Introduction to Machine Learning
Work 2
Case-based reasoning exercise

Contents

1 Case-based reasoning exercise

1.1 Introduction

In this exercise you will learn about case-based reasoning and feature selection. You will apply these techniques to a classification task. It is assumed that you are familiar with the concept of cross-validation. If not, you can read this paper:

R. Kohavi. 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.

For the validation of the different algorithms, you need to use a T-Test or a Friedman Test in conjunction with a Nemenyi test, or another statistical method. Next reference is a **reading proposal** on this topic:

Janez Demšar. 2006. Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7 (December 2006), 1-30.

The above article details how to compare two or more learning algorithms with multiple data sets.

1.2 Methodology of the analysis

As in the previous work assignment, you will analyze the behavior of the different algorithms by comparing the results in a pair of well-known data sets from the UCI repository. In that case, you will also use the class as we are testing several supervised learning algorithms. In particular, in this exercise, you will receive the data sets defined in .arff format but divided in ten training and test sets (they are the *10-fold cross-validation* sets you will use for this exercise).

This work is divided in several steps:

1. Improve the parser developed in Work 1 in order to use the class attribute, too. You should write a function in MatLab called `parser_arff_file('filename.arff')` that saves the information from a training or testing file in a `TrainMatrix` or `TestMatrix`. You need to normalize all the numerical attributes in the range [0..1]. Next you have an example of how to normalize one attribute of your data and how to get your original data back:

```
bla = 100.*randn(1,10)

minVal = min(bla);
maxVal = max(bla);

norm_data = (bla - minVal) / ( maxVal - minVal )
your_original_data = minVal + norm_data.*(maxVal - minVal)
```

2. Write a MatLab function that automatically repeats the process described in previous step for the 10-fold cross-validation files. That is, read automatically each training case and run each one of the test cases in the selected classifier.

3. Write a MatLab function for classifying each instance from the `TestMatrix` using the `TrainMatrix` to a classifier called `cbrAlgorithm(…)`. You decide the parameters for this classifier. This CBR function will call each one of the phases of the CBR. Write a function for each one of the phases of the CBR. **Justify your implementation and add all the references you have considered for your decisions.** An interesting paper that details quite well the CBR cycle is:

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.

   a. For the retrieval, `cbrRetrievalPhase()` function, you should consider the Euclidean distance and a new similarity metric, not the Minkowski with a different `r` parameter. Look at the bibliography and choose another one. Assume that the retrieval phase returns the `K` most similar instances (i.e., also known as cases) from the `TrainMatrix` to the `current_instance`.

   b. For the reuse, `cbrReusePhase()`, you may decide a policy. For example, you may consider to use the most similar retrieved case or alternatively to use a voting policy. The majority class in all the examples will be used to provide a solution to the `current_instance`.

   c. For the revision, `cbrRevisionPhase()`, you do not need to implement it for the CBR purpose. The revision phase will serve you for analyzing if the algorithm has correctly solved the new problem (i.e., `current_instance`). It will be used for the evaluation of the algorithm. In this phase you can store your results in a memory data or in a file.

   d. For retaining, `cbrRetentionPhase()`, you will implement three policies. First policy retains all the new solved cases, and you will store the solution with the real one. This approach will be called `cbrFullRetentionPhase()`. Second policy, `cbrNoRetentionPhase()`, will never retain a case, independently of the revision. Finally, third policy is free. You decide the implementation by looking at the bibliography. For example, a simple policy is to use the class information to decide if a case is stored or not. When the case has not been solved correctly, the new case will be stored in the case base.

At the end, you will have a CBR with two retrieval functions (Euclidean and the one you have considered), a reuse function (that you decide) and three retention policies (2 simple ones and another one selected by you). You should analyze the behavior of these functions in the CBR algorithm and decide which combination results in the best CBR algorithm. Extract conclusions by analyzing **two large** enough data sets. At least one of these data sets will contain numerical and nominal data.

4. Modify the retrieval similarity function in the CBR so that it implements a weighted function called `weightedCBRRetrievalPhase()`. Each weight denotes if an attribute is considered or not for the similarity. A weight value of 1.0 denotes that the attribute will be used by the similarity. By contrast, a weight value of 0.0 shows that the attribute is useless and it is not going to be used.

   The weights can be extracted using a weighting metric or a feature selection algorithm. You may choose **two algorithms** (filter or wrapper, as you wish). Use them as a preprocessing step. For example, you can use ReliefF, Information Gain, or the Correlation, among others. There are several MatLab toolboxes that also include most of the well-known metrics for feature selection algorithms. You can use the implementations that exist in MatLab for your feature selection implementations.

