Reading minds using classification algorithms on fMRI data

Caroline Froehlich[1] **, Alexandre R. Franco**[2] **, Felipe Meneguzzi**[1]

[1]Faculdade de Informática , [2]Faculdade de Engenharia
Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS)

carol.frohlich@gmail.com, {alexandre.franco,felipe.meneguzzi}@pucrs.com.br

Abstract. *Functional Magnetic Resonance Imaging (fMRI) is a non-invasive method to obtain brain images that indirectly shows neuronal activation. With fMRI scans, we are able to measure areas of the brain that are active in time during extension of the exam, which are often transformed into a time-sequence of images. These images are then analyzed by human experts to infer information of interest. Recent work has used machine learning algorithms to extract more complex information from fMRI scans. In this paper we propose to use a classification based algorithm to differentiate, at each time point during the scan, whether a single patient is performing a task or not. We process the data to generate examples when the patient is performing a task or resting, and experiment different parameters for the classification algorithm to achieve a high success rate.*

1. Introduction

Functional Magnetic Resonance Imaging (fMRI) is an imaging test that indirectly measures neural activity over time. Some neural activity patterns are known to indicate a person's cognitive state or indicate if a patient has a neuropsychological disorder. Cognitive disorders such as autism [Just et al. 2012], alzheimer's disease [Woodard et al. 2009] and dementia [Woodard and Sugarman 2012] are known to have been successfully identified using fMRI data. Moreover, it has been shown that using fMRI, we are able to predict the cognitive state of patients (predict in what they are thinking or seeing) [Mitchell et al. 2003]. Additionally given its non-intrusiveness to patients, fMRI is widely recognized as a powerful diagnostic tool for conditions with a neurological basis. In this context, learning disabilities such as dyslexia are known to have a strong neurological basis [Wolf and Bowers 1999]. However, no standard test used today is able to detect or predict dyslexia precisely [Hoeft et al. 2011]. Thus, new tests are needed in order to fill this gap and our overarching objective is to reach this goal by using fMRI, an imaging test able to indirectly detect neural activity while people are resting or performing cognitive tasks.

There are a number of problems that need to be overcome when analyzing fMRI data to discover such patterns. For instance, there is a very large amount of information to analyze at the same time, since a regular test contains about 30,000 voxels. Furthermore, the difference between an activated portion of the brain and a deactivated portion is often minimal. Thus, there is significant potential for using machine learning algorithms to find complex brain activation patterns. Machine learning can work with a huge amount of portions of the brain and they are sensitive to minimal changes in the fMRI test.

In this paper, we report on our initial efforts to use classification techniques on fMRI data as a diagnostic tool for dyslexia sufferers by calibrating such algorithms to distinguish when a patient is engaged with a mental task or not. Namely, we use a classification technique to predict whether a single patient is performing a task or resting at any given time during the fMRI scan. Specifically, we train a Support Vector Machine (SVM) learning algorithm with fMRI data from a single patient, in which we have partitioned the training data as time points of the fMRI scan. We show that the resulting classifier has an average of 75% accuracy on the final test partition of the dataset.

This paper is organized as follows. Section 2 reviews the background in fMRI and machine learning, describing the SVM algorithm in detail. Section 3 presents the data we use and how we transform it to be the input of the SVM algorithm. Section 4 describes our model of the experiment with the fMRI data, while Section 5 shows the results of these experiments. Finally, in Section 6 we conclude with final considerations and point the direction of future work.

2. Background

2.1. fMRI

Neuroimaging involves different techniques to acquire images of the brain, each of which have distinct purposes, measuring and showing views of how a subject's brain is or how it works. Each exam has a distinct spatial and temporal resolution, and detect different tissues.

For example, MRI (Magnetic Resonance Imaging) is a neuroimaging test that shows a static view of the brain anatomy in detail. MRI data have a good spatial resolution, showing with millimeter accuracy brain tissue of white matter and gray matter. With the MRI scanner we can acquire a high resolution 3D image of the brain showing these tissues.

We are interested in fMRI (functional Magnetic Resonance Imaging) [Huettel et al. 2004] , that is a 4D functional neuroimage that consists of a time series of 3D images of the brain. While MRI acquires one high resolution image showing the brain structure, fMRI acquires many low resolution images in order to detect the activation of brain regions over time, since its purpose is to map brain activity [Biswal et al. 1995]. Compared to EEG, fMRI data has a high spatial resolution (it shows the brain details in millimeters, as MRI), and a low temporal resolution (because the MRI scanner typically takes a couple of seconds to adquire a single volume).

