Dynamic k-NN Classification Based on Region Homogeneity

24th European Conference on Advances in Databases and Information Systems

Stefanos Ougiaroglou
stoug@ihu.gr
Dept. of Information and Electronic Engineering
International Hellenic University

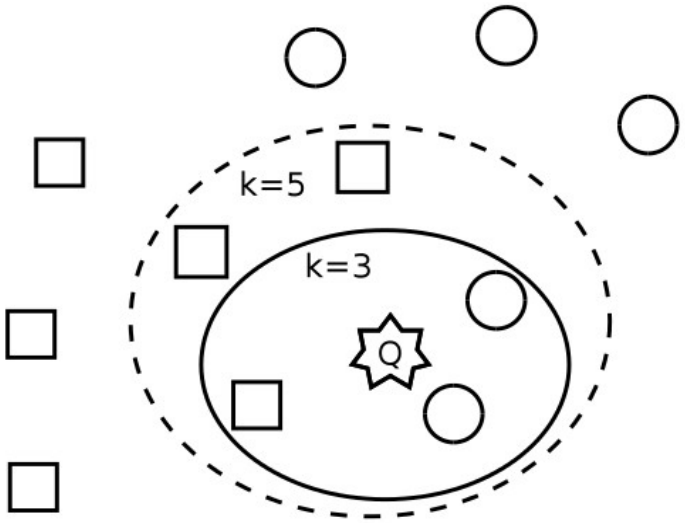
Georgios Evangelidis
gevan@uom.edu.gr
Dept. of Applied Informatics
University of Macedonia

Konstantinos Diamantaras
kdiamant@ihu.gr
Dept. of Information and Electronic Engineering
International Hellenic University

k-NN Classification

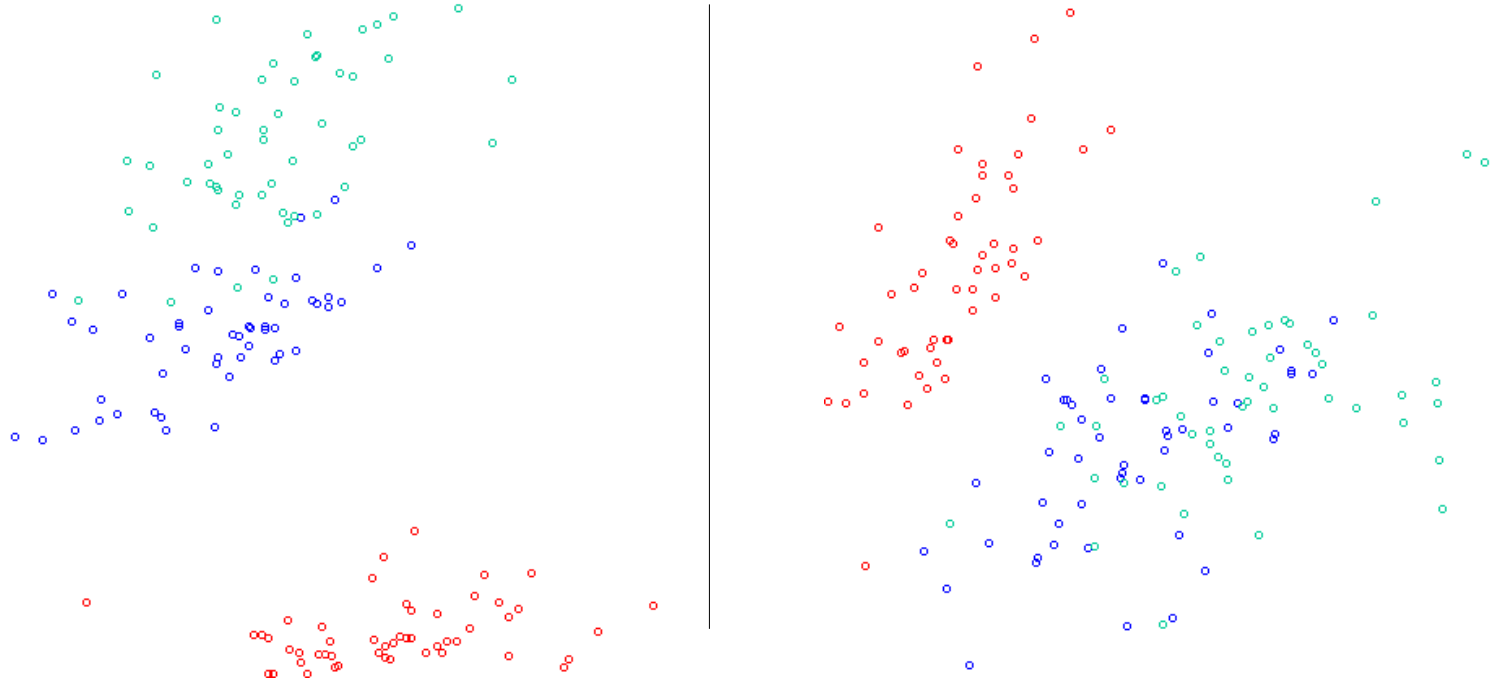
k-NN classifier works by searching the available training data for the k nearest items to the unclassified item. Then, these neighbors determine the class where the new item belongs to.

