2014 5th International Conference on Information and Communication Systems (ICICS)

The Influence of Class Order on Rule Induction Results

Christina B. Class
Department of Computer Science
German Jordanian University
Amman, Jordan
cclass@acm.org

Abstract— **The basic sequential rule induction algorithm does not define a specific class order. Class order is generally selected based on heuristics. We have implemented the basic rule induction algorithm together with several discretization and evaluation methods and varying class orders. Tests in several basic data sets show that the class order considerably influences the resulting accuracy of sequential rule induction. While the class order sometimes has a higher influence than the tested rule quality measures (FOIL and Likelihood Ratio) and the simple discretization algorithms, there is no single class order that provides best results for all tested combinations of discretization and rule quality measure approaches in all cases.**

Keywords— rule induction;class order; attribute discretization

I. INTRODUCTION

Rule induction has been discussed for many years. Many different algorithms have been described in the literature and compared using different data sets. While rule induction is quite straight forward for nominal and ordinal attributes, numerical attributes pose the additional difficulty that they must be discretized. Several discretization methods have been proposed and studied formally (e.g. [1] and [2]).

Many rule induction algorithms based on a sequential covering algorithm have been proposed. They differ in different points like:

- Identification of attribute/value pairs for numerical attributes
- Selection measure for comparing the rule quality
- Termination condition
- Existence of a pruning step

Numerous modifications have been proposed and compared to other algorithms using different statistical tests [3]. The modifications tackled one or more steps of the basic algorithm and could improve specific quality measures for the classifier (like e.g. accuracy). While those modifications helped generating reasonable classifiers, it is not exactly clear to which extend *specific individual changes* improved the quality of rule classification.

In order to understand the influence of single changes to the basic rule induction algorithm, we have implemented a basic rule induction framework in R[1]. The main purpose of our framework is not in defining a new classifier or replacing existing optimized implementations. Our framework allows for parameterization to study, among others, the influence of the class order as well as of different quality measures and attribute discretization as well as on the quality of the rule induction and its performance. It is flexible and allows for easily exchanging or adding new elements to the rule induction algorithm. Tests can be started and run with few lines of code. All results are stored for future analysis. It can, therefore, be used to run extensive tests on different data sets to evaluate the influence of different parameters on the rule induction quality.

As it is quite easy to run different tests for different combinations of influencing factors, we have run several extensive tests on sample data sets. As part of the tests we have reversed the order in which the different classes have been taken into account for rule induction and have observed that for some combinations of rule quality measures and discretization the reverse class order led to much better or much worse results. E.g. in the Iris data set we reached an accuracy of 86% for the original class order and of 98% for the reverse class order for the same discretization (6 equal width bins) and evaluation (Foil Gain) method. In order to study this effect closer, we have run our tests on all six different class orders of the Iris, CMC and the Wine data set, having three classes, and on the two class orders on the Pima Indian Diabetes data set with 2 classes.

In this paper we will shortly present our framework we used for testing. We will then present some results and the observed differences in detail and equally present values from the confidence interval estimation using Student's t-test. We will present performance versus accuracy results before drawing a first conclusion.

II. THE RULE INDUCTION FRAMEWORK AND ITS PARAMETRIZATION

The implemented framework is based on the basic sequential covering algorithm for rule induction and does not include additional steps like a pruning step. By testing and adding more and more of the specific features and algorithms (e.g. for attribute discretization) we will gradually be able to

[1] http://cran.r-project.org/

This research was supported by an internal research grant from German Jordanian University.

978-1-4799-3023-4/14/$31.00 ©2014 IEEE

observe and study the effect of numerous combinations on the quality of the produced classifier.

The framework allows for easily storing all relevant information, like the used training and test sets, the induced rules as well as statistical information on errors in the training and the test set. This allows for running different rule induction algorithms on exactly the same training and test sets in the future. The stored data allow for analysis using a large variety of different statistical evaluation methods on the obtained results.

The basic sequential covering algorithm applies a brute force approach and is therefore costly. Among others, the costs base on the number of split points, as rules might be created and tested for all possible split points. The number of split points depends on the discretization method used. An interesting question is the relationship of the quality of a rule induction algorithm and its performance. As indicator for the performance we count the number of different rules that are tested during the rule induction. Other performance indicators might be added later on.

