TAN trees for fMRI Brain Image Classification

Kit Chen
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
kchen1@andrew.cmu.edu

1 Introduction

Functional Magnetic Resonance Imaging, or fMRI, is a 3D brain imaging technique that can be used to observe which portions of the brain are activated by different types of physical or mental stimuli. The technology opens the door to new ways of studying human cognitive processes, in particular, the challenge of finding a predictable pattern in the brain images produced while the human subject is in a certain cognitive state. The historical approach to such a problem has been to keep the human subject in a particular cognitive state during some time interval by repeatedly stimulating them with the same activity (such as reading a variety of words), and then averaging the brain images collected over the interval. However, there is another, machine learning approach to the problem, that allows for the analysis of patterns in individual images without having to average over a time interval. It has been shown that trained machine learning classifiers can be employed to successfully find relevant features in fMRI brain images and link them to specific cognitive states [7]. Examples of classifiers that have already been tried include Gaussian Naive Bayes, Support Vector Machines, and k Nearest Neighbor approaches.

In particular, the Naive Bayes classifier is competitive with state-of-the-art classifiers like C4.5 despite its unrealistic and overly strong assumption that observed features are conditionally independent given a class assignment. Friedman et. al. has proven that the Naive Bayes classifier can be improved by relaxing the condtional independence assumption using a Tree-augmented Naive Bayes Network (TAN).[4] This method is shown to outperform Naive Bayes while maintaining its computational simplicity and robustness. This paper explores the use of Tree-augmented Naive Bayes (TAN) in comparison to Naive Bayes. In particular, I focus on solutions to the problem of learning Naive Bayes and TAN networks with continuous fMRI data. Both data discretization methods and Gaussian Naive Bayes/TAN methods are discussed. A coarse frequency discretization is applied to the data which was used for both Naive Bayes and TAN network classifiers. This was then compared with an implementation of the Gaussian Naive Bayes method.

2 Problem Definition

Our goal is to decode which cognitive activity is being performed, given an fMRI image taken while the human subject performs one of a set of known cognitive activities. The problem can be framed as training a machine learning classifier to output a cognitive state class given a sequence of brain scans from time t_1 and t_2.

$$f : \text{fMRI-sequence}(t_1, t_2) \rightarrow CognitiveState$$

The specific classification problem presented in this paper classifies fMRI scans into two classes: intervals during which the subject viewed a sentence then picture in which the sentence is negated and non-negated.

$$f : \text{fMRI-sequence}(t_1, t_2) \rightarrow \{Negated, NotNegated\}$$

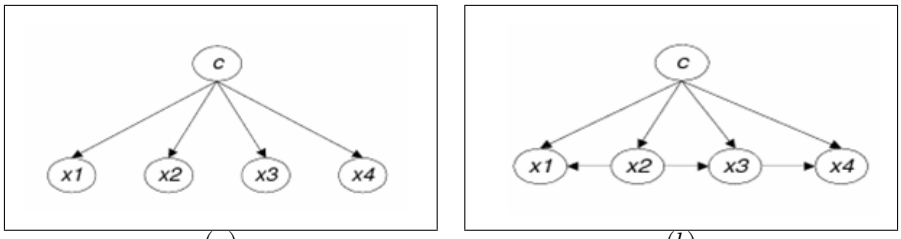
3 Proposed Method

We focus on training two flavors of bayesian classifiers: Naive Bayes and Tree-augmented Naive Bayes (TAN). These methods require discrete-valued datasets in order to compute the conditional probabilities used to do Bayesian inference. Thus, an interesting problem posed by fMRI data is how to adapt either the data or the classifier algorithms to deal with the continuous-valued data. We explore various data discretization methods, as well as the Gaussian Naive Bayes and Gaussian TAN methods.

