Structure and Parameter Learning in Bayesian Networks with Applications to Predicting Breast Cancer Tumor Malignancy in a Lower Dimension Feature Space

Danielle Maddix
AA238 Final Project
December 9, 2016

***Abstract*—Bayesian structure and parameter learning can be used in predicting the probability of whether a breast tumor is malignant or benign. Bayesian structure learning is implemented using a K2 Search over the space of diagonally acyclic graphs (DAGs) to obtain a Bayesian structure with a higher Bayesian score than that of the Bayesian network corresponding to the Naive Bayes model. We can model the conditional probability distributions (CPDs) for the continuous observed features, as gaussians and the discrete class label as Bernoulli distributed. Parameter learning using maximum likelihood estimation (MLE) is used to learn the mean and variance for the continuous CPDs and the observed counts for the discrete probabilities. A key advantage of this method is that it can be used for feature selection to reduce the dimensionality of the space. d-separation and the chain rule are used to eliminate variables conditionally independent from the class c. Various metrics are used to compare the methods, where probabilities are weighted with less negative effect than misclassifications and a reward function is defined to see which methods have more false negatives, the more dangerous outcome in this application. Precision-recall and ROC curves are also utilized to compare the methods and see the relation between rounding these probabilities at various thresholds.**

I. Introduction

A. Problem Description

Decision making under uncertainty has a wide variety of applications in scientific computing. One particular application in the biological sciences to be investigated is in regards to predicting whether a breast tumor is malignant or benign. Even though there are predictive medical procedures that are available for diagnosis, one test may not be definitive enough and there can be error margins of either false positives or false negatives. An algorithm which could take into account many test diagnostics and make a prediction has potential to have a broad impact in the medical field. In fact, the medical literature is already becoming rich in such methods as seen in [1], [2], [3], [4], with the potential goal of patents having to undergo fewer extensive tests. It is clear that some features are very predictive, such as the size of the tumor, but only considering this feature cannot give definitive results.

The comprehensive dataset utilized is available from the Breast Cancer Wisconsin (Diagnostic) Dataset on the UC Irvine Machine Learning Repository [5]. The dataset is fairly rich in examples with $m = 569$ patients. It consists of a matrix with 32 columns, where each row consists of a patient sample. For a given row, the first such column is the patient ID and so ignored in this study and the second column is the label M for the malignant class, $c = 1$ and B for benign class, $c = 0$. The remaining $n = 30$ columns consist of the observation vector, $\mathbf{o} \in \mathbb{R}^n$. There are ten distinct continuous observations, namely the radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry and fractal dimension [5]. For each of these, the mean, standard error and worst case measurements are reported, where the first ten columns correspond to the mean, columns 11-20 correspond to the standard error and columns 21-30 correspond to the worst case measurements. The class distribution is given by 357 benign samples ($\sim$62.7%) and 212 malignant samples ($\sim$37.3%). The output is the probability that the tumor is malignant given the input data, that is $p(c_i^1 \mid \mathbf{o}_i)$ for patient i. By analyzing this dataset, we would like to determine the subset of the observed features that are the most relevant in predicting this probability.

B. Prior Work

Past work in [1], [2], [3], [4] on this dataset has been in regards to machine learning classification problems, where instead of outputting a probability displaying the uncertainty in the prediction, a label is assigned to the tumor being malignant or not. This problem reduces to computing the decision boundary between the malignant and benign sets. It was shown in [5] that the sets are linearly separable using all 30 input observed features. Moreover, the best predictive accuracy, as measured by k-fold cross validation (CV), for $k = 10$ was obtained using one separating plane in the 3-dimensional feature space of worst area, worst smoothness and mean texture. This was obtained using the common optimization method, linear programming (LP), as described in [6]. In particular, the separating plane was computed using the Multisurface Method-Tree (MSM-T), which is a classification method that solves a LP to construct a decision tree [7]. The relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes [5].

