Cluster-oriented instance selection for classification problems

Author

Saha, Soumitra, Sarker, Partho Sarathi, Al Saud, Alam, Shatabda, Swakkhar, Newton, MA Hakim

Published

2022

Journal Title

Information Sciences

Version

Accepted Manuscript (AM)

DOI

10.1016/j.ins.2022.04.036

Cluster-Oriented Instance Selection for Classification Problems

Soumitra Saha*, Partho Sarathi Sarker*

Department of Computer Science and Engineering, University of Global Village, C&B Road, Barishal-8200, Bangladesh

Alam Al Saud*, Swakkhar Shatabda*

Department of Computer Science and Engineering, United International University, Plot-2, United City, Madani Avenue, Badda, Dhaka-1212, Bangladesh

M A Hakim Newton*

School of Information and Physical Sciences, The University of Newcastle University Drive, Callaghan, NSW 2308, Australia
Institute for Integrated and Intelligent Systems, Griffith University, 170 Kessels Road, Nathan, QLD 4111, Australia

Abstract

More training instances could lead to better classification accuracy. However, accuracy could also degrade if more training instances mean further noises and outliers. Additional training instances arguably need additional computational resources in future data mining operations. Instance selection algorithms identify subsets of training instances that could desirably increase accuracy or at least do not decrease accuracy significantly. There exist many instance selection algorithms, but no single algorithm, in general, dominates the others. Moreover, existing instance selection algorithms do not allow direct controlling of the instance selection rate. In this paper, we present a simple and generic cluster-oriented instance selection algorithm for classification problems. Our proposed algorithm runs an unsupervised K Means Clustering algorithm on the training instances and with a given selection rate, selects instances from the centers and the borders of the clusters. On 24 benchmark classification problems, when very similar percentages of instances are selected by various instance selection algorithms, K Nearest Neighbours classifiers achieve more than 2%-3% better accuracy when using instances selected by our proposed method than when using those selected by other state-of-the-art generic instance selection algorithms.

Keywords: Instance Selection, Data Reduction, Classification Problems

*Soumitra Saha and Partho Sarathi Sarker have contributed equally and are joint first authors.

Email addresses: ssoumitra45@gmail.com (Soumitra Saha), parthosarker93@gmail.com (Partho Sarathi Sarker), alamalsaud5@gmail.com (Alam Al Saud), swakkhar@uiu.ac.bd (Swakkhar Shatabda), mahakim.newton@newcastle.edu.au,mahakim.newton@griffith.edu.au (M A Hakim Newton)

1. Introduction

Large datasets are very common in many application areas that include data mining, text categorization, financial forecasting, multimedia databases, genome sequences, and meteorological, financial, industrial and science repositories. Datasets comprise instances and instances comprise attributes. Machine learning algorithms use training instances from the datasets to train their learning models, which are then used in making predictions about testing instances. Classification problems deal with learning and prediction of classes associated with the instances. More training instances usually lead to better classification accuracy. For example, the asymptotic classification error of K nearest neighbours (KNN) algorithm [23] tends to the optimal Bayes error when the number of instances per class tends to infinity [18].

Unfortunately, with more data, outliers and noises contained in the training data tend to affect the performance more. Further, the large volume of data also needs more computational resources for storing, processing, and future analysis. To deal with these issues, *data reduction* techniques are often used: outliers and noises are removed and the most relevant part of the data is extracted. Data reduction could be performed by feature selection or instance selection algorithms, or both. For feature selection, techniques such as low variance filers, high correlation filters, principal component analysis (PCA), backward feature elimination, and forward feature construction methods are used. In this paper, we deal with instance selection for data reduction in classification problems.

Many instance selection algorithms have been developed so far. These algorithms have used condensation [11], elimination [21], editing [47], merging [9], selecting typical [49] or atypical [1] instances, clustering [31, 12], boosting [15], margin maximisation [15], representativeness [50], evidence theory [22], affinity propagation [43], global density [29], ranking [8], and local-sensitive hashing [4] techniques. Moreover, decremental optimisation algorithms [46], evolutionary algorithms [42], agent based algorithms [14], neural network based algorithms [24], and support vector machine (SVM) based algorithms [41] have also been developed for instance selection. All these algorithms have various advantages and disadvantages, but no single algorithm, in general, dominates the others. Moreover, existing instance selection algorithms do not allow direct controlling of the instance selection rate. So the quest for an effective instance selection algorithm keeps going.

In this paper, we present a simple and generic cluster-oriented instance selection (CIS) algorithm for classification problems. Given the number of clusters and a selection rate as parameters, CIS first runs unsupervised K Means Clustering (KMC) algorithm to identify clusters in the training instances and then using distance based measures, selects instances from the centers and the borders of the clusters. While central instances capture the cluster characteristics, the border instances protect each cluster from the other clusters. On 24 benchmark classification problems, when very similar percentages of instances are selected by various instance selection algorithms, K Nearest Neighbours (KNN) classifiers achieve more than 2%-3% better accuracy when using instances selected by our proposed method than when using those selected by the state-of-the-art generic instance selection methods. CIS could also select only 50% training instances losing just 1%–2% accuracy by Gaussian Naive Bayes (GNB) and Linear Support Vector Machine (LSVM) classifiers.

The rest of the paper is organised as follows: Section 2 covers preliminaries of machine learning and instance selection, Section 3 explores existing work related to instance selection, Section 4 describes our proposed CIS algorithm in details, Section 5 provides our experimental results, Section 6 presents our overall discussion and Section 7 concludes the paper.

2. Preliminaries

We briefly provide preliminaries of relevant machine learning and instance selection approaches.

2.1. Machine Learning Approaches

Let D be a dataset comprising $|D|$ *examples* or *samples* or *instances*. Also, let each instance $i \in D$ be a tuple $\langle a_1^i, \cdots, a_{|A|}^i \rangle$ comprising a set of $|A|$ attributes $A = \{a_1, \cdots, a_{|A|}\}$. Note that each instance can essentially be viewed as a point in an $|A|$ dimensional space.

Given a *training dataset* T comprising given *training instances*, a *hypothesis set* H comprising viable hypotheses, and a *performance metric* P to be optimised, a *machine learning algorithm* L finds a hypothesis $h \in H$ such that the performance metric P is optimised [6]. Hypothesis h is then used in making predictions about each instance i in a *testing dataset*.

A *supervised machine learning* algorithm strives to capture input-output relationships from *labeled data*, in which each training instance i is associated with an output o. An *unsupervised learning* algorithm strives to identify hidden functional relations from *unlabeled data*, in which each training instance i not associated with any output o beforehand. Both supervised and unsupervised learning algorithms strive to make predictions about the output of the testing instances since the outputs associated with the testing instances are not known and hence are required.

In classification based machine learning, functional outputs are classes and the performance metric is usually the classification accuracy. For an instance i, we denote its class by $c(i)$. Also, for a training dataset T, we denote the set $\{c(i) : i \in T\}$ of $|C|$ classes by C. SVMs and KNNs are supervised classifiers. For unsupervised learning of classification, usually clustering approaches are used. Clusters act as proxy classes. KMC is an unsupervised clustering algorithm.

2.2. Instance Selection Algorithms

Instance selection is essentially a *data reduction* method since instances are in effect data. Assume a training dataset T, a hypothesis set H, and a performance metric P. Also, assume a machine learning algorithm L finds a hypothesis $h_T \in H$ that achieves the best performance metric P_T. An *instance selection* algorithm selects a significantly small dataset $S \subset T$ such that the same machine learning algorithm L finds a hypothesis $h_S \in H$ that achieves a performance metric P_T, which is desirably better than P_T but at least not much worse than P_T [6].

In instance selection, a *distance measure* is normally used in determining the similarity between pairs of instances. In this paper, we use Euclidean distance measure. Given two instances i and i', their *Euclidean distance* $d(i, i') = \sqrt{\sum_{a \in A}(a^i - a^{i'})^2}$ is computed from their attribute differences.

2.3. K-Means Clustering Algorithm

KMC [28] is an unsupervised and iterative machine learning algorithm. Given a training dataset T and a predefined positive integer k, KMC partitions the instances in T into k non-overlapping clusters. The clustering procedure is as follows: start from k clusters denoted by k cluster centers that are chosen either randomly or from instances; then in each iteration, assign each instance to the cluster with the center closest to the instance and update the cluster centers by computing the means of the instances assigned to the cluster; stop the iterations, when there is no significant change in the cluster centers; return the clusters with the instances assigned to.

2.4. K Nearest Neighbours Algorithm

KNN [23] is a simple supervised machine learning algorithm. For a given positive integer k, keeping all training instances, KNN just assigns to a testing instance the output value most common among the top k nearest neighbours of the instance, where the proximity of the instances is measured by using a distance metric defined over the input attributes of the instances.

The performance of KNN depends on the value k. A large k generally reduces the effect of noises on the classification but makes boundaries between classes less distinct. A good k could be selected by heuristic algorithms or just by running KNN a few times with various k values.

KNN performs well with small data. However, for large data, KNN needs much memory to hold the entire training data and then needs much time to revisit each training instance to compute the distance from the given test instance. So to use KNN efficiently, data reduction techniques are usually sought to eliminate noises and outliers from the given data and to keep only the required and good data. Instance selection algorithms are thus relevant in this context.

