

# Classification of fMRI Data Using Density Based Support Vector Machines

Zahra Nazari<sup>1</sup>, Dongshik Kang<sup>1</sup>, M. Reza Asharif<sup>1</sup>, Yul-Wan Sung<sup>2</sup> and Seiji Ogawa<sup>2</sup>

<sup>1</sup>Department of Information Engineering, University of the Ryukyus, Okinawa, Japan

<sup>2</sup>Kansei Fukushi Research Institute, Tohoku Fukushi University, Sendai, Japan

**Abstract:** *Interpreting brain images requires analysis of complex and multivariate data. Machine learning algorithms are the most popular and widely used analysis approaches to train classifiers to decode stimuli, behaviours, mental states, and other variables of interest from functional Magnetic Resonance Imaging (fMRI) data and thereby show the data contain enough information about them. In the present study, multivariate statistical pattern recognition methods including Support Vector Machines (SVM) and Density Based Support Vector Machines (DBSVM) were used to classify fMRI volumes, and the results of SVM and DBSVM classifiers were comparatively analysed..*

**Keywords:** *Support Vector Machines, fMRI data classification, Density Based Support Vector Machines*

## 1. Introduction

The advent of fMRI in the early 1990s, especially BOLD fMRI, by Ogawa and colleagues has provided a revolutionary means for non-invasively probing spatiotemporal variations of brain function. While conventional MRI results in snapshots of what's inside the body, fMRI produces movies starring the brain. Many researchers from all over the world are using fMRI in different areas such as improving lives, treating disorders, addressing social problems, exploring the mind and etc. Over the past years a variety of different fMRI experiments have been done and revealed functional characteristics of the human brain depending on certain tasks. By recording the activity pattern of human brain as images of 3D voxel (volume element), we are able to visualize the picture of the pattern in relation to a specific task and find statistical differences in brain activity patterns by comparing with different tasks. A challenging problem is to predict, through brain activity patterns with a classifier, high cognitive functions such as whether the human subject is reading a sentence or looking at an object, or whether the subject is reading an ambiguous or non-ambiguous sentence, etc. In the last few years there has been growing interest in the use of machine learning classifiers for analysing fMRI data. A growing number of studies have shown that machine learning is a useful tool to estimate exciting new information from neuro-imaging data [1-3].

Machine learning is a most powerful approach to train the classifier and then use it to discriminate between different cognitive states. Machine learning techniques take into account the full spatial pattern of brain activity, measured simultaneously at many locations, and exploit the inherent multivariate nature of fMRI data. The application of machine learning techniques to fMRI data has been referred to as multi-voxel pattern analysis (MVPA) and it generally entails four steps. First step is to select the set of voxels that will enter the multivariate analysis. With respect to the selection of voxels, the analysis may be massively multivariate and consider all brain voxels simultaneously or may be limited to a subset of voxels from one region-of-interest (ROI) in which case the dimensionality of the multivariate space is greatly reduced. Second is to represent stimulus-evoked

brain activity as a point in a multidimensional space, i.e. as the pattern of intensity values at selected voxels (multi-voxel patterns analysis, MVPA) [4]. Third step is to use a subset of trials with which a classifier is trained and the optimal separating boundary (hyper-surface) between different conditions in this multidimensional space is defined. Several methods including SVMs, linear discriminant analysis (LDA), Gaussian Naïve Bayes (GNB) and Neural classifiers have been used for this purpose. During training, a map coding for the relative contribution of each voxel to the discrimination of conditions (discriminative map) can be directly obtained for all linear classifiers. Fourth and the last step is to test the capability of the trained classifier to accurately discriminate the experimental conditions when new data are presented (generalization) [1, 5-8].

The purpose of the present study is to apply standard SVM and Density Based SVM to the problem of fMRI data classification and compare their results. Statistical pattern recognition algorithms are designed to learn and later classify multivariate data points based on statistical regularities in the data set. Among all of these algorithms, SVM is one of the best and mostly used algorithms in different practical problems. In this study, we attempted to classify the two categories of subjects who were with closed and open eyes in the same dark room.

## 2. Support Vector Machines

Data classification process using SVM includes two stages: learning is the first stage, the aim of which is to analyze labeled data and learn a mapping from  $x$  to  $y$  where  $y = \{1, \dots, C\}$  (with  $C$  being the number of classes) and to build a classifier. The second stage is predicting which is using the established model for predicting on novel inputs. SVM is one of the most successful classification algorithms and its important property is that the determination of the model parameters corresponds to a convex optimization problem, and so any local solution is also a global optimum. The basis of the theory of SVM for classification problems will be reviewed in the following [9, 10].