In MRI, we can observe in high detail how the brain structure is organized. In fMRI, we observe the level of activation of the brain, that means which brain region is activated when performing a specific task and how brain regions work together. For example, we can test which parts of the brain increase their activity when a subject is performing a task while inside the MRI scanner, such as moving the left hand or checking if two words rhyme. When more than one region is activated during a task, fMRI data can indicate how these brain parts interact: if they work synchronously (both regions are activated at the same time), when one is activated, the other deactivate, or if they are not related (the activation of one is not dependent on the other).

While the purpose of fMRI is to detect neural activity during experiments, it does not measure neural activity directly. Instead, it measures the fluctuation of oxygenated and deoxygenated hemoglobin in the blood flow. As oxygenated and deoxygenated hemoglobin have different magnetic properties, fMRI can detect changes in the hemoglobin molecule. Local neural activity increases oxygen consumption in cells, that increases local oxygenated hemoglobin level, so fMRI detects neural activity by measuring the change of deoxygenated to oxygenated hemoglobin, called BOLD (Blood Oxygen Level Dependent). Approximately two seconds after neural activity, there is change in the BOLD signal, that lasts 4 seconds, and we can see this changes in neuroimage.

We can obtain much information from fMRI data. First, we can see the brain function when someone is performing a cognitive or motor task, mapping the regions where there is more neural activity. Second, we can identify the cognitive state of a subject in a given time using fMRI data.

We can find brain activation patterns in subjects that are performing the same task by looking for brain regions that are activating or communicating in the same way. Also, we can identify when a single subject is performing a task by repeating the task several times and look for the brain regions that are always activated during the task. These patterns form biomarkers, that in this context are a set of characteristics (brain region activations) that identify a cognitive state.

In this paper, we are concerned with a specific type of fMRI test called task-based fMRI (used as a synonym of fMRI). In the task based-fMRI, the subject is required to perform a cognitive task, such as checking if 2 words rhyme. Stimuli (words) are presented on a screen for a few seconds while the subject is inside the scanner. Then, the subject has a few seconds to answer if the words rhyme, and a few seconds to rest. Each stimulus is repeated a number of times in order to identify clearly the brain regions that the task activates. In this type of experiment, we look for brain areas that are more activated during a trial. But knowing which brain areas are activated during a trial is not enough. Because the brain never ceases activity, data from this experiment shows voxels that are always activated when subjects are not performing any task in particular, and this information is not important to task-based experiments, where we want to see voxels that are activated in a specific task. Therefore, we acquire subjects baseline data, that is when the subject is not performing any task in particular and subtract the baseline data from task-based data to see just voxels involved in the task.

We divide the fMRI test in blocks. A block is a set of trials that a subject performs in row. For example, perform the rhyme task 6 times in row and then rest for a long period. Beside the task block, there are rest blocks, where the subject looks at a screen showing a fixation-cross (a black screen with a white cross in the middle) and not thinking in anything in particular. The baseline data is acquired in this block.

2.2. Support Vector Machine

Machine learning is an area of AI that studies techniques and algorithms to learn unknown functions from data [Russell and Norvig 2010], [Mitchell 1997]. There are many machine learning techniques, each of which is suitable for particular combinations of data and target functions. In this paper, we work with a class of machine learning algorithms called supervised learning. This type of algorithm solves the classification problem, namely, given examples E with labels Y, we want to learn a function that maps the examples to the labels (also called classes). Once this function is learned, it can generalize what it has learned from the previous examples and predict the label of the new example that the algorithm itself has never seen before.

More specifically, we work with a machine learning algorithm called Support Vector Machine (SVM) [Vapnik 2000]. SVM solves the binary classification problem, which is when each example is classified as being in one of two classes: positive and negative. SVM converts each example into a point in a Cartesian plane, and tries to find a hyperplane that divides the Cartesian plan in two: on one side there should be all positive examples while on the other side there should be all the negative examples. The type of hyperplane that divides the examples is determined by the kernel we use along with the SVM algorithm. A kernel is the type of function that we use to create the hyperplane, for example, a linear kernel creates a linear hyperplane that divides the examples. However, since not all examples can be linearly divided we sometimes need a non-linear kernel.

In this paper, we work with a non-linear kernel called RBF (Radial basis function) kernel [Powell 1987]. An RBF kernel has 2 parameters we can adjust: C and γ. The C parameter determines how the algorithm treats misclassified examples; if the value of C is low, the SVM training algorithm tries to find a smooth hyperplane, even if some examples are misclassified. Conversely, if the value of C is high, the SVM training algorithm tries to find a hyperplane that correctly classifies all examples. The γ parameter determines the level of influence a single example has on the boundary of the resulting hyperplane. Thus, if γ is set to a low value, each example makes the hyperplane that represents the class much bigger, whereas if γ is high, each example does not increase the hyperplane that represents the class so much.