3.1 Data Preprocessing

The fMRI data is collected from human subjects, pre-processed to remove unwanted artifacts due to head motion, signal drift, and other sources. Then all voxel activities are normalized by mean values during fixation conditions. The resulting image data is still very high-dimensional, noisy, and sparse. We hope to extract the most relevant features to reduce the dimensionality of the data before using it train the two classifiers for comparison: Naive Bayes and Tree-augmented Naive Bayes (TAN). Several feature (voxel) extraction methods have been employed by Mitchell et. al [7]. One involves selecting the n most discriminating voxels (Discrim). A second involves selecting the n most active voxels (Active). The one employed by this paper is Active feature selection as it has been shown that this method out-performs discriminative methods [7].

3.2 Naive Bayes vs. TAN Networks with Discrete



(a) (b)

Figure 1: (a) Naive Bayes network structure (b)TAN network structure

The Naive Bayes network has a very simple structure. This structure encodes the strong conditional independence assumption among attributes. The class node has no parents and is the parent node for each and every attribute node. Additionally, attribute nodes are restricted to only being child nodes 3.2(a). Thus the joint probability as represented by this structure is given as the class prior probabilities multiplied by the class conditional probabilities:

$$p(c, x_1, x_2, ..., x_n = p(c) \prod_i p(x_i|c)$$

The TAN network improves on Naive Bayes by relaxing the strong conditional independence assumption. Like Naive Bayes, the class node has no parents and is the parent for each attribute node. However, unlike Naive Bayes, each attribute can have one so-called *augmenting edge* pointing to it. These augmenting edges encode statistical depencies among attributes 3.2(b). Thus, the joint probability of this structure depends on probabilities conditioned not only on class, but also on an attribute parent node pa_{x_i}as well.

$$p(c, x_1, x_2, ..., x_n) = p(c) \prod_{i=1}^{n} p(x_i|pa_{x_i}, c)$$

From these network structures and corresponding joint distributions, we can compute class predictions $\widehat{C}(X)$

$$\widehat{C}(X) = argmax_c P(C|X) \propto P(C) \prod_i P(X|C)$$

In the case of Naive Bayes, the network structure is already known. However, TAN network structures must be learned. The procedure for doing this is based on the Chow-Liu algorithm which uses a conditinal mutual-information function between two attributes to construct the maximum-likelihood tree by finding the maximal weighted spanning tree in a graph. The steps are shown below [8].

1. Compute the conditional mutual information given C between each pair of distinct variables,
$$I(X_i; X_j|C) = \sum_{x_i, x_j, c} \widetilde{P}(x_i, x_j, c) log \frac{\widetilde{P}(x_i, x_j|c)}{\widetilde{P}(x_i)\widetilde{P}(x_j)}$$
where $\widetilde{P}(\cdot)$ is an empirical distribution (computed using the training data). Intuitively, this quantity represents the gain in information of adding X_i as a parent of X_j given that C is already a parent of X_j.
2. Build a complete undirected graph on the features $X_1, ..., X_n$ whre the weight of the edge between X_i and X_j is $I(X_i, X_j|C)$. Call this graph G_F.
3. Using Kruskal or Prim's algorithm, find a maximum weighted spanning tree on G_F. Call it T_F.
4. Pick an arbitrary node in T_F as the root and set the direction of all the edges in T_F to be outward from the root. Call the directed tree T'_F.
5. The structure of the TAN model consists of a Naive Bayes model on the joint probability $P(C, X_1, ..., X_n)$ augmented by the edges in T'_F.

3.3 Data Discretization for Naive Bayes and TAN

As it has been said, basic Naive Bayes and TAN network classifiers do not accomodate continuous-valued features. Thus, in order to test these classifiers, an effort is be made to discretize the data into N number of discrete values. One popular and simple method is range discretization in which the range of data values is divided into N range partitions and all the data values within those ranges take on the same value. The problem is that this method ignores the distribution of the underlying data causing some partitions to be over-populated and others to be empty.