In [8], machine learning classification algorithms were implemented to compute linear decision boundaries, based on the prior literature's analysis that the data is linearly separable. The supervised learning algorithms tested were logistic regression, linear and quadratic Gaussian Discriminant Analysis (GDA) and Support Vector Machines (SVM). Logistic regression is a discriminative learning algorithm directly modeling the conditional probability $p(c \mid \mathbf{o})$ using a logistic function. The fitting parameter, $\boldsymbol{\theta} \in \mathbb{R}^{n+1}$, including the intercept terms are computed via the MLE and then an optimization algorithm is used to find the optimal $\boldsymbol{\theta}$. In GDA, a model is built for both the malignant tumors, $p(\mathbf{o}|c^1)$ and for the benign tumors, $p(\mathbf{o} \mid c^0)$. It then learns $p(c^1 \mid \mathbf{o})$ using Bayes' Rule, namely

$$p(c^1 \mid \mathbf{o}) = \frac{p(\mathbf{o} \mid c^1)p(c^1)}{p(\mathbf{o} \mid c^0)p(c^0) + p(\mathbf{o} \mid c^1)p(c^1)}. \tag{1}$$

In linear GDA, the posterior densities are assumed to be multivariate gaussians with means $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1 \in \mathbb{R}^n$ and the same covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{nxn}$, as calculated using MLE counts. The prior, $p(c)$, is assumed to be Bernoulli distributed. In the quadratic case, we have two separate covariance matrices $\boldsymbol{\Sigma}_0$ and $\boldsymbol{\Sigma}_1$. Since the denominator is the same for both $p(c^1 \mid \mathbf{o})$ and $p(c^0 \mid \mathbf{o})$, GDA assigns a positive malignant label of 1 if the numerator of $p(c^1 \mid \mathbf{o})$ is larger than the numerator of $p(c^0 \mid \mathbf{o})$. Lastly, SVM solves a convex optimization problem to maximize the distance between the points and the decision boundary. It is key to desire to maximize this distance, since the points near the decision boundary represent a higher level of uncertainty. In this study, the k-fold CV is also computed, using $k = 10$ and linear GDA receives the lowest error of 4.46%, using a large number of 29 features. Furthermore, the error, according to the various metrics, namely hold-out CV and k-fold CV decreases as the number of observed features increases until some upper bound. The absolute error counts the number of misclassifications, namely

$$\frac{\sum_{i=1}^{m_{\text{test}}} |c_i - \tilde{c}_i|}{m_{\text{test}}}, \tag{2}$$

where c_i is the exact label, $\tilde{c}_i$ is the predicted label and m_{test} is the test set size. A key limitation of these methods is that they are forced to make a classification, even if there is a high uncertainty in the probability. This can lead to a large number of false negatives or false positives. Moreover, it shows that a high dimensional feature space is required for the best results.

II. Methods

The goal of this work is to identify the important features to reduce the problem from a high-dimensional feature space to a smaller one. To do so, structure learning is implemented to learn an optimal Bayesian network and is compared to the results of Bayesian network from Naive Bayes. MLE is utilized to estimate the parameters from the network, comparing both continuous gaussian to discrete CPDs. The output of this method will be the probability of the tumor being malignant, given the observations connected to the class, c, in the Bayesian network. Let $\mathbf{o} \in \mathbb{R}^n$ denote the n observations and $\tilde{\mathbf{o}} \in \mathbb{R}^{\tilde{n}}$ contain the components of $\mathbf{o}$, which are not conditionally independent of c, as inferred from the Bayesian network. Then, using the definition of conditional probability, the desired probability is given by

$$p(c \mid o_{1:n}) = p(c \mid \tilde{o}_{1:\tilde{n}}) = \frac{p(c, \tilde{o}_{1:\tilde{n}})}{p(\tilde{o}_{1:\tilde{n}})}. \tag{3}$$

We use our inference methods to infer the joint distributions using the chain rule for Bayesian networks, namely $p(x_1, \ldots, x_n) = \prod_{i=1}^{n} p(x_i | pa_{x_i})$, where pa_{x_i} are the parents of x_i in the Bayesian network [9]. By the law of total probability, the denominator is simply $\sum_{c=0}^{1} p(c, \tilde{o}_{1:\tilde{n}})$. So $p(c, \tilde{o}_{1:\tilde{n}})$ is the only probability needed to compute using the chain rule and then we can simply normalize.

A. Naive Bayes with gaussian CPDs

In this first method, we assume that the Bayesian structure is known and follows the Naive Bayes model. The Naive Bayes assumption is that $(o_i \perp o_j \mid c)\ \forall i \neq j$. Thus, the only edges in this simplified Bayesian network are from c to each of the observed features, since the observations are conditionally independent of each other. The Bayesian network for the Naive Bayes assumption for the first 10 mean observed features is shown below.

