Machine Learning assignment
"Yeast" data set

Tom Jaspers (105808)
Vrije Universiteit Brussel, Belgium

June 9, 2014

Abstract

This paper examines the multi-class imbalanced classification problem of identifying location of yeast proteins using various classifiers (Decision trees, k-Nearest Neighbor, Artificial Neural Networks). We investigate two feature selection techniques, Relief-F and Minimum Description Length Method, and show their importance by comparing the performances of the classifiers before and after applying feature selection.

1 Introduction

In this first section we give details about our used Yeast data set, and look at how to the deal with the class imbalance it presents. Furthermore, we present the three learning algorithms that we used to classify the data, along with the two feature selection algorithms we have applied.

The second section shows our experimental setup and the various efforts undertaken to tune the algorithms. The results from these experiments are shown in Section 3.

We conclude the paper in the fourth and final section.

1.1 Yeast data

The yeast data set contains 1462 rows, of which 9 are duplicates, leaving us with 1453 items. It contains 8 continuous, noisy, predictor variables (various scores based on their amino acid sequence), and 10 classes (indicating the location of the yeast protein).

We performed a split of 75/25 to use for training, and hold out for testing, leaving us with 1090 training samples and 363 testing samples. The testing samples were held out until the day of paper writing.

From the class distribution shown in Table 1 we can conclude that the class distribution is highly skewed. Indeed, the majority of instances belong to four of the 10 classes (CYT, ME3, MIT, or NUC) making this an imbalanced problem.

CYT	ERL	EXC	ME1	ME2	ME3	MIT	NUC	POX	VAC
438	5	35	43	51	162	244	425	20	30
30.14%	0.34%	2.41%	2.96%	3.51%	11.15%	16.80%	29.25%	1.38%	2.06%

Learning from imbalanced class problems is considered to be one of the 10 challenging problems in machine learning [21]. The problem has been well studied in the case binary classification [11], and is now active research topic w.r.t. multi-class imbalance [20].

Strategic re-sampling is a common technique, where data is removed (undersampling) or synthetic data is generated (oversampling), for which many different methods exist [2]. Cost-sensitive learning has been shown to often outperform random re-sampling [8]. More recently, Active Learning approaches have been proposed to dealing with this problem [4, 7].

1.2 Learning algorithms

1.2.1 Decision trees

A decision tree is a structure where the internal nodes represent a test condition based on the value of one of the attributes, and the leaf nodes contain class values [12]. An instance is a assigned a label by following the nodes based on the tests until a leaf node is reached.

One of the most popular and commonly used implementations is C4.5 [13]. It choses the test condition at each node using the entropy measure, such that the best attribute test allows the highest difference of entropy between the parent node and children nodes.

More recently Quilan has released a successor to C4.5, namely C5.0[1], incorporating several improvements, such as Adaptive Boosting. In this report, we evaluate both.

1.2.2 Instance based learning

In instance based learning there is no explicit generalization, but the hypotheses are constructed directly from the training instances [17]. An immediate drawback from this is that the hypothesis complexity can grow with the data size, making it more computational (and memory) intense than generalized learning algorithms.

The k-nearest neighbors (k-NN) is a simple, but common form of instance learning. It uses a distance metric (e.g., Euclidean) to label to an unseen case, assigning the class most common among its k nearest neighbors [1]. Ties are broken at random.

1.2.3 Artificial Neural Network (ANN)

Artificial Neural Networks are a learning model that uses multiple layers of interconnected simple units (perceptrons). It is inspired on biological learning systems that consist of large networks of neurons. ANNs are robust to errors and are often successfully applied in fields as computer vision or speech recognition [12].

We used the *nnet* package provided in R, which fits a single-hidden-layer neural network, and use cross-validation to select an appropriate number of hidden units.

