KNN AND CONDENSED KNN

Experiment Report

SEPTEMBER 17, 2014

AJAY VIJAYAKUMARAN NAIR

Contents

kNN

testY = testknn(trainX, trainY, testX, k)
trainX - the training sample where letters A-Z of the alphabet are represented as 16 dimensional vectors
trainY - the class each representation in trainX represents as a 1 dimensional vector
testX - the 16D representation of sample vectors for which the class needs to be predicted
k - the number of nearest neighbours to look for in deciding the class of unknown sample
testY - the predicted class of each of the samples in testY

The algorithm calculates the pairwise distance between each vector in trainX to each vector in testX. For each of the vectors in testX, the k closest vectors in trainX are found. These will be the vectors with the smallest distance between them. The majority of the classes of these k nearest vectors is considered as the class of the unknown sample. The majority is calculated using mode function and in cases of a tie, the first appearing class is selected. To calculate distance between two vectors, the Euclidean distance function (pdist2) is used.

Condensation based on 1NN

condensedIdx = condensedata(trainX, trainY)
trainX - the training sample where letters A-Z of the alphabet are represented as 16 dimensional vectors
trainY - the class each representation in trainX represents as a 1 dimensional vector
condensedIdx - the indices of trainX which have been retained as condensed sample

The algorithm randomly picks up a sample from trainX and is added to a subset. With this subset, the rest of trainX is classified using 1NN. From the resulting prediction, an incorrectly classified sample is picked up and is added to the subset. The resulting subset is now used to reclassify the remaining trainX. This process continues until the entire trainX is correctly classified using the subset.

Assumptions

- It is assumed that all vector representations of the alphabet provided consists of all 16 attributes in the same order.
- The condensation is run once for each of the sample size selections and testknn and condensedknn works with the same sample selection. (The sample selection is random from the first 15000 records of the data set). The test data is always the remaining 5000 vectors
- To validate the condensation results, the condensed training samples were passed onto testknn as training data and the sample data which was condensed was passed as test data.

Observations

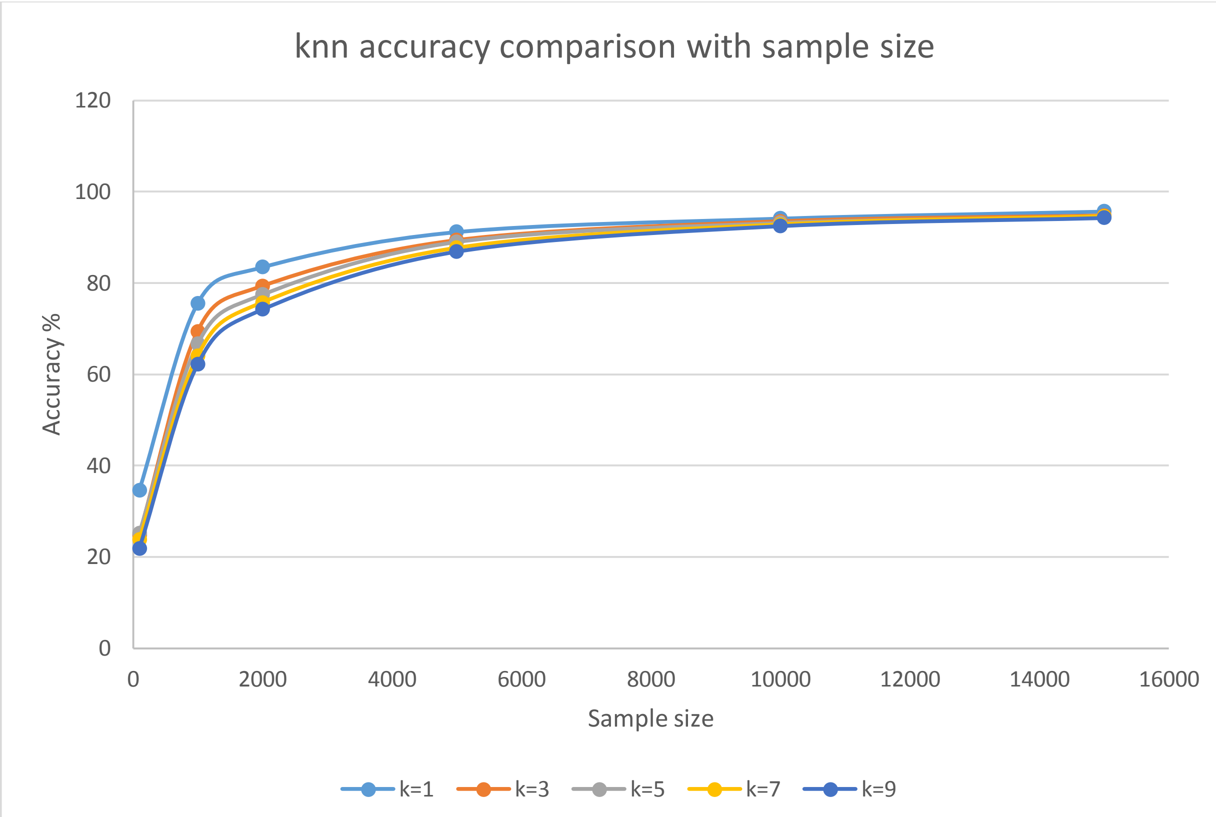
- Zero classification error proved the consistency of the condensed sample with the training sample.
- Initially, the condensation algorithm used double loops to reclassify the remaining samples. This was proving to be inefficient and was replaced with vectorized functions. The result proved ~70% of execution time improvement.
- Different distance functions apart from Euclidean was tried. Chebychev and Minkowski both decreased the accuracy compared to Euclidean.
- Condensation, as expected, proved to decrease the accuracy of predictions compared to non-condensed knn.