### 2.1. Hard Margin SVM

The linearly separable case is the easiest classification problem which is rare in practice. In this case data pairs can be classified perfectly and the empirical risk can be set to zero. In linearly separable cases, among all the separating hyperplanes which minimize the empirical risk, the one with the largest margin is required. This can be expressed as the idea that a classifier with a smaller margin will have a higher expected risk. Suppose that a set of 2-dimensional labeled training points  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  is given and each of them has a class label  $y \in \{-1, 1\}$  which denotes the two classes separately. During the learning stage the machine finds parameters  $w$  and  $b$  of the decision function  $f(x)$  given as:

$$f(x) = \text{sgn}(x \cdot w^T + b) \quad (1)$$

where  $w$  is the weight vector and  $b$  is the bias. SVM, after learning by training points can produce an output for unknown data point according to above decision function (1). The linearly separable data points can be classified by solving the following quadratic program:

$$\begin{cases} \min \frac{1}{2} \|w\|^2 \\ y_i(x_i \cdot w^T + b) \geq 1 \quad i = 1, \dots, N \end{cases} \quad (2)$$

## 2.2. Soft Margin SVM

In previous section the training points were assumed that are linearly separable and the resulting support vector machine will give exact separation of the training points which is not very realistic. Sometimes in real-world problems the training points are overlapped (slightly nonlinear) and some samples cannot be classified correctly and the constraint in (2) will not be satisfied. Therefore classification violation must be allowed in the SVM. In practice the soft margin will be allowed. This approach allows some training points to be on the wrong side of the separating hyperplane, but with a penalty that increases with the distance from hyperplane. To do this, the nonnegative variable  $\xi \geq 0$  will be used to measure the amount of this violation and (2) will be modified to (3):

$$\left\{ \begin{array}{l} \min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \\ y_i(x_i \cdot w^T + b) \geq 1 - \xi_i \quad i = 1, \dots, N \\ \xi_i \geq 0 \quad i = 1, \dots, N \end{array} \right. \quad (3)$$

where  $C \cdot \sum \xi_i$  is the distance of error samples to their correct places. Parameter  $C > 0$  (the only free parameter in SVM) controls the trade-off between slack variable penalty and the margin [9, 10].

## 2.3. Non-linear SVM

In case of considerable class overlapping (seriously nonlinear) of the training points, soft margin SVM classifiers are unable to separate the samples into classes appropriately. Therefore SVM transforms samples  $x$  from original input space to a higher dimensional feature space by a non-linear vector mapping function  $\Phi(x) = R^n \rightarrow F$ . However the vector mapping function ( $\Phi$ ) leads to high computational expenses. Thus, this transformation can be performed by kernel function which allows more simplified representation of the data. Polynomial, Sigmoidal, and Gaussian (RBF) are some popular kernel functions for this kind of transformation [9-12].

The different distribution in the feature space enables the fitting of a linear hypersurface in order to separate all samples into the classes. Classification is easier in higher dimensions, but computation is costly. The resulting separating hypersurface in feature space will be optimal in the sense of being a maximal margin classifier with respect to training points. The vector  $\Phi(x_i)$  in the feature space corresponds to vector  $x_i$  in the original space. The solution in the SVM does not depend directly to input vectors, rather to dot product between input vectors, and so the dot product of  $\Phi(x_i) \cdot \Phi(x_j)$  is needed. It would be preferable to be able to define the dot product directly rather than defining the mapping  $\Phi$  explicitly. The kernel function computes the dot product of training points in feature space and there will be no need to define  $\Phi$  explicitly. By using Lagrange multiplier and kernel method, the QP for nonlinear cases is as below:

$$\left\{ \begin{array}{l} \max \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j K(x_i, x_j) \\ \sum_{i=1}^N y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C \quad i = 1, \dots, N \end{array} \right. \quad (4)$$

### 3. Density Based Support Vector Machines

The main goals of Density Based SVM which is introduced in reference (9), is reducing effects of outliers, maximizing margin, providing better generalization, and adjusting the decision boundary according to the density of data sets. Meanwhile Density Based SVM reduces the number of support vectors which decreases computational complexity. It is noteworthy that in Density Based SVM, input vectors are those which are in highest-confidence area of data set and they are more informative than other input vectors [9].

Density Based SVM can detect outliers or data points which are out of the densely populated area. To detect these outliers, first the densely populated area of a data set should be determined. The data points which are located in the densely populated area will be considered as important (meaningful) points and other as less important (meaningless) which can be misclassified or ignored. Although the concept of population density is used to develop Density Based SVM, formulas are different. In this method the distance (Euclidean & Mahalanobis) between data points of one data set plays the main role to determine the area with high population density [9].

