ABSTRACT

Single nucleotide polymorphisms (SNPs) are genetic markers that may be used to identify the causes and risks of cancer. The sheer volume of data generated by SNP studies is difficult to analyze by hand. Machine learning techniques have been developed to address the types of data and the sizes of data sets provided by these studies in an efficient manner. We discuss the applicability of 5 machine learning techniques to the classification of cancer patients using SNP data. The techniques include decision trees, naive Bayes, neural networks, support vector machines, and clustering methods.

Keywords

cancer, machine learning, single nucleotide polymorphism

1. INTRODUCTION

Cancer is a genetic disease in which certain cells break free of normal growth controls. A number of factors can contribute to the DNA damage which leads to this uncontrolled cell division. Some of these factors are genetic conditions which increase the risk of the initiation of cancerous growth. The identification of these risk factors is of major interest in cancer biology.

Single nucleotide polymorphisms (SNPs) are single nucleotide variations of DNA. (see Table 1) While many of these small variations have no effect, some may influence certain drug reactions[2] or cause susceptibility to disease, including cancer. Some SNP's which may have no direct impact on health may be linked to other nearby genes which do have an effect. Technological advances have improved the efficiency

Figure 1: Sample SNP: BRCA1 Breast Cancer Early Onset.

tgaaacAgagcgaatgact - normal
tgaaacAgagcgaatgact - variant
SNP 1750018 [3]

with which we can now study large quantities of SNPs in significant numbers of individuals.

Although technological advances[4, 5] now allow us to study many SNPs concurrently over larger populations, the increasing amounts of data reduce the practicality of unaided human analysis[6]. Advances in the acquisition of data must be matched by advances in the analysis of data. Machine learning techniques[6, 7] are computational approaches by which patterns in known data can be learned and applied to new data. These intelligent systems can be applied to the domain of SNP data from cancer patients.

We propose the use of machine learning techniques to analyze the large amount of data produced by studies of single nucleotide polymorphisms in cancer patients. The aim of our project is to improve diagnosis and treatment of cancer. Specifically, we wish to determine if the available SNP data contains sufficient information to be able to classify cancer patients. This knowledge may help to improve clinical diagnosis and risk assessment which can lead to earlier, more specific, and more effective treatments. We also would like to identify genes which may be most significant causal factors in the development of cancer. With this information new treatments can be developed that better target the genes that are implicated by the SNP data. Finally, we want to determine which machine learning techniques will enable us to extract this information from the data set in the most efficient and accurate manner possible. It is this three-fold purpose upon which our project is based.

The problem of diagnosing cancer can be very challenging as there are many factors that influence the onset, manifestation, and progression of cancer. These factors can create ‘noise’ for a learning system that is focused on solely one aspect of patient data. In addition, real world clinical and molecular data can be difficult to obtain. In our data set, missing data is a major problem that affects the techniques required for data analysis.
2. BACKGROUND

2.1 PolymX

This machine learning project lies within the larger framework of the PolymX project.

The PolymX project aims to develop, implement, and document a broad spectrum molecular analysis of human cancers and their correlations to certain clinical outcomes. Working with pathologists and surgeons from a variety of regional hospitals, PolymX researchers are assembling a multi-tumor site tissue bank. These samples are then analyzed using the most recent advances in genomics, proteomics, metabolomics and bioinformatics. It is hoped that the wealth of molecular information collected by PolymX researchers will lead to major advances in the fight against cancer. [1]

The PolymX project will amalgamate many sources of data in order to improve patient diagnosis and treatment. The SNP data that we analyze in our project is a small part of that data. We intend to make the classification of cancer patients as accurate and efficient as possible using this single data source. Thus, when multiple data sources are brought together the Polymx system can be robust and practical.

We may consider the entire PolymX project as the performance system in this scenario. (see Figure 2) The 'SNP Learner' that we develop in this project is a single input into that larger performance system. The 'SNP Learner' is the learning system with which we are concerned.

To this point the PolymX project has been focused on data collection. The SNP data, in fact, arrived prior to and during the course of our project. This will be the first analysis of this data.

2.2 Cancer Research and SNPs

Single nucleotide polymorphisms have been used in a variety of ways for cancer research. It has been observed that certain SNPs indicate a predisposition to cancer [8, 9, 10, 11, 12, 13]. It has also been found that some SNPs may alter the effectiveness of various drug treatments [14, 15]. This, of course, is of huge concern to the pharmaceutical industry and many studies are currently underway.

SNP analysis has often been done with smaller numbers of SNPs in large populations. These analyses have typically been done using disequilibrium linkage analysis in order to identify correlations between a particular SNP or group of SNPs and a particular disease. This technique may commonly be used to map the location of the gene responsible for the disease.

Although many large gene expression studies have been done [16], to our knowledge no studies have been done to analyze a large group of SNPs (in our case, 209) using machine learning techniques as we propose. It is likely, however, that studies are taking place within the pharmaceutical industry that we are not aware of.

