Fast and Accurate k-Nearest Neighbor Classification using Prototype Selection by Clustering

Stefanos Ougiaroglou, Georgios Evangelidis
Dept. of Applied Informatics
University of Macedonia
Thessaloniki, Greece
{*stoug, gevan*}*@uom.gr*

***Abstract*—Data reduction is very important especially when using the k-NN classifier on large datasets. Many prototype selection and generation algorithms have been proposed aiming to condense the initial training data as much as possible and keep the classification accuracy at a high level. The Prototype Selection by Clustering (PSC) algorithm is one of them and is based on a cluster generation procedure. Contrary to many other prototype selection and generation algorithms, its main goal is the fast execution of the data reduction procedure rather than high reduction rate. In this paper, we demonstrate that the reduction rate and the classification accuracy of PSC can be improved by generating a larger number of clusters. Moreover, we compare the performance of the particular algorithm with two state-of-the-art algorithms, one selection and one generation, using six real life datasets. The experimental results indicate that the classification performance of the Prototype Selection by Clustering algorithm is comparable with that of its competitors when using many clusters.**

***Keywords*-Classification; Clustering; k-Nearest Neighbors; Data Reduction; Prototype Selection and Generation;**

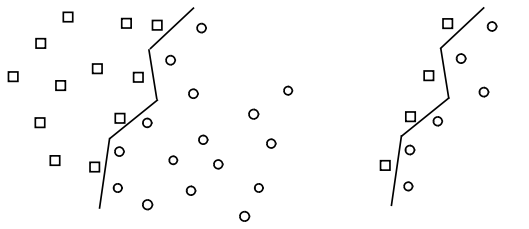
I. Introduction

The k-Nearest Neighbor (k-NN) classifier [6] is an effective lazy classifier. It classifies a new item by searching for its k nearest training items (neighbors) according to a distance metric. Then, the new item is classified to the most common class defined by the majority vote of its k nearest neighbors. Ties (two or more classes collecting the same number of votes) can be resolved by choosing either randomly one of the common classes or the class of the nearest neighbor.

The k-NN classifier is a widely-used classification algorithm because: (i) it is simple, (ii) it is easy to implement, and, (iii) it can be exploited in various application domains. On the other hand, since the distances between a new item and all items in the Training Set (TS) must be computed, the main drawback of the algorithm is the high computational cost that can render its execution prohibitive for large datasets. In other words, the computational cost depends on the size of TS. The more items in TS, the higher the cost involved.

The drawback of the high cost is an active research issue during the last decades. Many methods have been proposed that speed-up the nearest neighbor search. Particularly, the aforementioned drawback can be dealt with by using either a multi-attribute indexing method [27], [21] or a Data Reduction Technique (DRT)[1]. Contrary to indexing methods, DRTs have the extra benefit of the reduction of storage requirements. This work focuses on DRTs.

Data Reduction Techniques [24], [10], [23], [26], [16], [14], [12], [4], [18] are based on the following simple idea: they attempt to build a small set that represents the initial TS as accurately as possible. This small representative set is called Condensing Set (CS). Many DRTs build their CS using the following strategy. They consider that the items that lie in the "internal" data area of a class (i.e., far from decision borders) can be removed without significant loss of classification accuracy. Thus, they put in CS only the "useful" for classification data which are items that lie in the close-class-border data areas. Figure 1 depicts this strategy. The idea is that the k-NN classifier will be able to have similar classification accuracy using either TS or CS. However, a scan of CS involves much lower computational cost than that of TS.