Our basic rule induction framework supports conjunctive rules only.

In our current implementation of the framework in R we have parameterized several relevant elements for rule induction:

- The order of the classes for the main loop of sequential rule induction.
- Operators that are used for numerical attribute value pairs. We support the 6 operators <,≤ =,≠,>, and≥.
- the categorical as well as numerical attributes (dimensions) to be included
- The discretization method for numerical attributes including parameters. We equally allow for several discretization functions to be applied for the same attribute. The union of all derived split points will be used during rule induction.
- The function to use for rule quality measurement as well as the threshold to be applied.
- The minimum percentage of true positives a rule must cover to be included in the rule set. If a rule is below this threshold, induction for the current class will be stopped.

The current implementation supports so far the following discretization functions:

- equal frequency and equal width bins as simple unsupervised discretization methods. [4]
- oneR: the simple discretization method that was implemented in the 1R* algorithm [5] as an example for a simple supervised method.

As rule quality measurement function we have implemented

- FOIL Gain using the formula as described in [6]
- Likelihood Ratio equally using the formula described in [6]. We assume a two class model (member of the class or not member of the class) to simplify the calculation.

In order to generate human readable results we have implemented several methods that generate verbalized output.

The framework provides in addition supporting functions:

- to create a list of 10 folds of a specified size out of a data set
- to create a test and a training data set based on a folds list. An index is specified as parameter which determines which of the 10 folds is used as test set.
- to run the tests. This function receives a list of 10 fold sets (each containing 10 folds) and the relevant parameters for calling the rule induction function. It runs 10 rounds of 10-fold cross-validation and returns the results. This function allows for parameterization of the method to create the initial class list (e.g. natural order, reverse order, etc.).

All intermediate results (after each fold and after each round) are stored so that the rule induction can be interrupted and resumed later on.

III. Application of the Rule Induction Framework for Testing and Evaluation

As mentioned before, the main target of our rule induction framework lies in the possibility to flexibly change elements of rule induction and measure the quality of the classifier in terms

TABLE I. Definition of the Basic Tests

Test Nb.	discretization	parameter	rule quality measure
1	equal width bins	6 bins	FOIL
2			Likelihood Ratio
3		8 bins	FOIL
4			Likelihood Ratio
5	equal frequency bins	6 bins	FOIL
6			Likelihood Ratio
7		8 bins	FOIL
8			Likelihood Ratio
9	equal widths and equal frequency bins	each 6 bins	FOIL
10			Likelihood Ratio
11		each 8 bins	FOIL
12			Likelihood Ratio
13	1R		FOIL
14			Likelihood Ratio

of accuracy. We equally have recorder the number of tested rules as an indicator for the performance of the rule induction algorithm. For each defined test, we have run 10 rounds of different tests on different data sets. For each round we randomly split the data set in 10 folds and carry out the 10-folds cross validation. The 10 folds are saved to allow for testing rule induction with different parameters and discretization methods on the same folds for all the different tests.

We have defined 14 different basic tests as described in Table I.

Rule induction for a specific class was stopped:

- when no tuple belonging to the class was left in case of FOIL gain
- when no tuple belonging to the class was left or when the inducted rule did not have at least a ratio of 50% true positives covered in case of Likelihood ratio. To apply the statistics for Likelihood ratio we have chosen a threshold of 5.

For all tests we have stored the complete data including:

- used folds
- inducted rule sets
- confusion matrices
- statistics including the errors in the training set (90 % of the data) and the test set for each rule induction as well as the number of rules tested to induct this set of rules. In order to estimate the error rate for the given classifier we add the number of errors in each cross-validation fold test.

All tests have been run on the same folds to achieve comparability.