3. Data

The data we used in our experiments was obtained in a series of fMRI exams aimed at evaluating the proficiency in reading and comprehension of subjects diagnosed with dyslexia. Such tests are collectively known as language tests, since their aim is to activate the brain regions associated with language skills. The particular language test performed by the patients during an imaging session involves what is known as a pseudo-word tasks. In this task, a visual stimulus in the form of a written word is presented on a screen and patients have a few seconds to answer whether the word presented is a real word or not. The pseudo-word task aims to activate brain regions related to language processing, such regions are different than the areas activated when patients are resting. Thus, our goal is to differentiate the neural activity pattern when the subject is performing the pseudo-word task and when the subject is resting (i.e. not thinking in anything in particular).

Therefore, the fMRI data generated by this test can be classified according to the stimulus being presented to the subject, namely: pseudo-word task (when a word is being shown to the subject) and rest (when no word is being presented to the subject). In the pseudo-word part, patients have to decided if a certain written word displayed in a screen is real (e.g. dream) or not (e.g. cra), indicating if the word is real by pressing a button. Each time a word is displayed to the subject is called a *trial*. A trial lasts 8 seconds, in which the word was presented for 7 seconds followed by a 1 second fixation-cross. Each block has 48 seconds and contains 6 trials. In the resting block, a fixation-cross was presented for 30 seconds. The duration of the entire scanning session was 272 seconds, composed of 4 pseudo-word blocks and 2 rest blocks. 2 extra rest blocks lasting 10 seconds were added between pseudo-word blocks.

We acquired 1 image for each two seconds of the test, which was subsequently upsampled by a factor of 2, yielding a total of 272 images and performed 2 transformations in the data in order to obtain the training examples. The fMRI scanner used to acquire data generates 3D images with 3x3x3 mm voxels. First, we transformed the images from a voxel size of 3x3x3mm into images of voxels with 2x2x2m using the AFNI neuroimaging tool[1] [Cox 1996] in order to apply an image segmentation technique based on image templates. Thus, because the template is 2x2x2mm, we need to adjust the test data to fit the template voxel size. Second, we used a template that divides the brain into 116 distinct regions [Alemán-Gómez et al. 2006], grouping voxels that belong to the same region. Basically, this template is a bit mask, which, for each region, turns off all voxels that does not belong to that region, and then extracts the time series of that region by calculating the average brain activation of the voxels at each second. By using this mask, we obtain 116 time series data representing the average activation of each region. After processing the data, we have a 272x116 matrix, where the 116 columns are the brain regions from the template and each line is the average brain activation in 1 second. Thus, each cell of this matrix is the average brain activation in one region in a given time, represented by a real number.

We now want to create 2 types of examples with the resulting matrix: when the subject is performing a task; and when the subject is resting. For the task examples, we have to look at the 7 seconds in which the word is presented, calculating the average activation of each brain region in that time. As we have seen in Section 2, the BOLD signal takes 2 seconds to start and lasts only 4 seconds. Then, from the total 7 seconds of the task, we have to skip 2 seconds in order to wait for the BOLD signal to occur and look at the next 4 seconds, which is the amount of time that the bold signal lasts. Thus, from the 7 seconds of a task, we calculate the average activation on seconds 3,4,5 and 6.

After applying such preprocessing on the task blocks, we are left with 24 task examples (4 blocks with 6 tasks each). For the remaining examples, we use the 2 rest blocks. For each rest block, we remove the first 4 seconds, waiting for the BOLD signal clear after the task block. Consequently since each rest block lasts 30 seconds (4 of which were removed to wait for the BOLD signal to clear), we are able to create 7 examples in sequence of the remaining 26-second block, where each example is the average brain activation in 4 seconds.

Finally, we have the examples to use with the classification algorithm: 24 task examples and 14 rest examples (7 examples for each block), for a total of 38 examples. Each example is a vector of 116 elements, where each element is the average brain activation in one region for 4 seconds.

4. Experiments

Our goal is to train an SVM classifier to differentiate between task and rest examples. For this purpose, we search for parameter values that we have seen in 2.2 in order to obtain a high prediction rate. We used the SVM algorithm from LIBSVM [Chang and Lin 2011] and performed the 6 steps below in our experiments.