- Accuracy depends on the selection of *k*. Tedious cross-validation tasks are performed to determine “best” *k*. It is unique and constant for all unclassified instances
- A large *k* value renders the classifier more noise tolerant since it examines larger neighbourhoods. It has to be used when the classes are not well separated and when the data contains noise.
- A small *k* renders the classifier noise sensitive and should be used on training sets with well-separated classes



Motivation

- Even the “best” *k* value can not be optimal. Real-life datasets may have quite different structure in different regions of the metric space.



- A classifier that uses a fixed *k* value may be less accurate than a classifier that utilizes a different *k* value for each instance *x* that needs to be classified depending on the region where *x* lies.

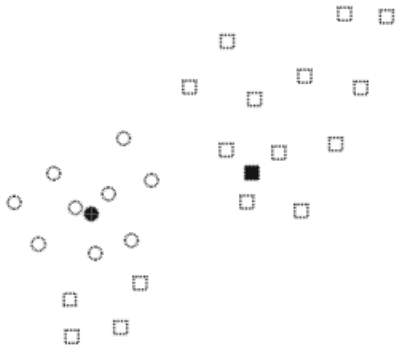
Contribution

Region Homogeneity based Dynamic k-NN classifier (rhd-kNN)

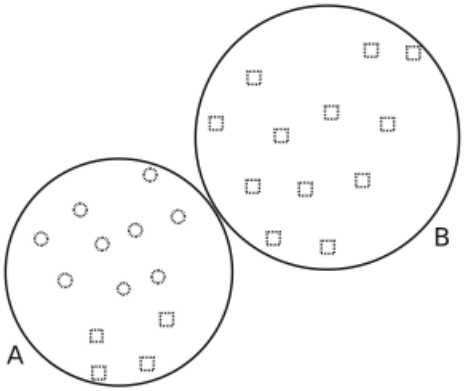
- A parameter free k-NN classifier that it uses a dynamic *k* value depending on nature of the region where the instance to be classified lies.
- rhd-kNN utilizes heuristics that dynamically adjust the k value.
- The heuristics are based on a data structure constructed k-means clustering pre-processing task that builds homogeneous clusters. The data structure (SHC) holds the cluster centroids and information about the region of each centroid.
- When a new instance *x* needs to be classified, the nearest centroid *c* from SHC is retrieved. Then based on *c*, k is appropriately adjusted and *x* is classified by searching the *k* nearest neighbours in the training set.

rhd-kNN: SHC Construction

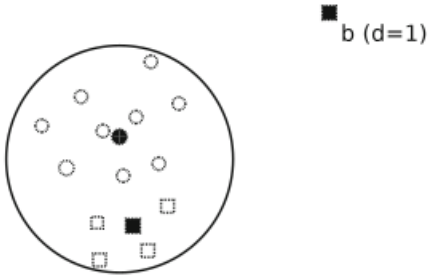
(a) initial data



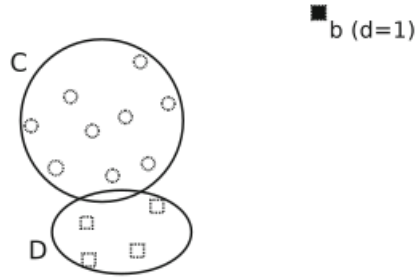
(b) initial class means



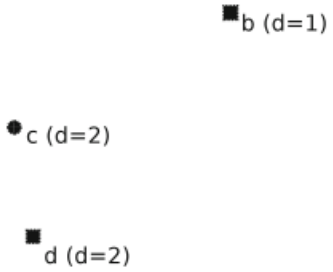
(c) k-means on initial data



(d) Cluster centroid and class means

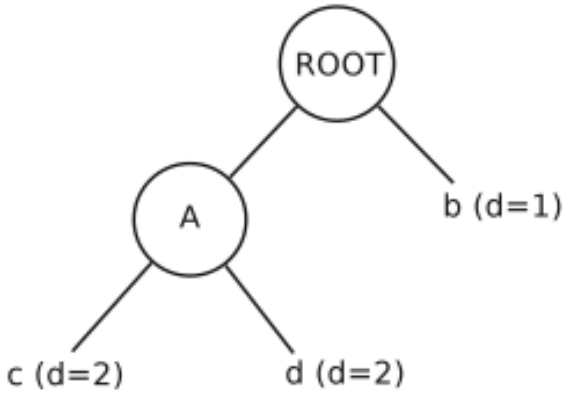


(e) k-means on a non-homogeneous cluster



(f) final set of cluster centroids

rhd-kNN: Heuristics (1/2)