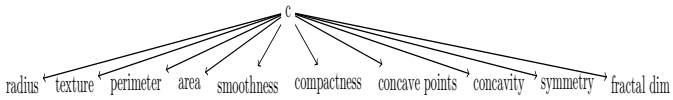

Fig. 1: Bayes Net for Naive Bayes for mean features

Since c has no parents in this model and each observation, o_i has c as its only parent, using the chain rule we obtain

$$p(c, \tilde{o}_{1:\tilde{n}}) = p(c) \prod_{i=1}^{\tilde{n}} p(\tilde{o}_i \mid c) \tag{4}$$

Thus, our desired probability is as follows:

$$p(c^1 \mid o_{1:n}) = \frac{\prod_{i=1}^{\tilde{n}} p(\tilde{o}_i \mid c^1)p(c^1)}{\prod_{i=1}^{\tilde{n}} p(\tilde{o}_i \mid c^0)p(c^0) + \prod_{i=1}^{\tilde{n}} p(\tilde{o}_i \mid c^1)p(c^1)} \tag{5}$$

Since we are assuming a specific Bayesian network for this case, there is only the parameter learning step to compute these probabilities. Since c is binary, it can only take on 2 discrete values and so it can be estimated by one independent parameter $\theta = p(c^1)$, where $p(c^0) = 1 - p(c^1)$. The parameters are computed using a training set of size m_{train}. From MLE parameter learning,

$$p(c^1) = \frac{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(c_i^1)}{m_{\text{train}}}, \tag{6}$$

where $\mathbb{1}$ is the indicator function and so $\mathbb{1}(c_i^1) = 1$, if $c_i = 1$ and 0 otherwise. Thus, this simply reduces to the counting the number of malignant samples in the training set and dividing by the total number of training samples [9].

The next step is to determine an appropriate model for the $p(\tilde{o}_i \mid c)$, for each $i = 1 : \tilde{n}$. From the success of GDA in [8],

it is clear that modeling the CPDs for the continuous features as gaussians is a good approximation. Thus, we compute the MLE parameters for a single variable gaussian, namely $\hat{\mu}_1$, $\hat{\sigma}_1^2$ and $\hat{\mu}_0$, $\hat{\sigma}_0^2$ on the subset of the data containing the malignant and benign samples, respectively. Let $m_{\text{ben}} = m_{\text{train}} - m_{\text{mal}}$. This results in the standard statistical estimates below:

$$\begin{aligned}\hat{\mu}_0 &= \frac{\sum_{i=1}^{m_{\text{train}}} \tilde{o}_i \mathbb{1}(c_i^0)}{m_{\text{ben}}}, \hat{\sigma}_0^2 = \frac{\sum_{i=1}^{m_{\text{train}}} (\tilde{o}_i - \hat{\mu}_0)^2 \mathbb{1}(c_i^0)}{m_{\text{ben}}} \\ \hat{\mu}_1 &= \frac{\sum_{i=1}^{m_{\text{train}}} \tilde{o}_i \mathbb{1}(c_i^1)}{m_{\text{mal}}}, \hat{\sigma}_1^2 = \frac{\sum_{i=1}^{m_{\text{train}}} (\tilde{o}_i - \hat{\mu}_1)^2 \mathbb{1}(c_i^1)}{m_{\text{mal}}}\end{aligned} \tag{7}$$

This formula is similar to Equation (1), where the major difference is that GDA is using MLE to compute multivariate gaussian parameters and here we are using MLE for a product of single variable gaussians. Moreover, since the output is a probability, the denominator is computed in this case.

After using the training data to learn the parameters, we loop over the test data to predict the probability of malignancy on anew patient's tumor given the observations. Thus, for patient with label c and observation $\tilde{\mathbf{o}}$, $p(\tilde{o}_i \mid c^{j\in\{0,1\}}) = \frac{1}{\sqrt{2\pi\hat{\sigma}_{ij}^2}} \exp(-\frac{(\tilde{o}_i - \hat{\mu}_{ij})^2}{2\hat{\sigma}_{ij}^2})$. Equation (5) is then utilized for the final probability.