(a) Training Set (b) Condensing Set

Figure 1: Data Reduction

DRTs are categorized into two main algorithm categories: (i) Prototype Selection (PS) [10], and (ii) Prototype Generation (PG) [24]. Both have as motivation the construction of CS. However, they differ on the way that this is achieved. PS algorithms select the close-border items and place them

[1]Data reduction has two points of view: (i) item reduction, and, (ii) dimensionality reduction. We consider them from the item reduction point of view.

in CS. On the other hand, PG algorithms generate new items by summarizing similar TS items. Efficient PG algorithms generate many items for the close-class-border data areas and few (or none) items for the non close-class-border data areas. We should mention that although editing algorithms, such as the ENN-rule [25], constitute a subcategory of PS algorithms, they have a different motivation. They attempt to improve the accuracy rather than reduce the computational cost by removing the noisy items and "smoothing" the decision borders. The reduction rate of DRTs depend mainly on the level of noise in TS. Their success usually implies the execution of editing before the application of the main data reduction routine [16]. The interested reader can find recent reviews with taxonomies and comparisons of PS and PG algorithms respectively in [10] and [24]. Other DRT reviews can be found in [23], [26], [16], [14], [12], [4], [18].

Lopez et al, have recently proposed a PS algorithm called Prototype Selection by Clustering (PSC) [19]. Its main goal is the fast construction of CS (low pre-processing cost). High reduction rate and accuracy continue to be desirable but constitute secondary goals. PSC is based on cluster generation. The main goal of PSC is achieved by the creation of a small number of clusters for each class. Of course, there are many other DRTs based on clustering. Some of them are the Self-Generating Prototypes (SGP) algorithms [9], the Prototype Generation and Filtering (PGF) [15] and the Symbolic Nearest Mean Classifier (SNMC) [7].

The motivation of this paper is to examine whether the creation of a larger number of clusters can improve the classification accuracy and the reduction rate of PSC algorithm as well as how the goal of low pre-processing cost is affected. The contribution of the paper is an extensive experimental study that compares our improved version of PSC with two state-of-the art DRTs, the PS algorithm CNN-rule [13] and the PG algorithm RSP3 [22].

The rest of this paper is organised as follows. Section II considers the CNN-rule as well as the family of RSP algorithms. Section III presents the PSC algorithm and explores how the construction of multiple clusters can improve its performance. Finally, Section IV presents the experimental results, and Section V concludes the paper and gives future directions.

II. State-of-the-art DRTs

A. Condensing Nearest Neighbor Rule

Hart proposed one of the first and well-known PS algorithms, the Condensing Nearest Neighbor (CNN) rule [13]. Many other approaches extend or are based on the idea of CNN-rule. Some known are: the Reduced NN rule [11], the Selective NN rule [20], the Modified CNN rule [8], the Fast CNN rule [3] and the IB algorithms [1].

However, CNN-rule remains the PS algorithm of reference until today and it is used in experimental studies for comparison purposes.

CNN-rule tries to remove the non-close-class-border (or "internal") items as follows. It uses two lists, S and T. Initially, a TS item is placed in S and all other items are placed in T. Then, the CNN-rule attempts to classify the items of T by scanning the items of S and applying the 1-NN rule. If an item is wrongly classified, it is moved to S. The procedure continues while there are moves from T to S. The final list S constitutes the CS.

The main idea of CNN-rule is that if an item is misclassified, it is close to a border data area and so it must be placed into the CS. Contrary to many other DRTs, CNN-rule determines the CS size automatically (i.e., without user-defined parameters).

B. Reduction by Space Partitioning

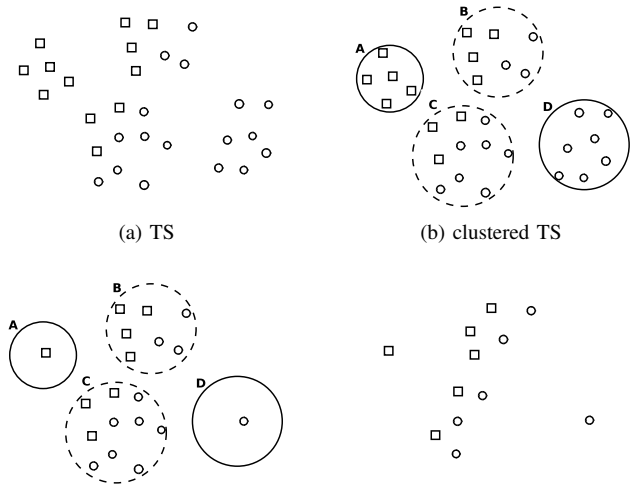
Chen and Jzwik [5] have proposed an effective PG algorithm which constitutes the ancestor of the Reduction by Space Partitioning (RSP) [22] algorithm family. We call it Chen and Jzwik Algorithm (CJA).

