Adaptive Case Base Reasoning

Introduction to Machine Learning

University of Barcelona

December 29, 2015

Authors:
KAZANCLI, Erol
LEYVA, María

Contents

1 Adapting the parser

We adapted the parser so that it can read 'REAL' and nominal values as well as numeric ones. We had to deal with NaN values in such a way that for real values we replaced the NaN values with the mean of the attribute. We normalized the numerical and real values afterwards. We also changed the distance function to take account of the nominal values. According to our distance algorithm, if two nominal values are the same for two data points, the contribution of this dimension to the distance is zero. Otherwise, the contribution is one.

For the 10-fold cross validation we made the analysis in 10 iterations, each time taking another training and test set, which are already organized as 10-fold cross data. We averaged the accuracy, execution time and case base size in the end.

2 Maintenance Algorithms

We implemented two case base maintenance algorithms: All-k-nn and IB2

2.1 All-k-nn

Edited Nearest Algorithm is an algorithm in which, cases correctly classified by the majority of the k-nearest neighbours are kept for the case base, and the rest of the data points, which are wrongly classified, are discarded. In all-k-nn algorithm we apply edited nearest neighbour algorithm six times, each time increasing the k (from 1 to 6). With this process we are removing the wrongly classified data points from the case base, thus getting rid of the noise and outliers, and smoothing the decision boundaries. We chose this algorithm, because after this process new data should be better classified in the absence of noisy data. We run the algorithm for all the folds of training data of the hypothyroid and kr-vs-kp data sets.

The result of the analysis:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Original size	3397	3394	3395	3395	3394	3394	3394	3394	3395	3396
Obtained size	3054	3043	3050	3051	3045	3060	3059	3061	3047	3024
% reduction	10.10	10.34	10.16	10.13	10.28	9.84	9.87	9.81	10.25	10.95
Execution time	152.52	151.83	150.82	149.63	147.74	151.41	153.98	150.81	150.43	148.84

Table 1: Results on using All-K-NN with the Hypothyroid set and its different training folds

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Original size	2875	2876	2877	2877	2876	2877	2876	2877	2878	2875
Obtained size	2521	2506	2518	2521	2518	2514	2507	2499	2509	2510
% reduction	12.31	12.87	12.48	12.37	12.45	12.62	12.83	13.14	12.82	12.70
Execution time	108.86	109.46	110.97	109.52	108.79	109.00	111.48	108.96	115.73	139.85

Table 2: Results on using All-K-NN with the Kr-vs-Kp set and its different training folds

As can be seen the case base is drastically reduced, a decrease of around 10 percent for the first dataset and 12-13 for the second one. However, as we can see from the execution time, this process is quite time costly, because it involves taking the distance from each point to each point, and then sorting these distances to find the k nearest neighbours. And we are repeating this process 6 times.

2.2 IB2

The other chosen Case-Base Maintenance Algorithm chosen belongs to another family, the Instance-Based Learning family. IB2 is basically a modified 1-Nearest-Neighbor algorithm, which basically adds to the Case-Base set a sample when this is not correctly classified by the 1-NN.

For this, we used the same two datasets as with All-k-NN and obtained the following results for the different folds:

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Original size	3397	3394	3395	3395	3394	3394	3394	3394	3395	3396
Obtained size	539	525	533	538	546	525	533	520	534	611
% reduction	84.13	84.53	84.30	84.15	83.91	84.53	84.30	84.68	84.27	82.01
Execution time	20.20	18.93	20.73	20.94	21.14	20.61	20.01	20.05	19.67	21.80

Table 3: Results on using IB2 with the Hypothyroid set and its different training folds

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Original size	2875	2876	2877	2877	2876	2877	2876	2877	2878	2875
Obtained size	268	247	240	236	232	227	227	228	229	228
% reduction	90.68	91.41	91.66	91.80	91.93	92.11	92.11	92.08	92.04	92.07
Execution time	12.51	11.35	11.96	11.08	10.94	10.89	8.90	9.76	12.16	11.88

Table 4: Results on using IB2 with the Kr-vs-Kp set and its different training folds

As we can observe, we get a very significant reduction of the training data set, around 84% for the first dataset and more than 90% for the other. Also, this algorithm is way faster than All-k-nn method, since it lasts almost 10 times less.