B. Bayesian Network with gaussian CPDs for mean features

We no longer assume that the Bayesian network is known and follows a Naive Bayes model. Instead, we use structure learning to learn a Bayesian network with a higher Bayesian score than the Bayesian score corresponding to the Bayesian network for Naive Bayes. We first consider the case of finding an optimal Bayesian structure for the first 10 mean features. Since these observed features are continuous, the first step is to discretize them into a specified number of bins, with the bin width inversely proportional to the number of bins, namely $\frac{\max(\tilde{o}_i) - \min(\tilde{o}_i)}{n_{\text{bins}}}$. This was done using an in-house MATLAB code, but could also be done using the `Discretizers.jl` library in Julia. For these experiments, the optimal parameter for the number of bins was 20. Using the same Gaussian model for the CPDs and Bernoulli prior for $p(c)$, we will now compute a Bayesian structure with a higher Bayesian score than that of Bayesian network shown in Figure 1 for Naive Bayes. This can be done using the `K2GraphSearch` in `BayesNets.jl` to search over the space of DAGs for such a graph with a higher Bayesian score than the previous graph with the appropriate conditional probability distributions specified for each node. Since the output is dependent on the ordering of the nodes given, we loop over 1000 random orderings of the nodes and store the Bayesian network with the highest Bayesian score, as shown below:

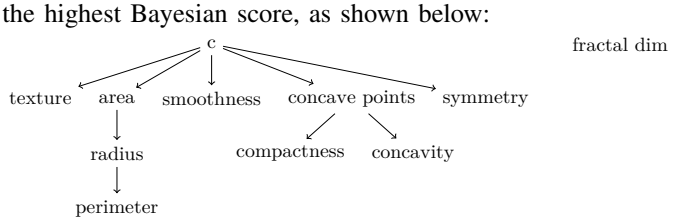

Fig. 2: Bayesian Net with gaussian CPDs for mean features

This Bayesian network can be used to determine the relevant observed features, by taking the subset of the observations with edges connected to c. Using this structure, we can perform inference using the conditional independence assumption that it encodes. Due to d-separation, several features can be eliminated from the model, reducing the dimensionality of the feature space. Thus, it is evident since the path contains a chain of nodes that given o_4, area, c is conditionally independent of o_1, radius and o_3, perimeter. Similarly, given o_7, concave points, c is conditionally independent of o_6, compactness and o_8, concavity. Furthermore, o_{10}, fractal dimension is unconnected and so is also conditionally independent of c. Thus, our model reduces to $p(c \mid o_{1:10}) = p(c \mid \tilde{o}_{1:5})$, where $\tilde{o} = (o_2, o_4, o_5, o_7, o_9)^T$. Since this subgraph follows the Naive Bayes assumption on $\tilde{o}$, Equations (4) and (5) hold and the implementation is the same from the prior subsection.

C. Bayesian Network with discrete CPDs for mean features

In this approach, we compute the Bayesian network assuming that the CPDs are discrete. As in the prior subsection, the first step is to discretize the continuous observations into 20 bins. We assume that each observation remains discrete and can assume integer values from 1 to 20. For this we use an in-house MATLAB code to find the optimal Bayesian network, given discrete distributions. We begin with a graph containing nodes $(c, o_{1:10})$ over the first mean features to compare to the method in the prior subsection. Given a random ordering of the nodes as input, we loop over the predecessors of each node and add a parent to the node that maximizes the Bayesian score [9]. We loop over 1000 random orderings of the nodes and return the DAG with the highest Bayesian score. The computed Bayesian network with a Bayesian score for this discrete model of -12,111.62 is shown below.

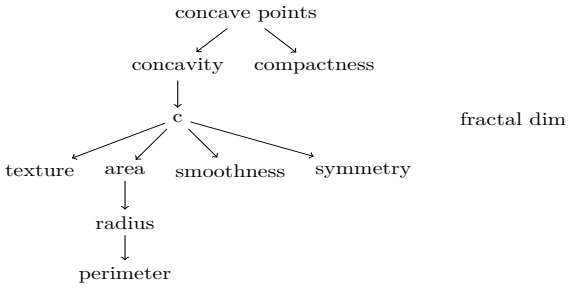

Fig. 3: Bayes Net with discrete CPDs mean features

Note that this is larger than the -13,473 Bayesian score for the discrete CPD model of the Bayes Net from Naive Bayes. Using the d-separation properties, it is evident that c is conditionally independent from every node, except for concavity, texture, area, smoothness and symmetry, reducing the dimensionality from 10 to 5. So, we have $p(c|o_{1:10}) = p(c|\tilde{o})$, where $\tilde{o} = (o_4, o_9, o_2, o_5, o_8)^T$. Using the chain rule, we get:

$$p(c, \tilde{o}_{1:\tilde{n}}) = \prod_{i=1}^{4} p(\tilde{o}_i \mid c) p(c \mid \tilde{o}_5) p(\tilde{o}_5) \tag{8}$$