2.3 Machine Learning

The field of machine learning is focused on building computational intelligent systems that improve over time [6, 7]. Machine learning has applications in a wide variety of domains, from suggesting books to buy to autonomously driving vehicles. A large subset of these applications consist of creating a 'classifier' of some type. It is this collection of applications we will focus on. Classifiers typically learn from 'training data'. Training data is supplied to the classifier with class labels for each training instance. With more training instances or experiences, the classifier learns how to distinguish between the various classes. It may evaluate its performance according to the effectiveness of its classification compared to the true value of test instances given to it. In this way, machine learning classifiers seek to improve their performance.

There are a variety of classifiers that have been developed due to different approaches to machine learning theory and various practical constraints. In our project we wish to study classifiers that, given SNP data to train on, will be able to distinguish classes of patients and, ideally, provide some indication of which SNPs were most influential in the decision.

We will evaluate decision trees [17, 18], naive Bayes classifiers[19], neural networks[20], support vector machines[21], and clustering algorithms[22]. We will determine which, if any, will be able to extract information from the SNP data for the improvement of diagnosis and treatment.

2.4 Cancer Research and Machine Learning

Machine learning techniques have been applied to a variety of applications in cancer research.

The data most often analyzed, however, is cDNA microarray data[23]. The techniques used on microarray data include support vector machines[24, 25] and clustering methods[26, 27]. Other techniques have been used as well.
3. METHODS

3.1 Data
The SNP data that we acquired from the Cross Cancer Institute consisted of patients of various conditions. The three conditions for which we had sufficient data to operate with were breast cancer (B), leukemia (L), and normal (N). These three classes were used throughout the experiments and were the target classifications for the machine learning algorithms.

For each patient the data contained an array of SNP values from 0 to 3. As each human normally has two copies of each gene, these SNP values represented the combinations of normal or variant for the two gene copies. (see Figure 3) ’1’ represents having two normal genes (homozygous normal). ’2’ represents having one normal gene and one variant gene (heterozygous). ’3’ represents having two variant copies of the gene (homozygous variant). ’0’ represents an unknown result. An unknown result may be due to experimental error or due to lack of the gene expected in the SNP location. As such, a ’0’ may be a significant outcome.

For mathematical notation, we can represent the data set as \( S = \{ S_{1,j}, S_{2,j}, \ldots, S_{n,j}, C_j \} \) where \( S \) is the data set, \( S_{i,j} \) is the is the value of a certain SNP \( i \) for a certain patient \( j \), \( C_j \) is the class of that patient \( j \), and \( n \) is the number of SNPs being considered.

The original SNP data had many missing values. These values were predominantly found in the ‘normal’ data that we used as control data. These holes can be seen in the lower half of the data set in Figure 4. The missing data was assigned values of ’0’ or ‘unknown’.

3.2 Process Overview
In order to establish the utility of the SNP data and the most effective machine learning method by which to analyze it, we passed the data through some preprocessing steps and then through each of the given classifiers. (see Figure 5) Various methods were attempted for the preprocessing step, as discussed below. Parameters were also adjusted on each of the classifiers in order to get optimal results using the given processed data set.

3.3 Preprocessing
The original data set was missing a large amount of information for certain sets of patients and SNPs. It was necessary to preprocess the data set in order to minimize the effect of the missing data on the analysis. As seen in Figure 4 there is meaningless regularity in the data due to the missing data. As the classifiers are built to recognize patterns, any classifier \( h \) will recognize patients with large amounts of missing data as ‘normal’. For such a data set the sample error in experiments will be low and seem to indicate success in learning. The true error in the entire distribution of patients in the world, however, will be quite high since this pattern will not hold in general. As a result, the classifier will appear to do very well on experimental data but will
Table 1: Data Sets created by splitting on patient class.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Patient Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN</td>
<td>cancer, normal</td>
</tr>
<tr>
<td>BN</td>
<td>breast cancer, normal</td>
</tr>
<tr>
<td>LN</td>
<td>leukemia, normal</td>
</tr>
<tr>
<td>BL</td>
<td>breast cancer, leukemia</td>
</tr>
<tr>
<td>BLN</td>
<td>breast cancer, leukemia, normal</td>
</tr>
</tbody>
</table>

Table 2: Distribution of data among patient classes.

<table>
<thead>
<tr>
<th>Patient Type</th>
<th>Patient Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>breast cancer</td>
<td>139</td>
</tr>
<tr>
<td>leukemia</td>
<td>33</td>
</tr>
<tr>
<td>normal</td>
<td>136</td>
</tr>
</tbody>
</table>

perform very poorly when used on real world data. Various preprocessing steps were used to prevent overfitting.

3.3.1 Pruning

Some particular patients (rows) and SNPs (columns) in the data set contained a large amount of missing data. One method which was used to prevent this missing data from interfering with the results was by simply pruning those rows or columns completely from the data set. After some testing, two representative pruning thresholds, 0.1 and 0.5, were chosen.