CJA initially retrieves the pair of the most distant items, X and Y in TS. These items define the diameter of the dataset. Then, TS is split into two subsets S_X and S_Y. The TS items closer to X are put in S_X, whereas the ones closer to Y are put in S_Y. The aforementioned splitting procedure is applied recursively on each created subset and stops when a predefined number of subsets is built. In the end, for each subset S, CJA generates an item C by averaging the items in S. C is labeled by the most common class in S. The user must determine the algorithm parameter that defines the number of prototypes that will be generated.

The RSP algorithm family includes three algorithms that are based on the CJA idea. RSP1 generates as many prototypes as the different classes in the subset. RSP1 computes a larger CS than that of CJA. However, it aspires to improve accuracy since it takes into account all TS items. RSP1 and RSP2 differ on how they select the next subset to be split. RSP1 uses the subset diameter as the splitting criterion, based on the idea that the subset with the larger diameter may include more items, and so, a higher reduction rate could be achieved. In contrast, RSP2 uses as its splitting criterion the highest overlapping degree. This criterion considers that the items that belong to a class must be as close to each other as possible. RSP3 iteratively splits all non-homogeneous subsets until they do not include items from other classes. RSP3 is the only non-parametric RSP algorithm. It automatically determines the size of CS. Considering RSP3 algorithms, we conclude that they generate few prototypes for representing non close-class-border areas, and many prototypes for representing close-class-border areas.

III. Prototype Selection by Clustering

Prototype Selection by Clustering (PSC) [19] is a recently proposed PS algorithm whose main goal is the fast execution



(a) TS (b) clustered TS

(c) finding non-border prototypes (d) finding border prototypes

Figure 2: Prototype Selection by Clustering

of the reduction procedure. In order to achieve that, it adopts the fast and well-known k-means clustering algorithm [17].

PSC is based on the idea that homogeneous clusters include items that lie in non-close-border data areas. On the other hand, non-homogeneous clusters include close-border items. Initially, PSC uses k-means clustering in order to partition the training data into clusters. For each homogeneous cluster, the nearest to the cluster mean item is put in CS. For each non-homogeneous cluster, the items that define the decision boundaries are placed into the CS.

More formally, PSC, initially, creates $|C|$ clusters, C_i where $i = 1, 2, \ldots, |C|$. For each homogeneous cluster C_i, PSC places the nearest item $p \in C_i$ to the cluster mean in CS. This item is the prototype that represents the whole data area of that cluster and is called a non-border prototype. For each non-homogeneous cluster C_i, PSC chooses a set of prototypes as follows. Initially, it finds the majority class T_M in C_i. Then, for each item $p_j \in T_i$, $i \neq M$, it puts in CS item $p_M \in T_M$ that is the nearest to $p_j \in T_i$. Also, it puts in CS, item $p_{C_i} \in T_i$ that is the nearest to p_M (p_{C_i} may be different than p_j). The prototypes collected from a non-homogeneous cluster are called border prototypes.

The PSC routine is summarized in figure 2. k-means identifies four clusters in TS. Clusters A and D are homogeneous. For these clusters, PSC keeps the nearest items to the cluster means as non-border prototypes. They represent the corresponding clusters data area. On the other hand, Clusters B and C are non-homogeneous. Thus, PSC analyzes the cluster items and keeps only the border prototypes by applying the methodology described in the previous paragraph.

