UCI Qualitative Bankruptcy Data Set Classification Problem

Ricardo Fusco

Universidade de Évora, Évora, Portugal
February 7, 2017
ricardo.fusco2@gmail.com

Abstract

With the current economic crisis worldwide it is only natural that a considerable number of companies end up being dragged towards bankruptcy. The information regarding bankruptcy prediction gathered by experts in this area is still considered quite important due to the subjectivity attached to their predictions. There has been quite an extensive number of studies on financial predictions namely bankruptcy prediction using financial databases, but in the other hand the number of studies that applied data mining methods and tools to study and represent experts' more subjective and qualitative opinions and decisions are a bit scarcer than other kinds of studies.

The objective here is to classify and apply data mining algorithms like rule based algorithms(Zero Rule and One Rule), Naive Bayes and Bayesian Belief Networks and Decision tree based methods(J48) to the data gathered from the experts' decisions regarding several aspects related to companies' internal risks.

Contents

1 Introduction

The chosen data set was the Qualitative Bankruptcy one [5][4], this problem is one of a social-economic nature. With the current economic crisis in so many countries it is only natural that a considerable number of companies end up being dragged towards bankruptcy, the recession still has an impact on the economy and businesses overall nowadays [2]. The information regarding bankruptcy prediction gathered by experts in this area is still considered quite important due to the subjectivity attached to their predictions.[3]

The objective here is to simulate and automatize these predictions, representing in a way the predictions that would be made by the experts. The objective here will be to solve a classification problem classifying the companies as going towards bankruptcy or not. Data classification remains as one of the most important issues for most business applications.

This data set and the parameters used for collecting it were referred in a paper by Myoung-Jong Kim and Ingoo Han [3] where genetic algorithms were used in order to do the rule extraction process, stating that the use of genetic algorithms shows indeed a better predictive accuracy than said neural networks and inductive learning methods.

2 Data

This study consist in a data set with 7 attributes (the 7th attribute is the class attribute which classifies either as bankruptcy or non bankruptcy) and 250 different instances.

Regarding the attributes there are 6 different attributes which have the same domain, each attribute can hold one of 3 possible values, positive (P), average (A) and negative (N) and the prediction/classification will be either as bankruptcy (B) or non-bankruptcy (NB). The 6 attributes present in the data are the companies' internal risks, Industrial Risk (IR), Management Risk (MR), Financial Flexibility (FF), Credibility (CR), Competitiveness (CO), and Operating Risk (OP). Details about the risk factors can be found on the next table 1.

Risk factor	Variables	Risk components
Industry risk	IR	Government policies and International agreements
		Cyclicality
		Degree of competition
		The price and stability of market supply
		The size and growth of market demand
		The sensitivity to changes in macroeconomic factors
		Domestic and international competitive power
		Product Life Cycle
Management risk	MR	Ability and competence of management
		Stability of management
		The relationship between management/owner
		Human resources management
		Growth process/business performance
		Short and long term business planning, achievement and feasibility
Financial Flexibility	FF	Direct financing
		Indirect financing
		Other financing (Affiliates, Owner, Third parties)

Credibility	CR	Credit history
		The reliability of information
		The relationship with financial institutes
Competitiveness	CO	Market position
		The level of core capacities
		Differentiated strategy
Operating Risk	OP	The stability and diversity of procurement
		The stability of transaction
		The efficiency of production
		The prospects for demand, for product and service
		Sales diversification
		Sales price and settlement condition
		Collection of A/R
		Effectiveness of sale network

Table 1: Details about qualitative risk factors

As referred previously, this data set was used to build a set of rules using genetic algorithms in the paper by Myoung-Jong Kim and Ingoo Han [3], and there is mention of bankruptcy prediction on several papers using several algorithms and approaches like support vector machines, neural networks, etc, (https://scholar.google.com/scholar?q=bankruptcy+prediction&hl=en&as_sdt=0&as_vis=1&oi=scholart&sa=X&ei=9ClhU9yrB8mpsQTzu4DwDg&ved=0CCkQgQMwAA), but no explicit mention of this data set in particular in any place.