Patients were pruned from the data set if the proportion of missing data for that particular patient was greater than the given threshold.

SNPs were pruned from the data if the amount of missing data for that SNP and any class of patient was greater than the threshold. For example, assume a set containing 300 total patients and pruned using the 0.1 threshold. If for a particular SNP 20 data points were missing, that SNP would not necessarily be pruned as this is less than 30. If, however, there were 100 normal patients and data was missing from 20 of these, the SNP would be pruned since that class would exceed the missing data threshold of 0.1. This style of pruning was done to avoid the problem of missing data being particularly concentrated in a single class. In this case, a large amount of normal data was missing.

It should be noted that although using the 0.5 threshold greatly reduced the total amount of missing data in the data set, it also removed a disproportionate number of ‘normal’. For this reason we also used the 0.1 threshold data set which had more missing data but a more balanced set of patient classes (about 1:1).

3.3.2 Patient class comparisons

Five unique data sets were also created on the basis of patient class. The three main classes were ‘breast cancer’, ‘leukemia’, and ‘normal’. The data sets were then created from various combinations of these as laid out in Table 1.

We tried using the BL and BLN data sets in order to determine if our methods could differentiate between different types of cancer. The results on the BLN data set were lower than the BN and LN sets individually, which indicated that there were patterns within each cancer type which the learning techniques had difficulty distinguishing when the data sets were combined. Unfortunately, the small amount of leukemia data (33 patients) led to very poor results in the LN, BL, and BLN data sets. Since the leukemia data made up only from 10% to 25% of each data set (see Table 2), each classifier could simply classify all the patients as the larger class and the error would only be 10% to 25% but no actual classification would be occurring. For this reason we did not focus on the leukemia data sets any longer in the analysis.

We tried the CN data set by mixing the breast cancer and leukemia patients into one class to see if the classifiers could find a trend in the cancer patients in general. The accuracy was consistently lower. Upon checking the most significant attributes of each analysis, it was clear that the attributes important in breast cancer were outweighing any important attributes in the combined set because of the small amount of leukemia data. As the leukemia seemed to only provide noise with no additional information in this data set, we had little confidence in those results and removed the set from consideration.

With these results, the main data set that we focused on for our analysis was the BN (breast cancer/normal) data set with about 135 patients of each class.

3.3.3 Sampling

Another technique that was used to reduce the effects of missing data on the results was sampling. Using this imputation technique, unknown values were filled in with a random sampling of values from the distribution seen in the known values. The distribution used for unknown SNP values in a particular patient class was taken only from the known values within the same patient class. Given $S_{ij}$ is the value of a certain SNP $i$ for a certain patient $j$ and $C_j$ is the class of that patient $j$.

$$S_{ij} \in \{0, 1, 2, 3\}$$

$$C_j \in \{B, L, N, C\}$$

$$P(S_{ij} = s | C_j = c) = \frac{\sum_{j} |C_j = c, S_{ij} = s|}{\sum_{j} |C_j = c|}$$

Unknown data were assigned values according to the distribution given by Equation 3. This filled out data that was not pruned so that it would not create meaningless and easily recognizable patterns. The resulting data set is shown in Figure 6.

Although we hope that this sampling prevents overfitting of the results, there is also the danger that correlations between SNPs may be lost in the the randomness of the sampling. As seen in the Results, this affects each machine learning technique differently since some assume independence (naive Bayes) between the SNPs and some do not.

3.4 Cross-validation

Another technique used to prevent overfitting of the data is that of cross-validation. With this technique the data set
3.5 Classification Technologies

3.5.1 Decision Trees

The nature of the SNP data set pointed to potentially learning the cancer classifier using a straightforward and common machine learning mechanism: the decision tree. While having a high dimensionality each attribute within a tuple has only 3 or 4 potential discrete values. As a result of this small attribute domain range, decision trees created based on this data would have a small branching factor providing a quick and compact classifier. An additional benefit of the decision tree method was the ability to easily derive human understandable information from the structure of the tree. In this form of early stage interdisciplinary research, the reasons behind the results are as important as the results themselves.

The actual algorithm used to generate the decision tree was C4.5 [18] This algorithm was chosen over other simpler learners such as ID3, because of its ability to prune the decision tree. Since the sample data had a both large number of attributes and was noisy the trees produced by ID3 would be excessively deep and would likely be inaccurate on untrained data. These problems would be magnified by the relatively small data set that was available for training. By performing pessimistic post pruning C4.5 would significantly reduce the size of the decision tree by removing the irrelevant, and theoretically evenly distributed, attribute branches. This would in theory generalize the decision tree and result in better accuracy.

We confirmed the WEKA analysis using the CART program[35]. Similar results were found.