In an alternate method, known as frequency discretization, an effort is made to create equally-populated partitions. Compared to range discretization, this method creates a more

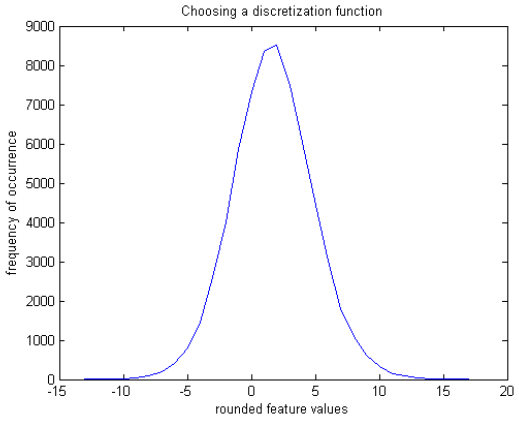


Figure 2: A plot of the frequencies of rounded data values takes on a roughly Gaussian distribution for fMRI data.

evenly-varied discretized data set. The frequency discretization method has been successfully employed with other continuous-valued datasets describing biological phenomena in the case of predicting protein fold structures [1]. With this precedent, it was the discretization method explored in this report.

In the case of fMRI data, a plot of the frequencies of rounded data values reveals a roughly Gaussian distributed frequency plot 2. The data is cleaved at the mode of the frequency plot leading to two roughly equivalent frequency intervals.

It is of note that there have been other discretization methods as suggested by [3], [5], and [6]. Most noted is a method developed by Fayyad and Irani which discretizes the data according to two criterion: one which involves finding the discretization which minimizes a recursive entropy heuristic coupled with a Minimum Description Length (MDL) criterion to limit the number of intervals produced over the space. Friedman and Goldszmidt attempt to jointly learn the discretization policy and the Bayesian network over the discretized data with a modified MDL approach. However, this is noted to be computationally costly and still in its exploratory stages.

3.4 Non-discretization Methods of Dealing with Continuous Data

Another approach to dealing with continuous data is to alter the algorithm itself to accomodate such a dataset. In this case, the data values take on a Gaussian disstribution, which naturally lends to being modeled in a Gaussian manner. Thus, we explore the Gaussian Naive Bayes and Gaussian TAN network algorithms.

The Gaussian Naive Bayes Algorithm assumes that the class conditional probabilities $\prod_i P(X_i|C)$ are Gaussian distributed with mean $\mu_{x_i|c}$ and variance $\sigma^2_{x_i|c}$. These parameters are calculated directly from the data, plugged into a Gaussian distribution, and multiplied by the class prior probabilities $P(C = c)$ to obtain the joint distribution, from which classification can be performed.

The Gaussian TAN Algorithm is almost the same as discrete-data TAN Algorithm except with slight modifications to the mutual information function for structure learning, and the parameters required to write conditional probabilities. However, once these have been obtained and the joint distribution can be written, the prediction step is exactly as before. While this method is not implemented and tested experimentally in this report, it is pre-

sented for personal interest [2].

For structure learning, the same algorithm is used as the discrete-data TAN, however, the following mutual information function is used instead:

$$I(X_i, X_j|C) = -\frac{1}{2}\sum_{c=1}^{|C|} P(C=c)log(1-\rho^2_{(i,j)|c})$$

where $\rho^2_{(i,j)|c}$ is the correlation coefficient between X_i and X_j given the class label c.

$$\rho_{(i,j)|c}) = \frac{cov(X_i, X_j|C=c)}{\sigma_{X_i|C=c}\sigma_{X_j|C=c}} = \frac{E(X_iXj|c) - E(X_i|c)E(X_j|c)}{\sigma_{X_i|c}\sigma_{X_j|c}}$$