Thus, our desired probability $p(c^1 \mid o_{1:n})$ is as follows:

$$\frac{\prod_{i=1}^{4} p(\tilde{o}_i \mid c^1) p(c^1 \mid \tilde{o}_5)}{\prod_{i=1}^{4} p(\tilde{o}_i \mid c^0) p(c^0 \mid \tilde{o}_5) + \prod_{i=1}^{4} p(\tilde{o}_i \mid c^1) p(c^1 \mid \tilde{o}_5)} \tag{9}$$

Note that $p(\tilde{o}_5)$ cancels from the numerator and denominator, since it does not depend on c. To perform inference, we must first learn the parameters from the network. For each i, we must compute $p(\tilde{o}_i \mid c)$. Since c is binary and $\tilde{o}_i$ can take on 20 values, this can be represented by 38 independent parameters, using the fact that probabilities sum to 1. The MLE parameters $\hat{\theta}_k = p(\tilde{o}_i^k | c)$ are simply the observed counts in the data, where $\tilde{o}_i = k$ corresponding to the appropriate malignant and benign samples, as given below:

$$\begin{aligned} p(\tilde{o}_i^k \mid c^1) &= \frac{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(\tilde{o}_i^k, c_i^1)}{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(c_i^1)} = \frac{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(\tilde{o}_i^k, c_i^1)}{m_{\text{mal}}} \\ p(\tilde{o}_i^k \mid c^0) &= \frac{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(\tilde{o}_i^k, c_i^0)}{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(c_i^0)} = \frac{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(\tilde{o}_i^k, c_i^0)}{m_{\text{ben}}} \end{aligned} \tag{10}$$

Lastly, to calculate $p(c \mid \tilde{o}_5)$, we have 20 independent parameters, since c is binary and $\tilde{o}_5$ can take on 20 discrete values. So,

$$p(c^1 \mid \tilde{o}_5^k) = \frac{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(\tilde{o}_5^k, c_i^1)}{\sum_{i=1}^{m_{\text{train}}} \mathbb{1}(\tilde{o}_5^k)} \tag{11}$$

Now that we have the necessary parameters to compute the final probability for new patients, we loop over the discretized test data and calculate the above probabilities for the discrete values of $\tilde{\mathbf{o}}$.

D. Bayesian Network with gaussian pdfs for all features

Since the gaussian CPDs performed better than the discrete model, as seen in the Results section, we compute an optimal Bayesian structure using all 30 features, assuming the gaussian model. The computed Bayesian network is given below, displaying only the nodes not conditionally independent of c.

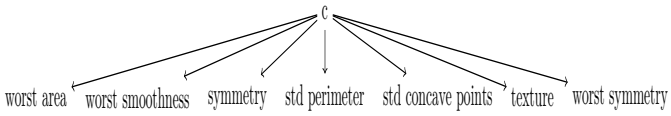

Fig. 4: Bayes Net computed using all 30 features

Worst area, worst smoothness and mean texture were selected, which were also the features cited as obtaining the best results in [5]. This resulting Bayesian network also follows the Naive Bayes assumption that the observations are conditionally independent of each other. Thus, Equations (4) and (5) also hold for $\tilde{o} = (o_{24}, o_{25}, o_9, o_{13}, o_{17}, o_2, o_{29})^T$ and the implementation follows from above. This is a new contribution from the work in [8], since it only looked at subset of features, as stored as consecutive elements in the feature vector, whereas here we have the key features selected from each category.

III. Results and Discussion

We define the following metric to compare the various methods, namely

$$\frac{\sum_{i=1}^{m_{\text{test}}} (p(c_i^1 \mid \tilde{\mathbf{o}}_i) - c_i)^2}{m_{\text{test}}} \tag{12}$$

Due to the square, it penalizes probabilities less than the maximum error for misclassifications, that is predicting 1 when $c_i = 0$ or 0 when $c_i = 1$.

For the Naive Bayes method, we investigate the optimal number of features to obtain the lowest metric. To do so, the k-fold CV is computed, which only holds out $\lceil m/k \rceil$ of the training data each time for testing and trains on the remaining portion. The result is the average of these $k = 10$ error metrics. As expected and similar to the findings in [8], this testing error decreases as the number of features is increased from 20 to 25 to a minimum of approximately 0.06 and then begins to increase again. We would like to avoid using such a large feature dimension space to obtain the minimum metric.