3.5.2 Naive Bayes

The naive bayes learning algorithm is a conceptually simple idea. The algorithm predicts classes by choosing the most probable class, based on the available evidence.

$$\text{argmax}_{\text{class}} P(\text{class} | \text{snp}_1, \text{snp}_2, ..., \text{snp}_n)$$

By using Bayes theorem, and counting the occurrences of each SNP and classes within a dataset the probabilistic classification of a new sample can be made computationally feasible.

The use of naive bayes in context of this problem was promising for several reasons. The first reason is that naive bayes are adept at handling noisy data due to their probabilistic nature. The biological nature of the data also introduces the possibility that the ‘real’ cancer classifier is in no way discrete. For example two people may have identical SNP results, but due to other factors, one individual may have cancer while the other does not. The learned cancer classifier using Naive bayes will take into account such cases and create (within the limitations of the data) probabilities for a patient having cancer or not. Using simple Naive bayes, in this dataset context, provides some causality, which can be useful for further investigation by biological researchers. A final benefit is that in the future as more information and known probabilities become available, they can be added into a naive bayes model to improve its predictive accuracy.

With using naive bayes an important assumption must be stated. This assumption, used to simplify bayes calculation within the learning algorithm, is that each SNP attribute is independent of every other one. This assumption is reasonable to make, but may not always hold in all cases.

3.5.3 Neural Nets

Neural networks also appear as a natural approach to learn a cancer classifier from the SNP data. Neural networks are robust in dealing with highly dimensional noisy data. The long training time for neural networks was not considered a problem in this application. The offline learning of the cancer classifier could be given an arbitrary large amount of time. The most significant drawback with using neural networks in this setting is the difficulty of extracting human readable information from the trained neural network. While this disadvantage inhibits the direct study of the causality within classifier, the predictive performance of the classifier is itself worth investigating.

This study utilized two forms of neural networks: basic perceptron and multilayer feedforward networks. Both types of networks use an input unit for each SNP attribute of a patient and have only one output unit that classifies the patient as having cancer or not having cancer. The training of both types of networks used the traditional back propagation [6] algorithm with a learning rate of 0.3 and 0.2 momentum to a maximum of 500 epochs.
3.5.4 Support Vector Machines

The Support Vector Machine (SVM) was introduced in COLT ’92 by Boser, Guyon, and Vapnik. It has been greatly developed ever since and become one of the most efficient learning methods applied to the real applications today, which include text categorisation, hand-written character recognition, image classification and biosequence analysis. The basic idea of SVM is to construct a largest margin separating hyperplane to classify objects into two classes. In computational biology, a lot of problems can be considered as classification problems, for example, gene classification, protein secondary structure prediction and so on. The issues we deal with in this project belong to the category of gene classification. Thus we choose Support Vector Machine to analyze our SNP data set.

Typically, if most data points in the training set are linear separable, we only need to build a linear classifier using linear SVM.

3.5.4.1 Linear SVM

Given training set as a sequence of labeled points in n-dimensional space, the task of Linear SVM is to find an optimal separating hyperplane. Assume we get a hyperplane that separates the training data. The equation of this hyperplane is \( w \cdot x + b = 0 \). Let \( d_+ \) be the shortest distance from the hyperplane to the nearest positive examples and \( d_- \) for the nearest negative examples. The margin \( m \) of the separating hyperplane is then \( d_+ + d_- \). The aim of linear SVM is to find the separating hyperplane with the largest margin \( m \).

An example is showed in Figure 7. The solid line is the solution hyperplane, the margin is the distance between the two parallel dashed lines, the circled negative and positive examples are called support vectors. The number of the support vectors is usually small in comparison with the size of the training set. We can see that the optimal hyperplane is defined only by the support vectors and if we remove all other training points, the hyperplane will not be affected.

However, there are two limitations with using linear SVM. The first is that linear SVM can’t handle the non-separable cases. For instance, the linear can’t construct a separating hyperplane for the input data in Figure 8. The second is that linear SVM can’t efficiently eliminate the effect of noise data. In order to overcome these two shortcomings, we need to introduce non-linear SVM with kernel functions.

3.5.4.2 Non-Linear SVM

The basic idea of non-linear SVM is to map the input data points into the high-dimensional feature space by some non-linear mapping and then construct a hyperplane in the feature space.

A simple example of this process is shown in Figure 8. The kernel function should be defined in such a way that the non-separable input data can be mapped into separable features. The method for building a hyperplane in the feature space basically is the same as applied in the linear SVM. Therefore, the critical point for applying non-linear SVM is how to choose the kernel functions. Three classical kernels are widely used today, which are polynomial kernels, Gaussian kernels and sigmoid kernels.

3.5.4.3 Polynomial kernels

The general form of a polynomial kernel is defined by:

\[
K_{poly}(\vec{x}, \vec{x'}) = (s \cdot \vec{x} \cdot \vec{x'} + c)^d
\]

(4)

where \( d \) is the degree of polynomial, \( s \) and \( c \) are constant numbers.

