Algorithms & Techniques for Efficient and Effective Nearest Neighbours Classification

PhD Thesis

Stefanos Ougiaroglou

Dept. of Applied Informatics
School of Information Sciences,
University of Macedonia,
Thessaloniki, Greece

Thesis publications (1/4)

Journal papers:

- Stefanos Ougiaroglou, Georgios Evangelidis, "**RHC: Non-parametric cluster-based data reduction for efficient *k*-NN classification**", Pattern Analysis and Applications, Springer, accepted with minor revision, revision is under review (I.F.: 0.814)
- Stefanos Ougiaroglou, Georgios Evangelidis, Dimitris A. Dervos "**FHC: An adaptive fast hybrid method for *k*-NN classification**", Logic Journal of the IGPL, Oxford journals, accepted with major revision, revision is under review (I.F.: 1.136)
- Stefanos Ougiaroglou, Georgios Evangelidis, "**Efficient data abstraction using weighted IB2 prototypes**", Computer Science and Information Systems (ComSIS), to appear (I.F.:0.549)
- Stefanos Ougiaroglou, Georgios Evangelidis, "**Efficient *k*-NN Classification based on Homogeneous Clusters**", Artificial Intelligence Review, Springer (I.F.: 1.565)
- Stefanos Ougiaroglou, Georgios Evangelidis, "**Efficient editing and data abstraction by finding homogeneous clusters**", under review

Thesis publications (2/4)

Book chapters:

- Stefanos Ougiaroglou, Leonidas Karamitopoulos, Christos Tatoglou, Georgios Evangelidis, Dimitris A. Dervos, “**Applying prototype selection and abstraction algorithms for efficient time series classification**”, In “Artificial Neural Networks-Methods and Applications in Bio-/Neuroinformatics (Series in Bio-/Neuroinformatics)”, Springer, to appear

Thesis publications (3/4)

Conference papers (1/2):

- Stefanos Ougiaroglou, Georgios Evangelidis, "**EHC: Non-parametric Editing by finding Homogeneous Clusters**", FoIKS 2014, Springer/LNCS 8367, pp. 290-304, Bordeaux, France, 2014
- Stefanos Ougiaroglou, Georgios Evangelidis, "**AIB2: An Abstraction Data Reduction Technique based on IB2**", BCI 2013, ACM ICPS, pp. 13-16, Thessaloniki, Greece, 2013
- Stefanos Ougiaroglou, Leonidas Karamitopoulos, Christos Tatoglou, Georgios Evangelidis, Dimitris Dervos, "**Applying general-purpose Data Reduction Techniques for fast time series classification**", ICANN 2013, Springer/LNCS 8131, pp. 34-41, Sofia, Bulgaria, 2013
- Stefanos Ougiaroglou, Georgios Evangelidis, "**A Fast Hybrid k-NN Classifier based on Homogeneous Clusters**", AIAI 2012, IFIP AICT 381, Springer, pp. 327-336, Halkidiki, Greece, 2012
- Stefanos Ougiaroglou, Georgios Evangelidis, "**Efficient Dataset Size Reduction by finding Homogeneous Clusters**", BCI 2012, ACM ICPS, pp. 168-173, Novi Sad, Serbia, 2012
- Stefanos Ougiaroglou, Georgios Evangelidis, "**Fast and Accurate k-Nearest Neighbor Classification using Prototype Selection by Clustering**", PCI 2012, IEEE CPS, pp. 168-173, Piraeus, Greece, 2012

Thesis publications (3/4)

Conference papers (2/2):

- Stefanos Ougiaroglou, Georgios Evangelidis, Dimitris A. Dervos, "**An Adaptive Hybrid and Cluster-Based Model for speeding up the k-NN Classifier**", HAIS 2012, Springer/LNCS 7209, pp. 163-175, Salamanca, Spain, 2012
- Stefanos Ougiaroglou, Georgios Evangelidis, "**A Simple Noise-Tolerant Abstraction Algorithm for Fast k-NN Classification**", HAIS 2012, Springer/LNCS 7209, pp.210-221, Salamanca, Spain, 2012
- Stefanos Ougiaroglou, Georgios Evangelidis, Dimitris A. Dervos, "**A Fast Hybrid Classification Algorithm based on the Minimum Distance and the k-NN Classifiers**", SISAP 2011, ACM, pp. 97-104, Lipari island, Italy, 2011
- Stefanos Ougiaroglou, Georgios Evangelidis, Dimitris A. Dervos, "**An Extensive Experimental Study on the Cluster-based Reference Set Reduction for speeding-up the k-NN Classifier**", IC-ININFO 2011, pp. 12-15, Kos island, Greece, 2011
- Stefanos Ougiaroglou, Georgios Evangelidis, "**WebDR: A Web Workbench for Data Reduction**", under review

A. Background knowledge & Related work

k-NN Classification (1/2)

A classifier is a data mining algorithm that attempts to map data to a set of classes

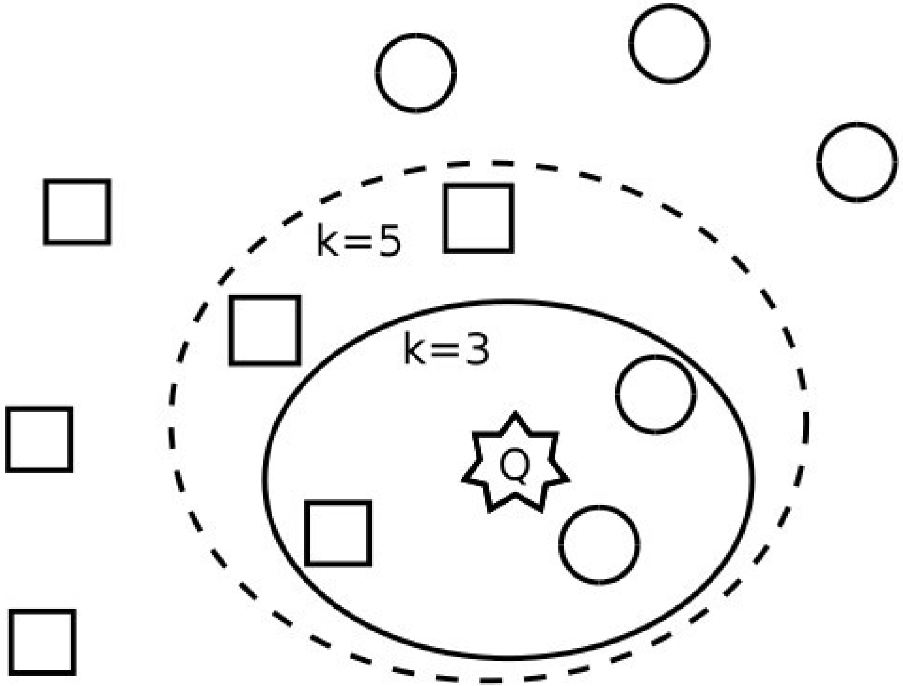
***k*-NN Classifier**:

- Extensively used and effective lazy classifier
- Easy to be implemented
- It has many applications
- It works by searching the database for the *k* nearest items to the unclassified item
- The *k* nearest items determine the class where the new item belongs to
- The “closeness” is defined by a distance metric

k-NN Classification (2/2)

k-NN example

- **k=3**, query point is assigned to class “circle”
- **k=5**, it is assigned to class “square”



Weaknesses / Thesis motivation

High computational cost: *k*-NN classifier needs to compute all distances between an unclassified item and the training data

e.g., 100,000 training items * 50,000 new items = 5 Billions distances

High storage requirements: The training database must be always available

Noise sensitive algorithm: Noise and mislabeled data, as well as outliers and overlaps between regions of classes affect classification accuracy

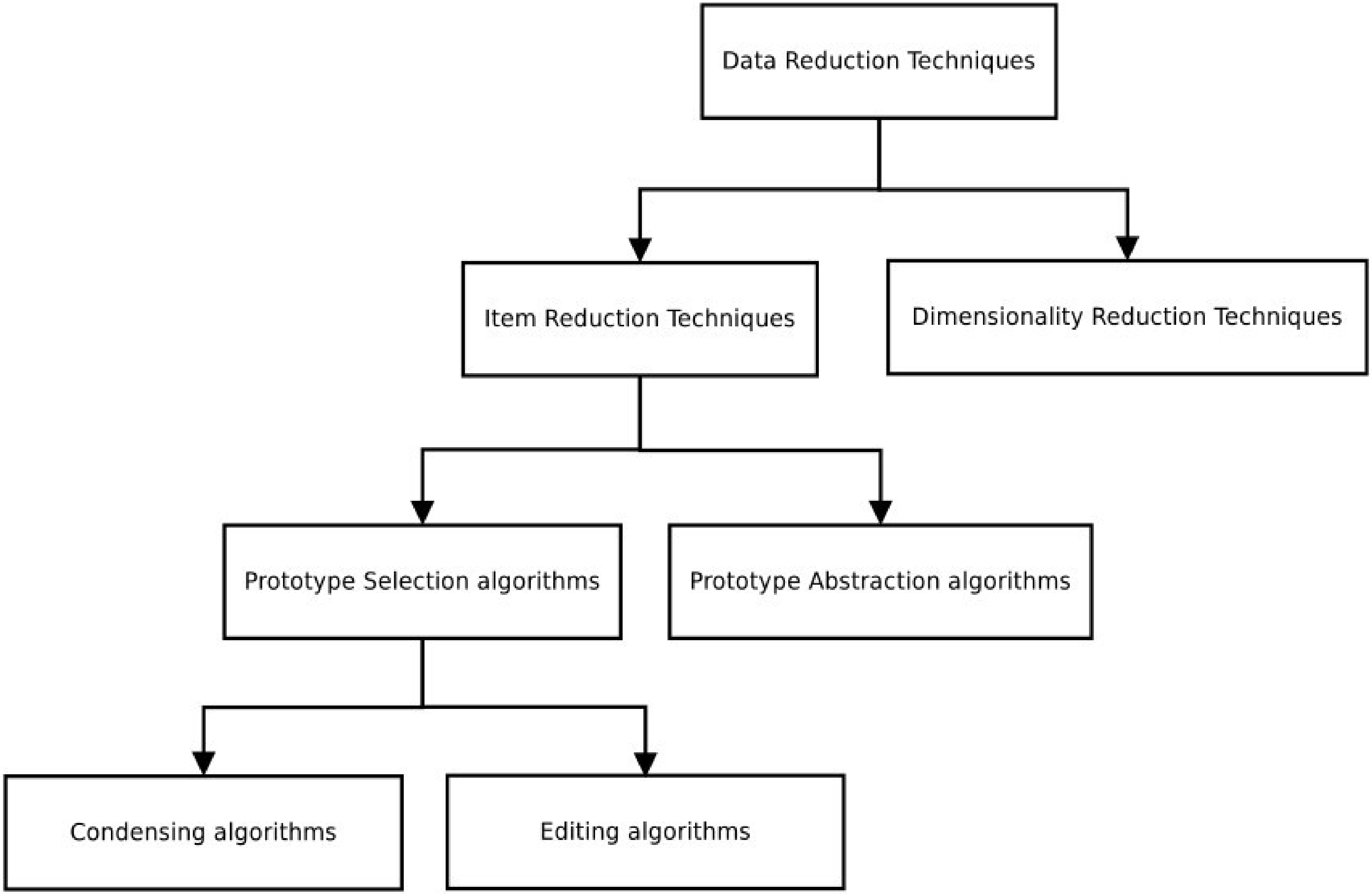
Method categories for efficient and effective *k*-NN classification

- Data Reduction Techniques (DRTs)

- Cluster-based methods (CBMs)

- Multi-attribute Indexing methods

Data Reduction Techniques (1/6)


Data Reduction Techniques
Item Reduction Techniques
Dimensionality Reduction Techniques
Prototype Selection algorithms
Prototype Abstraction algorithms
Condensing algorithms
Editing algorithms

Data Reduction Techniques (2/6)

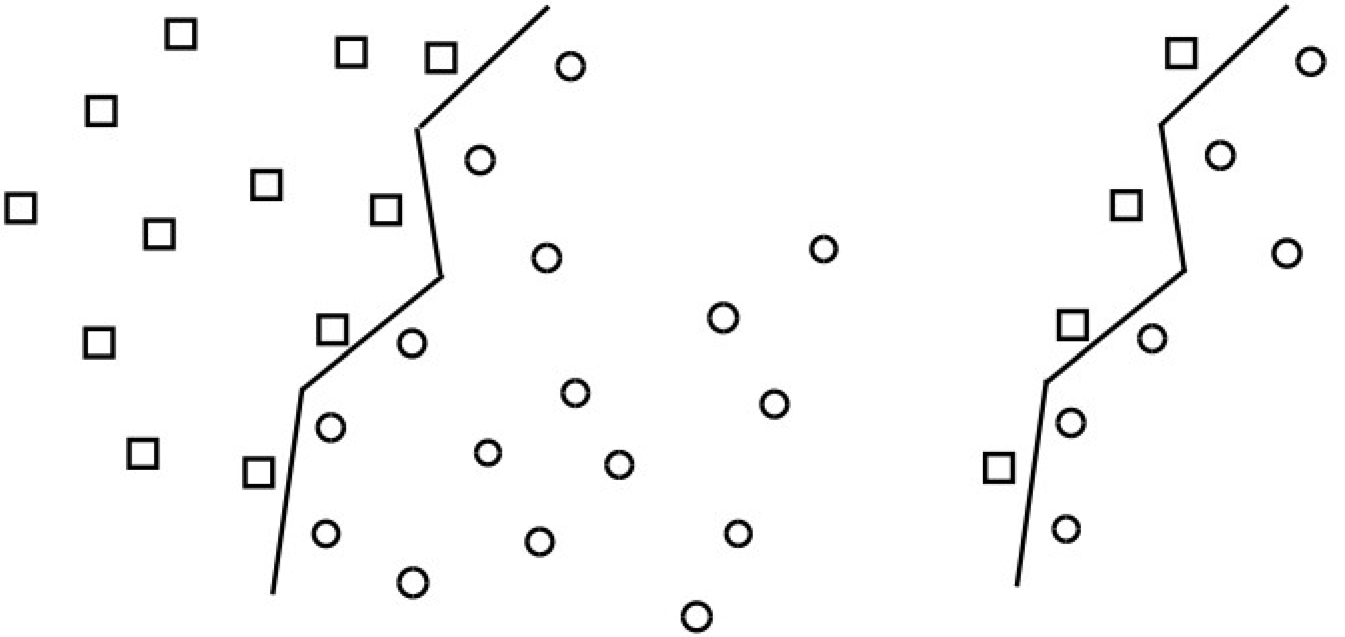
Condensing and Prototype Abstraction (PA) algorithms

- They deal with the drawbacks of high computational cost and high storage requirements by building a small representative set **(condensing set)** of the training data

- Condensing algorithms **select** and PA algorithms **generate** prototypes

- The idea is to apply *k*-NN on this set attempting to achieve as high accuracy as when using the initial training data at much lower cost and storage requirements

Data Reduction Techniques (3/6)



(a) Training set

(b) Condensing set

Data Reduction Techniques (4/6)

Editing algorithms

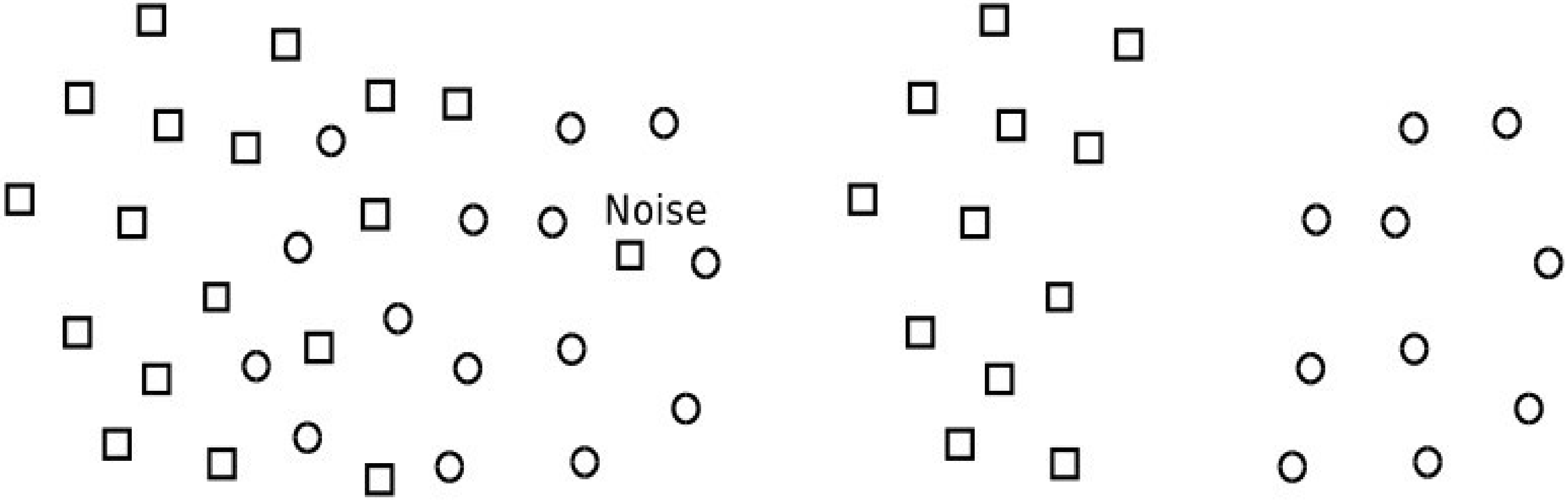
- They aim to improve accuracy rather than achieve high reduction rates

- They remove noisy and mislabeled items and smooth the decision boundaries. Ideally, they build an a set without overlaps between the classes

- The reduction rates of PA and condensing algorithms depend on the level of noise in the training data

- Editing has a double goal: accuracy improvement and effective application of PA and condensing algorithms

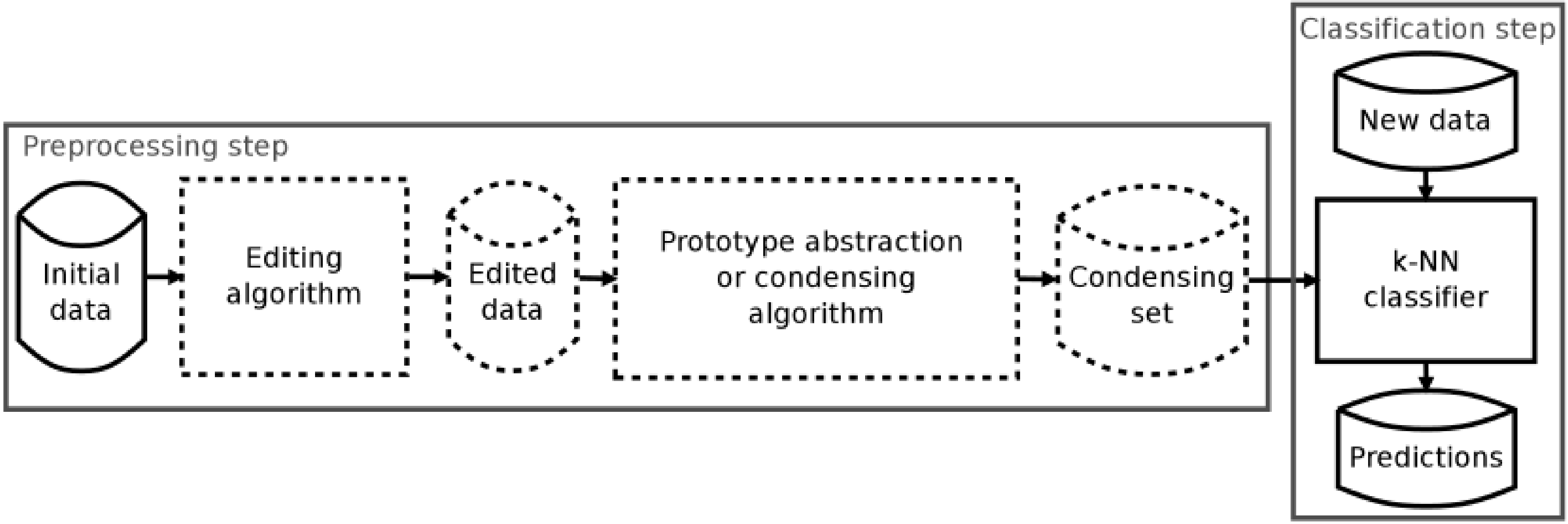
Data Reduction Techniques (5/6)



(a) Initial training set

(b) Edited set

Data Reduction Techniques (6/6)



Cluster-based Methods (CBMs)

CBMs idea:

- They pre-process the training data and placed them into clusters

- For each new item, they dynamically form a training Subset (reference set) of the initially data that is used to classify new data

- The training subset is the union of some clusters

- Contrary to DRTs, CBMs do not reduce the storage requirements

DRTs & CBMs implemented during the PhD

Prototype Selection algorithms

Condensing algorithms:

- Hart's Condensed Nearest Neighbour rule (CNN-rule)
- Instance Based learning 2 (IB2)
- Prototype Selection by Clustering (PSC)

Editing algorithms

- Edited Nearest Neighbour rule (ENN-rule)
- All-*k*-NN
- Multiedit

Prototype Abstraction algorithms

- Reduction by Space Partitioning 3 (RSP3)

Cluster-based methods

- Hwang and Cho method (HCM)

B. Contribution: Data Reduction Techniques

Reduction through Homogeneous Clusters (1/5)

Motivation/Weaknesses of Prototype Abstraction and condensing algorithms:

- They usually involve a costly, time-consuming preprocessing step on the training set
- Many algorithms are parametric
- The resulting condensing set may depends on the order of items in the training set
- Although some algorithms can achieve high RR, the accuracy of the classifier is affected
- Although some algorithms produce condensing sets that achieve accuracies close to those achieved by the non-reduced training sets, RR are not high

Reduction through Homogeneous Clusters (2/5)

Properties of RHC:

- It is an abstraction DRT
- Fast execution of the reduction procedure (low pre-processing cost)
- High reduction rates
- High classification accuracy
- Non-parametric algorithm
- It is based on the well-known *k*-Means clustering
- Its condensing set does not depend on the order of the training data

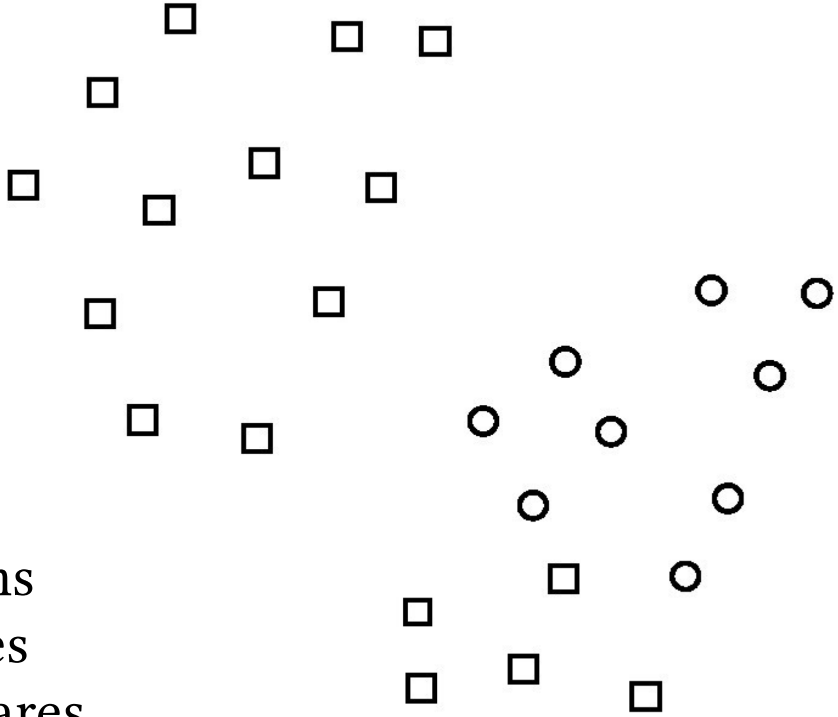
Reduction through Homogeneous Clusters (3/5)

RHC idea:

- RHC continues constructing clusters until all of them are homogeneous
- A cluster is homogeneous if all items that have been assigned to it are of a specific class
- The centroids of the homogeneous clusters constitute the condensing set

Reduction through Homogeneous Clusters (4/5)

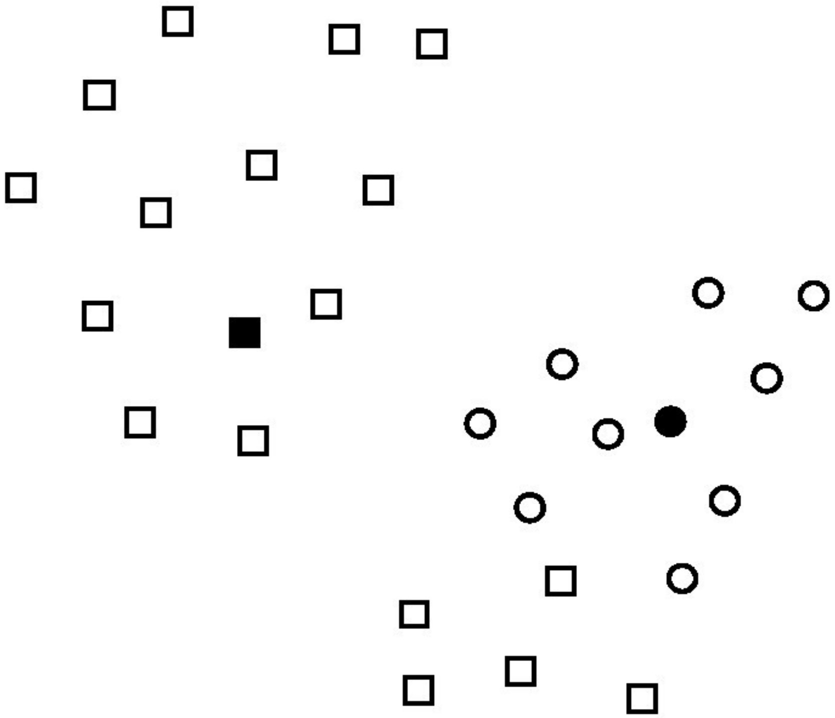
Initially, RHC considers the dataset as a non-homogeneous cluster



26 items
9 circles
17 squares

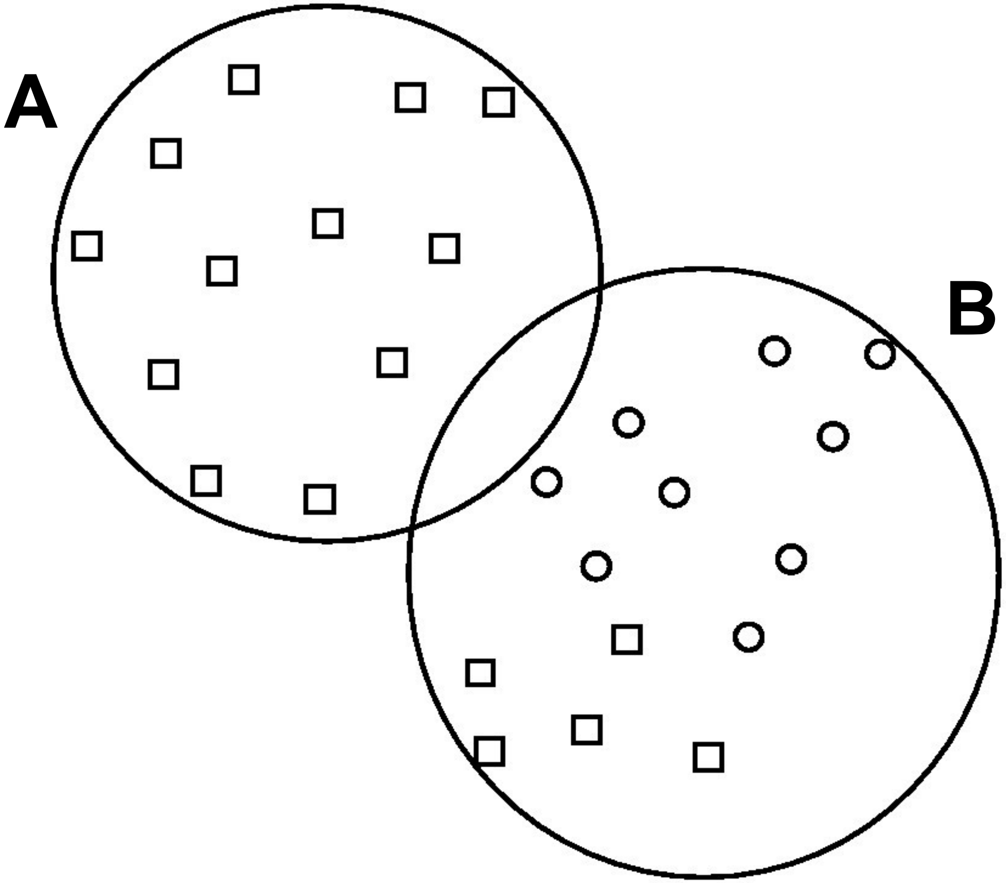
Reduction through Homogeneous Clusters (4/5)

RHC computes the mean item for each class in the data (class-mean)



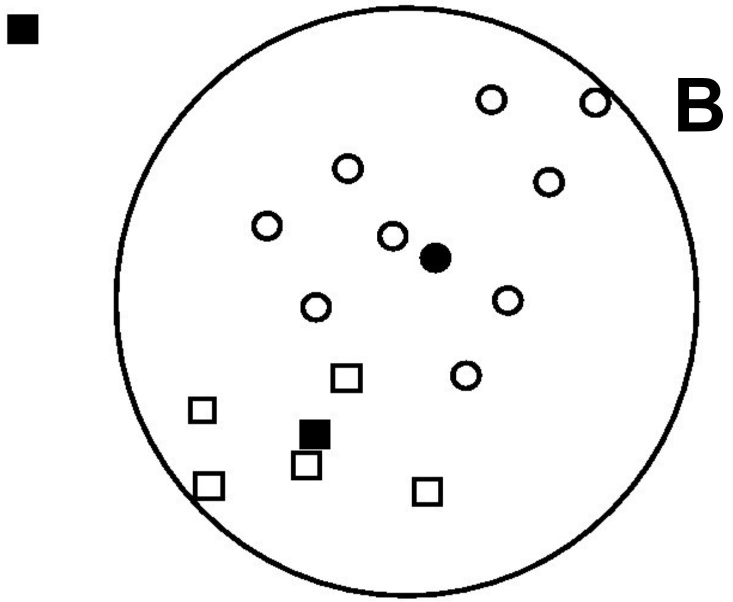
Reduction through Homogeneous Clusters (4/5)

RHC executes *k*-means clustering using the two class-means as initial means and builds two clusters



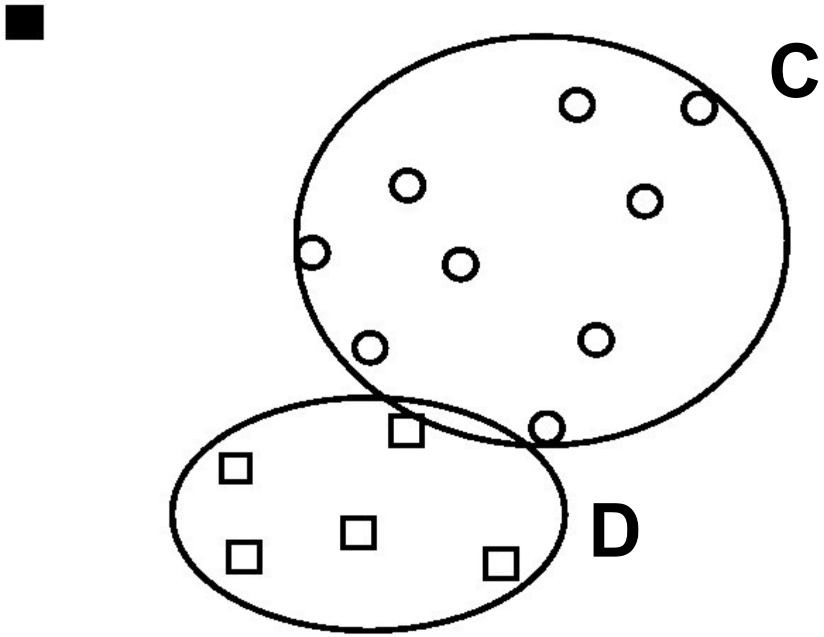
Reduction through Homogeneous Clusters (4/5)

RHC stores the cluster-mean of cluster A to the condensing set and computes a class-means for each class in B



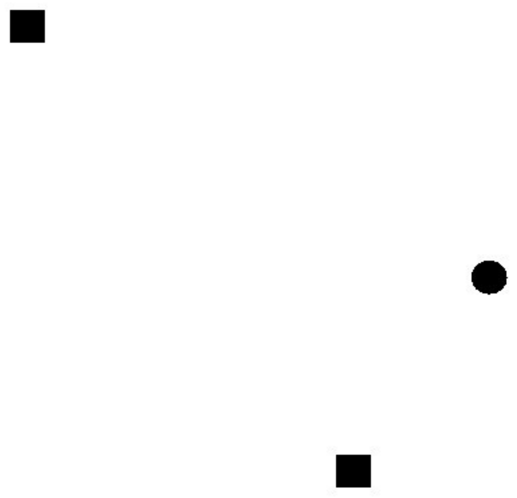
Reduction through Homogeneous Clusters (4/5)

k-means is executed on the data of B using as initial means the class-means and produces two clusters



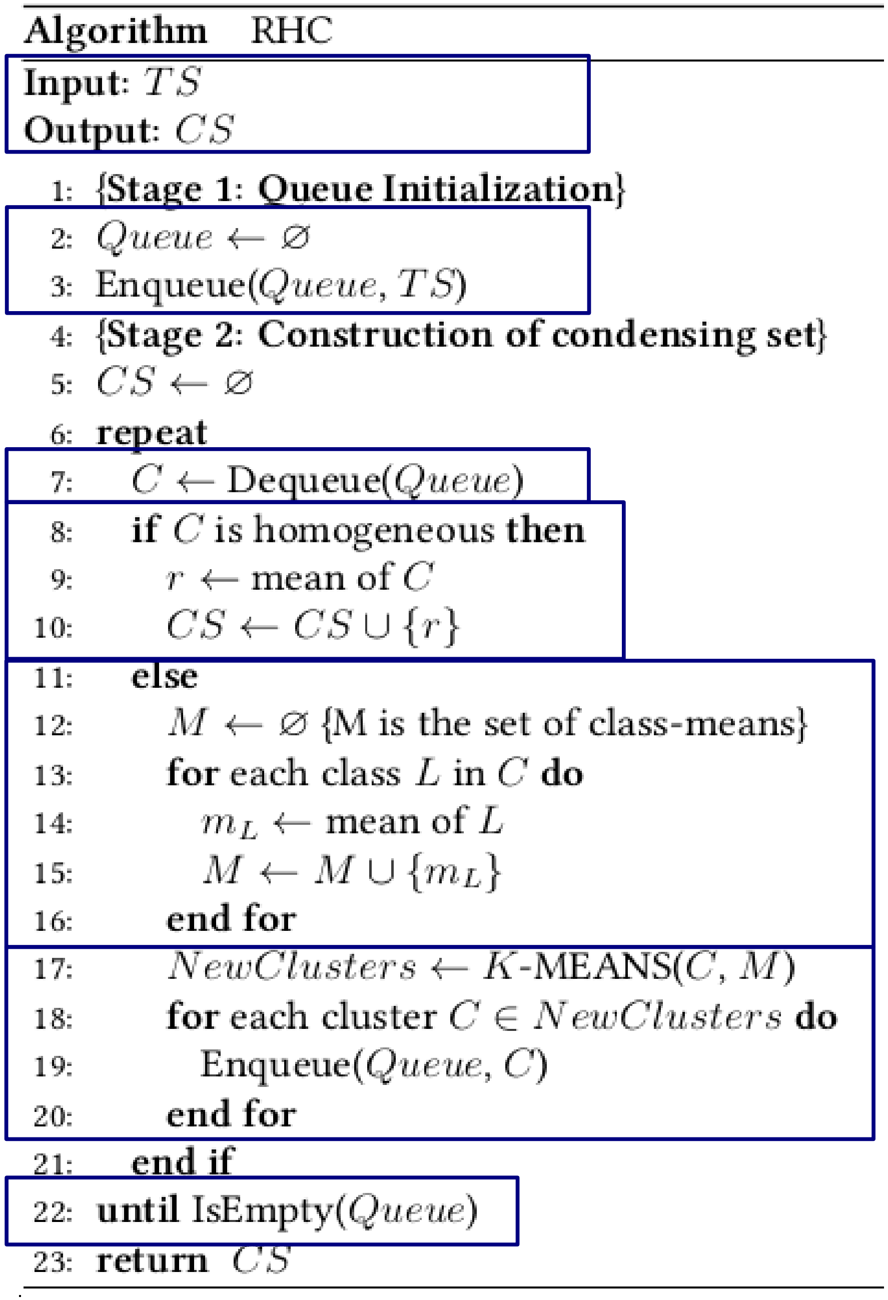
Reduction through Homogeneous Clusters (4/5)

C and D are homogeneous. RHC stores their means to the condensing set



RHC Condensing set

Reduction through Homogeneous Clusters (5/5)



```
Algorithm  RHC
Input: $TS$
Output: $CS$
{Stage 1: Queue Initialization}
$Queue \leftarrow \varnothing$
Enqueue($Queue$, $TS$)
{Stage 2: Construction of condensing set}
$CS \leftarrow \varnothing$
repeat
    $C \leftarrow$ Dequeue($Queue$)
    if $C$ is homogeneous then
        $r \leftarrow$ mean of $C$
        $CS \leftarrow CS \cup \{r\}$
    else
        $M \leftarrow \varnothing$ {M is the set of class-means}
        for each class $L$ in $C$ do
            $m_L \leftarrow$ mean of $L$
            $M \leftarrow M \cup \{m_L\}$
        end for
        $NewClusters \leftarrow K$-MEANS($C$, $M$)
        for each cluster $C \in NewClusters$ do
            Enqueue($Queue$, $C$)
        end for
    end if
until IsEmpty($Queue$)
return  $CS$
```

The dynamic RHC algorithm (1/4)

Motivation:

- Most DRTs are memory-based. This implies that the whole training set must reside in main memory. Thus, they are inappropriate for large datasets that cannot fit into main memory or for devices with limited main memory
- Most DRTs cannot consider new training items after the construction of the condensing set. They are inappropriate for dynamic/streaming environments where new training items are gradually available

The dynamic RHC algorithm (2/4)

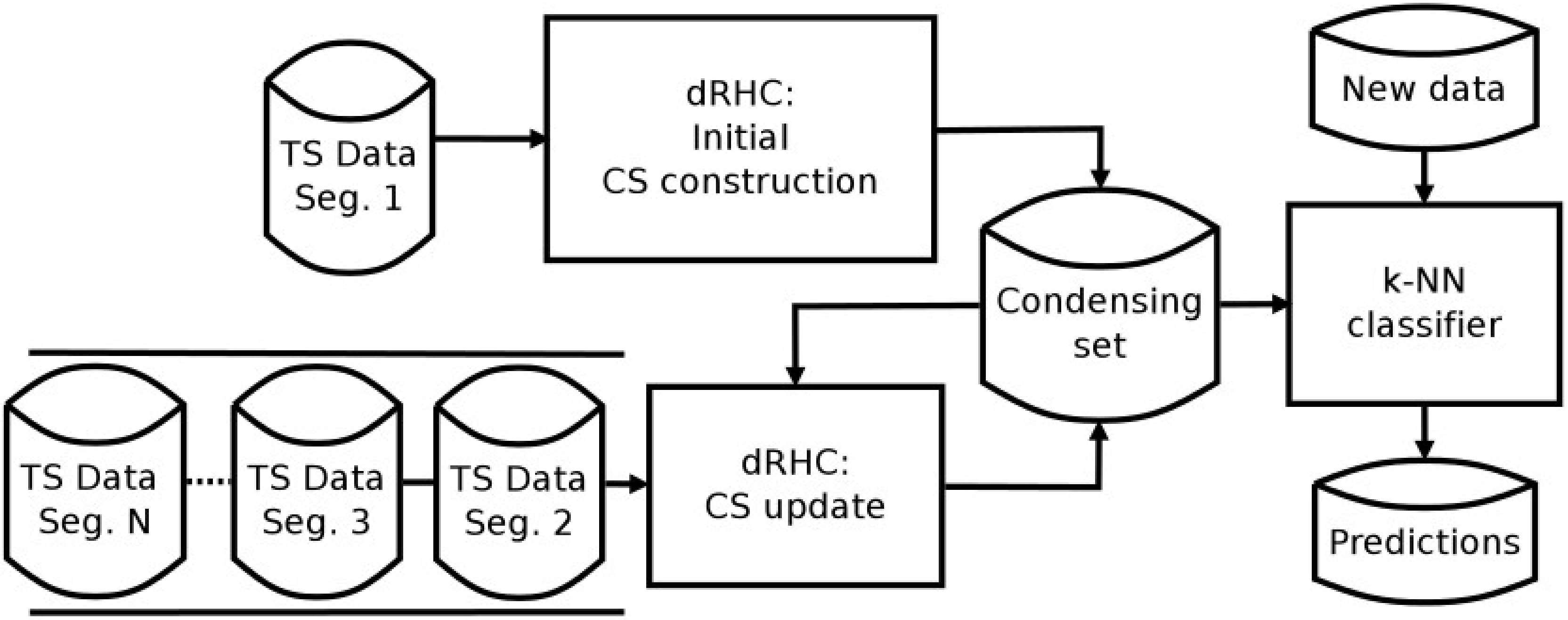
Properties of dynamic RHC (dRHC)

- dRHC is an incremental version of RHC which inherits all the good properties of RHC
- dRHC is a dynamic prototype abstraction algorithm that incrementally builds its condensing set.

