy is not necessary. Given the complete distance matrix of the training set, a standard ROC analysis is performed for every class-member in the first cycle. The object achieving the highest ROC-criterion is then chosen as starting prototype. In each of the following $k-1$ iteration cycles, one dummy only, i.e. a randomly chosen prototype with negative cutoff, is added until all k prototypes are gained.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

(2) The prototypes are permuted after each iteration step to store the recently renewed object with the highest index $j \leq k$ according to (1). Thus, at most only $j-1$ preliminary prototypes need to be inquired for better ones in the following cycle. Prototype j has proven to be the optimum choice in the previous cycle.

(3) When performing a generalised ROC-analysis for the prototype i, only the distances to those objects need to be examined which are not predicted as class members according to any of the other just fixed prototypes (see Sect. 2.1).

(4) In some occasional instances, the algorithm might even stop when some of the dummies have not been substituted. The best possible classification may already have been reached with fewer prototypes than envisaged beforehand. If this has happened, all remaining dummies are removed from the final parameter set.

Application to multi class problems

The classification algorithm described so far accomplishes the selection of prototypes and cutoffs for one pre-agreed class. In order to solve a multi class problem, this procedure is performed separately for every single class in a one against all approach. Thus, at the end of the analysis of training data, only a few prototypes and respective cutoffs have been selected for each of the classes.

In anticipation of subsequent multi class diagnostics, the empirical functions of positive predictive values (PPV) are supplementally stored for each of the prototypes. The positive predictive values can easily be derived in the course of ROC-analysis. It is the proportion of correctly predicted class members with respect to all predicted class members, both justified by the prototype under consideration. For the prototype x_i of class C_j at distance d^*, it is

$$PPV_i = PPV_i(d^*) = \left|\left\{y \in C_j : d(y,x_i) \leq d^*\right\}\right| / \left|\left\{y \in O : d(y,x_i) \leq d^*\right\}\right|.$$

Given the estimated parameters of all single class problems and the corresponding empirical PPV functions, the distances of any object of unknown class to all prototypes must be calculated first. Only those prototypes with distances not higher than their corresponding cutoffs are further considered. If exactly one prototype is left, its class is predicted for the object. If more than one prototype comes into question, the object is then assigned to the class of the prototype with the highest related PPV. If all distances were higher than the corresponding cutoffs, the object is assigned to an additional residual class. This result should clarify that a reliable classification of that object is not possible based on the data given.

The ruling based on PPV facilitates a balanced diagnostic decision for all classes; and the extraneous residual class ensures a high specificity of classification even if it is tried for an object that indeed does

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

not belong to any of the trained classes.

Software

The elucidated multi prototype classifier was implemented in R (R Core Team, 2012), version 2.15.2 (2012-10-26). The sources of the corresponding R-functions "fit.prototypes.r" and "predict.prototypes.r" will be made available on request.

3 EVALUATION EXAMPLES

The performance of the classification system presented in this paper is demonstrated at various application examples, and in comparison with other methods. In this context, time expenditure and accuracy of prediction are the main aspects of evaluation.

For the comparison of final classification results, six multi class problems have been selected. They are based on example data sets provided by the UCI Machine Learning Repository (Bache and Lichman, 2013). All data sets comprise moderately many features and classes. All features are numeric and have been treated as continuous data. The number of included objects enables a serious and fast classification. One data set (Statlog shuttle) comprises many more objects than the others. So, distinctions in computation time requirements can be illustrated better. The chosen data examples cover various fields of application: biomedical data, image data, and physical data.

3.1 Data sets

Cardiotocography data set (Ayres-de Campos et al., 2000). The data set consists of 21 measured features of fetal heart rate and uterine contraction on 2126 cardiotocograms ("Cardiotocography') classified by expert obstetricians with respect to both, a morphologic pattern and a fetal state. In this collection fetal state classes (NSP) are considered, only. Their frequencies are

1	2	3
1655	295	176

.