3. Related Work

Instance selection and instance elimination are two complementary approaches to achieve data reduction. Given a training dataset T, an instance selection algorithm keeps a selected subset S of the training instances and thus discards the other $R = T \setminus S$ training instances. On the other hand, an *instance elimination* algorithm eliminates R training instances from T and thus keeps the remaining $S = T \setminus R$ training instances. Considering the fact that both instance selection and elimination algorithms lead to the same consequence of keeping S and discarding R and that sometimes such categorising is difficult, we cover both types of algorithm together.

3.1. Nearest Neighbour Based Algorithms

The nearest neighbour (NN) [11] algorithm is a KNN algorithm with $k = 1$. Both KNN and NN store all training instances. The condensed nearest neighbour (CNN) [11] algorithm iteratively selects S instances from T such that the correct prediction could still be made for each instance in T while using S. The reduced nearest neighbour (RNN) [21] algorithm iteratively eliminates instances from S of CNN and thus obtains $S' \subset S$ such that the correct prediction could still be made for each instance in T using S'. The edited nearest neighbour (ENN) [47] algorithm eliminates from T each instance for which the actual class does not match with the class predicted by using the rest of the instances in T. Another NN based algorithm [9] merges each two closest training instances of the same class into one instance by averaging such that each original instance in T could still be correctly predicted. The selective nearest neighbour (SNN) [35] selects the smallest S such that any instance in T is closer to its nearest neighbour in S than any instance of a different class. The instance based learning (IBL) algorithm [1] uses "concept descriptions" which are basically S. One IBL version iteratively stores in S from T only those instances that are incorrectly predicted by NN while using S. Another IBL version discards from S the instances that lead to incorrect predictions more than a given threshold. The typical instance based learning (TIBL) algorithm [49] stores instances that are "typical" in terms of *family resemblance* [36] measured as the ratio of intra-concept similarity to the inter-concept similarity; in this context, the IBL [1] actually stores "atypical" instances. The instance pruning techniques (IPT) [45] algorithm removes any instance i from T such that the prediction accuracy of the instances that have i as their nearest neighbour is not affected. The iterative case filtering (ICF) algorithm [5] removes an instance i when i has

more nearest neighbours from different classes than the number of instances that have i as their nearest neighbours from different classes. The SV-kNNC algorithm [41] uses SVM to eliminate training instances. The class conditional nearest neighbor (CCNN) [30] algorithm selects instances analysing a graph constructed by using relations among instance pairs $\langle i, i' \rangle$ such that i' is in the nearest neighbour of i among those instances (other than i) that are all in one of the classes in T.

The decremental reduction optimization procedure (DROP) [46] algorithms are variants of RNN and ENN algorithms mainly to fine tune and eliminate noises. Another KNN based instance selection method [15] uses boosting [33] and margin maximisation [30] after using ENN to remove noisy instances. The RIS [8] algorithm computes a ranking score for each instance $i \in T$ based on its distance $d(i, i')$ with each other instance $i' \in T$ and also based on whether $c(i)$ matches with $c(i')$; a positive score is given when $c(i) = c(i')$ and a negative score when $c(i) \neq c(i')$. RIS then iterates over all instances in T in the descending order of their ranking scores and includes in S an instance i such that there exists no instance $i' \in S$ with $c(i) = c(i')$ or if i is more than a certain distance away from any instance $i' \in S$ having $c(i) = c(i')$. Besides the original version RIS1, RIS has two other versions. One version (RIS2) performs scaling of values within the class while the original version (RIS1) performs scaling globally. Another version (RIS3) eliminates instances having very low scores and then performs selection. The RIS algorithms have been evaluated using KNN over 24 datasets. The representative-based instance selection (RBIS) algorithm [50] selects "representative" instances by measuring respresentativeness of the instances within their own classes and against other classes. There are two RBIS variants for balanced and imbalanced data and these variants are evaluated on a few datasets only. The evidential instance selection (EIS) algorithm [22] uses an evidence theory [39, 16, 17] to eliminate instances based on their degrees of conflicts with their neighbouring instances mainly to improve KNN performance.

3.2. Instance Clustering Based Algorithms

A focusing based instance selection algorithm [34] performs some kind of clustering of the instances and selects a 'leader instance' from each cluster such that the leader instance is within a certain distance from other instances in the cluster. A heuristic algorithm [13] computes similarity coefficients among the instances in T, then performs clustering of the instances using those coefficients, and selects a small number of instances from each cluster. A democratic algorithm [20] divides T into small disjoint subsets and run instance selection algorithms on each subset. The prototype selection by clustering (PSC) [31] algorithm performs costlier fuzzy C-means clustering of the instances. PSC then selects only one instance from a homogeneous cluster having all instances belonging to the same class. For a non-homogeneous cluster having instances from multiple classes, PSC finds the majority class and then for each instance in the other classes selects the nearest instance from the majority class and also the nearest instance from its own class. A cluster based instance selection algorithm [12] performs agent based optimisation search algorithm within each cluster. A two-step under-sampling approach called cluster-based instance selection (CBIS) [43] reduces data samples in a majority class. CBIS uses affinity propagation as clustering algorithm in the first phase and then uses existing instance reduction algorithms to reduce instances.

A parameter-free hybrid instance selection algorithim based on local sets with natural neighbours (LSNaNIS) [27] removes internal noises from classes, condenses internal samples, performs smoothing of class boundaries, and retains border samples. The global density-based instance selection (GDIS) [29] algorithm uses certain global density and relevance functions to select instances are in the borders of the classes. The enhanced global density-based instance selection (EGDIS)

improves GDIS by improving the reduction rate. The border point extraction based on local-sensitive hashing (BPLSH) [4] algorithm uses locality-sensitive hash functions to identify instances that are in the same classes and then to keep instances from the class boundaries and eliminates nonessential instances to accelerate the training process for SVM-based classifiers. A three way clustering based approach was proposed in a recent work [40] finding a compact representation of the data in the clusters. In another recent work [48], the authors have used multi-view based method for instance retrieval within a self-weighted learning framework.

3.3. Search Based Optimisation Algorithms

Instance selection could be viewed as a search-based optimisation problem. Consequently, evolutionary algorithms have been developed for instance selection [19, 42]. Also, a multi-objective evolutionary algorithm[10] and a memetic algorithm [26] comprising a single-point memetic structure and an accelerated local search procedure have developed. Moreover, agent based population algorithms [14, 12] perform both instance selection and elimination. A fuzzy clustering based approach uses genetic algorithms for instance selection [25].

3.4. Machine Learning Based Algorithms

The neural network ensemble editing (NNEE) [24] algorithm uses an ensemble of neural networks to eliminate "suspicious" instances [38]. A filtering-based instance selection algorithm (FIS) [37] removes unusual instances using self-organising maps and neural networks and thus mapping instances to a reduced lower dimensional space to find closeness or similarity. Manifold reduction methods are often used to deal with the curse of dimensionality [3, 44].

3.5. Proposed CIS vs Existing Algorithms

Our proposed CIS is an instance clustering based algorithm. CIS uses KMC where k is typically $|C|$. CIS then sorts the instances within each cluster based on their distances from the centers of the respective clusters. Finally, with a given selection rate s, CIS selects instances that are close to the centers of the clusters and that are in the border regions of the clusters.

Our proposed CIS is generic over classifiers and we show its effectiveness over three classifiers such as KNN, GNB, and LSVM. Compared to other clustering based instance selection algorithms, CIS is simpler as it just adopts a very simple way to include or exclude instances from clusters obtained by the K-Means clustering algorithm. Moreover, unlike other instance selection or reduction algorithms, CIS is more flexible as it allows direct controlling of the instance selection or reduction rate and thus also provides a way to trade off with the classification accuracy level.

4. Proposed Method

We describe our CIS algorithm and show its time and memory complexity. We also show its performance visually using a toy example.

4.1. Algorithm Details

The main idea of our CIS is to exploit the tacit relation between classes and clusters. So we perform clustering of the training instances. Then, we want to capture cluster characteristics. The instances around the cluster centers usually do that. Also, we want to protect the clusters from other clusters. The instances towards the border of the clusters help achieve that. However, the

instances at the middle of the clusters are normally covered by the center and border instances, and so we do not select them. Consequently, our instance selection algorithm has two steps: clustering and selection. We describe the steps below. Algorithm 1 shows the pseudocode of our CIS algorithm.

Algorithm 1 Cluster Oriented Instance Selection

Input: training instances T, number of clusters k, selection rate s
Output: a subset S of selected training instances from T

1: $Q \leftarrow \mathsf{Kmeans}(T, k)$
2: $S \leftarrow \emptyset$ // initially empty
3: **for each** cluster q in Q **do**
4: $\quad n \leftarrow$ number of instances in q
5: $\quad j \leftarrow$ the center of the cluster q
6: $\quad I \leftarrow$ set of instances in cluster q
7: $\quad d_i \leftarrow d(i, j)$ where each $i \in I$
8: $\quad L \leftarrow$ sort I in ascending order of d_i
9: $\quad S_{\mathsf{center}} \leftarrow$ first $\frac{1}{2}ns$ instances from L
10: $\quad S_{\mathsf{border}} \leftarrow$ last $\frac{1}{2}ns$ instances from L
11: $\quad S \leftarrow S \cup S_{\mathsf{center}} \cup S_{\mathsf{border}}$
12: **end for**

Clustering. Given a number k of clusters, in Line 1 of Algorithm 1, we run KMC on T. We typically use c, the number of classes in T, as the value of k. For hyper-spherical classes, the number of classes is a good choice. For irregular-shaped classes, or when classes are split geometrically, larger k values might lead to better performance. We run experiments to try larger k values.