Condensation algorithm observations

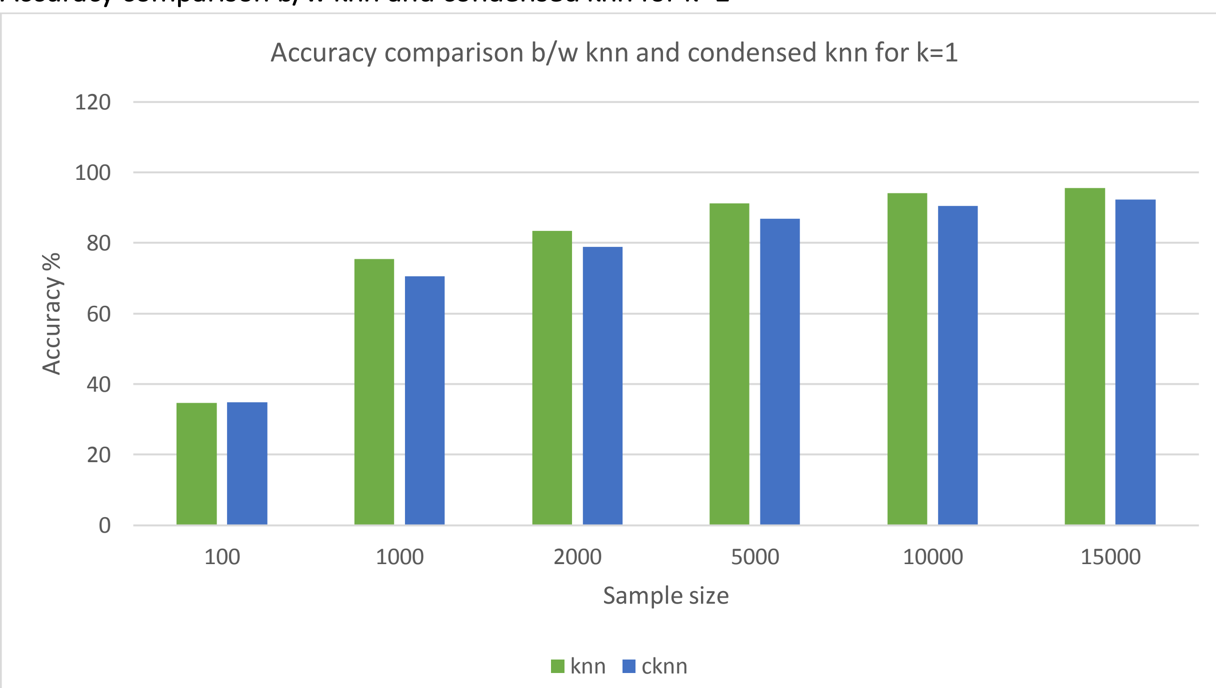
Sample size	Condensed size	Condensing ratio	Time taken in seconds	Time in minutes
100	85	15%	0.25	0
1000	454	55%	3.94	0
2000	704	65%	20.08	0
5000	1325	74%	188.7	3
10000	1958	80%	884.73	15
15000	2482	83%	2239.09	37