We have used four different data sets that have all been downloaded from [7]:

- The *Iris data* set is one of the most well-known data sets and consists of 150 tuples with 4 attributes. It has 3 classes and in the data set the classes are equally distributed. For the Iris data set we have created ten sets of ten folds each containing 5 tuples of all three classes. We have, thus, created stratified folds.
- The *Wine* data set equally has three classes. Its tuples have 13 different numerical attributes. We did not normalize the attribute values. 59 tuples belong to class 1, 71 to class 2 and 48 to Class 3. We have created stratified folds by selecting values with replacement for the lower frequency classes. Equally for the last two folds of class 1 we have run selection with replacement to create folds of equal size.
- The *CMC* data set (contraceptive method choice) has 2 numerical and 7 categorical attributes. 629 tuples belong to class 1, 333 to class 2 and 511 to class 3. For CMC we have equally created stratified folds by using selection with replacement. Each fold contains 63 tuples of each of the three classes.
- For the Pima Indian *Diabetes* data set we have removed all 0 values. Removing values of 0 (except for the number of pregnancies) was proposed in [8] as those 0 values are physiologically impossible. After removing those data tuples, 392 tuples remained. 262 of the tuples belong to the class "tested negative" while 130 tuples have the class "tested positive". We did not stratify the folds for the diabetes data set. For each set of folds we have created 10 folds containing 40 tuples using selection with replacement.

As initial validation of our rule induction results, we compare the obtained accuracy results to results from two literature references [8] and [9] as depicted in TABLE II.

We have run all fourteen tests on all possible class orders for all data sets. In each test we have run 10 times a 10-fold cross validation. For each specified class order we have thus run 1400 tests. The best and worst test accuracy that we could reach for the simple rule induction algorithm implementation are depicted in TABLE III.

For all data sets we could achieve a set of inducted rules that has an accuracy of the expected range. In all data sets we equally have achieved sets of rules with much lower accuracy. All tests have been based on the *same* set of folds using the same implementation of the rule induction algorithm. We *only* have changed the discretization method used (equal width and equal frequency bins, 1R), the rule quality measure (FOIL and Likelihood ratio) as well as the class order.

In the following sections we present some of the results related to different class orders in more detail.

IV. Test Results: Accuracy

As an example we present the results for the Iris data set. We choose this data set here as the used folds have been stratified and no replacement has been applied when selecting the data for the folds.

The Iris data set has three classes. TABLE VI. depicts the minimum and maximum achieved accuracy for each type of test and the class order that has achieved these results. 1 refers to Iris setosa, 2 to Iris versicolor and 3 to Iris virginica. The test number refers to the numbers specified in TABLE I.

While the class orders 213 and 231 in the Iris data set have always low results, the class order 132, e.g., has low (64%) to

TABLE II. Expected Accuracy Values

data set	expected accuracy according to [8]	expected accuracy according to [9]
Iris		92% to 96 %
Wine		89 % to 93 %
CMC	40 % to 57 %	
Diabetes	69 % to 78 %	71% to 74%

TABLE III. Minimum and Maximum Accuracy in Tests

data set	minimum accuracy	maximum accuracy
Iris	36%	98%
Wine	31%	92%
CMC	35%	49%
Diabetes	33%	75%

TABLE VI. MINIMUM AND MAXIMUM ACCURACY FOR ALL TESTS IN THE IRIS DATA SET

test number	minimum accuracy		maximum accuracy	
	accuracy	*class order*	*accuracy*	*class order*
1	52%	213	98%	132, 312, 321
2	36%	213	94%	321
3	86%	213,231	93%	132, 312, 321
4	64%	231	96%	132,312
5	82%	213,231	97%	132
6	62%	231	93%	132, 312
7	82%	213	91%	312
8	63%	231	94%	123,132, 312
9	52%	213	98%	132, 312, 321
10	36%	213	94%	321
11	82%	213,231	87%	132
12	62%	231	93%	132, 312
13	86%	213,231	93%	132, 321
14	64%	231	96%	132, 312

excellent results (98 %). Not for all tests the same class order gives best or worst results. E.g., the class order 123 obtains results in the lower, the mid and the upper range for different tests.

In TABLE IV. we depict the highest and lowest accuracy results for all four data sets together with information about the class order and the test number.

As we can see in TABLE IV. we do not have a general result of the tests nor the general class order that produces highest accuracy values for the given data sets. E.g. in the Wine data set the best and the worst accuracy results have been produced using the same class order. We can equally not make general conclusions, which of the applied discretization methods is better. The same holds for the rule quality measure approach.

