TAN network classifers for fMRI data

Michael P. Ashley-Rollman
mpa@cs.cmu.edu

Lucia Castellanos Pérez-Bolde
lucia@cmu.edu

Abstract

The goal of this work is to try a new solution for the classification task of predicting when a subject is reading a sentence versus perceiving a picture using brain imaging data (fMRI). fTAN, an algorithm that both relaxes the conditional independence assumptions from a traditional Naive Bayes classifier, and assumes continuous variables, is implemented and compared to other methods.

1 Introduction

A particular portion of the generic problem of decoding mental states from brain activity in humans is addressed in this work. The aim is at trying a new approach for the classification task of predicting when a subject is reading a sentence versus perceiving a picture read from brain imaging data (fMRI).

Previous attempts have used Gaussian Naive Bayes classifiers, support vector machines, and k-nearest neighbors. In this work some conditional independence assumptions of the Gaussian Naive Bayes classifier are removed, in a way that each attribute is assumed to be conditionally independent of every other attribute, given the class variable and one other attribute. This probabilistic structure corresponds to the tree augmented naive Bayes (TAN) approach [1], where a restricted Bayes network is built with the property that each attribute node has got at most two parents, namely the class variable and another attribute.

fMRI data is continuous, extremely high dimensional, sparse, and noisy. The learning scheme used to analyze these data must take into account such characteristics.

Specifically tailored feature selection methods have been successfully used to reduce dimensionality and noise before [2]. In this work these methods are employed as a preprocessing step for the data.

Continuous attributes can not be incorporated directly into the original TAN algorithm [1] since this model assumes discrete variables. There are two ways to tackle this problem: either discretize the data before applying the algorithm, or make some assumptions about the distribution of the features and incorporate the distributions into the learning scheme. In particular, conditional Gaussian distributions can be used to deal with continuous variables avoiding a discretization step. Intuitively, this latter approach is preferred over discretizing the data because discretization implies loss of information. In this work the conditional Gaussian assumptions are incorporated as suggested in [3].

In section 2 a brief overview of previous work in decoding mental states from brain imaging data and of the theory on bayesian network classifiers is exposed. In section 3 a detailed description of the process to analyse the data used in this work is presented. In section 4

the experiment design is explained and in 5 the results are summarized. Finally, in section 6 our conclusions are reported and an outline of future work is presented.

2 Previous work

2.1 fMRI data analysis

Haynes and Rees [4] review gives an overview of the work done for decoding mental states from brain imaging. However, with views to the particular problem addressed in this work, Mitchell's *et al.* work [2] is more representative because they showed three different fMRI studies demonstrating the feasibility of training classifiers to distinguish a variety of cognitive states based on single-interval fMRI observations. In particular, they trained classifiers to categorize whether a subject was being presented a sentence or a picture during a particular time interval in a fMRI session. In fact, they learnt a function of the form:
f:fMRI-sequence$(timeinterval) \rightarrow \{Picture, Sentence\}$.

As a pre-processing step, they applied some *feature selection* methods that improved classification accuracy in all of their studies. The success of these methods consisted in taking into consideration the signal when the subject is neither viewing a picture nor a sentence, but simply fixating on the screen data. They called these methods *activity based feature selection*.

Among the most successful methods was the one that selected the n most active voxels (*Active*), that consists in selecting voxels according to a t-test. The idea is to compare the voxel's fMRI activity in examples belonging to a specific class to its activity in examples belonging to fixation periods. Then the n voxels with greatest t statistic are selected. n is selected to minimize the mean error of all single subject classifiers trained for the specific study. In the *picture versus sentence* study $n = 240$ features where selected.

The method n most active voxels per region of interest (*roiActive*) is similar to *Active* but ensuring that voxels are selected uniformly from the user specified regions of interest within the brain. In the case of *picture versus sentence* study, these regions of interest are: *CALC*, *LIPL*, *LT*, *LTRIA*, *LOPER*, *LIPS* and *LDLPFC*. A method based on the average of active voxels per ROI was also tried.

Mitchell *et al.* built Gaussian Naive Bayes (GNB) classifiers, linear Support Vector Machine (SVM) and k-Nearest Neighbors. They found that Gaussian Naive Bayes (GNB) and linear Support Vector Machine (SVM) classifiers outperform k- Nearest Neighbors in all three studies.