Statlog image segmentation data set (Brodley, 1990). Segments from seven images (classes) were evaluated for several pixel measures. Each of the 2310 instances represents a 3x3 region. Attribute no. 3 was removed because of identical entries. So, this data set ("Statlog Image Segmentation") comprises 18 features to predict seven equally sized classes:

1	2	3	4	5	6	7
330	330	330	330	330	330	330

.

Statlog shuttle data set (Catlett, 1994). NASA Shuttle database deals with the positioning of radiators in the Space Shuttle. In the original, it consists of 43500 training objects and 14500 test objects. Each instance is described by nine continuous attributes and is assigned to one of seven classes with the frequencies

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

1	2	3	4	5	6	7
45586	50	171	8903	3267	10	13

Classes 6 and 7 are too small for substantial classification and are omitted afterwards which consequently results in the frequencies

1	2	3	4	5
45586	50	171	8903	3267

All 57977 remaining objects are united into one data set ("Statlog Shuttle") in advance of subsequent cross-validation.

Statlog vehicle silhouettes data set (Siebert, 1987). The data set ("Statlog Vehicle Silhouettes") provides 18 features extracted from the silhouettes of four types of vehicle. The vehicle may have been viewed from one of many different angles. The overall number of examples amounts to 846, its class frequencies are:

bus	opel	saab	van
218	212	217	199

Waveform database generator (version 1) data set (Breiman et al., 1984). This artificial data set ("Waveform Data Generator") was simulated for classification purpose (for details see Breiman et al. (1984)). Three classes of waves were generated from a combination of two of three "base" waves. Each instance consists of 21 attributes, all of which include noise. A total of 5000 instances was uniformly partitioned into the classes:

1	2	3
1657	1647	1696

Yeast data set (Horton and Nakai, 1996, 1997) (modified). This data set was collected for predicting the cellular localisation sites of proteins. In the original data, eight numeric attributes of altogether 1484 objects were provided, and proteins were divided into ten classes (Horton and Nakai, 1996). The numbers of objects of several classes, however, were fairly low. For the purpose under investigation, all three membrane proteins (ME1, ME2, ME3) have been united to class MEM, therefore. Furthermore, all other small groups (EXC, VAC, POX, ERL) have been pooled to a residual class RES. The original class frequencies were

CYT	MIT	NUC	ME1	ME2	ME3	ERL	EXC	POX	VAC
463	244	429	44	51	163	5	35	20	30

whereas the frequencies of the modified data set ("Yeast Protein Localisation") are

CYT	MIT	NUC	MEM	RES
463	244	429	258	90

Descriptive parameters of all applied data sets are gathered in Tab. 2.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

Table 2. Numbers of objects, features, and classes of evaluation data sets.

Data Set	Number of		
	Objects	Features	Classes
Cardiotocography	2126	21	3
Statlog Image Segmentation	2310	18	7
Statlog Shuttle	57977	9	5
Statlog Vehicle Silhouettes	846	18	4
Waveform Data Generator	5000	21	3
Yeast Protein Localisation	1484	8	5

3.2 Methods

For demonstration and evaluation, the described classification algorithm is compared with several widely used classifiers. Data-based and feature-based methods have been selected as well to get a general impression of the performance.

Multi prototype classifier. Multi prototype classification ("PRO") has been performed for all data sets, each with a series of different parameters. Euclidean, Manhattan, or Canberra distances were applied to the data. The maximum number of prototypes was set to 3, 5, or 7.

k-Nearest neighbour classifier (Tan et al., 2005). The implementation of the k-nearest neighbour algorithm ("kNN") of the R package "knnflex" (Brooks, 2007) was utilised. Again, Euclidean, Manhattan, or Canberra distances were applied to the data. The number of neighbours considered for classification was chosen as 1, 3, or 5.

Support vector machine (Kohonen, 2001). The support vector machine algorithm ("SVM") of the R package "e1071" (Meyer et al., 2012) was utilised.

CART (Breiman et al., 1984). This classical variant of decision trees ("CART") provided by the R package "rpart"(Therneau et al., 2012) was applied.

C4.5 (Quinlan, 1993). The implementation of the R package "RWeka" (Hornik et al., 2009; Witten and Frank, 2005) was used for this algorithm ("C4.5").