3 Adaptive Case Based Reasoning

3.1 Retrieve

At this stage the most similar cases from a case base are retrieved for a new case. We use k nearest neighbour algorithm to retrieve the k nearest data points to be used in the later stages. We used k = 3, 5 and 7

3.2 Reuse

We use a majority vote mechanism to choose a class to the new case. Therefore the majority class in the k nearest points is assigned to the new case. We do not apply the adaptation phase because we only dealt with classification problems.

3.3 Revise

We assign a goodness value of 0.5 to each point in the beginning. At the revision stage we update the goodness values for the k nearest points we have found. The goodness value is increased by a learning rate (0.1 in our implementation) for the majority class, which is the class for the new case. The goodness values are decreased for the other k nearest neighbours. These goodness values are going to be used in the oblivion stage. The oblivion stage is also considered a part of revision process, therefore we implemented it at this stage. For the oblivion we check the goodness values of the k nearest data points. If the goodness value is below the initial goodness value, which is 0.5, then the point is discarded, in other words forgotten.

3.4 Retain

We implemented Degree of Disagreement Retention Strategy (DD), Minimal Goodness Retention Strategy(MG), Always Retain Strategy(AR) and Never Retain Strategy(NR) for retention.

3.4.1 Degree of Disagreement Retention Strategy (DD)

Degree of Disagreement retention is an unsupervised strategy based of disagreement among the k-nearest neighbours. Degree of disagreement is calculated according to the equation

$$d = \frac{remainingcases}{(classes - 1) * majoritycases}$$

where remainingcases is the number of cases in the k-nearest neighbour which do no not have the majority class, classes is the number of classes in the case base and majoritycases is the number of cases in the k-nearest neighbours with the majority class. If the disagreement is bigger than a threshold, the case is retained because it is thought that it brings valuable information to resolve future cases. If d is smaller than the threshold than this means it will not bring additional information to the case base. In our case threshold = 0.2

3.4.2 Minimal Goodness Retention Strategy (MG)

Minimal goodness retention is an unsupervised retention strategy, where the maximum goodness of the majority class in the k-nearest neighbours is compared to a threshold, that is the average of the maximum and minimum goodness of the same class in the whole casebase. If this value is found less than the threshold the case is kept and assigned an initial goodness value of 0.5. Otherwise it is discarded. This strategy aims to increase the diversity among the low-goodness groups and restrict the growth of high-goodness groups, which are well-adapted.

3.4.3 Always Retain Strategy (AR)

In this retention strategy every new case is retained and given an initial goodness value of 0.5

3.4.4 Never Retain Strategy (NR)

In this retention strategy every new case is discarded.

4 Testing the Algorithm

We tested the adaptive case base reasoning algorithm with different retention strategies without or combined with oblivion for three different k = 3, 5, 7. We used ten-fold cross validation using two different large enough data sets containing numeric and nominal values and averaged the accuracies, execution times and case base sizes obtained from each cross validation.

4.1 Hypothyroid

The accuracy rates, the execution times and the case base sizes can be seen in the tables below.