1. We transformed the fMRI data in examples for the SVM algorithm, as we have seen in Section 3.

[1] AFNI is a popular neuroimaging software available at: `http://afni.nimh.nih.gov`

2. We scaled the features in data, so all features in all examples are real values between [-1,1]. Scaling data prevents overfitting, it reduces the differences between the features, minimizing the problem of one feature having much more importance than the others [Pereira et al. 2009].
3. In order to maximize the prediction accuracy, we have to find the best parameters for the SVM algorithm with the data, train the algorithm with these parameters and test the algorithm success rate. However, we cannot use the same examples for finding good parameters, training and testing. Using the same data for all of these steps can overfit the results. Therefore, we separated the data in 3 sets: the tuning set, which we use to find good parameters for the algorithm; the training set, which we use to train the algorithm using the parameters we have found and perform cross-validation and the test set, which we use to validate the classifier by measuring its expected accuracy.
4. After splitting the data, we need to find good parameters for the SVM algorithm using the tuning set. For this purpose, we use the RBF kernel along with the SVM algorithm, which maps the examples in a high dimensional space and fits well non-linear data [Keerthi and Lin 2003]. We use cross-validation several times with predefined parameters in order to optimize these parameters using Grid Search [Nelder and Mead 1965]. The Grid Search procedure gives us the parameters that generated the best prediction rate results.
5. We trained the SVM algorithm again setting the new parameters we found with Grid Search using the data from the test set, so the results are not overfitted. We use the training set to measure the SVM accuracy with leave-one-out cross-validation.
6. We validated the results by measuring the trained SVM true-accuracy with the test set, that the algorithm has never seen before.

5. Results

We now evaluate the accuracy of the experiments described in Section 4 using the data from 3 patients (referred to as P001, P002 and P003). In our experiments, each patient data was used to train a separate classifier.

As described in step 3, the data of each patients is split into 3 sets: the tuning set, the training set and the testing set. Table 1 shows the amount of examples of task and rest we use in each set, as the number of examples is unbalanced (there are more examples of task than rest).

set	task	rest	total
tuning	10	5	15
train	12	7	19
test	2	2	4

Table 1. Data of a single patient split in 3 sets

Table 2 shows the result of the experiment using the data of each patient. First, we present the RBF kernel parameter values, C and γ, that we find when tuning the SVM algorithm. Second, the cross-validation accuracy using the training data and the new

Patient	C	γ	cross-validation accuracy	true-accuracy
P001	2	0.0078125	78.94%	75%
P002	2	0.0078125	73.68%	100%
P003	128	0.00048828125	42.1%	50%

Table 2. Results of the experiments described in section 4

parameter values. Finally, we show the expected accuracy of the trained SVM in the test set.

When looking at the results of Table 2, we note that we work with a limited number of examples. To measure the expected accuracy there are only 4 examples, so that if the SVM algorithm misclassifies one example we lose 25% of the expected accuracy. Notice that the accuracy for the SVM classifier generated with P001 and P002 data is high, and their respected tuning parameters are the same. We observe that the selected parameters are not distant from the the starting parameters ($C = 8$, $\gamma = 0.5$). On the other hand, the SVM algorithm accuracy for P003 data is equal to random guessing, and the parameter values are far from the other patients data and and the starting parameters from the tuning algorithm.

In order to verify if all the steps in section 4 contribute to increase the prediction rate of the SVM algorithm, we created other 3 experiments by removing some important steps. In this first experiments, we do not scale the data before tuning the algorithm parameters, training and testing. In the second experiment, we do not boost the algorithm parameters. And in the third experiment, we do not scale the data nor boost the parameters. The true accuracy of each experiment is in table 3.

Patient	no scale	no parameter boosting	no scale and no parameter boosting
P001	75%	50%	50%
P002	100%	100%	50%
P003	75%	50%	50%

Table 3. True accuracy when we do not scale the data or boost the parameters

The results of table 3 show two important facts. First, scaling is not very helpful on this data. Although the true accuracy for P001 and P002 data when scaling or not is the same, the true accuracy of P003 data increases when not scaling. Looking at the not scaled data, we notice that it is between [-4, 3]. Thus, the difference between the data values is not significant enough to perform scaling, and the true accuracy gets better when not scaling. Second, when not boosting the parameters of the SVM algorithm, only the true accuracy of P001 data decreases. As the difference between the default parameter values and the parameter values found from the tuning algorithm are close, not tuning the parameter does not change dramatically the true accuracy from P001 and P002 data. The tuning algorithm seems not to have found good parameters for P003 data, as they are too different from P001 and P002 parameter values and the resulting accuracy is the same as random guessing. Thus, changing the parameter values from P003 data does not change the accuracy.

