12

COMPARING SYMBOLIC AND SUBSYMBOLIC MACHINE LEARNING APPROACHES TO CLASSIFICATION OF CANCER AND GENE IDENTIFICATION

Werner Dubitzky, Martin Granzow, Daniel Berrar
German Cancer Research Center,
Intelligent Bioinformatics Systems Group,
Heidelberg, Germany

Abstract: Microarray experiments provide the scientific community with huge amounts of data. Without appropriate methodologies and tools significant information and knowledge hidden in these data may not be discovered. Therefore, there is a need for methods capable of handling and exploring large data sets. The field of data mining and machine learning provides a wealth of methodologies and tools for analyzing large data sets. Generally, such methods can be categorized into symbolic and subsymbolic methods. In the context of the acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) data sets provided by CAMDA 2000, we investigated and compared two representatives of classical machine learning approaches: *decision trees* (symbolic) and *artificial neural networks* (subsymbolic). The experiments indicated that the symbolic approach yields more accurate results than the neural network method.

Key words: Microarray analysis, gene expression profiling, decision tree, artificial neural network

INTRODUCTION

Classification of human tumors into distinguishable entities is preferentially based on clinical, pathohistological, enzyme-based histochemical, immunohistochemical, and in some cases cytogenetic data. Such an approach to classification often provides classes containing tumors that show similarities but differ strongly with respect to some important aspects, such as clinical course, treatment response, or survival. Data

obtained from high-throughput experiments (e.g., cDNA microarrays for profiling gene expression in tissues) have the potential to address this problem. However, microarray experiments generate huge amounts of data that is frequently characterized by many more descriptors (independent variables) than observation entities (cases). This raises the question as to which methods could be reliably used to analyze such data.

Based on microarray gene expression data, we investigated the performance of two popular machine learning [Mitchell, 1997] techniques in the context of molecular classification of cancer and identification of potentially relevant genes. The techniques in question are *decision trees* and *artificial neural networks* (see chapter 1 for a more general discussion of these techniques). Decision trees belong to the class of so-called symbolic methods. The attractiveness of decision trees is largely due to their ability to express the learned models as symbolic rules that can readily be understood by humans. Neural network approaches, on the other hand, represent their learned knowledge as patterns of connectivity that exist among the nodes of the network. This type of knowledge representation is sometimes called subsymbolic, and is not readily intelligible by humans. In their standard implementation both methods do not readily lend themselves to the determination of single-event probabilities of their predictions. However, this does not necessarily mean that these methods cannot be statistically validated.

Rather than devising special-purpose algorithms, our intention was to use standard, off-the-shelf software tools for analyzing the data provided for the CAMDA competition [CAMDA, 2000]. As a basis for our comparative study we chose two of the most popular algorithms currently available in proprietary machine learning and data mining software: the decision tree algorithm C5.0 and the well-known backpropagation algorithm for neural networks [Witten & Frank, 2000; Berry & Linoff, 1997; Dayhoff, 1996]. For both algorithms we used the proprietary implementation realized in the data mining tool Clementine from SPSS [SPSS]. Besides using standard tools, our motivation was to contribute to the understanding of the underlying techniques, their benefits and their limitations.

DESCRIPTION OF PROBLEM

A recent study [Golub et al., 1999] analyzed gene expression data from a training set of 38 (27 ALL, 11 AML) and a test set of 34 (20 ALL, 14 AML) leukemia patients for class discovery (cluster analysis) and prediction (discriminant analysis). Out of 7,070 genes, they found 1,400 genes to be highly expressed in ALL or AML. From those they picked an arbitrary total

of 50 genes with the highest association with disease type and used them for prediction. The aim of our analysis is to investigate

- the relative classification performance of a standard symbolic (decision tree C5.0) and a standard subsymbolic (backpropagation artificial neural network) machine learning method; and
- the value of these methods with respect to their capability to identify the most relevant genes.

DESCRIPTION OF EXPERIMENTS

The general approach of our analysis was to directly use all expression data (except the control data) without further processing.