3.3 Validation

Limited 10-fold cross-validation. In each of the data analyses, a limited 10-fold cross-validation has been performed. The data set is randomly partitioned into ten almost equal-sized subsets, therefore. In each of the ten steps, one subset serves as test data as in usual cross-validation (Tan et al., 2005; Hastie et al., 2009). The remaining nine subsets together constitute the preliminary training data. It is randomly reduced to maximum 100 objects of each class if more are present. With this limitation of the training data sets, the estimation of model parameters is based on the information of at most 100 objects of each class for all classification procedures. Although it might not lead to the best classification result possible,

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

Table 3. Optimum distance measures and numbers of prototypes or neighbours, respectively.

Data Set	Multi prototypes		k-Nearest neighbours	
	Metric	No. of prototypes	Metric	No. of neighbours
Cardiotocography	Canberra	7	Canberra	1
Statlog Image Segmentation	Manhattan	7	Manhattan	1
Statlog Shuttle	Euklidean	7	Manhattan	1
Statlog Vehicle Silhouettes	Euklidean	7	Manhattan	1
Waveform Data Generator	Euklidean	3	Manhattan	5
Yeast Protein Localisation	Manhattan	5	Manhattan	5

it should do for comparison.

Adjusted Rand index. The summarised true and predicted class frequencies are compared by the adjusted Rand index ("ARI") (Vinh et al., 2010; Hubert and Arabie, 1985; Vinh et al., 2009). It is corrected for the bias of different class sizes and is therefore considered as main criterion of success.

Accuracy. The plain accuracy ("ACC") or naive percentage as the proportion of correctly classified objects disregards different numbers of objects in the various classes. This index is calculated in percentage terms for additional consideration.

Number of iteration cycles. In case of multi prototype classifier application, the number of iteration cycles is recorded for every single class approach. This oblique processing parameter affects the effective calculation time.

Computation time. Computation time is recorded for training and prediction parts of the calculations, separately. It should demonstrate the operationality of the applied methods for sundry applications.

Software. All calculations were performed using R (R Core Team, 2012), Version 2.15.2 (2012-10-26).

4 RESULTS

All data sets have been analysed as described in Sect. 3.2. In case of multi prototype or k-nearest neighbour classifiers, the best classifications with respect to distance measure and number of prototypes or neighbours, respectively, were used for comparison. Adjusted Rand index was applied as selection criterion. The best choices of the processing parameters are listed in Tab. 3.

The classification performances of the compared methods were measured by ARI and ACC. All results are visualised in Fig. 1. They vary with data sets and with methods. The values of ARI range from 0.183 .. 0.269 for the yeast data to 0.808 .. 0.991 for the shuttle data. The corresponding ACC ranges are 52.4 .. 60.8 (yeast) and 94.5 .. 99.7 (shuttle).

In every performed basic multi prototype classification, i.e. single class analysis, the number of iteration cycles was recorded. Fig. 2 shows boxplots for all data sets dependent on the number of prototypes each class. According to the applied cross-validation procedure, one boxplot summarises ten

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

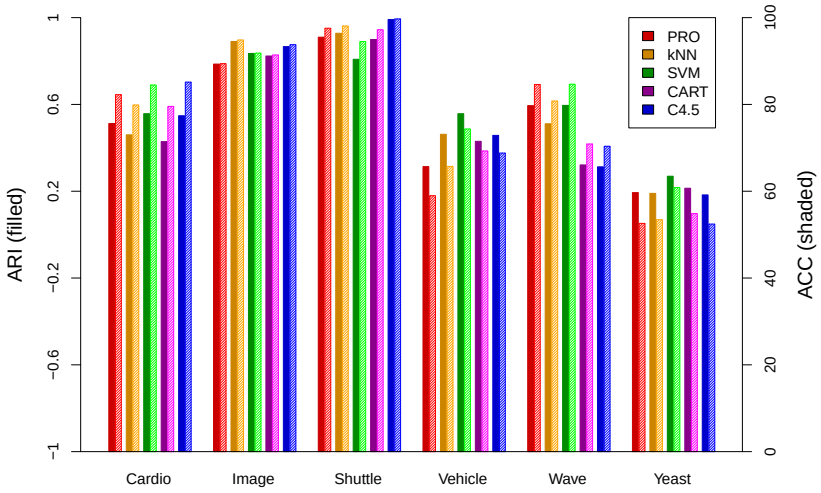

Figure 1. Adjusted Rand indices and accuracies of the classification results.

