Classification of Brain States from fMRI Data Using Machine Learning Techniques

Ashish Farmer, Yash Shah, Haixuan Sun

1. The Problem

Inferring about the extent of craving in nicotine dependent subjects in response to certain cues is an important aspect to learn about the mind of the individual. This information can be used for rehabilitation purposes. In this project, we propose to use machine learning techniques to infer about the feeling of craving in an individual using the fMRI data collected when the individual was shown some visual cues (ex. Fig.1). We aim to cast our problem into the framework of "Multi-task" learning problems for classification, and come up with a generalized classifier that can accurately classify across different subjects and also classify data from a new subject not used to train the classifier. We also explore the use of Hidden Markov Models as classifiers for our application.

Figure 1: Examples of images presented to the subject to influence their brain activity while scanning.

2. Motivation and Introduction

Smoking addiction or nicotine dependence is a major health concern and nicotine craving can be a persistent and disturbing feature of such addiction. Studies have reported that nicotine dependence level of subjects is associated with greater BOLD fMRI activation [1, 2] and craving for cigarettes in response to smoking cues. The use of machine learning techniques has been increasing in fMRI data analysis[3, 4, 5], mostly because of real-time capabilities of such multivariate pattern analysis techniques[6]. This helps to understand the neural mechanisms involved and throws light on the possible application of real-time fMRI as a neurofeedback tool to enable self-regulation of their feelings (craving) similar to that shown in[7] where individuals are shown to be capable of learning to directly control activation of localized brain regions that are associated with pain perception and regulation.

In this project, we propose to use support vector machines to examine brain fMRI data of nicotine dependent subjects and analyze difference in activation when presented with smoking-related visual cues, previously used in [8]. Once a model is trained, it is possible

to exploit the fast nature of support vector testing and predict the subjects brain state with every acquired image. Furthermore, we wish to generalize our classifier to work on data from a subject not used in training. We investigate the problem by casting it into the framework of Multi task learning [9, 10, 11]. We also look into using Hidden Markov Models(HMM) for classification by drawing an analogy between fMRI classification and speech processing[12] and face recognition[13] where HMMs have been widely used.

fMRI datasets are inherently high dimensional because of the very large number ($\sim 10^6$) of features (pixel intensities). This implies that the number of classifier parameters increase, resulting into computationally expensive estimation procedures and reduced accuracy of the estimated parameters. Thus it is vital to perform some kind of dimensionality reduction before classification, so that we remove some features irrelevant to classification, and hence reduce noise. For dimensionality reduction, we explore feature selection techniques which are more intuitive and it is easy to map back into the original feature space for interpretation.

3. Notation and Formal Setup

3.1 Multi Task Learning

Consider a setting where we observe data from T "related" tasks, and for simplicity we assume same number of samples(m) for each task. Thus, we have T input spaces $\mathcal{X}_l, 1 \leq l \leq T$ and T output spaces $\mathcal{Y}_l, 1 \leq l \leq T$. We assume $\mathcal{X}_l = \mathcal{X}(= \mathbb{R}^d)$ for all l and $\mathcal{Y}_l = \mathcal{Y} = \{-1, 1\}$, a more generalized treatement of the problem can be found in[11]. Thus, we have our training data $\{(x_{il}, y_{il}) : 1 \leq i \leq m, 1 \leq l \leq T\} \subset \mathcal{X}_l \times \mathcal{Y}_l$, where $(x_{il}, y_{il})_{i=1}^m$ for a given l is sampled from an unknown distribution P_{XY}^l. Let $\mathcal{P}_{\mathcal{X}\times\mathcal{Y}}$ denote the set of probability distributions on $\mathcal{X} \times \mathcal{Y}$, and our goal is to find a family of T functions $f_l : \mathcal{X} \to \mathcal{Y}, 1 \leq l \leq T$ such that function f_l minimizes $\mathbb{E}_{P_{XY}^l}[L(y_l, x_l)]$ for some loss function $L(\cdot, \cdot)$. One approach to solve this problem is to solve an SVM for each l independently and obtain all the classifier functions, however in doing so we assume that the tasks are independent, and we do not take advantage of relatedness of tasks. Multi task learning [9] offers a framework to encode the relationships between different tasks while simultaneously solving for the classifiers. It has been shown that this leads to an improvement in performance, as we essentially increase the number of training data points because now we consider training point from other classes too. Evegniou *et.al.*[11] extended the concept of Multi task learning to SVMs.