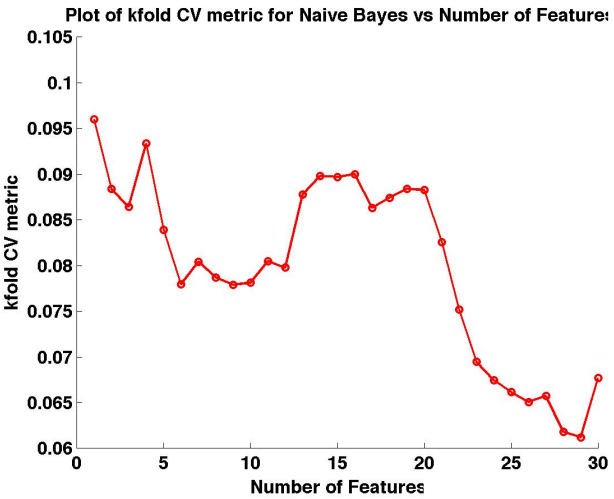

Fig. 5: Naive Bayes k-fold CV for various number of features

The results for the various methods are summarized below:

TABLE I: Probability Metric Results

	NB 10	Gaussian 10	Discrete 10	Gaussian 30
Training Error	0.0743	0.0559	0.0404	0.0456
k-fold CV	0.0781	0.0625	0.0803	0.0520

For the first three methods, we only consider the first 10 mean features and in the final method, we consider all 30 features. The first row is the training error, which is simply training on the entire dataset and then using this training set as our test set. The training error should clearly be lower than the k-fold CV in every case. The k-fold CV is more informative on the effectiveness of the method because it shows the method's potential to generalize, which is the desirable feature that given a subset of the dataset, it can properly diagnose new patients.

The discrete CPD method using the first 10 features to calculate the Bayesian network has the lowest training error,

but the highest k-fold CV test error. This illustrates one of the negative effects of using MLE parameter learning for a limited dataset. If the training set data has no occurrence of $\tilde{o}_i = k$ and the given c value, then it falsely assigns a probability of 0. This increases the number of false negatives, as indicated by the lower recall for this method in Table II. The training set, on the other hand, is representative of each observation having the discrete values. A Bayesian parameter learning approach with Beta and Dirichlet distributions to represent these counts from the dataset [9] may lead to improvement. With Naive Bayes we need a large number of features, 29 to be precise, to obtain a k-fold CV error of approximately 0.06, whereas with structure learning to learn the Bayesian network of c and the observed features, there are only 7 key features resulting in a k-fold CV error of 0.0520.

Another key component of this research is determining how to choose the appropriate class, using the probability as well as other factors specific for this application. Thus, we need to round the probabilities according to some threshold parameter, ϵ. If $p(c^1 \mid \tilde{\mathbf{o}}) \geq \epsilon$, we diagnose a malignant tumor with label $\tilde{c} = 1$ and if $p(c^1 \mid \tilde{\mathbf{o}}) < \epsilon$, we diagnose a benign tumor with label $\tilde{c} = 0$. The simplest ϵ to choose as a first attempt is 0.5. Using the k-fold CV and rounding to compute the labels $\tilde{c}$, we calculate the precision, which is a measure of the number of false positives, the recall, which is a measure of the number of false negatives and the specificity. Precision depends on the computed class label and is defined as $p(c^1 \mid \tilde{c}^1) = \frac{TP}{TP+FP}$, where TP is the number of true positives and FP is the number of false positives. Recall and specificity depend on the exact label and are defined as $p(\tilde{c}^1 \mid c^1) = \frac{TP}{TP+FN}$ and $p(\tilde{c}^0 \mid c^0) = \frac{TN}{FP+TN}$, respectively, where FN is the number of false negatives and TN is the number of true negatives.

TABLE II: Precision, Recall and Specificity for $\epsilon = 0.5$

	NB 10	Gaussian 10	Discrete 10	Gaussian 30
Precision	87.16%	93.18%	90.60%	91.16%
Recall	87.81%	87.35%	83.71%	88.37%
Specificity	94.69%	97.10%	96.04%	96.67%

Is is evident that the Bayesian network over all 30 observations from the gaussian CPDs has the highest recall. We now investigate different values of ϵ to increase the recall, since the decision should take into account the consequences of a misclassification, rather than just rounding up or down [9]. In this case, a false negative has higher consequences than a false positive because diagnosing the tumor as benign when it is really malignant could be deathly, whereas for a false positive more tests may be required to confirm. In the below figure, we vary ϵ from 0.001 to 0.999, with equally spaced intervals of 0.001 and plot the precision versus the recall for each ϵ. The small values of ϵ correspond to the highest recall and lowest precision. Thus, as ϵ is decreasing from 0.9999 to 0.001, the precision is decreasing with respect to the recall. This makes sense, since a large ϵ implies that we label more benign samples and so we do not have as many false positives and a small ϵ means that we label more malignant samples and so we do not have as many false negatives.