3 Algorithms/Proposed Solutions

The classification techniques used were rule based methods (Zero Rule and One Rule), Naive Bayes and Bayesian Belief Networks and Decision tree based methods(J48).

Regarding the data set used, the data set file (.arff) was uploaded to the **Weka** software. Weka is an open source data mining software made in Java which has a collection of machine learning algorithms for data mining tasks and these algorithms can be used on a data set or they can be used and called inside the Java code being developed, it has tools for "data preprocessing, classification, regression, clustering, association rules, and visualization".

3.1 Zero Rule

Applying the Zero Rule algorithm [11] first on the data which is the most simple one which calculates/predicts the mode for this example due to having only nominal attributes (mean for numeric attributes). The **K-fold cross-validation** [14] algorithm was used here , it divides the data set into k parts/subsets and it repeats the holdout method k times. Each of the times it repeats, one of the k subsets is used as the test set and the other k-1 subsets are joined in order to make a training set. Afterwards the average error across all k trials is computed. The obtained data was then the following:

```
=== Run information ===

Scheme:       weka.classifiers.rules.ZeroR
Relation:     bankrupt_qualitative
Instances:    250
Attributes:   7
              IR
              MR
              FF
              CR
              CO
              OP
              Class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

ZeroR predicts class value: NB

Time taken to build model: 0.06 seconds



=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         143               57.2    %
Incorrectly Classified Instances       107               42.8    %
Kappa statistic                          0
Mean absolute error                      0.4898
Root mean squared error                  0.4949
Relative absolute error                100      %
Root relative squared error            100      %
Total Number of Instances              250

=== Detailed Accuracy By Class ===

         TP Rate FP Rate  Precision  Recall   F-Measure  MCC     ROC Area  PRC Area  Class
         0.000   0.000    0.000      0.000    0.000      0.000    0.483     0.420     B
         1.000   1.000    0.572      1.000    0.728      0.000    0.483     0.564     NB
Wtd.Avg. 0.572   0.572    0.327      0.572    0.416      0.000    0.483     0.502

=== Confusion Matrix ===

   a   b   <-- classified as
   0 107 |   a = B
   0 143 |   b = NB
```

In this case the classifier was just able to predict correctly the most frequent instance which is not always the best choice or the correct classification. Being an algorithm that selects the most frequent value it's not too Far-fetched to get these values with a poor performance of only 57.2% of correctly classified instances, being of no use in this particular case unlike other data

sets in which it may indeed be a good index to evaluate performance.

3.2 One Rule

Applying the One Rule [10] algorithm on the data, which is also a pretty straightforward one, it uses the minimum-error attribute for prediction, discretizing numeric attributes, the **K-fold cross-validation** algorithm was also used here as in every classification algorithm (standard 10-fold). The results were the following:

```
=== Run information ===

Scheme:       weka.classifiers.rules.OneR -B 6
Relation:     bankrupt_qualitative
Instances:    250
Attributes:   7
              IR
              MR
              FF
              CR
              CO
              OP
              Class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

CO:
P -> NB
A -> NB
N -> B
(246/250 instances correct)


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         246               98.4    %
Incorrectly Classified Instances         4                1.6    %
Kappa statistic                          0.9672
Mean absolute error                      0.016
Root mean squared error                  0.1265
Relative absolute error                  3.2667 %
Root relative squared error             25.5606 %
Total Number of Instances              250

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.963     0.000     1.000      0.963     0.981      0.968     0.981     0.979     B
               1.000     0.037     0.973      1.000     0.986      0.968     0.981     0.973     NB
Wtd.Avg.       0.984     0.021     0.984      0.984     0.984      0.968     0.981     0.975
```

```
=== Confusion Matrix ===

   a   b   <-- classified as
 103   4 |   a = B
   0 143 |   b = NB
```

As we can see in the results, the results were quite positive with an excellent performance of **98.4%** correctly classified instances, a result drastically different than that of the Zero Rule proving to be a good measure of performance because it examines every attribute and chooses the best and most predictable one.