Selection. All lines except Line 1 in Algorithm 1 are related to the selection phase. As we see form the algorithm, for each cluster q found by KMC, we sort the n instances in the cluster q in the order of their distances from the cluster center j. Then, for a given selection rate $s \in [0, 1]$, we take $\frac{1}{2}ns$ instances having the smallest distances from the cluster center and further $\frac{1}{2}ns$ instances having the largest distances from the cluster center.

4.2. Complexity Analysis

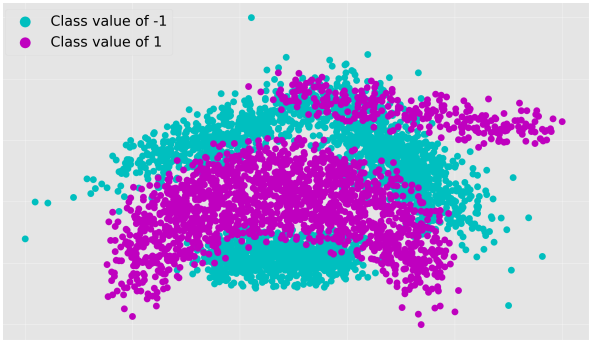
The following lemma describes the time and memory complexity of the CIS algorithm.

Threorem 1. *The proposed CIS algorithm described in Algorithm 1 has a time complexity of $\mathcal{O}(klmn)$ and a space complexity $\mathcal{O}(mn)$ where k is the number of clusters to be found by K-Means clustering algorithm, l is the number of iterations K-Means clustering algorithm will run for, m is the number of attributes in each instances, and n is the number of instances.*

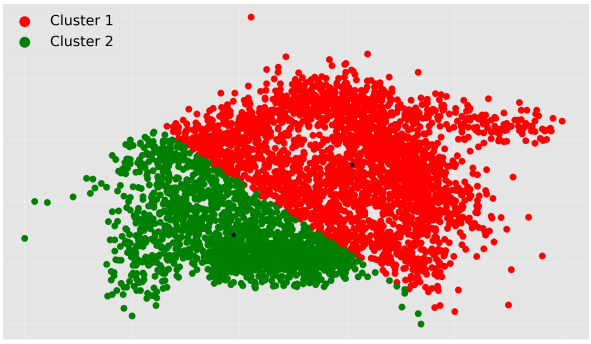
Proof. Running K-Means algorithm in Algorithm 1 has a time complexity of $\mathcal{O}(klmn)$ and a memory complexity of $\mathcal{O}(mn)$. The for loop in Algorithm 1 runs for k times but combining all laps together, sorting all instances in the clusters altogether needs $\mathcal{O}(n \log n)$ time and $\mathcal{O}(n)$ memory. □

4.3. A Toy Example

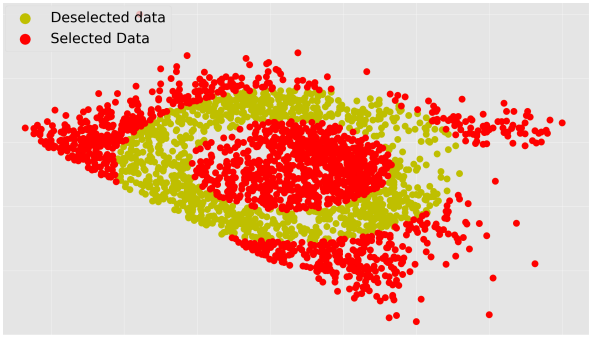
For visualisation of instance selection by our proposed CIS algorithm, we use the banana dataset as a toy example. The banana dataset is collected from the KEEL repository [2]. It has two attributes, two classes and 5300 instances. Figure 1 (a) shows the original dataset while Figure 1 (b) shows the instances after running KMC. Figure 1 (c) and (d) show the CIS selected instances from the two clusters respectively. Finally, Figure 1 (e) shows all CIS selected instances while Figure 1 (f) shows all RIS selected instances from the dataset. RIS [8] is a recent ranking-based instance selection algorithm and has been evaluated on the same 24 datasets that we use. From the figures, it is clear that CIS selects instances only from the centers and the borders of the clusters while RIS selects instances from all over the dataset.



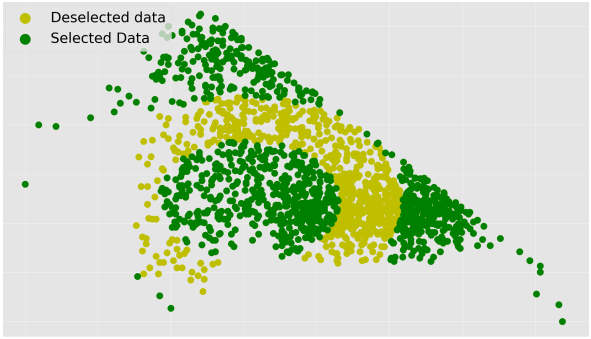
(a) all training instances with actual classes



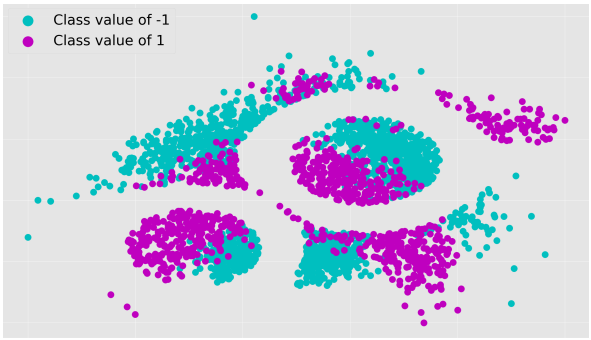
(b) all training instances after clustering



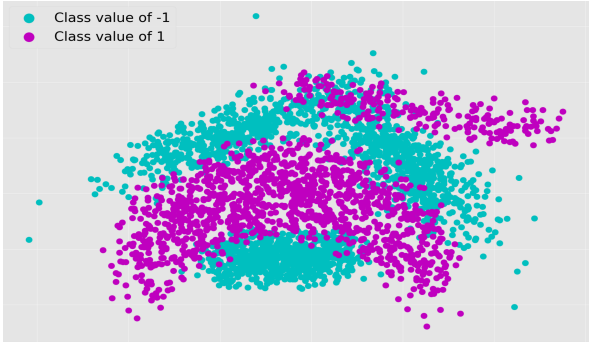
(c) CIS selected instances from cluster 1



(d) CIS selected instances from cluster 2



(e) instances selected by CIS



(f) instances selected by RIS

Figure 1: Instances selected by CIS and RIS in the Banana dataset

5. Experimental Results

We have implemented the CIS algorithm using `Python 3.8` and `scikit-learn 0.23.1` [32]. The parameters of the algorithms from the `scikit-learn` are kept default unless specifically mentioned. No hyper-parameter tuning has been performed on the classification algorithms or models. All experiments have been run on a computer with 2-core Intel(R) Core(TM) i3 CPU @ 2.10GHz, 12GiB memory, and Ubuntu 20.04 operating system.

We compare CIS with KNN based algorithms RIS [8] and EIS [22], and clustering based algorithms GDIS [29] and EGDIS [29]. RIS have been evaluated on 24 datasets. EIS, GDIS, and EGDIS uses a number of datasets but 17 of them are in the datasets used by RIS. We do not compare with KNN based algorithm RBIS [50] since it was evaluated only on a few datasets. We do not compare with clustering based algorithms CBIS [43], LSNaNIS [27], and BPLSH [4]. CBIS has been evaluated with multi-layer perceptron (MLP) based ensemble classifiers. LSNaNIS uses datasets that have only 4 datasets common with the 24 datasets that we use. BPLSH mainly works with SVM but CIS is generic over classifiers. Note that we do not compare CIS with any search based or machine learning based algorithms since those algorithms are quite different types compared to the cluster oriented CIS algorithm.

We further describe benchmark datasets, experimental settings, and results and analyses.

5.1. Benchmark Datasets

In our experiments, we have used 24 datasets that are listed in Table 1. These datasets have been taken from KEEL repository [2]. These datasets are widely used in evaluating instance selection and elimination algorithms that include RIS [8], EIS [22], GDIS [29], and EGDIS [29]. As we see, these datasets vary in the numbers of instances, attributes, and classes.

5.2. Experimental Settings

Considering the most number of 24 datasets common with RIS [8], we have followed the same experimental setup used by RIS. In RIS experiments, for each of the 24 datasets, 10 folds have been created from the original set of instances. Then, in each of 10 runs for each dataset, 1 designated fold out of the 10 folds is used in testing and the rest 9 folds are collectively used in training. We have obtained the exact instances in each of the 10 folds used by RIS in each dataset. So essentially all results reported in this paper could be directly compared with the results provided by RIS [8].