1.3 Feature selection

Dash and Liu provide a large review of feature selection techniques [3]. They categorize based on the evaluation method (e.g., distance measure, information measure, ...) and the generation method (random, heuristic, or complete). We have chosen to experimentally evaluate two of these algorithms, namely Relief-F and Minimum Description Length Method (MDLM).

1.3.1 Relief-F

Relief-F [10] is an extension of the original Relief feature selection algorithm used in binary classification [9]. Amongst other improvements, Relief-F was made suitable for multi-class problems.

It has been shown that Relief-F is able to handle noise [3, 16], it can work continuous, discrete, and nominal data [3], making it suited for our specific problem. We used the implementation in the *FSelector* package in R.

1.3.2 Minimum Description Length Method (MDLM)

The Minimum Description Length Method (MDLM) belongs to the category of complete generative feature selection methods, using information gain as its measure [18]. It thus searches all possible subsets ($2^8 = 256$ in our case) and outputs the best one. As the name suggests, it is based on the Minimum Description Length

[1] C5.0: http://www.rulequest.com/see5-info.html

principle, a formalization of Occam's Razor, that states that the best hypothesis for a given set is the one that leads to the best compression of the data [14].

Dash and Liu have shown that MDLM is able to handle noise, and works on continuous data [3], making it appropriated for our problem. We used the implementation in the *CORElearn* package in R [15].

2 Experiments

2.1 Performance measure

Most algorithms are mainly compared based on their accuracy, despite it not always being representative in imbalanced problems [19]. Indeed, one could trivially achieve high accuracy in an imbalanced binary classification by labeling all cases as the majority class. An alternative to this is plotting the receiver operating characteristics (ROC) graph. The area under the ROC curve (AUC) then provides a single scalar value between 0.5 and 1 to represent the performance, where 0.5 is as good as random, and 1.0 means no misclassification [5].

This measure is mainly used for binary classification, but has been extended for multi-class problems by n-way averaging the AUCs of one-vs-all classification [6]. The R package *pROC* [15] provides an implementation for this multi-class AUC measure.

2.2 Decision tree

Using the standard settings for the tree function in R we can generate a tree with 9 terminal nodes (representing 7 out of 10 class labels), and a misclassification error rate of 0.4028. The decision tree used five out of the eight features, namely {alm, gvh, mcg, mit, nuc}. Attempts to prune this tree gave no improvement, and we have thus used as is (referring to it as the *simple tree* in following sections). We have shown the resulting tree in Figure 1.

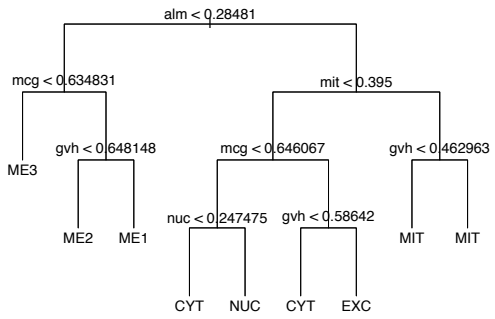
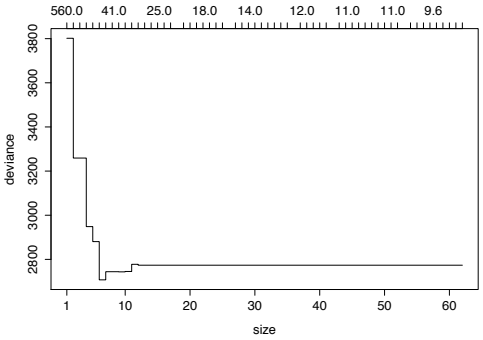


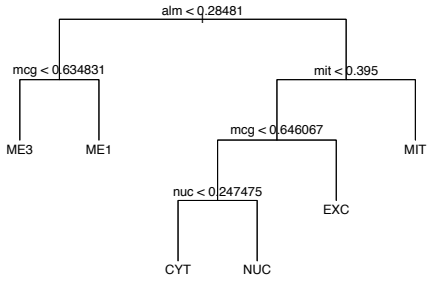
Figure 1: Decision tree (C4.5) obtained using the default settings. There are 9 terminal nodes that represent 7 out of 10 class labels. The features selected by the learning algorithm are {alm, gvh, mcg, mit, nuc}.