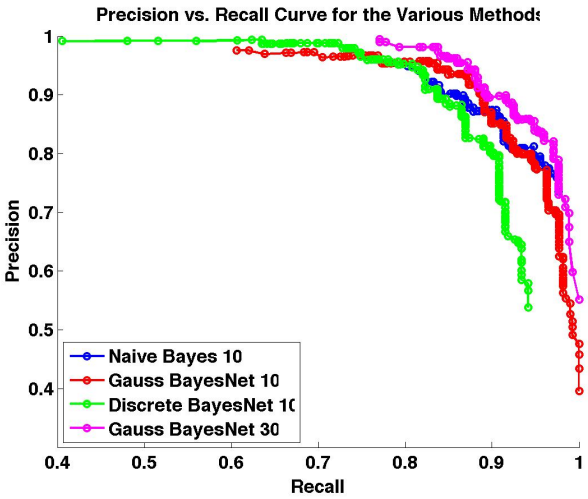

Fig. 6: Precision-Recall Curve for various ϵ

From this curve, we see that the Bayesian network computed using Gaussian CPDs with all 30 features is clearly the preferable method, since for the same recall, it has higher precision. For example, if we require a recall $\geq 90\%$ using this method, from the plot we can compute the desired tolerance with the maximum precision. For example, $\epsilon = 0.32$ corresponds to 90.82% recall and 89.83% precision. If we require a higher recall of at least 95%, then we round up to malignant for every probability greater than $\epsilon = 0.0650$. The corresponding precision is 83.96% for a recall of 95.28%. Since the precision does not drop drastically with such a low tolerance, this indicates another key feature of this method that it does not contain many uncertain probabilities around 0.5. Thus, there are several values near the extreme values of 0 and 1, unaffected by this rounding.

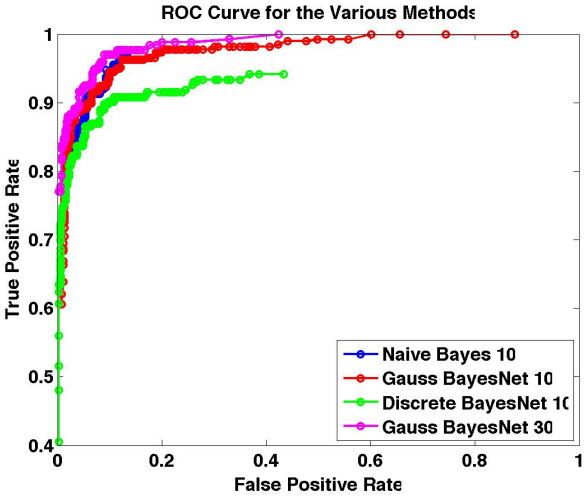

Fig. 7: ROC curve for various ϵ

The ROC curve above is another useful curve in measuring the comparative effectiveness of each method for classification. It plots the recall, that is, the true positive rate, versus $1-$ specificity, that is, the false positive rate. It is evident again that the Bayesian network computed from the 30 features with gaussian CPDs performs the best. This is indicated, since it has the steepest curve, representing a high positive rate with a low false positive. This shows that all the methods form good separators, since a random classifier would be given by the line connecting (0,0) and (1,1).

Lastly, we consider a reward system, which assigns a reward of -1 to a false negative and $-\lambda$ for $0 \leq \lambda \leq 1$ to a false positive. We vary λ and compute the plots for the various methods. We can define the following reward metric,

$$\frac{\sum_{i=1}^{m_{\text{test}}} (c_i(1 - p(c_i^1 \mid \tilde{\mathbf{o}}_i))^2 + \lambda(1 - c_i)p(c_i^1 \mid \tilde{\mathbf{o}}_i)^2)}{m_{\text{test}}}. \quad (13)$$

It is again clear that the Bayesian network from the gaussian CPDs over all 30 features has the largest reward regardless of the value of λ. This also illustrates that the initial gap between Naive Bayes with 10 features and the Bayesian network from the gaussian CPDs over 10 features is much smaller than the final, revealing that Naive Bayes has a smaller number of misclassified negatives than misclassified positives. On the contrary, the gap is large for Discrete Bayes Net and Naive Bayes initially, indicating that Discrete Bayes Net has a larger number of misclassified false negatives than false positives, due to the small gap at $\lambda = 1$ This was explained due to the MLE discrete parameter learning and lack of data for every bin. Note that for $\lambda = 1$, we get the negative k-fold CV testing error from Table I.