3.5.4.4 Gaussian Kernels

The Gaussian Kernel is defined by:

\[
K_{Gaussian}(\vec{x}, \vec{x'}) = exp(-\frac{||\vec{x} - \vec{x'}||^2}{2\delta^2})
\]

(5)

where \( \delta \) is a parameter. The smaller value of \( \delta \) is suitable for the complex shape while the larger value is used for handling the smooth shape.
3.5.4.5 Sigmoid Kernels

The sigmoid kernel is defined by:

\[ K_{\text{sigmoid}}(\mathbf{x}, \mathbf{x}') = \frac{1}{1 + e^{-(\mathbf{x} \cdot \mathbf{x}' + \theta)}} \]  

where \( k \) is called gain and \( \theta \) is called threshold.

We didn’t find any documents talking about how to choose a kernel function for a specific data set. We have to try all kernel functions with different values of parameters. The SVM tool we used is SVM \( M^{\text{light}} \) which is an implementation of SVM for the problem of pattern recognition and for the problem of regression. The detailed description of SVM \( M^{\text{light}} \) is in http://svnlight.joachims.org/.

The experiments show that only applying polynomial kernel can get better classification accuracy. Using other kernels can’t get a good hyperplane for our SNP data set. The experiments also indicate that the value of polynomial degree has significant effect to the construction of hyperplane. The Figure 9 demonstrates how the change of polynomial degree affects the prediction accuracy. This result is based on all data sets we generated. The accuracy of degree 1 is equal to that of linear SVM. We can see the non-linear SVM using polynomial kernel doesn’t work with polynomial degree larger than 7. We can also get that the biggest improvement is around 14% in average using kernel functions for our SNP data set. The detailed results are presented in the section 4.

3.5.5 Clustering

The clustering is the process of grouping a set of data points with highest similarity. As we know, the clustering method is a unsupervised learning method. Since we have got the label values for all data tuples, we only need using supervised learning methods to build our models, such as SVM, decision tree and neural network we have applied. However, little work has been done to apply machine learning methods to analyze the SNP data set. So we still want to have a try to employ clustering methods and see if we can find the hidden interesting patterns in SNP data set. In this project, we choose OPTICS to analyze our SNP data set.

OPTICS is a Density-Based clustering method. The density of a data point is measured by the number of its neighborhood points. A density-based cluster is a set of density-connected objects that is maximal with respect to density-reachability. The basic idea of a density-based clustering method is that Given a value of radius, the neighborhood of each point in a cluster contains at least a minimum number of objects.

There are two reasons for us to select OPTICS. Firstly, OPTICS is a hierarchical clustering method. Unlike other clustering methods, the number of clusters is not needed to be specified. Therefore, OPTICS is very suitable to find the undiscovered patterns in the data set. Secondly, OPTICS is not sensitive to the parameter of similarity distance, which is used to define the reachability of a data point. Many clustering methods are very sensitive to this parameter, such as DBSCAN. A little bit change of this parameter will get completely different results. Usually, this parameter is empirically specified but difficult to determine for high-dimensional data sets. To overcome this limitation, OPTICS firstly sorts all data points by reachability distance and extracts the clusters based on these sorted data. Therefore, the performance of OPTICS is not affected too much by this parameter only if the value of this parameter is big enough to get a good result.

3.6 Confusion Matrices

The measure of classifier confusion is an important aspect of results obtained through learning cancer classification through the SNP dataset. The level of classifier confusion can be viewed as a "confusion matrix"; an n-by-n matrix, where n is the size of the class domain, is populated by the correct and incorrect classifications for each class. By examining the distributions of values within the confusion matrix the error bias of the classifier (if any) will become evident. A classifier with a high prediction accuracy rating may simply be an artifact of an uneven distribution of sample classes, an important flaw that will be shown clearly within the confusion matrix. The distribution of error is an equally important point from an application perspective. For learning cancer results from SNP data this importance is quite clear. The misclassification of patients as having cancer when they do not is very unfortunate. However the misclassification of patients not having cancer when they do is a far worse event. The confusion matrix indicates the individual rate of error for each type of misclassification and can be used to rate the classifier.

4. RESULTS

As mentioned earlier, due to poor preliminary results with other data sets, only the BN (breast cancer/normal) data set is included in the final analysis. Both the 0.1 and 0.5 pruning thresholds were compared.
Table 4: Naive Bayes Classifier Accuracy

<table>
<thead>
<tr>
<th>Attribute Selection</th>
<th>Sampling 0.1</th>
<th>Sampling 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-sampled</td>
<td>97.5%</td>
<td>90.5%</td>
</tr>
<tr>
<td>Sampled</td>
<td>98.2%</td>
<td>90.5%</td>
</tr>
</tbody>
</table>

Table 5: Results for given Pruning Threshold (PT) on sampled and unsampled data using a naive Bayes classifier