Seeing as this simple tree will only be able to classify the 7 classes for which it has leaves, there is an immediate restriction as we will be unable to correctly classify 3 classes. We have thus attempted to grow a more complex tree, that allows for more overfitting. This 62-leaved tree was then pruned (using cross-validation) to one of 6 terminal nodes.

We have also tested the C5.0 decision tree algorithm using the default configuration (see Results section), but no additional tuning was performed on it.



(a) Plot of the cross-validation results

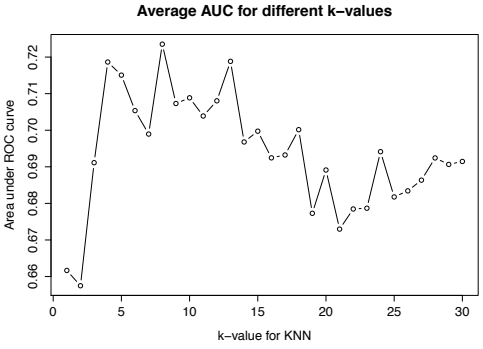


(b) The pruned (formerly complex) tree

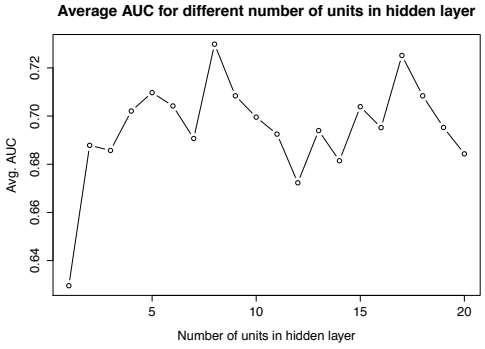
Figure 2: Results from cross-validation to prune the complex tree (a), and the resulting pruned tree (b).

2.3 *k*-NN and ANN with full featureset

We show the results of tuning the algorithms to find the optimal values for k in our k-NN algorithm, and the optimal number of units in hidden layer for ANN algorithm. We used the full featureset (all 8) for this, and present our results in the figure below.



(a) Best AUC (0.723527) using $k = 8$



(b) Best AUC (0.729850) using 8 hidden units

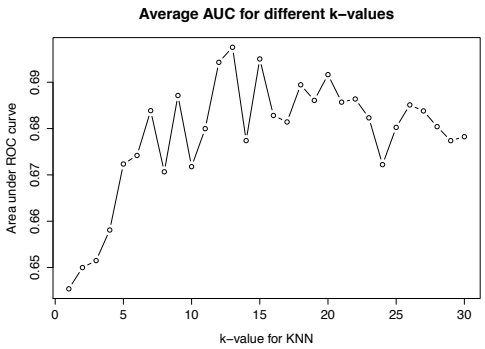
Figure 3: Results from 5-fold cross-validation to select the optimal value for k in our k-NN algorithm (a), and optimal number of units in hidden layer for ANN algorithm (b), using the full feature set.

2.4 *k*-NN and ANN with Relief-F features

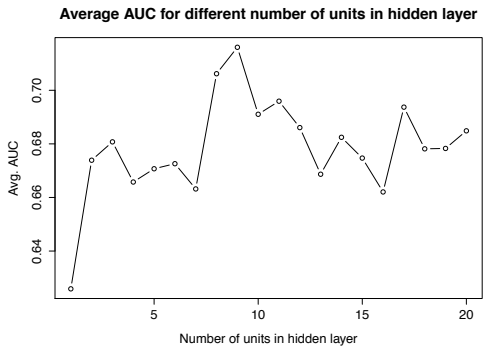
We applied the Relief-F feature selection algorithm, and choose the five[2] best features provided, namely {alm, gvh, mcg, nuc, vac}. Four out of the five features are same as those selected by the decision tree. Instead of the *mit* feature, Relief-F selected *vac*.

