CSE512
Machine Learning
Project Presentation

Learning Cognitive states using Brain fMRI

Sriganesh N(109928706)
Vasudev B(110166551)

Motivation and Objective

OBJECTIVE:

Identify subject's cognitive state - i.e. whether they are viewing an image or sentence - based on fMRI images

MOTIVATION:

Less Amount of Preprocessing.
High Dimensional data. [Feature Selection].
Learn different algorithms based on different assumptions.
Comparative study between these algorithms.

Dataset

- Dataset consists of 54 trials.
- During each trial the subject is shown two stimuli - an image and a sentence .
- fMRI images captures the blood-oxygen level of specific regions of the brain.
- Each image contains ~5000 voxels.

Feature Selection (Voxels)

Two selection criteria were used to restrict the features:

1. **Voxels based on ROI (Regions of Interest)**

- Filter the voxels belonging to specific anatomic regions of interest. [Domain Knowledge]
- Select 7 ROIs best suited to distinguish the cognitive state of the subject.
- This reduced the set of features to ~1700.

Feature Selection(Voxels)

2. n-most active voxels

- Perform *t*-test to compare activity between class stimulus and the rest period.
- Top 100, 200, 300, 400 and 500 voxels are selected.
- Perform pairwise-time averaging between adjacent fMRI-samples to reduce noise in the data.

Classifiers Used

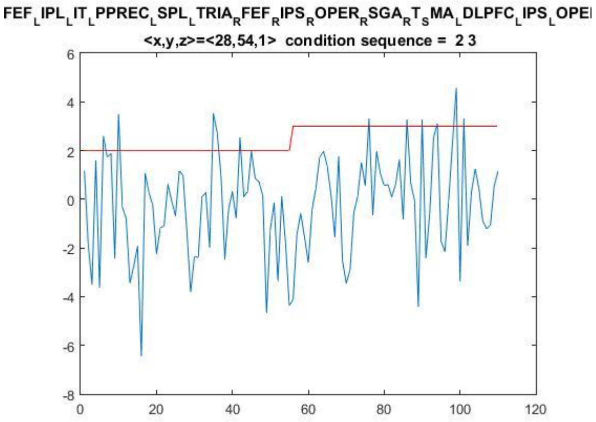
1. Gaussian Naive Bayes [Conditional Independence]
2. Bayesian Networks [Conditional Dependence]
    a. Tree Augmented Naive Bayes [TAN]
    b. k-dependence TAN.

Based on different similarity measures between features

3. Linear SVM (LibLinear) [High Dim - Linear Kernel SVM]
4. k-NN [Distance based similarity measure]

Why GNB?

- Classify- assuming each feature contributes independently of each other. [Conditional Independence assumption.]
- Continuous Data.
- Simple Assumption: Features follow a Gaussian distribution.



TAN and KD-BN

- Relaxed conditional independence.
- Relation between features are characterised by conditional mutual info gain [Correlation coefficient]
- 3 phase:
  - Pre-Processing [Estimate relation between features]
  - Structural Learning [Produce dependency structure]
  - Inference [Predict based on dependency structure]

Structure Learning, Parameter Estimation

Information Gain -> $I_P(A_i;A_j|C) = -\frac{1}{2}\sum_{c}^{C} P(c)\log\left(1-\rho_c^2(A_i,A_j)\right)$

Correlation Coefficient $\rho_c(A_i, A_j) = \frac{\sigma_{ij|c}}{\sqrt{\sigma_{i|c}^2 \sigma_{j|c}^2}}$

Likelihood of tree|data $LL(B_T \mid D) = \sum_{i=1}^{N}\sum_{j=1}^{n} \log\left[P_D\left(A_j^i \mid \Pi_i\right)\right]$

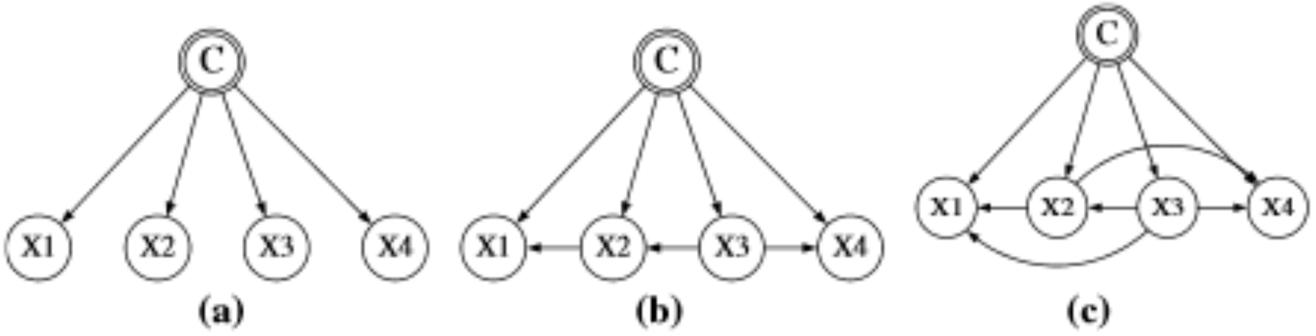
Inference $P(c \mid a) \propto p(c,a) = P(c)\, p(a \mid c) = P(c)\prod_{i=1}^{n} p(a_i \mid \Pi_i)$

