Application of Machine Learning on Gene expression data for breast cancer

Group#3 "CHRZ" members
David Charubini <dcharubi@cs.uml.edu>
James Hardin (team lead) <runiok@yahoo.com>
Sri Reddy <reddy22222@gmail.com>
Haiyang Zhang <hzhang@cs.uml.edu>

Introduction

Microarray techniques were developed in the late 90's to simultaneously measure the expression levels of thousand of genes. It’s a great improvement over previous DNA hybridization techniques. In organisms, the DNA transcribes into RNA, which in turn translates into proteins. The measurements of a protein is much more difficult than the RNA levels, which is measured by the microarray. In an experiment, usually 100-300 samples are used, and each sample is applied to a chip with thousands of probes.

Due to the large amount of data, machine learning methods are necessary to extract useful information. In supervised learning, we can classify the cancer types based on the gene expression levels. The accurate diagnosis of cancer types is important to prescribe proper treatments for patients. In unsupervised learning, cluster algorithms are applied to the genes and/or samples to discover the sets of genes with similar function, or new cancer types.

The dataset

Breast Cancer Gene Expression
There are 232 sample/cases in the initial dataset. The cases come from normal persons and breast cancer patients. 1544 features are available. They are expression levels of 1544 genes. Each sample is classified into one of the 6 classes including 1 normal and 5 types of breast cancers, which is the attribute to be predicted by our learning algorithms.

Each gene/sample data value is represented by a positive or negative floating point number between -5.983 and 7.307.

Problem Description

The problem given to our group requires evaluating the accuracy of decision tree learning, bagging decision trees, boosting decision trees, SVM or SMO in Weka and random forests on the gene expression data involving breast cancer. The implementation of the learning algorithms takes place within the Weka software package. Five fold cross validation is used on the data and is done outside of Weka to maintain an accurate comparison between the learning methods. The goal of the project is to test for a 99 percent level of significance in the difference between the accuracy of the best performer and the accuracy of the other learners. In addition, the problem requires reporting the level of significance and results. The p-values are also needed for comparing the different learning methods.

The difference in accuracies between learning methods is calculated for each comparison. The average of the differences is then calculated and used for calculating the standard deviation. At this point, the T-Statistic is determined for a 99% confidence interval with four
degrees of freedom since there was 5 fold cross validation. The T-Statistic is multiplied times the standard deviation and added or subtracted with the mean accuracies to give the final confidence interval.

**Possible learning problems**
Supervised learning: classify the cancer types based on the gene expression levels.
Unsupervised learning: cluster the genes and/or patients to discover the sets of genes with similar function, or new cancer types.

**Learning Algorithms**

**Decision Trees**

Nodes in a decision tree involve testing a particular attribute. Leaf nodes give a classification that applies to all instances that reach the leaf. If the attribute that is tested at a node is nominal, the number of children at that node represents the attribute values.

**Bagging**

Bagging is an ensemble method used to combine the outputs of multiple learning methods into a single output. The output value with the greatest number of votes usually wins. Weights can be used on the output.

**Boosting**

Boosting is unusual in the sense that it takes advantage of unstable characteristics in learning methods. It focuses more on learning incorrect classifications than other methods studied so far. By focusing more weight on these classifications, they become more important during the learning process.

**Random Forest**

This method uses an ensemble of decision trees where the subset of attributes for each tree is selected randomly from a total set of attributes. The classification values with the largest vote are chosen. Each tree is allowed to grow without pruning.

**Support Vector Machine**

The simplest SVM is a linear classifier that separates the data using a hyperplane into two classes. Sequential Minimal Optimization, or SMO which is used in Weka, can break the problem into subproblems. This could be an explanation as to why the SVM in the project was able to outperform other learning methods even though it classifies more than two values. The data class has six values in total. SVMs use a kernel function which is usually a polynomial, radial basis or sigmoid. The classification function can be described as \( f(x) = \text{sign}(w \cdot x - b) \) where \( w \) is a vector. One of the central ideas is to maximize the margin between the support vectors which are data points.

**Week 1 results**
Five different learning algorithms were evaluated in this project. In the final analysis, the best performing learning method appears to be the Support Vector Machine which is represented by SMO in Weka. Although it is designed for binary classification it is most simplistic form, it adapted itself in Weka to perform classifications of six values with the best overall accuracy on average when compared to decision trees, bagging, boosting and random forest. The following tables are provided to display the results.

We have tested different combinations of arguments in each algorithm, for example, in random forest, arguments with the following ranges are tested:


**Performance Table**

<table>
<thead>
<tr>
<th>Learning Method</th>
<th>Partition 1 Accuracy</th>
<th>Partition 2 Accuracy</th>
<th>Partition 3 Accuracy</th>
<th>Partition 4 Accuracy</th>
<th>Partition 5 Accuracy</th>
<th>Average Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decision Tree</td>
<td>71.7391</td>
<td>67.3913</td>
<td>58.6957</td>
<td>69.5652</td>
<td>70.8333</td>
<td>67.64492</td>
</tr>
<tr>
<td>Bagging</td>
<td>89.1304</td>
<td>76.087</td>
<td>63.0435</td>
<td>82.6087</td>
<td>85.4167</td>
<td>79.25726</td>
</tr>
<tr>
<td>Boosting</td>
<td>89.1304</td>
<td>76.087</td>
<td>76.087</td>
<td>91.3043</td>
<td>83.3333</td>
<td>83.1884</td>
</tr>
<tr>
<td>Random Forest</td>
<td>91.3043</td>
<td>78.2609</td>
<td>71.7391</td>
<td>93.4783</td>
<td>85.4167</td>
<td>84.03986</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>95.6522</td>
<td>73.913</td>
<td>80.4348</td>
<td>89.1304</td>
<td>91.6667</td>
<td>86.15942</td>
</tr>
</tbody>
</table>

**99% Confidence interval of the best performer:**

SVM's mean accuracy: 86.15942%
k = 5

Standard Deviation = $\sqrt{\text{Sum}(\text{Acc}[i]-\text{Average})^2/(k(k-1))} = 3.95$

$T(99,k-1) = 4.604$

**Confidence interval = mean +/- T*SD = 86.2% +/- 18.2%**

**Paired Comparison**

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Mean Delta</th>
<th>SD(mean Delta)</th>
<th>T-Stat = Mean/SD</th>
<th>p-value</th>
<th>Significant difference: p&lt;0.01 ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT-SVM</td>
<td>18.5145</td>
<td>3.08</td>
<td>6.01</td>
<td>0.0039</td>
<td>YES</td>
</tr>
</tbody>
</table>
Since the objective was a 99% confidence interval, the T value for this level of significance would be 4.6041 with a k factor of 5 which represents the number of subsets. The formulas used in calculating averages, standard deviation, confidence intervals and p-values have come from the Mitchell textbook, lecture slides and other sources.

**Conclusion**

This project involved testing and comparing the different learning algorithms. The lone decision tree had the least mean accuracy while the Support Vector Machine had the greatest. Given the large number of genes serving as attributes as well as six classification values, all the learning methods did reasonably well when compared to chance. Even though the SVM was the best performer, its accuracy was still limited to 86 percent. However, by changing the kernel functions, validation methods and other optimization parameters, it is still possible to achieve higher accuracy. The overall performance of the learning algorithms indicates that they will be useful tools for bio-informatics research.

**References**

4) Kristin Bennett & Colin Campbell, 2000, "Support Vector Machines: Hype or Hallelujah?", SIGKDD Explorations, Volume 2, Issue 2