Of course, the selected number of border and non-border prototypes depends on the number of clusters that are initially created ($|C|$). The higher the $|C|$ value, the more homogeneous clusters are generated and the more non-border prototypes are collected. In contrast, the larger the clusters, the more border prototypes selected and the lower reduction rate achieved. Lopez et al considered a small number of clusters in order to achieve fast execution of the algorithm. In particular, in their experiments [19], they built only $r \times j$, $j = 2, 4, \cdots, 10$, clusters, where r is the number of discrete classes. We claim that a larger number of clusters could improve the classification performance in terms of accuracy and reduction rate.

IV. Performance Evaluation

A. Experimental Setup

We compared the performance of CNN-rule, RSP3 and PSC by applying the k-NN classifier on the condensed sets they generate. In all cases, we used the k value that achieved the highest classification accuracy. Furthermore, we used a 5-cross validation schema on six real life datasets distributed by the KEEL Dataset Repository[2][2]. Thus, we run five training/testing set k-NN experiments for each dataset and each algorithm and we report the averages. Of course, only the TS was preprocessed by the reduction algorithms. We used the five already constructed pairs of Training/Testing splits hosted by the KEEL repository. The six datasets are summarized in Table I. None of them has missing values. All algorithm runs were executed on the original datasets, i.e., without normalization or any other data transformation. Moreover, we used euclidean distance as the distance metric.

Table I: Dataset description

dataset	Size	Attr.	Classes
Letter Recognition (LIR)	20000	16	26
Pen-Digits (PD)	10992	16	10
Landsat Satellite (LS)	6435	36	6
Shuttle (SH)	58000	9	7
Texture (TXR)	5500	40	11
Phoneme (PH)	5404	5	2

The three algorithms are compared by estimating three metrics: (i) Accuracy (Acc.), (ii) Reduction Rate (R.R.) and Processioning Cost (P.C.) in terms of million (M) distance computations (we counted the distances computed during the procedure of CS construction). For each dataset, we present one diagram for each metric. For PSC, we built 24 CSs. Each one was built by using different number of clusters, $k = r \times CL$ clusters, where r is the number of discrete classes. CL takes 25 different values: $CL = 2, 4, 6, 8, 10, 20, \cdots, 190, 200$. The x-axis of each comparison diagram represents the CL values. CNN-rule and RSP3 are not parametric approaches and so their performance is not affected when varying the CL value.

[2]http://sci2s.ugr.es/keel/datasets.php

B. Comparisons

Figures 3-8 present the comparison measurements of the three methods on the six datasets. Each figure includes three diagrams, one for each metric, i.e., Acc., R.R. and P.C. The Accuracy diagrams include one extra curve for the measurements achieved by the Conventional k-NN classifier (Conv-k-NN), i.e., k-NN over the original training data (without condensing).

In all cases 3-8, CNN-rule executed faster and achieved higher reduction rate than RSP3. On the other hand, with the exception of dataset PH, RSP3 achieved higher accuracy measurements than CNN-rule. In some cases, the accuracy of RSP3 is close to the one of Conv-k-NN.

With the exception of dataset PH (figure 8), the highest reduction rate in all datasets are achieved when $10 \leq CL \leq 50$. This means that the corresponding k-NN classifiers executed faster than the classifiers built using the rest of the CL values. In the case of PH, reduction rate continues to improve with higher CL values.

Moreover, in the cases of LIR (figure 3), PD (figure 4), LS (figure 5), and TXR (figure 7) datasets, for $CL \leq 50$, the preprocessing cost measurements were lower than or close to those of RSP3. In the case of dataset SH (figure 6), which is the largest one, RSP3 generates its CS at an extremely high computational preprocessing cost. This is the result of the farthest point computations in the subsets created during RSP3 execution.