As the discretization method of 1R as described in [5] is a supervised one compared to the other applied discretization methods, one might suggest that the tests based on 1R discretization are less sensitive to class order. But as TABLE V. shows high differences in the achieved accuracy based on different class orders can also be observed when this discretization method is used. E.g. the two different class orders lead to contrary results in the Pima Indian Diabetes data set. While one class order performs well with FOIL gain as rule quality measure the other, reverse class order performs well using Likelihood ratio as rule quality measure.

TABLE IV. TESTS AND CLASS ORDER WITH MINIMUM AND MAXIMUM ACCURACY

data set	minimum accuracy		maximum accuracy	
	class order	*test*	*class order*	*test*
Iris	213	2 and 10	132,312, 321	1 and 9
Wine	321	13	321	7
CMC	this result is not clear, many tests have lowest accuracy values		213	all 7 tests with Likelihood ratio
Diabetes	12	all 7 tests with Likelihood ratio	21	2, 10 and 14

TABLE V. MINIMUM AND MAXIMUM ACCURACY VALUES FOR 1R DISCRETIZATION METHOD

data set	rule quality measure	minimum accuracy	maximum accuracy
Iris	FOIL	86%	93%
	Likelihood	64%	96%
Diabetes	FOIL	48 %	71 %
	Likelihood	33 %	75 %
CMC	FOIL	35%	36%
	Likelihood	38%	49%
Wine	FOIL	31%	80%
	Likelihood	33%	55%

V. STUDENT'S T-TEST STATISTICS RESULTS FOR CLASS ORDER DEPENDENCY

In order to validate our assumption that class order plays a relevant role in the accuracy of the basic sequential rule induction algorithm, we have calculated the Student's t-test value for a 2-sided confidence interval of 95% and a paired test. We have created lists of errors for the ten different fold sets for which a cross validation was carried out for each test. We have then applied the Student's t-test to compare the differences in the error lists between two different class orders for a given test. We have thus calculated the values for all combinations of class order for each single test. As we have run 14 tests for the Iris, Wine and CMC data set, we have retrieved 210 t-values for the 6 different classes. As the Diabetes data set has 2 classes, 14 t-values have been computed.

In the *Diabetes* data set, all t-test values are either below -16 or above 40 and therefore we can conclude beyond reasonable doubt that the number or classification errors, and thus accuracy, depends on the class order we use. Looking in detail at the 28 results of the diabetes data set (which has two classes) we can see that the class order (tested negative, tested positive) has better results using the FOIL rule quality measure whereas Likelihood ratio is better for the class order (tested positive, tested negative).

In the *CMC* data set the t-statistics values indicate a class order dependency for more than half of the class order and test combinations. In 99 cases the hypothesis that the accuracies for the given two class orders in a given test are the same could not be falsified using the t-statistics.

The different t-values for the *Wine* data set show for around 45% of the combinations a dependency of the accuracy value on the class order. For more than 50% of the tests the dependency of class orders could not be verified using the t-

statistics. The same class order with the same rule quality measure method produces the lowest and the highest accuracy.

In the *Iris* data set, 35 of the tested combinations do not have a distinguished t-value. In addition the t-statistics could not be calculated for 19 pairs (e.g. due to division by 0). As for the class order it is difficult to make conclusions based on the results. E.g. class order 123 produces one of the lowest results (38%) as well as very good results (94%) both applying the Likelihood ratio for rule quality measurement.

Our results indicate that class order indeed influences the result of rule induction. The influence itself depends on the data set, as well as on the discretization and the rule quality measure approach. The obtained results do not allow us to make a recommendation for the best class order for all given data sets.

VI. Test Results: Accuracy over Performance

The basic sequential rule induction algorithm is a brute force approach that tests all possible attribute combinations to induct rules. Brute force algorithms can be expensive and therefore it is worth to not only look at the accuracy of the classifier but equally at the costs to retrieve it. Our implementation keeps track of the number of rules that have been tested with the rule quality measure during rule induction. We use this as an indicator for the rule induction performance.

The number of tested rules depend on the number of split points for numeric attributes and thus on the used discretization method. It also depends on how fast all tuples are covered by rules. As rules are tested according to the class order, the number of tested rules equally depends on the class order.