3.3 Naive Bayes

The Naive Bayes classifier [9] [15] uses estimator classes in order to classify the data and is based on applying the Bayes' theorem with strong independence assumptions between the features (naive). The results were the following:

```
=== Run information ===

Scheme:       weka.classifiers.bayes.NaiveBayes
Relation:     bankrupt_qualitative
Instances:    250
Attributes:   7
              IR
              MR
              FF
              CR
              CO
              OP
              Class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Naive Bayes Classifier

                Class
Attribute           B     NB
               (0.43) (0.57)
============================
IR
  P               27.0   55.0
  A               29.0   54.0
  N               54.0   37.0
  [total]        110.0  146.0

MR
  P               12.0   52.0
  A               24.0   47.0
  N               74.0   47.0
```

```
  [total]        110.0  146.0

FF
  P               2.0   57.0
  A               5.0   71.0
  N             103.0   18.0
  [total]       110.0  146.0

CR
  P               4.0   77.0
  A              18.0   61.0
  N              88.0    8.0
  [total]       110.0  146.0

CO
  P               1.0   92.0
  A               5.0   53.0
  N             104.0    1.0
  [total]       110.0  146.0

OP
  P              20.0   61.0
  A              25.0   34.0
  N              65.0   51.0
  [total]       110.0  146.0


Time taken to build model: 0.16 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         248               99.2    %
Incorrectly Classified Instances         2                0.8    %
Kappa statistic                          0.9837
Mean absolute error                      0.0088
Root mean squared error                  0.0568
Relative absolute error                  1.7944 %
Root relative squared error             11.478  %
Total Number of Instances              250

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.991    0.007    0.991      0.991    0.991      0.984    1.000     1.000     B
               0.993    0.009    0.993      0.993    0.993      0.984    1.000     1.000     NB
Wtd.Avg.       0.992    0.008    0.992      0.992    0.992      0.984    1.000     1.000

=== Confusion Matrix ===

   a   b   <-- classified as
 106   1 |   a = B
   1 142 |   b = NB
```

As we can see in the data this classifier yielded excellent results using cross-validation obtaining a performance of **99.2%** correctly classified instances, due to it assuming that each prediction is independent from the value of the other predictions doing in this case a correct classification of the data. The performance for the training set was of **100%** correctly classified instances.

3.4 Bayesian Belief Network

Bayes network learning [1] [6] [12] uses various search algorithms and quality measures. The bayes network classifier provides datastructures (network structure, conditional probability distributions, etc.) and facilities common to Bayes Network learning algorithms like K2 and B. In this case the K2 algorithm was used.

K2 [8] is a bayes network learning algorithm that uses a hill climbing algorithm restricted by an order on the variables. And the bayes network uses SimpleEstimator which estimates the conditional probabilities tables of a bayes network once the structure has been learned (estimates the probabilities directly from the data).

Applying the Bayes Network classifier on the data set using 10-fold the following results were obtained:

```
=== Run information ===

Scheme:       weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAY
Relation:     bankrupt_qualitative
Instances:    250
Attributes:   7
              IR
              MR
              FF
              CR
              CO
              OP
              Class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=7 #classindex=6
Network structure (nodes followed by parents)
IR(3): Class
MR(3): Class
FF(3): Class
CR(3): Class
CO(3): Class
OP(3): Class
Class(2):
LogScore Bayes: -1452.0937532983926
```

```
LogScore BDeu: -1484.9732528492011
LogScore MDL: -1490.5287117274383
LogScore ENTROPY: -1421.51045025416
LogScore AIC: -1446.51045025416


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         249               99.6    %
Incorrectly Classified Instances         1                0.4    %
Kappa statistic                          0.9918
Mean absolute error                      0.0076
Root mean squared error                  0.0523
Relative absolute error                  1.5487 %
Root relative squared error             10.578  %
Total Number of Instances              250

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.991    0.000    1.000      0.991    0.995      0.992    1.000     1.000     B
               1.000    0.009    0.993      1.000    0.997      0.992    1.000     1.000     NB
Wtd. Avg.      0.996    0.005    0.996      0.996    0.996      0.992    1.000     1.000

=== Confusion Matrix ===

   a   b   <-- classified as
 106   1 |   a = B
   0 143 |   b = NB
```