<table>
<thead>
<tr>
<th>Attribute Selection</th>
<th>Sampling 0.1</th>
<th>Sampling 0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-sampled</td>
<td>79.2727%</td>
<td>89.4437%</td>
</tr>
<tr>
<td>Sampled</td>
<td>79.2727%</td>
<td>88.9474%</td>
</tr>
</tbody>
</table>

4.1 Decision Trees

The decision tree results displayed in Table 4 show the best classification accuracy with using the C4.5 learning of a cancer classifier by using an information gain splitting criteria. The results are however some what misleading. In the 0.1 sampled and non sampled data sets, the learning algorithm is finding large sets of zeros and former zeros to match and use as a distinguishing feature of the normal patients. This is an artifact of the original incomplete data set as and such those results can not be considered as indicative. The results of using 0.5 data set reflect more accurately the capability of decision trees to classify cancer from this form of SNP data. The results for those data sets are not affected by the zeros as the offending normals are removed. On those data sets the decision tree performs as well as the naive bayes algorithm (shown below) as a classifier.

The top (root) branches of the decision tree can be interpreted as the most informative and distinguishing. Some of the top level tree nodes are the attributes: CDX2_2_T201C, CYP11B1_6_A_G135A, STAT2_3_C_T201C, ARO1_1_T201C, ACE_1_C201A.

These attributes are major factors in determining whether a patient has cancer within the available SNP data set.

4.2 Naive Bayes

The naive bayesian networks displayed almost the best overall classification accuracy. On the 0.5 data sets the classifier performed at about 89% accuracy and about 79% accuracy for 0.1 data. What is more interesting in these results is the distribution of the error made for each data set by the naive bayes classifier. By examining the four confusion matrices for the four data sets (Table 6, 7, 8, and 9) several observations are made. First the distribution of error for the 0.5 data sets is quite even. That is to say the naive bayes classifier is not achieving a high accuracy by simply always guessing one classification, but rather is really using the available data to predict the class. A second point of interest is that, while both the sampled and un-sampled data for the 0.1 data set had exactly the same predictive accuracy, the sampled data set error distribution was better distributed.

These two ideas can be summarized by looking at the mean squared error $\text{error}(B) + \text{error}(N)/2$ of each classification (Table 10). The mean squared error is a good additional indicator of the true abilities of a classifier.

The mean squared error for 0.5 non-sampled and 0.5 sampled do not vary much, while the 0.1 non-sampled is mean squared error is significantly higher that for the 0.1 sampled.

4.3 Neural Nets

Before delving into the predictive results of neural networks on the four primary data sets some more preliminary results. In initial neural network investigations, using a hand-selected subset of the full SNP data set, both single layer (perceptron) and multi-layer networks were experimented with. The results of these initial trials were significantly informative and gave direction to all further neural network use within the scope of the SNP data set. As shown in Tables 12 and 13 the results using both types of neural networks, with otherwise identical learning parameters, produced almost identical results. The classification accuracy with using single layer network is 85.4077% and 83.2618% with using multi-layer network.

Since the multi-layer network is a great deal more expressive in the range of classifiers it can produce the a perceptron, which can only represent linearly separating classifiers, this result leads to inference that the 'true' cancer classifier is a linearly separator. Due to this result all further experimentation with neural networks use only the single layer perceptron network type.

Table 11 summarizes the predictive accuracy of perceptrons with the four primary data sets. Three of the four data sets

4.2 Naive Bayes

The naive bayesian networks displayed almost the best overall classification accuracy. On the 0.5 data sets the classifier performed at about 89% accuracy and about 79% accuracy for 0.1 data. What is more interesting in these results is the distribution of the error made for each data set by the naive bayes classifier. By examining the four confusion matrices for the four data sets (Table 6, 7, 8, 9) several observations are made. First the distribution of error for the 0.5 data sets is quite even. That is to say the naive bayes classifier is not achieving a high accuracy by simply always guessing one classification, but rather is really using the available data to predict the class. A second point of interest is that, while both the sampled and un-sampled data for the 0.1 data set had exactly the same predictive accuracy, the sampled data set error distribution was better distributed.

These two ideas can be summarized by looking at the mean squared error (error(B)^2 + error(N)^2)/2 of each classification (Table 10). The mean squared error is a good additional indicator of the true abilities of a classifier.

The mean squared error for 0.5 non-sampled and 0.5 sampled do not vary much, while the 0.1 non-sampled is mean squared error is significantly higher that for the 0.1 sampled.

4.3 Neural Nets

Before delving into the predictive results of neural networks on the four primary data sets some more preliminary results. In initial neural network investigations, using a hand-selected subset of the full SNP data set, both single layer (perceptron) and multi-layer networks were experimented with. The results of these initial trials were significantly informative and gave direction to all further neural network use within the scope of the SNP data set. As shown in Tables 12 and 13 the results using both types of neural networks, with otherwise identical learning parameters, produced almost identical results. The classification accuracy with using single layer network is 85.4077% and 83.2618% with using multi-layer network.