	DD	MG	AR	NR	DDO	MGO	ARO	NRO
k=3	0.9316	0.9327	0.9321	0.9316	0.9332	0.9332	0.9319	0.9332
k=5	0.9303	0.9316	0.9300	0.9316	0.9311	0.9311	0.9295	0.9311
k=7	0.9305	0.9311	0.9295	0.9311	0.9300	0.9300	0.9287	0.9300

Table 5: Accuracy rates with different retention strategies and k

	DD	MG	AR	NR	DDO	MGO	ARO	NRO
k=3	31.7310	32.0957	33.6466	31.0536	30.4261	31.0394	32.1180	30.9280
k=5	30.6446	31.7261	32.4270	30.5530	30.3915	31.1701	31.4526	30.1374
k=7	31.2897	32.2883	32.3688	31.0667	30.5722	30.5837	33.3191	31.5629

Table 6: Execution Times with different retention strategies and k

	DD	MG	AR	NR	DDO	MGO	ARO	NRO
k=3	3395	3504	3772	3394	3372	3512	3754	3372
k=5	3405	3483	3772	3394	3359	3473	3742	3359
k=7	3401	3464	3772	3394	3345	3440	3731	3345

Table 7: Case Base Sizes with different retention strategies and k

As can be seen from the tables above three retention strategies with oblivion, namely DDO, MGO and NRO have the best accuracy rates with k = 3. When we check the execution times, the best performance belongs to DDO with k = 3. As for the case base, we see the case base did not increase much for the DD, and it decreased for the DDO due to oblivion. This might be because of the nature of our data in which there was not much disagreement between the k-nearest neighbours for most of the cases. We see the biggest increase for AR, which is expected. For MG there is a substantial increase in the case base size and it also has the best accuracy among the non-oblivion methods. And as expected we see the case base sizes decreased for the oblivion strategy NRO. The case base increased for the oblivion strategies MGO and ARO since the oblivion rate was smaller than the addition from the new cases. Comparing the accuracies and the execution times, the winner for this data set is DDO, although this is due more to the oblivion than the DD part. From this data set we clearly see that oblivion strategy works to improve the accuracy but may add additional execution time.

We compared the performances of IB2, allknn(first processing the data and applying NR) and alknn with DDO using k = 3, which was the best performing algorithm in the previous analysis. As can be seen from the Table 4, IB2 performed quite badly for our data set, probably the boundary

	Accuracy	Exec time	Case Base Size
IB2	0.76385	5.6583	540
IB2 with DDO	0.84095	5.1407	458
Allknn	0.93081	31.4779	3049
Allknn with DDO	0.93081	30.8542	3047

Table 8: Comparison of Maintenance Algorithms and DDO with Allknn and IB2

data points are not enough to represent the data due to the nature of our specific data. It can be seen that the case base size is dramatically reduced for IB2. We see that applying All knn before DDO algorithm reduced the performance of DDO (see Table1 and Table4) slightly, probably because in DDO a similar selection process is already applied by oblivion.

4.2 Kr-vs-Kp

	DD	MG	AR	NR	DDO	MGO	ARO	NRO
k=3	0.9581	0.9578	0.9584	0.9578	0.9565	0.9568	0.9584	0.9565
k=5	0.9653	0.9653	0.9650	0.9653	0.9618	0.9628	0.9625	0.9618
k=7	0.9621	0.9625	0.9621	0.9625	0.9553	0.9549	0.9556	0.9553

Table 9: Accuracy rates with different retention strategies and k

	DD	MG	AR	NR	DDO	MGO	ARO	NRO
k=3	22.9155	22.5724	23.7740	22.4462	22.3204	22.4948	23.8525	22.5513
k=5	23.5834	23.2561	23.9792	23.3970	23.0669	24.0632	25.0111	24.2672
k=7	24.3910	23.5488	24.9622	23.1161	21.5022	22.4401	23.2154	21.3417

Table 10: Execution Times with different retention strategies and k

	DD	MG	AR	NR	DDO	MGO	ARO	NRO
k=3	2962	2817	3196	2876	2792	2904	3112	2791
k=5	3012	2882	3196	2876	2711	2794	3024	2705
k=7	2962	2879	3196	2876	2617	2643	2930	2609

Table 11: Case Base Sizes with different retention strategies and k

For this dataset we can observe that the best K value seems to be 5, instead of 3, which was the one we were using for the former dataset.

The best methods here seem to be *MG* and *NR*, which get the best accuracies, with 96%.