Let the family of SVM classifier functions be written as: $f_l(x) = u_l^T x, u_l \in \mathbb{R}^d$, thus the multi task SVM problem is to estimate the parameters $u = (u_l, 1 \leq l \leq T) \in \mathbb{R}^{Td}$ as minimizer of a regularization function

$$R(u) = \frac{1}{Tm} \sum_{l=1}^{T} \sum_{i=1}^{m} L(y_{il}, u_l^T x_{il}) + \gamma J(u) \tag{1}$$

where γ is a positive regularization parameter and $J(u)$ is a homogenous quadratic function of u, that is $J(u) = u^T E u$ for a $Td \times Td$ matrix E that captures relationship between different tasks.

Solution of Eq.(1) involves estimating Td parameters, which can be very large if T or d is very large. The following analysis as in [11] shows that the solution of Eq.(1) is equivalent

to a single task learning method for an appropriate choice of kernel which would depend on the data points as well as an additional dimension which encodes the relationship between different tasks. Thus, we transform the data into a higher dimensional space with dimension $p \geq Td$ using a linear feature map Φ such that we have the classifier

$$f_l(x) = w^T \Phi(x) + b = w^T B_l x + b, w \in \mathbb{R}^p \tag{2}$$

for some $p \times d$ matrix B_l. Eq.(2) is equivalent to learning the coefficients $u_l = B_l^T w$ and for existance of unique solution, we must have B_l to be a full rank matrix for all l. If we select $B_l = B_0, 1 \leq l \leq T$, then it is easy to see by plugging into Eq.(2) that we will have $f_1 = f_2 = ... = f_T$, that is all the tasks are the same task. Thus, B_l indeed encodes the coupling information between the tasks.

Writing a data point x from task t as an ordered pair (x, t), we have the kernel

$$K((x,t),(s,q)) = \Phi((x,t))\Phi((s,q)) = x^T B_t^T B_q s, \qquad x, s \in \mathbb{R}^d, 1 \leq t, q \leq T \tag{3}$$

which is called *linear multi task kernel*, as it is bilinear for x and s for fixed t and q. With these transformatons, the optimization problem in Eq.(1) can be written as:

$$S(w) = \frac{1}{Td} \sum_{l=1}^{T} \sum_{i=1}^{m} L(y_{il}, w^T B_l x_{il}) + \gamma w^T w, \qquad w \in \mathbb{R}^p \tag{4}$$

Taking $L(\cdot, \cdot)$ to be the hinge loss, we can write Eq.(4) as the usual SVM problem:

$$\begin{aligned} \min_{w,b} & \sum_{l=1}^{T} \sum_{i=1}^{m} \xi_{il} + \gamma ||w||^2 \\ s.t. \quad & y_{il}(w^T B_l x_{il} + b) \geq 1 - \xi_{il} \\ & \xi_{il} \geq 0 \end{aligned} \tag{5}$$

which is analogous to solving a single task SVM[14] with Tm training samples. Therefore, we obtain the optimal solution of Eq.(5) of the form

$$f_q^* = \sum_{i=1}^{m} \sum_{l=1}^{T} \alpha_{il} K((x_{il}, l), (x, q)), \qquad x \in \mathbb{R}^d, 1 \leq q \leq T \tag{6}$$

by solving the dual quadratic program:

$$\begin{aligned} & \max_{\alpha_{il}} \sum_{l=1}^{T} \sum_{i=1}^{m} \alpha_{il} - \frac{1}{2} \sum_{l=1}^{T} \sum_{q=1}^{T} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_{il} y_{il} \alpha_{jq} y_{jq} K((x_{il}, l), (x_{jq}, q)) \\ s.t. \qquad & 0 \leq c_{il} \leq \frac{1}{2\gamma} \qquad \text{and} \qquad \sum_{i=1}^{m} \sum_{l=1}^{T} \alpha_{il} y_{il} = 0 \end{aligned} \tag{7}$$