Accuracy Computation. The classification accuracy in each run is computed over the designated fold used in testing in that run and the mean accuracy for the dataset is computed over the 10 runs. To evaluate a given instance selection algorithm, the same procedure is repeated again, but in each of the 10 runs for a dataset, the instance selection algorithm is used on the combined training instances from the other nine folds and only the selected instances are used in training while the testing instances in the designated fold remain the same. The procedure to compute accuracy in each run of the dataset and the mean accuracy over 10 runs of the dataset remains the same.

Note that while CIS uses the term instance *selection rate* s, RIS uses instance *reduction rate* r. Actually, instance reduction rate r is complementary to instance selection rate s since $r + s = 1$. In this paper, we show both r and s in percentages. We show accuracy in percentage as well.

Henceforth, we use NoSel to denote no instance selection algorithm has been used.

Table 1: A summary of the datasets used in the experiments.

ID	Dataset	Instances	Attributes	Classes
1	adult	45,222	14	2
2	appendicitis	106	7	2
3	balance	625	4	3
4	bupa	345	6	2
5	coil2000	9822	85	2
6	connect-4	67,557	42	3
7	contraceptive	1473	9	3
8	haberman	306	3	2
9	hayes-roth	160	4	3
10	heart	270	13	2
11	ionosphere	351	33	2
12	led7digit	500	7	10
13	marketing	6876	13	9
14	monk-2	432	6	2
15	movement-libras	360	90	15
16	pima	768	8	2
17	satimage	6435	36	7
18	segment	2310	19	7
19	titanic	2201	3	2
20	vowel	990	13	11
21	wine	178	13	3
22	winequality-red	1599	11	6
23	winequality-white	4898	11	7
24	yeast	1484	8	10

5.3. Comparison with RIS Variants

We compare CIS with the recent algorithm RIS [8]. RIS has been shown to be significantly better than ENN [47], DROP3 [46], and ATISA1 [7]. Nevertheless, Table 2 shows the maximum mean accuracy values achieved by KNN over $k \in \{1, 3, 5\}$ when NoSel, CIS and RIS algorithms are used. Table 2 also shows instance reduction rates of the RIS versions.

As mentioned before, RIS has three versions: RIS1, RIS2, and RIS3. We have obtained RIS executable programs and have run ourselves on the 24 datasets. As per Table 2, RIS versions help KNN achieve accuracy values 67.18%–70.10% with reduction rates 54.17%–66.26%. Among three RIS versions, RIS1 helps KNN obtain the best mean accuracy value with the least mean reduction rate. RIS2 and RIS3 have higher reduction rates but help achieve lower accuracy values than what RIS1 does. So, RIS1 has been selected as the best RIS version [8].

Our CIS algorithm has a parameter s that is the rate of instance selection. We create three CIS versions CIS50, CIS60, and CIS70 with instance selection rates 50%, 60%, and 70% respectively. As such CIS50, CIS60, and CIS70 have instance reduction rates 50%, 40%, and 30% respectively. CIS has another parameter k for the number of clusters to be obtained by KMC. In this experiment, for each dataset, we have used the number of classes $|C|$ as the value of k. As per Table 2, CIS versions help KNN achieve accuracy values 77.79%–80.50% with reduction rates 30%–50%. Among

Table 2: Maximum mean accuracy values (%) obtained by KNN over $k \in \{1, 3, 5\}$ with NoSel, CIS and RIS algorithms. Also, the instance reduction rates (%) of the RIS versions. The emboldened value for each dataset is the best accuracy. Both for accuracy and reduction rates, the higher the better.

Dataset		Maximum Mean KNN Accuracy over $k \in \{1, 3, 5\}$							Reduction Rate		
ID	Name	NoSel	CIS50	CIS60	CIS70	RIS1	RIS2	RIS3	RIS1	RIS2	RIS3
1	adult	77.06	77.01	78.89	**82.32**	75.43	77.09	77.13	47.25	47.25	47.25
2	appendicitis	86.64	86.72	86.73	90.64	84.70	**92.26**	**92.26**	70.64	72.42	72.84
3	balance	76.80	88.00	89.12	**90.25**	89.93	87.70	85.95	76.53	84.76	90.91
4	bupa	66.94	65.44	**66.30**	66.03	61.43	62.31	62.34	43.83	50.18	56.14
5	coil2000	93.47	93.09	93.24	93.40	**94.03**	87.78	87.77	31.01	27.10	25.11
6	connect-4	58.11	81.21	83.82	**87.03**	65.83	65.59	65.80	52.81	46.74	95.3
7	contraceptive	51.19	65.58	67.88	**68.90**	49.15	48.95	49.15	26.14	38.36	41.37
8	haberman	71.90	73.23	72.24	72.24	**74.16**	70.23	69.62	49.20	58.02	64.78
9	hayes-roth	76.88	71.88	75.00	**77.50**	71.94	66.40	64.44	44.79	48.42	51.47
10	heart	66.67	88.52	92.22	**93.70**	82.59	84.07	85.19	50.99	61.77	77.86
11	ionosphere	85.18	97.72	98.00	**98.29**	92.89	90.92	91.22	75.50	74.64	74.89
12	led7digit	70.60	69.60	67.40	69.60	**73.34**	68.84	76.44	16.53	57.76	92.53
13	marketing	29.74	56.24	61.20	**66.09**	18.34	15.53	16.06	82.96	72.30	50.43
14	monk-2	96.55	93.50	96.05	**97.45**	92.11	92.84	94.00	72.79	83.04	87.27
15	movement-libras	83.61	84.44	86.11	**89.17**	80.11	60.56	59.22	62.28	70.73	72.58
16	pima	71.74	73.43	74.09	**74.87**	69.00	68.08	67.57	53.78	52.82	53.27
17	satimage	89.00	92.52	94.17	**95.70**	25.13	18.31	18.20	52.74	60.82	13.76
18	segment	96.28	94.11	95.93	**96.84**	92.60	91.82	91.65	88.96	90.51	91.51
19	titanic	76.47	**75.19**	73.60	72.19	69.06	56.75	56.21	17.61	34.28	68.34
20	vowel	63.74	81.92	83.84	87.87	**87.88**	66.06	65.56	76.90	83.45	84.39
21	wine	74.18	98.33	98.33	**98.89**	93.95	93.32	97.25	86.64	86.15	87.83
22	winequality-red	44.96	61.61	61.61	**63.10**	51.46	48.67	50.61	42.01	64.83	69.35
23	winequality-white	40.92	44.57	45.14	**46.26**	45.06	43.39	42.92	41.83	60.69	62.91
24	yeast	53.64	53.10	52.63	**53.58**	42.37	46.85	45.75	36.27	56.02	60.57
Mean		70.93	77.79	78.90	**80.50**	70.10	66.85	67.18	54.17	61.70	66.26

three CIS versions, CIS70 obtains the highest accuracy value with the least reduction rate. As the mean reduction rates increase from CIS70 to CIS60 to CIS50 by 10% each time, the accuracy values gradually decrease by 1.1%–1.6% each time, making the total decrease of 2.71% in accuracy from CIS70 to CIS50. The highest decreases in accuracy from CIS70 to CIS50 are about 6% in `vowel` and `connect-4` datasets. Interestingly, in CIS50 compared to CIS70, the accuracy values actually increase by about 3% and 1% in `titanic` and `haberman` datasets respectively. However, in `coil2000` and `led7digit` datasets, reduction rates do not affect the accuracy values much.

When NoSel algorithm is used, Table 2 shows KNN achieves the mean accuracy value 70.93%. Compared to this accuracy obtained using NoSel, all three CIS versions help obtain at least 5.86% larger accuracy values while all three RIS versions cause KNN to degrade accuracy values by 0.83% to 3.75%. Overall, CIS50, CIS60, and CIS70 help KNN improve accuracy values in 15, 16, 19 datasets respectively while RIS1, RIS2, and RIS3 help improve in 11, 10, 11 datasets respectively. Moreover, as per the emboldened values in Table 2, CIS versions help KNN obtain the best accuracy values in 19 out of 24 datasets while RIS version do so only in 5 datasets.

Table 3: Wilcoxon signed rank test p-values. Underlined p-values denote pairwise accuracy differences are not statistically significant at 95% confidence level.

	CIS50	CIS60	CIS70	RIS1	RIS2	RIS3
NoSel	0.00964	0.00427	0.00027	0.43644	0.20897	0.26435
CIS50		0.00205	0.00013	0.00135	0.00006	0.00024
CIS60			0.00008	0.00050	0.00005	0.00018
CIS70				0.00006	0.00006	0.00004
RIS1					0.00695	0.06426
RIS2						0.47608

We perform Wilcoxon signed rank test with 95% confidence level on the accuracy values obtained by KNN (as shown in Table 2) when NoSel, CIS, and RIS algorithms are used. The p-values for pair-wise comparisons are shown in Table 3. As we see from the top row, CIS versions help KNN obtain significantly better accuracy values than what NoSel algorithm does. However, the differences of accuracy values due to the RIS versions from the the accuracy values due to NoSel algorithm are statistically not significant. Next from middle three rows in Table 3 we see, KNN accuracy values due to all CIS versions are statistically different from each other and also from the accuracy values due to all RIS versions. However, from last two rows in Table 3 we see among RIS versions, KNN accuracy values due to RIS1 and RIS2 are statistically different, but the accuracy value due to RIS3 is not statistically different from accuracy values due to RIS1 and RIS2.

