1 Introduction

The challenge of cancer treatment has been to identify distinct tumor types to enable the selection of specific therapies aimed to maximize efficacy and minimize toxicity. Improvements in cancer classifications have thus been central in the advances of cancer treatment. Today, most cancer classifications have been based primarily on morphological appearance of the tumor cell determined by the observation of pathologists. Although cancer classification using such method has improved over the past decades, there is still a need for a fully automated and less subjective method for cancer diagnosis. Recent studies demonstrated that DNA microarray data could provide for cancer classification at the gene expression level.

A DNA microarray is a collection of microscopic DNA spots; each typically represents the DNA sequence of a single gene, arrayed on a solid surface by covalently attachment. A typical use of the microarray is the hybridization of fluorescently or radioactively labeled cDNA extracted and polymerized from mRNA in a cell. By measuring the intensity of hybridization on each microscopic spot, DNA microarray can be used to determine the amount of mRNA expressed in the cell. Since cancer cells have different expression patterns than normal cells, given that DNA microarray has the ability to quantify gene expression level in different location and time, it has became a potential useful tool in classifying patients [CW03].

Given the nature of cancer microarray data, which usually consists of a few hundred samples with thousands of genes as features, the analysis has to be carried out carefully. Work in such a high dimensional space is extremely difficult if not impossible, not to mention the problem of overfitting. To solve these problems, a natural approach is to select relevant genes that have the highest predictive power.

In this paper, we consider several general approaches to feature selection for cancer prediction.

2 Methods

2.1 Data set

The data sets we applied feature selection methods was acquired from the published gene expression data of 117 breast cancer patients [vTvDvV^{+}02]. Of all the genes (≈ 25,000), only a small portion of genes is expected to present clinical importance. Of 117 samples, 97 samples are clinically well-annotated. In this work we aimed at selecting out genes having discriminative potential in classifying cancer cells’ progression stage. For a simplicity, [vTvDvV^{+}02] considers patients with follow-up survival years ≤ 5 yrs were considered “malignant” but those with > 5 yrs were considered “benign”. We followed the same categorization.

2.2 Feature Selection

In this section, we describe the feature selection techniques used to select the set of gene expressions that have the most predictive power in determine whether the patient exhibits the expression pattern of a benign or malignant cell type. We used three feature selection approaches and compare the performance of these approaches. To summarize briefly, our initial approach proceeds in three phases. In the first phase, we used unconditional
univariate mixture modeling to provide initial filtering of the features, and to provide a discretization for the second phase. In the second phase, we computed information gain of each feature and filter out features having low information gain. Finally, in the third phase, we used a more computationally intense procedure of Markov blanket filtering to choose candidate feature subsets that are then passed to a classification algorithm. The second approach utilized a technique known as lasso regression. Finally, the third approach we used was a simple t-test technique.

2.2.1 Approach 1: Information Gain and Markov Blanket Filtering

**Mixture model** By the nature of cDNA microarrays experiment, we can reasonably assume that the state of gene expression can be either “active” or “inactive” [XJK01]. Since these states are latent or hidden, we can infer most probable state assignment using a Gaussian mixture model; thus we can represent for each gene’s log-likelihood as

$$
l(\theta; D) = \sum_{n} \log p(x_n) = \sum_{n} \log \sum_{i \in \{0,1\}} p(z_{ni})p(x_n | z_{ni}) = \sum_{n} \log \sum_{i \in \{0,1\}} \pi_i N(\mu_i, \sigma_i^2)
$$

where \(i\) denotes latent variable assignment (cluster), \(n\) denotes \(n\)th sample, and \(z_{ni}\) denotes an indicator of \(n\)-th sample having \(i\) latent variable. All samples in each \(i\)-th cluster share the same prior \(\pi_i\) and parameters for Normal, \(\mu_i, \sigma_i\). We can easily derive the following EM algorithm. For each latent variable \(k \in \{0,1\},\)

- **E-step:**
  $$\tau_n^k \text{ def } \mathbb{E}_{p(z|x)}(z_n^k) = \frac{\pi_k N(x_n | \mu_k, \sigma_k)}{\sum_{i} \pi_i N(x_n | \mu_i, \sigma_i)}$$