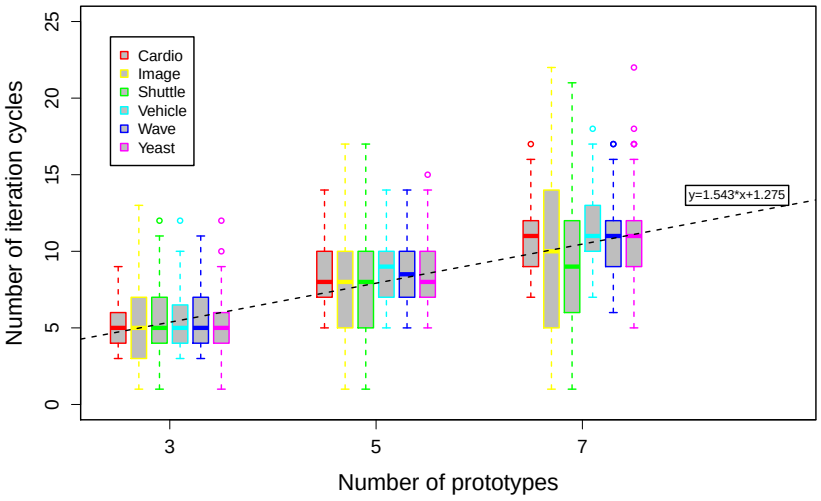

Figure 2. Number of iteration cycles of the milti prototype classifier.

repetitions of the number of cycles for each of the classes of the underlying data set, and for each of the metrics investigated. The estimated parameters of a linear regression of the entire resulting data are 1.543 (inclination) and 1.275 (intercept). The data support the assumption that the number of iteration cycles is of the same order as the processed number of prototypes, although it cannot be evidenced with it.

The computational effort for the prediction part of classification is of paramount interest in diagnostic applications. There are large speed differences between the methods. The corresponding results are depicted in Fig. 3 on a logarithmic time scale, therefore.

5 DISCUSSION

In this paper, a novel, universal distance-based classification procedure has been introduced. For each class to be recognised, this classifier computes a number of prototypes and corresponding cutoffs from

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

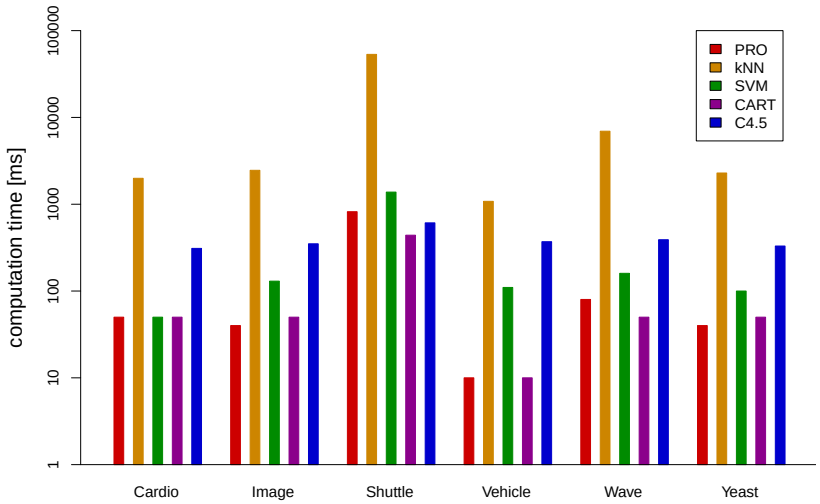

Figure 3. Computation time need of the prediction part.

the training data. The approximate solution of the underlying discrete optimisation problem has been realised by implementing a greedy algorithm which runs in $\mathcal{O}(k^2 n^2 log(n))$ time where k is the number of prototypes to be selected and n is the number of training objects.