- When a new instance *x* needs to be classified, rhd-kNN finds the 1-nearest centroid *c* in SHC and its corresponding *d*.
- Then, one heuristic is employed to determine *k* based on *d*. rhd-kNN classifies *x* by finding the *k* nearest neighbours in the training set.
- We expect that rhd-k-NN will outperform the k-NN classifiers that use fixed *k* for datasets that contain a mixture of well-separated and not well-separated classes

rhd-kNN: Heuristics (2/2)

- $k=d$: It defines k to be equal to the depth of the 1-nearest centroid in SHC. This heuristic is used as a baseline
- $k=2^d$: This heuristic tends to examine an extremely large number of nearest neighbours, especially when d is greater than 9. Thus, in our experiments, we manually set $k = 2^9$ when $d > 9$.
- $k=d^2$: This heuristic is a trade-off between the above two heuristics
- $k=(d\times(d+1))/2$: This heuristic determines k by mapping values of d to the following arithmetic sequence: 1, 3, 6, 10, 15, 21, 28, . . .
- k= $\lfloor e^{\sqrt{d}} \rfloor$: This is a more conservative heuristic than the previous one. It uses the 2, 4, 5, 7, 9, 11, 14, . . . sequence for determining k

Experiments (1/4)

Dataset	Size	Attributes	Classes
Balance (bl)	625	4	3
Banana (bn)	5300	2	2
Ecoli (ecl)	336	7	8
Iris	120	4	3
Letter Recognition (lir)	20000	16	26
Landsat Satellite (ls)	6435	36	6
Magic G. Telescope (mgt)	19020	10	2
Pen-Digits (pd)	10992	16	10
Phoneme (ph)	5404	5	2
Pima (pm)	615	8	2
Shuttle (sh)	58000	9	7
Twonorm (tn)	7400	20	2
Texture (txr)	5500	40	11
Yeast (ys)	1484	8	10

We built two additional “noisy” versions (e.g., txr30) of the 14 datasets by adding 10% and 30% random uniform noise

We compared the performance of rhd-kNN against 6 k-NN classifiers that use fixed k values:

- The widely used 1-NN
- The 5-NN and the 10-NN classifiers
- The common rule-of-thumb approaches:

$$k = \sqrt{N} \quad \text{and} \quad k = \sqrt{\frac{N}{2}}$$

- The k-NN classifier with the “best” tuned *k* (by applying a 5-fold cross validation)

Experiments (2/4)

Accuracy measurements

- Almost in all cases an rhd-kNN classifier can achieve higher accuracy than the accuracy achieved by k-NN with *k*=1, *k*=5, *k*=10
- rhd-kNN classifiers almost always outperform RoT classifiers
- The comparison between the “best” k-NN and rhd-kNN reveals that at least one of the rhd-kNN classifiers can achieve higher accuracy than that of the best k-NN classifier in 18 datasets, while in 2 datasets, a rhd-kNN approach is as accurate as the “best” k-NN.
- Contrary to the best k-NN classifier, rhd-kNN achieves that performance without the need of any input parameter and tedious and costly parameter tuning procedures
- The *k*=(*d*×(*d*+1))/2 heuristic seems to be an ideal approach since it achieves high accuracy even when the dataset contains noise. In 9 datasets it is more accurate than best k-NN

Data	Best k value	Best k-NN	1-NN	5-NN	10-NN	RoT $k=\sqrt{N}$	RoT $k=\sqrt{\frac{N}{2}}$	rhd-kNN $k=d$	rhd-kNN $k=2^d$	rhd-kNN $k=d^2$	rhd-kNN $k=(d\times(d+1))/2$	rhd-kNN $k=\lfloor e^{\sqrt{d}} \rfloor$
AVG:	19.42	**87.63**	76.77	84.33	86.72	85.33	86.15	84.63	86.19	86.82	87.02	86.70

Experiments (3/4)