#### 3.1. Density Based SVM with Euclidean Distance

Euclidean distance measures the distance between two points by formula (5) in Euclidean space. Suppose that a set of 2-dimensional data  $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$  is given. First the Euclidean distance between all data points of one class should be calculated. For example the Euclidean distance between point 1 and 2, 3, ..., n and the Euclidean distance between point 2 to 1, 3, ..., n and so on.

$$D(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2} \quad (5)$$

The next step is to sum up all distances for each point. For example the total distance for point 1 is  $d_1 = [D(1,2) + D(1,3) + \dots + D(1,n)]$  where n is the number of data points in one data set. The total distances for all data points of one data set is needed to calculate the average distance which will be used to determine data points which are inside or outside of densely populated area. The average distance can be calculated by formula (6).

$$Average_d = \frac{\sum_{j=1}^N \sum_{i=1}^N \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}}{n} \quad (6)$$

$if d_i > Average_d \rightarrow x_i = outlier$

After calculating the  $Average_d$  by (6), those data points with ( $d \leq Average_d$ ) should go to group 1 which is the new training set and others to group 2. The space which is occupied by group 1 is the area with high population density and data points inside this group will be considered as important data points. Those data points in group 2 will be considered as less important or outliers and they will not contribute in training phase [9].

---

#### Algorithm:

---

- 1- For each data point  $x_i$ :
    - Calculate the Euclidean distance between  $x_i$  and all other data points by (6)
    - Sum up all the distances calculated for one point as d
  - 2- Sum up all d values as  $total_d$ .
  - 3- Divide  $total_d$  by number of data points of one set as  $Average_d$  by (7)
  - 4- Set all data points with ( $d \leq Average_d$ ) in one group
  - 5- New group contains the most important data points and others will be considered as outliers.
-

## 4. Functional MRI Data

Functional MRI measurement of brain was performed by a 3 T MRI scanner (Verio, Siemens Co., Erlangen, Germany). Functional images were acquired from 26 subjects in resting-state in the dark room. The brain data of 13 subjects were acquired with the eyes open and the other 13 subjects with the eyes closed. The scan time was 5 minutes (300s) with repetition time of 2 seconds and 34 slice per an imaging volume. Image resolution was  $3.4 \times 3.4 \times 3.4 \text{ mm}^3$ . The brain data of 34 slices were separated into 116 ROIs (Region of Interest) by using AAL (Automated Anatomical Labeling) atlas as the mask and so a voxel has 150 samples on the temporal axis. For time series data of 150 sampling points of 116 ROIs, the correlation coefficient between each ROI was calculated, and finally a  $116 \times 116$  correlation coefficient matrix was obtained for each subject.

### 4.1. Experiment

The aim of our experiment is to classify the fMRI data which is acquired from the 26 subjects. Before using the classification algorithm we used Principle Component Analysis (PCA) algorithm to reduce the dimensionality of the data [13]. The procedure is explained in the following figure (Fig. 1). After reducing the dimensionality of the data, we used standard Support Vector Machines as well as Density Based Support Vector Machines algorithm to classify the fMRI data into two groups. K-fold cross validation which  $k=5$  was used for this experiment. Finally we made a comparison between the results of normal SVM and Density Based SVM which is shown in the table below (Table 1).

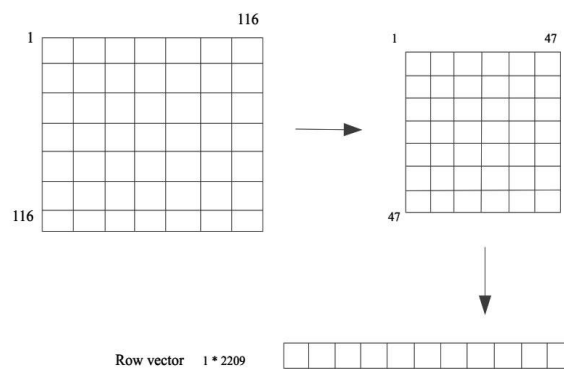


Fig. 1: Dimension reduction/feature selection using PCA.

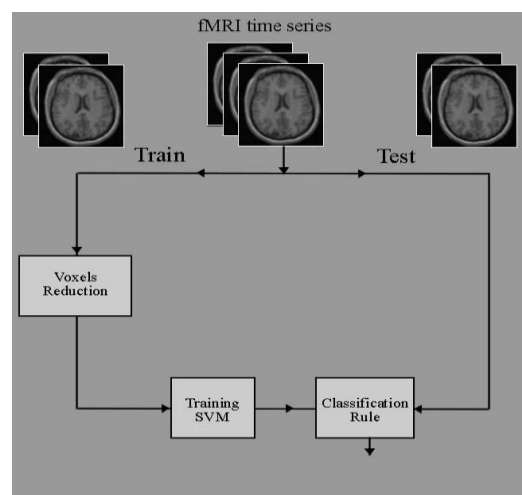


Fig.2: General description of fMRI data classification.