In contrast to the concepts of centroids or medoids, it is not necessary that the centres of the hyperspheres coincide with regions of high point density. The multi prototype classifier models the shape of the points belonging to the class by an overlap of different-sized hypersheres. This global distance-based approach is universally applicable. It is based on a simple geometric model. No other assumptions, e.g. about the distribution of point distances or about clusters or subclusters, were made. Nevertheless, the choice of an appropriate distance measure is of great importance for the classification performance, as always in distance-based methods.

The presented method can be utilised in case of multi-class problems as well. For this purpose, a one against all approach has been performed. This makes a future extension of the entire diagnostic method for additional classes easily possible. An object has been finally assessed to that class of maximum positive predictive value when in doubt. Objects not recognised as a member of any of the classes have been assigned to an additional residual class.

For classification methods applied in diagnostics, the introduction of a residual class opens the chance to avoid unnecessary misdiagnoses and to examine those objects by other tools. This ensures a high level of confidence. Although the extensive evaluation of more pairwise classification results may enhance the overall performance in general (Allwein et al., 2000), the choice of a one against all approach seems to be sufficient. The results are comparable to those of other classification procedures and demonstrate excellent agreements of the class labels. Thus, multi prototype classifier can be used alternatively to other techniques.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

Classification results of some few methods applied to various data sets have been compared by the poor accuracy index, and by the adjusted Rand index. ACC is biased by differently sized classes. In general, ARI is a good and reliable measure of classification success and should be preferred. In the cases considered in this paper, however, both indices would not lead to dissent conclusions when used apartly.

The classification performances visualised in Fig. 1 vary with both, data sets and methods. So, the decision which procedure should be applied depends on the specific task. The results convey the impression that multi prototype classifier is comparable to the tools considered and works more or less effectively, like others else.

Time consumption is generally hard to compare between different computer programs. Each of them is possibly programmed by other persons, and in different contexts, and may differ in effectivity. This applies to the specific implementations of the classifiers described in Sect. 3.2, too. Furthermore, PRO is interpreting R-code, only. The other examined methods make partly use of compiled segments. The differences in computation time, as presented in Fig. 3, should consequently be interpreted with care. However, some patterns are expected and plausible.

For all data sets, kNN is the slowest prediction method. The distances of an investigated object to all objects of the reference set must be calculated for it. Just for that reason, PRO was developed. The prediction using C4.5, with the exception of shuttle data, was also relatively slow. This might be due to relatively large trees necessary for good classification. PRO, however, has proven to be one of the fastest methods. This is an important benefit for potential utilization such as in high-throughput diagnostics.

The findings of this exploration can be condensed in three main statements. Multi prototype classifier is a universal classification algorithm which is comparable to the other examined procedures concerning its overall performance. With the introduction of a residual class, PRO realises the opportunity to tag objects which cannot be classified at a sufficient level of confidence. The depicted multi prototype classification technique enables a rapid class prediction even for large amounts of data.

ACKNOWLEDGMENTS

The author would like to acknowledge UCI Machine Learning Repository (Bache and Lichman, 2013). These data sets are very helpful for developing, testing, and comparing algorithms for data analysis. Providing that material will also help readers to practically reproduce presented results.

REFERENCES

Allwein, E. L., Schapire, R. E., and Singer, Y. (2000). Reducing multiclass to binary: a unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1:113–141.

Ayres-de Campos, D., Bernardes, J., Garrido, A., Marques-de Sá, J., and Pereira-Leite, L. (2000). SisPorto 2.0: a program for automated analysis of cardiotocograms. *J Matern Fetal Med*, 9(5):311–8.

Bache, K. and Lichman, M. (2013). UCI machine learning repository.

Breiman, L., Friedman, J., Stone, C., and Olshen, R. (1984). *Classification and Regression Trees*. Chapman & Hall, New York.

Brodley, C. (1990). Statlog (Image Segmentation) Data Set. Vision Group, University of Massachusetts.