Firstly, we aimed at comparing the performance of the neural network and decision tree methods on the basis of (a) a 5-fold cross-validation procedure applied to *all* 72 cases plus the original training/test set (henceforth referred to by cross-validation index cv0), and (b) a validation only on the basis of the provided training/test data split. Secondly, we aimed at identifying relevant genes based on the best-performing model. And thirdly, we aimed at finding a way of usefully combining neural networks and decision trees.

As both the backpropagation neural network algorithm and the C5.0 decision tree algorithm allow some control of the learning behavior through parameter settings, we decided after some exploratory screening to run three sets of experiments with different parameter configurations for either approach. So we did three times 6-fold cross-validation runs with different decision tree configurations and three times 6-fold cross-validation runs with different neural networks configurations.

Cross-Validation

We have randomly subsampled the entire set of 72 cases into five training sets (n=15) and five test sets (n=57). Henceforth, these cross-validation samples are referred to by the cross-validation indices: cv1, cv2, ..., cv5. In addition to these five training/test set pairs, we used the original training set (n=38) and test set (n=34) as the sixth cross-validation pair; this is referenced by the cross-validation index cv0. The distribution of the tumors for these six cross-validation training/test set pairs is shown in Table 1.

Table 1. Distribution of tumors over the cross-validation data sets.

Tumor	cv 0	cv 1	cv 2	cv 3	cv 4	cv 5
	Training Sets					
AML	11	17	19	21	20	21
ALL	27	40	38	36	37	36
total	38	57	57	57	57	57
	Test Sets					
AML	14	8	6	4	5	4
ALL	20	7	9	11	10	11
total	34	15	15	15	15	15

To establish the performance of the six models and parameter configurations (3 times decision tree, 3 times neural network), we trained each model/parameter configuration six times (on each of the six training sets) and applied the resulting classifier to the corresponding test set. Then we computed the individual performances for each test set and the average performance, using absolute accuracy and *lift* measures, of each model/parameter configuration over the six test runs. See tabular presentation of these results in

Table *2* and Table 3. To determine the relative performances between the decision tree and neural network approach, we picked the best-performing (according to average performance over six test runs) of the three decision tree classifiers and the best-performing of the three neural network classifiers and established through *t*-statistics if the performance of the two is statistically significant. Please refer to the section on Statistical Validation of Best-Performing Models for a detailed description of the methods used.

Table 2. Cross-validation performances.

	ANN Test Set Performance			**Decision Tree Test Set Performance**		
cv / size	**Acc. Th: 90%**	**Acc. Th: 85%**	**Acc. Th: 80%**	**Standard**	**10-fold boo.**	**20-fold boo.**
cv 0; n=34	79.41*	79.41	70.59	91.17	91.17	91.18
cv 1; n=15	86.67	80.00	46.67	66.66	86.66	86.66
cv 2; n=15	86.67	86.67	93.33	80.00	100.00	100.00
cv 3; n=15	60.00	93.33	60.00	93.33	100.00	100.00
cv 4; n=15	66.67	73.33	66.67	86.66	86.67	86.66
cv 5; n=15	93.33	93.33*	93.33*	86.66	86.66	93.33
avg	**78.79**	**84.35**	**71.77**	**84.08**	**91.86**	**92.97**

*Legend: Acc. Th. = training accuracy threshold; boo. = boosting; * = pre-set threshold range could not be achieved.*

Table 3. Summary of results.

	Neural Network						
Test Set(s)	**Accuracy %**			**Lift**			**Parameters**
	AML	ALL	TOT	AML	ALL	TOT	
Best CAMDA; cv = 0	**50.00**	**100.0**	**79.41**	**2.42**	**1.25**	**1.84**	***a* = 87.5% (no >90%)**
Best All (cv = 0 .. *n*)	67.08	94.63	84.35	2.61	1.55	1.98	85% < *a* < 90%
Best Single Test Set	75.00	100.0	93.33	3.74	1.25	2.50	cv=5; *a* = 91.7%
All 3 Configurations	54.74	92.76	78.30	2.04	1.52	1.66	
	Decision Tree						
Test Set(s)	AML	ALL	TOT	AML	ALL	TOT	
Best CAMDA; cv = 0	**92.85**	**90.00**	**91.18**	**2.10**	**1.61**	**1.85**	**Std., 10fb, 20fb**
Best All (cv = 0 .. *n*)	92.56	92.62	92.97	2.64	1.54	2.09	20-fold boosting
Best Single Test Set	100.0	100.0	100.0	3.74	1.36	2.56	cv=2, cv=3 10/20fb
All 3 Configurations	90.94	88.28	89.64	2.44	1.36	2.56	