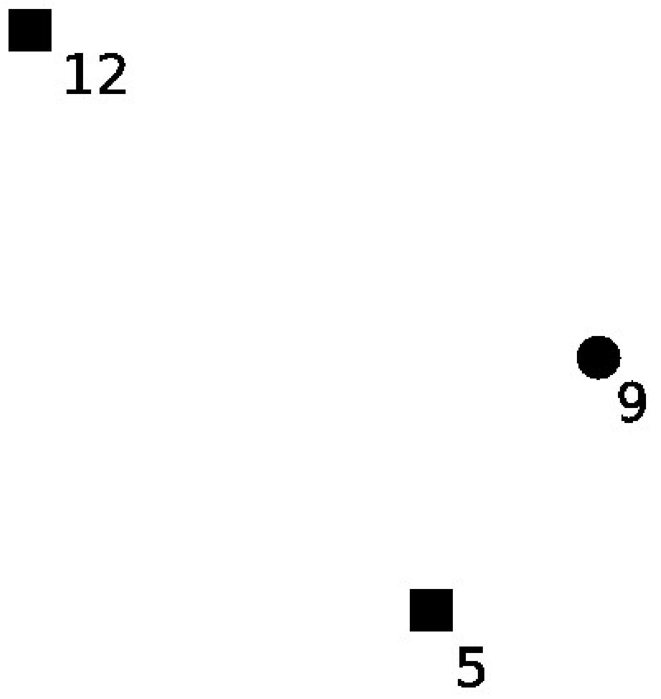
Therefore:

- dRHC is appropriate for dynamic/streaming environments where new training data is gradually available
- dRHC is appropriate for very large datasets that can not fit in main memory

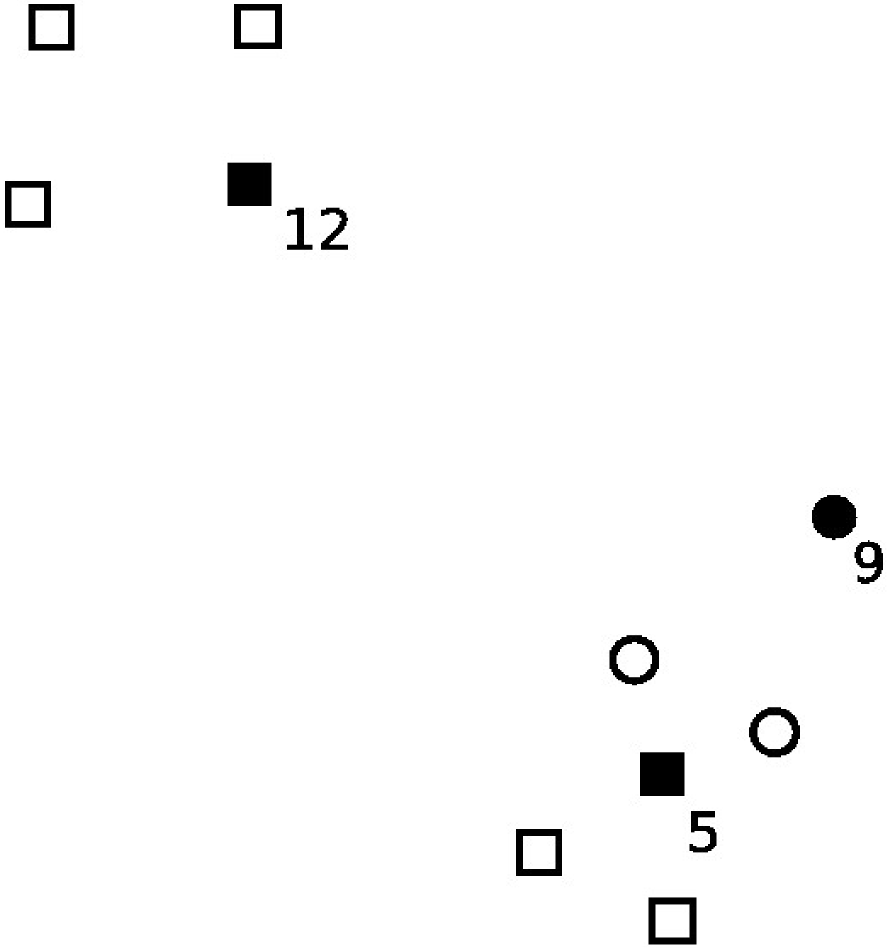
The dynamic RHC algorithm (3/4)



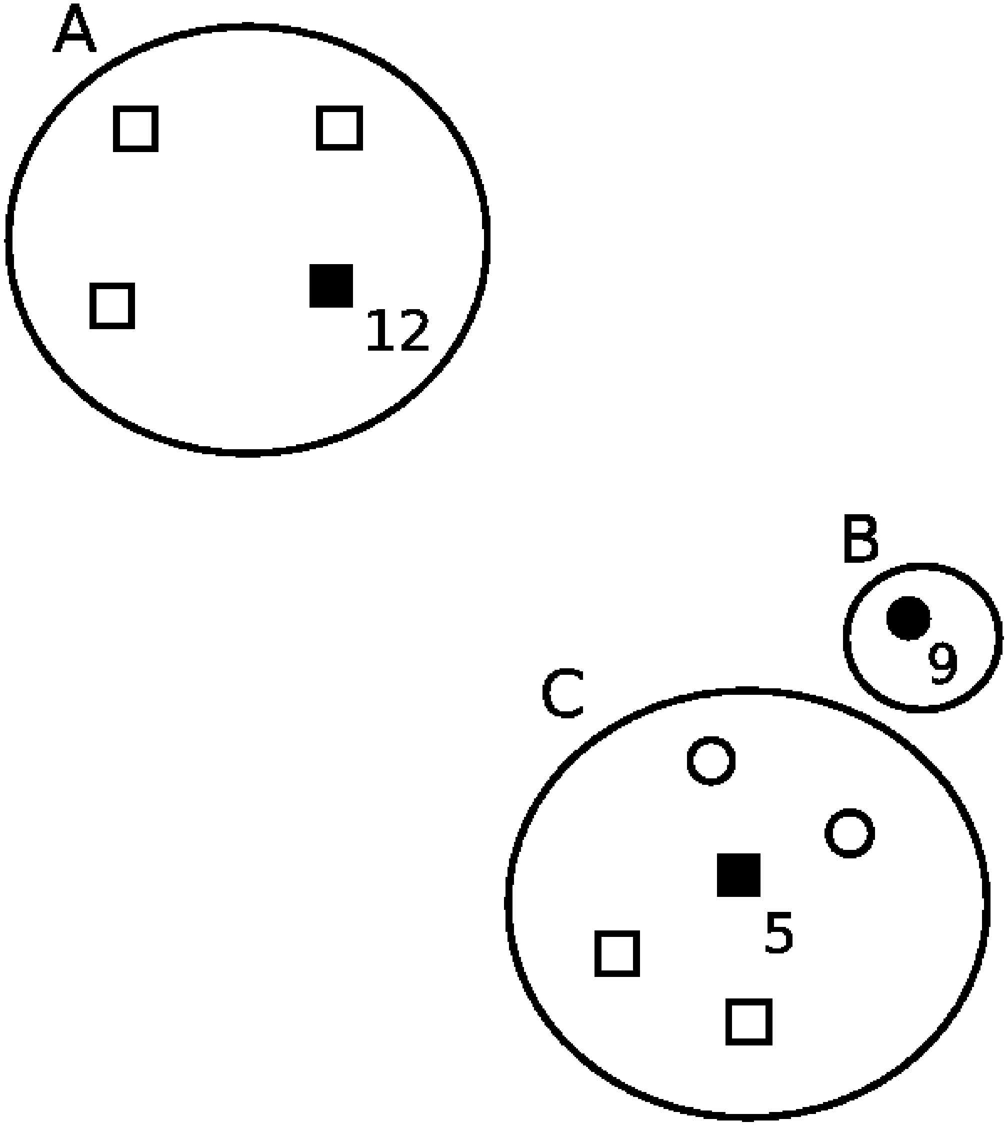
The dynamic RHC algorithm (4/4)



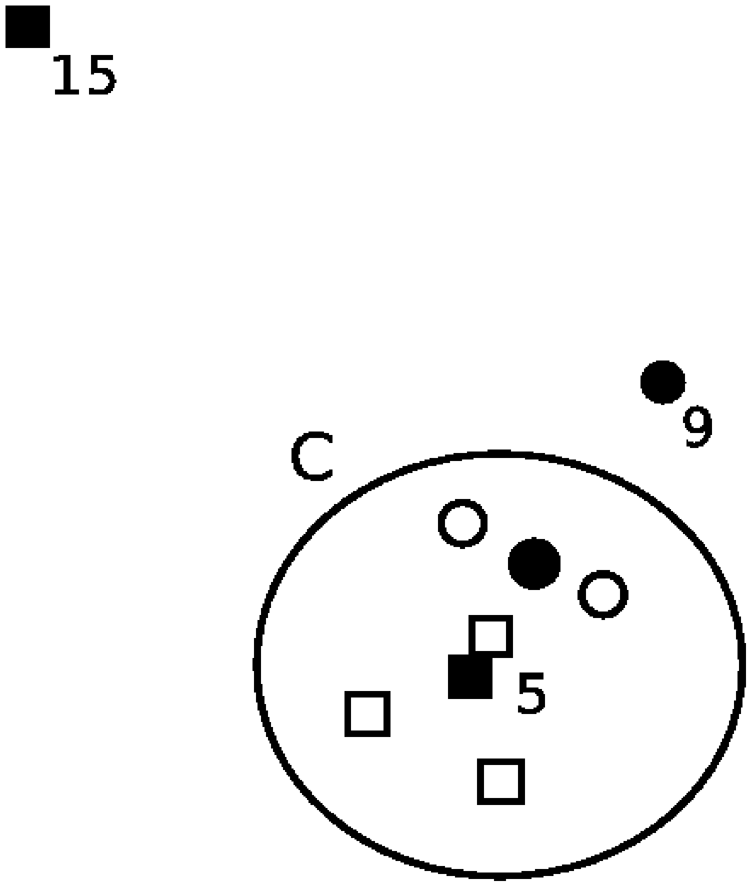
The dynamic RHC algorithm (4/4)



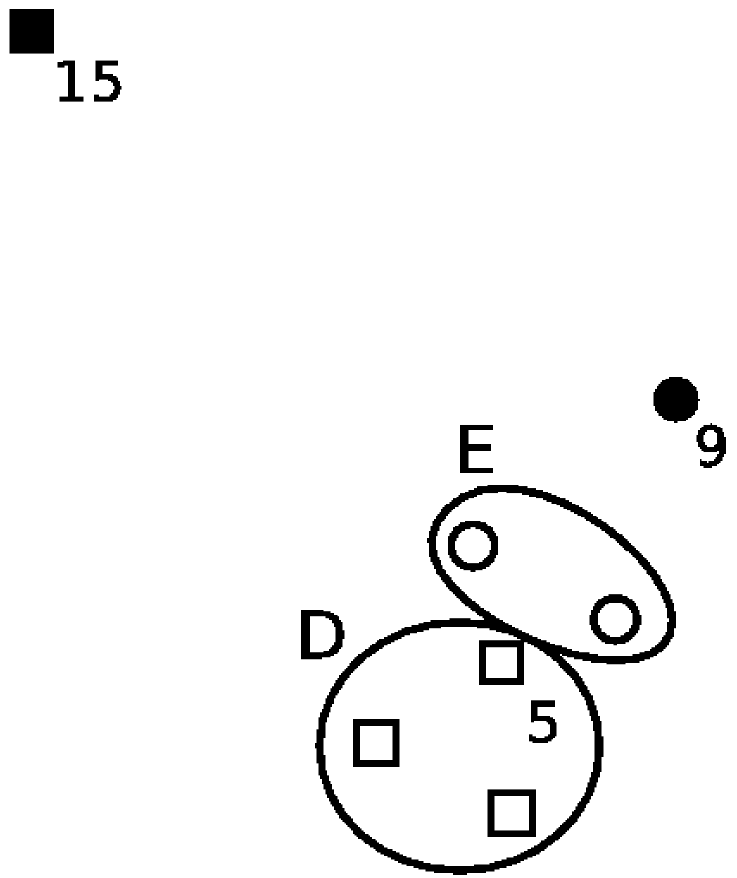
The dynamic RHC algorithm (4/4)



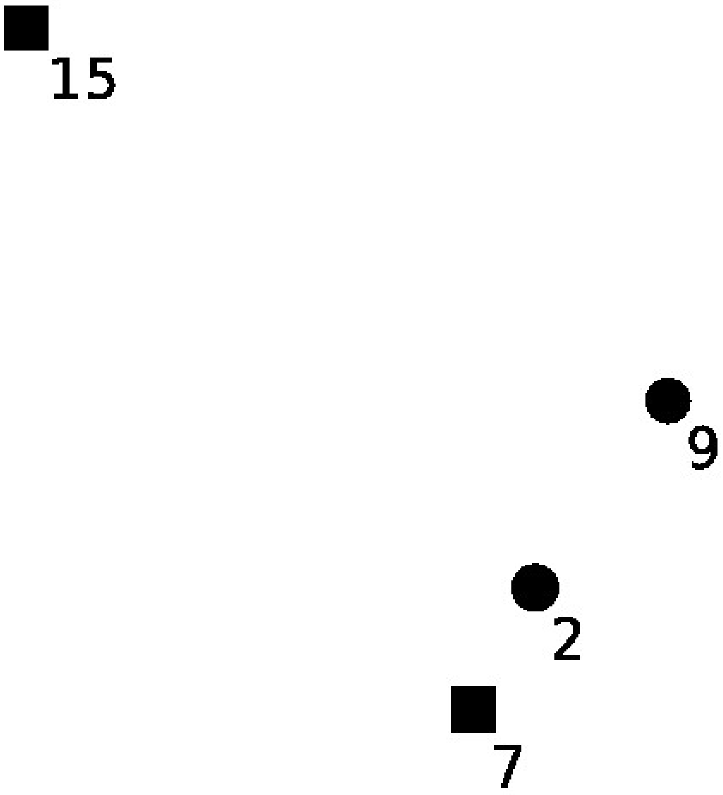
The dynamic RHC algorithm (4/4)



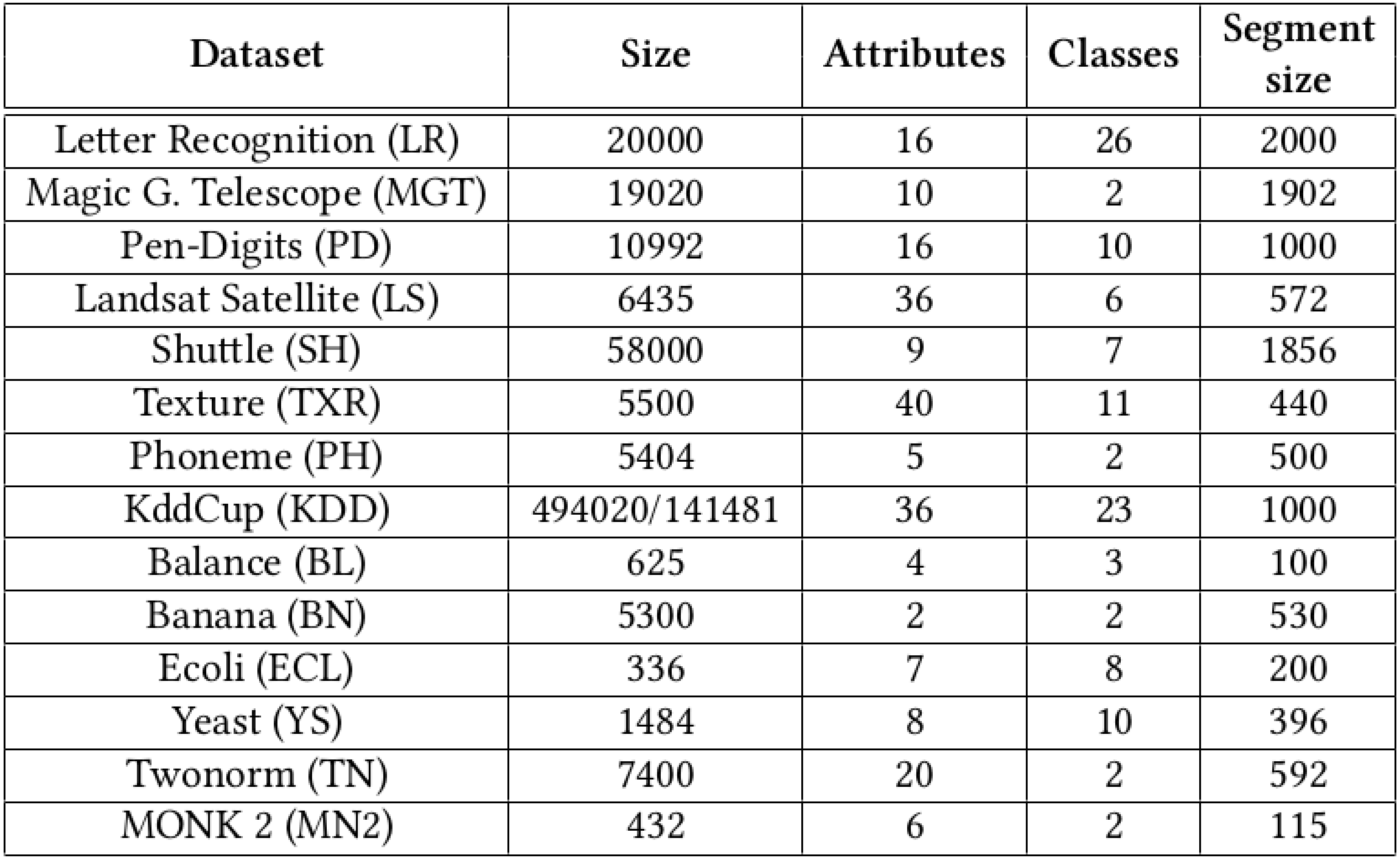
The dynamic RHC algorithm (4/4)



The dynamic RHC algorithm (4/4)



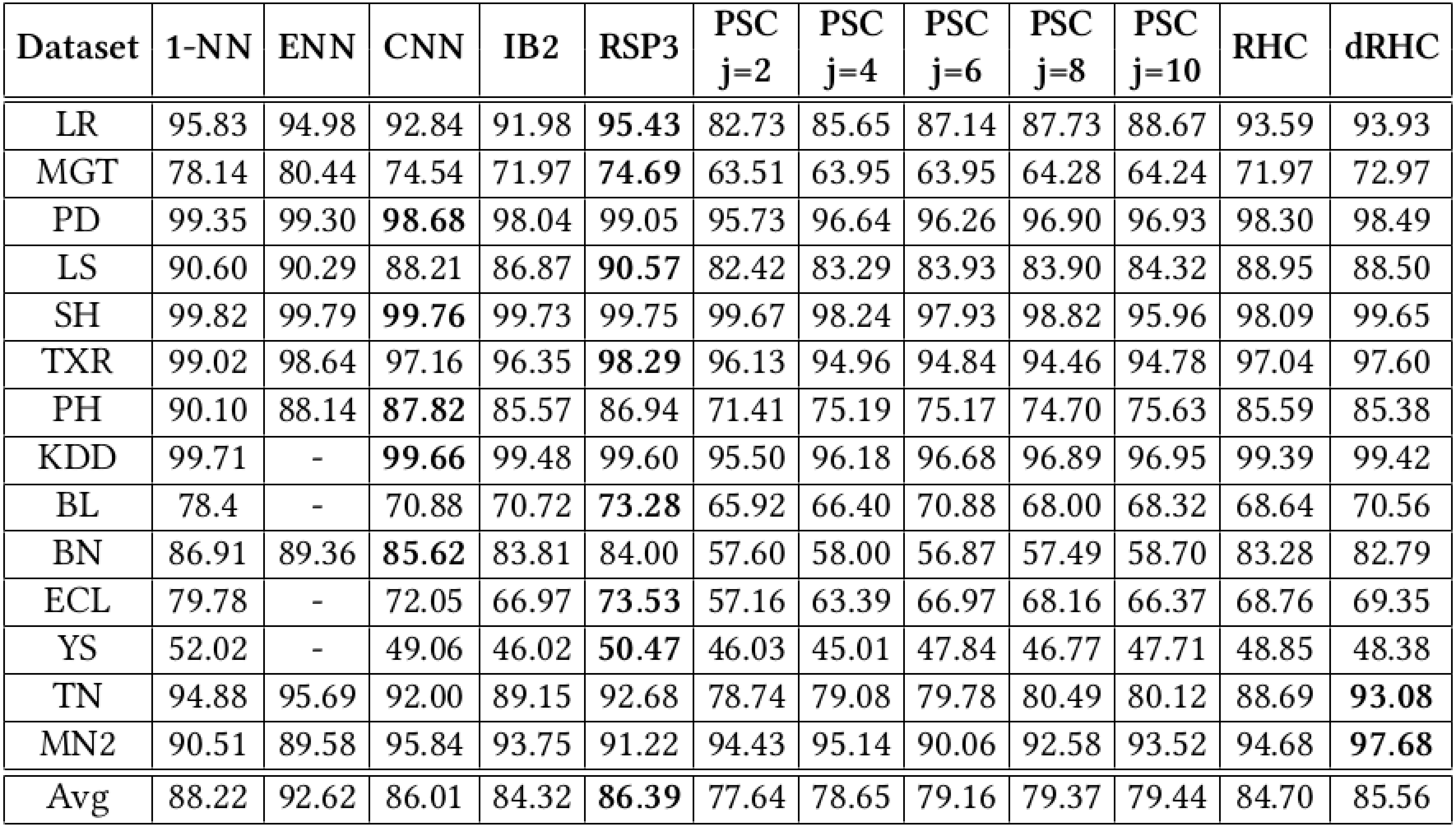
RHC & dRHC: Experimental study (1/9)



Dataset	Size	Attributes	Classes	Segment size
Letter Recognition (LR)	20000	16	26	2000
Magic G. Telescope (MGT)	19020	10	2	1902
Pen-Digits (PD)	10992	16	10	1000
Landsat Satellite (LS)	6435	36	6	572
Shuttle (SH)	58000	9	7	1856
Texture (TXR)	5500	40	11	440
Phoneme (PH)	5404	5	2	500
KddCup (KDD)	494020/141481	36	23	1000
Balance (BL)	625	4	3	100
Banana (BN)	5300	2	2	530
Ecoli (ECL)	336	7	8	200
Yeast (YS)	1484	8	10	396
Twonorm (TN)	7400	20	2	592
MONK 2 (MN2)	432	6	2	115

RHC & dRHC: Experimental study (2/9)

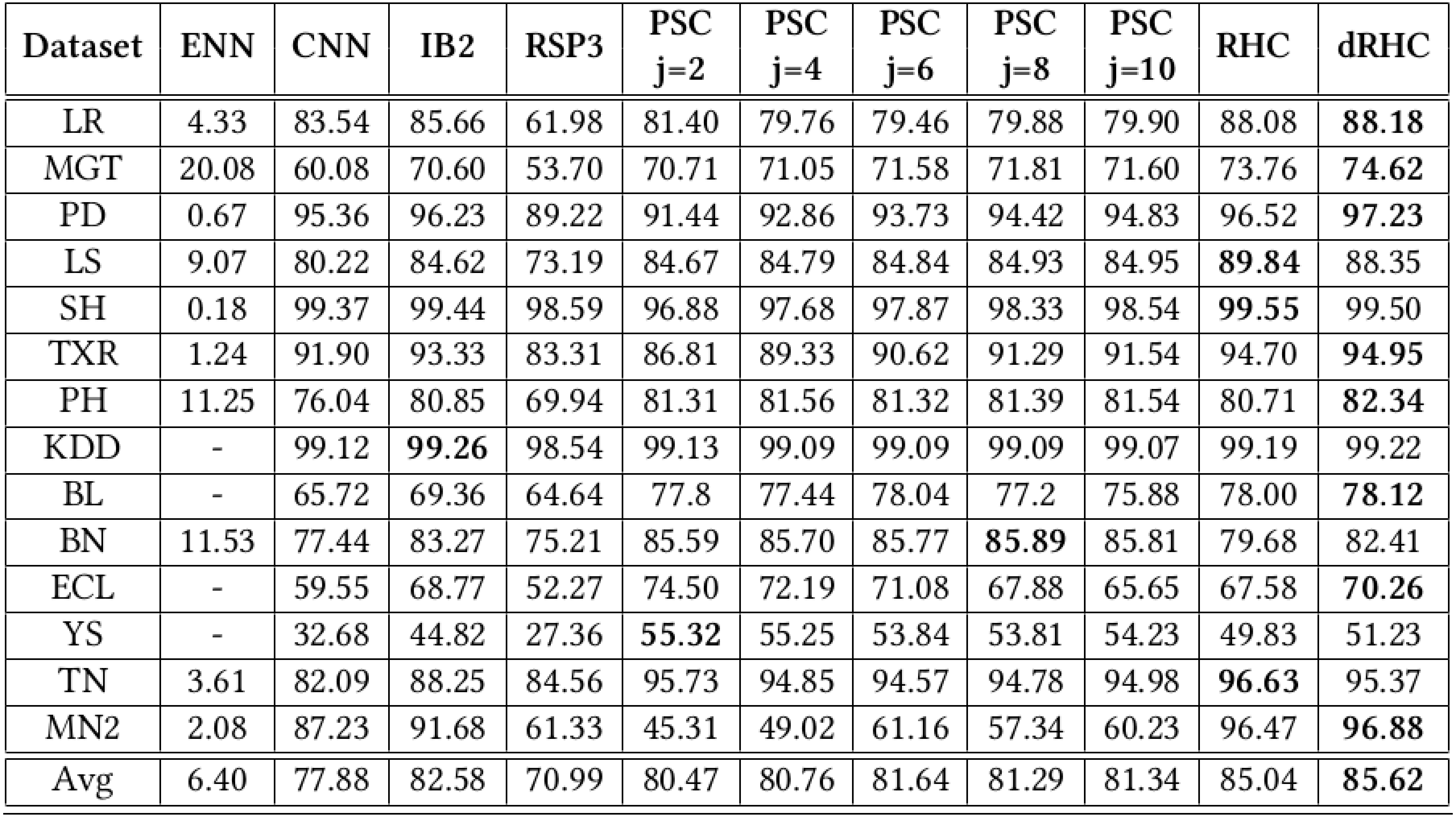
Accuracy / non-edited data



Dataset	1-NN	ENN	CNN	IB2	RSP3	PSC j=2	PSC j=4	PSC j=6	PSC j=8	PSC j=10	RHC	dRHC
LR	95.83	94.98	92.84	91.98	**95.43**	82.73	85.65	87.14	87.73	88.67	93.59	93.93
MGT	78.14	80.44	74.54	71.97	**74.69**	63.51	63.95	63.95	64.28	64.24	71.97	72.97
PD	99.35	99.30	**98.68**	98.04	99.05	95.73	96.64	96.26	96.90	96.93	98.30	98.49
LS	90.60	90.29	88.21	86.87	**90.57**	82.42	83.29	83.93	83.90	84.32	88.95	88.50
SH	99.82	99.79	**99.76**	99.73	99.75	99.67	98.24	97.93	98.82	95.96	98.09	99.65
TXR	99.02	98.64	97.16	96.35	**98.29**	96.13	94.96	94.84	94.46	94.78	97.04	97.60
PH	90.10	88.14	**87.82**	85.57	86.94	71.41	75.19	75.17	74.70	75.63	85.59	85.38
KDD	99.71	-	**99.66**	99.48	99.60	95.50	96.18	96.68	96.89	96.95	99.39	99.42
BL	78.4	-	70.88	70.72	**73.28**	65.92	66.40	70.88	68.00	68.32	68.64	70.56
BN	86.91	89.36	**85.62**	83.81	84.00	57.60	58.00	56.87	57.49	58.70	83.28	82.79
ECL	79.78	-	72.05	66.97	**73.53**	57.16	63.39	66.97	68.16	66.37	68.76	69.35
YS	52.02	-	49.06	46.02	**50.47**	46.03	45.01	47.84	46.77	47.71	48.85	48.38
TN	94.88	95.69	92.00	89.15	92.68	78.74	79.08	79.78	80.49	80.12	88.69	**93.08**
MN2	90.51	89.58	95.84	93.75	91.22	94.43	95.14	90.06	92.58	93.52	94.68	**97.68**
Avg	88.22	92.62	86.01	84.32	**86.39**	77.64	78.65	79.16	79.37	79.44	84.70	85.56

RHC & dRHC: Experimental study (3/9)

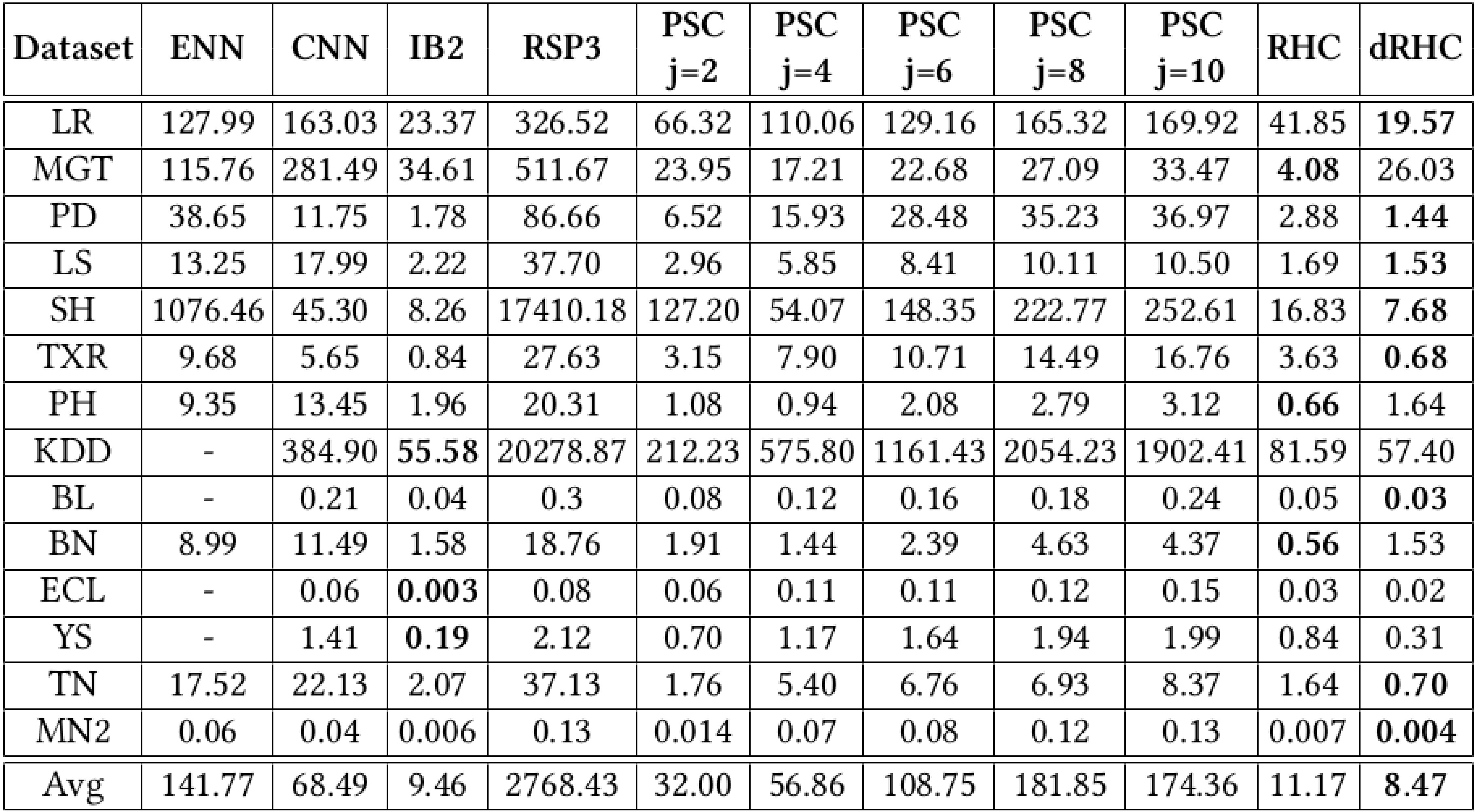
Reduction Rate / non-edited data



Dataset	ENN	CNN	IB2	RSP3	PSC j=2	PSC j=4	PSC j=6	PSC j=8	PSC j=10	RHC	dRHC
LR	4.33	83.54	85.66	61.98	81.40	79.76	79.46	79.88	79.90	88.08	**88.18**
MGT	20.08	60.08	70.60	53.70	70.71	71.05	71.58	71.81	71.60	73.76	**74.62**
PD	0.67	95.36	96.23	89.22	91.44	92.86	93.73	94.42	94.83	96.52	**97.23**
LS	9.07	80.22	84.62	73.19	84.67	84.79	84.84	84.93	84.95	**89.84**	88.35
SH	0.18	99.37	99.44	98.59	96.88	97.68	97.87	98.33	98.54	**99.55**	99.50
TXR	1.24	91.90	93.33	83.31	86.81	89.33	90.62	91.29	91.54	94.70	**94.95**
PH	11.25	76.04	80.85	69.94	81.31	81.56	81.32	81.39	81.54	80.71	**82.34**
KDD	-	99.12	**99.26**	98.54	99.13	99.09	99.09	99.09	99.07	99.19	99.22
BL	-	65.72	69.36	64.64	77.8	77.44	78.04	77.2	75.88	78.00	**78.12**
BN	11.53	77.44	83.27	75.21	85.59	85.70	85.77	**85.89**	85.81	79.68	82.41
ECL	-	59.55	68.77	52.27	74.50	72.19	71.08	67.88	65.65	67.58	**70.26**
YS	-	32.68	44.82	27.36	**55.32**	55.25	53.84	53.81	54.23	49.83	51.23
TN	3.61	82.09	88.25	84.56	95.73	94.85	94.57	94.78	94.98	**96.63**	95.37
MN2	2.08	87.23	91.68	61.33	45.31	49.02	61.16	57.34	60.23	96.47	**96.88**
Avg	6.40	77.88	82.58	70.99	80.47	80.76	81.64	81.29	81.34	85.04	**85.62**