In this work a TAN classifier, that relaxes the Gaussian Naive Bayes assumptions, is tried out with views to at least yield competitive results with the two leading classifiers.

2.2 Bayesian Networks and Probabilistic Graphical Models

Bayesian Networks classifiers based on probabilistic graphical models are described in detail in [1] and have proven to be effective. A Bayesian Network is a directed acyclic graph of nodes representing variables and arcs representing conditional independence relations between the variables. Bayesian Networks assume that each random variable follows a conditional probability function given a specific value of its parents. These conditional probability functions are usually assumed to be multinomial, that is, the networks handle discrete values.

In a problem setting where there are continous variables, there are two ways of proceeding: discretize the data in advance, and consequently risk lossing some information; or incorporate the continuous variables in the learning scheme by assuming that continous

variables are sampled from a Gaussian distribution. The latter kind of network is known as a conditional Gaussian network.

A solution to a classification task can be obtained considering the Bayes rule together with the fact that probabilistic graphical models can be used to encode the joint distribution among the predictor variables, based on the conditional independence represented by the graph structure. Indeed, a general classification problem that considers only predictor variables and that would like to be addressed in the framework of a conditional Gaussian network can be described as follows [3].

Let C be the class variable, and $\{X_i\}_{i=1}^n$ the continuous predictor variables. Consider that the class variable C is the root of the graph. Thus, if Π_i denotes the set of parents of variable i, then $\{C\} \subseteq \Pi_i$ for all i.

In order to classify an instance $\mathbf{x} = (x_1, ..., x_n)$ the class with the highest a posteriori probability $P(c|\mathbf{x})$ should be selected. The classification process can be done in the following way:

$$P(c|\mathbf{x}) \propto p(c, \mathbf{x}) = P(c)p(\mathbf{x}|c) = P(c)\prod_{i=1}^{n} p(x_i|\pi_i)$$

Where π_i denotes the value of Π_i. Also:

$p(x_i|\pi_i)$ is distributed $N(m_{i|c}, v_{i|c})$ and:

$$m_{i|c} = \mu_{i|c} + \sum_{j=1}^{n_i} \beta_{ij|c}(x_j - \mu_{j|c})$$

$$v_{i|c} = \frac{\left|\Sigma_{X_i,\Pi_i'|c}\right|}{\left|\Sigma_{\Pi_i'}|c\right|}$$

$\Pi_i^{'}$ is the set of continuous predictors that are parents of X_i, *i.e.* $\Pi_i^{'} = \Pi_i \backslash \{C\}$. $\Sigma_{S|c}$ is the covariance matrix of the set of variables S conditioned to the class value $C = c$. $\beta_{ij|c}$ is the regression coefficient of X_i on X_j conditioned on the class value $C = c$, and is defined as $\beta_{ij|c} = \frac{\sigma_{ij|c}}{\sigma^2_{j|c}}$. $\sigma_{ij|c}$ is the covaraiance between the variables X_i and X_j conditioned to c, and $\sigma^2_{j|c}$ is the variance of X_i conditioned to c.

3 Proposed Method

3.1 Theoretical approach

The *tree augmented Bayesian network structures (TAN structures)* are a type of Bayesian network that loosen the conditional independence assumption made by Naive Bayes structures, in the sense that TAN structures allow probabilistic dependencies among variables (*predictors*). TAN structures consist of graphs with edges from the class variable to the predictors, and with edges between predictors taking into account that the maximum number of parents of a variable is two: another variable and the class.

Friedman *et al.* [1] proposed the TAN algorithm to train TAN structure based classifiers. This algorithm only works with discrete features. There has been other work that propose a pre-discretization step to deal with continous variables, however this step usually results in lose of information.

Pérez *et al.* [3] proposed the *filter tree augmented naive Bayes (fTAN)*, that adapts the original TAN algorithm [1] to continuous variables by defining the *mutual information* between two continuous variables that are jointly distributed given C as a bivariate normal. Indeed, if C is a multinomial random variable, and the joint density function of variables X_i and X_j conditioned to $C = c$ follows a bivariate normal distribution with a vector of means $\mu_{ij|c}$ and a covariance matrix $\Sigma_{ij|c}$, then the mutual information between variables X_i and X_j conditioned to C is such that:

$$I(X_i; X_j|C) = -\frac{1}{2} log(1 - \rho_c^2(X_i, X_j))$$

where $\rho_c(X_i, X_j) = \frac{\sigma_{ij|c}}{\sqrt{\sigma_{i|c}^2 \sigma_{j|c}^2}}$ is the correlation coefficient between X_i and X_j conditioned to the value $C = c$.