Legend: 10fb = 10-fold boosting; 20fb = 20-fold boosting

Parameter Configurations

The selection of model parameter configurations was largely determined by the implementation of the two approaches we chose (the Clementine data mining tool version 5.0.1 [SPSS]) and some initial exploratory experimentation.

C5.0 is the successor of C4.5 [Quinlan & Quinlan, 1997]. It is a complex and sophisticated decision tree algorithm with many parameter options. After some testing we chose to use the following three settings:

- *Standard*. In this mode the algorithm does not attempt to perform any boosting.
- *10-fold boosting*. Here, the algorithm attempts 10-fold boosting. Notice, depending of the actual performance of individual runs, there may be less than 10 models generated.
- *20-fold boosting*. Same as above, but with a fold of 20.

For all other setting we used the default values: *Gain Ratio* = ON, *Pruning Severity* = 70%, and *Minimum Record Per Branch* = 2. We did not specify any non-default costs for misclassification.

The neural network algorithm in Clementine supports various algorithms and parameter options. Critical for us was the option *dynamic* = ON, as this mode dynamically searches and evaluates different network topologies. After some experimentation we decided that option *Stop On = Accuracy* is most critical. If this option is selected one must specify a percentage value that determines a prediction accuracy threshold (for the training set). The

learning process is terminated once this threshold is reached. To avoid overfitting, we chose the following training accuracy thresholds: 80%, 85%, and 90%. For all other settings we used defaults. Similar to the boosting approach, that does not guarantee *n* models when *n*-fold boosting is specified, the training accuracy may linger just below a threshold (e.g., 80%) and then jump to 98%. Thus, for a given training set, it may not be possible to obtain a final training accuracy within a preset accuracy *interval* (e.g., [80%,85%]).

This leads to three runs of 6-fold (cv0, ..., cv5) cross-validation analysis using the C5.0 decision tree model with the settings *Standard*, *10-fold boosting*, and *20-fold boosting*, and to three runs of 6-fold (cv0, ..., cv5) cross-validation with the neural network backpropagation algorithm using the training accuracy threshold settings 80%, 85%, and 90% respectively. See also

Table *2* and Table 3 under Results and Discussion.

The Lift Measure

To glimpse into performance beyond simple accuracy or error measures, we used a measure frequently employed in machine learning called *lift*. The lift measures the strength of an effect against some expected "baseline". This measure takes into account the frequency of a class in the data set (test set), how often the class is predicted by the classifier, and how many *correct* predictions are made for the class.

Given the set of class labels, $C = \{c_1, ..., c_n\}$ and the set of cases, $S = \{x_1, ..., x_m\}$, let $act(x_j)$ denote the *actual* class (label) of case x_j and $prd(x_j)$ the class (label) *predicted* for x_j by a classifier. Then the *lift* for a particular class c_i, $lift(c_i)$, is measured by the prior probability, $p(act(x_j)=c_i)$, of class c_i occurring in S, and the conditional probability, $p(act(x_j)=c_i \mid prd(x_j)=c_i)$ of class $act(x_j)=c_i$ given the prediction, $prd(x_j)=c_i$, as follows:

$$lift(c_i) = \frac{p(act(x_j) = c_i \mid prd(x_j) = c_i)}{p(act(x_j) = c_i)} \qquad [1]$$

For example, let there be 15 cases in the test set, S, and let 3 of them have the class label A. So $p(act(x_j)=A) = 3/15 = 1/5$. If a model predicts the class A for 6 out of these 15 cases, and gets 2 of them correct, then $p(act(x_j)=A \mid prd(x_j)=A) = 2/6 = 1/3$. And then $lift(A) = (1/3) / (1/5) = 5/3 = 1.66$. Notice, the lift of 1.66 is much better than the lift of 1.00 which would be achieved if the model predicted class A for *all* elements in the set S, that is $p(act(x_j)=A \mid prd(x_j)=A) = 3/15 = 1/5$. This would of course lead to an

accuracy (for class A) of 100%, giving a false impression of the actual performance of the model.