The dual program in Eq.(7) is same as solving the dual problem for single task SVM with

Tm training samples with kernel K. An example of linear multi task kernel is:

$$K((x,l),(s,q)) = (1-\lambda+\lambda T\delta_{lq})x^T s, \qquad x,s \in \mathbb{R}^d, 1 \leq l,q \leq T \tag{8}$$

Here λ is the coupling parameter between 0 and 1. It is easy to see that if λ is small, every task influences the other tasks heavily, that is there is a strong coupling between the tasks, whereas $\lambda = 1$ means that the tasks are learnt independently. The coupling parameter λ and the regularization parameter γ can be selected using cross validation.

3.2 Hidden Markov Model

HMM consists of two interrelated processes: an underlying unobservable Markov chain with a fixed number of hidden states(N), an observable random process related to the hidden process, a state transition probability matrix(P) and the initial state probability distribution(Π) and a set of probability density function associated with each of the states. Let $S = \{s_1, s_2, ..., s_N\}$ denote the set of hidden states and $V = \{v_1, v_2, ..., v_M\}$ be the set of possible observation symbols (also called the codebook). Let B be the observation probability, *i.e.* $B = b_j(k)$, where $b_j(k) = P\{O_t = v_k | q_t = s_j\}, 1 \leq k \leq M, 1 \leq j \leq N$, where O_t is the observed symbol at time t. Let $\Pi = \pi(i) = P\{q_1 = s_i\}, 1 \leq i \leq N$ be the initial distribution on the hidden states, then a HMM is defined by the triplet $\lambda = (A, B, \Pi)$. Typically B is assumed to be a mixture of the form $b_i(O) = \sum_{k=1}^{m} c_{ik} N(O, \mu_{ik}, U_{ik})$, where c_{ik} is the mixture coefficient of k^{th} mixture in state i, and $N(O, \mu_{ik}, U_{ik})$ is a multivariate Gaussian density with mean μ_{ik} and covariance matrix U_{ik}. And the parameters of HMM can be computed using Viterbi, and Baum-Welch algorithms[12].

In the first proposed method, we assume one HMM generating craving data for all individuals and one HMM generating non craving data for all individuals. For the second method, we assume a different pair of HMMs for each of the subjects. During training phase, we build a HMM λ^v by optimizing the model parameters (A, B, Π) for each family of training sequences, that optimize the likelihood of the training set observations. We classify each test image(O) by selecting the family of HMM that maximizes the posterior probability. That is, the decision rule is:

$$\hat{y} = \arg\max_{1 \leq v \leq V} \; P(O|\lambda^v) \tag{9}$$

3.3 Generalizing the classifier

A simple generalization of the classifier would be to compute $\hat{y}_l = f_l^*(x)$ for $1 \leq l \leq T$ for a test point $x \in \mathbb{R}^d$ and take a majority vote. We propose a method that combines the idea of Nearest Neighbor classifier and the SVM. Let μ_l^+ and μ_l^- denote the centers of training data from task l corresponding to the label $+1$ and -1 respectively. That is:

$$\mu_l^+ = \frac{1}{|\{y_{il} = +1 : i \in \{1, 2, ..., m\}\}|} \sum_{x \in \{x_{il} | y_{il} = +1, 1 \leq i \leq m\}} x \tag{10}$$

Similarly we define μ_l^-. For a given unlabeled test point x, we compute the SVM decision $\hat{y}_l = f_l^*(x)$ for $1 \leq l \leq T$, and then decide the label $\hat{y}_k$ for x where k is given as:

$$k = \arg\max_r \ ||x - \mu_r^{\hat{y}_r}|| \tag{11}$$

where $\mu_r^{\hat{y}_r} = \mu_r^+$ if $\hat{y}_r = +1$ and vice versa.

3.4 Dimensionality Reduction

In this project, we explored feature subset selection techniques of dimensionality reduction which makes it easier to map data back into original space for interpretation. Mainly two types of feature extraction methods are employed called filter methods and wrapper methods. Filter methods involve ranking the features by certain criteria and then selecting and retaining the top-ranked features, whereas, wrapper methods usually involve retaining those features that improve the performance of the classifier and excluding the ones that do not have much impact.