We now compare CIS50 and RIS1 head to head. Note that CIS50 is the worst CIS version while RIS1 is the best RIS version in terms of the accuracy values obtained by KNN while using the respective instance selection algorithms. Also, note that the reduction rates for CIS50 and RIS1 are 50% and 54.17% respectively and so are very close. Table 4 shows CIS50 helps obtain better accuracy values in 17 out of 24 datasets while RIS1 does so in 7 datasets. The mean accuracy difference is 7.69% in favor of CIS50 with respect to RIS1. As per Table 3, this difference is statistically significant. CIS50 leads to $> 10\%$ better accuracy values than RIS1 in `connect-4`, `contraceptive`, `marketing`, `satimage`, `winequality-red`, and `yeast` datasets but RIS1 leads to about 6% better accuracy value in `vowel` dataset. We conclude CIS50 greatly outperforms RIS1.

Although RIS1 has a mean instance reduction rate of 54.17% (see Table 2), the reduction rates actually vary from 16.53% for `led7digit` to 88.96% for `segment` dataset. On the other hand, CIS50 has a reduction rate of 50% regardless of the dataset. So for a more vis-a-vis comparison, we have run a CIS version named CISR on each dataset with the instance reduction rate the same as RIS1 has in that dataset. Table 4 shows CISR helps obtain better accuracy values in 16 out of 24 datasets while RIS1 does so in 8 datasets. The mean accuracy difference is 6.36% in favor of CISR with respect to RIS1. As per Table 4 bottom row, this difference is statistically significant at 95% confidence level of the Wilcoxon signed rank test. CISR leads to $> 10\%$ better accuracy values in `connect-4`, `contraceptive`, `marketing`, `satimage`, `winequality-red`, and `yeast` datasets while RIS1 leads to $> 10\%$ better accuracy value in `vowel` dataset. We conclude with CISR's superiority over RIS1. Notice that as per the last row of Table 4, the accuracy difference achieved due to CISR is not statistically different from CIS50 at 95% confidence level of the Wilcoxon signed rank test.

Overall, our conclusion is that CIS is significantly better (at least 6%) in accuracy than RIS when mean reduction rates are the same or similar at about 50%–54.17%. The accuracy values obtained by using CIS could be further increased if the reduction rates are further reduced or in

Table 4: Maximum mean accuracy values (%) obtained by KNN over $k \in \{1, 3, 5\}$ with CIS50, RIS1, and CISR where CISR is CIS with the reduction rate the same as RIS1 has in each dataset. The underlined values are the best accuracy values between CIS50 and RIS1 columns while the emboldened values are the best accuracy values between RIS1 and CISR columns. Bottom line: Wilcoxon signed rank test p-values for CIS50 and RIS1 against CISR.

ID	Dataset	CIS50	RIS1	CISR
1	adult	77.01	75.43	**78.07**
2	appendicitis	86.72	84.70	**85.73**
3	balance	88.00	**89.93**	86.10
4	bupa	65.44	61.43	**66.31**
5	coil2000	93.09	**94.03**	93.38
6	connect-4	81.21	65.83	**80.50**
7	contraceptive	65.58	49.15	**69.51**
8	haberman	73.23	**74.16**	73.23
9	hayes-roth	71.88	71.94	**75.63**
10	heart	88.52	82.59	**90.00**
11	ionosphere	97.72	**92.89**	92.30
12	led7digit	69.60	**73.34**	71.00
13	marketing	56.24	18.34	**43.05**
14	monk-2	93.50	**92.11**	88.15
15	movement-libras	84.44	80.11	**82.50**
16	pima	73.43	69.00	**72.92**
17	satimage	92.52	25.13	**93.24**
18	segment	94.11	**92.60**	88.01
19	titanic	75.19	69.06	**74.51**
20	vowel	81.92	**87.88**	73.03
21	wine	98.33	93.95	**97.78**
22	winequality-red	61.61	51.46	**61.55**
23	winequality-white	44.57	45.06	**45.61**
24	yeast	53.10	42.37	**52.83**
Mean		77.79	70.10	**76.46**
Wilcoxon signed rank test		0.13786	0.01539	p-value

other words, the selection rates are further increased. Henceforth, in our further experiments, we use CIS50.

5.4. Comparison with GDIS, EGDIS, and EIS

Table 5 shows the mean accuracy values and reduction rates obtained by KNN with $k = 3$ when using CIS50, GDIS [29], EGDIS [29], and EIS [22]. Note that we take the results of GDIS, EGDIS, and EIS from the publication of EIS which includes 17 out of 24 datasets used by CIS and RIS. As we see from the table, EIS obtains the best accuracy values in 9 out 17 datasets and best reduction rates in 13 datasets. However, note that EIS is specifically designed for KNN and also it optimises its instance selection using the value of k and performing a grid search over its algorithmic parameters. This somewhat explains the higher performance of EIS with KNN with a specific k over the other instance selection algorithms. In contrast, CIS, GDIS, and EGDIS are generic over any classification algorithm. So in Table 5, we also observe the performance among

these three algorithms only and we find that CIS performs the best in 11 out of 17 datasets.

In Table 5, we also show the p-values of the Wilcoxon signed rank test on pairwise difference between the accuracy values obtained by KNN when the instance selection algorithms are used. As we see from the tbale, with 95% confidence level, CIS50 is not significantly different from any of the three algorithms but EIS is significantly different from GDIS and EGDIS.

Table 5: Left: Mean accuracy values (%) obtained by KNN with $k = 3$ when using CIS50, GDIS, EGDIS, and EIS. Also, the instance reduction rates (%) of GDIS, EGDIS, and EIS while CIS50 has a reduction rate of 50%. The emboldened value for each dataset is the best accuracy over the four instance selection algorithms while the underlined values for each dataset is the best accuracy over GDIS, EGDIS, and CIS. Both for accuracy values and reduction rates, the higher the better. Results of GDIS, EGDIS, and EIS are collected from the publication of EIS. Right: p-values for Wilcoxon signed rank test on the pairwise difference of the accuracy values.

Dataset		Mean Accuracy				Reduction Rate		
ID	Name	CIS50	GDIS	EGDIS	EIS	GDIS	EGDIS	EIS
2	appendicitis	86.7	76.6	86.0	**95.0**	63.4	97.1	97.1
4	bupa	**65.4**	55.6	55.5	64.1	35.8	95.4	89.8
5	coil2000	92.0	93.7	93.7	**96.5**	80.2	84.6	85.8
7	contraceptive	**55.9**	49.0	47.0	54.5	86.3	88.1	77.0
8	haberman	72.3	70.3	69.7	**75.1**	63.2	67.9	95.6
9	hayes-roth	63.8	**66.4**	64.5	66.2	58.9	79.4	80.6
10	heart	83.0	84.1	85.3	**93.0**	75.8	80.4	95.6
12	led7digit	64.4	68.9	76.5	**82.3**	62.9	58.8	96.2
13	marketing	**40.1**	26.5	30.2	33.8	16.2	96.4	61.1
14	monk-2	93.5	92.9	94.1	**95.7**	76.6	61.4	96.6
16	pima	72.9	68.1	67.6	**73.2**	69.1	78.2	95.1
17	satimage	89.4	88.6	87.4	**91.3**	82.9	87.1	98.2
18	segment	93.4	92.8	90.7	**97.9**	80.2	83.7	97.8
19	titanic	**73.6**	63.6	65.9	72.1	53.4	99.8	99.8
20	vowel	**76.1**	66.1	65.6	67.3	78.6	30.1	94.3
21	wine	**98.3**	93.3	97.3	97.5	89.6	83.9	98.3
24	yeast	50.7	**57.1**	50.1	55.9	84.8	84.3	78.5
Mean		74.79	71.96	72.18	**77.14**	68.11	79.98	90.43

Wilcoxon Signed Rank Rest

p	GDIS	EGDIS	EIS
CIS50	0.05	0.055	0.1031
GDIS		0.567	0.0005
EGDIS			0.0003

5.5. Effect of CIS on KNN with Varying K

In Tables 2 and 4, we have shown the maximum mean accuracy values obtained by KNN over $k \in \{1, 3, 5\}$ while using NoSel, CIS, and RIS algorithms. In doing that, we have just followed the procedure used in RIS [8]. However, we further investigate how the accuracy values vary if specific k values are used with KNN. Table 6 (top) shows the mean accuracy values over 24 datasets when using various k values and various instance selection algorithms. When CIS50, CIS60, and CIS70 algorithms are used, $k = 1$ leads to the best KNN accuracy values while $k = 5$ is best when NoSel algorithm is used. This is interesting and the reason is with instances reduced, the classification decisions somewhat tend to rely on fewer neighbours. Table 6 (bottom) shows that the accuracy differences for pairs of k values with NoSel and CIS versions are statistically significant except in one case where k values are 1 and 3 and NoSel algorithm is used. Taking the best k options for NoSel, CIS50, CIS60, and CIS70 versions, we have four best versions and we perform another set of pairwise Wilcoxon signed rank tests with confidence level 95%. The statistical difference between all pairs are significant with p-values at most 0.037.