Looking at the percentage of correctly classified instances using the 10 fold cross-validation (**99.6%**) and the accuracy for each class based on the reported divergence between the network distribution on file and the one learned, we can see that it is pretty accurate regarding the initial training set with **100%**. The numbers for the accuracies and the amount of correctly classified instances are calculated by enumerating all possible instantiations of all variables.

As we can observe the best result so far was using the bayes network classifier obtaining **99.6%** performance.

3.5 J48

This classification algorithm generates a pruned or unpruned C4.5 [13] decision tree in order to classify the data [7]. The results were the following:

```
=== Run information ===

Scheme:       weka.classifiers.trees.J48 -C 0.25 -M 2
Relation:     bankrupt_qualitative
Instances:    250
Attributes:   7
              IR
              MR
              FF
              CR
              CO
              OP
              Class
Test mode:    10-fold cross-validation

=== Classifier model (full training set) ===

J48 pruned tree
------------------

CO = P: NB (91.0)
CO = A
|   CR = P: NB (26.0)
|   CR = A: NB (25.0)
|   CR = N: B (5.0/1.0)
CO = N: B (103.0)

Number of Leaves  :  5

Size of the tree :  7


Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         245               98      %
Incorrectly Classified Instances         5                2      %
Kappa statistic                          0.959
Mean absolute error                      0.0234
Root mean squared error                  0.1366
Relative absolute error                  4.7865 %
Root relative squared error             27.6068 %
Total Number of Instances              250

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure  MCC      ROC Area  PRC Area  Class
               0.963     0.007     0.990       0.963   0.976      0.959     0.993     0.986     B
               0.993     0.037     0.973       0.993   0.983      0.959     0.993     0.994     NB
Wtd Avg.       0.980     0.024     0.980       0.980   0.980      0.959     0.993     0.991
```

```
=== Confusion Matrix ===

   a   b   <-- classified as
 103   4 |   a = B
   1 142 |   b = NB
```

Looking at the results we can see we have less performance than the all the previously experimented classifiers except for the zero rule, obtaining a performance of **98%** correctly classified instances. In this example there is a 7 dimensional space which is created for the decision tree within the classes taking into account the 7 existent attributes. The attribute with the greatest information gain is chosen to make the decision and then repeats in the attribute with the least information gain. In this case it doesn't work as intended because the generated tree is a more complex tree than that of the naive bayes and bayesian network resulting in a slight loss in performance with **99.2%** correctly classified instances.

3.6 Gain and attribute evaluator

Using the InfoGainAttributeEval we can evaluate the worth of an attribute by measuring the information gain with respect to the class. Using the Ranker search method, the ranks attribute by their individual evaluations with the attribute evaluator we can obtain the best indexes to test the performance. The following results were obtained:

```
=== Run information ===

Evaluator:    weka.attributeSelection.InfoGainAttributeEval
Search:       weka.attributeSelection.Ranker -T -1.7976931348623157E308 -N -1
Relation:     bankrupt_qualitative
Instances:    250
Attributes:   7
              IR
              MR
              FF
              CR
              CO
              OP
              Class
Evaluation mode:    evaluate on all training data



=== Attribute Selection on all input data ===

Search Method:
Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 7 Class):
Information Gain Ranking Filter
```

```
Ranked attributes:
 0.9018  5 CO
 0.5845  3 FF
 0.533   4 CR
 0.1061  2 MR
 0.0586  6 OP
 0.0459  1 IR

Selected attributes: 5,3,4,2,6,1 : 6
```

So looking at the results we can see that the yielded results were exceptionally good and we can try to select the attribute (or more than one) with the greatest information gain which in this case is the 5th one (CO - Competitiveness) to our advantage and check the performance.