In the cases of LS (figure 5) and SH (figure 6), PSC could not reach the accuracy levels of the other two methods, but it was quite close. In all other datasets (figures 3, 4, 7, 8), PSC achieved higher accuracy measurements than CNN-rule and RSP3. Furthermore, PSC classifiers with CSs built using CL values greater than 10, were more accurate and achieved higher reduction rate than those built by lower CL values (Lopez et al case [19]). However, the generation of these CSs was an "expensive" procedure since it computed more distances. For non-dynamic environments, there is no need for periodical CS reconstruction. Thus, we claim that these measurements may not be so significant since the CS is built only once.

We conclude that PSC is an adaptive algorithm that can be used either for fast CS generation but with lower reduction rate and accuracy (this is the scenario presented by Lopez et al), or for accurate and fast k-NN classification but with "expensive" CS generation. The desirable performance can be achieved by tuning the CL parameter.

V. Conclusions

In this paper, we presented and compared three known DRTs, namely, CNN-rule, RSP3, and PSC. In addition, we demonstrated how the creation of a large number of clusters can improve the performance of PSC. The experimental measurements derived by a cross-validation schema on six real life datasets indicate that PSC can reach and exceed the

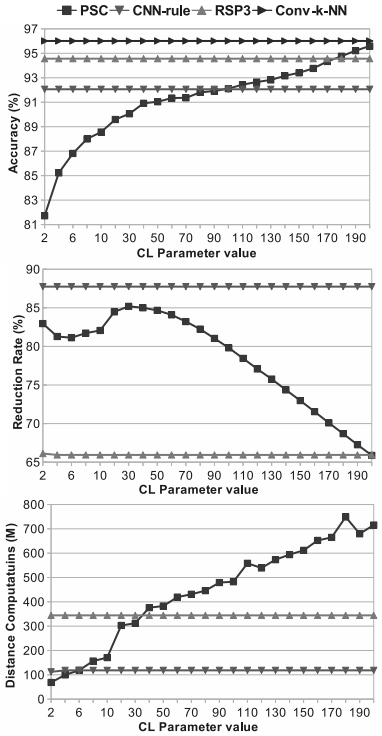


Figure 3: LIR (Acc., R.R., P.C.)

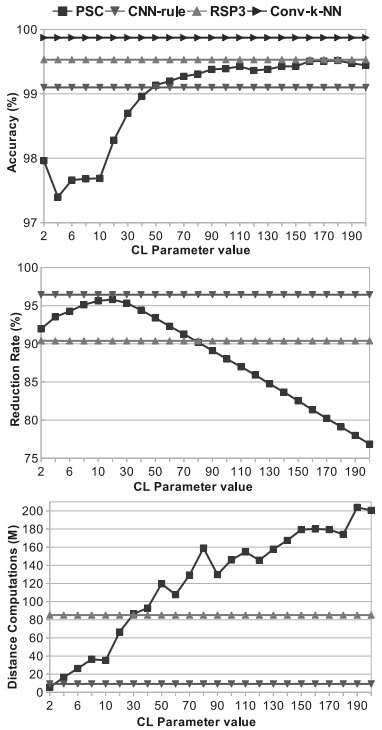


Figure 4: PD (Acc., R.R., P.C.)

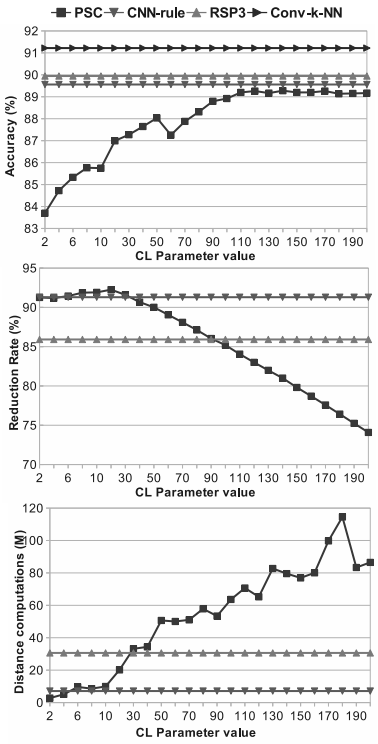


Figure 5: LS (Acc., R.R., P.C.)