In this manner the fTAN algorithm can be stated in an analogous way to the original TAN algorithm [1] as follows:

1. For every pair of attributes $\{X_i, X_j\}$ where $i, j \in \{1, 2, .., n\}$ and $i \neq j$ obtain $I(X_i; X_j|C)$.
2. Build a complete undirected graph with the set $\{X_i\}_{i=1}^n$ as the set of vertices. Assign a weight $I(X_i; X_j|C)$ to the edge connecting X_i with X_j for all $i, j \in \{1, 2, .., n\}$ and $i \neq j$.
3. Build a maximum weighted spanning tree.
4. Select one root variable and set the direction of all edges to be outward from it, obtaining a directed tree.
5. Add a vertex C and add an arc from C to each variable X_i.

A TAN model built like this is guaranteed to maximize the loglikelihood of the data [1]. However, it is important to note that the structural likelihood maximization does not necessarily imply a predictive error minimization.

The fTAN algorithm constructs a complete TAN structure, meaning that all variables are included in the structure, all possible conditional independence assumptions are represented and no more dependencies can be allowed. This may imply that some redundant variables and irrelevant edges could be added [3].

The computational cost of fTAN is polynomial in the number of variables, that is, it holds the Naive Bayes's computational simplicity.

Pérez *et al.* [3] showed empirically that even if the data sets do not obey the Gaussian distribution assumption, these methods perform competitively well compared to others.

3.2 Solution in context

In this work the fTAN algorithm was implemented, and tried in the framework of *picture versus sentence* problem in fMRI data. This algorithm deals with the continous data from the fMRI measurements and relaxes the Naive Bayes classifier assumptions.

To deal with the noise and high dimensionality of the data, the analysis included a preprocessing step of feature selection as explained in section 2.1. Cross-validation is used with views to deal with sparsity of the data, that is, in order to maximize the information extracted from the data sets. Since variables are continuous, it is assumed that the variables present a Gaussian distribution.

Table 1: fTAN algorithm errors

FEATURE SELECTION	Avg.	4799	4820	4847	5680	5710
Active(240)	0.19	0.06	0.14	0.32	0.16	0.28
roiActive(240)	0.27	0.08	0.28	0.40	0.26	0.34
roiActiveAvg(120)	0.09	0.10	0.12	0.06	0.12	0.06

Table 2: Time series Gaussian Naive Bayes errors

FEATURE SELECTION	Avg.	4799	4820	4847	5680	5710
All features	0.20	0.20	0.20	0.20	0.20	0.20
Active(240)	0.14	0.08	0.12	0.14	0.14	0.20
roiActive(240)	0.19	0.18	0.18	0.20	0.18	0.20
roiActiveAvg(120)	0.05	0.08	0.04	0.04	0.06	0.04

4 Experiments

The experiments were run on the brain imaging data of subjects 4799, 4820, 4847, 5680, and 5710 individually using a specific variable selection method, a particular classifier and a fixed data presentation.

Three different classifiers were employed in the study, namely Gaussian Naive Bayes classifiers, Support Vector Machines and fTAN. The feature selection methods considered were *Active*, *roiActive* and *Average roiActive*. These methods were also employed by Mitchell *et al.* [2]. In addition, for the Naive Bayes classifiers and the Support Vector Machines, all of the features were also considered. In the fTAN algorithm using all the features is infeasible due to memory constraints.

The original data sets for each subject consisted of a series of 54 time point measurements per voxel. Due to memory constraints, in order to apply the fTAN algorithm the 54 time points were averaged and a unique measurement per voxel was considered. With views to fairly compare the algorithms, in addition to considering the time series data for the Naive Bayes classifier and for the Support Vector Machines, the averaged data was also tried in these two classifiers.

All experiments were run with cross-fold validation leaving out one example on each fold.