The schedule for these steps is as follows:

- Week 1. Steps 1, 2
- Week 2. Steps 3, 4

1.3 Work to deliver

In this work, you will implement a case-based reasoning and a feature selection algorithm. You may select 2 data sets (large enough to extract conclusions) for your analysis. At the end, you will find a list of the data sets available.

You will use your code in MatLab to extract accuracy results. The accuracy measure is the average of correctly classified cases. That is the number of correctly classified instances divided by the total of instances in the test file. For the evaluation, you will use a T-Test or Friedman algorithm.

From the accuracy results, you will extract conclusions showing graphs of such evaluation and reasoning about the results obtained.

In your analysis, you will include several considerations.

1. You will analyze the CBR (with no weighting or selection). You will analyze which is the most suitable combination of phases for the CBR. The one with the highest accuracy. This CBR combination will be named as standard CBR.

2. Once you have decided the best CBR combination. You will analyze it in front of using this combination with two feature selection algorithms. The idea is to extract conclusions of which feature selection algorithm is the best one.

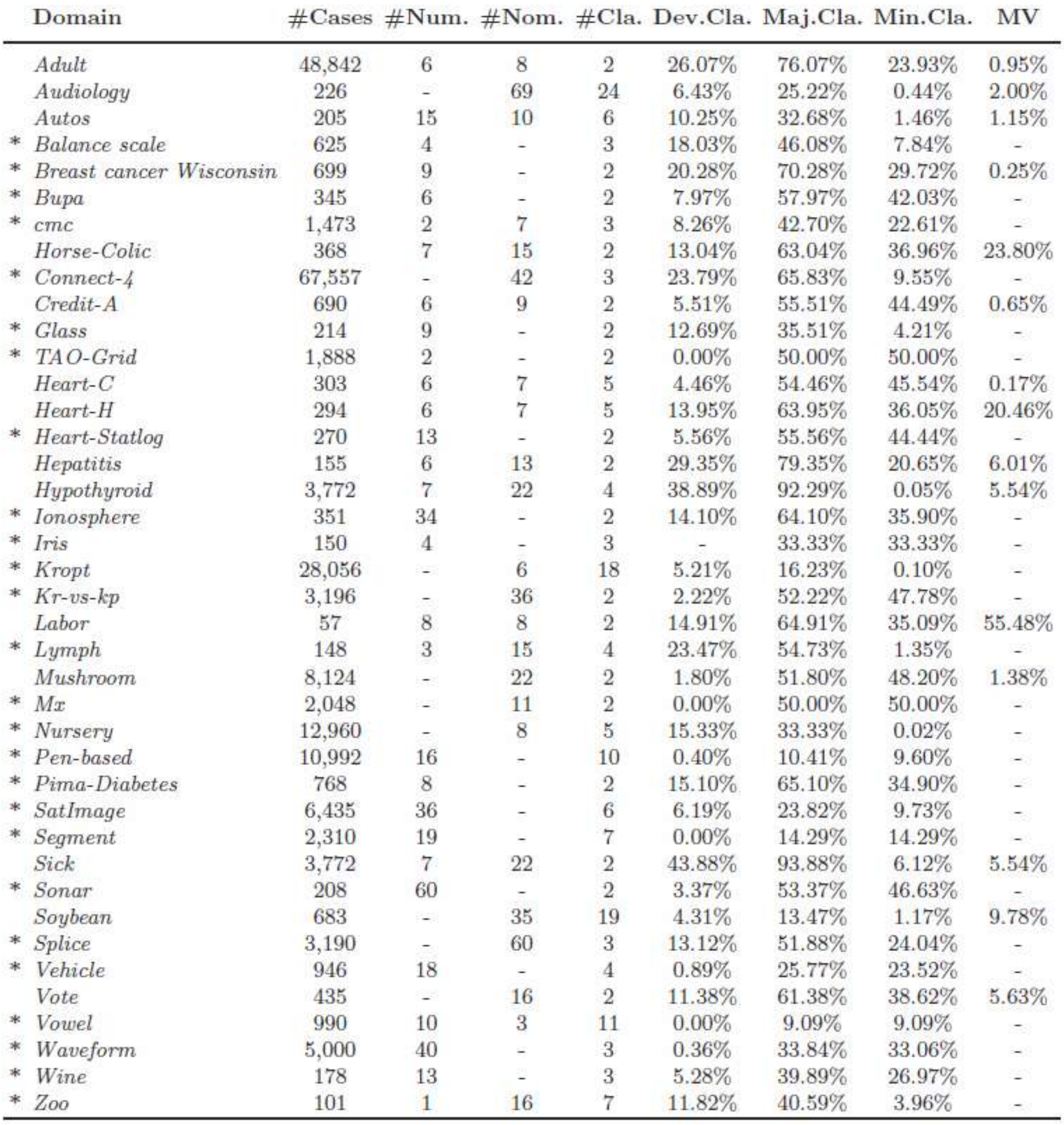
   For example, some of questions that it is expected you may answer with your analysis:

- Which is the best value of K for the retrieval and reuse phase?
- Which is the best similarity metric, among those implemented?
- Did you find differences among the standard CBR and the CBR that also uses feature selection?