Since the multi-layer network is a great deal more expressive in the range of classifiers it can produce the a perceptron, which can only represent linearly separating classifiers, this result leads to inference that the ‘true’ cancer classifier is a linearly separator. Due to this result all further experimentation with neural networks use only the single layer perceptron network type.

Table 11 summarizes the predictive accuracy of perceptrons with the four primary data sets. Three of the four data sets

Table 6: Confusion matrix for 0.1 Non-Sampled Data Set

<table>
<thead>
<tr>
<th>Classification</th>
<th>Label</th>
<th>B</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>133</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>51</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 7: Confusion matrix for 0.5 Non-Sampled Data Set

<table>
<thead>
<tr>
<th>Classification</th>
<th>Label</th>
<th>B</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>129</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>11</td>
<td>41</td>
</tr>
</tbody>
</table>

Table 8: Confusion matrix for 0.1 Sampled Data Set

<table>
<thead>
<tr>
<th>Classification</th>
<th>Label</th>
<th>B</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>115</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>33</td>
<td>103</td>
</tr>
</tbody>
</table>

Table 9: Confusion matrix for 0.5 Sampled Data Set

<table>
<thead>
<tr>
<th>Classification</th>
<th>Label</th>
<th>B</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>130</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>N</td>
<td>13</td>
<td>39</td>
</tr>
</tbody>
</table>
produce very similar accuracies of about 85%. The sampled and 0.1 data set however showed a relatively weak ability for neural net classification with 77% predictive accuracy. The likely reason for the lower performance of the sampled data sets is the reduction of distinctive values within the data. By sampling, the differences between the normal and cancer patients are reduced and the distinction is blurred. The 0.5 shows this effect to a far smaller degree because of the small amount of sampling that is performed on it.

Looking deeper into the results the confusion matrix of the 0.5 non-sampled data set (Table 14) reveals a somewhat bias split among the error for the neural network. Out of the 52 normal patients in the set, the perceptron misclassified 15 data points only an accuracy of 71%, while the cancer classification scored 91%. While the accuracy was not quite so high using the 0.5 sampled data set, the error distribution is slightly better (Table 15).

Generally due to the nature of neural networks, the models can be difficult to interpret. In the case of the single layer networks (perceptrons) it is possible to look at the weights of the input nodes and determine the largest contributors to the result.

The attributes (and associated variant type) shown in Table 16 are the largest contributors or detractors (when the weight is negative) for a patient having breast cancer, based on the neural network trained on available SNP data.

4.4 Support Vector Machines

The results from using SVM are shown in the table 17. Because using 0.5 pruning threshold removed most partially known data tuples, the sampling process doesn’t have much effect on the selection of support vectors. So the classification accuracies are the same for both sampled data set and unsampled data set. The highest accuracy is 87.27% for the

Table 10: Data Sets created by splitting on patient class.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Mean Squared Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1 Non-Sampled</td>
<td>7.12%</td>
</tr>
<tr>
<td>0.5 Non-Sampled</td>
<td>2.43%</td>
</tr>
<tr>
<td>0.1 Sampled</td>
<td>4.43%</td>
</tr>
<tr>
<td>0.5 Sampled</td>
<td>3.39%</td>
</tr>
</tbody>
</table>

Table 11: Neural Network Classifier Accuracy

<table>
<thead>
<tr>
<th>Attribute Selection</th>
<th>0.1</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-sampled</td>
<td>89.909%</td>
<td>86.319%</td>
</tr>
<tr>
<td>Sampled</td>
<td>71.4543%</td>
<td>85.2632%</td>
</tr>
</tbody>
</table>

Table 13: Confusion matrix for Multi-layer Network on Hand-Sampled Data Set

<table>
<thead>
<tr>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>N</td>
</tr>
</tbody>
</table>

Table 14: Confusion matrix for 0.5 Neural Nets Non-Sampled Data Set

<table>
<thead>
<tr>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>N</td>
</tr>
</tbody>
</table>

Table 15: Confusion matrix for Neural Nets 0.5 Sampled Data Set

<table>
<thead>
<tr>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>B</td>
</tr>
<tr>
<td>N</td>
</tr>
</tbody>
</table>