RHC & dRHC: Experimental study (4/9)

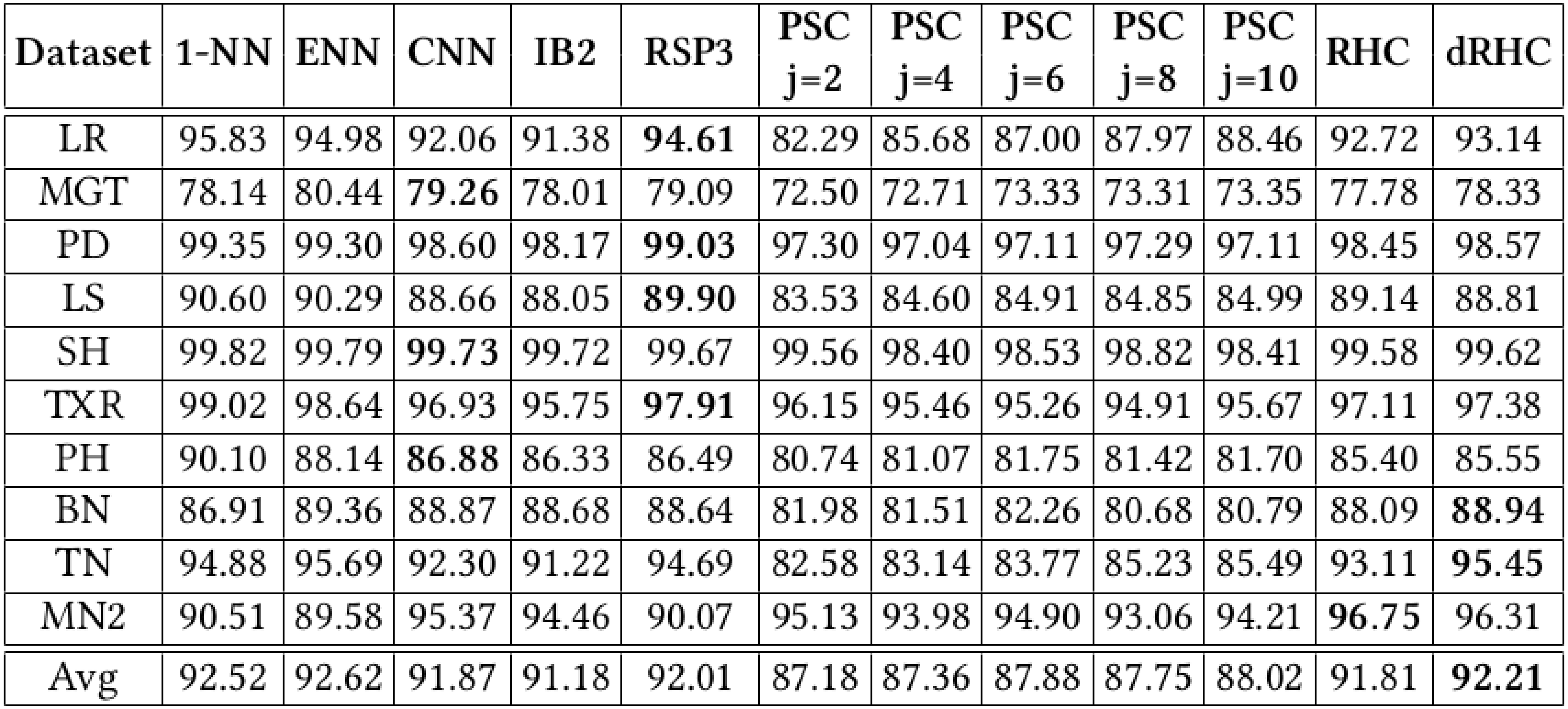
Preprocessing Cost / non-edited data



Dataset	ENN	CNN	IB2	RSP3	PSC j=2	PSC j=4	PSC j=6	PSC j=8	PSC j=10	RHC	dRHC
LR	127.99	163.03	23.37	326.52	66.32	110.06	129.16	165.32	169.92	41.85	**19.57**
MGT	115.76	281.49	34.61	511.67	23.95	17.21	22.68	27.09	33.47	**4.08**	26.03
PD	38.65	11.75	1.78	86.66	6.52	15.93	28.48	35.23	36.97	2.88	**1.44**
LS	13.25	17.99	2.22	37.70	2.96	5.85	8.41	10.11	10.50	1.69	**1.53**
SH	1076.46	45.30	8.26	17410.18	127.20	54.07	148.35	222.77	252.61	16.83	**7.68**
TXR	9.68	5.65	0.84	27.63	3.15	7.90	10.71	14.49	16.76	3.63	**0.68**
PH	9.35	13.45	1.96	20.31	1.08	0.94	2.08	2.79	3.12	**0.66**	1.64
KDD	-	384.90	**55.58**	20278.87	212.23	575.80	1161.43	2054.23	1902.41	81.59	57.40
BL	-	0.21	0.04	0.3	0.08	0.12	0.16	0.18	0.24	0.05	**0.03**
BN	8.99	11.49	1.58	18.76	1.91	1.44	2.39	4.63	4.37	**0.56**	1.53
ECL	-	0.06	**0.003**	0.08	0.06	0.11	0.11	0.12	0.15	0.03	0.02
YS	-	1.41	**0.19**	2.12	0.70	1.17	1.64	1.94	1.99	0.84	0.31
TN	17.52	22.13	2.07	37.13	1.76	5.40	6.76	6.93	8.37	1.64	**0.70**
MN2	0.06	0.04	0.006	0.13	0.014	0.07	0.08	0.12	0.13	0.007	**0.004**
Avg	141.77	68.49	9.46	2768.43	32.00	56.86	108.75	181.85	174.36	11.17	**8.47**

RHC & dRHC: Experimental study (5/9)

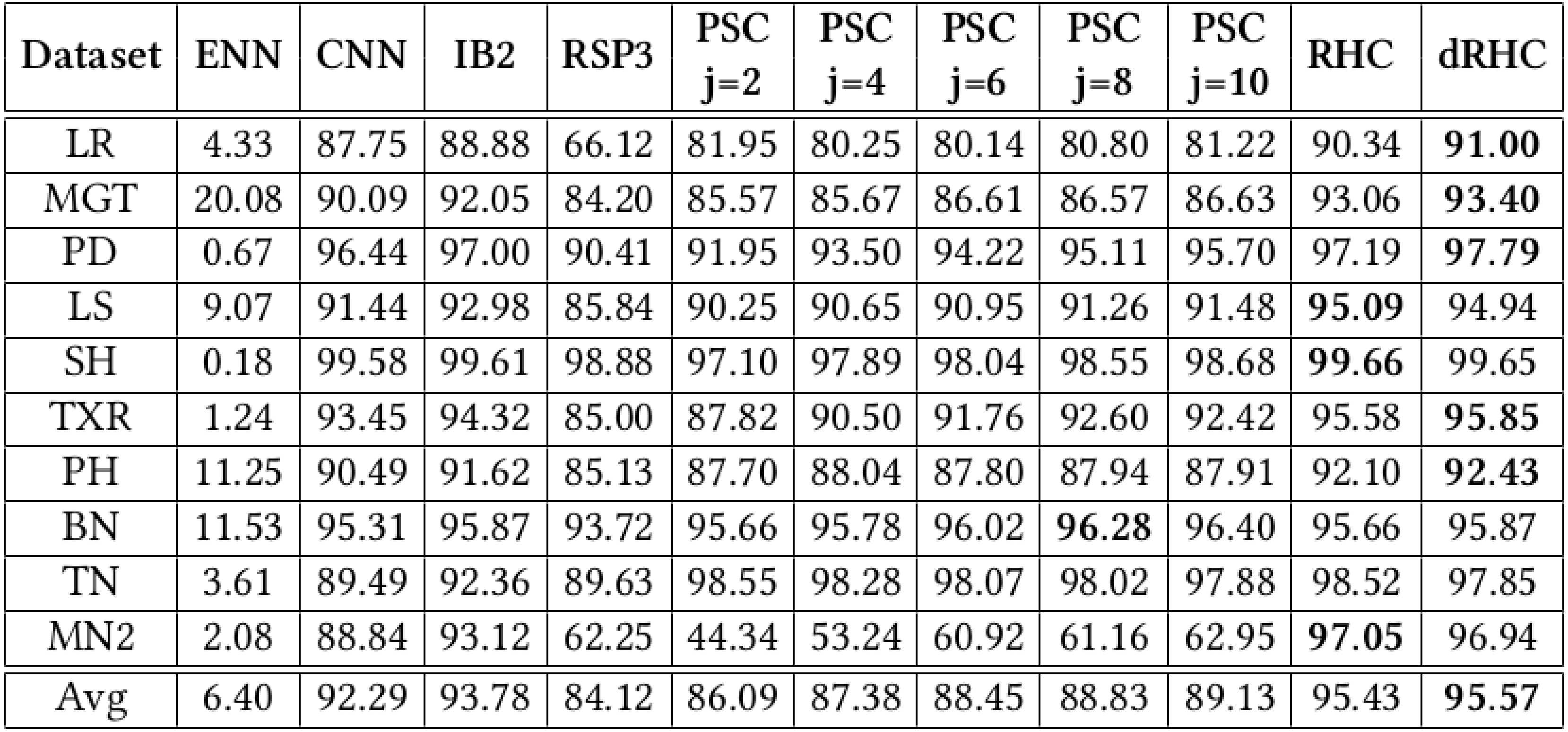
Accuracy / edited data



Dataset	**1-NN**	**ENN**	**CNN**	**IB2**	**RSP3**	**PSC j=2**	**PSC j=4**	**PSC j=6**	**PSC j=8**	**PSC j=10**	**RHC**	**dRHC**
LR	95.83	94.98	92.06	91.38	**94.61**	82.29	85.68	87.00	87.97	88.46	92.72	93.14
MGT	78.14	80.44	**79.26**	78.01	79.09	72.50	72.71	73.33	73.31	73.35	77.78	78.33
PD	99.35	99.30	98.60	98.17	**99.03**	97.30	97.04	97.11	97.29	97.11	98.45	98.57
LS	90.60	90.29	88.66	88.05	**89.90**	83.53	84.60	84.91	84.85	84.99	89.14	88.81
SH	99.82	99.79	**99.73**	99.72	99.67	99.56	98.40	98.53	98.82	98.41	99.58	99.62
TXR	99.02	98.64	96.93	95.75	**97.91**	96.15	95.46	95.26	94.91	95.67	97.11	97.38
PH	90.10	88.14	**86.88**	86.33	86.49	80.74	81.07	81.75	81.42	81.70	85.40	85.55
BN	86.91	89.36	88.87	88.68	88.64	81.98	81.51	82.26	80.68	80.79	88.09	**88.94**
TN	94.88	95.69	92.30	91.22	94.69	82.58	83.14	83.77	85.23	85.49	93.11	**95.45**
MN2	90.51	89.58	95.37	94.46	90.07	95.13	93.98	94.90	93.06	94.21	**96.75**	96.31
Avg	92.52	92.62	91.87	91.18	92.01	87.18	87.36	87.88	87.75	88.02	91.81	**92.21**

RHC & dRHC: Experimental study (6/9)

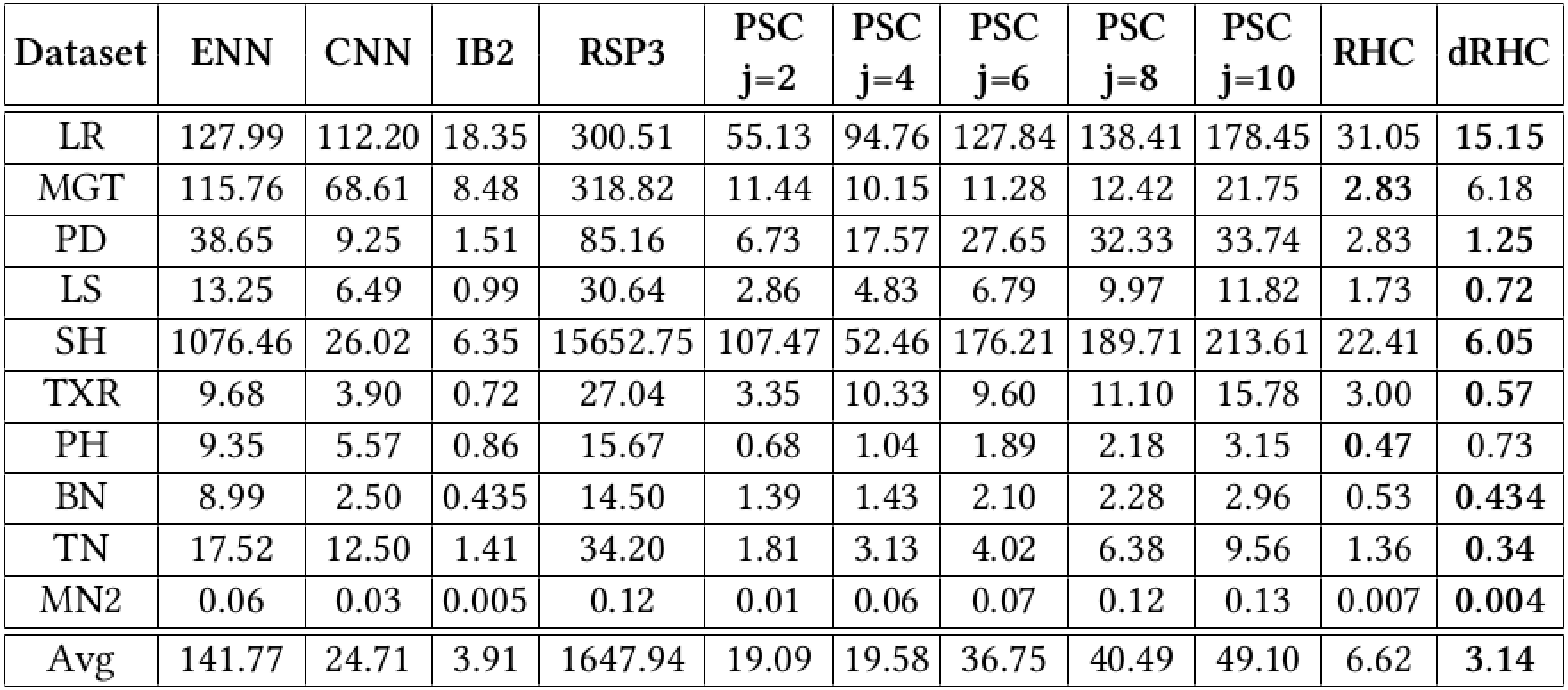
Reduction Rate / edited data



Dataset	ENN	CNN	IB2	RSP3	PSC j=2	PSC j=4	PSC j=6	PSC j=8	PSC j=10	RHC	dRHC
LR	4.33	87.75	88.88	66.12	81.95	80.25	80.14	80.80	81.22	90.34	**91.00**
MGT	20.08	90.09	92.05	84.20	85.57	85.67	86.61	86.57	86.63	93.06	**93.40**
PD	0.67	96.44	97.00	90.41	91.95	93.50	94.22	95.11	95.70	97.19	**97.79**
LS	9.07	91.44	92.98	85.84	90.25	90.65	90.95	91.26	91.48	**95.09**	94.94
SH	0.18	99.58	99.61	98.88	97.10	97.89	98.04	98.55	98.68	**99.66**	99.65
TXR	1.24	93.45	94.32	85.00	87.82	90.50	91.76	92.60	92.42	95.58	**95.85**
PH	11.25	90.49	91.62	85.13	87.70	88.04	87.80	87.94	87.91	92.10	**92.43**
BN	11.53	95.31	95.87	93.72	95.66	95.78	96.02	**96.28**	96.40	95.66	95.87
TN	3.61	89.49	92.36	89.63	98.55	98.28	98.07	98.02	97.88	98.52	97.85
MN2	2.08	88.84	93.12	62.25	44.34	53.24	60.92	61.16	62.95	**97.05**	96.94
Avg	6.40	92.29	93.78	84.12	86.09	87.38	88.45	88.83	89.13	95.43	**95.57**

RHC & dRHC: Experimental study (7/9)

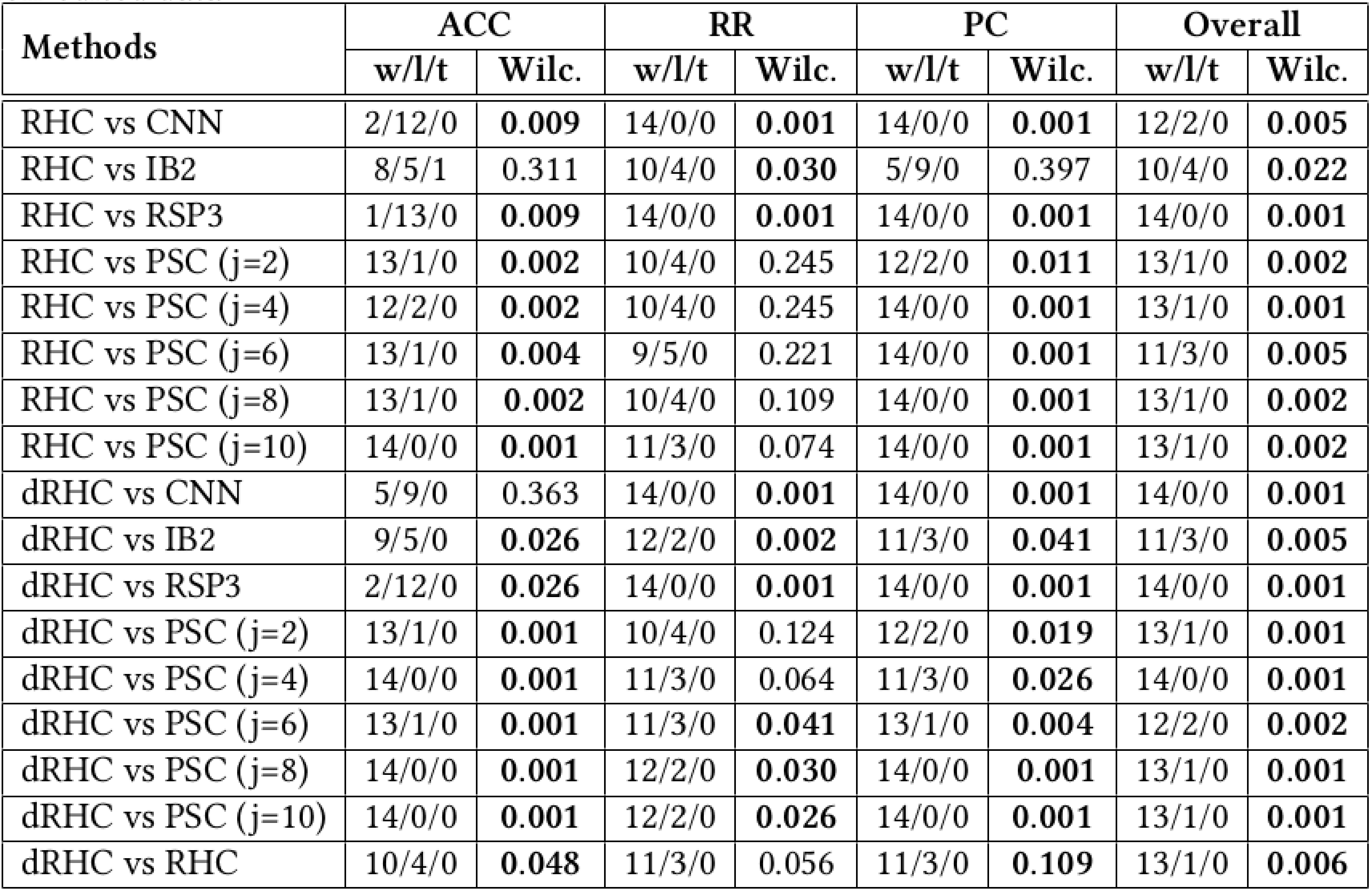
Preprocessing Cost / edited data



Dataset	**ENN**	**CNN**	**IB2**	**RSP3**	**PSC j=2**	**PSC j=4**	**PSC j=6**	**PSC j=8**	**PSC j=10**	**RHC**	**dRHC**
LR	127.99	112.20	18.35	300.51	55.13	94.76	127.84	138.41	178.45	31.05	**15.15**
MGT	115.76	68.61	8.48	318.82	11.44	10.15	11.28	12.42	21.75	**2.83**	6.18
PD	38.65	9.25	1.51	85.16	6.73	17.57	27.65	32.33	33.74	2.83	**1.25**
LS	13.25	6.49	0.99	30.64	2.86	4.83	6.79	9.97	11.82	1.73	**0.72**
SH	1076.46	26.02	6.35	15652.75	107.47	52.46	176.21	189.71	213.61	22.41	**6.05**
TXR	9.68	3.90	0.72	27.04	3.35	10.33	9.60	11.10	15.78	3.00	**0.57**
PH	9.35	5.57	0.86	15.67	0.68	1.04	1.89	2.18	3.15	**0.47**	0.73
BN	8.99	2.50	0.435	14.50	1.39	1.43	2.10	2.28	2.96	0.53	**0.434**
TN	17.52	12.50	1.41	34.20	1.81	3.13	4.02	6.38	9.56	1.36	**0.34**
MN2	0.06	0.03	0.005	0.12	0.01	0.06	0.07	0.12	0.13	0.007	**0.004**
Avg	141.77	24.71	3.91	1647.94	19.09	19.58	36.75	40.49	49.10	6.62	**3.14**

RHC & dRHC: Experimental study (8/9)

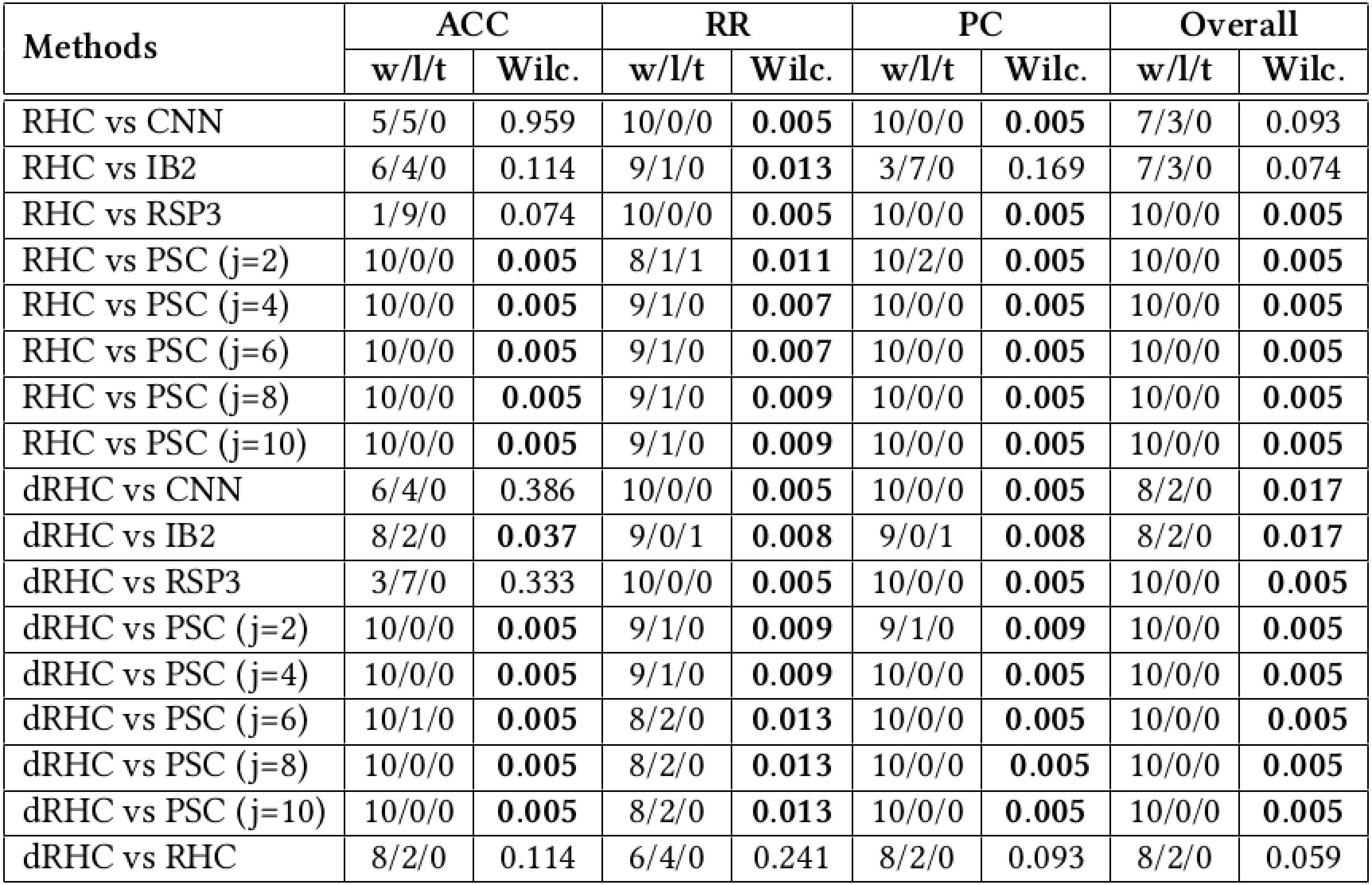
Wilcoxon signed ranks tests / non-edited data



Methods	ACC		RR		PC		Overall	
	w/l/t	Wilc.	w/l/t	Wilc.	w/l/t	Wilc.	w/l/t	Wilc.
RHC vs CNN	2/12/0	**0.009**	14/0/0	**0.001**	14/0/0	**0.001**	12/2/0	**0.005**
RHC vs IB2	8/5/1	0.311	10/4/0	**0.030**	5/9/0	0.397	10/4/0	**0.022**
RHC vs RSP3	1/13/0	**0.009**	14/0/0	**0.001**	14/0/0	**0.001**	14/0/0	**0.001**
RHC vs PSC (j=2)	13/1/0	**0.002**	10/4/0	0.245	12/2/0	**0.011**	13/1/0	**0.002**
RHC vs PSC (j=4)	12/2/0	**0.002**	10/4/0	0.245	14/0/0	**0.001**	13/1/0	**0.001**
RHC vs PSC (j=6)	13/1/0	**0.004**	9/5/0	0.221	14/0/0	**0.001**	11/3/0	**0.005**
RHC vs PSC (j=8)	13/1/0	**0.002**	10/4/0	0.109	14/0/0	**0.001**	13/1/0	**0.002**
RHC vs PSC (j=10)	14/0/0	**0.001**	11/3/0	0.074	14/0/0	**0.001**	13/1/0	**0.002**
dRHC vs CNN	5/9/0	0.363	14/0/0	**0.001**	14/0/0	**0.001**	14/0/0	**0.001**
dRHC vs IB2	9/5/0	**0.026**	12/2/0	**0.002**	11/3/0	**0.041**	11/3/0	**0.005**
dRHC vs RSP3	2/12/0	**0.026**	14/0/0	**0.001**	14/0/0	**0.001**	14/0/0	**0.001**
dRHC vs PSC (j=2)	13/1/0	**0.001**	10/4/0	0.124	12/2/0	**0.019**	13/1/0	**0.001**
dRHC vs PSC (j=4)	14/0/0	**0.001**	11/3/0	0.064	11/3/0	**0.026**	14/0/0	**0.001**
dRHC vs PSC (j=6)	13/1/0	**0.001**	11/3/0	**0.041**	13/1/0	**0.004**	12/2/0	**0.002**
dRHC vs PSC (j=8)	14/0/0	**0.001**	12/2/0	**0.030**	14/0/0	**0.001**	13/1/0	**0.001**
dRHC vs PSC (j=10)	14/0/0	**0.001**	12/2/0	**0.026**	14/0/0	**0.001**	13/1/0	**0.001**
dRHC vs RHC	10/4/0	**0.048**	11/3/0	0.056	11/3/0	**0.109**	13/1/0	**0.006**

RHC & dRHC: Experimental study (9/9)

Wilcoxon signed ranks tests / edited data



Methods	ACC		RR		PC		Overall	
	w/l/t	Wilc.	w/l/t	Wilc.	w/l/t	Wilc.	w/l/t	Wilc.
RHC vs CNN	5/5/0	0.959	10/0/0	**0.005**	10/0/0	**0.005**	7/3/0	0.093
RHC vs IB2	6/4/0	0.114	9/1/0	**0.013**	3/7/0	0.169	7/3/0	0.074
RHC vs RSP3	1/9/0	0.074	10/0/0	**0.005**	10/0/0	**0.005**	10/0/0	**0.005**
RHC vs PSC (j=2)	10/0/0	**0.005**	8/1/1	**0.011**	10/2/0	**0.005**	10/0/0	**0.005**
RHC vs PSC (j=4)	10/0/0	**0.005**	9/1/0	**0.007**	10/0/0	**0.005**	10/0/0	**0.005**
RHC vs PSC (j=6)	10/0/0	**0.005**	9/1/0	**0.007**	10/0/0	**0.005**	10/0/0	**0.005**
RHC vs PSC (j=8)	10/0/0	**0.005**	9/1/0	**0.009**	10/0/0	**0.005**	10/0/0	**0.005**
RHC vs PSC (j=10)	10/0/0	**0.005**	9/1/0	**0.009**	10/0/0	**0.005**	10/0/0	**0.005**
dRHC vs CNN	6/4/0	0.386	10/0/0	**0.005**	10/0/0	**0.005**	8/2/0	**0.017**
dRHC vs IB2	8/2/0	**0.037**	9/0/1	**0.008**	9/0/1	**0.008**	8/2/0	**0.017**
dRHC vs RSP3	3/7/0	0.333	10/0/0	**0.005**	10/0/0	**0.005**	10/0/0	**0.005**
dRHC vs PSC (j=2)	10/0/0	**0.005**	9/1/0	**0.009**	9/1/0	**0.009**	10/0/0	**0.005**
dRHC vs PSC (j=4)	10/0/0	**0.005**	9/1/0	**0.009**	10/0/0	**0.005**	10/0/0	**0.005**
dRHC vs PSC (j=6)	10/1/0	**0.005**	8/2/0	**0.013**	10/0/0	**0.005**	10/0/0	**0.005**
dRHC vs PSC (j=8)	10/0/0	**0.005**	8/2/0	**0.013**	10/0/0	**0.005**	10/0/0	**0.005**
dRHC vs PSC (j=10)	10/0/0	**0.005**	8/2/0	**0.013**	10/0/0	**0.005**	10/0/0	**0.005**
dRHC vs RHC	8/2/0	0.114	6/4/0	0.241	8/2/0	0.093	8/2/0	0.059

Editing through Homogeneous Clusters (1/3)

Motivation/Drawbacks of editing algorithms:

- Since all editing algorithms either extend ENN-rule or are based on the same idea, they are parametric. Their performance is dependent on costly trial-and-error procedures
- They require high preprocessing cost

Contribution

- Development of a novel, fast, non-parametric editing algorithm that is based on a *k*-means clustering procedure that forms homogeneous clusters

Editing through Homogeneous Clusters (2/3)

EHC properties

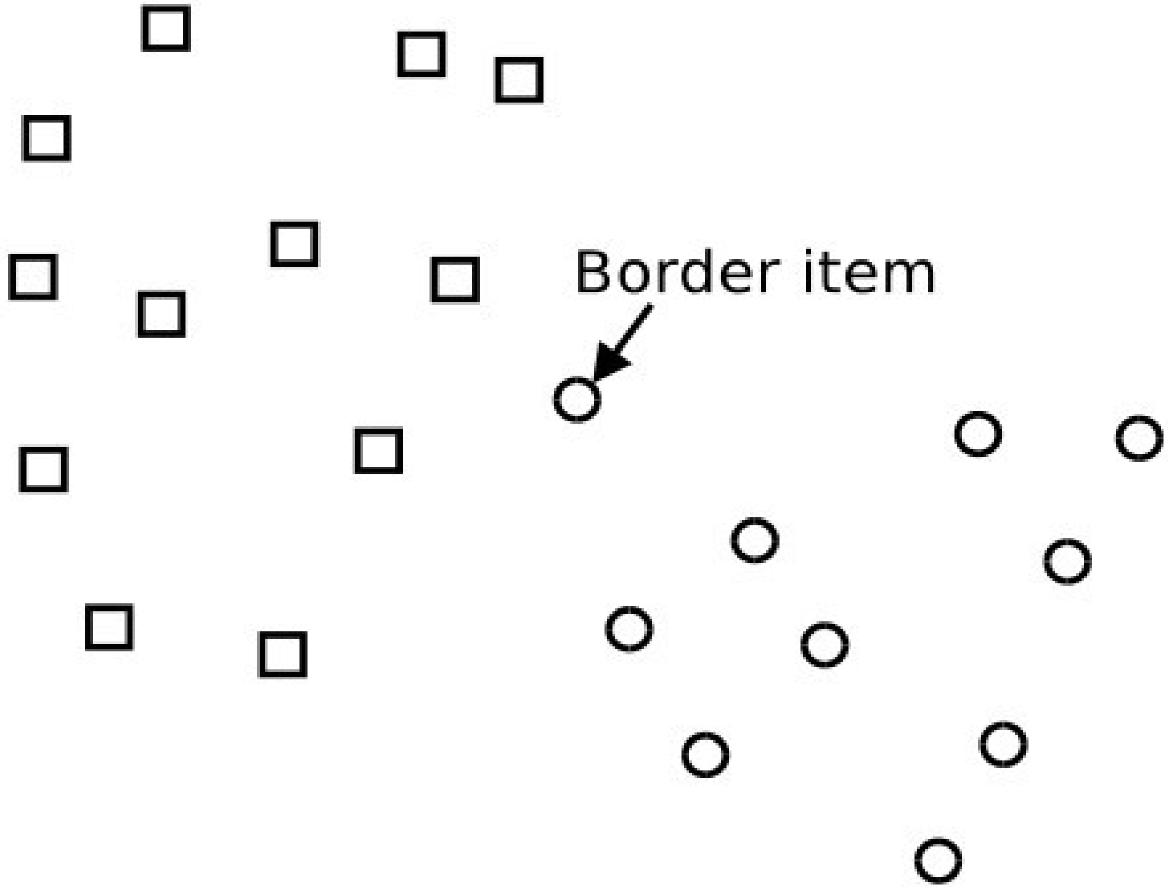
- It follows completely different strategy from that of ENN-based approaches
- Fast execution
- Non-parametric
- It is based on *k*-means clustering

EHC idea:

- It continues constructing clusters until all of them are homogeneous
- It removes the clusters that contain only one item (they are considered as outliers, noise or close-border items)

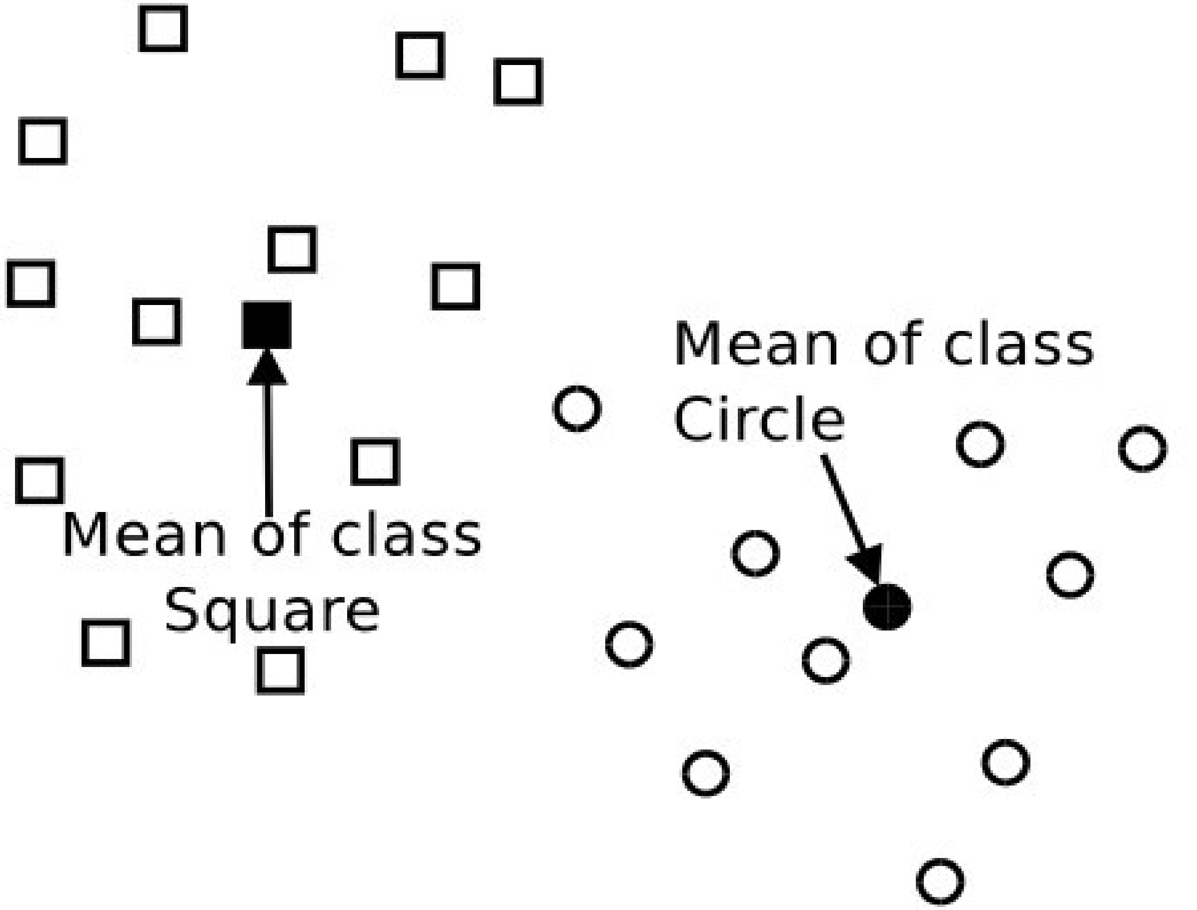
Editing through Homogeneous Clusters (3/3)