Table 6: Accuracy values (top) obtained by KNN with $k \in \{1, 3, 5\}$ while using NoSel and CIS algorithms, and p-values of Wilcoxon signed rank test (bottom) for pairwise comparisons. Emboldened accuracy values are the best ones while the underlined p-value is where the difference is not statistically significant at 95% confidence level.

	NoSel			CIS50			CIS60			CIS70		
Mean	1	3	5	1	3	5	1	3	5	1	3	5
Accuracy	67.93	68.70	**69.48**	**76.24**	73.78	72.30	**77.30**	74.63	73.25	**78.71**	75.84	74.41

Wilcoxon												
Signed	p	3	5	p	3	5	p	3	5	p	3	5
Rank Test	1	0.066	0.036	1	0.022	0.014	1	0.027	0.013	1	0.034	0.014
	3		0.018	3		0.024	3		0.029	3		0.020

Figure 2 shows the accuracy values for the 24 datasets as obtained by KNN while using $k \in \{1, 3, 5\}$ and CIS50. For NoSel, CIS60, and CIS70, the graphs are similar.

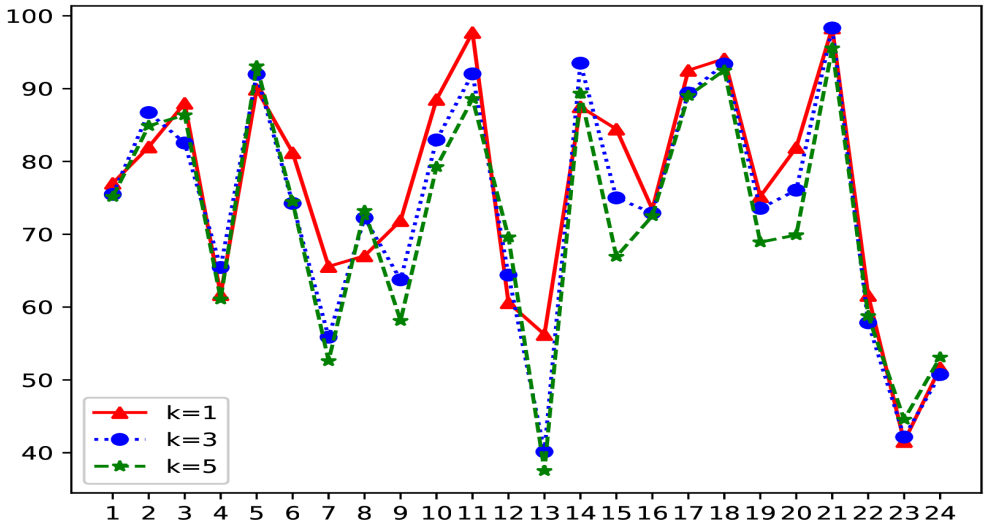


Figure 2: Accuracy values (y-axis) for 24 datasets (x-axis) as obtained by KNN while using $k \in \{1, 3, 5\}$ and CIS50.

5.6. Effect of Varying K in KMC on CIS

An important parameter of CIS is k, which is the number of clusters to be identified by KMC algorithm. So far for any given dataset, the value of k has been kept equal to the number classes $|C|$ in the dataset. We further investigate how various values of k affect the accuracy levels. For this, we use KNN with $k = 3$. Moreover, we use CIS50 but the value of k for KMC is chosen from $\{|C|, 2|C|, 3|C|, 4|C|\}$. Table 7 shows the accuracy values and also the p-values of Wilcoxon signed rank test at 95% confidence interval. As we see the performance of CIS50 improves in 21 out of 24 datasets, if the number of clusters is increased. However, the accuracy values due to $k = |C|$ are statistically different from that due to other larger k values. In particular, no statistically significant difference is observed among accuracy values due to $2|C|$, $3|C|$, and $4|C|$.

The cluster oriented instance selection of CIS tacitly assumes that the clusters are hyper-spherical. When classes are irregular or scattered, using the class number as the cluster number might not be the best option. In such a case, a large number of clusters might help capture the distributed or segmented density of the instances and thus better represent a given dataset. As

Table 7: Accuracy values (left) obtained by KNN with $k = 3$ and CIS50 where in KMC, $k \in \{|C|, 2|C|, 3|C|, 4|C|\}$ and p-values (right) of pairwise Wilcoxon signed rank tests. Emboldened accuracy values are the best for each dataset while underlined p-values are where differences are not statistically significant at 95% confidence level.

ID	Dataset	$\|C\|$	$2\|C\|$	$3\|C\|$	$4\|C\|$
1	adult	75.48	79.19	**79.46**	79.42
2	appendicitis	86.72	89.73	**91.64**	**91.64**
3	balance	82.54	83.82	**85.60**	**85.60**
4	bupa	65.44	66.02	**66.61**	66.60
5	coil2000	91.97	92.14	92.19	**92.32**
6	connect-4	74.24	74.05	74.47	**74.58**
7	contraceptive	55.87	**57.15**	57.09	56.61
8	haberman	72.25	**74.85**	72.24	71.28
9	hayes-roth	63.75	**69.38**	66.88	61.25
10	heart	82.96	85.55	85.93	**86.66**
11	ionosphere	92.03	**92.30**	91.73	89.47
12	led7digit	64.40	64.40	**67.00**	64.20
13	marketing	40.14	41.41	41.47	**41.83**
14	monk-2	93.50	94.89	95.13	**96.06**
15	movement-libras	**75.00**	74.72	74.72	74.17
16	pima	72.91	**73.95**	73.69	74.08
17	satimage	89.40	90.50	**90.69**	90.43
18	segment	93.38	93.59	94.24	**95.24**
19	titanic	**73.56**	71.92	71.42	68.83
20	vowel	76.06	77.07	79.09	**80.91**
21	wine	**98.33**	**98.33**	**98.33**	**98.33**
22	winequality-red	57.85	58.23	**58.48**	57.73
23	winequality-white	42.16	43.88	**44.81**	43.90
24	yeast	50.74	**52.63**	52.09	51.29
	Mean	73.78	74.99	**75.21**	74.68

Wilcoxon Signed Rank Test

p	$2\|C\|$	$3\|C\|$	$4\|C\|$
$\|C\|$	0.001	0.001	0.042
$2\|C\|$		<u>0.199</u>	<u>0.903</u>
$3\|C\|$			<u>0.181</u>

we see from the results, just using twice the number of classes as the number of clusters might be sufficient. Similar types of experiments have been done in PSC [31].

5.7. Effect of CIS on Various Classifiers

So far we have used KNN as the classifier. We further investigate how CIS affects the accuracy levels obtained by other classifiers. We choose SVM with linear kernel (LSVM) and Gaussian Naive Bayes (GNB). Moreover, we choose CIS50 (with $k = |C|$) as the instance selection algorithm. We follow the same procedure of 10 fold cross validation as used so far and compute the mean accuracy values. Table 8 shows the mean accuracy values obtained by KNN (with $k = 3$), LSVM, and GNB classifiers while using NoSel and CIS50. Thus, these results are directly comparable to those in the previous sections and also to the results in RIS [8]. Table 8 also shows the p-values of Wilcoxon signed rank test at 95% confidence level.

As we see from Table 8, CIS helps KNN, GNB, and LSVM improve accuracy levels in 17, 13, and 6 out of 24 datasets. Overall, with 50% instance reduction rate, CIS leads to better mean accuracy (5.08% improvement) for KNN, but slightly worse mean accuracy (1%–2% loss) for GNB

Table 8: Mean accuracy values (top) obtained by various classifiers using NoSel and CIS50; Also, p-values of Wilcoxon signed rank test (bottom). The emboldened values for each dataset for each classifier is the best accuracy value. The underlined p-value denotes the difference is not statistically significant at 95% confidence level.

		KNN		GNB		LSVM	
ID	Dataset	NoSel	CIS50	NoSel	CIS50	NoSel	CIS50
1	adult	75.48	**75.49**	78.88	**79.42**	79.45	**82.00**
2	appendicitis	83.82	**86.72**	**84.73**	79.18	**87.73**	84.00
3	balance	74.08	**82.54**	83.84	**88.31**	**91.69**	87.52
4	bupa	64.61	**65.44**	56.22	**57.12**	**67.80**	63.45
5	coil2000	**92.95**	91.97	17.22	**17.86**	**94.03**	93.97
6	connect-4	56.19	**74.24**	54.95	**61.14**	**65.83**	65.74
7	contraceptive	48.55	**55.87**	46.71	**48.00**	**51.60**	51.12
8	haberman	70.60	**72.25**	74.22	**74.85**	71.89	**74.17**
9	hayes-roth	**65.63**	63.75	**67.50**	66.25	56.25	**56.26**
10	heart	66.67	**82.96**	**84.07**	80.00	**83.70**	82.22
11	ionosphere	83.18	**92.03**	**88.33**	79.49	86.90	**91.46**
12	led7digit	**69.60**	64.40	**69.20**	60.60	75.00	**77.00**
13	marketing	28.01	**40.14**	30.70	**30.99**	**31.75**	31.31
14	monk-2	**96.55**	93.50	86.58	**91.89**	**80.55**	78.48
15	movement-libras	**78.06**	75.00	58.33	**64.17**	**70.00**	68.61
16	pima	69.01	**72.91**	**75.77**	69.14	**77.20**	75.64
17	satimage	89.00	**89.40**	**79.27**	75.81	**85.64**	80.72
18	segment	**95.06**	93.38	**79.91**	74.03	**96.15**	89.39
19	titanic	72.11	**73.56**	**77.33**	77.19	**77.60**	76.97
20	vowel	63.13	**76.06**	59.70	**60.10**	**59.90**	49.90
21	wine	71.44	**98.33**	96.63	**97.75**	96.63	**98.33**
22	winequality-red	44.15	**57.85**	**54.47**	48.09	**58.04**	55.41
23	winequality-white	38.38	**42.16**	**44.38**	32.23	**52.90**	47.45
24	yeast	**52.43**	50.74	14.01	**25.19**	**55.45**	54.98
Mean		68.70	**73.78**	**65.12**	64.17	**73.07**	71.50
Wilcoxon signed rank test p-value		0.0082		0.2389		0.0139	

and LSVM. However, for GNB, the accuracy drop is not statistically significant. These results show the strength of CIS as an instance selection algorithm over various classifiers.