Brooks, A. D. (2007). *knnflex: A more flexible KNN*. R package version 1.1.1.

Catlett, J. (1994). Statlog (Shuttle) Data Set. Basser Department of Computer Science, University of Sydney, N.S.W., Australia.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. (2009). *Introduction to Algorithms*. The MIT Press, Cambridge, MA, USA, 3rd edition.

De Bruyne, K., Slabbinck, B., Waegeman, W., Vauterin, P., De Baets, B., and Vandamme, P. (2011). Bacterial species identification from MALDI-TOF mass spectra through data analysis and machine learning. *Syst Appl Microbiol*, 34(1):20–9.

Gordon, A. (1999). *Classification*. Monographs on Statistics and Applied Probability. Chapman & Hall/CRC.

Greiner, M., Pfeiffer, D., and Smith, R. (2000). Principles and practical application of the receiver-operating characteristic analysis for diagnostic tests. *Preventive Veterinary Medicine*, 45(1-2):23–41.

Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, 143(1):29–36.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.

Hong, S. P., Shin, S., Lee, E. H., Kim, E. O., Ji, S. I., Chung, H. J., Park, S. N., Yoo, W., Folk, W. R., and Kim, S. (2008). High-resolution human papillomavirus genotyping by MALDI-TOF mass spectrometry. *Nat. Protocols*, 3(9):1476–1484.

Hornik, K., Buchta, C., and Zeileis, A. (2009). Open-source machine learning: R meets weka. *Comput. Stat.*, 24(2):225–232.

Horton, P. and Nakai, K. (1996). A probabilistic classification system for predicting the cellular localization sites of proteins. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pages 109–115. AAAI Press.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

Horton, P. and Nakai, K. (1997). Better prediction of protein cellular localization sites with the k nearest neighbors classifier. In *Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology*, pages 147–152. AAAI Press.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2(1):193–218.

Karger, A., Ziller, M., Bettin, B., Mintel, B., Schares, S., and Geue, L. (2011). Determination of serotypes of shiga toxin-producing escherichia coli isolates by intact cell matrix-assisted laser desorption ionization-time of flight mass spectrometry. *Applied and Environmental Microbiology*, 77(3):896–905.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer, Berlin, 3rd edition.

Martinetz, T. M. and Schulten, K. J. (1991). A "neural-gas" network learns topologies. In Kohonen, T., M"akisara, K., Simula, O., and Kangas, J., editors, *Artificial Neural Networks*, pages 397–402. North-Holland, Amsterdam, North-Holland, Amsterdam.

Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., and Leisch, F. (2012). *e1071: Misc Functions of the Department of Statistics (e1071), TU Wien*. R package version 1.6-1.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

R Core Team (2012). *R: A Language and Environment for Statistical Computing*. R Core Team, Vienna, Austria. ISBN 3-900051-07-0.

Siebert, J. P. (1987). Vehicle recognition using rule based methods. Turing Institute Research Memorandum TIRM-87-018. Technical report, Turing Institute.

Tan, P., Steinbach, M., and Kumar, V. (2005). *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Therneau, T., Atkinson, B., and Ripley, B. (2012). *rpart: Recursive Partitioning*. R package version 4.1-0.

Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1073–1080, New York, NY, USA. ACM.

Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.*, 11:2837–2854.

Witten, I. H. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Wong, J., Schwahn, A., and Downard, K. (2010). FluTyper - an algorithm for automated typing and subtyping of the influenza virus from high resolution mass spectral data. *BMC Bioinformatics*, 11(1):266.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

Ziller, M., Gussmann, M., and Karger, A. (2011). Mathematical aspects of classifying mass spectra. diagnostic discrimination of influenza-subtypes. In Dress, A., Biebler, K.-E., Cieslik, D., and Spillner, A., editors, *The Math of Flu*, volume 17 of *Biometrie und Medizinische Informatik - Greifswalder Seminarberichte*, pages 101–123. Shaker, Aachen.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1180v1 | CC-BY 4.0 Open Access | rec: 15 Jun 2015, publ: 15 Jun 2015

Figure 1(on next page)

Figure 1

Adjusted Rand indices and accuracies of the classification results.