Removal of a border item



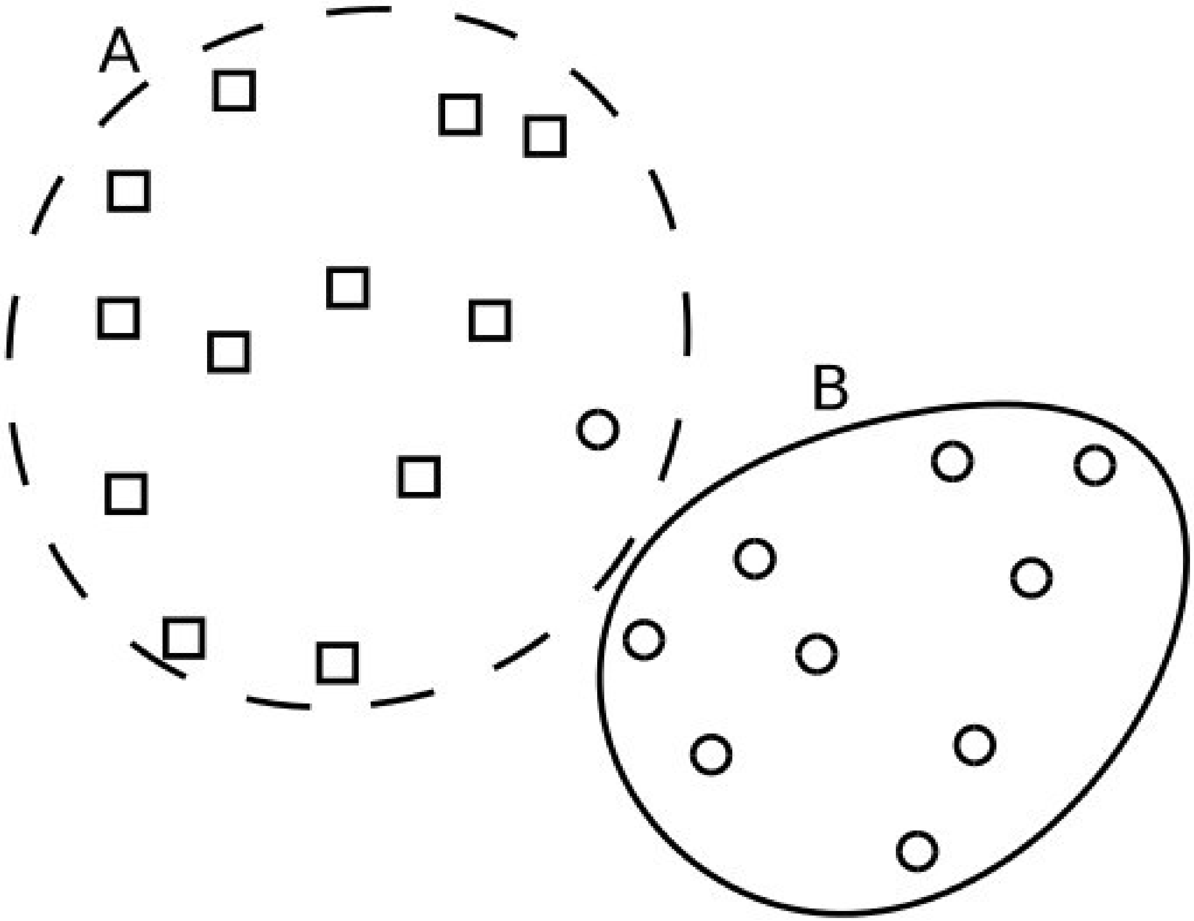
Editing through Homogeneous Clusters (3/3)

Removal of a border item



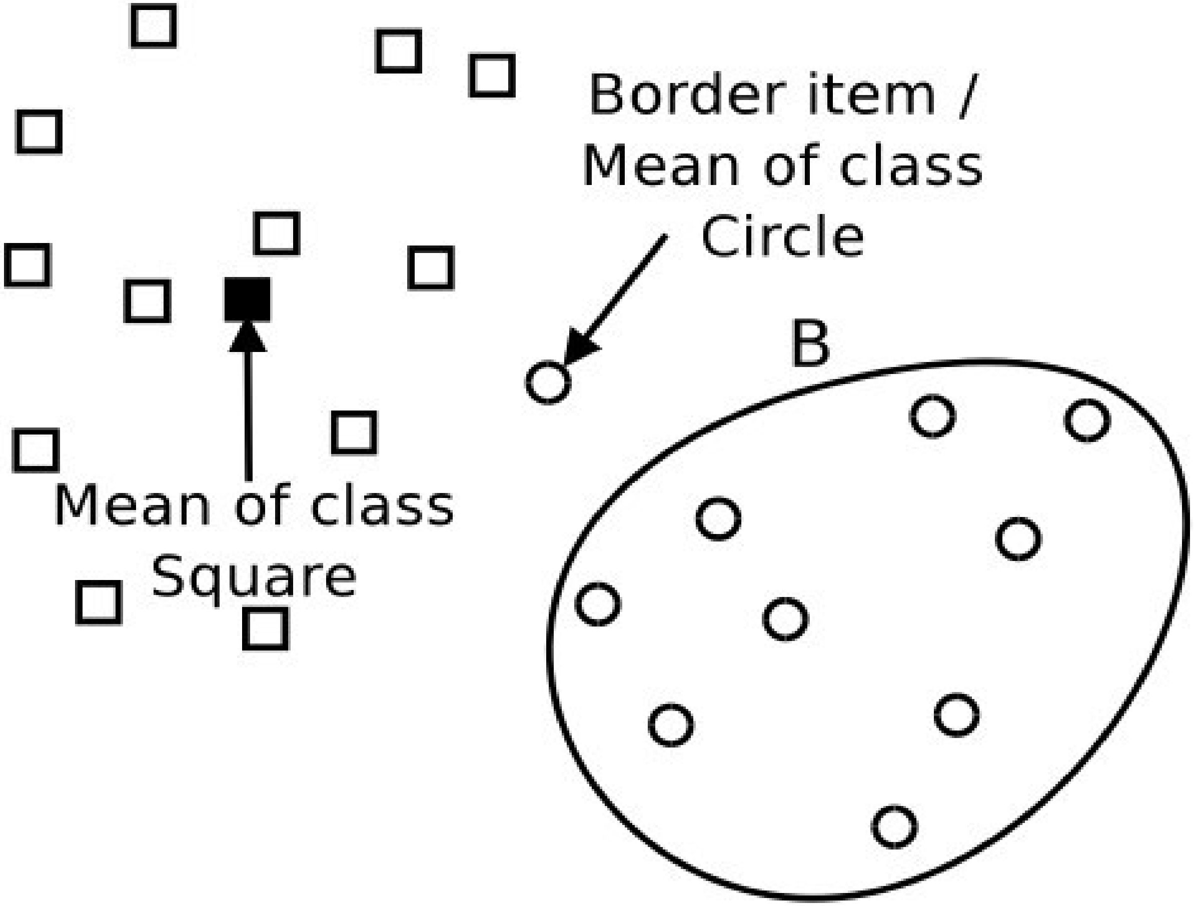
Editing through Homogeneous Clusters (3/3)

Removal of a border item



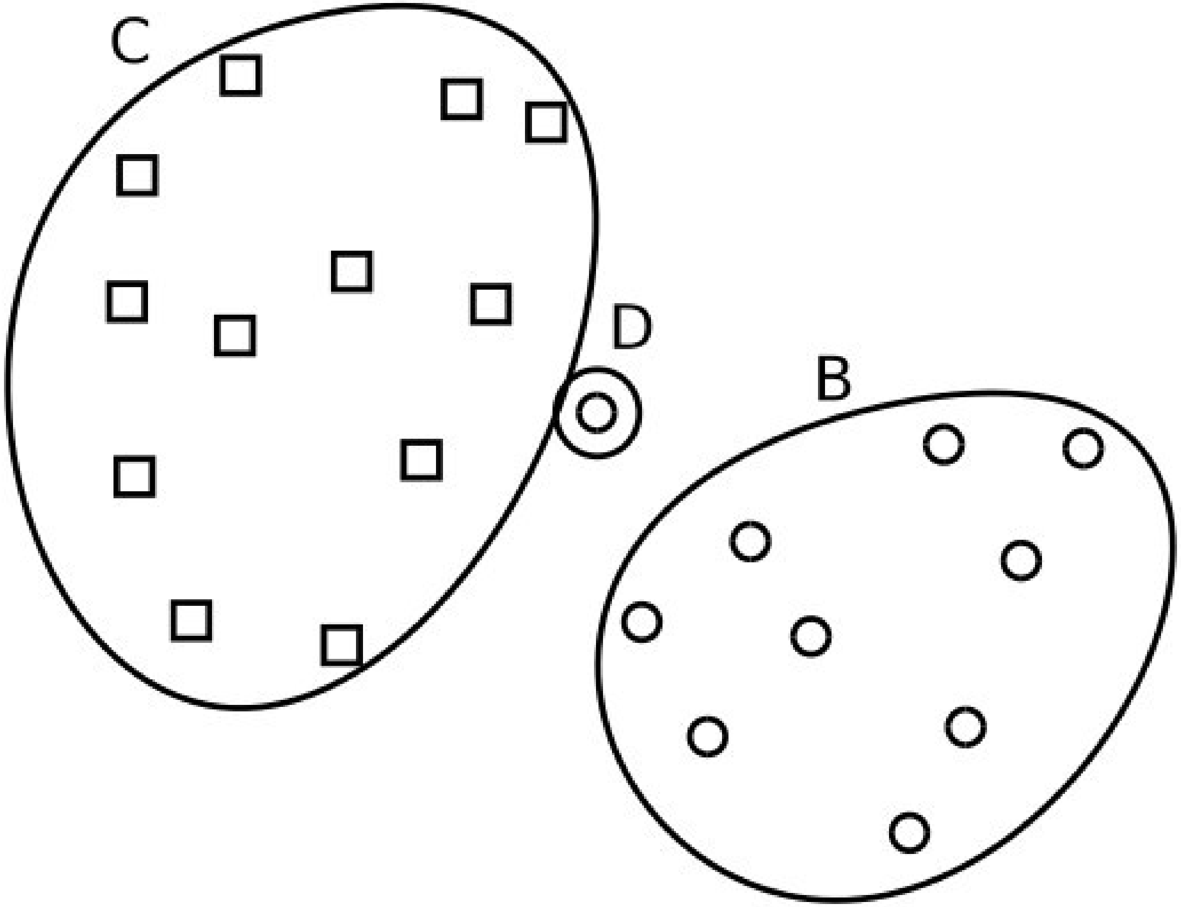
Editing through Homogeneous Clusters (3/3)

Removal of a border item



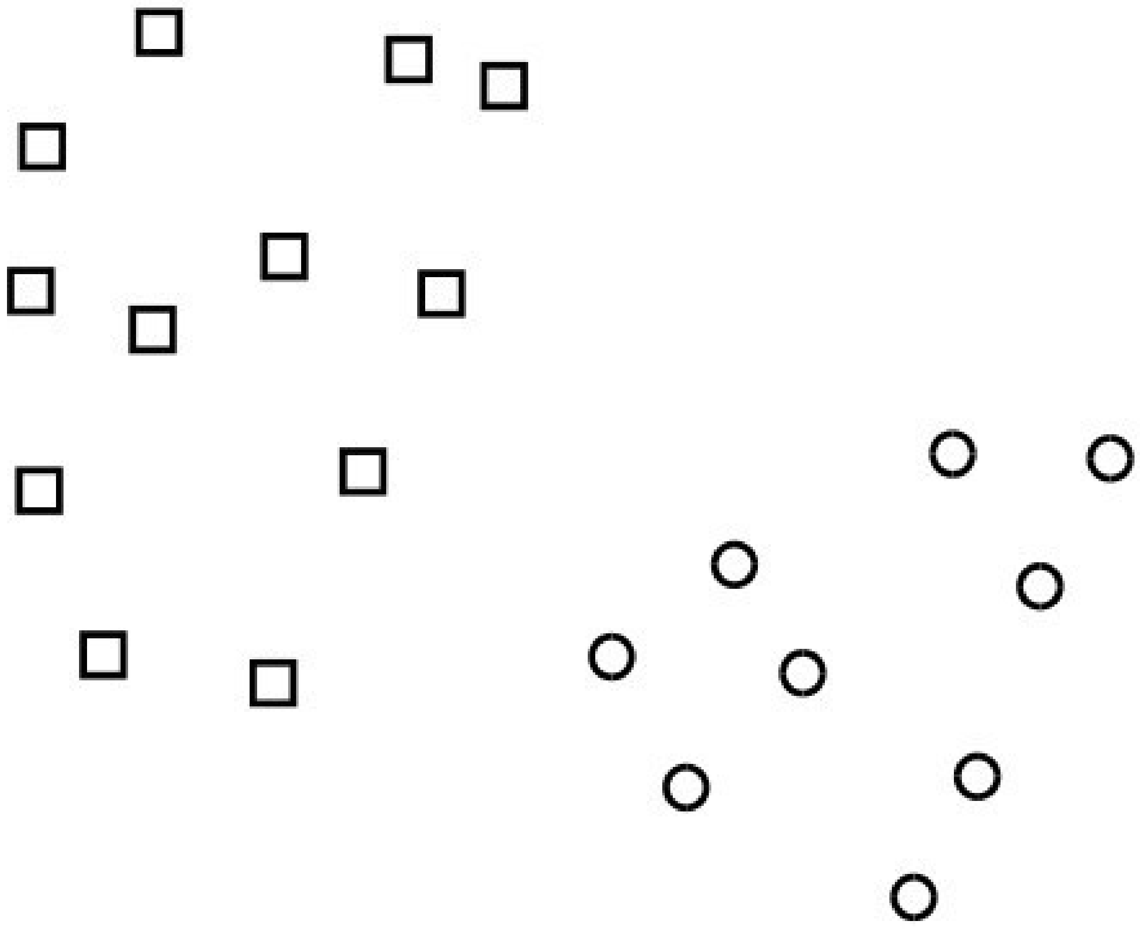
Editing through Homogeneous Clusters (3/3)

Removal of a border item



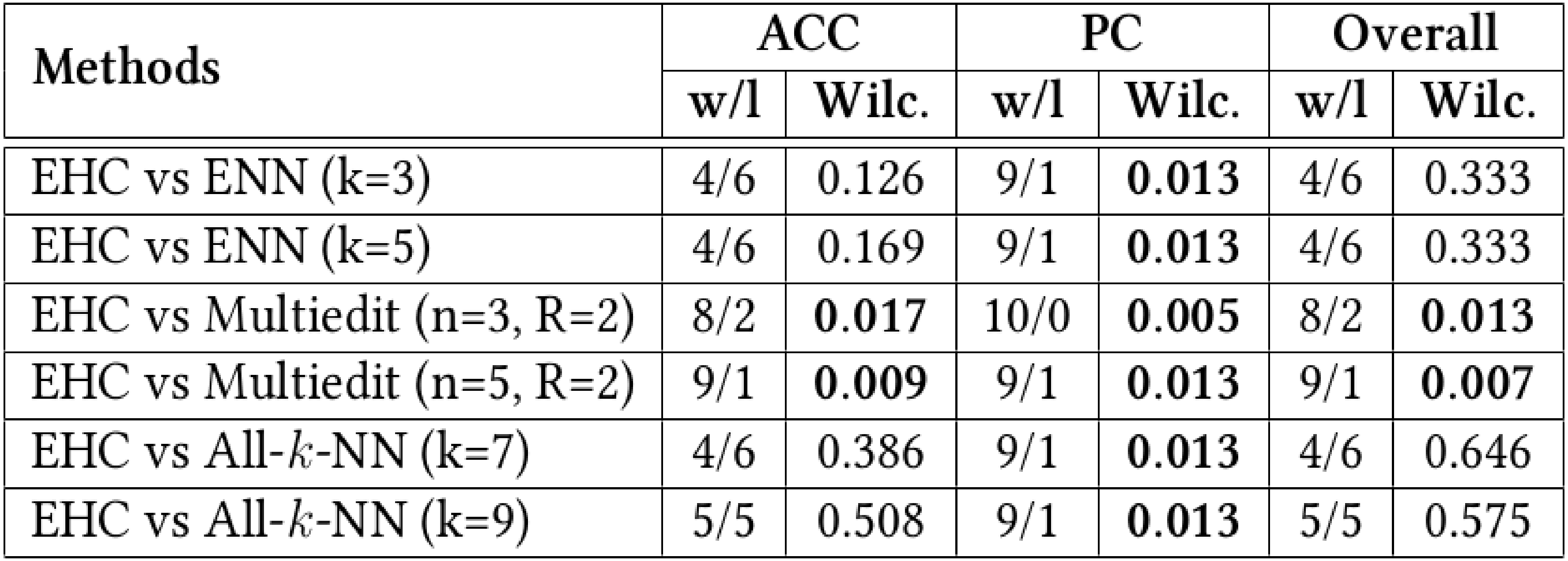
Editing through Homogeneous Clusters (3/3)

Removal of a border item



Dataset		1-NN	ENN	ENN	Multiedit	Multiedit	All$k$NN	All$k$NN	EHC
			($k=3$)	($k=5$)	($n=3$, $R=2$)	($n=5$, $R=2$)	($k=7$)	($k=9$)	
MGT	Acc	78.14	80.44	80.57	76.75	75.26	80.76	**80.86**	79.52
	RR	-	20.08	19.20	39.98	42.36	29.67	30.38	10.70
	PC	-	115.76	115.76	2,839.55	1,447.93	115.76	115.76	**4.08**
LS	Acc	**90.60**	90.30	90.43	86.79	86.03	90.12	90.16	**90.55**
	RR	-	9.07	9.27	24.13	26.17	13.92	14.51	3.11
	PC	-	13.25	13.25	266.22	139.53	13.25	13.25	**1.69**
PH	Acc	**90.10**	88.14	87.53	80.77	79.72	86.55	86.23	**89.06**
	RR	-	11.25	11.93	34.14	36.91	17.92	19.30	7.36
	PC	-	9.35	9.35	166.22	53.71	9.35	9.35	**0.66**
LIR	Acc	**95.83**	94.98	94.87	70.94	58.35	94.28	94.00	**95.23**
	RR	-	4.33	4.44	43.43	56.59	7.31	7.97	3.95
	PC	-	127.99	127.99	7,214.38	2,900.53	127.99	127.99	**41.85**
BN	Acc	86.91	89.36	89.55	89.83	**90.38**	89.509	89.79	88.60
	RR	-	11.53	10.98	20.12	21.64	17.10	17.51	10.65
	PC	-	8.99	8.99	106.69	60.26	8.99	8.99	**0.56**
ECL	Acc	79.78	81.57	81.86	63.10	46.11	81.26	80.66	**82.16**
	RR	-	20.45	20.45	47.29	60.15	28.63	30.48	17.01
	PC	-	0.036	0.036	0.100	0.055	0.036	0.036	**0.035**
PM	Acc	68.36	71.87	71.75	71.36	68.89	72.65	**73.30**	70.32
	RR	-	30.16	29.43	53.07	58.96	45.56	46.24	16.59
	PC	-	0.19	0.19	0.51	0.26	0.19	0.19	**0.06**
YS	Acc	52.16	56.47	57.07	52.90	50.54	58.29	**58.42**	54.45
	RR	-	45.73	43.89	74.34	80.93	59.90	61.25	29.58
	PC	-	0.70	0.70	1.19	**0.58**	0.70	0.70	0.84
LS-n	Acc	82.58	89.64	89.74	86.47	85.55	89.73	**89.84**	87.55
	RR	-	19.82	18.45	38.33	40.19	29.64	30.17	10.93
	PC	-	13.25	13.25	139.02	78.43	13.25	13.25	**2.00**
PH-n	Acc	82.14	**86.94**	86.70	81.31	79.29	86.31	85.90	86.16
	RR	-	21.20	20.61	44.93	49.85	33.29	34.68	17.66
	PC	-	9.35	9.35	52.65	31.74	9.35	9.35	**0.71**
AVG	Acc	80.66	82.97	**83.01**	76.02	72.01	82.95	82.92	82.36
	RR	-	19.36	18.87	41.98	47.38	28.29	29.25	12.75
	PC	-	29.89	29.89	1,078.65	471.30	29.89	29.89	**5.25**

EHC: Experimental study



Methods	ACC		PC		Overall	
	w/l	Wilc.	w/l	Wilc.	w/l	Wilc.
EHC vs ENN (k=3)	4/6	0.126	9/1	**0.013**	4/6	0.333
EHC vs ENN (k=5)	4/6	0.169	9/1	**0.013**	4/6	0.333
EHC vs Multiedit (n=3, R=2)	8/2	**0.017**	10/0	**0.005**	8/2	**0.013**
EHC vs Multiedit (n=5, R=2)	9/1	**0.009**	9/1	**0.013**	9/1	**0.007**
EHC vs All-k-NN (k=7)	4/6	0.386	9/1	**0.013**	4/6	0.646
EHC vs All-k-NN (k=9)	5/5	0.508	9/1	**0.013**	5/5	0.575

Simultaneous editing and data abstraction by finding homogeneous clusters

Editing and Reduction through Homogeneous Clusters (ERHC):

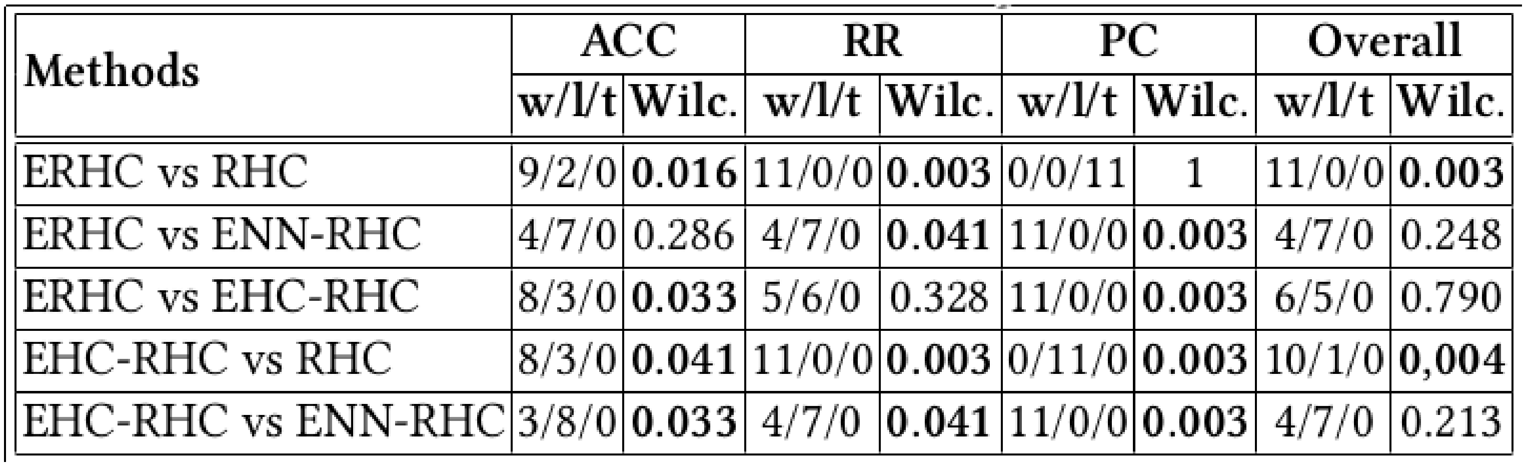
- Integration of EHC idea in RHC

- ERHC is a variation of RHC that can effectively handle datasets with noise (High reduction rates regardless the level of noise in the data)

- ERHC differs from RHC in one point: If an one-item cluster is identified, it is removed, i.e., ERHC does not build a prototype for this cluster

Dataset		Conv-1-NN	RHC	ENN-RHC	EHC-RHC	ERHC
LIR	Acc	95.825	**93.585**	92.720	93.045	92.690
	RR	-	88.081	90.343	90.383	**92.029**
	PC	-	**41.844**	159.039	73.710	**41.844**
PD	Acc	99.354	98.299	98.453	98.472	**98.626**
	RR	-	96.516	97.189	**97.589**	97.448
	PC	-	**2.882**	41.489	5.521	**2.882**
SH	Acc	99.822	98.095	**99.597**	98.481	98.038
	RR	-	99.550	99.658	99.669	**99.690**
	PC		**16.827**	1098.864	32.695	**16.827**
TXR	Acc	99.018	97.036	97.109	96.873	**97.364**
	RR	-	94.705	95.582	95.732	**95.936**
	PC	-	**3.629**	12.675	6.133	**3.629**
BN	Acc	86.906	83.283	**88.094**	87.019	88.000
	RR	-	79.684	**95.660**	93.000	90.330
	PC	-	**0.562**	9.519	1.014	**0.562**
LS	Acc	90.598	88.951	**89.138**	88.392	89.013
	RR	-	89.841	**95.062**	92.273	92.949
	PC	-	**1.693**	14.984	3.192	**1.693**
MGT	Acc	78.144	71.966	**77.781**	74.716	77.014
	RR	-	73.757	**93.057**	83.843	84.456
	PC	-	**4.082**	118.591	7.480	**4.082**
PH	Acc	90.100	85.585	85.400	86.158	**86.565**
	RR	-	80.708	**92.098**	89.008	88.053
	PC	-	**0.658**	9.812	1.161	**0.658**
PM	Acc	68.358	63.281	**72.653**	69.927	69.793
	RR	-	63.577	**91.792**	80.977	80.065
	PC	-	**0.062**	0.219	0.103	**0.062**
LS-n	Acc	82.580	78.819	**88.578**	84.817	85.377
	RR	-	76.632	**95.361**	88.465	87.560
	PC	-	**1.999**	14.744	3.637	**1.999**
PH-n	Acc	82.143	75.407	83.993	81.255	**84.030**
	RR	-	64.246	**92.019**	86.394	81.910
	PC	-	**0.706**	116.164	1.180	**0.706**
Avg	Acc	88.441	84.937	**88.501**	87.196	87.865
	RR	-	82.482	**94.347**	90.667	90.039
	PC	-	**6.813**	145.100	12.348	**6.813**

ERHC: Experimental study



Methods	ACC		RR		PC		Overall	
	w/l/t	Wilc.	w/l/t	Wilc.	w/l/t	Wilc.	w/l/t	Wilc.
ERHC vs RHC	9/2/0	**0.016**	11/0/0	**0.003**	0/0/11	1	11/0/0	**0.003**
ERHC vs ENN-RHC	4/7/0	0.286	4/7/0	**0.041**	11/0/0	**0.003**	4/7/0	0.248
ERHC vs EHC-RHC	8/3/0	**0.033**	5/6/0	0.328	11/0/0	**0.003**	6/5/0	0.790
EHC-RHC vs RHC	8/3/0	**0.041**	11/0/0	**0.003**	0/11/0	**0.003**	10/1/0	**0,004**
EHC-RHC vs ENN-RHC	3/8/0	**0.033**	4/7/0	**0.041**	11/0/0	**0.003**	4/7/0	0.213

The AIB2 algorithm (1/6)

- IB2 is an one-pass and incremental variation of the condensing CNN-rule
- We improve the performance of IB2 by considering the idea of prototype abstraction
- Our contribution is the development of an abstraction version of IB2 (AIB2) and an experimental study
- AIB2 is faster and achieves higher reduction rates than CNN-rule and IB2. AIB2 achieves higher accuracy than IB2

The AIB2 algorithm (2/6)

IB2 is a fast one-pass version of CNN-rule

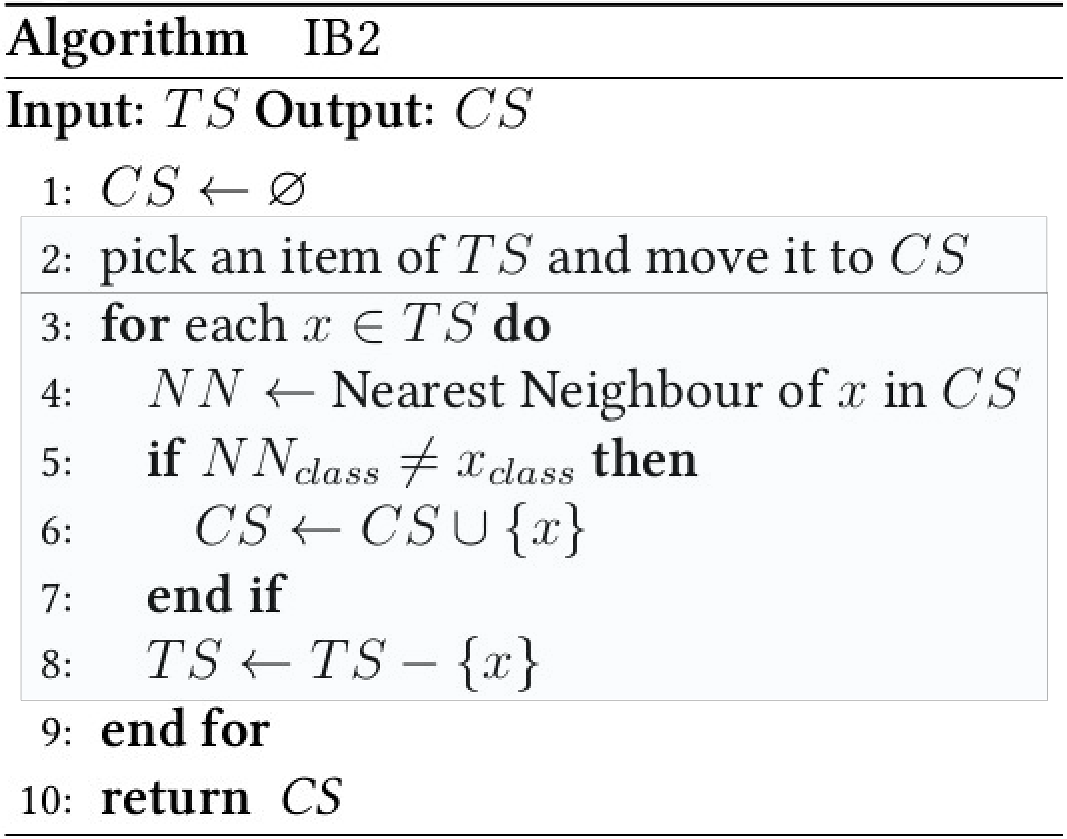
Like CNN-rule:

- IB2 is non-parametric
- IB2 is order dependent
- IB2 tries to keep only the close-border items

Contrary to CNN-rule:

- IB2 builds its condensing set **incrementally** (appropriate for dynamic/streaming environments)
- IB2 does not require that all training data reside into the main memory

The AIB2 algorithm (3/6)



Algorithm IB2

Input: TS **Output:** CS

```
$CS \leftarrow \varnothing$
pick an item of $TS$ and move it to $CS$
for each $x \in TS$ do
    $NN \leftarrow$ Nearest Neighbour of $x$ in $CS$
    if $NN_{class} \neq x_{class}$ then
        $CS \leftarrow CS \cup \{x\}$
    end if
    $TS \leftarrow TS - \{x\}$
end for
return $CS$
```

The AIB2 algorithm (4/6)

AIB2 idea: The prototypes should be at the center of the data area they represent

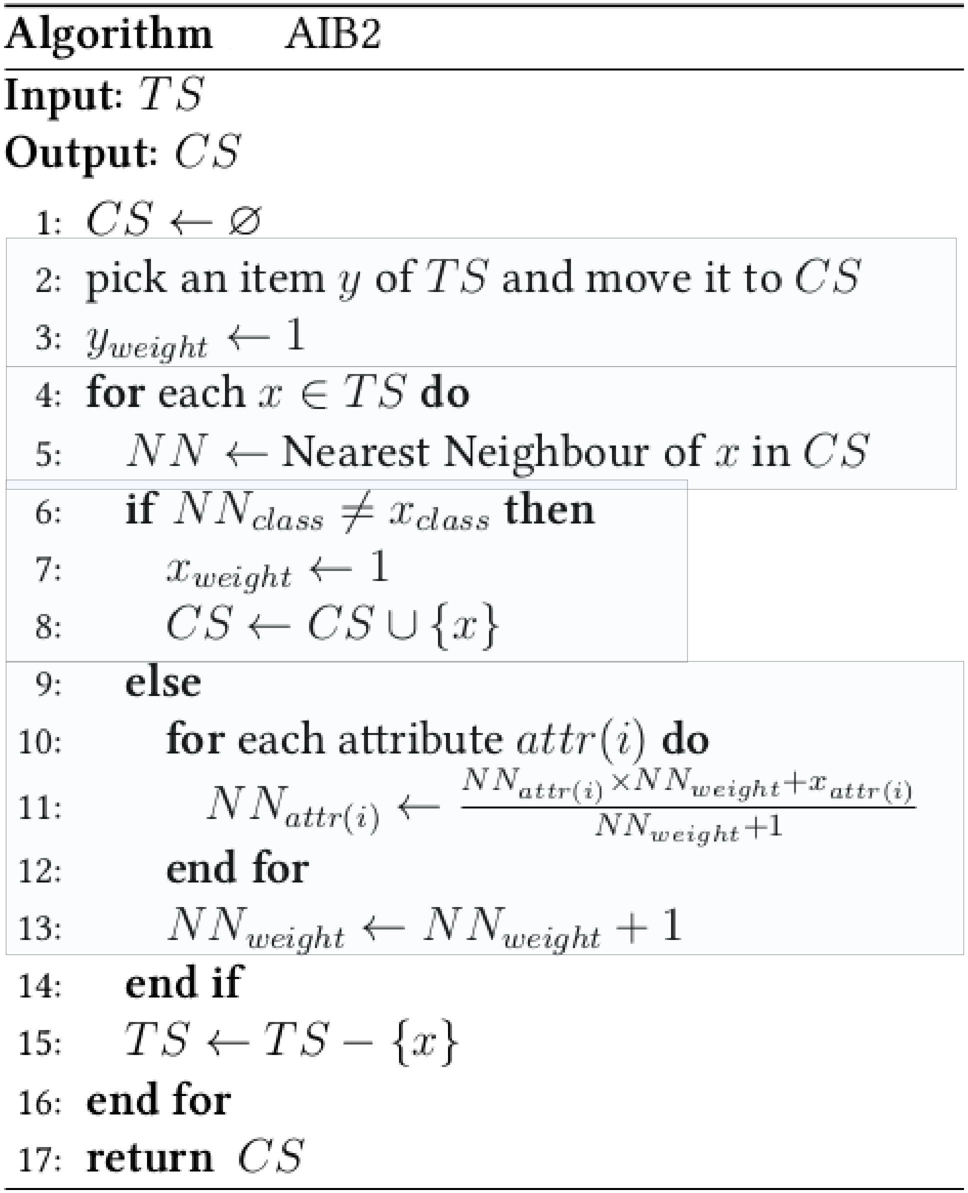
To achieve this:

- AIB2 adopts the concept of prototype weight which denotes the number of items it represents
- The weight values are used for updating the prototype in the multidimensional space

Result:

- Higher classification accuracy (Better prototypes)
- Higher reduction rates (Fewer items enter condensing set)
- Lower preprocessing cost (Fewer items enter condensing set)

The AIB2 algorithm (5/6)



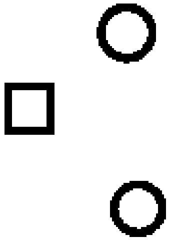
Algorithm AIB2

Input: TS

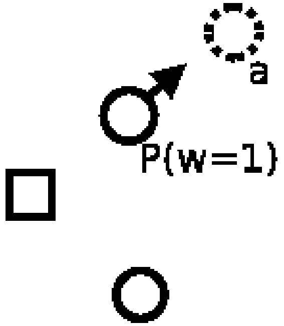
Output: CS

```
$CS \leftarrow \varnothing$
pick an item $y$ of $TS$ and move it to $CS$
$y_{weight} \leftarrow 1$
for each $x \in TS$ do
    $NN \leftarrow$ Nearest Neighbour of $x$ in $CS$
    if $NN_{class} \neq x_{class}$ then
        $x_{weight} \leftarrow 1$
        $CS \leftarrow CS \cup \{x\}$
    else
        for each attribute $attr(i)$ do
            $NN_{attr(i)} \leftarrow \frac{NN_{attr(i)} \times NN_{weight} + x_{attr(i)}}{NN_{weight} + 1}$
        end for
        $NN_{weight} \leftarrow NN_{weight} + 1$
    end if
    $TS \leftarrow TS - \{x\}$
end for
return $CS$
```

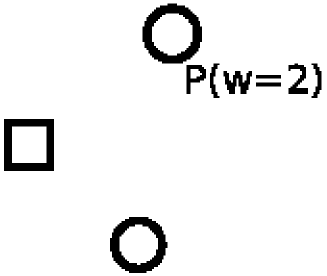
The AIB2 algorithm (6/6)