Our data show that the best accuracy does not necessarily imply the lowest performance (i.e. highest number of tested rules). In the *Iris* data set, the test with the highest accuracy had the lowest number of tested rules, i.e. the highest performance. (See Fig. 2)

In the *Diabetes* data set the number of tested rules to achieve highest accuracy lies in the middle and is 53% of the number of tested rules for an expensive rule induction that achieves second highest accuracy of 74%. (See Fig. 1)

In the *CMC* data set, the best accuracy requires the highest number of tested rules. In the cases of highest accuracy there

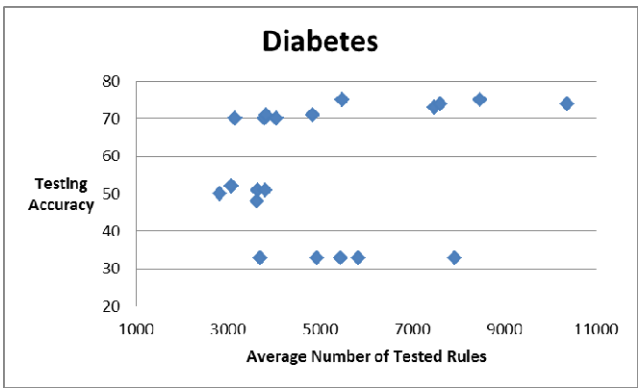

Fig. 1. Performance vs. accuracy for the Pima Indian Diabetes data set

are equally remarkable differences. One test achieving an accuracy of 49% (the highest one for our CMC rule induction) requires 80 % of the tested rules of another test with the same accuracy which is a significant performance gain. (See Fig. 3)

In the *Wine* data set, very good results could be achieved with a relatively low number of tested rules. An accuracy of 91% could be achieved with using 70% of the number of tested rules that was used for an accuracy of 92%. It is also interesting to note that the tests with the lowest accuracy had quite low performance, i.e. many rules were tested to achieve a set of rules that had low accuracy. (See Fig. 4)

Comparing accuracy and performance we can equally not draw a conclusive picture. For some data sets high accuracy can be achieved with high performance (relatively few number of rules to be tested) in other data sets the required number of tested rules to achieve high accuracy is medium or even high. Depending on the discretization method and class order, similarly high accuracy can be achieved with significantly differing performance.

VII. Summary and Conclusion

We have implemented a flexible rule induction framework for sequential rule induction to test the influence of different discretization methods, rule quality measures and class orders on the accuracy and performance of rule induction. Applying