- **M-step:**
  $$\arg\max_{\mu_k, \sigma_k} \mathbb{E}_{p(z|x)}(l(\mu, \sigma; D))$$

$$\mu_k = \frac{\sum_n \tau_n^k x_n \pi_k}{\sum_n \tau_n^k \pi_k}, \quad \sigma_k^2 = \frac{\sum_n \tau_n^k (x_n - \mu_k)^2}{\sum_n \tau_n^k \pi_k}$$

We iterate until the parameters (or log-likelihood) converge. The resulting parameters designate two Gaussian distributions. Then we used these two distributions to quantize data using the method shown in [XJK01], where each sample \(x_n\) is to be assigned according to \(\arg\max_z p(z_n | x_1:N)\).

**Information Gain** Once we quantized all the features, we computed information gain of each feature. The information gain is commonly used as a surrogate for approximating a conditional distribution. To compute information gain, we used the following equation:

$$I_{\text{gain}}(X_i) = H(P(Y = 0), P(Y = 1)) - \sum_{k=0}^{1} P(X_i = k) H(P(Y = 0 | X_i = k), P(Y = 1 | X_i = k))$$

The information gain provides a simple initial filter with which we can filter out less informative features for classification based on the conditional distribution approximation.

**Markov Blanket Filtering** Features that pass the information gain filter (in our case, 6000 features) are fed into a more computationally intensive feature selection technique known as the Markov Blanket Filtering [KS96]. In essence, the algorithm utilizes the idea of conditional independence to reduce feature subset. By finding a feature, \(F_i\), that is independent to other features given a small subset of features, \(M\), (or commonly known as the Markov blanket of \(F_i\)), we should be able to omit such feature without compromising the accuracy of class prediction. In Koller and Sahami’s paper, the authors defined the cross-entropy as:

$$\delta_G(F_i | M_i) = \sum_{f_M, f_i} \text{Pr}(M_i = f_M, F_i = f_i) \times D(\text{Pr}(C|M = f_M, F_i = f_i)) || \text{Pr}(C|M = f_M)$$

where \(D(P||Q) = \sum_{x} P(x) \log(P(x)/Q(x))\) is the Kullback-Leibler divergence. Since the goal is to find a small non-redundant feature subset, and those features that form an approximate Markov blanket are most likely to be more strongly correlated to \(F_i\), we construct a candidate Markov blanket for \(F_i\) by collecting \(k\) features that have the highest Pearson’s correlations with \(F_i\), where we defined \(k = 10\) in our experiment. Using this definition, we iteratively finds an approximate Markov blanket for each feature and rank the features based on their cross-entropy. The approximate algorithm is given as the following:
Initialize $G \leftarrow F$
repeat
  for each $F_i \in G$ do
    let $M_i$ be the set of $k$ features having the highest correlation with $F_i$
    compute $\delta_G(F_i|M_i)$ for each $i$
  end for
  $G \leftarrow G - F_i$, s.t. $F_i \in \min \delta_G(F_i|M_i)$
until $G = \emptyset$

We used this top $n$ ranked feature set with 10-fold cross validations to select the best feature subset that has the best cross-validation error.

### 2.2.2 Approach 2: Lasso Regression

### 2.2.3 Approach 3: T-test

The third feature selection approach we used is a simple statistical test, Welch’s t-test. The Welch’s t-test is intended for use with samples having unequal variance. In this approach, we ranked the features based on the t-test score. Since t-test test the null hypothesis that the means of the two normally distributed populations are equal, ranking the features based on t-test allowed us to find features that have the means furthest apart from each other with low variances overlap. Consequently, this allows us to keep features that have the highest predictive power for classification.

\[
t = \frac{\bar{m}_1 - \bar{m}_0}{\sqrt{\frac{s_1^2}{N_1} + \frac{s_0^2}{N_0}}}
\]

### 2.3 Classification Algorithms

We used a Gaussian Naive Bayes classifier, K-nearest neighbor classifier, and Support Vector Machine classifier in our study. In this section, we provide a brief description of these classifiers.