Current condensing set



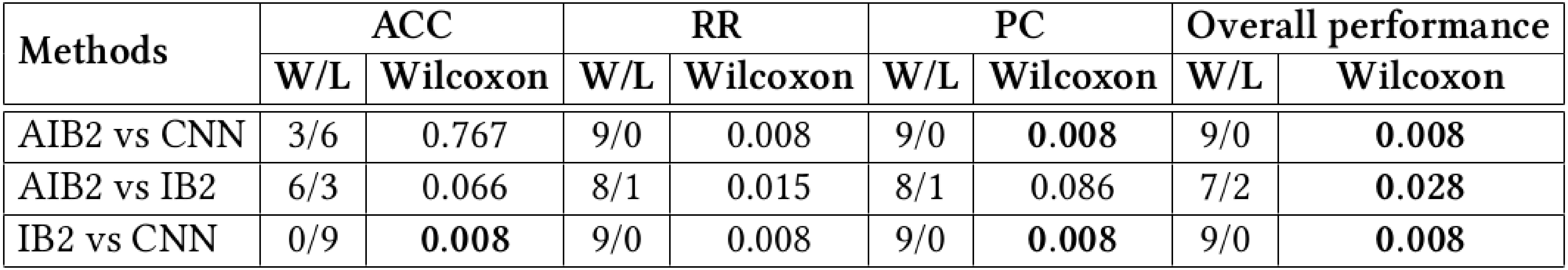
Arrival of a new item



Repositioning of the nearest prototype

Dataset		Conv-1-NN	CNN-rule	IB2	AIB2
LIR	Acc:	95.83	92.84	91.98	**94.12**
	RR:	-	83,54	85.66	**88.12**
	PC:	-	163.03	23.37	**20.10**
MGT	Acc:	78.14	**74.54**	71.97	73.36
	RR:	-	60.08	70.60	**71.90**
	PC:	-	281.49	34.61	**33.05**
MGT-ENN	Acc:	80.44	**79.26**	78.01	78.81
	RR:	-	87.62	90.07	**91.06**
	PC:	-	68.61	8.48	**7.65**
PD	Acc:	99.35	**98.68**	98.04	98.33
	RR:	-	95.36	96.23	**97.19**
	PC:	-	11.75	1.78	**1.38**
LS	Acc:	90.60	88.21	86.87	**89.42**
	RR:	-	80.22	84.62	**86.72**
	PC:	-	17.99	2.22	**1.92**
SH	Acc:	99.82	**99.76**	99.73	99.72
	RR:	-	99.37	99.44	**99.46**
	PC:	-	45.30	8.26	**7.89**
TXR	Acc:	99.02	97.16	96.35	**97.69**
	RR:	-	91.90	93.33	**94.95**
	PC:	-	5.65	0.84	**0.66**
PH	Acc:	90.10	**87.82**	85.57	84.92
	RR:	-	76.04	80.85	**81.75**
	PC:	-	13.45	1.96	**1.84**
KDD	Acc:	99.71	**99.66**	99.48	99.41
	RR:	-	99.12	**99.26**	99.21
	PC:	-	384.90	**55.58**	58.78
Average	Acc:	92.56	**90.88**	89.78	90.64
	RR:	-	85,92	88.90	**90.04**
	PC:	-	110.24	15.23	**14.81**

AIB2: Experimental study



Methods	ACC		RR		PC		Overall performance	
	W/L	**Wilcoxon**	**W/L**	**Wilcoxon**	**W/L**	**Wilcoxon**	**W/L**	**Wilcoxon**
AIB2 vs CNN	3/6	0.767	9/0	0.008	9/0	**0.008**	9/0	**0.008**
AIB2 vs IB2	6/3	0.066	8/1	0.015	8/1	0.086	7/2	**0.028**
IB2 vs CNN	0/9	**0.008**	9/0	0.008	9/0	**0.008**	9/0	**0.008**

General purpose DRTs for efficient time series classification (1/3)

DRTs has been recently exploited for fast time series classification (Both are parametric):

1. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: Insight: efficient and effective instance selection for time-series classification. 15th Pacific-Asia conference on Advances in knowledge discovery and data mining – Part II. pp. 149–160. PAKDD'11, Springer (2011)
2. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. 23rd international conference on Machine learning. pp. 1033–1040. ICML '06, ACM (2006)

Motivation:

- State-of-the-art non-parametric DRTs have not been evaluated on time series data
- The idea of Prototype Abstraction has not been adopted for fast time series classification
- RHC and AIB2 have not been evaluated on time series data

General purpose DRTs for efficient time series classification (2/3)

State-of-the-art non-parametric DRTs are evaluated on time series data:

- Original time series data (using all attributes)
- A reduced dimensionality representation of the same time series data (12 dimensions, using Piecewise Aggregate Approximation - PAA)

DRTs evaluated:

- **Two condensing algorithms:**
  Condensing Nearest Neighbor (CNN) rule
  The IB2 algorithm
- **Three prototype abstraction algorithms:**
  Reduction by Space Partitioning v3 (RSP3)
  Reduction through Homogeneous Clusters (RHC)
  The AIB2 algorithm

General purpose DRTs for efficient time series classification (3/3)

RSP3 achieved the highest accuracy. However, it is the slowest method in terms of both preprocessing and classification cost

RHC, AIB2 and IB2 have much lower preprocessing cost than CNN-rule and RSP3

RHC and AIB2 build the smallest condensing sets

RHC and AIB2 are usually more accurate than IB2 and CNN-rule

The 1-NN classification on the 12-dimensional datasets is very fast while accuracy remains at high levels

Conclusion: One can obtain efficient time series classifiers by combining condensing or prototype abstraction algorithms with time-series dimensionality representations

Data Reduction through *k*-Means clustering

The thesis proposes the use of the means generated by *k*-means clustering as a simple noise-tolerant approach (R*k*M algorithm)

For each class, R*k*M builds a number of clusters and their means are placed into CS as prototypes of the class

The noisy items of a class are represented by a mean item lying in the main area of the class. So R*k*M is a more noise-tolleran DRT

Examination of how the performance of two state-of-the-art DRTs (CNN-rule and RSP3) are affected by the addition of noise

Prototype Selection by Clustering (PSC)

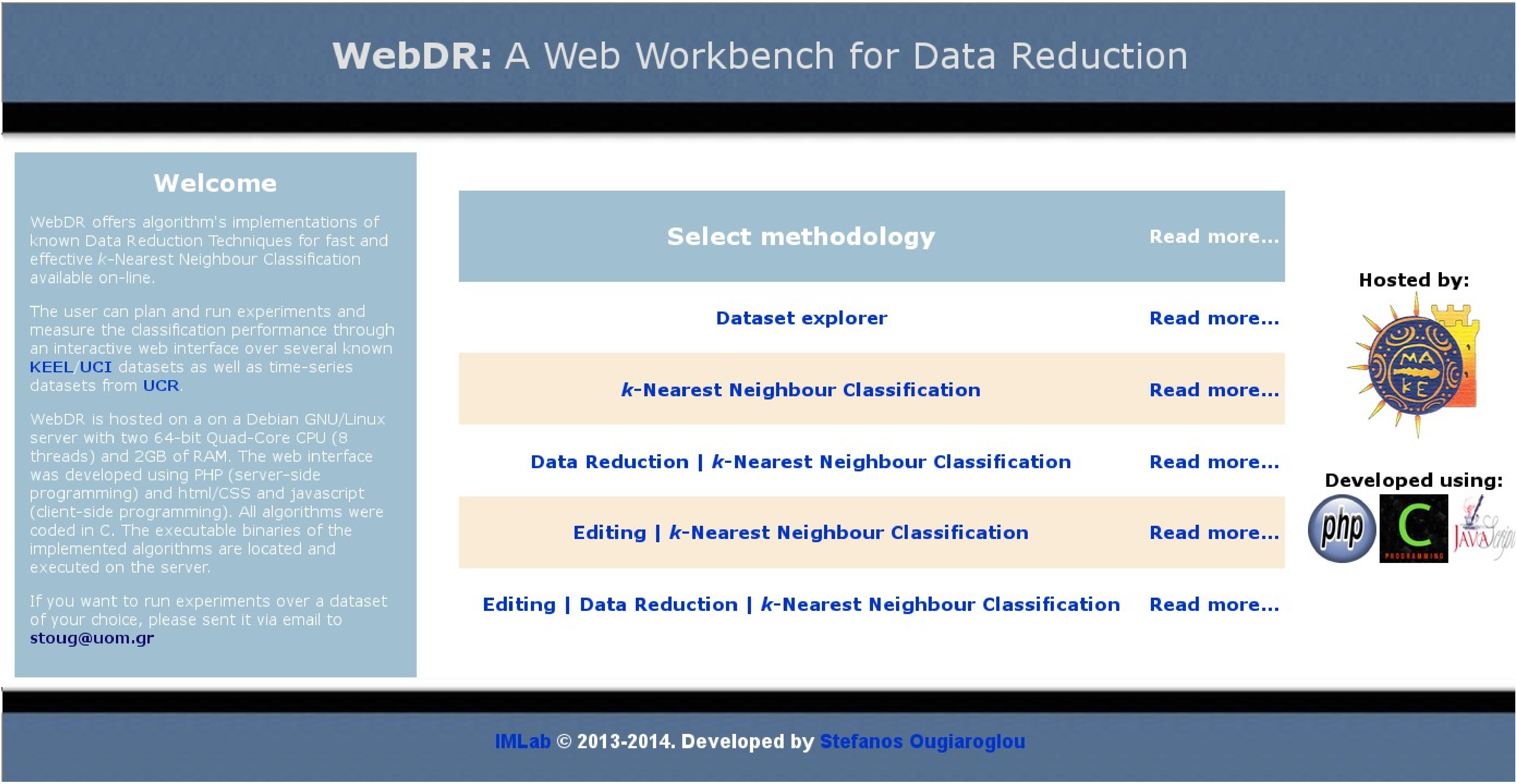
PSC is a recently proposed condensing algorithm whose main goal is the fast execution of data reduction rather than high reduction rates

PSC is parametric. The user should provide the number of clusters that will be built. The main goal of PSC is achieved by using a small number of clusters

The thesis demonstrates that the reduction rate and the classification accuracy achieved by PSC can be improved by generating a large number of clusters

WebDR: A Web Workbench for Data Reduction

(https://ilust.uom.gr/webdr)



WebDR: A Web Workbench for Data Reduction

Welcome

WebDR offers algorithm's implementations of known Data Reduction Techniques for fast and effective *k*-Nearest Neighbour Classification available on-line.

The user can plan and run experiments and measure the classification performance through an interactive web interface over several known KEEL/UCI datasets as well as time-series datasets from UCR.

WebDR is hosted on a on a Debian GNU/Linux server with two 64-bit Quad-Core CPU (8 threads) and 2GB of RAM. The web interface was developed using PHP (server-side programming) and html/CSS and javascript (client-side programming). All algorithms were coded in C. The executable binaries of the implemented algorithms are located and executed on the server.

If you want to run experiments over a dataset of your choice, please sent it via email to **stoug@uom.gr**

Select methodology	**Read more...**
Dataset explorer	**Read more...**
***k*-Nearest Neighbour Classification**	**Read more...**
Data Reduction | *k*-Nearest Neighbour Classification	**Read more...**
Editing | *k*-Nearest Neighbour Classification	**Read more...**
Editing | Data Reduction | *k*-Nearest Neighbour Classification	**Read more...**

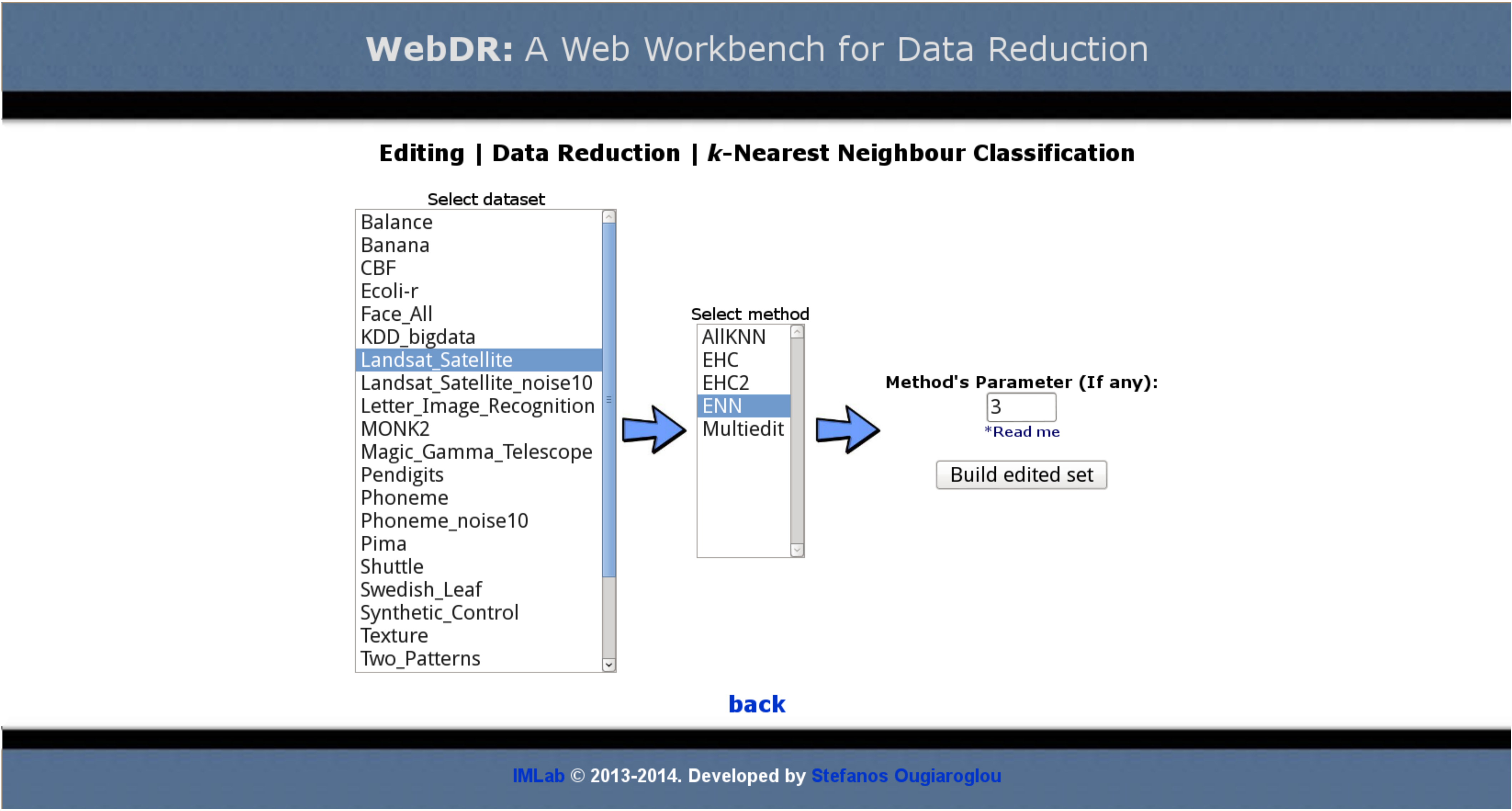
Hosted by:

Developed using:

IMLab © 2013-2014. Developed by Stefanos Ougiaroglou

WebDR: A Web Workbench for Data Reduction

(https://ilust.uom.gr/webdr)



WebDR: A Web Workbench for Data Reduction

Editing | Data Reduction | *k*-Nearest Neighbour Classification

Select dataset

- Balance
- Banana
- CBF
- Ecoli-r
- Face_All
- KDD_bigdata
- Landsat_Satellite
- Landsat_Satellite_noise10
- Letter_Image_Recognition
- MONK2
- Magic_Gamma_Telescope
- Pendigits
- Phoneme
- Phoneme_noise10
- Pima
- Shuttle
- Swedish_Leaf
- Synthetic_Control
- Texture
- Two_Patterns

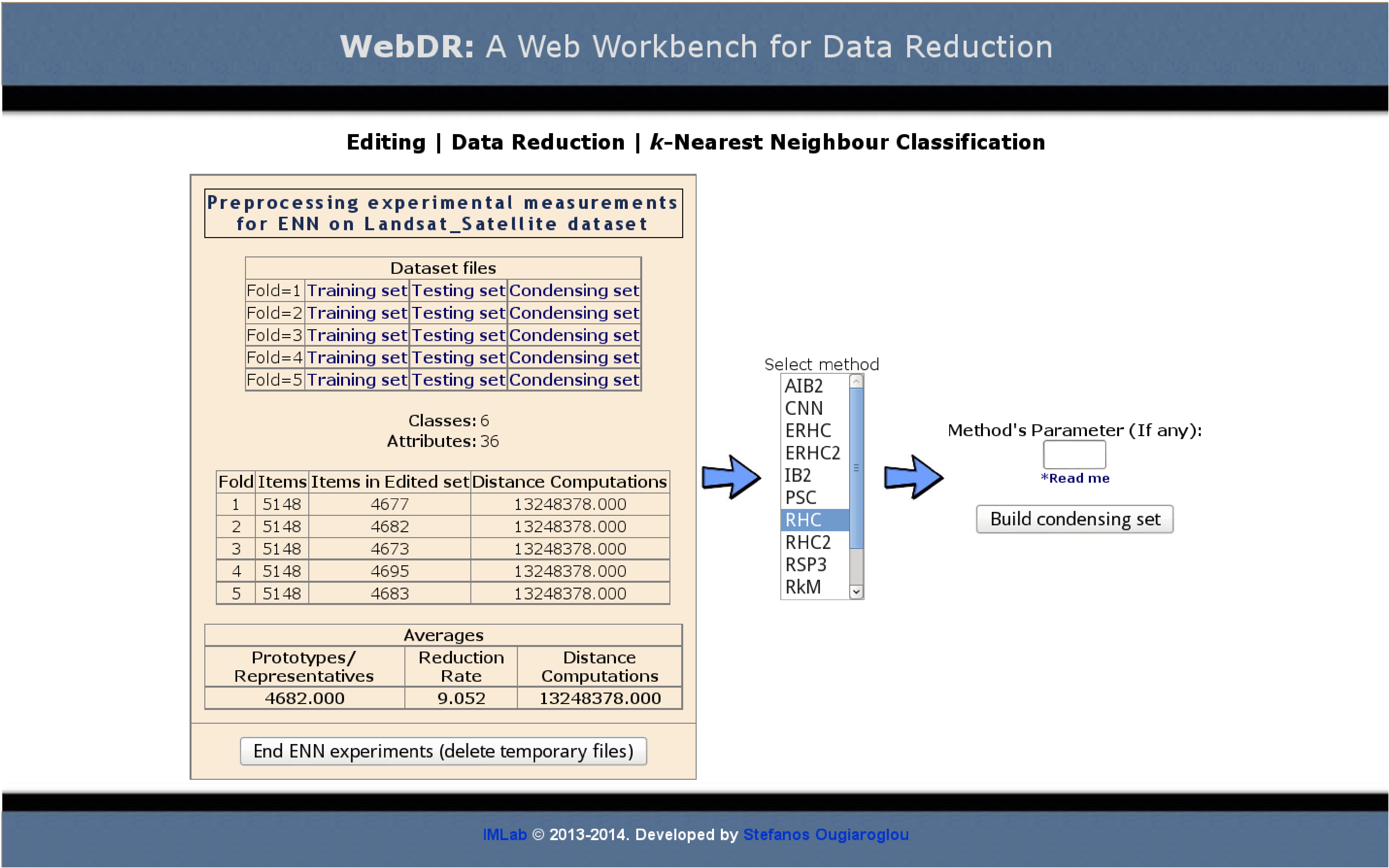
Select method

- AllKNN
- EHC
- EHC2
- ENN
- Multiedit

Method's Parameter (If any): 3

IMLab © 2013-2014. Developed by Stefanos Ougiaroglou

WebDR: A Web Workbench for Data Reduction
(https://ilust.uom.gr/webdr)



WebDR: A Web Workbench for Data Reduction

Editing | Data Reduction | *k*-Nearest Neighbour Classification

Preprocessing experimental measurements for ENN on Landsat_Satellite dataset

Dataset files			
Fold=1	Training set	Testing set	Condensing set
Fold=2	Training set	Testing set	Condensing set
Fold=3	Training set	Testing set	Condensing set
Fold=4	Training set	Testing set	Condensing set
Fold=5	Training set	Testing set	Condensing set

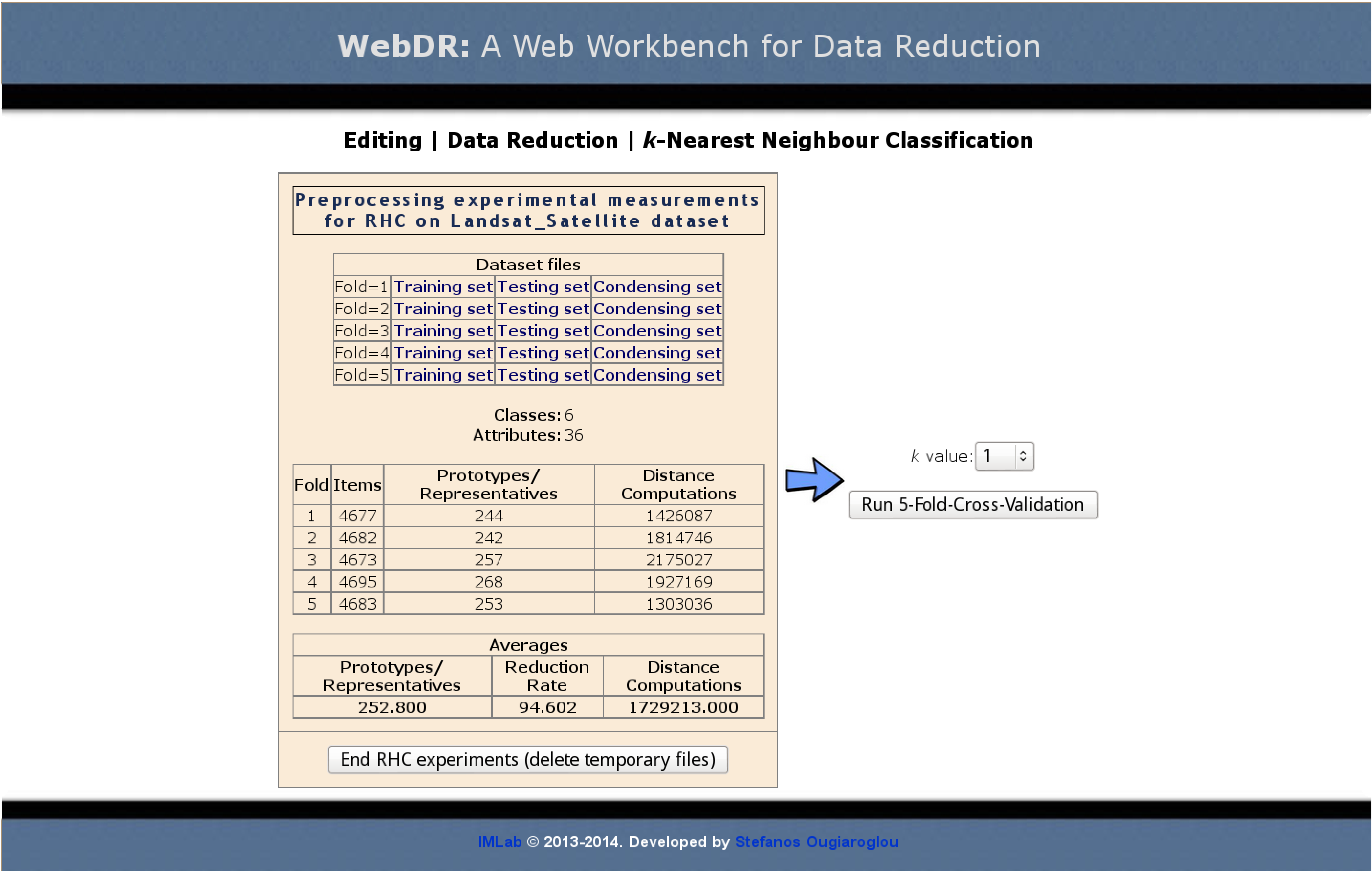
Classes: 6
Attributes: 36

Fold	Items	Items in Edited set	Distance Computations
1	5148	4677	13248378.000
2	5148	4682	13248378.000
3	5148	4673	13248378.000
4	5148	4695	13248378.000
5	5148	4683	13248378.000

Averages		
Prototypes/ Representatives	Reduction Rate	Distance Computations
4682.000	9.052	13248378.000

IMLab © 2013-2014. Developed by Stefanos Ougiaroglou

WebDR: A Web Workbench for Data Reduction
(https://ilust.uom.gr/webdr)



WebDR: A Web Workbench for Data Reduction

Editing | Data Reduction | *k*-Nearest Neighbour Classification

Preprocessing experimental measurements for RHC on Landsat_Satellite dataset

Dataset files			
Fold=1	Training set	Testing set	Condensing set
Fold=2	Training set	Testing set	Condensing set
Fold=3	Training set	Testing set	Condensing set
Fold=4	Training set	Testing set	Condensing set
Fold=5	Training set	Testing set	Condensing set

Classes: 6
Attributes: 36

Fold	Items	Prototypes/ Representatives	Distance Computations
1	4677	244	1426087
2	4682	242	1814746
3	4673	257	2175027
4	4695	268	1927169
5	4683	253	1303036

Averages		
Prototypes/ Representatives	Reduction Rate	Distance Computations
252.800	94.602	1729213.000

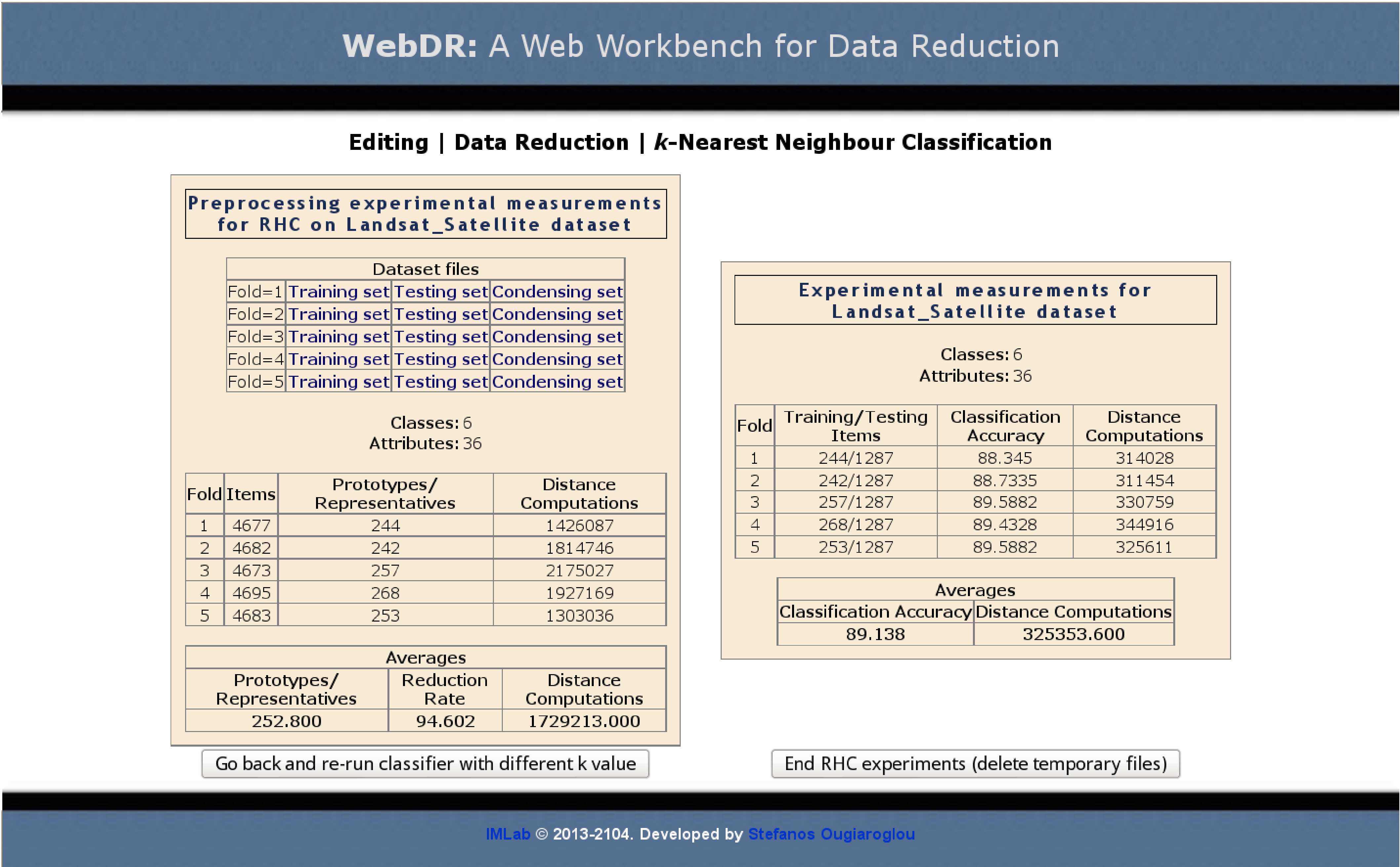
k value: 1

End RHC experiments (delete temporary files)
Run 5-Fold-Cross-Validation

IMLab © 2013-2014. Developed by Stefanos Ougiaroglou

WebDR: A Web Workbench for Data Reduction

(https://ilust.uom.gr/webdr)



WebDR: A Web Workbench for Data Reduction

Editing | Data Reduction | *k*-Nearest Neighbour Classification

Preprocessing experimental measurements for RHC on Landsat_Satellite dataset

Dataset files			
Fold=1	Training set	Testing set	Condensing set
Fold=2	Training set	Testing set	Condensing set
Fold=3	Training set	Testing set	Condensing set
Fold=4	Training set	Testing set	Condensing set
Fold=5	Training set	Testing set	Condensing set

Classes: 6
Attributes: 36

Fold	Items	Prototypes/ Representatives	Distance Computations
1	4677	244	1426087
2	4682	242	1814746
3	4673	257	2175027
4	4695	268	1927169
5	4683	253	1303036

Averages		
Prototypes/ Representatives	Reduction Rate	Distance Computations
252.800	94.602	1729213.000

Go back and re-run classifier with different k value

Experimental measurements for Landsat_Satellite dataset

Classes: 6
Attributes: 36

Fold	Training/Testing Items	Classification Accuracy	Distance Computations
1	244/1287	88.345	314028
2	242/1287	88.7335	311454
3	257/1287	89.5882	330759
4	268/1287	89.4328	344916
5	253/1287	89.5882	325611

Averages	
Classification Accuracy	Distance Computations
89.138	325353.600

End RHC experiments (delete temporary files)

IMLab © 2013-2104. Developed by Stefanos Ougiaroglou

C. Contribution: Hybrid Speed-up methods

Fast Hybrid classification based on Minimum distance and the *k*-NN classifiers (1/7)

Motivation

- Fast classification without costly preprocessing (without using DRTs or Indexes)

Contribution:

- We purpose a Fast, Hybrid and Model-free classification algorithm (FHCA) and two variations that combine the MDC and the conventional *k*-NN classifier

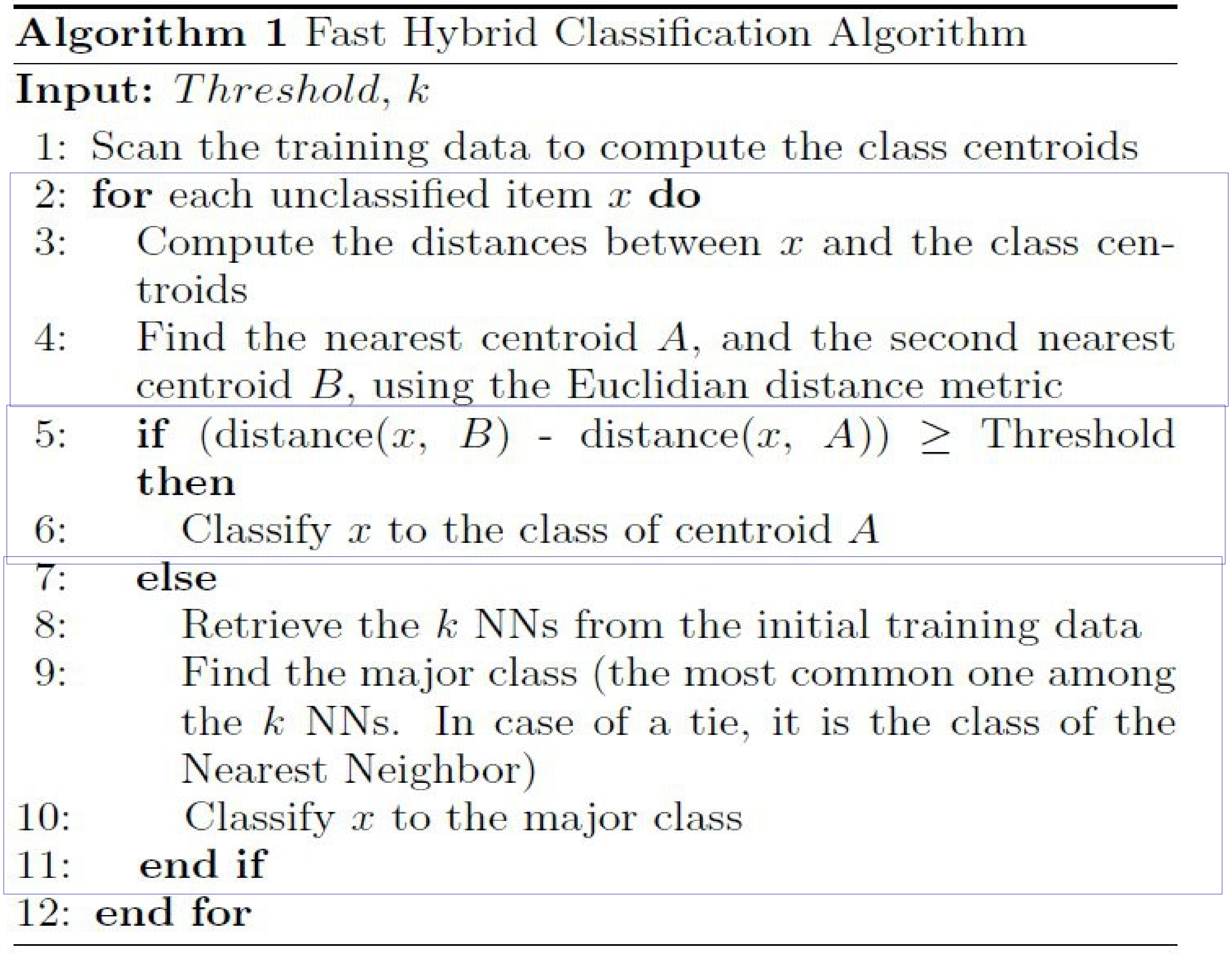
- It avoids expensive preprocessing procedures and so, It can be applied for repeated classification tasks in dynamic databases

Fast Hybrid classification based on Minimum distance and the *k*-NN classifiers (2/7)

Basic idea:

- FHCA search for the nearest neighbors in a small dataset which includes only a representative for each class
- Then, it tries to classify the new item to the class of a representative
- Upon failure to meet the set acceptance criteria, classification proceeds by the conventional *k*-NN classifier
- Each representative is computed by calculating the average value of the items that belong to each one class
- The main algorithm (FHCA) and the two variations (FHCA-V1 & FHCA-V2) differ to each other on the set acceptance criteria that they involve

Fast Hybrid classification based on Minimum distance and the *k*-NN classifiers (3/7)



Algorithm 1 Fast Hybrid Classification Algorithm

Input: *Threshold*, k

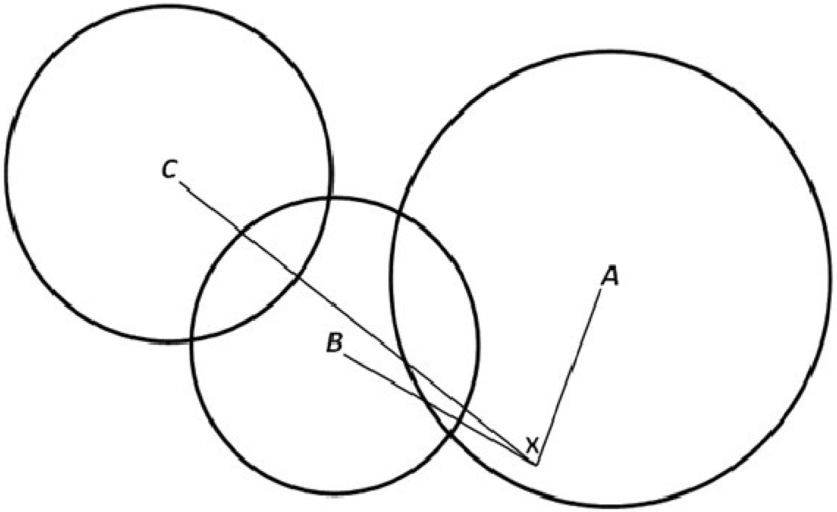
```
Scan the training data to compute the class centroids
for each unclassified item $x$ do
    Compute the distances between $x$ and the class centroids
    Find the nearest centroid $A$, and the second nearest centroid $B$, using the Euclidian distance metric
    if (distance($x$, $B$) - distance($x$, $A$)) $\geq$ Threshold then
        Classify $x$ to the class of centroid $A$
    else
        Retrieve the $k$ NNs from the initial training data
        Find the major class (the most common one among the $k$ NNs. In case of a tie, it is the class of the Nearest Neighbor)
        Classify $x$ to the major class
    end if
end for
```

Fast Hybrid classification based on Minimum distance and the *k*-NN classifiers (4/7)