5.8. Time Complexity Comparison

Algorithms CIS, GDIS, EGDIS, and EIS have been implemented on different programming language and on different machine learning platforms. Moreover, we compare our results with the results reported by the other methods. It is not possible for us to run all the programs on a single machine to compare the actual running times head to head. In this paper, we rather compare the theoretical time complexities of the mentioned instance selection algorithms.

Theorem 1 in Section 4.2 shows that CIS has a time complexity of $\mathcal{O}(klmn)$, k is the number of clusters to be found by K-Means clustering algorithm, l is the number of iterations K-Means clustering algorithm will run for, m is the number of attributes in each instances, and n is the number of instances. After analysing the algorithms of RIS, GDIS, and EGDIS, we found their

time complexities to be $\mathcal{O}(n^2m)$, $\mathcal{O}(n^2mk)$, $\mathcal{O}(n^2mk)$ respectively. For EIS, the time complexity is found to be $\mathcal{O}(vpn^2mk)$, where $v \times p$ is the size of the grid search space in EIS.

5.9. Clustering Quality after Instance Selection

To observe the quality of clustering of the dataset when NoSel, CIS50, CIS60, and CIS70, we compute Davies-Bouldin index (DBI) for each dataset and show the DBI value in Figure 3. The DBI value measures some kind of mean dispersion of the datapoints inside each class. As we see from the figure, before instance selection and after selection, the DBI values do not appear be much different. We perform Wilcoxon signed rank test with DBI values for CIS50, CIS60, and CIS70 against NoSel and the p-values are 0.27, 0.22, and 0.06 respectively. With 95% confidence level, these p-values clearly do not indicate significant differences.

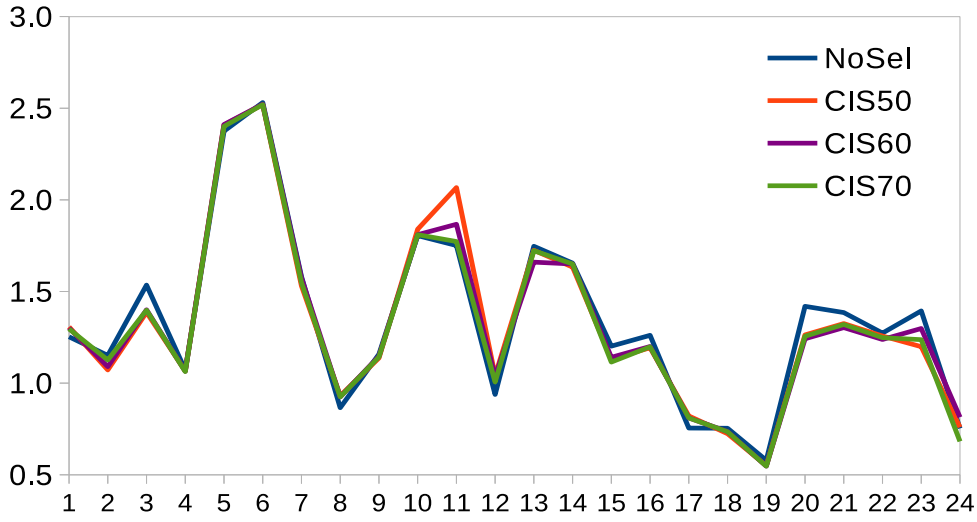


Figure 3: Davies–Bouldin indexes (y-axis) for datasets (x-axis) when using NoSel, CIS50, CIS60, and CIS70.

6. Overall Discussion

As noted before, our results are directly comparable with those in RIS [8] on all of the 24 datasets. Also, our results are directly comparable with GDIS [29], EGDIS [29], and EIS [22] on 17 datasets. Combining all results, Figure 4 shows the mean accuracy values against mean reduction rates for various classifiers using various instance selection algorithms. In the figure, we have included results of ENN [47], DROP3 [46], and ATISA1 [7]. Further, LSVM-NoSel, KNN-NoSel, GNB-NoSel, LSVM-CIS50, KNN-CIS50, and GNB-CIS50 results in the chart are from Table 8. Furthermore, CIS50, CIS60, CIS70, RIS1, RIS2, RIS3 results in the chart are from Table 2 and CISR results are from Table 4. Note that GDIS, EGDIS, and EIS results are on 17 datasets and are from Table 5 while for the results of the other algorithms are on 24 datasets. With very high reduction rates, accuracy values are very low. Overall, the chart shows CIS's strength in achieving high accuracy values with about 50% instance reduction rates.

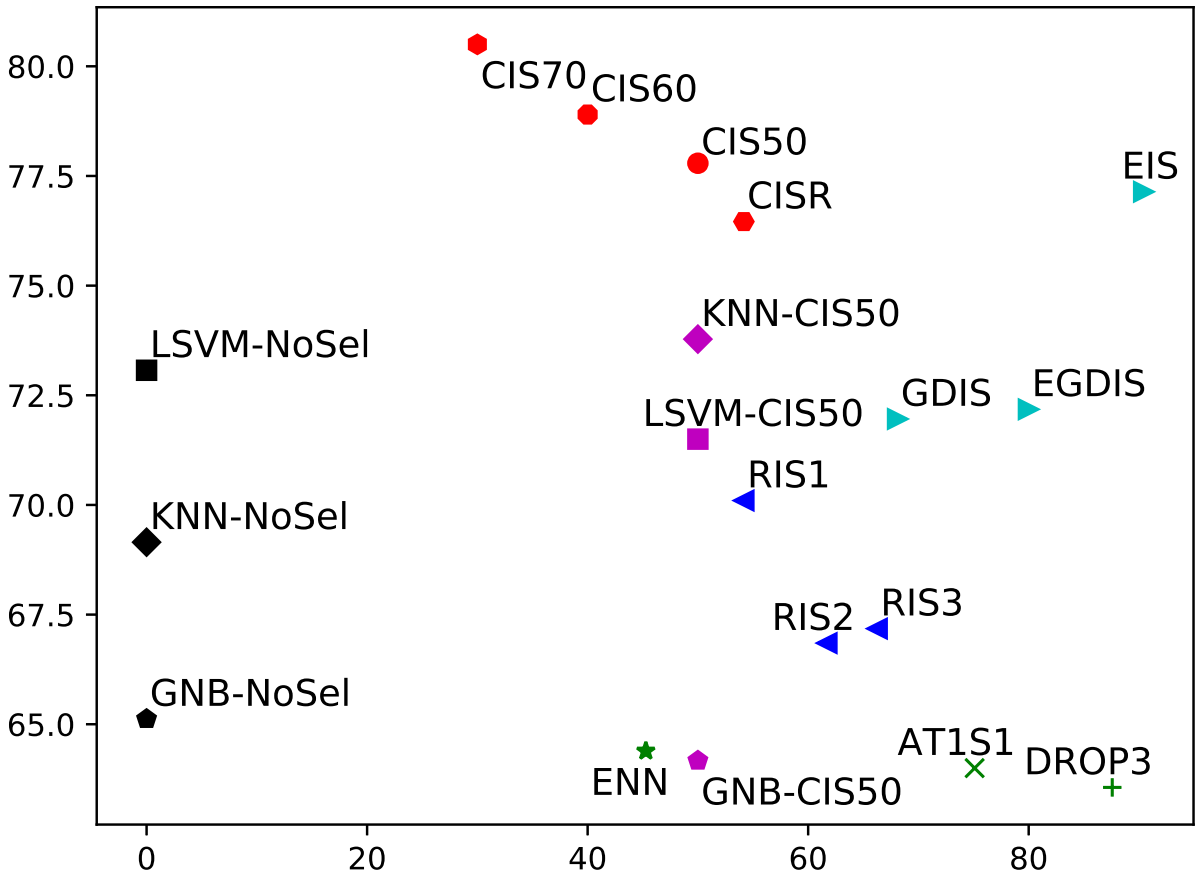


Figure 4: Mean (%) accuracy values (y-axis) vs mean (%) instance reduction rates (x-axis) for various classifiers using various instance selection algorithms. In both axes, the larger the better and there is a trade-off scenario.