4 Results

After removing the attributes with less information gain and running all the classification algorithms previously mentioned the performance was actually lower using the algorithm that had the best performance (**bayes network classifier**), the other algorithms the performance was similar but with higher relative absolute errors, which means that taking out the attributes with less information gain decreases the performance, so the best way to classify this set of data is with all the initial attributes present in the data and using the bayes network classifier in order to obtain the maximum performance possible and classify the data almost 100% correctly with a Kappa value of almost 1 (0.9918) correctly classifying 245 instances out of 250. Besides the bayes network, the naive bayes, J48 and One Rule algorithms also yielded great results having high performances.

5 Conclusions and future work

With the use of some of the classification algorithms available to test their performance while classifying the data was a positive experience. Most of the algorithms applied had a great performance classifying correctly most of the data and it could be ascertained the one with the best performance for this specific case. With the current economic status across the world it is very positive that this kind of data mining solutions and bankruptcy predictions are starting to "automatize" a bit of the evaluation needed to ascertain if a company is probably going to go into bankruptcy or not.

This is definitely a good theme to pick up later and try other perspectives, other classification algorithms and try new approaches to this data set and develop code with some changes to the algorithms and the data set itself to see how is it that other approaches influence this problem and how it can evolve to something that can be of benefit to companies in order to "audit" their current state and predict their economic status in the future.

References

[1] Remco R. Bouckaert. *Bayesian Network Classifiers in Weka for Version 3-5-7*. http://www.cs.waikato.ac.nz/remco/weka.bn.pdf, 2008.

[2] Marc Davis. *The Impact Of Recession On Businesses*. http://www.investopedia.com/articles/economics/08/recession-affecting-business.asp, 2008.

[3] Myoung-Jong Kim and Ingoo Han. Expert systems with applications 25 (2003) 637–646. *The discovery of experts' decision rules from qualitative bankruptcy data using genetic algorithms*, 2003.

[4] M. Lichman. *UCI Machine Learning Repository*. http://archive.ics.uci.edu/ml. University of California, Irvine, School of Information and Computer Sciences, 2013.

[5] UCI Machine Learning Repository. *Qualitative Bankruptcy Data Set*. http://archive.ics.uci.edu/ml/datasets/Qualitative_Bankruptcy, 2014.

[6] Weka. *Bayes Network algorithm*. http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/BayesNet.html.

[7] Weka. *J48 algorithm*. http://weka.sourceforge.net/doc.dev/weka/classifiers/trees/J48.html.

[8] Weka. *K2 algorithm*. http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/net/search/global/K2.html.

[9] Weka. *Naive Bayes*. http://weka.sourceforge.net/doc.dev/weka/classifiers/bayes/NaiveBayes.html.

[10] Weka. *OneR Classifier*. http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/OneR.html.

[11] Weka. *ZeroR Classifier*. http://weka.sourceforge.net/doc.dev/weka/classifiers/rules/ZeroR.html.

[12] Wikipedia. *Bayesian network*. https://en.wikipedia.org/wiki/Bayesian_network.

[13] Wikipedia. *C4.5 algorithm*. https://en.wikipedia.org/wiki/C4.5_algorithm.

[14] Wikipedia. *Cross-validation (statistics)*. https://en.wikipedia.org/wiki/Cross-validation_(statistics).

[15] Wikipedia. *Naive Bayes classifier*. https://en.wikipedia.org/wiki/Naive_Bayes_classifier.