3.4.1 Two Sample T-test Statistic Score (T-score)

A t-test is a statistical hypothesis test in which the test statistic follows a Student-t distribution. Here, we use the t test as a univariate feature selection algorithm where we rank features and select only a subset of features that is most discriminating between the class labels. First we partition the data into two sets A and B. Let μ_A, σ_A denote the average and variance of the data points of set A and μ_B and σ_B denote the average and variance of the data points in set B. Then we define the T-score for a feature i in the data as:

$$t^{(i)} = \frac{\mu_A^{(i)} - \mu_B^{(i)}}{\sqrt{\frac{\sigma_A^{(i)}}{|A|} + \frac{\sigma_B^{(i)}}{|B|}}} \tag{12}$$

We can select first $k < d$ features with the highest T-scores (most discriminating features) as our data. Optimal value of k can be determined using cross validation.

3.4.2 Weight Based Scores

This is an iterative multivariate feature selection algorithm where we start with all the features totrain a classifier and then discard the features with least weights (say 20% features with lowest weights) after each iteration. The assumption in this algorithm is that the weight assigned to each feature determines how significant the feature is for classification.

4. Experiments and Discussions

4.1 Dimensionality Reduction

The data collected from different nicotine dependent subjects is their brain activation maps when subjects are presented with a sequence of images with craving descriptors (Fig.1) for

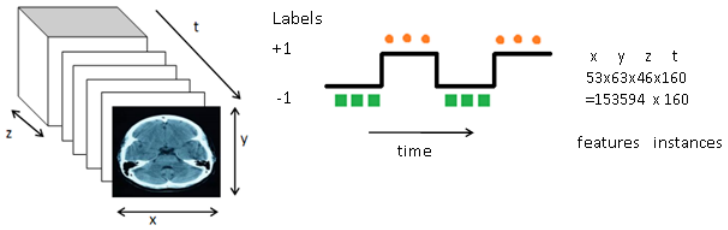
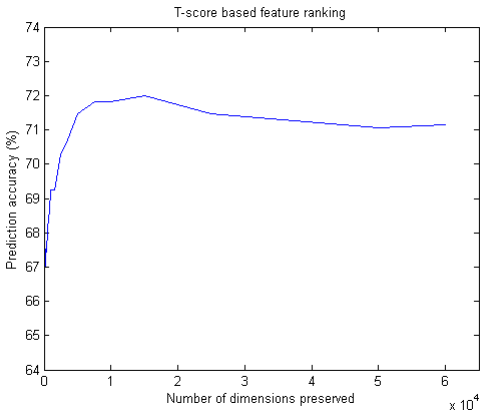


Figure 2: Alternating blocks of different conditions are presented to the subject and 3-dimensional volumes are captured across time. Each of these volumes is labeled as belonging to either of two classes (+1 or -1).

a fixed length of time followed by a sequence of images that has no relation to nicotine or smoking. The data is three dimensional because it has brain images for slices at regular spatial intervals as well as temporal intervals (See Fig.2). Each brain map at every point of time has roughly 150000 features, and we have such data points for 160 points in time. Two runs of data were acquired for each of the 16 subjects using both the paradigms. All the data was normalized to an MNI atlas space using SPM. It was then smoothed using a gaussian kernel with FWHM 8mm to denoise the data. The data was then standardized to have zero mean and unit variance. To solve the SVM problem, we used the SMO code given for one of the homeworks.