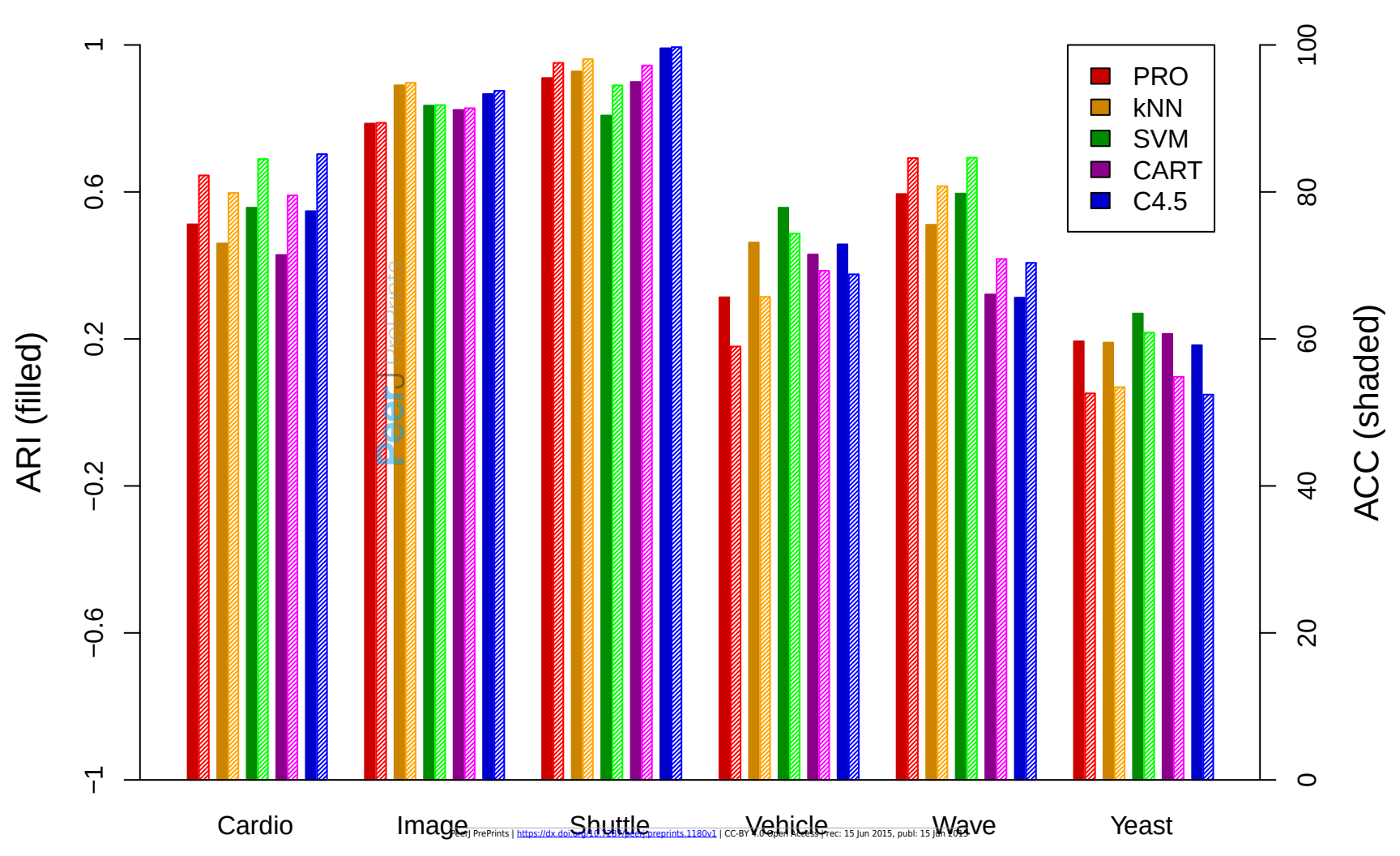

Figure 2(on next page)

Figure2

Number of iteration cycles of the milti prototype classifier.