We then re-ran cross-validation on the the k-NN and ANN learning algorithms to find optimal values for k and the number of hidden units, respectively.

[2]The amount of features our decision tree ended up with.



(a) Best AUC (0.6975) using $k = 13$



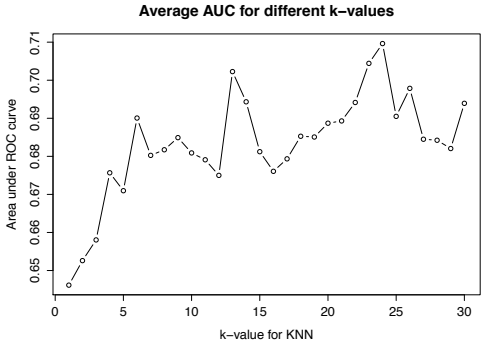
(b) Best AUC (0.7160) using 9 hidden units

Figure 4: Results from 5-fold cross-validation to select the optimal value for k in our k-NN algorithm (a), and optimal number of units in hidden layer for ANN algorithm (b), using the feature set obtained through the Relief-F feature selection algorithm.

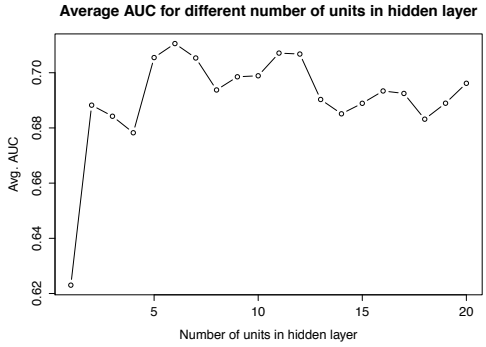
2.5 *k*-NN and ANN with MDL features

We applied the MDL feature selection algorithm to our dataset, and obtained the same five best features as our decision tree had obtained, namely {alm, gvh, mcg, mit, nuc}. The decision tree uses a form of information gain (namely difference in entropy) as a splitting criterion, which is the same kind of measure used by this MDL algorithm, and thus explains why the same set of features was obtained by both.

In order to find optimal values for k for k-NN and the number of hidden units for ANN, we re-ran cross-validation on these algorithms.



(a) Best AUC (0.7096) using $k = 24$



(b) Best AUC (0.7106) using 6 hidden units

Figure 5: Results from 5-fold cross-validation to select the optimal value for k in our k-NN algorithm (a), and optimal number of units in hidden layer for ANN algorithm (b), using the feature set obtained through the MDML feature selection algorithm.

2.6 Results

After tuning the algorithms, we trained them on the full training set of 1090 items. In the table below, we present our results from testing the trained classifiers on the independent, held-out test set of 363 items. We show the used configuration for the algorithms, and the set of features used.

Table 1: Results obtained from testing the classifiers on the held-out testing set, after they were trained on the full training set.

Algorithm	Configuration	Features	AUC
C4.5 simple	Default	{alm, gvh, mcg, mit, nuc}	0.5981
C4.5 pruned	$mincut = 2$, $minsize = 5$, $mindev = 0.0025$	{alm, mcg, mit, nuc}	0.6026
C5.0	Default	All	0.6211
k-NN	$k = 8$	All	0.6710
	$k = 13$	{alm, gvh, mcg, nuc, vac}	0.5958
	$k = 24$	{alm, gvh, mcg, mit, nuc}	0.5821
ANN	8 hidden units	All	0.6150
	9 hidden units	{alm, gvh, mcg, nuc, vac}	0.6366
	6 hidden units	{alm, gvh, mcg, mit, nuc}	0.6335

The pruned, formerly more complex, C4.5 tree performs slightly better than the simple C4.5 tree. This is an interesting result, as the pruned tree only has 6 terminal nodes, compared to the 7 of the simple tree, and is thus unable to classify 4 out of 10 classes.