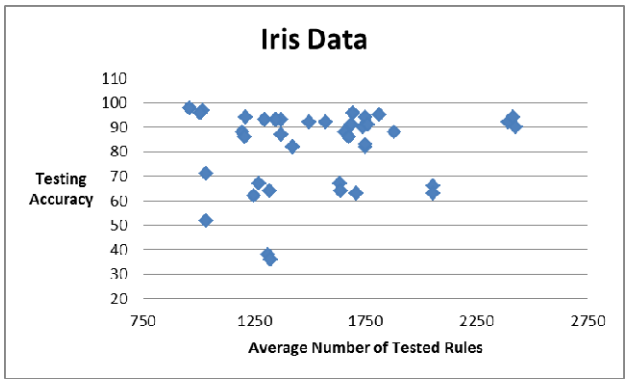

Fig. 2. Performance vs. accuracy for the Iris data set

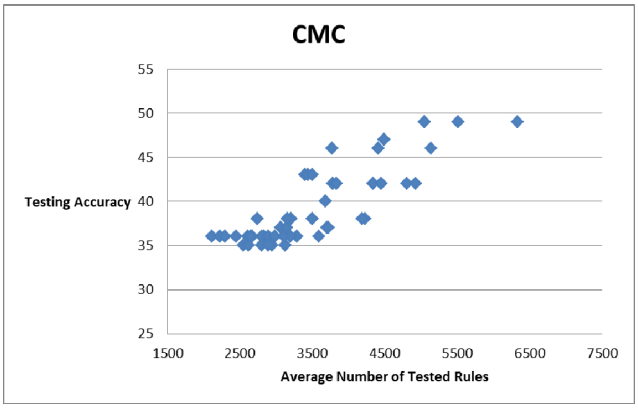

Fig. 3. Performance vs. accuracy for the CMC data set

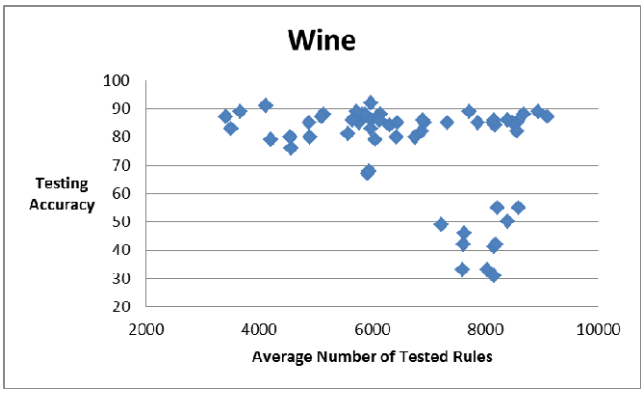

Fig. 4. Performance vs. accuracy for the Wine data set

the framework we have observed that results depend on the class order of sample data sets. In order to validate our assumption that class order influences significantly the result of rule induction, we have created a set of 14 different tests and run these tests on all different class order combinations of four well known data sets.

Using the basic rule induction approach and the simple discretization methods implemented so far in our framework, we could achieve results that were in the range of expected accuracy values. We equally have received results that were very inaccurate. We have applied the Student's t-statistics and identified in all data sets class order combinations (between 40 and 100%) that had a significant influence on the observed differences in achieved accuracy. Based on our framework we equally could compare the performance of rule induction, measured in the number of tested rules, with the accuracy. There is no general dependency, as sometimes the most accurate rule inductions have been achieved with the lowest, a medium or highest number of tested rules.

We have shown that class order has an influence on the accuracy of rule induction. But the influence is still subject to further investigation. So far there is a limited understanding on the influence of class order and heuristics are used. E.g. [10] discuss approached to determine the optimal class order for the Ripper Rule Induction algorithm.

The class order is part of different factors that influence the rule accuracy. Among them are the applied discretization method and the rule measure quality, as we could observe during our study. Together these parameters equally influence the performance of rule induction. In order to optimize rule induction it would be valuable to better understand these influences to set induction parameters and class order in an optimum way. We will continue our study by testing other well know data sets and equally including other discretization methods that are entropy based.

As one result of our study we would advise to include the applied class order(s) in all rule induction results to increase comparability, as the class order influences the accuracy of rule induction.

REFERENCES

[1] T. Elomaa and J. Rousu, "General and Efficient Multiplitting of Numerical Attributes", Machine Learning, Vol. 36, pp. 201-244, 1999.

[2] M. Boulle, "Khiops: A statistical discretization method of continuous attributes", Machine Learning, 55, 2004, pp. 53-69.

[3] J. Demšar, "Statistical Comparisons of Classifiers over Multiple Data Sets", Journal of Machine Learning Research, vol. 7, pp. 1-30, 2006.

[4] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features". In 12th International Conference on Machine Learning (ICML), pages 194–202, 1995.

[5] R. C. Holte, "Very Simple Classification Rules Perform Well on Most Commonly Used Datasets", Machine Learning 11, pages 63-91, 1993

[6] J. Han, and M. Kamber, "Data Mining Concepts and Techniques", 2nd ed., Waltham: Morgan Kaufmann Publishers, 2006.

[7] A. Frank and A. Asuncion. "UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]", School of Information and Computer Science, University of California, Irvine, 2010.

[8] T.-S. Lim, W-Y. Loh and Y.-. Shih, " A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-three Old and New Classification Algorithms", Machine Learning, Vol. 40, 2000, pp. 203-229

[9] M. Last, and O. Maimon, "A Compact and Accurate Model for Classification", IEEE Transactions on Knowledge and Data Engineering, Vol. 16, No. 2, 2002, pp. 203-215

[10] Ata, S.; Yildiz, O.T., "Searching for the optimal ordering of classes in rule induction," Pattern Recognition (ICPR), 2012 21st International Conference on , pp.1277-1280, 11-15 Nov. 2012