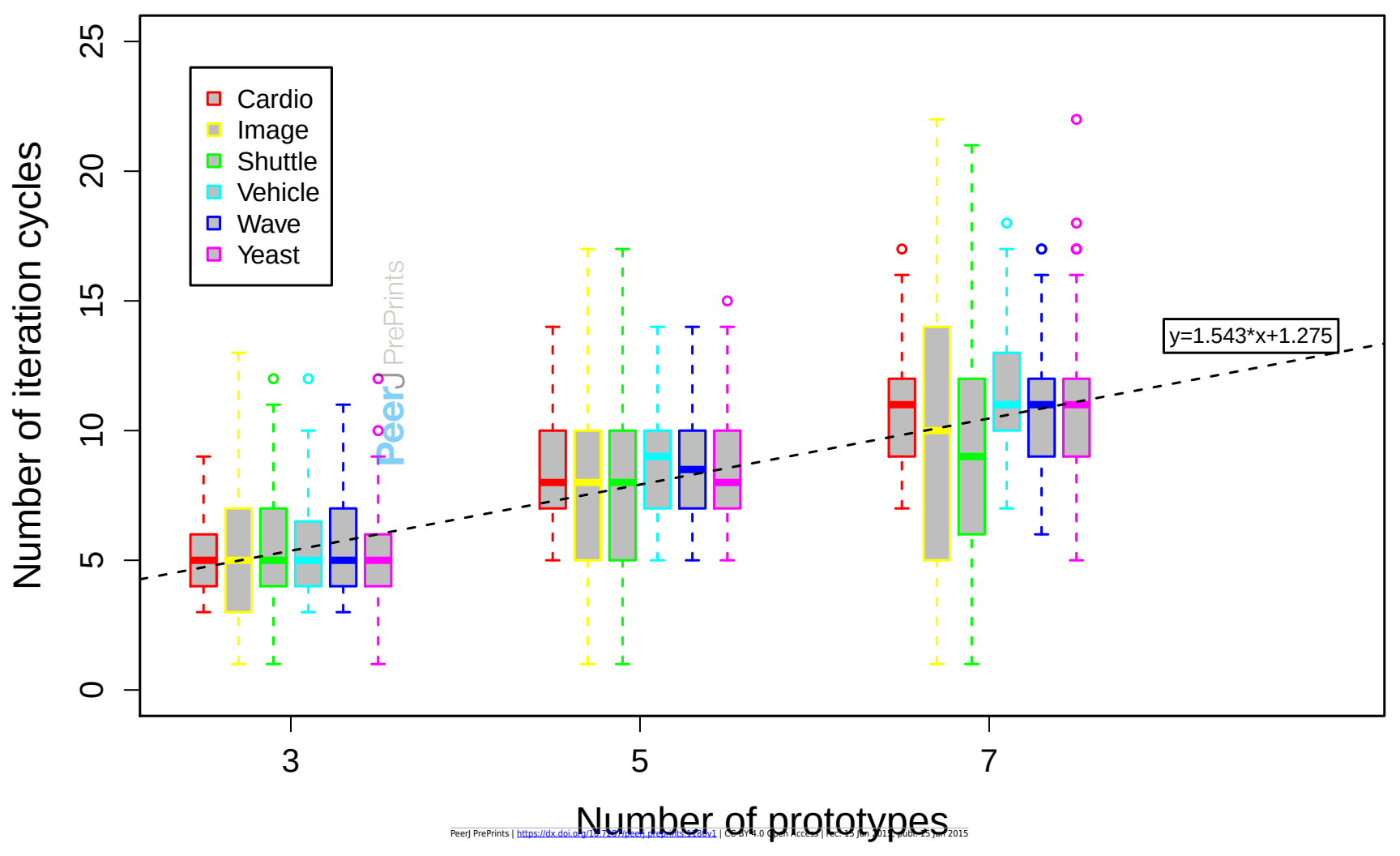

Figure 3(on next page)

Figure 3

Computation time need of the prediction part.