5 Results

The fTAN algorithm implementation was written within the existing fMRI `Matlab` framework.

In Table 1 the results of applying the fTAN algorithm to the data averaged over the 54 time points are shown. In Table 3, the corresponding Gaussian Naive Bayes classifier results are presented; whereas in Table 2 the algorithm was applied to the time series data. Table 4 contains the Support Vector Machine classifier results on the time series data, and Table 5 on the averaged data.

Tables 6, 7, and 8 show a comparison of the classifiers when Active, roiActive, and roiActiveAverage feature selection is used respectively.

Table 3: Averaged Gaussian Naive Bayes errors

FEATURE SELECTION	Avg.	4799	4820	4847	5680	5710
All features	0.14	0.14	0.06	0.14	0.18	0.18
Active(240)	0.09	0.12	0.08	0.08	0.08	0.10
roiActive(240)	0.08	0.14	0.08	0.04	0.10	0.06
roiActiveAvg(120)	0.10	0.12	0.08	0.06	0.10	0.12

Table 4: Time series Support Vector Machine errors

FEATURE SELECTION	Avg.	4799	4820	4847	5680	5710
All features	0.12	0.14	0.12	0.12	0.10	0.14
Active(240)	0.06	0.10	0.04	0.04	0.06	0.08
roiActive(240)	0.08	0.12	0.04	0.06	0.06	0.12
roiActiveAvg(120)	0.04	0.14	0.04	0.02	0.00	0.02

Table 5: Average Support Vector Machine errors

FEATURE SELECTION	Avg.	4799	4820	4847	5680	5710
All features	0.028	0.06	0.02	0.02	0.02	0.02
Active(240)	0.04	0.06	0.04	0.02	0.04	0.04
roiActive(240)	0.04	0.02	0.06	0.04	0.06	0.02
roiActiveAvg(120)	0.084	0.06	0.14	0.10	0.06	0.06

6 Conclusions

6.1 Discussion

In Table 8 fTAN is shown to be competitive in the averaged case against the other classifiers in the same modality, in fact, it beats the Averaged GNB classifiers on average; is the best on two of the five subjects; and, it beats or ties each of the other classifiers in three subjects. If fTAN were extended to support more random variables in the implementation it would likely be competitive on the time series data as well.

While the fTAN classifier performed well with roiActiveAvg feature selection method, it performed poorly with Active and roiActive as shown in Table 1 . Thus this classifier is affected by the feature selection methods differently than those studied in [2] where Active feature selection method yielded the best results. It is, however, interesting to note that fTAN was able to cut the error of GNB in half and match the SVM results for one subject with Active feature selection. It is also interesting to note that the variance of the results across the subjects is much higher for the fTAN algorithm than the other classifiers when using Active feature selection as shown in Table 6. It is interesting to note that fTAN performed consistently well on subject 4799 regardless of the feature selection method applied. This implies that fTAN is more sensitive to the training data it is provided than the other algorithms and demonstrates that it is competitive when given ‘good’ data. It is our conjecture that the sparsity of the training data results in a poor estimation of the dependency of the random variables on each other resulting in trees of questionable validity. When the tree does a good job of approximating the dependency relations the classifier is competitive and when the tree is less accurate the classifier does poorly.

Table 6: Classifiers with *Active* Feature selection

Classifiers	Avg.	4799	4820	4847	5680	5710
Averaged fTAN	0.19	**0.06**	0.14	0.32	0.16	0.28
Averaged GNB	0.09	0.12	0.08	0.08	0.08	0.10
Averaged SVM	**0.04**	**0.06**	**0.04**	**0.02**	**0.04**	**0.04**
Time Series GNB	0.14	0.08	0.12	0.14	0.14	0.20
Time Series SVM	0.06	0.10	**0.04**	0.04	0.06	0.08

Table 7: Classifiers with *roiActive* Feature selection

Classifiers	Avg.	4799	4820	4847	5680	5710
Averaged fTAN	0.272	0.08	0.28	0.40	0.26	0.34
Averaged GNB	0.084	0.14	0.08	**0.04**	0.10	0.06
Averaged SVM	**0.04**	**0.02**	*0.06*	**0.04**	**0.06**	**0.02**
Time Series GNB	0.188	0.18	0.18	0.20	0.18	0.20
Time Series SVM	0.08	0.12	**0.04**	0.06	**0.06**	0.12