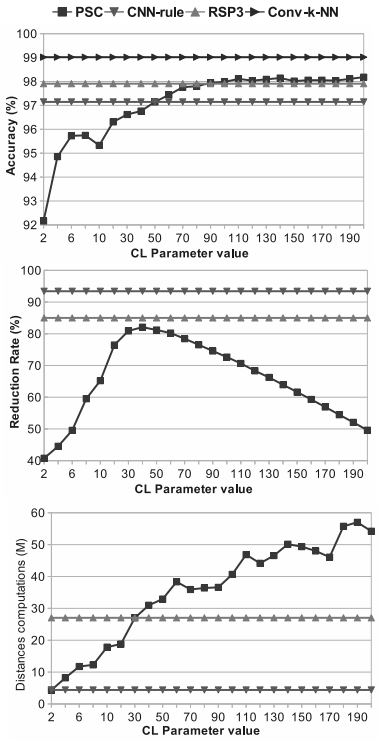


Figure 7: TXR (Acc., R.R., P.C.)

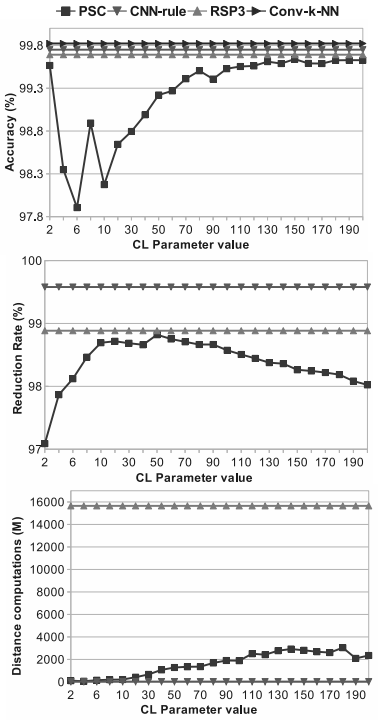


Figure 6: SH (Acc., R.R., P.C.)

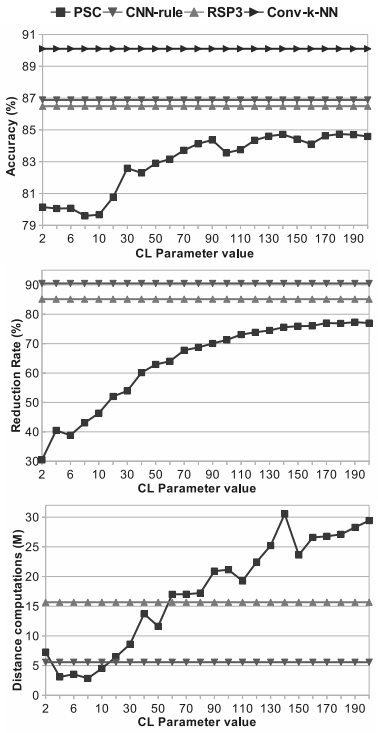


Figure 8: PH (Acc., R.R., P.C.)

classification performance of the other two state-of-the-art algorithms.

We plan to keep on examining the way clustering algorithms can be used for effective data reduction. In particular, we plan to develop fast and incremental data reduction algorithms that achieved high reduction rates without sacrificing the classification accuracy.

ACKNOWLEDGMENT

Stefanos Ougiaroglou is supported by a scholarship from the Greek State Scholarships Foundation (I.K.Y.)

REFERENCES

[1] D. W. Aha, D. F. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[2] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesús, S. Ventura, J. M. G. i Guiu, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera. Keel: a software tool to assess evolutionary algorithms for data mining problems. *Soft Comput.*, 13(3):307–318, 2009.