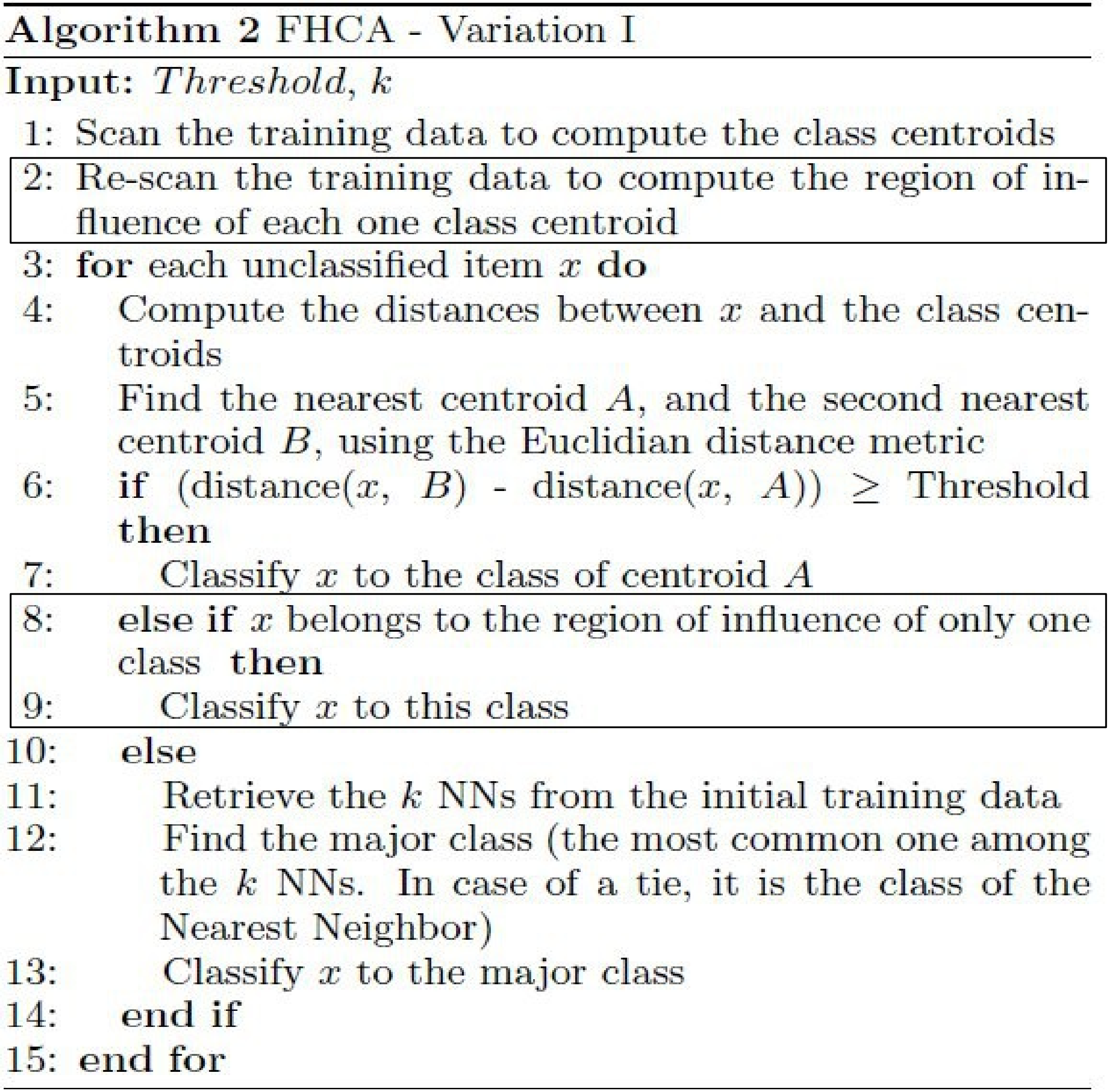
FHCA – Variation I

- FHCA-V1 attempts to classify even more new incoming items without falling back to the *k*-NN classifier
- It computes the region of influence of each one class
- The class region of influence is the average distance of the training set class items from the class centroid
- It uses the distance difference criterion and if it fails, it uses the Region of Influence Criterion (RIC)

RIC: If *x* lies within the region of influence of one class, xis classified to this class



Fast Hybrid classification based on Minimum distance and the *k*-NN classifiers (5/7)



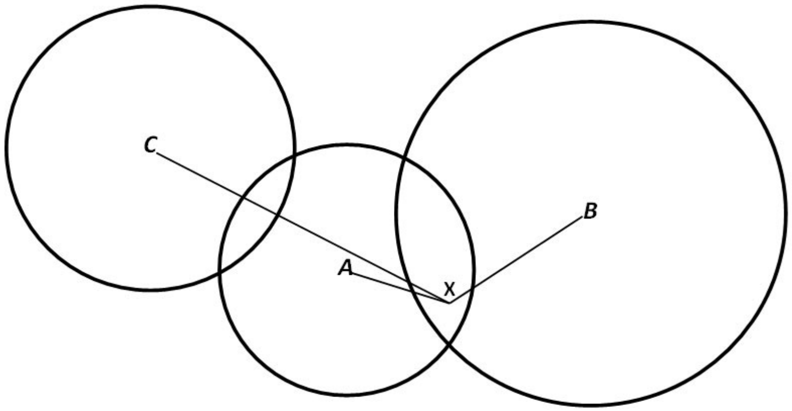
Algorithm 2 FHCA - Variation I

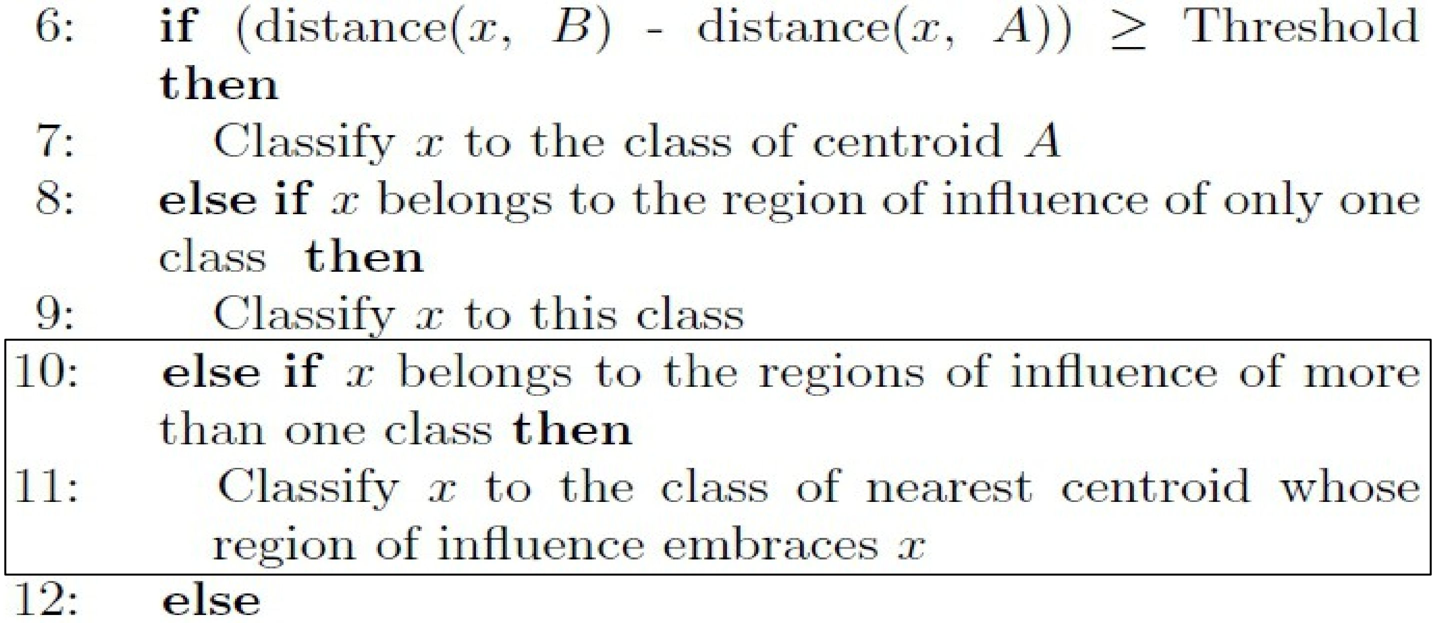
Input: *Threshold*, k

```
Scan the training data to compute the class centroids
Re-scan the training data to compute the region of influence of each one class centroid
for each unclassified item $x$ do
    Compute the distances between $x$ and the class centroids
    Find the nearest centroid $A$, and the second nearest centroid $B$, using the Euclidian distance metric
    if (distance($x$, $B$) - distance($x$, $A$)) $\geq$ Threshold then
        Classify $x$ to the class of centroid $A$
    else if $x$ belongs to the region of influence of only one class then
        Classify $x$ to this class
    else
        Retrieve the $k$ NNs from the initial training data
        Find the major class (the most common one among the $k$ NNs. In case of a tie, it is the class of the Nearest Neighbor)
        Classify $x$ to the major class
    end if
end for
```

Fast Hybrid classification based on Minimum distance and the *k*-NN classifiers (6/7)

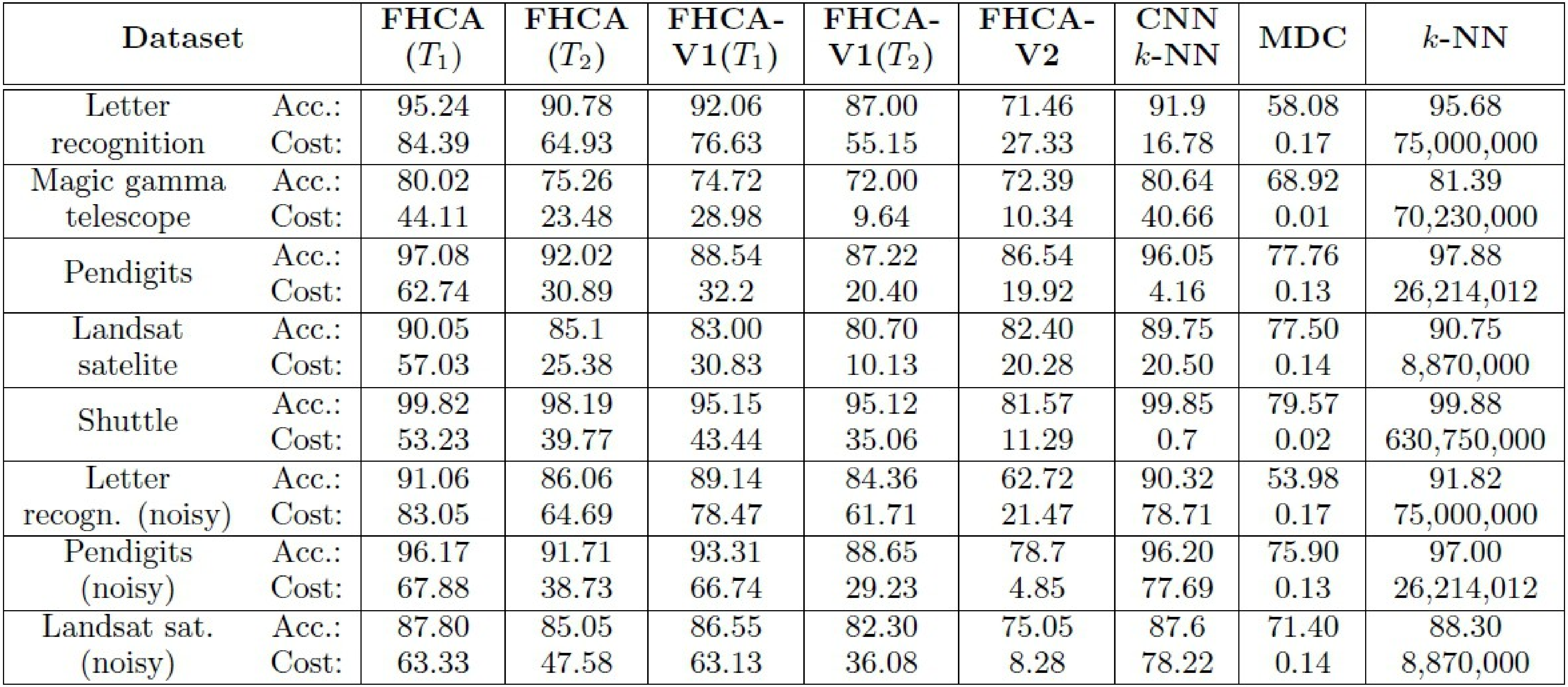
FHCA – Variation II





```
if (distance($x$, $B$) - distance($x$, $A$)) $\geq$ Threshold then
    Classify $x$ to the class of centroid $A$
else if $x$ belongs to the region of influence of only one class then
    Classify $x$ to this class
else if $x$ belongs to the regions of influence of more than one class then
    Classify $x$ to the class of nearest centroid whose region of influence embraces $x$
else
```

Fast Hybrid classification based on Minimum distance and the *k*-NN classifiers (7/7)



Dataset		FHCA (T_1)	FHCA (T_2)	FHCA-V1(T_1)	FHCA-V1(T_2)	FHCA-V2	CNN k-NN	MDC	k-NN
Letter recognition	Acc.:	95.24	90.78	92.06	87.00	71.46	91.9	58.08	95.68
	Cost:	84.39	64.93	76.63	55.15	27.33	16.78	0.17	75,000,000
Magic gamma telescope	Acc.:	80.02	75.26	74.72	72.00	72.39	80.64	68.92	81.39
	Cost:	44.11	23.48	28.98	9.64	10.34	40.66	0.01	70,230,000
Pendigits	Acc.:	97.08	92.02	88.54	87.22	86.54	96.05	77.76	97.88
	Cost:	62.74	30.89	32.2	20.40	19.92	4.16	0.13	26,214,012
Landsat satelite	Acc.:	90.05	85.1	83.00	80.70	82.40	89.75	77.50	90.75
	Cost:	57.03	25.38	30.83	10.13	20.28	20.50	0.14	8,870,000
Shuttle	Acc.:	99.82	98.19	95.15	95.12	81.57	99.85	79.57	99.88
	Cost:	53.23	39.77	43.44	35.06	11.29	0.7	0.02	630,750,000
Letter recogn. (noisy)	Acc.:	91.06	86.06	89.14	84.36	62.72	90.32	53.98	91.82
	Cost:	83.05	64.69	78.47	61.71	21.47	78.71	0.17	75,000,000
Pendigits (noisy)	Acc.:	96.17	91.71	93.31	88.65	78.7	96.20	75.90	97.00
	Cost:	67.88	38.73	66.74	29.23	4.85	77.69	0.13	26,214,012
Landsat sat. (noisy)	Acc.:	87.80	85.05	86.55	82.30	75.05	87.6	71.40	88.30
	Cost:	63.33	47.58	63.13	36.08	8.28	78.22	0.14	8,870,000

FHC: An adaptive fast hybrid method (1/9)

Motivation:

- Does the combination of the strategies of data abstraction and CBMs lead to fast and accurate classification?

The **contribution** is the development of an adaptive, hybrid and cluster-based method for speeding-up the k-NN classifier

- We develop a fast cluster-based preprocessing algorithm that builds a two level data structure. The first level stores a number of cluster means for each class. The second level stores the set of items belonging to each cluster
- We develop efficient classifiers that access either the first or the second level of the data structure and perform the classification

FHC: An adaptive fast hybrid method (2/9)

Two Level Data Structure Construction Algorithm (TLDSCA)

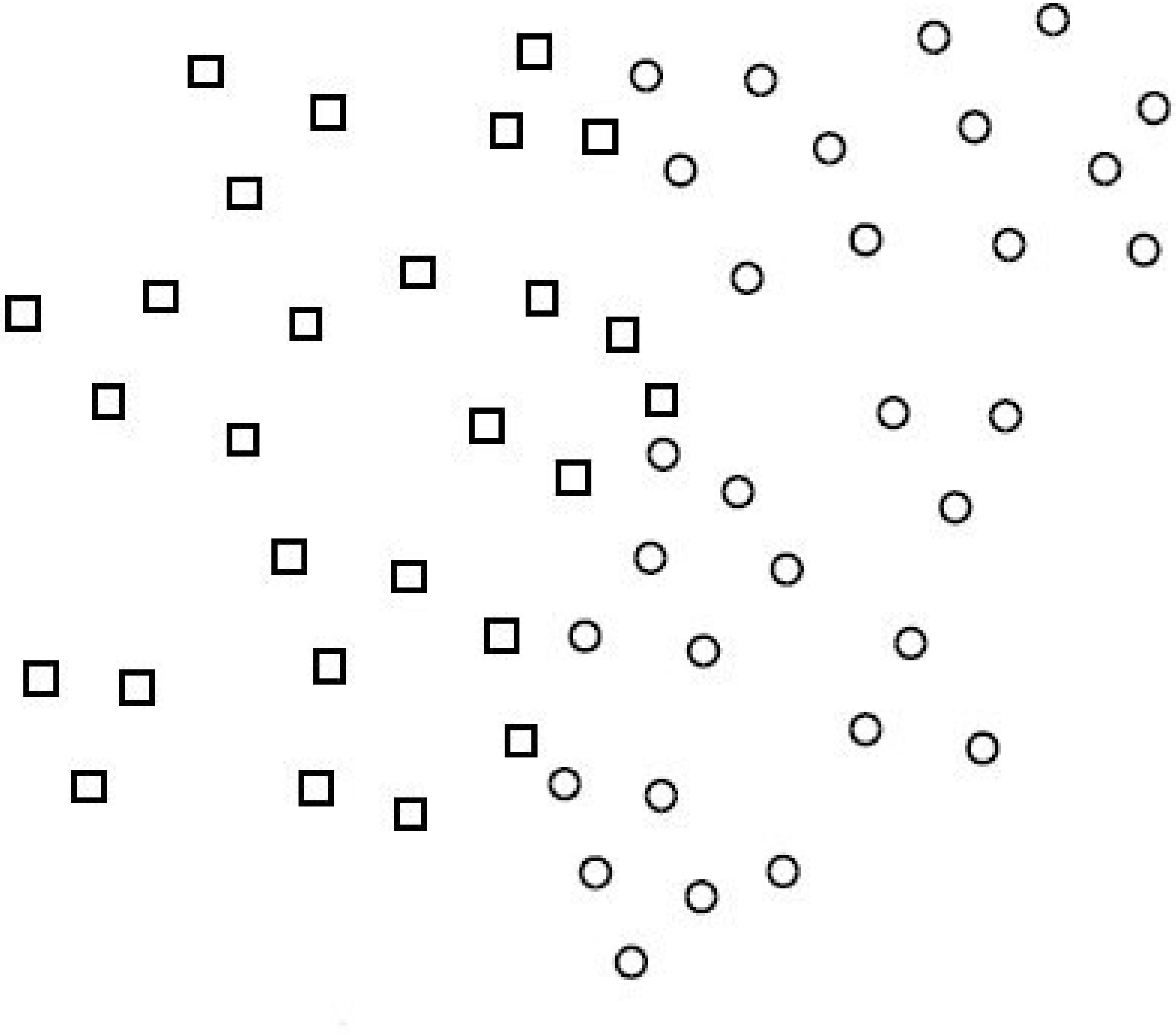
- For each class, it identifies a number of clusters
- First Level: A list of cluster centroids for all classes
- Second level: The "real" items of each cluster

Data Reduction Factor (DRF) determines the number of representatives (or the TLDS length). For each class *C*, the algorithm builds *NC* representatives

$$NC = \lceil \frac{X}{DRF} \rceil$$

X is the number of items that belong to class *C*

FHC: An adaptive fast hybrid method (3/9)



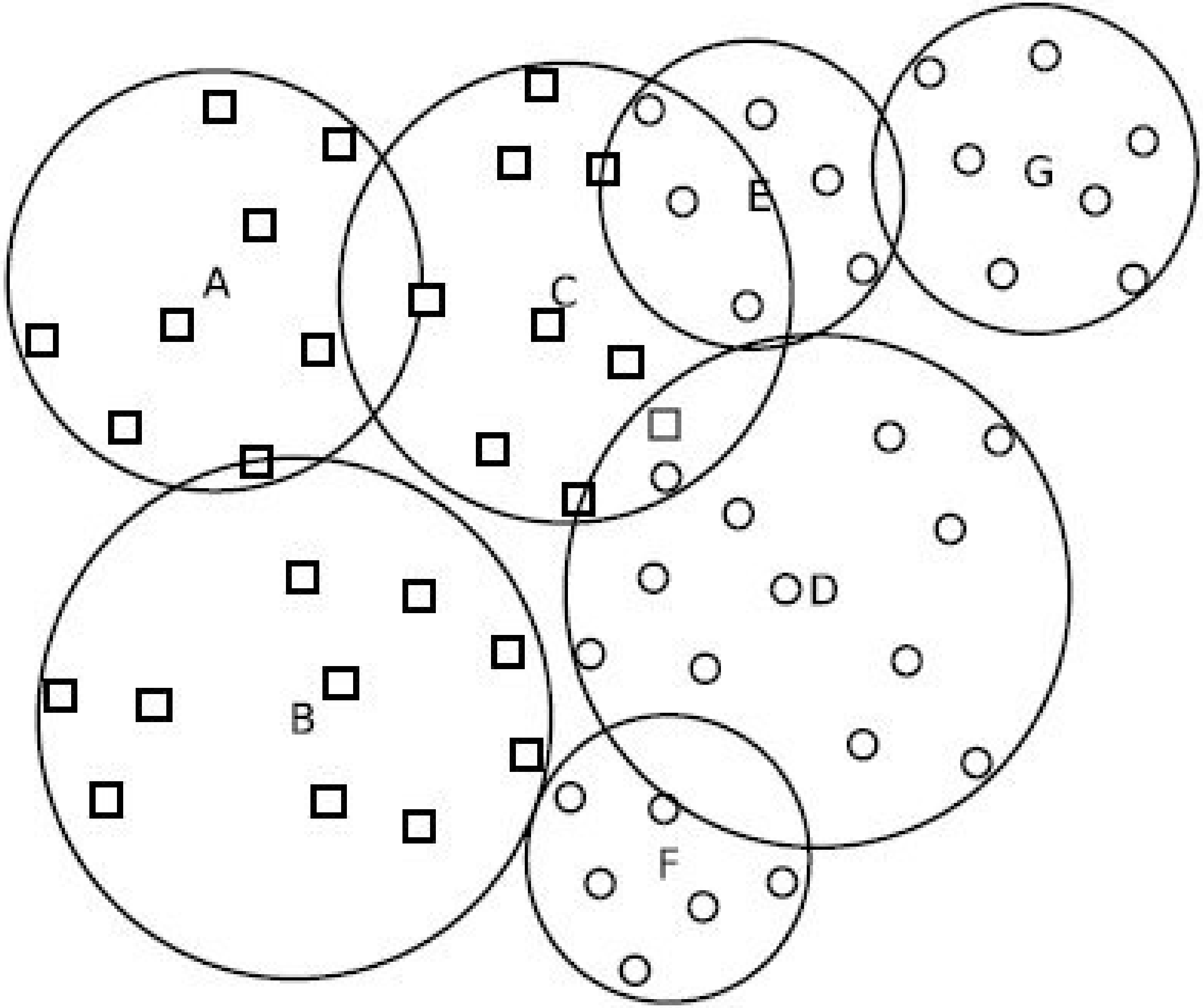
FHC: An adaptive fast hybrid method (4/9)

DRF = 10,
CIRCLE Items = 31, SQUARE Items = 27

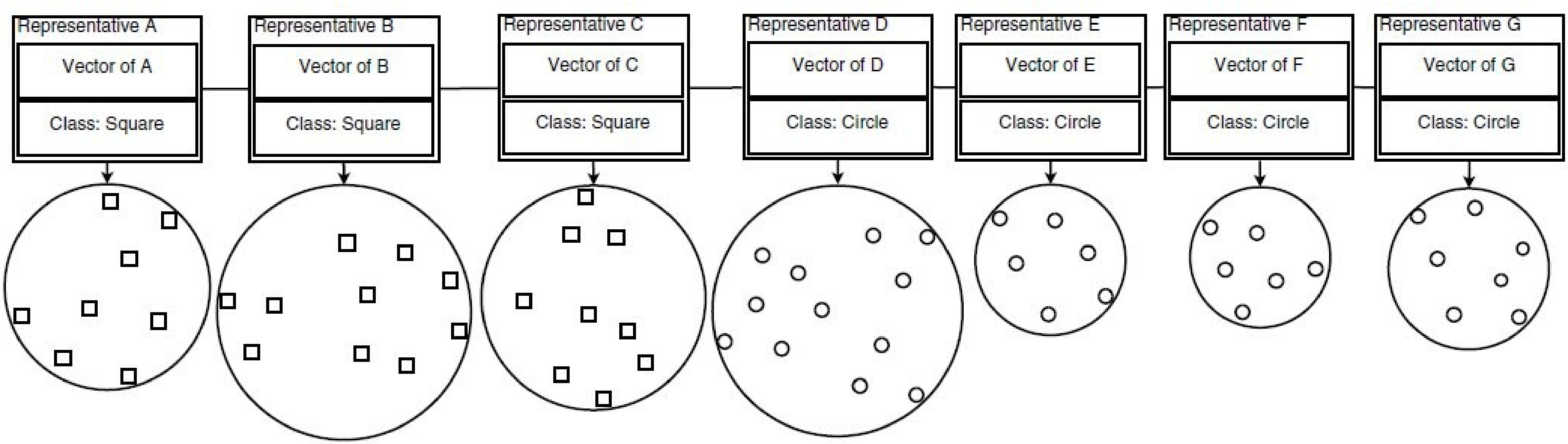
$$NC_{Circle} = \lceil \frac{31}{10} \rceil = 4$$

$$NC_{Square} = \lceil \frac{31}{10} \rceil = 3$$

FHC: An adaptive fast hybrid method (5/9)



FHC: An adaptive fast hybrid method (6/9)



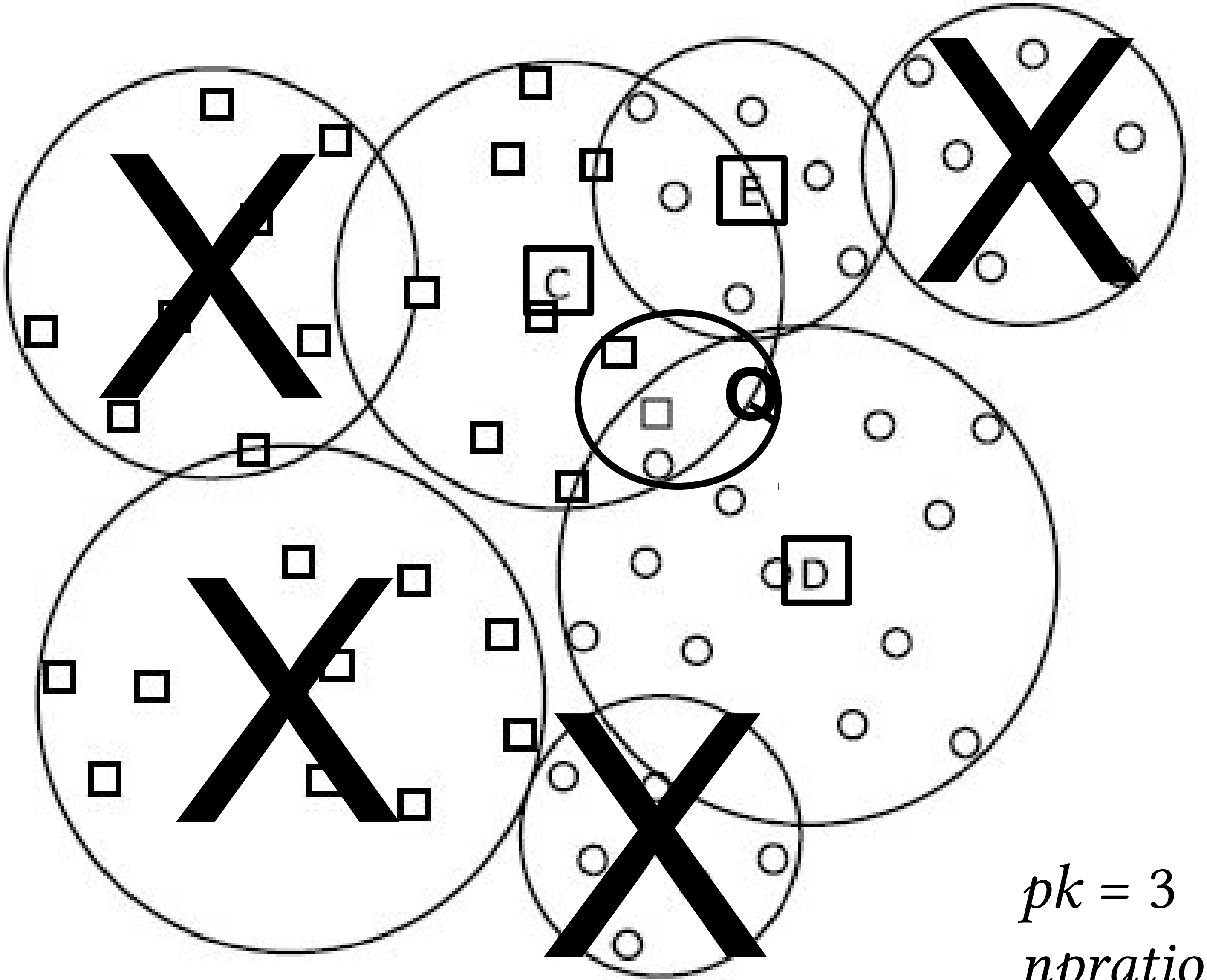
FHC: An adaptive fast hybrid method (7/9)

FHC-I:

- It accesses TLDS and make predictions
- For each new item *x*, it scans the first level of TLDS and retrieves the *pk* nearest representatives to *x*
- If the *npratio* parameter is satisfied, they determine the class where *x* belongs to
- Otherwise, *x* is classified by searching for the *k* "real" nearest neighbors within the clusters of the *pk* nearest representatives

The *pk* an*d npration* parameters let the user to define the desirable trade-off between accuracy and cost

FHC: An adaptive fast hybrid method (8/9)



pk = 3

npratio =1

FHC: An adaptive fast hybrid method (9/9)

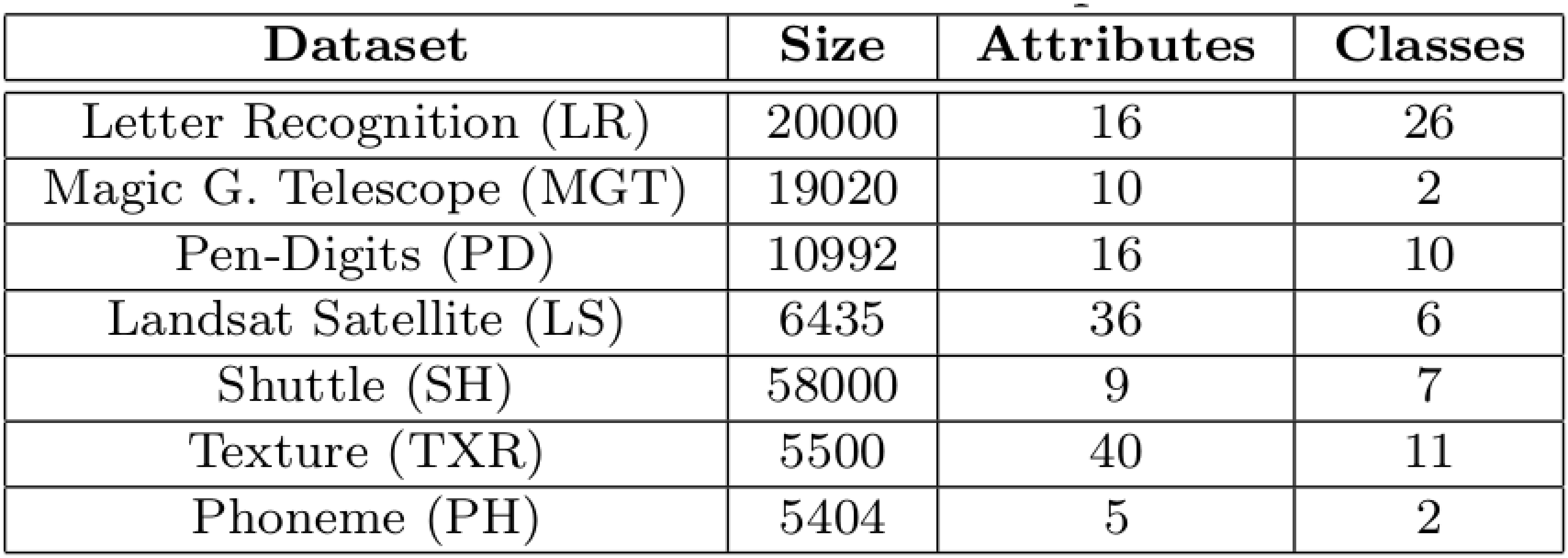
FHC-II:

- **Motivation:** in cases of non-uniform distributions, the probability of performing a second level search depends on which is the majority class of the first level search. Items belonging to rare classes are always classified by a second level search

- FHC-II attempts to better manage imbalanced datasets. It considers the sizes of the classes and tries to reduce “costly” second level searches.

- FHC-II estimates *npratio* instead of using a pre-specified value. The value of *npratio* is dynamically adjusted to be between a user-defined range and depends on the majority class determined by the first level search

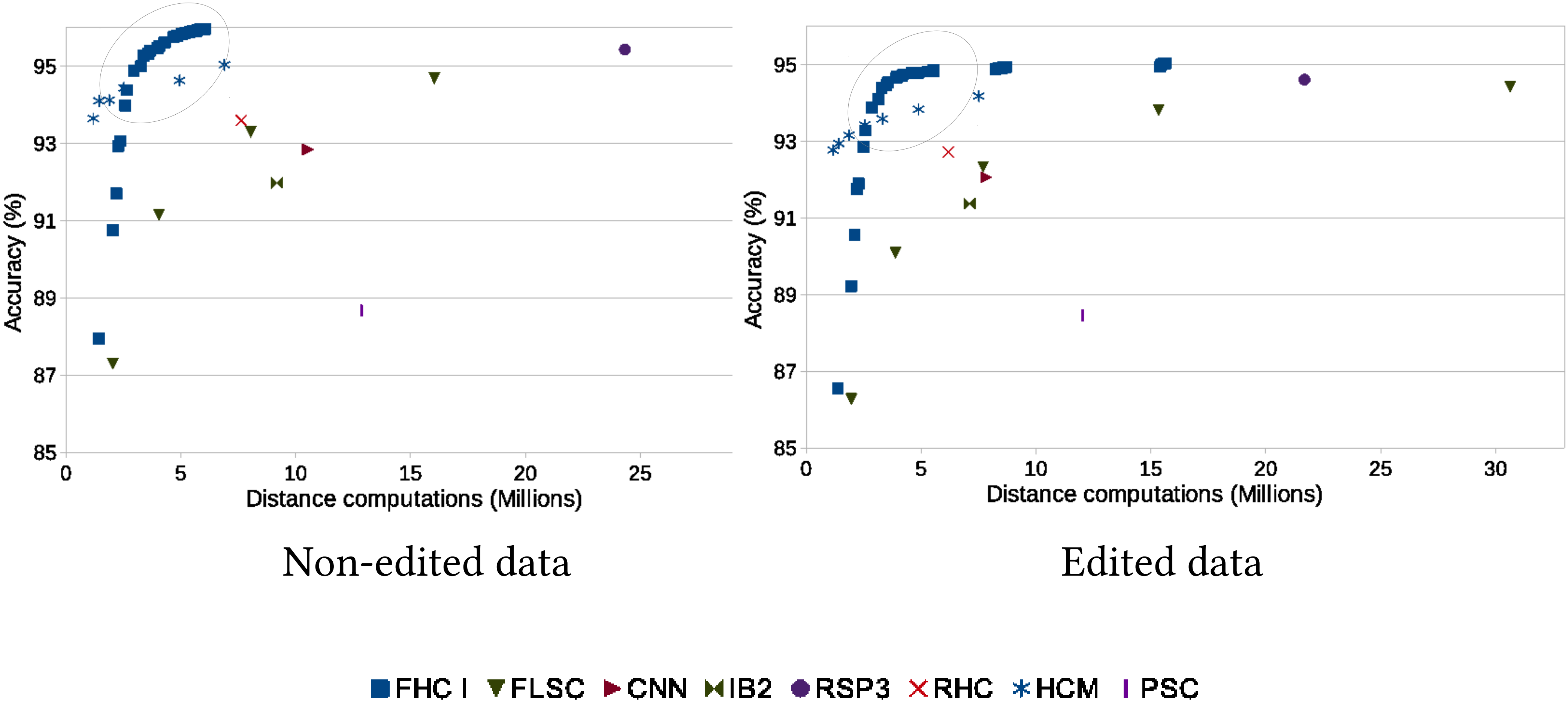
FHC: Experimental study (1/8)



Dataset	Size	Attributes	Classes
Letter Recognition (LR)	20000	16	26
Magic G. Telescope (MGT)	19020	10	2
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

FHC: Experimental study (2/8)

LIR dataset

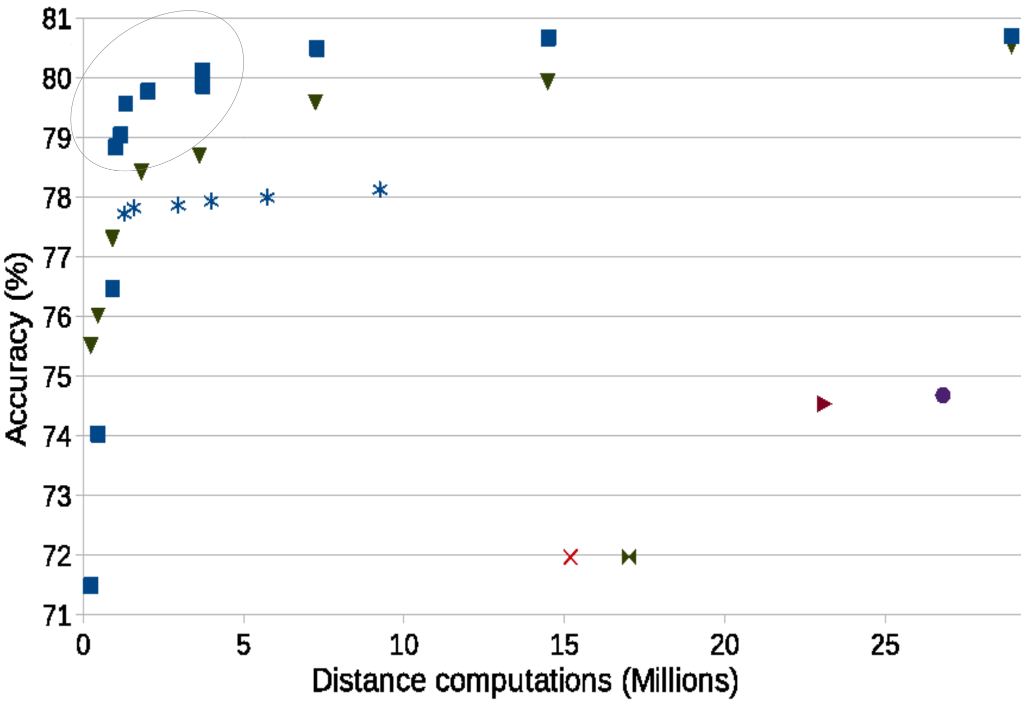


Non-edited data

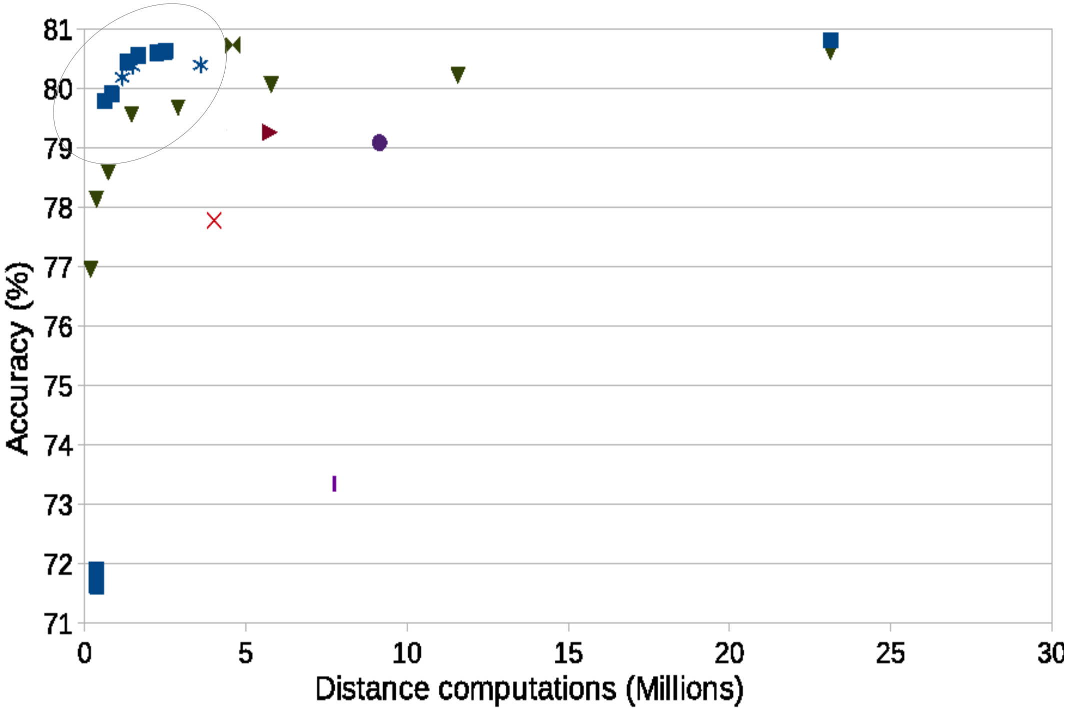
Edited data

FHC: Experimental study (3/8)