7. Conclusions

Instance selection algorithms help reduce training data, preferably increasing and at least not causing significant decrease of performance of machine learning algorithms. Reduced training data could contain fewer outliers and noises and might need less computational resources. However, many instance selection algorithms struggle to maintain high accuracy values while significantly reducing instances. Moreover, existing instance selection algorithms do not allow direct controlling of the instance selection rate. We propose a simple and generic cluster-oriented instance selection (CIS) algorithm, which selects instances from cluster centers and borders. On 24 benchmark classification problems, when very similar percentages of instances are selected by various instance selection algorithms, CIS helps K Nearest Neighbour classifiers to achieve more than $2\%-3\%$ better accuracy than what other state-of-the-art generic instance selection algorithms do. Moreover, CIS could also select only 50% training instances losing just 1%—2% accuracy by Gaussian Naive Bayes(GNB) and Linear Support Vector Machine (LSVM) classifiers. CIS assumes hyper-spherical clusters and uses Euclidean distances in instance selection. Following this work, consideration of other distance measures and addressing irregular clusters or scattered classes could be interesting

future directions.

References

[1] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[2] Jesús Alcalá-Fdez, Alberto Fernández, Julián Luengo, Joaquín Derrac, Salvador García, Luciano Sánchez, and Francisco Herrera. Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic & Soft Computing*, 17, 2011.

[3] Paolo Arena, Luca Patanè, and Angelo Giuseppe Spinosa. Data-based analysis of laplacian eigenmaps for manifold reduction in supervised liquid state classifiers. *Information Sciences*, 478:28–39, 2019.

[4] Mohammad Aslani and Stefan Seipel. Efficient and decision boundary aware instance selection for support vector machines. *Information Sciences*, 577:579–598, 2021.

[5] Henry Brighton and Chris Mellish. Identifying competence-critical instances for instance-based learners. In *Instance Selection and Construction for Data Mining*, pages 77–94. Springer, 2001.

[6] Doina Caragea, Adrian Silvescu, and Vasant Honavar. A framework for learning from distributed data using sufficient statistics and its application to learning decision trees. *International Journal of Hybrid Intelligent Systems*, 1(1-2):80–89, 2004.

[7] George DC Cavalcanti, Tsang Ing Ren, and Cesar Lima Pereira. Atisa: Adaptive threshold-based instance selection algorithm. *Expert systems with applications*, 40(17):6894–6900, 2013.

[8] George DC Cavalcanti and Rodolfo JO Soares. Ranking-based instance selection for pattern classification. *Expert Systems with Applications*, 150:113269, 2020.

[9] Chin-Liang Chang. Finding prototypes for nearest neighbor classifiers. *IEEE Transactions on Computers*, 100(11):1179–1184, 1974.

[10] Fan Cheng, Feixiang Chu, and Lei Zhang. A multi-objective evolutionary algorithm based on length reduction for large-scale instance selection. *Information Sciences*, 576:105–121, 2021.

[11] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[12] Ireneusz Czarnowski. Cluster-based instance selection for machine classification. *Knowledge and Information Systems*, 30(1):113–133, 2012.

[13] Ireneusz Czarnowski and Piotr Jedrzejowicz. An approach to instance reduction in supervised learning. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence*, pages 267–280. Springer, 2003.

[14] Ireneusz Czarnowski and Piotr Jedrzejowicz. Data reduction algorithm for machine learning and data mining. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 276–285. Springer, 2008.

[15] Aida de Haro-García, Gonzalo Cerruela-García, and Nicolás García-Pedrajas. Instance selection based on boosting for instance-based learners. *Pattern Recognition*, 96:106959, 2019.

[16] Arthur P Dempster. Upper and lower probabilities induced by a multivalued mapping. In *Classic works of the Dempster-Shafer theory of belief functions*, pages 57–72. Springer, 2008.

[17] Thierry Denœux, Didier Dubois, and Henri Prade. Representations of uncertainty in ai: beyond probability and possibility. In *A Guided Tour of Artificial Intelligence Research*, pages 119–150. Springer, 2020.

[18] Pierre A Devijver and Josef Kittler. *Pattern recognition: A statistical approach*. Prentice hall, 1982.

[19] Salvador Garcı, Isaac Triguero, Cristobal J Carmona, Francisco Herrera, et al. Evolutionary-based selection of generalized instances for imbalanced classification. *Knowledge-Based Systems*, 25(1):3–12, 2012.

[20] César García-Osorio, Aida de Haro-García, and Nicolás García-Pedrajas. Democratic instance selection: a linear complexity instance selection algorithm based on classifier ensemble concepts. *Artificial Intelligence*, 174(5-6):410–441, 2010.

[21] Geoffrey Gates. The reduced nearest neighbor rule (corresp.). *IEEE transactions on information theory*, 18(3):431–433, 1972.

[22] Chaoyu Gong, Zhi-gang Su, Pei-hong Wang, Qian Wang, and Yang You. Evidential instance selection for k-nearest neighbor classification of big data. *International Journal of Approximate Reasoning*, 138:123–144, 2021.

[23] Joseph Lawson Hodges. *Discriminatory analysis*. 11. USAF School of Aviation Medicine, 1950.

[24] Yuan Jiang and Zhi-Hua Zhou. Editing training data for knn classifiers with neural network ensemble. In *International symposium on neural networks*, pages 356–361. Springer, 2004.

[25] Mirosław Kordos, Marcin Blachnik, and Rafał Scherer. Fuzzy clustering decomposition of genetic algorithm-based instance selection for regression problems. *Information Sciences*, 587:23–40, 2022.

[26] Hoang Lam Le, Ferrante Neri, and Isaac Triguero. Spms-als: A single-point memetic structure with accelerated local search for instance reduction. *Swarm and Evolutionary Computation*, page 100991, 2021.

[27] Junnan Li, Qingsheng Zhu, and Quanwang Wu. A parameter-free hybrid instance selection algorithm based on local sets with natural neighbors. *Applied Intelligence*, 50(5):1527–1541, 2020.

[28] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[29] Mohamed Malhat, Mohamed El Menshawy, Hamdy Mousa, and Ashraf El Sisi. A new approach for instance selection: Algorithms, evaluation, and comparisons. *Expert Systems with Applications*, 149:113297, 2020.

[30] Elena Marchiori. Class conditional nearest neighbor for large margin instance selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2):364–370, 2009.

[31] J Arturo Olvera-López, J Ariel Carrasco-Ochoa, and J Francisco Martínez-Trinidad. A new fast prototype selection method based on clustering. *Pattern Analysis and Applications*, 13(2):131–141, 2010.

[32] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[33] Gunnar Rätsch, Takashi Onoda, and K-R Müller. Soft margins for adaboost. *Machine learning*, 42(3):287–320, 2001.

[34] Thomas Reinartz. A unifying view on instance selection. *Data Mining and Knowledge Discovery*, 6(2):191–210, 2002.

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

[36] Eleanor Rosch and Carolyn B Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive psychology*, 7(4):573–605, 1975.

[37] Marcio Rubbo and Leandro A Silva. Filtering-based instance selection method for overlapping problem in imbalanced datasets. *J*, 4(3):308–327, 2021.

[38] José Salvador Sánchez, Ricardo Barandela, Ana I Marqués, Roberto Alejo, and Jorge Badenas. Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24(7):1015–1022, 2003.

[39] Glenn Shafer. *A mathematical theory of evidence*. Princeton university press, 1976.

[40] Anwar Shah, Nouman Azam, Bahar Ali, Muhammad Taimoor Khan, and JingTao Yao. A three-way clustering approach for novelty detection. *Information Sciences*, 569:650–668, 2021.

[41] Anantaporn Srisawat, Tanasanee Phienthrakul, and Boonserm Kijsirikul. Sv-knnc: An algorithm for improving the efficiency of k-nearest neighbor. In *Pacific rim international conference on artificial intelligence*, pages 975–979. Springer, 2006.

[42] Chih-Fong Tsai, William Eberle, and Chi-Yuan Chu. Genetic algorithms in feature and instance selection. *Knowledge-Based Systems*, 39:240–247, 2013.

[43] Chih-Fong Tsai, Wei-Chao Lin, Ya-Han Hu, and Guan-Ting Yao. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences*, 477:47–54, 2019.

[44] Claudio Turchetti and Laura Falaschetti. A manifold learning approach to dimensionality reduction for modeling data. *Information Sciences*, 491:16–29, 2019.

[45] D Randall Wilson and Tony R Martinez. Instance pruning techniques. In *ICML*, volume 97, pages 400–411, 1997.

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

[47] Dennis L Wilson. Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man, and Cybernetics*, 3:408–421, 1972.

[48] Zhijian Wu, Jun Li, Jianhua Xu, and Wankou Yang. Subspace-based self-weighted multiview fusion for instance retrieval. *Information Sciences*, 592:261–276, 2022.

[49] Jianping Zhang. Selecting typical instances in instance-based learning. In *Machine Learning Proceedings 1992*, pages 470–479. Elsevier, 1992.

[50] Fei Zhao, Yang Xin, Kai Zhang, and Xinxin Niu. Representativeness-based instance selection for intrusion detection. *Security and Communication Networks*, 2021, 2021.