**Gaussian Naive Bayes Classifier** A Gaussian Naive Bayes classifier is a generative classification model. The model consists of a prior probability, $\pi_c$, and a Gaussian class-conditional density $N(\mu_c, \Sigma_c)$ for each class $c$. The probability of data point belong to class 1 (malignant) and class 0 (benign) is proportional to the following equation:

\[
P(y = 1|x, \hat{\mu}_1, \hat{\Sigma}_1) \propto \pi_1 \frac{1}{(2\pi)^{\frac{d}{2}} |\hat{\Sigma}_1|^\frac{1}{2}} \exp \left( -\frac{1}{2} (x - \hat{\mu}_1)^T \hat{\Sigma}_1^{-1} (x - \hat{\mu}_1) \right)
\]

\[
P(y = 0|x, \hat{\mu}_0, \hat{\Sigma}_0) \propto \pi_0 \frac{1}{(2\pi)^{\frac{d}{2}} |\hat{\Sigma}_0|^\frac{1}{2}} \exp \left( -\frac{1}{2} (x - \hat{\mu}_0)^T \hat{\Sigma}_0^{-1} (x - \hat{\mu}_0) \right)
\]

Given these two equations, we would classify a sample as malignant if $P(y = 1|x, \hat{\mu}_0, \hat{\Sigma}_0) \geq P(y = 0|x, \hat{\mu}_0, \hat{\Sigma}_0)$ or benign otherwise.

**K Nearest Neighbor Classification** We also used a simple $K$ nearest neighbor classification algorithm, setting $K$ equal to three. The distance metric that we used was the Pearson correlation coefficient. Since data points are most similar when they are highly correlated, we defined distance metric as the inverse of the absolute value of Pearson correlation coefficient, $\frac{1}{|p_{ij}|}$.

**Support Vector Machine with RBF kernel** A publicly available SVM classifier, LIBSVM [CL01], was used. In order to boost the performance we used RBF kernels. Also, since SVMs might be affected by a slack penalty $C$, we tuned the “magic” parameter by 10 fold cross-validation (Figure 3a), assuming all features are relevant. For our data, $C_{best} = 1.5$, but the CV error did not change dramatically. Then with this $C$ value, we further investigated the relationship between classification error (CV) and greedy features selection steps.
3 Results

3.1 Filtering Results

Mixture model As can be seen in Figure 1a, the mixture filtering considers only 6000 genes possess discriminative potentials. Yet the other features show that their two mixtures are highly overlapped thus the error probability (Eq. ??) is almost 1, i.e. $\log \epsilon \approx 0$ in Figure 1a. We neglected these features having poor predictive potentials. The number is greatly reduced comparing to initially given 22,000 genes.

Markov Blanket Figure 1c displays the values of $\delta_G(F_i|M_i)$ for each $F_i$. Genes are ordered in their removal sequence from left to right. Note that $\delta_G(F_i|M_i)$ increases from right to left. This ordering is then used for the subsequent greedy feature selection steps of the classification tasks.

3.2 Classification Results

Figure 2 shows the 10-fold cross validation error, training set error, and test set error for each of the three different classifiers. For each classification task, we train the learners with a subset of features ranging from the top one qualified feature ranked by Markov blanket filtering to top 1000 features ranked by Markov blanket filtering.
The figures show that for all classifiers, after an initial decrease in cross-validation errors, the classifiers quickly overfit the training data. In most cases, validation errors were the lowest during the first few hundred feature subset. We observed a trend of decreasing training error as more features are used for classifications but levels off as the number of features exceeds more than 300 features.

![Figure 3: SVM classification](image)

SVM shows a similar pattern (Figure 3b). Although a large number of features might guarantee a further decreased training error, the cross-validation error does not decrease after 1000 features added.

### 4 Discussion

From the cross-validation errors and test errors of the three classifier, we observed a relatively high validation and test error rate which was relatively consistent over several different random separation data into train and test set. Although this may be the nature of the dataset, we cannot rule out the poor performance is due to poor initial feature selection filtering using *unconditional mixture modeling*. In addition, we also suspect the result may be affected by an incorrect implementation of the Markov blanket filtering. In our next milestone, we will identify the cause of the poor performance (whether it’s man-made by us or nature of the data set) by reviewing the matlab code we implemented for Markov blanket filtering and implementing other feature selection techniques such as Lasso regression.
References