TABLE I: Results of fMRI data classification by SVM & DBSVM

fMRI data classification before dimensionality reduction			
SVM (linear)	DBSVM (linear)	SVM (RBF)	DBSVM (RBF)
61%	65%	67%	67%
fMRI data classification after dimensionality reduction			
SVM (linear)	DBSVM (linear)	SVM (RBF)	DBSVM (RBF)
76%	84%	88%	88%

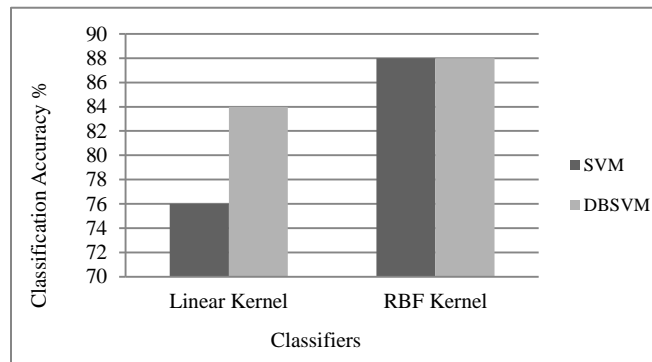


Fig.3: Comparison of classification results of SVM and DBSVM methods.

## 5. Conclusion

According to the results of classification of fMRI data by standard SVM and Density Based SVM, Density Based SVM performs better than standard SVM in case of linear kernel. Density Based SVM is considered to reduce the effects of noises and be able to improve the sensitivity of the SVM to noises and outliers. However there is no improvement in case of Radial Basis Function (RBF) kernel. The future work to be done is to use other dimension reduction methods as well as other pattern recognition algorithms to achieve a better classification result.

## 6. Acknowledgements

This study was supported by JSPS KAKEN Grant Number 26350995 and MEXT-Supported program for the Strategic Research Foundation at Private Universities, 2014-2018.

## 7. References

- [1] S. LaConte, S. Strother, V. Cherkassky, J. Anderson & X. Hu, "Support Vector Machines for Temporal Classification of Block Design fMRI Data", Elsevier Inc. January 2005, pp. 1-4.
- [2] R. A. Clay, "Functional Magnetic Resonance Imaging: A New Research Tool", American Psychological Association, 2007, pp. 2-10.
- [3] S.J. Peltier, J. M. Linsinski, D. C. Noll & S. M. LaConte, "Support Vector Machines Classification of Complex fMRI Data", 31th Annual International Conference of the IEEE EMBS, Minneapolis, Minnesota, USA, September 2009, pp 1-2.  
<http://dx.doi.org/10.1109/iembs.2009.5332805>
- [4] K. A. Norman, S. M. Polyn, G. J. Detre, J. V. Haxby, "Beyond mind-reading: multi-voxel pattern analysis of fMRI data", ELESVIER 2006, pp. 1-3.
- [5] C.Wu, "Feature Selection for fMRI Classification", Program of Computational Biology, Carnegie Mellon University. Research Report, pp. 1-4.

- [6] F. D. Martino, G. Valente, N. Staeren & J. Ashburner, "Combining Multivariate Voxel Selection and Support Vector Machines for Mapping and Classification of fMRI Spatial Patterns", Elsevier Inc., July 2008, pp. 1-3.
- [7] F. Pereira, T. Mitchell, M. Botvinick, "Machine Learning Classifiers and fMRI: a tutorial overview", Princeton University, July 2008, pp. 1-3.
- [8] D. D. Cox & R. L. Savoy, "Functional Magnetic Resonance Imaging (fMRI) brain reading: detecting and classifying distributed patterns of fMRI activity in human visual cortex", Elsevier Inc., December 2002, pp. 1-4.
- [9] Z. Nazari, D. Kang, "Density Based Support Vector Machines for Classification", International Journal of Advanced Research in Artificial Intelligence, Vol 4, Issue. 4. April 2015, pp. 1-7.  
<http://dx.doi.org/10.14569/ijarai.2015.040411>
- [10] C. M. Bishop, "Pattern Recognition and Machine Learning", Springer 2006, pp. 341-350.
- [11] V. N. Vapnik, "Statistical Learning Theory", AT&T Research Laboratories, John Wiley & Sons, 1998, pp. 19-25.
- [12] S. Abe, "Support Vector Machines for Pattern Classification", Springer 2010, pp. 21-43.  
<http://dx.doi.org/10.1007/978-1-84996-098-4>  
[http://dx.doi.org/10.1007/978-1-84996-098-4\\_2](http://dx.doi.org/10.1007/978-1-84996-098-4_2)
- [13] C. J. C. Burges, "Dimension Reduction: A Guided Tour", Microsoft Research, One Microsoft Way, Redmond, WA 98052-6399, USA, MSR Tech Report MSR-TR-2009-2013, pp. 4-12.