Using this parameters, we test the performance with the Maintenance Algorithms considered before. The results can be seen in the following table.

	Accuracy	Exec time	Case Base Size
IB2 NR	0.51693	2.1531	236
IB2 MG	0.51412	2.2253	240
Allknn NR	0.92333	21.6296	2512
Allknn MG	0.92333	21.8729	2515

Table 12: Comparison of Maintenance Algorithms and NR and MG

Here we can observe a huge loss of performance for the IB2 maintenance algorithm. It is much faster, but, on the other hand, it has 40% less accuracy. In this case, it is not recommendable to use it.

5 Feature Selection Before ACBR

We tried two methods for feature selection, ReliefF which is a built-in method in Matlab and correlation-based feature selection, which we implemented based on a paper titled 'Feature Selection Based on Mutual Correlation' (see ref. no.6). According to this algorithm we take the correlation between all attributes given the training set, and delete the one with the biggest average correlation. This step is executed until a desired number of features are obtained (each time taking the correlations again). We kept 10 out of 28 attributes for each method for our hypothyroid data set we used before. The performances of each method can be seen in Table 13. We see that relieff method offers an improvement on the previous DDO with k = 3 without feature selection(see table 5). However, for this data set we were not able to see very clearly which algorithm performed better. We tried with the kr-vs-kp data reducing the number of attributes from 35 to 10, which is a significant drop and with an MG retention mechanism. We see from Table 14 that with this data, besides offering a better performance than with all attributes used, relieff with 10 attributes performed much better than correlation based feature selection with 10 attributes.

	Accuracy	Exec time	Case Base Size
RELIEFF	0.94273	36.5842	3376
CORRELATION	0.92975	12.1661	3382

Table 13: Comparison of Feature Selection Algorithms using DDO with k = 3(hypothyroid data)

	Accuracy	Exec time	Case Base Size
RELIEFF	0.9687	26.5822	2893
CORRELATION	0.6858	9.58	2877

Table 14: Comparison of Feature Selection Algorithms using MG with k = 5(kr-vs-kp data)

6 Running the code

In order to run all our code, you can execute the W2 script included. This will interactively run all the scripts corresponding to all the exercises(namely ReadDatasetsScript, MaintenanceAlgorithmScript, ACBRscript, MaintainedACBRScript, WeightedACBRScript) showing in console the data this assignment asked for.

All this scripts can be executed separately, since the data are stored in .mat files, in order to make the execution faster and not to repeat executions unnecessarily.

Please note that the execution of each exercise block may take some time because two files for eight retention types over ten iterations are processed. If the execution time needs to be reduced the first ReadDataScript step may be skipped because the 10-fold data has already been read into mat files, and additionally one of the file executions can be commented out in each main exercise script. After each exercise block, the program will close all the windows and clear the variables and the command line.

References

[1] Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In Proceedings of the International Joint Conferences on Articial Intelligence IJCAI-95. (1995).

[2] Demsar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7 (December 2006), 1-30. (2006)

[3] Retrieval, reuse, revision and retention in case-based reasoning The Knowledge Engineering Review, Vol. 20, No. 03. (2005), pp. 215-240, doi:10.1017/S0269888906000646 by Ramon de Mantaras, David Mcsherry, Derek Bridge, et al.

[4] Overview of Maintenance for Case-Based Reasoning Systems. Abir Smiti and Zied Elouedi. International Journal of Computer Applications, Vol. 32, No. 2, 2011

[5] Adaptive case-based reasoning using retention and forgetting strategies. Knowledge-Based Systems. 24, 2 (March 2011), 230-247 by M. Salamó and M. López-Sánchez. 2011. DOI=10.1016/j.knosys.2010.08.003 http://dx.doi.org/10.1016/j.knosys.2010.08.003

[6] Feature Selection Based on Mutual Correlation. Michal Haindl, Petr Somol, Dimitrios Ververidis, and Constantine Kotropoulos http://muscle.ercim.eu/images/DocumentPDF/MP_473_FS.pdf