Now for parameter training to find the full joint distribution of the Gaussian-TAN model. The full joint distribution is given by:

$$p(c, x_1, x_2, ..., x_n) = p(c)\prod_{i=1}^{n} p(x_i|pa_{x_i}, c)$$

where pa_{x_i} is the additional (non-class) parent node of x_i which we now know after learning the structure with the modified mutual information function. The conditional distribution $p(X_i = x_i|pa_{x_i}, C = c)$ is given by a Gaussian distribution given class $C = c$ as such:

$$p(X_i = x_i|pa_{x_i}, C = c) \sim Normal_c(\mu_{x_i} + a \cdot pa_{x_i}, \sigma^2_{x_i} \cdot (1-\rho^2))$$

where μ_{x_i} and $\sigma^2_{x_i}$ are the mean and variance of feature x_i,

$$\rho = \frac{cov(x_i, pa_{x_i})}{\sigma x_i \sigma pa_{x_i}}$$

is the correlation coefficient between x_i and pa_{x_i}, and

$$a = \frac{cov(x_i, pa_{x_i})}{\sigma^2_{pa_{x_i}}}$$

4 Experiments

4.1 Discretized Data with Naive Bayes and TAN networks

The implementation of TAN networks used in this project was computationally expensive, especially in the calculation of mutual information between each parent and child nodes, and in the evaluation of the TAN network. The datasets had to be heavily restricted to allow for some results to be obtained in time for this report to be completed. I restricted the datasets to be only the top $N = 5, 10, 15$ Active Voxels. I also discretized coarsely into two partitions, in other words, I created binary data sets from the continuous data.

Then, the data was split 20-80 into a test and training set respectively. The instances chosen for the test set were drawn randomly and the classifiers were trained on the remaining instances, which made up the training set. Then the error for TAN and Naive Bayes was computed. This was repeated for fifty trials and then the errors were averaged to form the final classification error associated with the network. This approach in which test/train

instances were chosen randomly allowed for some variations in TAN network and Naive Bayes errors so that they could be compared on a trial by trial basis as well.

As expected, several trials gave high (greater than 50%) classification errors for TAN and Naive Bayes. However, there were also many trials with surprisingly low classification errors (less than 50%). The errors averaged over the fifty trials came out to be less than 50% over all suggesting that even with the limited number of features and the coarsely discretized data set, the predictions obtained from classifiers were better than random guessing. While Naive Bayes had a lower over-all average error that was lower than error for TAN networks, when the errors for both classifiers were compared on a trial by trial basis, the TAN algorithm outperformed Naive Bayes about 50% of the time, meaning for data this coarse, it didn't really make much of a difference to use TAN instead of Naive Bayes. Classification errors turned out to be relatively uniform across human subjects, perhaps also due to the coarse-discretization, which eliminated variations among subjects.

A table comparing the classification error (in percent) obtained for TAN and Naive Bayes Networks trained with 5, 10, and 15 Active Voxel features.

# Active Voxels	**TAN**	**Naive Bayes**
5	45.25	38
10	44.72	40.42
15	44.5	41.48

Firstly, these results show that even under the circumstances of extremely sparse and coarsely discretized data, both TAN and Naive Bayes algorithms are robust enough to extract some information from the data to provide predictions that are better than just random guessing. As for revealing differences between TAN and Naive Bayes, we would have expected TAN networks to provide more accurate predictions compared to Naive Bayes because of the relaxed conditional independence assumptions. However, these results suggest that for this coarse frequency discretization method, it is unclear which algorithm is better, and that if anything, it might be better to hold a strict conditional independence assumption instead of a more relaxed assumption. The classifiation error for Naive Bayes grows as the number of features selected increases as predicted by previous work [7]. The classification error doesn't seem to change much for TAN networks. This is somewhat expected since there isn't a huge difference between using 5, 10 or 15 active voxels. A greater difference would be expected if the number of active voxels used varied widely (say 5 to 1000) as shown in the next selection.