Table 16: Neural Network Contributing Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Contribution for Cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADPR1_4</td>
<td>T</td>
<td>1.3781</td>
</tr>
<tr>
<td>BDKRB2_1</td>
<td>G201A</td>
<td>Heterozygous</td>
</tr>
<tr>
<td>BDKRB2_2</td>
<td>G201A</td>
<td>Homozygous Nor</td>
</tr>
<tr>
<td>BDKRB2_2</td>
<td>C152T</td>
<td>Homozygous Nor</td>
</tr>
<tr>
<td>BDKRB2_2</td>
<td>C152T</td>
<td>Heterozygous</td>
</tr>
<tr>
<td>BRCA1_12</td>
<td>A201C</td>
<td>Homozygous Nor</td>
</tr>
<tr>
<td>BRCA1_12</td>
<td>A201C</td>
<td>Heterozygous</td>
</tr>
<tr>
<td>CYP3A_1_1</td>
<td>A201C</td>
<td>Heterozygous</td>
</tr>
<tr>
<td>CYP3A_1_1</td>
<td>A201C</td>
<td>Homozygous Var</td>
</tr>
<tr>
<td>GHRH_1_1</td>
<td>A201G</td>
<td>Heterozygous</td>
</tr>
<tr>
<td>GHRH_1_1</td>
<td>A201G</td>
<td>Homozygous Var</td>
</tr>
<tr>
<td>NFATC3_3</td>
<td>A201G</td>
<td>Heterozygous</td>
</tr>
<tr>
<td>NFATC3_3</td>
<td>A201G</td>
<td>Homozygous Var</td>
</tr>
</tbody>
</table>

Table 17: Results for given Pruning Threshold (PT) on sampled and unsampled data using a support vector machine (SVM) classifier.

<table>
<thead>
<tr>
<th>Classification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Unsampled</td>
</tr>
<tr>
<td>Sampled</td>
</tr>
</tbody>
</table>
unsampled data set using 0.1 pruning threshold. But this number is not convincing because a lot of partially known data tuples are picked up as support vectors. After using GIBB’s sampling method to fill in these unknown information, we only get 73.82% for the classification accuracy. This indicates that the unknown information are very significant to the classification. Depending on current SNP data set, we get around 80% in average for the classification accuracy.

Our experiment results show that the SVM is not the best tool for classifying our SNP data. This is a little deviant from our initial expectation since the SVM is a very new learning method and it has become one of the best tools in the computational biology. We thought the reasons are that all data types in our data set are categorial and there are only three possible values for every attribute. Because SVM uses the distance measurement to find the optimal hyperplane, applying it to our SNP data set can’t exhibit the full power of SVM.

4.5 Clustering

Unfortunately, the result of applying OPTICS is not good. We can only get one cluster no matter how we change the values of parameters. From the figure 10, we can see that only one cluster can be generated and the difference among the reachabilities of all data points is very small. It means all data points are crowded. We thought the reason for this result is somewhat the same as those in using SVM. Using the distance measurement between data tuples is difficult to separate them. Before we draw a conclusion, we also tried other clustering methods, which are CURE and ROCK. The results of using both methods are the same.

5. CONCLUSIONS

These results show that SNP data does contain sufficient information to be able to classify cancer patients. This knowledge may allow clinicians to use machine learning techniques in the future to identify individuals who are at higher risk for breast cancer. With this information, those individuals might be checked more often in order to catch tumors early.

Happily, the analysis confirms the results of earlier studies[10, 12] which identified certain genes as significant causal factors in the development of cancer. This agreement between our study and previous studies validates our the feasibility of our approach.

In particular, SNPs in BRCA2, a well-known breast cancer gene, were prominent in the analysis. The CYP gene seems to play a major role in our data set, confirming other studies[8]. The MLH tumor-suppressor gene, which disrupts DNA repair when mutated, was shown to be significant in breast cancer.

The caveat here is that the SNPs chosen are already a very select sampling of the entire genome and were selected as likely candidates for success in this study. Even in the case that all the genes may have some previously known effect, we hope that our analysis will help prioritize the most influential SNPs in a meaningful way.

The best techniques for our analysis seemed to be the decision trees and naive Bayes classifier, although some of the other classifiers were not far behind. It seems that the sampling during preprocessing was not as effective as we might have hoped. The pruning seemed a more effective way to reduce problems due to missing data while not introducing the noise that sampling seemed to.

From the perspective of aiding clinicians and researchers the decision trees and naive Bayes show a distinct advantage in that it is easier to extract causal information from these studies. For this reason, if the results of neural nets, SVMs, and clustering were only a little higher than the other two methods, we would probably still prefer the decision trees and naive Bayes.

5.1 Remaining problems and future research directions

To summarize, we are encouraged by the results we obtained in these experiments. Despite the large amount of missing data, some significant evidence was found. We anticipate that the previously known SNPs that appeared the most relevant in our analysis (such as BRCA2) will not be the most interesting part of this work. We expect that the less well characterized SNPs that appear nearly as significant will be the most interesting places for future work.

We anticipate that the results of using these methods will improve drastically in the use of a full data set. With more time, the parameters for the classifiers could certainly be improved for some increase in accuracy. We would also benefit from closer interaction with cancer biologists.

Overall, we feel that machine learning techniques will continue to be useful on SNP data sets for cancer research and beyond.

6. ACKNOWLEDGMENTS

We appreciate suggestions and support from Russ Greiner and the PolyomX data set from Brent Zanke with support from Jennifer Listgarten.

7. REFERENCES