[3] F. Angiulli. Fast nearest neighbor condensation for large data sets classification. *IEEE Trans. on Knowl. and Data Eng.*, 19(11):1450–1464, Nov. 2007.

[4] H. Brighton and C. Mellish. Advances in instance selection for instance-based learning algorithms. *Data Min. Knowl. Discov.*, 6(2):153–172, Apr. 2002.

[5] C. H. Chen and A. Jóźwik. A sample set condensation algorithm for the class sensitive artificial neural network. *Pattern Recogn. Lett.*, 17:819–823, July 1996.

[6] B. V. Dasarathy. *Nearest neighbor (NN) norms : NN pattern classification techniques*. IEEE Computer Society Press, 1991.

[7] P. Datta and D. F. Kibler. Learning symbolic prototypes. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 75–82, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[8] V. S. Devi and M. N. Murty. An incremental prototype set building technique. *Pattern Recognition*, 35(2):505–513, 2002.

[9] H. A. Fayed, S. R. Hashem, and A. F. Atiya. Self-generating prototypes for pattern classification. *Pattern Recogn.*, 40:1498–1509, May 2007.

[10] S. Garcia, J. Derrac, J. Cano, and F. Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.

[11] G. W. Gates. The reduced nearest neighbor rule. ieee transactions on information theory. *IEEE Transactions on Information Theory*, 18(3):431–433, 1972.

[12] M. Grochowski and N. Jankowski. Comparison of instance selection algorithms ii. results and comments. In *Artificial Intelligence and Soft Computing - ICAISC 2004*, volume 3070 of *Lecture Notes in Computer Science*, pages 580–585. Springer Berlin / Heidelberg, 2004.

[13] P. E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 14(3):515–516, 1968.

[14] N. Jankowski and M. Grochowski. Comparison of instances seletion algorithms i. algorithms survey. In *Artificial Intelligence and Soft Computing - ICAISC 2004*, volume 3070 of *Lecture Notes in Computer Science*, pages 598–603. Springer Berlin / Heidelberg, 2004.

[15] W. Lam, C.-K. Keung, and C. X. Ling. Learning good prototypes for classification using filtering and abstraction of instances. *Pattern Recognition*, 35(7):1491 – 1506, 2002.

[16] M. Lozano. *Data Reduction Techniques in Classification processes (Phd Thesis)*. Universitat Jaume I, 2007.

[17] J. McQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of 5th Berkeley Symp. on Math. Statistics and Probability*, pages 281– 298, Berkeley, CA : University of California Press, 1967.

[18] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler. A review of instance selection methods. *Artif. Intell. Rev.*, 34(2):133–143, Aug. 2010.

[19] J. A. Olvera-Lopez, J. A. Carrasco-Ochoa, and J. F. M. Trinidad. A new fast prototype selection method based on clustering. *Pattern Anal. Appl.*, 13(2):131–141, 2010.

[20] G. Ritter, H. Woodruff, S. Lowry, and T. Isenhour. An algorithm for a selective nearest neighbor decision rule. *IEEE Trans. on Inf. Theory*, 21(6):665–669, 1975.

[21] H. Samet. *Foundations of multidimensional and metric data structures*. The Morgan Kaufmann series in computer graphics. Elsevier/Morgan Kaufmann, 2006.

[22] J. S. Sánchez. High training set size reduction by space partitioning and prototype abstraction. *Pattern Recognition*, 37(7):1561–1564, 2004.

[23] G. Toussaint. Proximity graphs for nearest neighbor decision rules: Recent progress. In *34th Symposium on the INTERFACE*, pages 17–20, 2002.

[24] I. Triguero, J. Derrac, S. García, and F. Herrera. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 42(1):86–100, 2012.

[25] D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE trans. on systems, man, and cybernetics*, 2(3):408–421, July 1972.

[26] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

[27] P. Zezula, G. Amato, V. Dohnal, and M. Batko. *Similarity Search - The Metric Space Approach*, volume 32. Springer, 2006.