Statistical Validation of Best-Performing Models

One of the main intentions of this study was to compare the performance of a standard decision tree and a standard neural net within the context of microarray data analysis. To do so, we first selected the best performing decision tree and neural configuration with respect to the performance on *all* test sets, cross-validation indices cv0 to cv5 (see

Table *2*). The neural network performed best by allowing it to train to an accuracy of more than 85% and less than 90%, and the best-performing decision tree model was that with 20-fold boosting. We computed the z-value according to equation [2], the *mean prediction error*, $merr(M;X)$, of the model, M, over all predictions in the common test set, $X = \{x_1, \ldots, x_n\}$, according to equation [3], the individual prediction error, $err(M;x_i)$, of the model, M, for an individual case, x_i, according to equation [4], the combined standard error, $se(A;X,B;X)$, of two models, A and B, on the common test set, X, according to equation [5], and the combined variance, $var(A;X,B;X)$, in accordance with equation [6]. Finally, the p-value was taken from a standard *t*-statistics table. Table 7 shows a summary of these statistics. This procedure is also described in [Weiss & Indurkhya, 1998].

$$z = \frac{\left|merr(A;X) - merr(B;X)\right|}{se(A;X,B;X)} \qquad [2]$$

$$merr(M;X) = \frac{\sum_{i=1}^{n} err(M;x_i)}{n} \qquad [3]$$

$$err(M;x_i) = \begin{cases} 0 \text{ if } prd(M;x_i) = act(x_i) \\ 1 \text{ if } prd(M;x_i) \neq act(x_i) \end{cases} \qquad [4]$$

$$se(A;X,B;X) = \sqrt{\frac{var(A;X,B;X)}{n}} \qquad [5]$$

$$\text{var}(A;X,B;X) = \frac{1}{n}\sum_{i=1}^{n}\left[\left(err(A;x_i) - err(B;x_i)\right) - \left(merr(A;X) - merr(B;X)\right)\right]^2 \qquad [6]$$

In equations [2] to [6], n refers to the number of cases in the data set (test set X), and $act(x_i)$ denotes the *actual* class of case x_i and $prd(x_i)$ the class *predicted* for x_i by a classifier or model M (or A or B).

Identification of Genes

To identify which genes are most important, we first identified the best-performing decision tree and neural configuration with respect to the performance on the given CAMDA 2000 test set (n=34), cross-validation index cv0. For the decision tree approach *all* tree configurations yielded identical performance (31 out of 34 correctly classified), and for the neural network approach the configuration training accuracy threshold = 85% performed best (27 out of 34 correctly classified). In fact, within the given time limits for training, we could not get the network to achieve a training performance of more than 87.5% (see also

Table *2* and Table 3). The genes we identified are presented in the section on Results and Discussion.

RESULTS AND DISCUSSION

We first look at the individual results and performances of the two approaches and then compare and contrast them. Consult tables

Table *2* and Table 3 for this discussion.

Neural Network Results and Discussion

The best classification performance of the neural network method was obtained by interrupting the backpropagation learning process between 85% and 90% (average: 88.43%) predicted accuracy. In this case, the average classification accuracy over all 6 cross-validation runs was 84.35%. Training the net to a predicted accuracy, x, of $x > 90\%$ and $80\% < x < 85\%$ respectively resulted in lower actual prediction performances (78.79% in the former and 71.77% in the latter case).

Further analysis showed that although for each of the three neural net runs the ALL tumor was classified with a higher accuracy than the AML class: ALL avg. classification accuracy over all three runs: 92.76%, for AML: 54.74%. However, the lift measure for the AML class scored higher in each of the test runs: ALL avg. lift score over all three runs: 1.52, for AML: 2.04. This means that the model showed a significantly higher sensitivity/selectivity with regard to the AML class. See also

Table *2* and Table 3 for a detailed breakdown of these results.