Data	Best k value	Best k-NN	1-NN	5-NN	10-NN	RoT $k = \sqrt{N}$	RoT $k = \sqrt{\frac{N}{2}}$	rhd-kNN $k = d$	$k = 2^d$	$k = d^2$	$k = (d \times (d+1))/2$	$k = \lfloor e^{\sqrt{d}} \rfloor$
bl	42	89.60	79.20	86.40	89.60	89.60	89.60	84.80	88.80	89.60	**90.40**	89.60
bl10	43	88.80	68.80	86.40	87.20	89.60	89.60	88.00	**91.20**	89.60	90.40	87.20
bl30	32	**88.00**	59.20	69.60	79.20	83.20	83.20	73.60	83.20	87.20	82.40	76.80
bn	29	**90.66**	87.26	89.81	90.19	90.57	90.47	89.81	88.68	90.38	90.28	90.47
ecl	6	**91.05**	83.58	89.55	92.54	86.57	92.54	88.06	88.06	86.57	86.57	89.55
ecl10	6	89.55	77.61	88.06	92.54	88.06	89.55	86.57	91.05	**92.54**	91.05	91.05
ecl30	14	85.08	61.19	82.09	85.08	85.08	85.08	76.12	85.08	**88.06**	**88.06**	85.08
iris	11	93.33	90.00	93.33	93.33	93.33	93.33	93.33	93.33	93.33	**96.67**	93.33
lir	4	**95.78**	95.70	95.40	95.03	81.05	84.20	95.65	95.28	93.83	95.05	95.43
ls	8	**91.53**	89.98	91.53	91.14	86.09	87.34	91.22	90.68	89.59	90.99	90.83
ls10	8	**91.30**	81.66	90.52	90.99	86.09	87.10	90.75	90.37	89.90	89.98	90.75
ls30	13	88.27	61.23	82.21	89.04	86.25	86.79	82.98	88.58	89.36	**88.81**	87.72
mgt	10	**83.57**	80.13	82.97	83.57	80.97	81.65	83.36	78.97	80.86	81.86	83.39
mgt10	20	**83.10**	73.32	80.97	82.76	81.13	81.55	82.89	78.68	80.81	81.84	82.99
pd	1	99.05	99.05	99.09	98.73	95.04	96.13	**99.09**	98.59	98.32	98.54	98.91
pd10	8	98.86	89.40	98.95	98.73	95.13	96.36	**98.91**	98.73	98.36	98.64	**98.91**
pd30	12	**98.68**	69.75	94.18	98.45	94.90	96.36	92.58	98.04	98.23	98.41	97.59
ph	1	**88.70**	88.70	86.94	86.30	82.04	83.52	85.74	78.98	81.30	81.94	84.44
ph10	8	**86.67**	81.39	86.02	86.02	81.76	83.89	84.35	78.80	81.11	81.48	84.26
ph30	48	80.19	65.46	72.32	75.28	80.93	80.65	77.96	79.07	80.28	**80.93**	79.44

Experiments (4/4)

Data	Best k value	Best k-NN	1-NN	5-NN	10-NN	RoT $k = \sqrt{N}$	RoT $k = \sqrt{\frac{N}{2}}$	rhd-kNN $k = d$	$k = 2^d$	$k = d^2$	$k = (d \times (d+1))/2$	$k = \lfloor e^{\sqrt{d}} \rfloor$
ph	1	**88.70**	88.70	86.94	86.30	82.04	83.52	85.74	78.98	81.30	81.94	84.44
ph10	8	**86.67**	81.39	86.02	86.02	81.76	83.89	84.35	78.80	81.11	81.48	84.26
ph30	48	80.19	65.46	72.32	75.28	80.93	80.65	77.96	79.07	80.28	**80.93**	79.44
pm	48	76.47	77.12	78.43	81.05	75.16	77.78	80.39	72.55	74.51	73.86	**82.35**
pm10	18	78.43	72.55	76.47	78.43	75.16	78.43	75.82	77.12	77.78	**79.09**	77.78
pm30	47	73.86	64.05	64.71	69.94	71.90	71.24	71.24	73.86	71.90	73.20	**76.47**
sh	1	**99.96**	99.96	99.91	99.86	99.44	99.36	99.89	99.72	99.67	99.83	99.85
tn	49	97.97	95.54	97.70	97.77	97.77	98.04	97.77	**98.11**	97.70	97.91	98.04
tn10	47	97.77	84.39	95.47	97.10	97.57	97.50	97.10	**97.91**	97.43	97.70	97.70
tn30	50	**97.77**	67.23	79.19	86.08	97.43	97.57	86.69	**97.77**	97.03	97.57	94.73
txr	1	**98.73**	98.73	98.09	97.82	94.73	95.82	98.36	98.09	97.55	98.00	98.09
txr10	5	97.91	90.55	97.91	97.82	94.55	95.55	**98.09**	97.73	97.46	97.64	97.73
txr30	10	**97.55**	69.73	93.55	97.55	94.82	94.91	91.09	96.55	97.00	**97.55**	96.64
ys	14	58.45	49.32	57.10	59.12	61.15	60.14	55.41	61.15	60.14	**61.49**	57.43
ys10	13	58.45	44.93	52.37	58.11	60.14	58.11	51.01	59.46	**59.80**	57.43	57.43
ys30	14	56.76	36.82	45.61	55.41	58.78	59.46	44.26	50.00	**57.77**	56.08	48.99

THANK YOU
FOR YOUR
ATTENTION

Stefanos Ougiaroglou

stoug@ihu.gr
https://www.iee.ihu.gr/~stoug