(a) Plot of performance against data dimensionality for dimensionality reduction using T-scores

(b) Plot of performance against data dimensionality for dimensionality reduction using recursive feature elimination

Figure 3

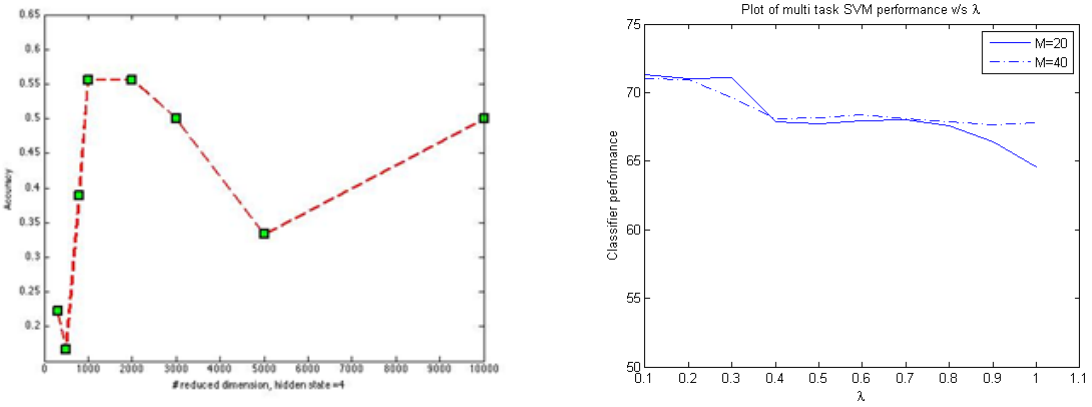
The data was preprocessed for dimensionality reduction, and the reduced dimension data is fed into SVM for classification. The goal of dimensionality reduction is to reduce the dimension without significant loss of information. To test for loss of information, we we trained an SVM for 16 subjects using run1 of their data and tested on run2, and plotted the average classification performance [1] against dimensionality of the data (See Fig.3). We see that the average prediction accuracy obtained by using $\sim$ 60000 features is 71%, whereas,

[1] We define Performance = 1 - Test error = $\frac{1}{n}\sum_{i=1}^{n} 1_{y_i=f(x_i)}$, n is number of test points

if only 15000 features are used for SVM training and testing, then the prediction accuracy is found to be 72%. This confirms that most discriminatory information is still preserved when the dimensionality is reduced to 1/4th since the classification performance is slightly better. Similarly, from Fig.(3b), we can see that the average prediction accuracy increases from ∼71% to about 72.7% when only the most significant 80% of the voxels are preserved and rest are ignored recursively until only 5000 voxels are left out. We see a little degradation in performance as dimension of the data increases because of noise due to more number of dimensions.

4.2 Classification

For classification we reduce the dimension of the data to 1000 features. We compute the HMM model parameters using first 10 craving images for eight subjects. For the initial states, we make guess on Gaussian random mixture probability density function and mixture matrix with respect to the training data into the observation distribution function B, and we assume random state transition probability distribution and initial states. The initial guess affects the accuracy of the HMM training, as the Baum-Welch algorithm only converges to the local maximal, thus we apply Kevin Murphys Bayes Net Toobox[2] that makes good guesses to initialize our HMM well. For the recognition, we would take the 10 images of the same states as a test family of sequence, apply the dimension reduction same as above, and take it into the evaluation of each HMM model likelihood. The performance of this classifier was just a little above 50% chance (See Fig.4(a)), however we see a better performance for method.2.



(a) Plot of performance of HMM classifier(Method.1) aginst the dimension of input data

(b) Plot of multi task classifier performance against λ for 20 and 40 training points per task(M).

Figure 4

In the second proposed method using HMMs, we scan each image and descompose it into different overlapping windows. We divide our 3D fMRI image into 30 equal size volumes that are all parallel to the ground surface. Each window consists of two neighboring volumes.

[2] `http://www.cs.ubc.ca/~murphyk/Software/HMM/hmm.html`

The window is the two topmost volumes, and then for each time, the window contains one below neighbor volume and we kick out the first volume, and so on. For each image of same experiment, more windows are added. In this way, a sequence of 10 vectors with each size 29 is generated. In our method, we apply the dimensionality reduction to the original data, and track the space position of each left point. Thus the dimension is reduced to 1/15 of the reduced dimension. In our assumptions, we assume the hidden states correspond to the areas of brain. In the initialization, we construct the initial state as the first hidden state, and the transform state matrix with zeros from j to i where $j > i$ and uniform for other values. Same as above method, we make random Gaussian pdf w.r.t the data and mixture matrix initialization. For recognition, the test images are applied to the above feature extraction that decomposed into different overlapping windows. For this method, the classifier performance was observed to be 68.33%.