4.2 Active Feature Selection

In the section above comparing Naive Bayes and TAN networks on sparse and coarse data, it was stated that to a degree, having a larger number of Active features increases the classification error. This is consistent with [7] which states that reducing the number of features increases classification accuracy. To check this, I took the continuous data set, selected an increasing number of voxels from 5 to 1000 and plotted the error of Gaussian Naive Bayes classifier. The results are shown in 3. It makes intuitive sense that you need a certain number of relevant voxels to do classification, but that using more than that will simply reduce classification accuracy. Thus, I expected the classification accuracy to increase up to a certain number of features selected and then decrease after a threshold. It turns out this threshold seems to happen at around 500 voxels selected. The number of voxels selected for the Naive Bayes and TAN classifiers in the previous section fell well below that (in the 5-15 voxel range). Thus, I probably shouldn't take the increasing classification error for Naive Bayes to mean something. And it explains why the TAN classifiers don't really decrease in accuracy as the number of active voxels selected increases.

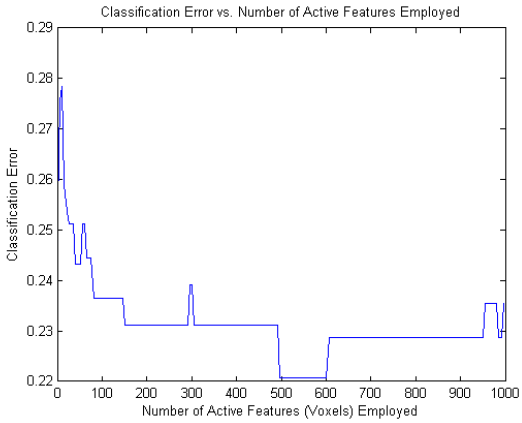


Figure 3: Classification error as a function of active (Voxels) features employed.

4.3 Gaussian Naive Bayes

The implementation of Gaussian Naive Bayes was computationally faster than discrete-data Naive Bayes, and could take advantage of the full range of values within continuous data. The classification error rates for Gaussian Naive Bayes turned out to be much lower compared with error rates of discrete Naive Bayes using the same number of selected Active features. It might have been that Gaussian TAN would have had reduced classification errors in comparison to normal TAN.

5 Conclusions

A very-coarse frequency-based discretization method was chosen to adapt the continuous data to Naive Bayes and TAN networks. It is seen that both Naive Bayes and TAN trees are robust algorithms that can decode cognitive states from even roughly-processed data. It is also shown that when data is this rough, it might be that Naive Bayes's strong conditional independence assumption may cause it to be a better predictor of cognitive state than TAN networks.

This report also reveals a Gaussian nature in the fMRI data that may lend naturally to using methods such as Gaussian Naive Bayes and Gaussian TAN networks. This algorithm is shown above but not explored experimentally.

Acknowledgments

Joseph Gonzalez (TAN help), Sue Ann Hong (TAN help), Andreas Krause (latex help), Indrayana Rustandi

References

[1] A. Chinnasamy, W.K. Sung, and A. Mittal. Protein structure and fold prediction using tree-augmented bayesian classifier. *Pacific Symposium on Biocomputing 9:387-398*, 2004.

[2] Ira Cohen, Nicu Sebe, Ashutosh Garg, Lawrence S. Chen, and Thomas S. Huang. Facial expression recognition from video sequences: temporal and static modeling.

[3] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features.

[4] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 1997.

[5] Nir Friedman and Moises Goldszmidt. Discretizing continuous attributes while learning bayesian networks.

[6] Nir Friedman, Moises Goldszmidt, and Thomas J. Lee. Bayesian network classification with continuous attributes: Getting the best of both discretization and parametric fitting.

[7] Tom M. Mitchell, Rebecca Hutchinson, Radu S. Niculescu, Francisco Pereira, Xuerui Wang, Marcel Just, and Sharlene Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004.

[8] Ajit Singh. Tree-augmented naive bayes. *Homework 2 Problem 7 of Probabilistic Graphical Models*, Fall 2006.