MGT dataset



Non-edited data

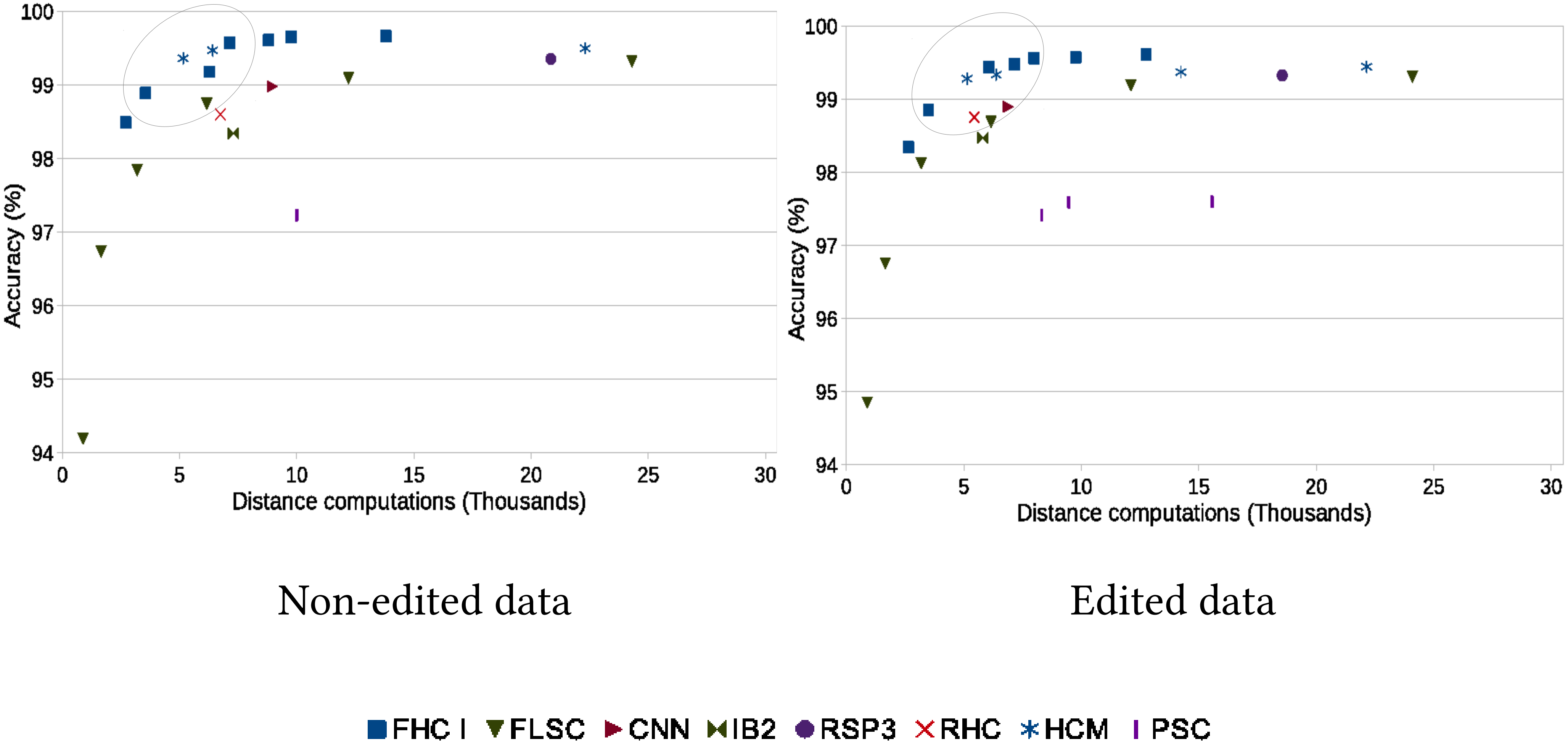


Edited data

■ FHC I ▼ FLSC ▶ CNN ⧓ IB2 ● RSP3 ✕ RHC ✱ HCM ❙ PSC

FHC: Experimental study (4/8)

PD dataset

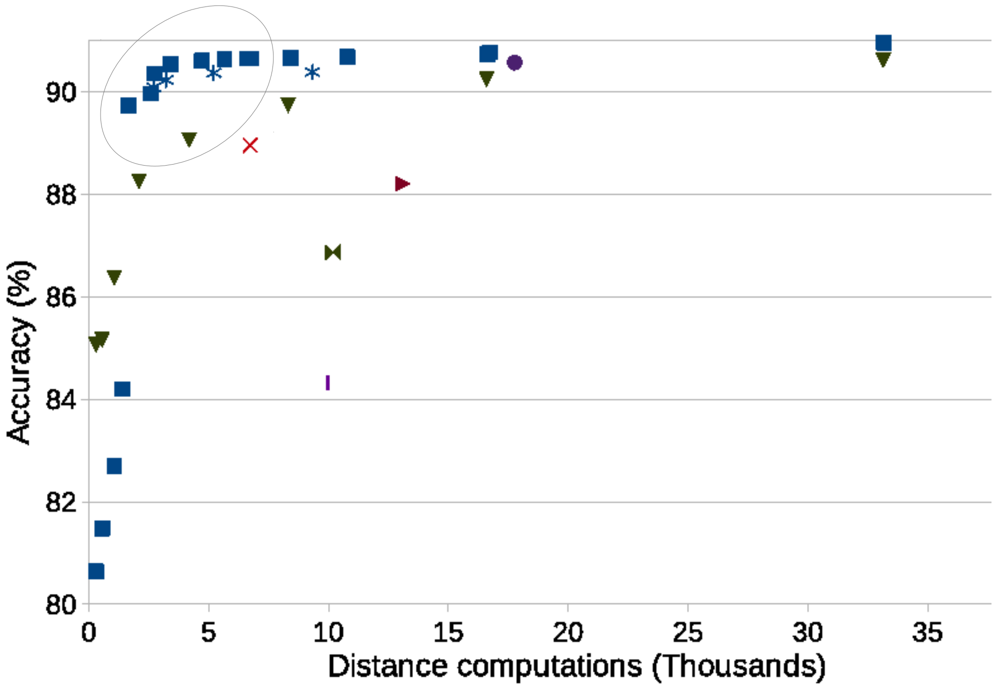


Non-edited data

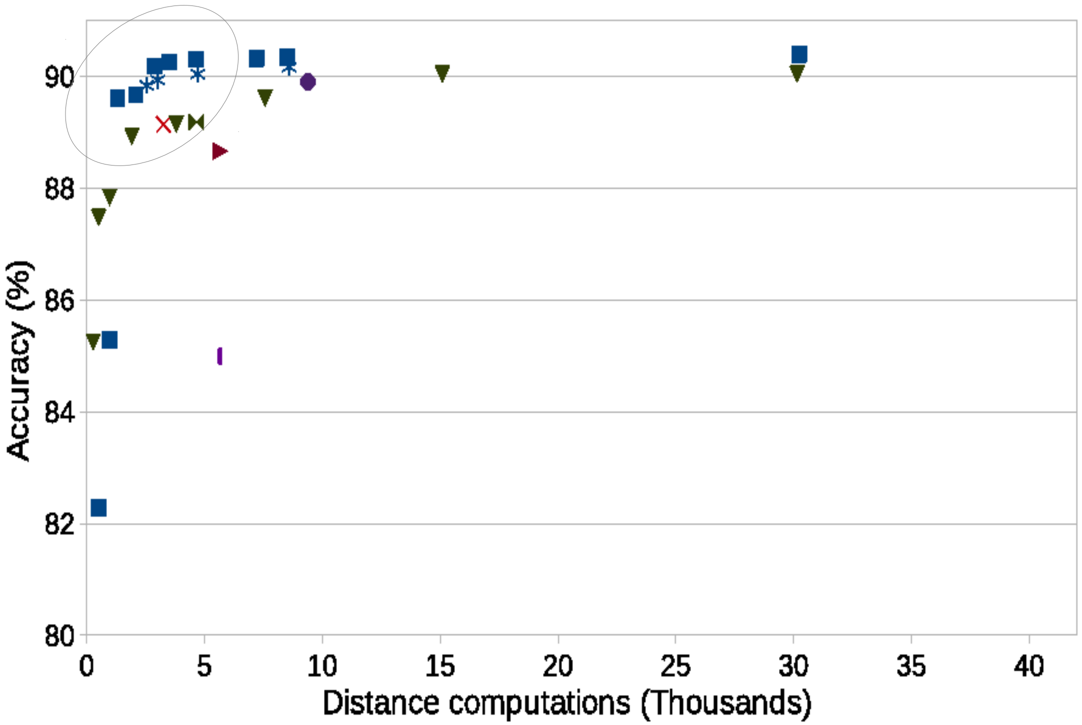
Edited data

FHC: Experimental study (5/8)

LS dataset



Non-edited data

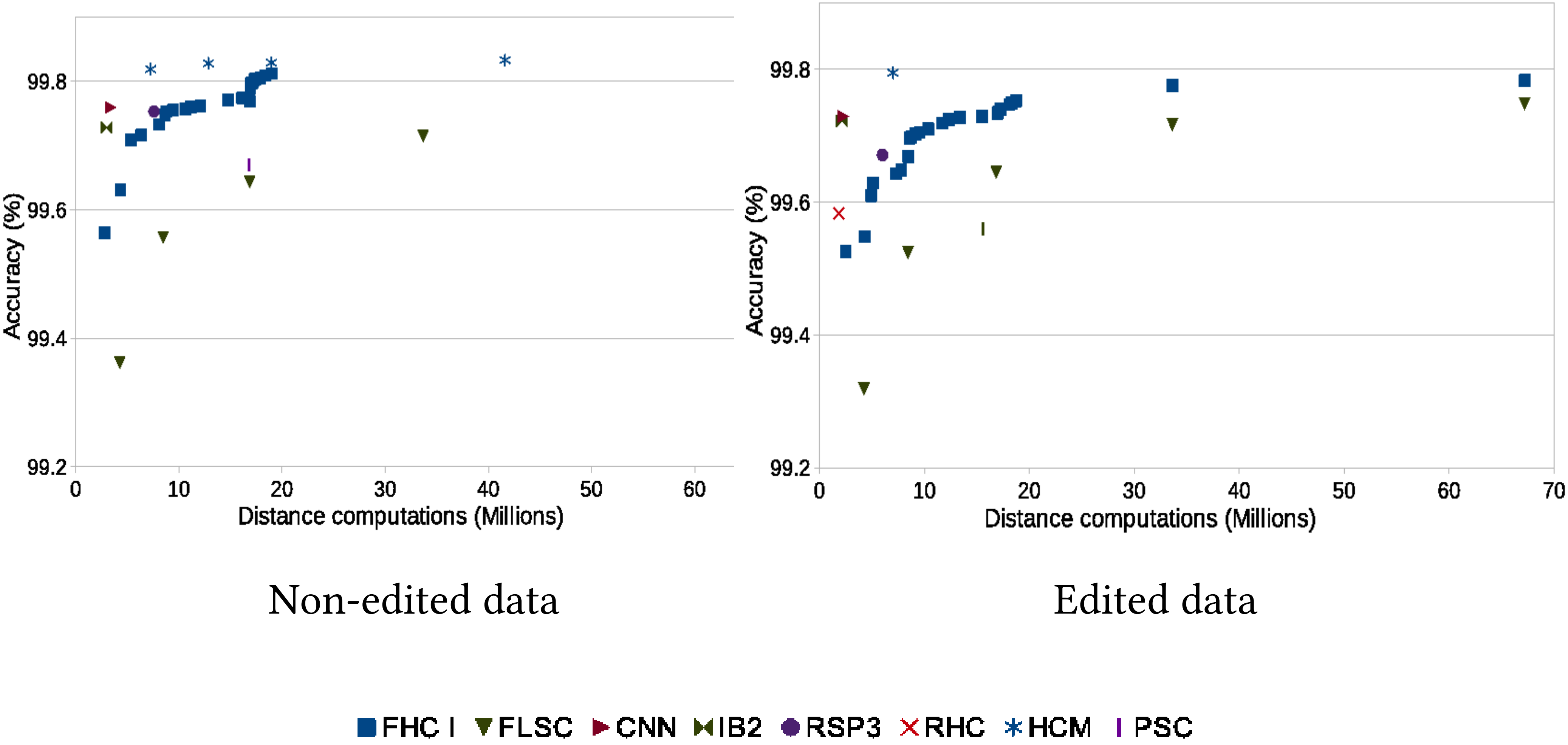


Edited data

■ FHC I ▼ FLSC ► CNN ⋈ IB2 ● RSP3 × RHC ✱ HCM | PSC

FHC: Experimental study (6/8)

SH dataset

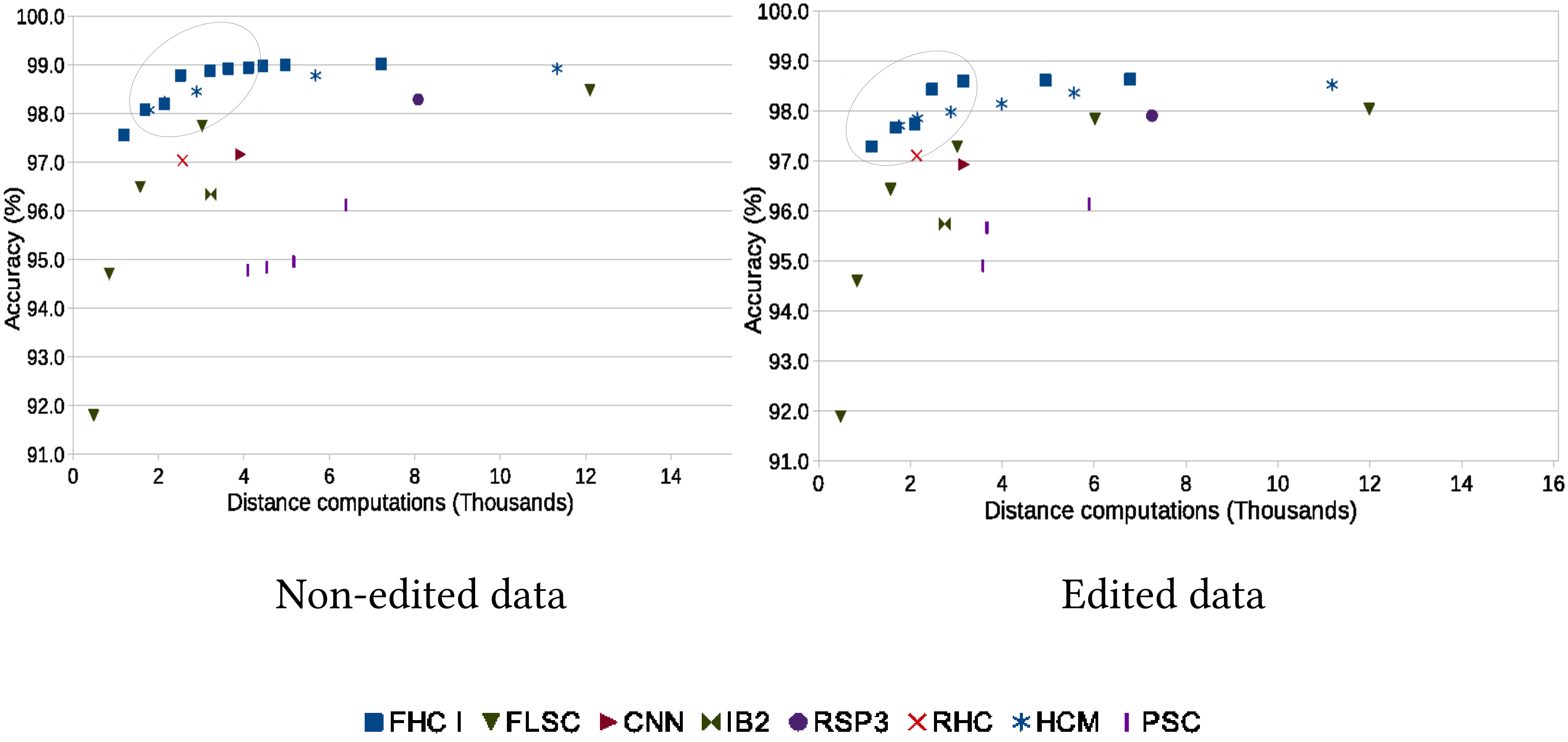


Non-edited data

Edited data

FHC: Experimental study (7/8)

TXR dataset

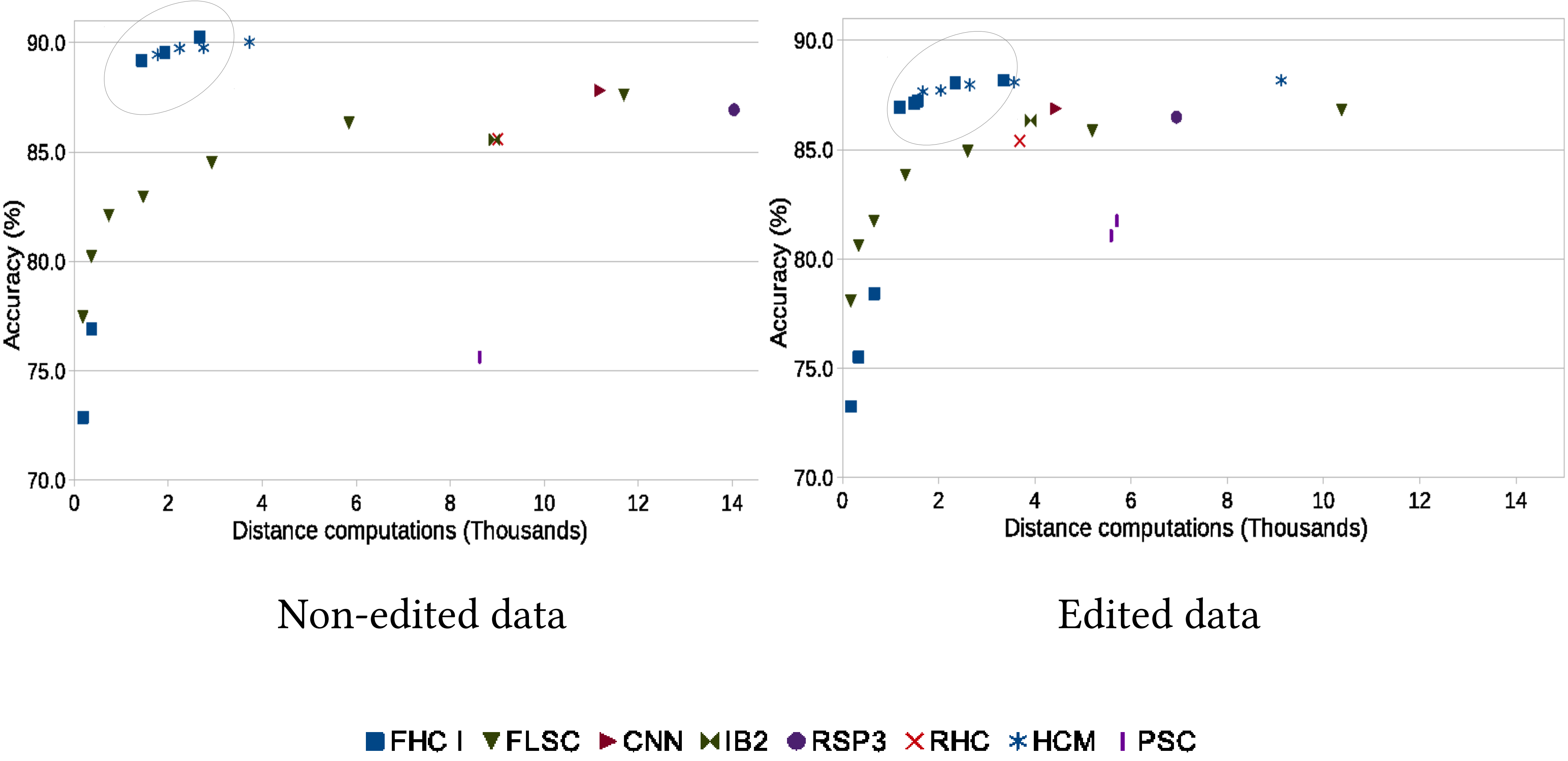


Non-edited data

Edited data

FHC: Experimental study (8/8)

PH dataset



Non-edited data

Edited data

Hybrid classification based on Homogeneous Clusters (1/5)

Motivation:

- TLDSCA and FHC include three parameters (*DRF, pk, npration*). The existence of these parameters may be characterized as weak points

Contribution:

- The development of non-parametric method that combines the idea of DRT with that of CBMs in a hybrid schema that follows the procedure of forming homogeneous clusters of RHC
- The development of a CBM which is applied in the condensing sets and is able to improve the performance of DRTs

Hybrid classification based on Homogeneous Clusters (2/5)

Speed-up Data Structure Construction Algorithm (SUDCA):

- It is non-parametric, pre-processing algorithm
- It builds the Speed-Up Data Structure (SUDS)
- It is based on the procedure of forming homogeneous clusters of RHC
- The length of SUDS is determined automatically without parameters

SUDS data levels:

- First level: A list of prototypes built by RHC
- Second level: Each prototype indexes the "real" cluster items which are stored in the second level

Hybrid classification based on Homogeneous Clusters (3/5)

When a new item *x* must be classified:

- HCAHC scans the first SUDS level and retrieves the *pk* nearest prototypes

- If all pk cluster prototypes vote a specific class, x is classified to this class (first level search)

- Otherwise, *x* is classified by searching the *k* “real” nearest items within the subset formed by the union of the clusters of the *pk* Prototypes (second level search)

Hybrid classification based on Homogeneous Clusters (4/5)

HCAHC can not characterized as neither DRT nor CBM. It is a hybrid method:

- First level search is an abstraction DRT (similar to RHC)
- Second level search is a CBM

HCAHC is a parametric algorithm. However *pk* can be determined by the empirical rule:

$$Rk=\lfloor\sqrt{|SUDS\square}\rfloor$$

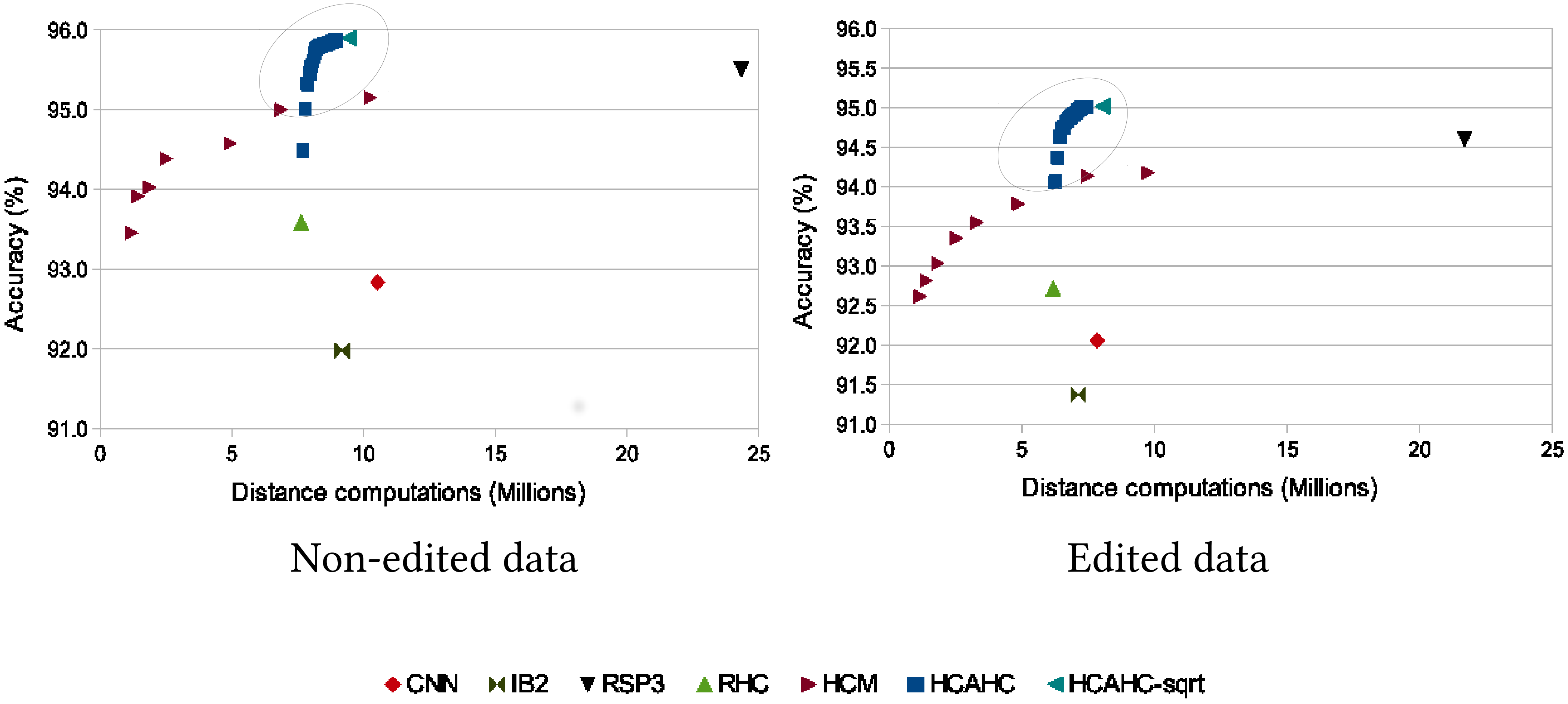
Hybrid classification based on Homogeneous Clusters (5/5)

SUDS classification method over condensing sets:

- We suggest the SUDS classification method to be applied on the data stored in a condensing set
- A classifier that uses SUDS will be executed faster than the *k*-NN classifier that searches for nearest neighbours in the condensing set. The classifier that uses SUDS prunes distance computations, without loss of accuracy
- Since SUDSCA is applied on a condensing set (i.e., a small dataset), the preprocessing overhead introduced will be almost insignificant
- The proposed classifier (HCA) avoids classification through first level search

HCAHC: Experimental study (1/7)

LIR dataset

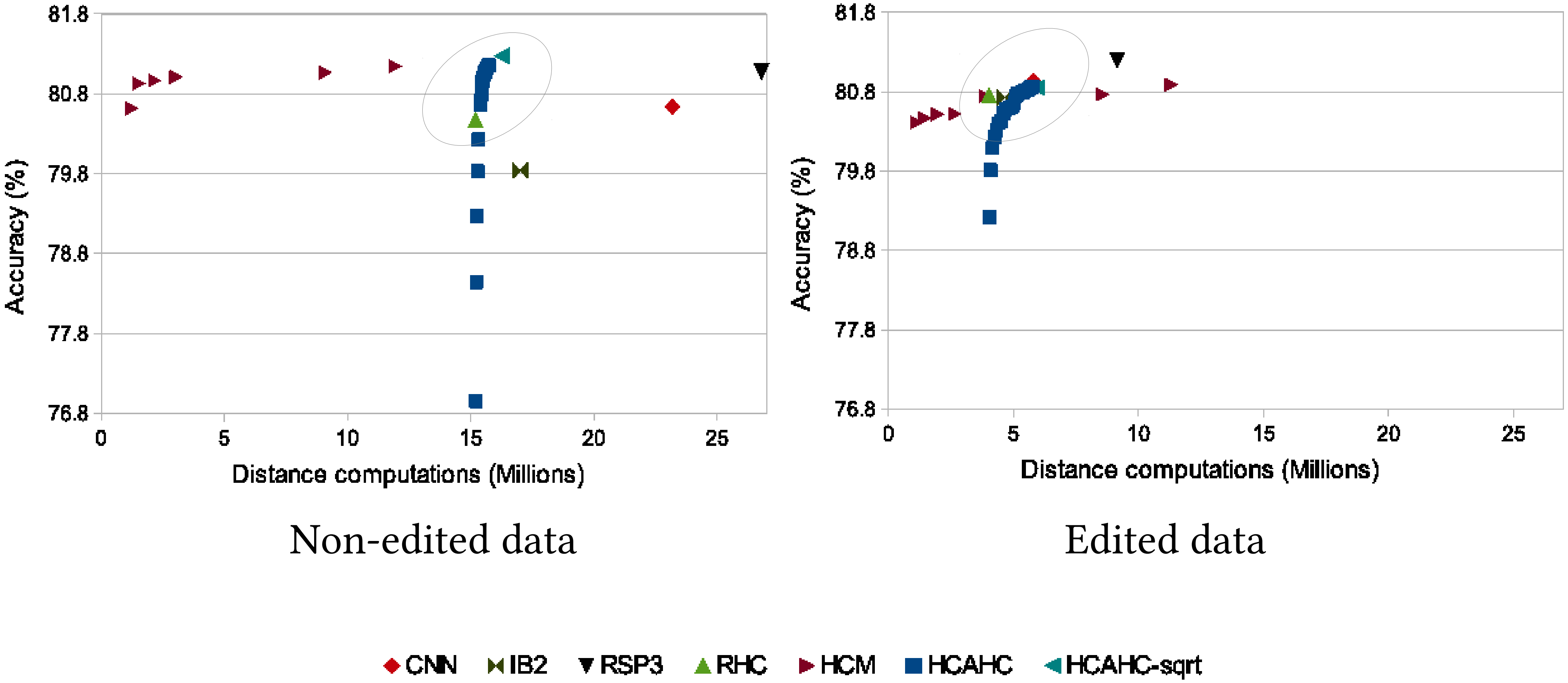


Non-edited data

Edited data

HCAHC: Experimental study (2/7)

MGT dataset

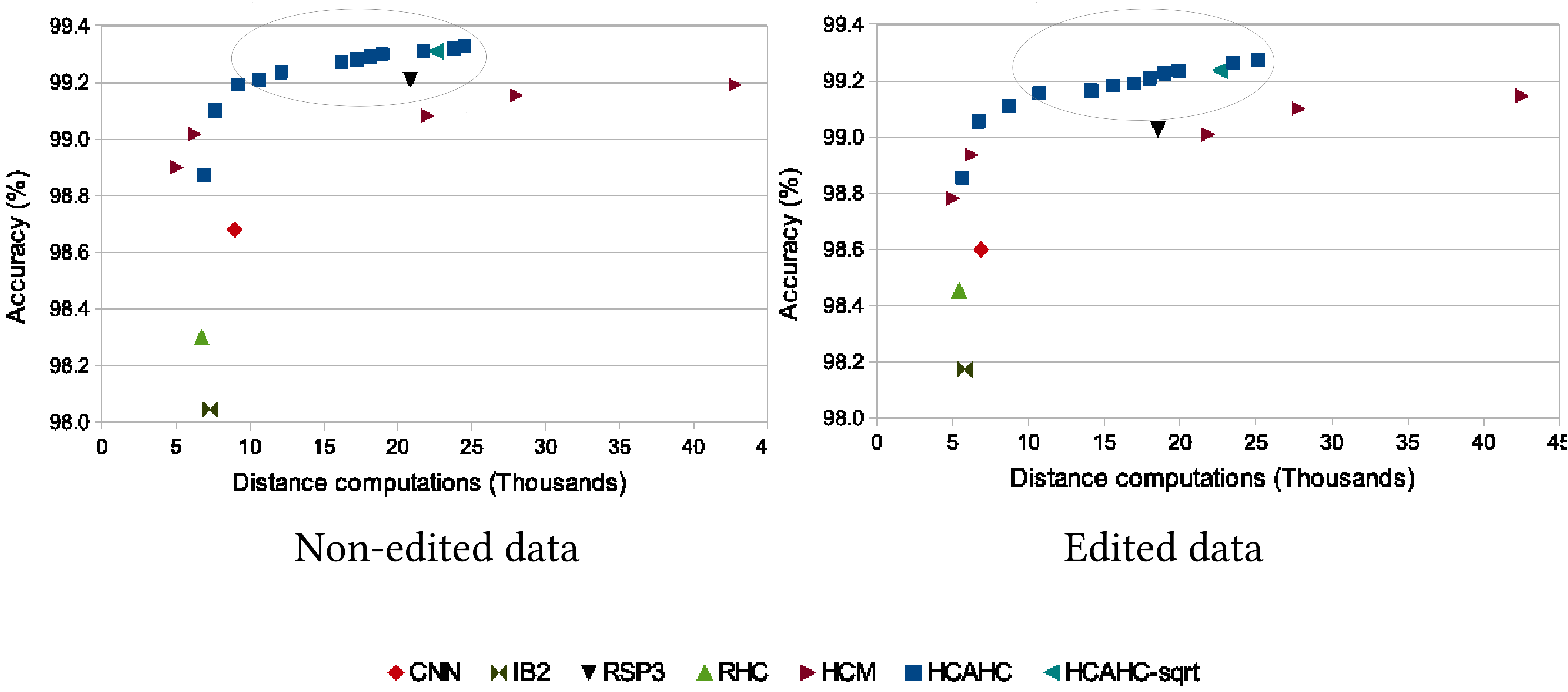


Non-edited data

Edited data

HCAHC: Experimental study (3/7)

PD dataset

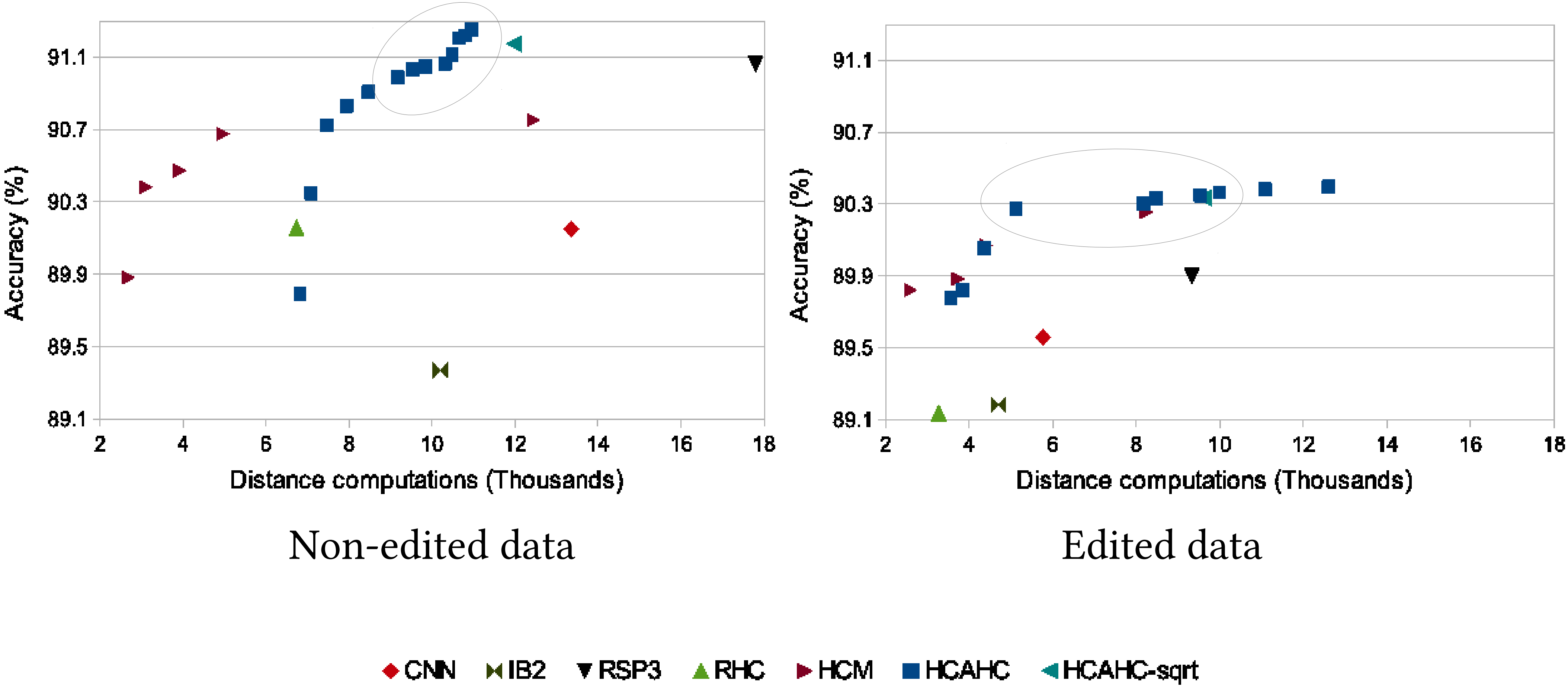


Non-edited data

Edited data

HCAHC: Experimental study (4/7)