Using 1000 features (dimensions), we trained an individual SVM (Eq.(2)) on the run 1 of the data for 16 subjects, and tested on run 2 for the same subject and observed 74.1% classification accuracy with a linear kernel. For multi-task SVM, we used cross validation to get the value of coupling parameter λ. First we split the training data into two sets and train on one of the sets and test on the other. These experiments were repeated for values of λ between 0 and 1 at intervals of 0.1 (See Fig.(4b)). The optimal value of the coupling parameter found using this method was $\lambda = 0.2$, which indicates that there is strong coupling between the tasks (subjects). Using $\lambda = 0.2$ and data from 16 subjects with 20 training data points for each subject, and testing on run2 for the same subject we obtained the average accuracy of the Multi task SVM (MT-SVM) to be 71.2%. There is a slight drop in the performance of the MT-SVM compared to single task SVM, because there are some subjects whose data (response) was observed to be very different than others. This is the case where the assumption of close coupling among all tasks fails, and performance drops. A way to handle such cases is to use different coupling parameters for different taks. Such an extension of the MT-SVM makes the problem non tractable as now we are left to select $\binom{T}{2}$ parameters. We plan to take up this treatement for generalizing the coupling parameter as a part of future work.

Table.1 shows a comparison between individual SVM and MT-SVM with the number of tasks and training data points per class. We see a clear advantage of using multi task learning methods for the scenerio where we have a lot of related tasks and very few training samples per task, which is very relevant to the fMRI classification setting. This validates the use and further exploration into multi task learning techniques for classification problems using fMRI data.

Another aspect investigated in this project was to generalize the classifiers for a subject not used to train the classifier. Table.2 shows the results of various generalization techniques we implemented, simplest one being to train a single classifier using data from all the subjects. This method is not very accurate because if one of the tasks is particularly very noisy, then it will greatly affect the classifier. Another extension is to use a majority vote over individual classifiers obtained using single task as well as multi task SVMs. As expected the generalization of MT-SVM classifiers using majority vote turned out to be more accurate as the individual classifiers are more accurate than single task. The proposed generalization method (sec.3.3) based on weighing the SVM votes based on Nearest Neighbor classifier out-

Tasks	Training points per task	Individual SVM	MT-SVM
5	20	56.50	62.50
5	40	70.63	63.25
5	60	72.25	70.38
10	20	57.50	70.56
10	40	66.81	70.38
10	60	69.81	70.81
15	20	58.38	71.08
15	40	65.50	71.00
15	60	70.29	71.17

Table 1: Comparison between individual SVM and MT-SVM as the number of tasks and training data per task changes.

performs the other discussed methods by at least 1% or more for most cases. For comparison, we also show performance of Nearest Neighbor classifier and k-Nearest Neighbor classifier with $k = 37$ for classifying the fMRI data.

Tasks	Data	One SVM	Indiv. SVM	MT-SVM	Proposed	NN	kNN
5	20	70.28	67.94	69.91	71.03	65.22	71.16
5	40	71.25	67.94	70.03	70.81	69.00	70.34
5	60	71.5	67.94	70.53	70.72	66.44	70.59
10	20	66.97	63.38	70.69	70.63	70.91	70.16
10	40	69.63	63.34	70.53	71.06	71.22	68.66
10	60	70.25	63.38	70.59	71.03	70.06	68.58

Table 2: Comparison between different classifier generalization methods.

5. Conclusions and Future Work

The investigation into dimensionality reduction confirms the idea of using a lower dimensional data for this application. We also explored the Multi Task SVM technique for classification, and showed that fMRI classification fits well into the framework of multi task learning. Hence we could improve the classification performance using MT-SVM. Finally we explored the techniques for generalizing the classifier, and proposed a method for selecting a classifier from the set of classifiers given by MT-SVM using principles of NN classification.