We have seen that our approach for obtaining a high prediction rate and differentiate task examples for rest examples need improvements, as we do not obtained the expected results from all patients data. Moreover, we learn which are the relevant steps for finding good results, and why some steps are unnecessary.

6. Conclusion

In this paper we described an application of machine learning algorithms to discover what a patient is doing at any given time in the fMRI scan session. To apply the machine learning algorithm to our problem, we transformed the fMRI test in readable data for the SVM algorithm and created experiments to test the true accuracy of the SVM algorithm using the data. in order to evaluate which steps of our main experiments are relevant, we created other smaller experiments and compared the results.

As future work, we aim to generalize the classifier so that we can train using the data from multiple patients and predict the classification of examples of data from an unknown patient. Moreover, we want generate an SVM classifier to differentiate when a patient is seeing a word or a pseudo-word.

Acknowledgement

We would like to thank Augusto Buchweitz from The Brain Institute of Rio Grande do Sul (InsCer) for making available the data collected under the ACERTA project. We would also like to thank CAPES for financial support under the OBEDUC project number 14385.

References

Alemán-Gómez, Y., Melie-García, L., and Valdés-Hernandez, P. (2006). Ibaspm: toolbox for automatic parcellation of brain structures. In *12th Annual Meeting of the Organization for Human Brain Mapping*, volume 27.

Biswal, B., Zerrin Yetkin, F., Haughton, V. M., and Hyde, J. S. (1995). Functional connectivity in the motor cortex of resting human brain using echo-planar mri. *Magnetic resonance in medicine*, 34(4):537–541.

Chang, C.-C. and Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Cox, R. W. (1996). Afni: software for analysis and visualization of functional magnetic resonance neuroimages. *Computers and Biomedical research*, 29(3):162–173.

Hoeft, F., McCandliss, B. D., Black, J. M., Gantman, A., Zakerani, N., Hulme, C., Lyytinen, H., Whitfield-Gabrieli, S., Glover, G. H., Reiss, A. L., et al. (2011). Neural systems predicting long-term outcome in dyslexia. *Proceedings of the National Academy of Sciences*, 108(1):361–366.

Huettel, S. A., Song, A. W., and McCarthy, G. (2004). *Functional magnetic resonance imaging*, volume 1. Sinauer Associates Sunderland.

Just, M. A., Keller, T. A., Malave, V. L., Kana, R. K., and Varma, S. (2012). Autism as a neural systems disorder: a theory of frontal-posterior underconnectivity. *Neuroscience & Biobehavioral Reviews*, 36(4):1292–1313.

Keerthi, S. S. and Lin, C.-J. (2003). Asymptotic behaviors of support vector machines with gaussian kernel. *Neural Computation*, 15(7):1667–1689.

Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45.

Mitchell, T. M., Hutchinson, R., Just, M. A., Niculescu, R. S., Pereira, F., and Wang, X. (2003). Classifying instantaneous cognitive states from fmri data. In *AMIA Annual Symposium Proceedings*, volume 2003, page 465. American Medical Informatics Association.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer journal*, 7(4):308–313.

Pereira, F., Mitchell, T., and Botvinick, M. (2009). Machine learning classifiers and fMRI: A tutorial overview. *NeuroImage*, 45(1, Supplement 1):S199–S209. Mathematics in Brain Imaging.

Powell, M. J. (1987). Radial basis functions for multivariable interpolation: a review. In *Algorithms for approximation*, pages 143–167. Clarendon Press.

Russell, S. J. and Norvig, P. (2010). *Artificial intelligence: a modern approach*, volume 2. Prentice hall.

Vapnik, V. (2000). *The nature of statistical learning theory*. springer.

Wolf, M. and Bowers, P. G. (1999). The double-deficit hypothesis for the developmental dyslexias. *Journal of Educational Pscyhology*, 91(3):415–438.

Woodard, J., Seidenberg, M., Nielson, K., Antuono, P., Guidotti, L., Durgerian, S., Zhang, Q., Lancaster, M., Hantke, N., Butts, A., et al. (2009). Semantic memory activation in amnestic mild cognitive impairment. *Brain*, 132(8):2068–2078.

Woodard, J. L. and Sugarman, M. A. (2012). Functional magnetic resonance imaging in aging and dementia: Detection of age-related cognitive changes and prediction of cognitive decline. In *Behavioral Neurobiology of Aging*, pages 113–136. Springer.