LS dataset

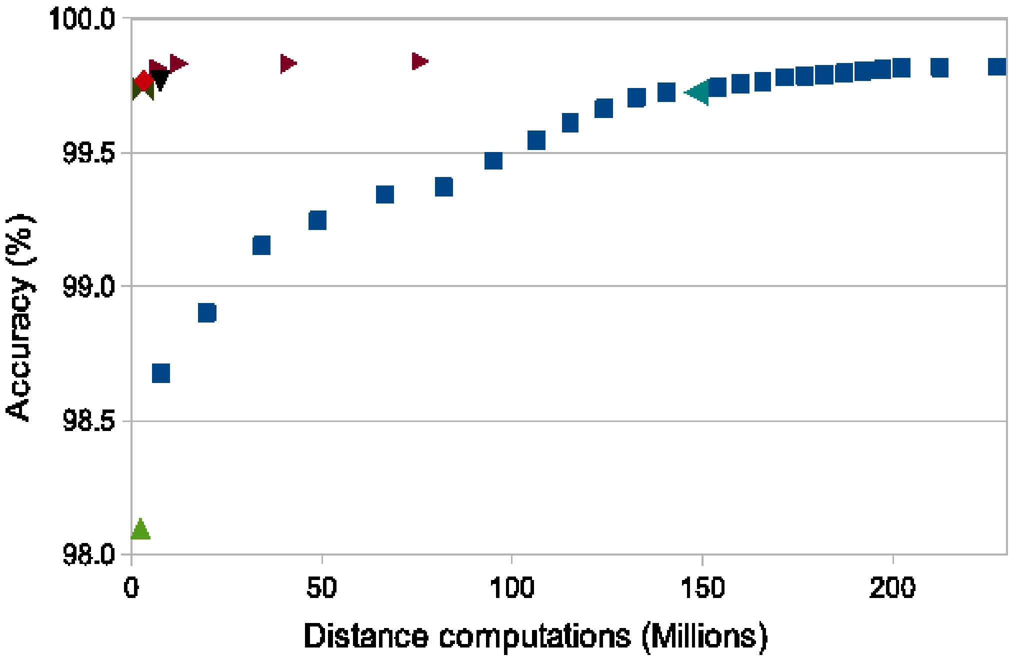


Non-edited data

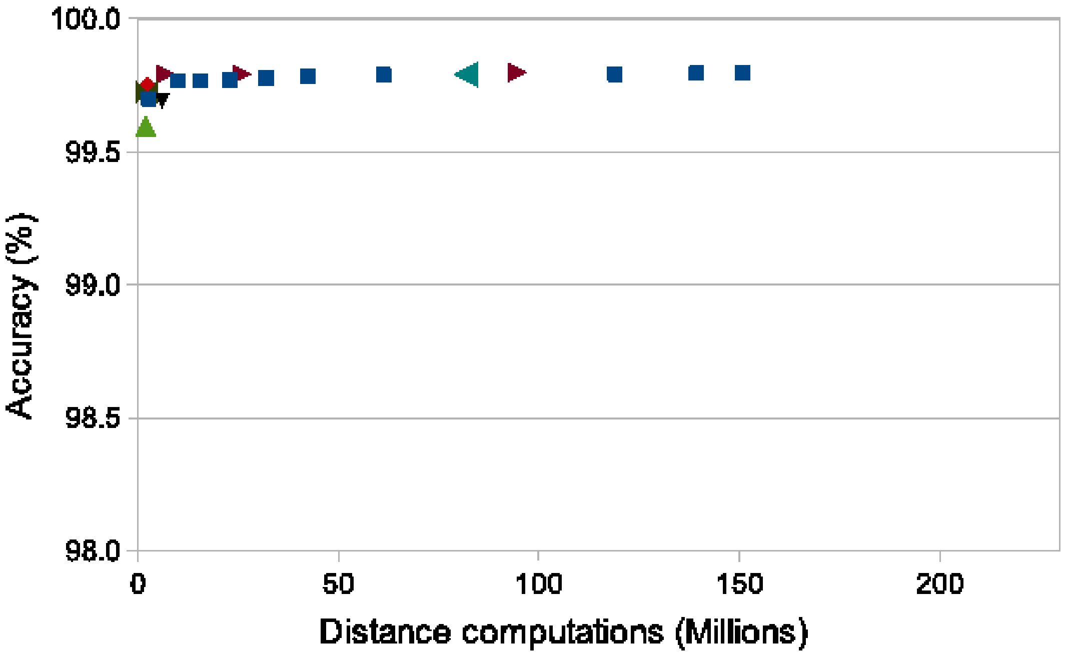
Edited data

HCAHC: Experimental study (5/7)

SH dataset



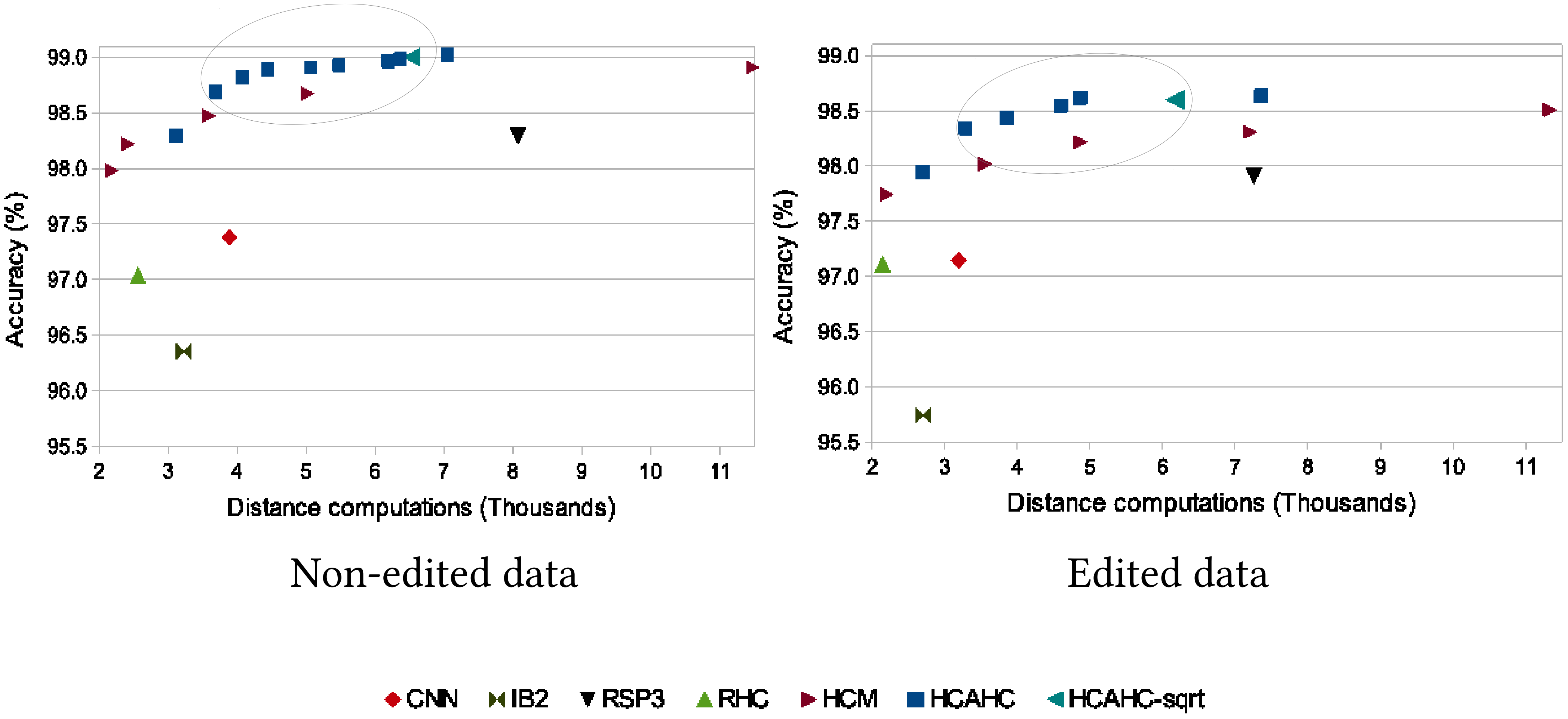
Non-edited data



Edited data

HCAHC: Experimental study (6/7)

TXR dataset

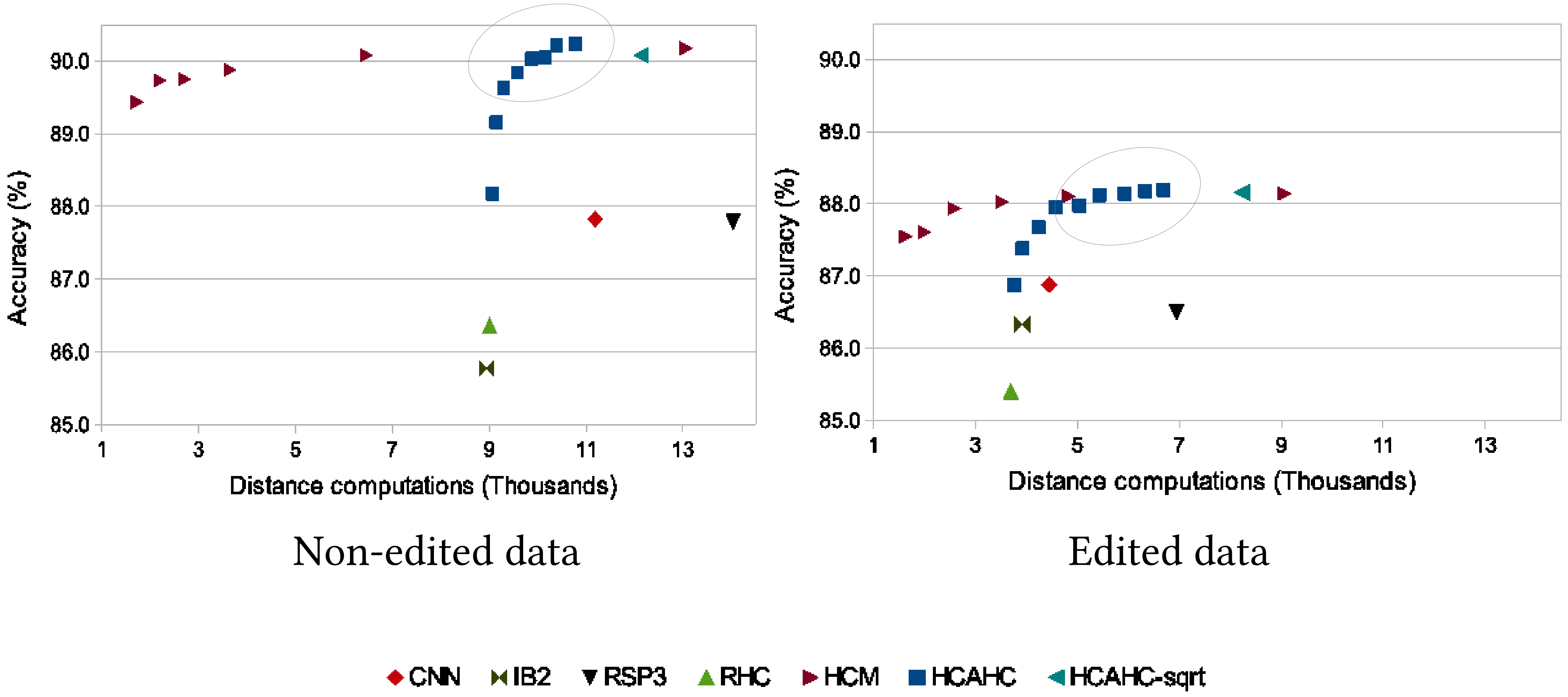


Non-edited data

Edited data

HCAHC: Experimental study (7/7)

PH dataset

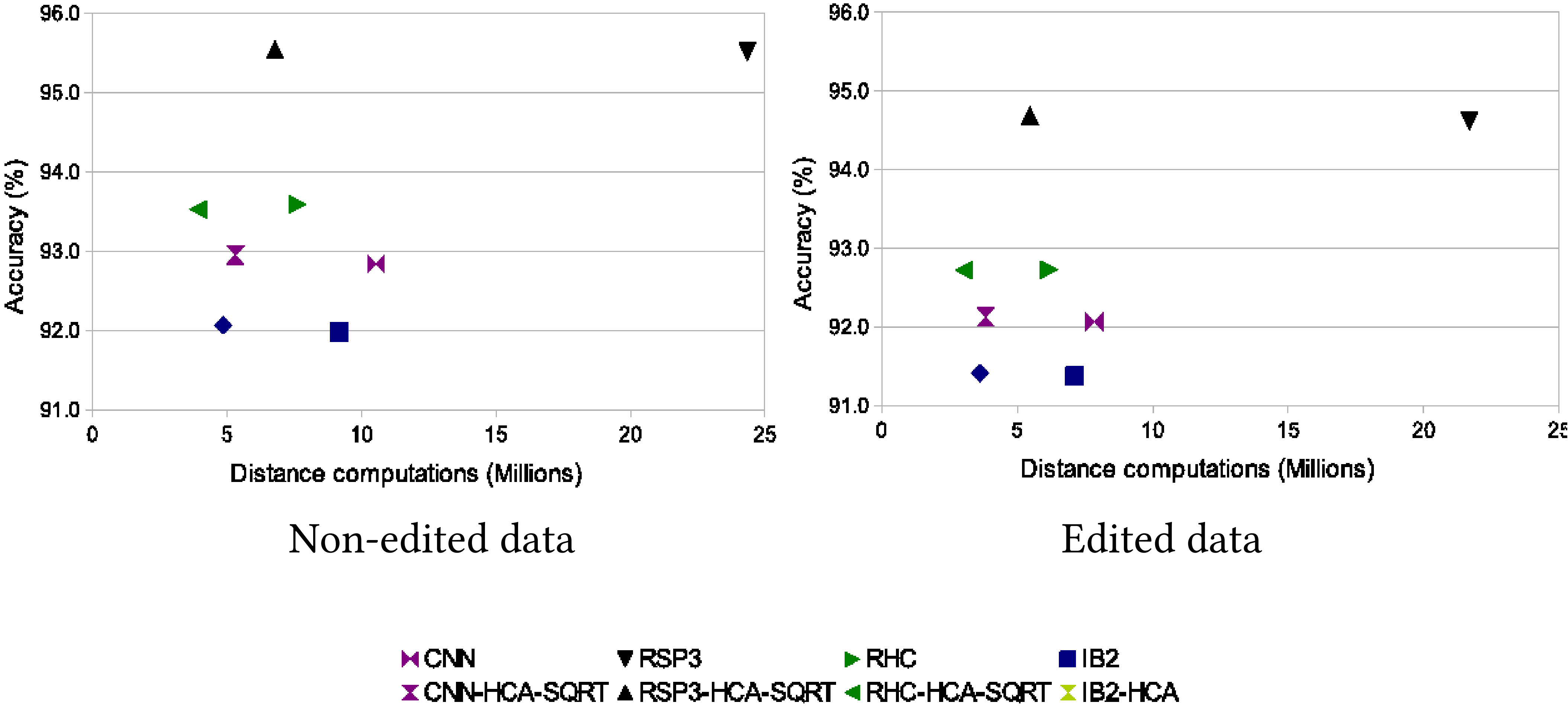


Non-edited data

Edited data

HCA: Experimental study (1/7)

LIR dataset

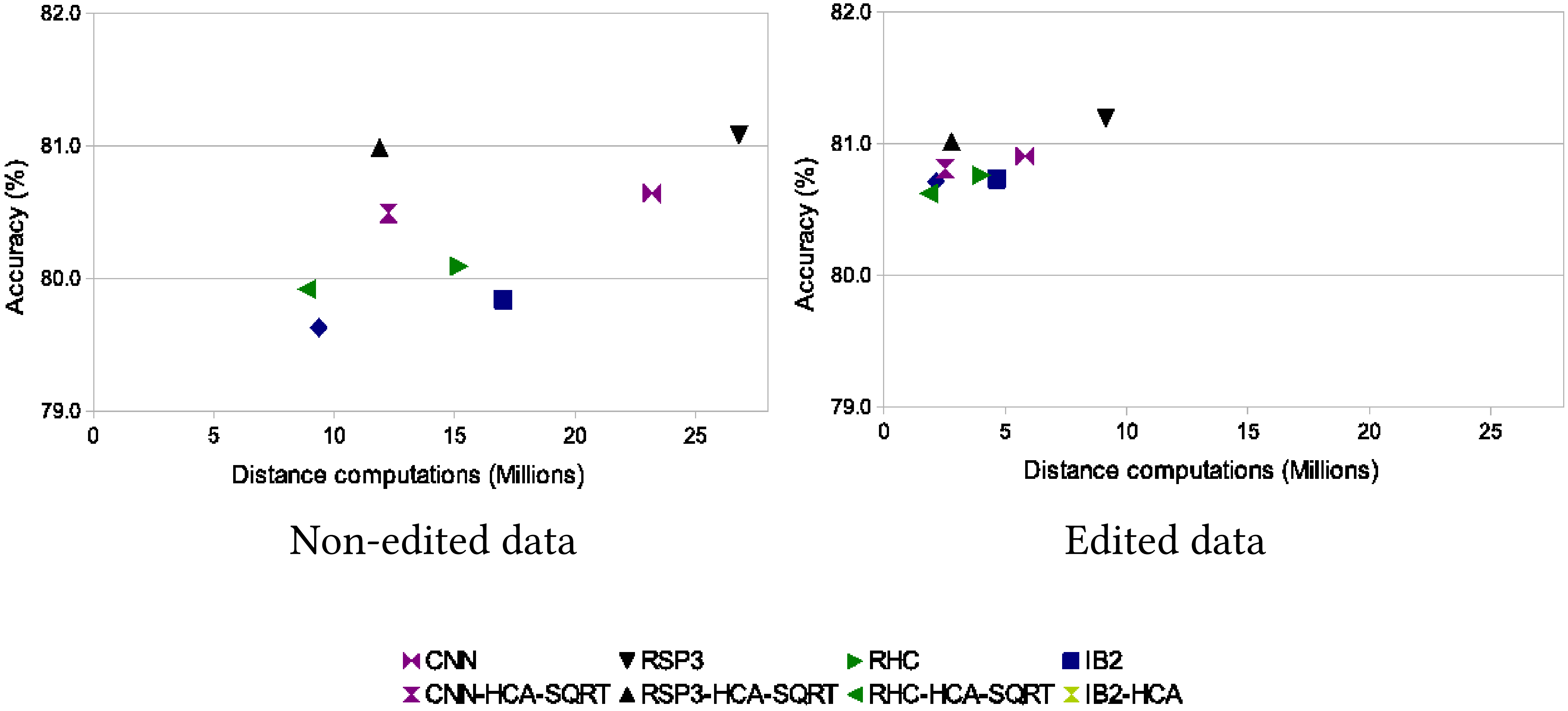


Non-edited data

Edited data

HCA: Experimental study (2/7)

MGT dataset

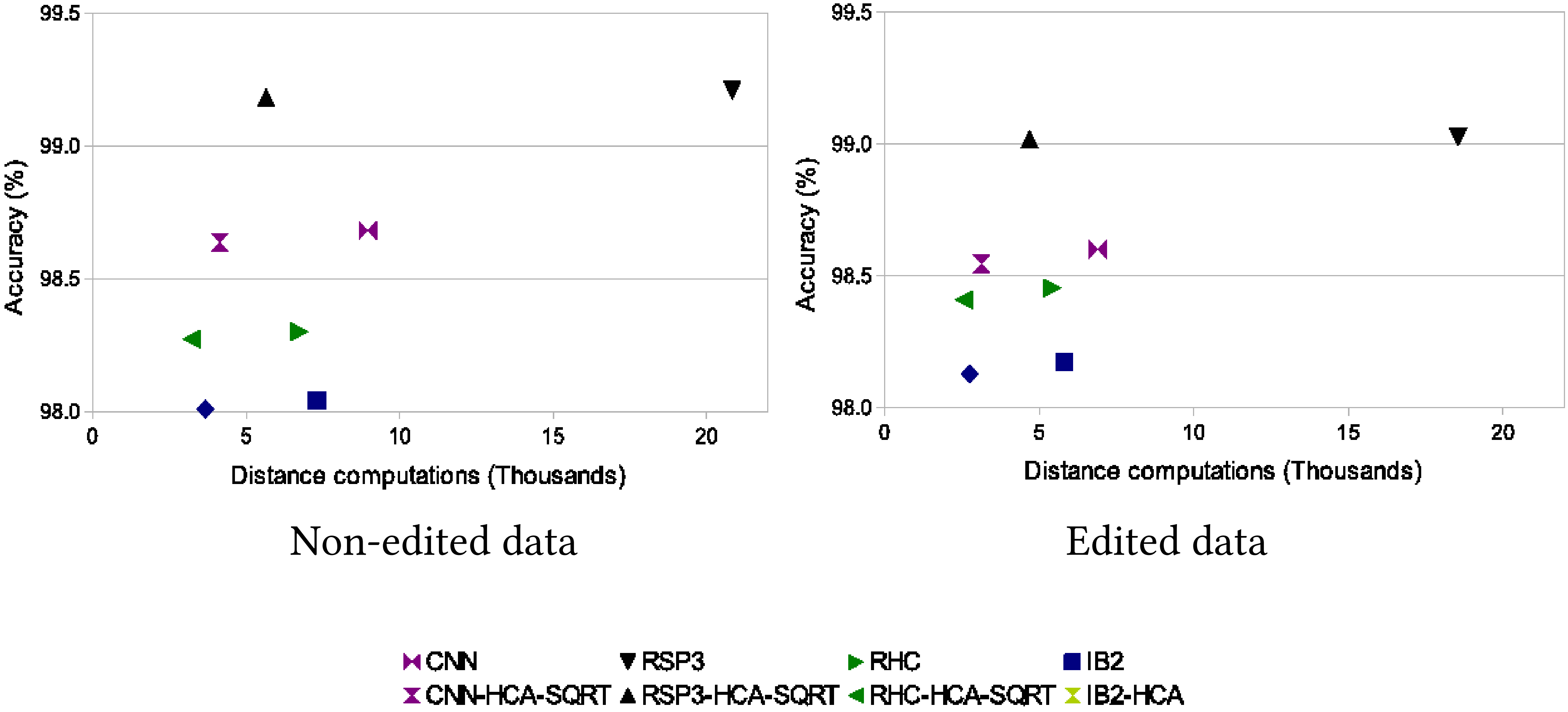


Non-edited data

Edited data

HCA: Experimental study (3/7)

PD dataset

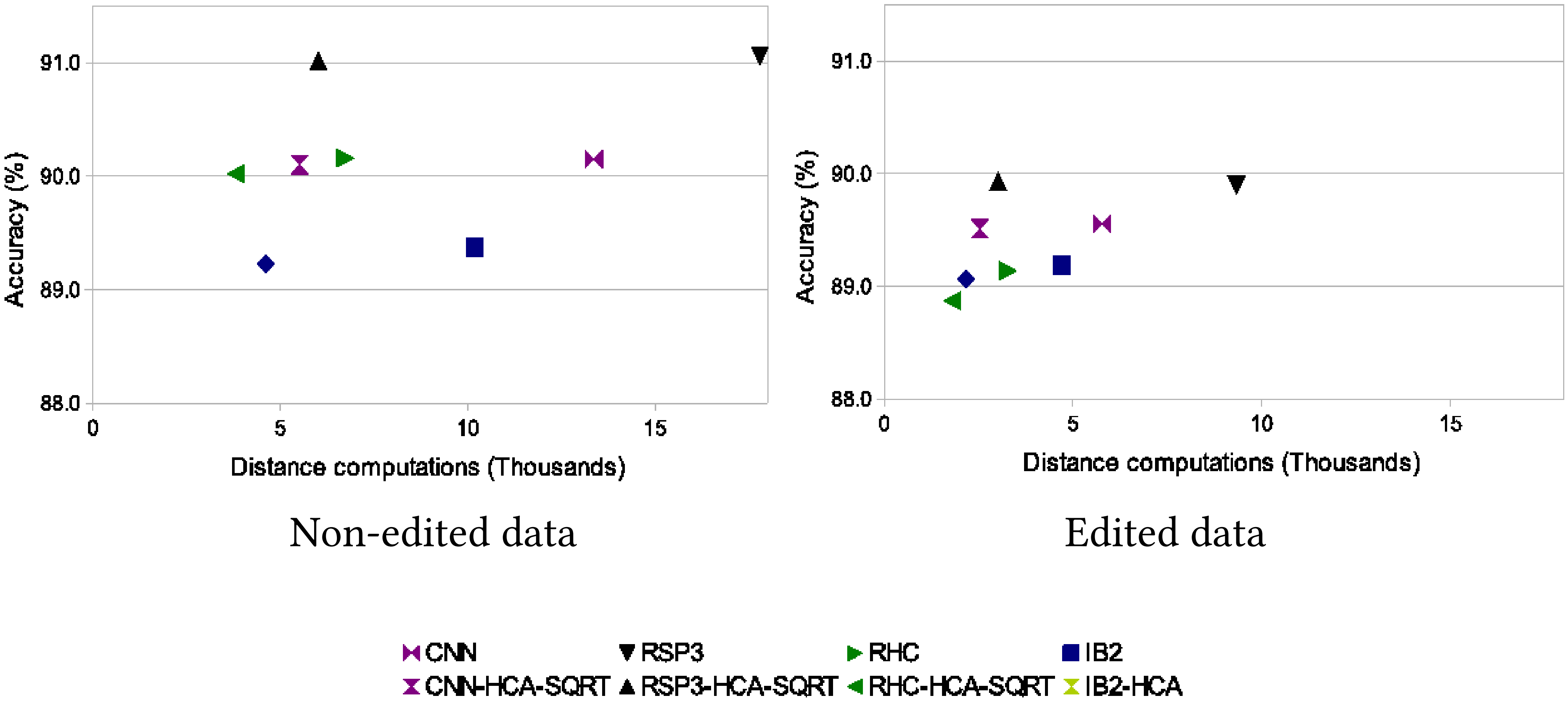


Non-edited data

Edited data

HCA: Experimental study (4/7)

LS dataset

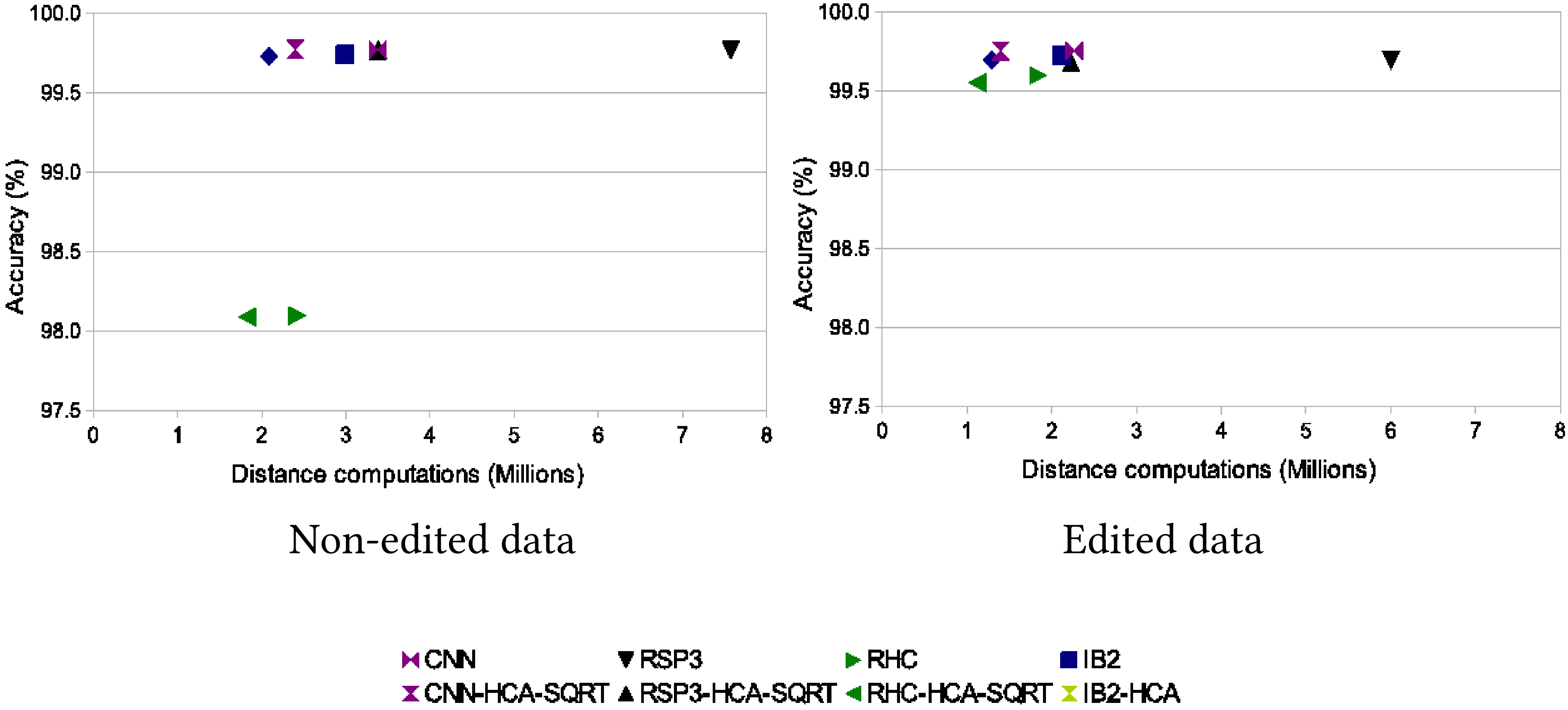


Non-edited data

Edited data

HCA: Experimental study (5/7)

SH dataset

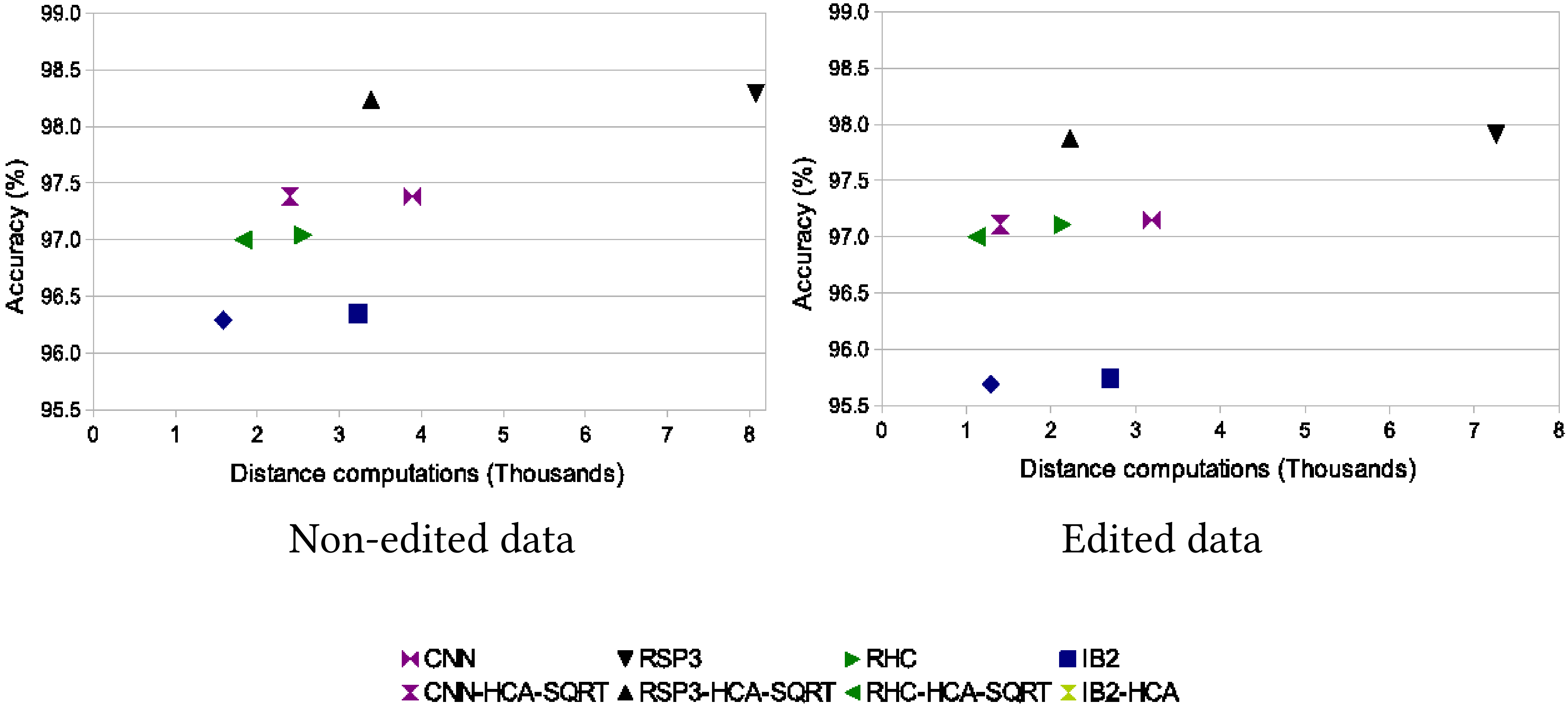


Non-edited data

Edited data

HCA: Experimental study (6/7)

TXR dataset

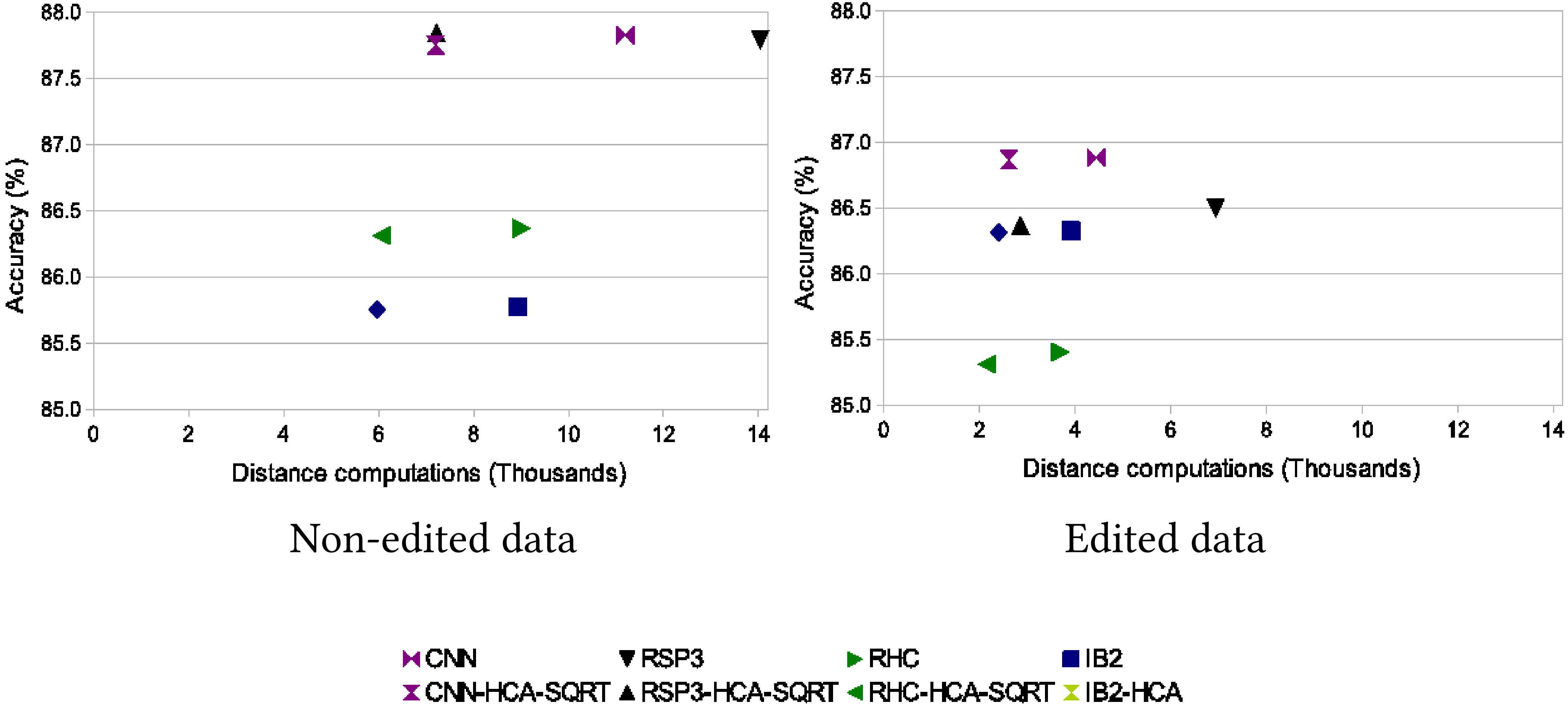


Non-edited data

Edited data

HCA: Experimental study (7/7)

PH dataset



Non-edited data

Edited data

Future work

Development of non-parametric one-pass DRTs that take into account the phenomenon of concept drift that may exist in data streams

Enhancements and modifications on existing algorithms and techniques so that they can cope with large and fast data streams (with or without concept drift)

Parallel implementations of DRTs for fast construction of condensing sets

Development of DRTs that can be applied in complex problems such as multi-label classification

DRTs for imbalanced training data

Thank you

for your attention