Experiment analysis

knn accuracy comparison with sample size



Accuracy comparison b/w knn and condensed knn for k=1



Detailed tabular results

Algorithm	k	Training sample size	Execution time in s	Accuracy %	True +ves	True -ves
knn	1	100	0.02	34.6	1730	3270
knn	1	1000	0.27	75.54	3777	1223
knn	1	2000	0.54	83.46	4173	827
knn	1	5000	1.34	91.18	4559	441
knn	1	10000	3.25	94.14	4707	293
knn	1	15000	4.35	95.68	4784	216
knn	3	100	0.12	24.76	1238	3762
knn	3	1000	0.34	69.4	3470	1530
knn	3	2000	0.64	79.3	3965	1035
knn	3	5000	1.45	89.34	4467	533
knn	3	10000	2.95	93.48	4674	326
knn	3	15000	4.41	94.74	4737	263
knn	5	100	0.15	25.24	1262	3738
knn	5	1000	0.36	66.64	3332	1668
knn	5	2000	0.6	77.42	3871	1129
knn	5	5000	1.47	88.96	4448	552
knn	5	10000	2.92	93.12	4656	344
knn	5	15000	4.39	94.5	4725	275
knn	7	100	0.1	23.8	1190	3810
knn	7	1000	0.38	64.04	3202	1798
knn	7	2000	0.61	75.72	3786	1214
knn	7	5000	1.39	87.66	4383	617
knn	7	10000	3.03	92.8	4640	360
knn	7	15000	4.42	94.6	4730	270
knn	9	100	0.1	21.88	1094	3906
knn	9	1000	0.33	62.18	3109	1891
knn	9	2000	0.6	74.2	3710	1290
knn	9	5000	1.39	86.88	4344	656
knn	9	10000	3.01	92.48	4624	376
knn	9	15000	4.57	94.26	4713	287
Condensedknn	1	85	0.02	34.76	1738	3262
Condensedknn	1	454	0.12	70.62	3531	1469
Condensedknn	1	704	0.16	78.86	3943	1057
Condensedknn	1	1325	0.37	86.84	4342	658
Condensedknn	1	1958	0.59	90.56	4528	472

Condensedknn	1	2482	0.71	92.28	4614	386
Condensedknn	3	85	0.11	22.44	1122	3878
Condensedknn	3	454	0.2	52.88	2644	2356
Condensedknn	3	704	0.27	60.44	3022	1978
Condensedknn	3	1325	0.42	75.3	3765	1235
Condensedknn	3	1958	0.68	81.5	4075	925
Condensedknn	3	2482	0.77	84.66	4233	767
Condensedknn	5	85	0.13	22.88	1144	3856
Condensedknn	5	454	0.19	51.58	2579	2421
Condensedknn	5	704	0.26	59.2	2960	2040
Condensedknn	5	1325	0.48	72.04	3602	1398
Condensedknn	5	1958	0.69	77.64	3882	1118
Condensedknn	5	2482	0.8	81.88	4094	906
Condensedknn	7	85	0.09	20.58	1029	3971
Condensedknn	7	454	0.2	48.16	2408	2592
Condensedknn	7	704	0.35	56.14	2807	2193
Condensedknn	7	1325	0.45	66.92	3346	1654
Condensedknn	7	1958	0.68	73.52	3676	1324
Condensedknn	7	2482	0.77	77.86	3893	1107
Condensedknn	9	85	0.1	18.6	930	4070
Condensedknn	9	454	0.24	45.76	2288	2712
Condensedknn	9	704	0.28	53.6	2680	2320
Condensedknn	9	1325	0.45	65.64	3282	1718
Condensedknn	9	1958	0.73	70.1	3505	1495
Condensedknn	9	2482	0.77	74.78	3739	1261