Recomputed Mean $m_{i|c} = \mu_{i|c} + \sum^{ni} \beta_{ij|c}\left(a_j - \mu_{j|c}\right)$

Recomputed Variance $v_{i|c} = \frac{\left|\Sigma_{ai,\Pi i|C}\right|}{\left|\Sigma_{\Pi i|C}\right|}$

Weighted difference (Beta) $\beta_{ji|c} = \frac{\sigma_{ij|c}}{\sigma_{j|c}^2}$

BN Structures

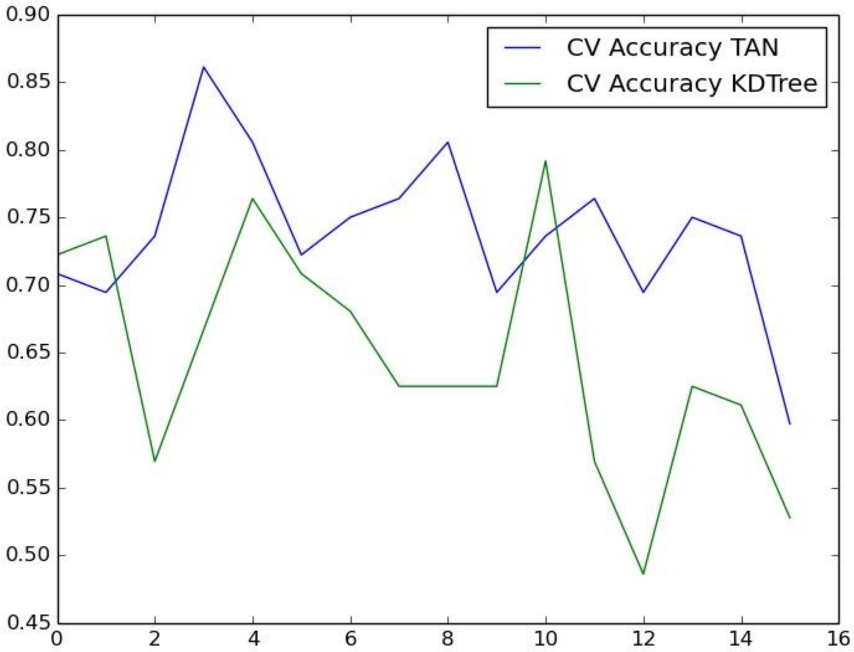


Naive Bayes

Tree Augmented Naive Bayes

k-Dependency Bayesian Network

TAN Cross Validation Results



Feature Selection: Active 100

TAN Average Accuracy: 0.738
k-DB Average Accuracy: 0.645

TAN

As expected, it performs better than SVM and GNB.

Test accuracies:

	Selected ROIs	Active 100	Active 200	Active 300	Active 400	Active 500
TAN	0.539	0.757	0.804	0.820	0.882	0.859
kDB	0.578	0.632	0.787	0.855	0.863	0.852
GNB	0.57	0.70	0.79	0.82	0.79	0.82

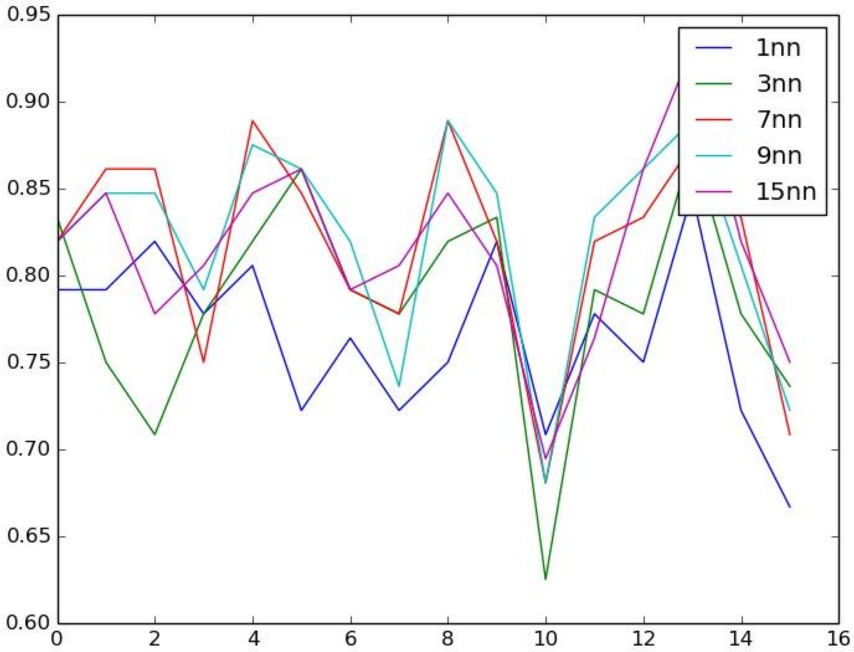
Why kNN?