Table 8: Classifiers with *roiActiveAvg* Feature selection

Classifiers	Avg.	4799	4820	4847	5680	5710
Averaged fTAN	0.092	0.10	0.12	*0.06*	0.12	*0.06*
Averaged GNB	0.096	0.12	*0.08*	*0.06*	0.10	0.12
Averaged SVM	*0.084*	**0.06**	0.14	0.10	*0.06*	*0.06*
Time Series GNB	0.052	0.08	**0.04**	0.04	0.06	0.04
Time Series SVM	**0.044**	0.14	**0.04**	**0.02**	**0.00**	**0.02**

6.2 Future work

A first expansion of the current project is to do experiments with a classifier for multiple subjects. The results of such experiments could be affected by differences on the intensity of fMRI responses to stimuli or on the spatial-temporal patterns of fMRI activation accross subjects. It is yet unclear if different brains present such a similar behaviour that implies similar activation patterns, but some evidence in [2] suggests that building classifiers in a set of subjects can be used to analyse novel subjects data. It would be interesting to try this expansion on the fTAN algorithm.

fTAN builds a complete TAN structure adding arcs in order of their conditional mutual information. However this algorithm presents three drawbacks. In the first place, the structural likelihood maximization performed by fTAN does not imply a predictive error minimization. In the second place, due to the completeness of the built TAN structure some redundant variables and irrelevant arcs could be included. Finally, building a maximum spanning tree requires huges amount of memory for problems like time series fMRI data such as the one considered here.

To tackle the first two problems Pérez *et al.* [3] suggested the *wrapper tree augmented naive Bayes* (wTAN) algorithm. This algorithm obtains a TAN structure, either complete or incomplete, by greedily searching in the space of allowed structures using as optimization function the estimated classification accuracy.

The third drawback can be solved as follows. Step 1 in the fTAN algorithm may consume

too much memory and the problem might be intractable, for instance, in the time series fMRI problem the number of features to consider in the *Active* case is $240 \times 54 = 12,960$. In theory, the information gain for each pair of features in the variable set needs to be computed. Instead, the suggestion is to substitute steps 1 to 3 in the original fTAN algorithm as follows. Let T be a graph initially empty; features represent vertices and information gain between features, represent the weights of the edges. *(1)* Select an arbitrary feature (a vertex) i; *(2)* compute the information gain between features i and s for all s, take this value as the weight between i and s that labels an edge going from i to s; *(3)* select the edge corresponding to weight $w_{i,k}$ such that $w_{i,k} = max_j \{w_{i,j}\}$ and add it to T; *(4)* keep in the queue the $n-1$ edges corresponding to the other $w_{i,j}$ weights such that $j \neq k$; *(5)* compute the information gain between feature k and every other feature s, and list these values $w_{k,s}$; *(6)* make comparisons between $w_{k,s}$ and $w_{i,s}$ and keep in the queue only $max \{w_{k,s}, w_{i,s}\}$ for each s; *(6)* pick the edge corresponding to the highest weight and consider the vertex that is not in T to apply the procedure as before in an analogous manner (repeat from *(5)*).

This procedure is equivalent to steps 1 to 3 in the original fTAN algorithm and needs only to keep $n-1$ values in memory at each time point, instead of a $n \times n$ matrix containing the information gain for each pair of features. Therefore in the problem studied in this work, a modified fTAN version with the above mentioned variant would allow the fTAN algorithm to classify the fMRI time series data without memory problems.

References

[1] Nir Friedman, Dan Geiger, and Moisés Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997.

[2] Tom M. Mitchell, Rebecca Hutchinson, Radu Stefan Niculescu, Francisco Pereira, Xuerui Wang, Marcel Just, and Sharlene Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57(1-2):145–175, 2004.

[3] Aritz Pérez, Pedro Larrañaga, and Iñaki Inza. Supervised classification with conditional gaussian networks: Increasing the structure complexity from naive bayes. *International Journal of Approximate Reasoning*, 43(1):1–25, 2006.

[4] John-Dylan Haynes and Geraint Rees. Decoding mental states from brain activity in humans. *Nature Reviews Neuroscience*, (7):523–534, 2006.