Further improvements are possible in the techniques explored in the project. For example, generalizing the multi task kernel to a case where we have different couplings between different tasks. Also, we only investigated the linear kernel for SVM and MT-SVM in this project. We would like to use higher order kernels like Gaussian or polynomial kernel and investigate into the problem.

In the brain state recognition from fMRI, HMM does not perfectly predict the pattern. This is because the difference between craving and non craving fMRI images is not great

enough. However, in the future, if we wish to study HMM more carefully and select model parameters that might give valuable information about the brain activity in response to craving.

Individual Effort

Ashish: Ashish worked on classification problem and investigated into MT-SVM theory and implemented the MT-SVM classifier. He worked on tuning the classifier parameters and training and testing on the data. He worked on generalizing the classifiers and implementing different techniques for the same.
Yash: Yash prepared all the fMRI data for classification. He worked on pre processing of data, normalizing the data to MNI atlas space and smoothing. He investigated into the dimensionality reduction techniques and implemented them. He analyzed the effects of using lower dimensional data on classification performance, and calculated optimal number of parameters required for good classification accuracy.
Ashish and Yash both contributed equally to doing the literature survey for the project and worked on the Project Description and Proposal documents.
Haixuan: Haixuan investigated the use of HMM as a classification technique. He investigated and implemented the two methods proposed for classification and evaluated their performance.
All three group members worked on their part of report, and have equal contribution in aggregating the content of the report as a whole.

References

[1] M. Smolka, M. Bhler, S. Klein, U. Zimmermann, K. Mann, A. Heinz, and D. Braus, "Severity of nicotine dependence modulates cue-induced brain activity in regions involved in motor preparation and imagery," *Psychopharmacology*, vol. 184, pp. 577–588, 2006. 10.1007/s00213-005-0080-x.

[2] F. J. McClernon, F. B. Hiott, J. Liu, A. N. Salley, F. M. Behm, and J. E. Rose, "Selectively reduced responses to smoking cues in amygdala following extinction-based smoking cessation: results of a preliminary functional magnetic resonance imaging study.," *Addiction Biology*, vol. 12, no. 3-4, pp. 503–512, 2007.

[3] F. Pereira, T. Mitchell, and M. Botvinick, "Machine learning classifiers and fmri: a tutorial overview.," *NeuroImage*, vol. 45, no. 1 Suppl, pp. S199–S209, 2009.

[4] S. Lemm, B. Blankertz, T. Dickhaus, and K.-R. Mller, "Introduction to machine learning for brain imaging," *NeuroImage*, vol. 56, no. 2, pp. 387 – 399, 2011.

[5] S. LaConte, S. Strother, V. Cherkassky, J. Anderson, and X. Hu, "Support vector machines for temporal classification of block design fmri data," *NeuroImage*, vol. 26, no. 2, pp. 317 – 329, 2005.

[6] S. M. LaConte, S. J. Peltier, and X. P. Hu, "Real-time fmri using brain-state classification.," *Human Brain Mapping*, vol. 28, no. 10, pp. 1033–1044, 2007.

[7] R. C. deCharms, F. Maeda, G. H. Glover, D. Ludlow, J. M. Pauly, D. Soneji, J. D. E. Gabrieli, and S. C. Mackey, "Control over brain activation and pain learned by using real-time functional mri," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 51, pp. 18626–18631, 2005.

[8] A. King, P. McNamara, M. Angstadt, and K. L. Phan, "Neural substrates of alcohol-induced smoking urge in heavy drinking nondaily smokers.," *Neuropsychopharmacology*, vol. 35, no. 3, pp. 692–701, 2010.

[9] R. Caruana, "Multitask learning," in *Machine Learning*, pp. 41–75, 1997.

[10] T. Evgeniou, "Regularized multi-task learning," pp. 109–117, 2004.

[11] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *Journal of Machine Learning Research*, vol. 6, pp. 615–637, 2005.

[12] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," in *Proceedings of IEEE*, vol. 77, pp. 257–286, IEEE, 1989.

[13] A. Nefian and I. Hayes, M.H., "Hidden markov models for face recognition," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 5, pp. 2721 –2724 vol.5, may 1998.

[14] V. Vapnik, *Statistical learning theory*. Wiley, 1998.