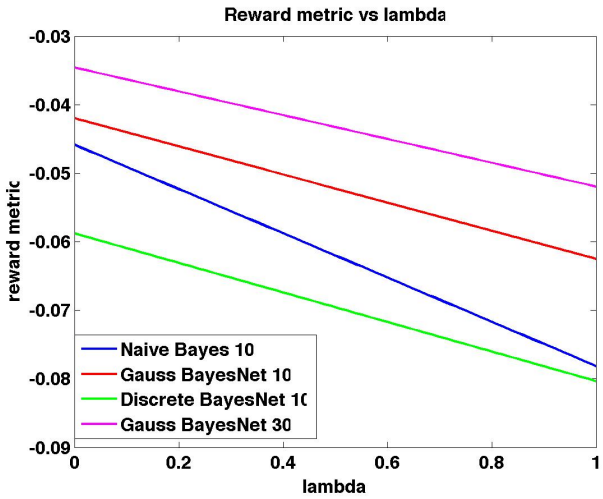

Fig. 8: λ reward metric curve

IV. Conclusion

The benefits of returning a probability, rather than a classification are clear because it identities the predictions with the highest uncertainties. Physicians can utilize these probabilities to only diagnose the disease if it is within some confidence level and for probabilities around 0.5 to recommend further testing. Various rounding and reward techniques can be investigated to determine a desired threshold to achieve a specific recall or precision. It is clear from the results that gaussians are good approximations for the conditional probability distributions of the continuous observed features. We see that utilizing structure learning can greatly improve upon the Bayesian structure corresponding to the Naive Bayes' assumption, by reducing the dimensionality of the problem and the error metric. This study also shows that using the mean features is not enough, since the worst case and standard error measurements are also valuable, by displaying the better performance of the structure and parameter learning using all 30 features, rather than just the first 10. It is also clear that using a discrete model with MLE estimates does not generalize as well to the test data for values of the discretized variables that are present in the test and not in the training data. Future work consists of experimenting with different conditional probability distributions, such as in hybrid Bayesian networks, where the variables are a mix of discrete, that is, c and continuous variables, the observed features. Potential distributions include logit and probit models.

It is evident that there are promising results in the area of applying decision making under uncertainty and machine learning to the realm of cancer diagnosis for potential use in collaboration with established medical tests and to help avoid evasive diagnostic tests on patients. This is yet another example of a promising connection between the scientific computing and medical fields.

References

[1] W. W.H., S. W.N., and M. O.L., "Machine learning techniques to diagnose breast cancer from fine-needle aspirates," *Cancer Letters*, vol. 77, pp. 163–171, 1994.

[2] M. O.L., W. W.H., and S. W.N., "Image analysis and machine learning applied to breast cancer diagnosis and prognosis," *Analytical and Quantitative Cytology and Histology*, vol. 17, pp. 77–87, 1995.

[3] W. W.H., S. W.N., H. D.M., and M. O.L., "Computerized breast cancer diagnosis and prognosis from fine needle aspirates," *Archives of Surgery*, vol. 130, pp. 511–516, 1995.

[4] M. O.L., W. W.H., S. W.N., and H. D.M., "Computer-derived nuclear features distinguish malignant from benign breast cytology," *Human Pathology*, vol. 26, pp. 792–796, 1995.

[5] UCI Machine Learning Repository Center for Machine Learning and Intelligent Systems, "Breast cancer wisconsin (diagnostic) data set," https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic), 1996.

[6] B. K.P. and M. O.L., "Robust linear programming discrimination of two linearly inseparable sets," *Optimization Methods and Software*, vol. 1, pp. 23–34, 1992.

[7] B. K.P, "Decision tree construction via linear programming," *Proceedings of the 4th Midwest Artificial Intelligence Cognitive Science Society*, pp. 97–101, 1992.

[8] M. D., "Diagnosing malignant versus benign breast tumors via machine learning techniques in high dimensions," http://cs229.stanford.edu/projects2014.html, 2014.

[9] K. M. J., *Decision Making Under Uncertainty: Theory and Application*. Cambridge, Massachusetts and London, England: MIT Press, 2015.