Training times for each neural network model was limited to a maximum of 5 minutes (on a 2 x Pentium III PC), and the obtained network topologies ranged from the most complex (6 nodes in the first, and 4 in the second hidden layer) to the least complex (2 nodes in the first, and 2 in the second hidden layer). The network topologies for the best-performing neural network configuration (85% < predicted accuracy < 90%) were 7070-6-4-1, 7070-3-3-1, 7070-4-2-1, 7070-2-2-1, 7070-5-4-1, and 7070-5-3-1 respectively. Incidentally, the topology notation follows the following neuron layer pattern: *input-hidden1-hidden2-output*.

Through sensitivity analysis we were able to identify which genes have the highest impact (relative importance) on the classification performance of the neural network. In *Table 4* we provide the first 15 genes of a sensitivity analysis carried out on the *entire* data set of 72 cases, based on a 7070-3-2-1 topology and a learning accuracy threshold setting of 90%. The full list can be obtained from the authors.

Table 4. Identified genes through neural network sensitivity analysis.

Gene Description	Gene Accession #	Sensitivity
ELA2 Elastatse 2, neutrophil	M27783_s_at	0.00902
(clone S31i125) mRNA, 3' end of cds	U61734_s_at	0.00838
PPGB Protective protein for beta-galactosidase (galactosialidosis)	M22960_at	0.00827
Homeotic Protein C6, Class I	HG3921-HT4191_f_at	0.00823
GB DEF = Myotubularin related protein 3 (MTMR3) gene, partial cds	U58034_at	0.00804
Oncoprotein 18 (Op18) gene	M31303_rna1_at	0.00797
Catalase (EC 1.11.1.6) 5'flank and exon 1 mapping to chromosome 11, band p13 (and joined CDS)	X04085_rna1_at	0.00797
GB DEF = GPSAT=glycophorin SAT [human, peripheral bloods, mRNA Partial, 407 nt]	S77893_s_at	0.00793
OBF-1 mRNA for octamer binding factor 1	Z49194_at	0.00791
LGALS3 Lectin, galactoside-binding, soluble, 3 (galectin 3) (NOTE: redefinition of symbol)	M57710_at	0.00787
GB DEF = HH2A/c gene	Z83742_at	0.00784
TCF11 Transcription factor 11 (basic leucine zipper type)	X77366_at	0.00779
KIAA0022 gene	D14664_at	0.00776
S100A2 gene, exon 1, 2 and 3	Y07755_at	0.00774

Decision Tree Results and Discussion

The best classification performance of the C5.0 decision tree method was obtained on the basis of 20-fold boosting (combination of multiple different

models). In this case the average classification accuracy over all 6 cross-validation runs was 92.98%. The result for 10-fold boosting was only marginally lower (91.87%). However, the non-boosting version of the decision tree only achieved an average classification accuracy of 84.09%. Interestingly, for the CAMDA 2000 training set (n=38) the boosting method was not able to derive multiple models, but repeated the known [Golub et al., 1999] result: Zyxin (accession code X95735_at) with an expression level of 938 as decision boundary. However, for many of the other cross-validation subsamples, boosting was able to identify multiple complementary models, thus indicating multiple genes and expression levels related to differentiating AML and ALL. The listing in Figure 1 shows a section of the trained decision trees for cv5 and 20-fold boosting. Indentations in the diagram indicate lower nesting levels, and the values in round brackets refer to *support* and *confidence* measures respectively.

```
Rule #1 - estimated accuracy 98.2% [boost 100.0%]:
    M23197_at =< 309 (34.0, 1.0) -> ALL
    M23197_at  > 309 (22.0, 0.955) -> AML
Rule #2 - estimated accuracy 96.4% [boost 100.0%]:
    M23197_at =< 390
        X71125_at =< 64 (38.6, 1.0) -> ALL
        X71125_at  > 64 (3.8, 0.605) -> AML
    M23197_at  > 390 (13.6, 1.0) -> AML
Rule #3 - estimated accuracy 94.6% [boost 100.0%]:
    M23197_at =< 390 (44.7, 0.957) -> ALL
    M23197_at  > 390 (11.3, 1.0) -> AML
Rule #4 - estimated accuracy 94.6% [boost 100.0%]:
    M31523_at =< 528 (26.2, 0.958) -> AML
    M31523_at  > 528 (29.8, 0.98) -> ALL
Rule #5 - estimated accuracy 92.9% [boost 100.0%]:
    L09209_s_at =< 911 (31.1, 1.0) -> ALL
    L09209_s_at  > 911 (24.9, 0.916) -> AML
...
```