- According to the data sets chosen, which feature selection algorithm provides you more advice for knowing the underlying information in the data set?
- In the case of the feature selection CBR, how many features where removed? Which are the features selected for each one the feature selection algorithms?
- Which criterion have you used to decide the features that should be removed?

Apart from explaining your decisions and the results obtained, it is expected that you reason each one of these questions along your evaluation. Additionally, you should explain how to execute your code. Remember to add any reference that you have used in your decisions.

You should deliver a word or pdf document as well as the code in MatLab in Racó at UPC by November, 11th, 2013.



	Domain	#Cases	#Num.	#Nom.	#Cla.	Dev.Cla.	Maj.Cla.	Min.Cla.	MV
	Adult	48,842	6	8	2	26.07%	76.07%	23.93%	0.95%
	Audiology	226	-	69	24	6.43%	25.22%	0.44%	2.00%
	Autos	205	15	10	6	10.25%	32.68%	1.46%	1.15%
*	*Balance scale*	625	4	-	3	18.03%	46.08%	7.84%	-
*	*Breast cancer Wisconsin*	699	9	-	2	20.28%	70.28%	29.72%	0.25%
*	*Bupa*	345	6	-	2	7.97%	57.97%	42.03%	-
*	*cmc*	1,473	2	7	3	8.26%	42.70%	22.61%	-
	Horse-Colic	368	7	15	2	13.04%	63.04%	36.96%	23.80%
*	*Connect-4*	67,557	-	42	3	23.79%	65.83%	9.55%	-
	Credit-A	690	6	9	2	5.51%	55.51%	44.49%	0.65%
*	*Glass*	214	9	-	2	12.69%	35.51%	4.21%	-
*	*TAO-Grid*	1,888	2	-	2	0.00%	50.00%	50.00%	-
	Heart-C	303	6	7	5	4.46%	54.46%	45.54%	0.17%
	Heart-H	294	6	7	5	13.95%	63.95%	36.05%	20.46%
*	*Heart-Statlog*	270	13	-	2	5.56%	55.56%	44.44%	-
	Hepatitis	155	6	13	2	29.35%	79.35%	20.65%	6.01%
	Hypothyroid	3,772	7	22	4	38.89%	92.29%	0.05%	5.54%
*	*Ionosphere*	351	34	-	2	14.10%	64.10%	35.90%	-
*	*Iris*	150	4	-	3	-	33.33%	33.33%	-
*	*Kropt*	28,056	-	6	18	5.21%	16.23%	0.10%	-
*	*Kr-vs-kp*	3,196	-	36	2	2.22%	52.22%	47.78%	-
	Labor	57	8	8	2	14.91%	64.91%	35.09%	55.48%
*	*Lymph*	148	3	15	4	23.47%	54.73%	1.35%	-
	Mushroom	8,124	-	22	2	1.80%	51.80%	48.20%	1.38%
*	*Mx*	2,048	-	11	2	0.00%	50.00%	50.00%	-
*	*Nursery*	12,960	-	8	5	15.33%	33.33%	0.02%	-
*	*Pen-based*	10,992	16	-	10	0.40%	10.41%	9.60%	-
*	*Pima-Diabetes*	768	8	-	2	15.10%	65.10%	34.90%	-
*	*SatImage*	6,435	36	-	6	6.19%	23.82%	9.73%	-
*	*Segment*	2,310	19	-	7	0.00%	14.29%	14.29%	-
	Sick	3,772	7	22	2	43.88%	93.88%	6.12%	5.54%
*	*Sonar*	208	60	-	2	3.37%	53.37%	46.63%	-
	Soybean	683	-	35	19	4.31%	13.47%	1.17%	9.78%
*	*Splice*	3,190	-	60	3	13.12%	51.88%	24.04%	-
*	*Vehicle*	946	18	-	4	0.89%	25.77%	23.52%	-
	Vote	435	-	16	2	11.38%	61.38%	38.62%	5.63%
*	*Vowel*	990	10	3	11	0.00%	9.09%	9.09%	-
*	*Waveform*	5,000	40	-	3	0.36%	33.84%	33.06%	-
*	*Wine*	178	13	-	3	5.28%	39.89%	26.97%	-
*	*Zoo*	101	1	16	7	11.82%	40.59%	3.96%	-