It is unsurprising that the C5.0 algorithm outperforms it predecessor, even with minimal tuning. Indeed, C5.0 is a much more complex model, applying boosting techniques to generate and combine multiple decision trees.

The results of the k-NNs are surprising in two ways. First is that it has demonstrated the best performance over all other learning algorithms, while using the full feature set. Seeing as k-NN relies on majority voting, it is sensitive to the class imbalance, and thus we had expected worse results. Secondly, despite being known for its sensitivity to irrelevant features, both of the feature selection algorithms have severely decreased the performance of k-NN. We can see that the model complexity increased as well, as it used just 8 neighbors using the full featureset, but 24 using the limited featureset obtained by MDLM.

However, both Relief-F and MDLM have provided performance boosts for the ANNs. We had expected the impact of the features would be less in the ANNs.

3 Conclusions

We have applied three widely used learning algorithms - Decision Trees, k-Nearest Neighbors, and Artificial Neural Networks - on the Yeast data set, a ten-class imbalanced data set with 8 predictor variables. Using the feature selection methods Relief-F and Minimum Description Length Method, we reduced the dimensionality to the same amount of features naturally used by the decision trees. We compared the learning algorithms on the full feature set, and on the two features sets obtained after feature selection.

We found that both Relief-F and MDL provided performance boost in the neural networks. However, the k-NN algorithm performed much worse using the limited featuresets, a surprising result considering that it is known to be sensitive to irrelevant features. The best performance was thus obtained with k-NN ($k = 8$) using the full featureset.

References

[1] David W Aha, Dennis Kibler, and Marc K Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.

[2] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *arXiv preprint arXiv:1106.1813*, 2011.

[3] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[4] Seyda Ertekin, Jian Huang, Leon Bottou, and Lee Giles. Learning on the border: active learning in imbalanced data classification. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 127–136. ACM, 2007.

[5] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[6] David J Hand and Robert J Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45(2):171–186, 2001.

[7] Timothy M Hospedales, Shaogang Gong, and Tao Xiang. Finding rare classes: Active learning with generative and discriminative models. *Knowledge and Data Engineering, IEEE Transactions on*, 25(2):374–386, 2013.

[8] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[9] Kenji Kira and Larry A Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, 1992.

[10] Igor Kononenko. Estimating attributes: analysis and extensions of relief. In *Machine Learning: ECML-94*, pages 171–182. Springer, 1994.

[11] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36, 2006.

[12] Tom M Mitchell. Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45, 1997.

[13] John Ross Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[14] Jorma Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, 1978.

[15] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frdrique Lisacek, Jean-Charles Sanchez, and Markus Mller. proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, 12:77, 2011.

[16] Marko Robnik-Šikonja and Igor Kononenko. Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, 53(1-2):23–69, 2003.

[17] S Russell and P Norvig. Artificial intelligence, a modern approach, the intelligent agent book. 2003. *ISBN: 0-13-080302-2*.

[18] Jacob Sheinvald, Byron Dom, and Wayne Niblack. A modeling approach to feature selection. In *Pattern Recognition, 1990. Proceedings., 10th International Conference on*, volume 1, pages 535–539. IEEE, 1990.

[19] Marina Sokolova, Nathalie Japkowicz, and Stan Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *AI 2006: Advances in Artificial Intelligence*, pages 1015–1021. Springer, 2006.

[20] Shuo Wang and Xin Yao. Multiclass imbalance problems: Analysis and potential solutions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(4):1119–1130, 2012.

[21] Qiang Yang and Xindong Wu. 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making*, 5(04):597–604, 2006.