Figure 1. Part of C5.0-generated decision tree for cv5, 20-fold boosting.

Further analysis showed that over all three C5.0 decision tree runs the AML class was classified with a higher accuracy than ALL (see

Table *2* and Table 3), with an average classification accuracy over all three runs: 90.94% for AML, and 88.28% for ALL. Moreover, the lift measure for the AML class scored significantly higher in each of the three test runs (ALL avg. lift score over all three runs: 1.50, for AML: 2.44). This means that the C5.0 decision tree model not only showed a significantly higher sensitivity/selectivity with regard to the AML class (when compared with ALL), but also a slightly higher precision. With regard to the ALL class both models showed comparable results regarding lift (sensitivity/selectivity) and precision (accuracy), but for AML the decision tree method clearly outperformed the neural net approach.

On an Intel-based PC (2 x Pentium III 600 MHz, 512 MB memory), training times of the C5.0 decision tree model construction ranged from 10-20 seconds for the non-boosting to 10-30 seconds for 10-fold boosting to approximately 100 seconds for 20-fold boosting.

Although in principle easy to implement, the identification of a full list of relevant genes with the decision tree implementation provided by Clementine turned out to be cumbersome. Furthermore, it was not possible to quantify the relative difference of the ranked genes. In Table 5 we provide a short list of genes (accession numbers) that were found to be most influential to the classification performance of the best decision tree model (20-fold boosting).

Table 5. Identified genes through decision tree analysis.

Gene Description	Gene Accession #
KIAA0181 gene, partial cds	D80003_at
CYSTATIN A	D88422_at
Glutamate Decarboxylase 1	HG2160-HT2230_at
Interferon gamma up-regulated i-5111 protein precursor	L07633_at
APLP2 Amyloid beta (A4) precursor-like protein 2	L09209_s_at
CD33 CD33 antigen (differentiation antigen)	M23197_at
CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)	M27891_at
PTX3 Pentaxin-related gene, rapidly induced by IL-1 beta	M31166_at
TCF3 Transcription factor 3 (binding factors E12/E47)	M31523_at
GB DEF = Amelogenin (AMELX) gene, 3' end of cds	M55418_at
DF D component of complement (adipsin)	M84526_at
GLUTATHIONE S-TRANSFERASE, MICROSOMAL	U46499_at
Melanoma growth stimulatory activity (MGSA)	X54489_rna1_at
TST Thiosulfate sulfurtransferase (rhodanese)	X59434_at
Max gene extracted from H.sapiens max gene	X66867_cds1_at
Zyxin	X95735_at

Decision Tree vs. Neural Net: Results and Discussion

After running 2 x 3 x 6 cross-validation tests, we picked the best-performing neural network (85% < predicted accuracy < 90%) and decision tree model (20-fold boosting) and compared them using *t*-statistics. It is perhaps interesting to first inspect a detailed breakdown of misclassification of the two best-performing models in *Table 6*. The best-performing decision tree misclassified three cases, and the best neural net failed to correctly classify seven cases. The only case misclassified by both models is case# 66, which gives rise to the assumption that this case was not labeled correctly in the first place or that it belongs to a new subcategory.