- Captures Similarity in terms of distance measure.
- Highly varied performance.

  Test Accuracy:

  - Based on ROIs: 0.5
  - Based on Active Voxels: 0.75 - 0.9
- Performance significantly improves after reducing dimensionality.
  - k-nn is susceptible to irrelevant features. [Noise]

k-NN Cross Validation Results



Feature Selection: Active 100

Cross Validation Average Accuracy Score for:

1nn: 0.7482
3nn: 0.7803
7nn: 0.8029
9nn: 0.8098
15nn: 0.8211

k-NN Test Results

	Selected ROIs	Active 100	Active 200	Active 300	Active 400	Active 500
1NN	0.534	0.764	0.969	0.964	0.967	0.968
3NN	0.547	0.784	0.955	0.960	0.963	0.961
7NN		0.815	0.937	0.934	0.934	0.947
9NN		0.82	0.930	0.925	0.928	0.936
15NN		0.814	0.912	0.917	0.905	0.922

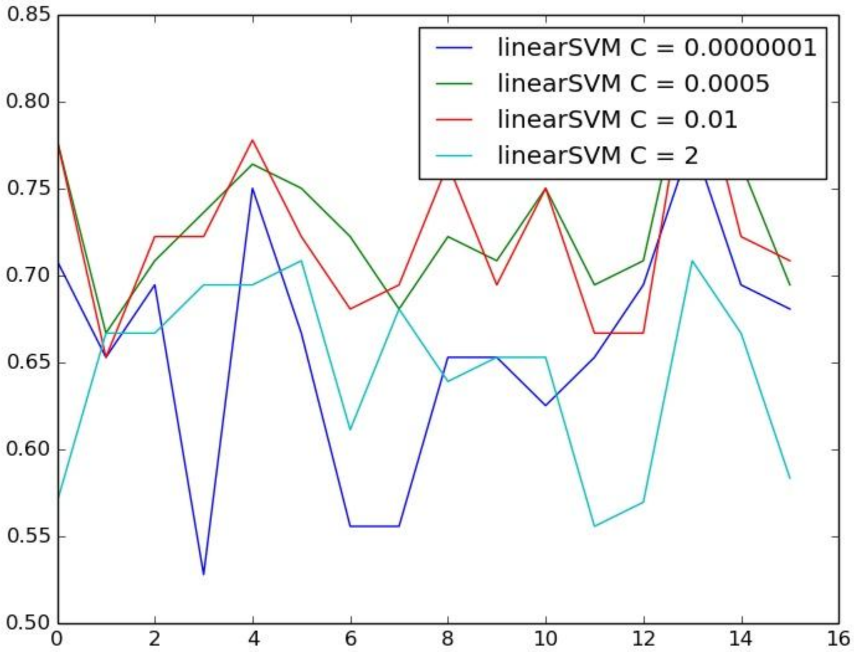
Why SVM?

1. High-dimensional data
2. Captures similarity between voxels (Linear Kernel)
3. Similarity only near the support vectors.

Test accuracies:

	Selected ROIs	Active 100	Active 200	Active 300	Active 400	Active 500
SVM	0.601	0.679	0.757	0.83	0.828	0.812

Linear SVM Cross Validation Results



Feature Selection: Active 100

Cross Validation Average Accuracy

C = 0.0000001	0.658
C = 0.0005	0.730
C = 0.01	0.722
C = 2	0.644

Conclusions/Learnings/Next Steps

1. Used different learning algorithms based on the nature of the data.
2. Capturing dependency between features contribute to increase performance in classification.
3. Study - how to reduce the effect of noise in the dependency structure.
4. Decrease the computational complexity of the algorithm [maximum spanning tree and DAG formation].