Table 6. Breakdown of misclassifications on CAMDA test set (n=34).

prd(ANN85)	prd(DT 20fb)	Actual Tumor Label	Case#
ALL	AML	ALL	68
ALL	AML	ALL	67
ALL	AML	AML	52
ALL	AML	AML	54
ALL	AML	AML	57
ALL	AML	AML	60
ALL	AML	AML	65
ALL	AML	AML	62
ALL	ALL	AML	66

The overall performances of the two models are depicted in Figure 2 (in the diagram, the numbers on the X-axis correspond to the discussed cross-validation indices as follows: 1 = cv0, 2 = cv1, ..., 6 = cv5).

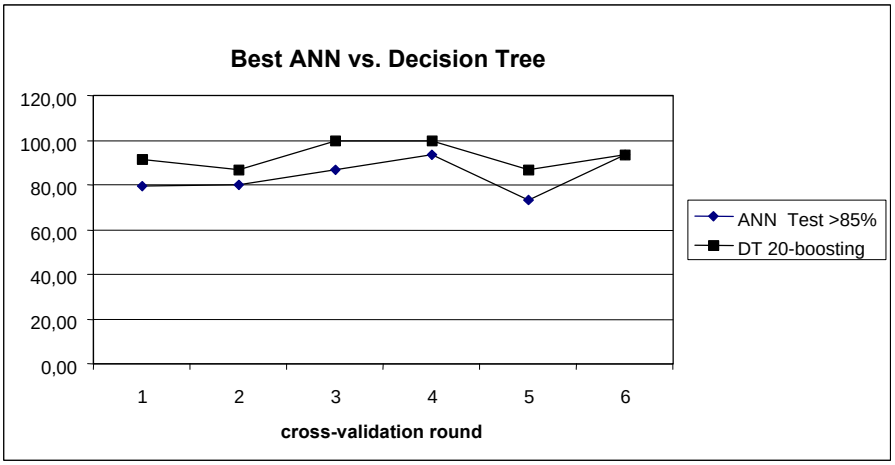

Figure 2. Overall classification performances of best-performing neural network and decision tree model (6-fold cross-validation).

It is easily seen from the diagram in Figure 2 that the best-performing decision tree model outperformed the best-performing neural network model (except in cv5 where both models fared equally). However, to decide if the performance differences are statistically significant is only possible after some statistical analysis. We carried out the analysis described in the section on Statistical Validation of Best-Performing Models [Weiss & Indurkhya, 1998]. Table 7 shows the results of this analysis.

Table 7. Statistics of best-performing models.

cv / size	merr ANN85	merr DT20	comb. SE	z-value	p-value
cv 0; n=34	0,21	0,09	0,08	3,64	< 0.01
cv 1; n=15	0,20	0,13	0,11	2,92	< 0.01
cv 2; n=15	0,13	0,00	0,09	1,52	n.s.
cv 3; n=15	0,07	0,00	0,06	1,04	n.s.
cv 4; n=15	0,27	0,13	0,13	3,11	< 0.01
cv 5; n=15	0,07	0,07	0,09	1,41	n.s.
avg	0,16	0,07	0,09	2,27	?

Legend: merr = mean error of model; ANN85 = neural net model with training accuracy threshold of 85%; DT20 = decision tree model with 20-fold boosting; comb. SE = combined standard error.

The results of the significance tests indicate that there are good reasons to believe that the performance difference of the two best-performing models is due to some systematic difference in their underlying learned mechanisms (rather than to chance variation). However, it would require further and more comprehensive tests to decide this with more confidence. The following is a list of observations and interpretations of the data analysis:

- both approaches can be used directly (no further preprocessing or discretization needed) with high-dimensional inputs (> 7000 genes) for molecular tumor classification and gene identification;
- the C5.0 decision tree classification model (a) shows higher precision and sensitivity levels, (b) provides an output format that is easy to interpret by humans (symbolic rules), and (c) is faster to train than the neural model; however, in the implementation we used it was cumbersome to generate a ranked list of important factors (genes) and quantify their relative importance;
- the neural network approach was treated unfairly in that it was not used in boosting (as this is not directly supported by the Clementine implementation); one could expect better performance (albeit with high computational costs) if the neural approach is used within a boosting framework;
- through sensitivity analysis the neural network approach was simpler and more precise with regard to ranking (plus relative performance) and identifying high-impact variables; however, this analysis carries high computational costs;

- Although significance analysis ranks all genes according to “impact” on classification, the analysis carries high computational cost when the number of genes is high ($n \sim 7000$);
- On the CAMDA training set (n=38) the decision tree was not able to come up with boosted (i.e., multiple) models as X95735_at (Zyxin) provided a perfect "split" that resulted in the following perfect-confidence rules: rule 1: ***if*** *X95735_at* £ *938* ***then*** *ALL (support=27.0, confidence=1.0)*, and rule 2: ***if*** *X95735_at > 938* ***then*** *AML (support=11.0, confidence=1.0)*. To identify additional high-impact rules required to remove X95735_at and re-run the learning step on the remaining factors. This procedure eventually yielded the factors listed in *Table 5.*

With regard to the genes found by the two approaches we can make the following observations/interpretations. Two among the 10 genes identified by the decision tree encoded for (1) CD33 differentiation antigen that is tissue specific for monocytic/myeloid lineage cells (accession# M23197_at), and (2) transcription factor 3 (E2A) (accession# M31523_at). This is part of the described fusion transcript PBX1-E2A occurring in B-cell precursor ALL with translocation t(1;19)(q23;p13.3). Among the 50 most important genes identified by the neural network, several belong to:

- *ras-oncogene pathway*: Oncoprotein 18/Stathmin, MAPKAP kinase 3 (3pK), HCK (Hemopoietic cell kinase)) (e.g., accession# U09578_at);
- *Hox-gene family*: HoxA9 (an Oncogene), HoxB3 and the potential effector (Retinoic acid receptor beta (RARb)) (e.g., accession# U82759_at);
- *apoptosis associated genes*: LTBP1, a member of the *TGF-ß signal pathway,* and Interleukin 16, a cytokine activating the SAPK (a member of the MAPK family) signaling pathway (e.g., accession# M34057_at);
- *cell-specific genes:* OBF-1/POU2AF1 that is involved in a form of B-cell leukemia, and LGALS3 that is widely distributed in various tissues with the notable exception of B- and T-lymphocytes (e.g., accession# M57710_at); and
- *potential tumor suppressor gene:* hSNF2ßb/BRG-1 (chromatin remodeling) (e.g., accession# D26156_s_at).

SUMMARY AND CONCLUSION

Based on the CAMDA 2000 competition ALL/ALL data [CAMDA, 2000; Golub et al., 1999], we compared two classical machine learning approaches with regard to their potential of identifying genes and their performance in terms of discriminant analysis. The experiments indicated that the symbolic (decision tree) approach yields more accurate results than the neural network method in terms of classification performance. However, we found the sensitivity analysis of the neural approach useful to obtain a ranked list of "influential" genes. Clearly, much more analysis experiments, statistical and biological validation is necessary to establish a comprehensive set of properties on machine learning approaches to microarray data analysis.

REFERENCES

Berry, M.J.A. and Linoff, G., "Data Mining Techniques: For Marketing, Sales, and Customer Support", John Wiley & Sons, 1997.

CAMDA 2000 Web site, http://bioinformatics.duke.edu/camda/camda00/.

Dayhoff, J.E., “Neural Network Architectures: An Introduction”, Thomson Computer Press, 1996.

Golub T.R., Slonim D.K., Tamayo P., Huard C., Gaasenbeek M., Mesirov J.P., Coller H., Loh M.L., Downing J.R., Caligiuri M.A., Bloomfield C.D., Lander E.S., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring", *Science* 286(5439) pp531-537, 1999.

Mitchell, T.M., "Machine Learning", McGraw-Hill, Singapore, 1997.

Quinlan, J.R. and Quinlan, R., "C4.5: Programs for Machine Learning", Morgan Kaufmann Series in Machine Learning, 1997.

SPSS: http://www.spss.com/datamine/, and Clementine User Group: http://www.spss.com/clementine/clug/ .

Weiss, S.M. and Indurkhya, N., "Predictive Data Mining: A Practical Guide", Morgan Kaufmann, Pub., San Francisco, CA, 1998.

Witten, I.H. and Frank, E., "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann Pub., 2000.