Online Learning for Real Time fMRI Classification

H. Xu[1], Y. T. Xi[1], R. Lee[2], and P. J. Ramadge[1]

[1]Electrical Engineering, Princeton University, Princeton, NJ, United States, [2]Princeton Neuroscience Institute, Princeton University, Princeton, NJ, United States

Introduction: Real-time functional magnetic resonance imaging (rtfMRI) enables investigations of brain activity during an ongoing experiment. Conventionally, inference (e.g. classification) from fMRI data is done off-line after all data is acquired. In contrast, rtfMRI generates inference as data collection proceeds and it offers several potential advantages: 1) directly reading a subject's cognitive state in real-time; 2) monitoring data quality and locating regions of interest as data is collected; 3) providing feedback to the subject based on the output of an adaptive real-time classifier; 4) adaptively changing an experimental stimulus based on real-time feedback to achieve better activation.

In rtfMRI, an effective and efficient algorithm is essential. Here we consider *online* algorithms that both learn and classify as data is being collected. The main challenges are that the algorithm must learn online to accurately predict brain states from high dimensional fMRI data (≥122,880 voxels per TR), and do so within tight time constraints (one TR = 2-3 secs). Our contribution is a new *online learning* algorithm, real-time conjugate gradient (rtCG), which attains competitive classification accuracy within the constraints required. It is closely connected to partial least squares (PLS), a popular offline analysis method. We demonstrate its effectiveness in actual real-time fMRI tests. Test results show that the online rtCG classifier: is fast (training time < 0.5 secs), is accurate (prediction accuracy around 90%), and yields better classification performance than standard PLS applied to a sliding window of recent data.

Methodology: In this study, we seek an efficient online algorithm with comparable performance to PLS [1]. PLS assumes that the input and output data are both generated by a small number of latent factors. It iteratively finds these factors by maximizing the covariance between input and output. PLS uses a relatively small number of factors, which ensures the stability of resultant predictor derived from it. PLS performs well, but is computationally expensive for rtfMRI. A more efficient algorithm in the streaming data setting, Incremental Sparse Bridge PLS (iSB-PLS) [2], is also computationally expensive on high dimensional fMRI data. [3] shows that PLS is a conjugate gradient (CG) [4] algorithm. This motivates our study of a CG based algorithm for rtfMRI.

Our online learning algorithm has two steps: (1) at time *t*, fMRI signal $x(t) \in R^k$ arrives and the classifier predicts its label $\hat{l}(t) = \{-1,1\}$; (2) the actual labe $l(t) = \{-1,1\}$ is provided and the classifier is updated. Here we consider the linear classifier $b(t) \in R^k$ with label prediction $\hat{l}(t) = sign[x(t)^T b(t-1)]$. For real-time application, we bring in a window parameter *h* and let $X(t,h) = [x(t-h+1),...,x(t)]$ and $y(t,h) = [l(t-h+1),...,l(t)]$. $b(t-1)$, $X(t,h)$ and $y(t,h)$ are inputs to the update step at time *t* that yields $b(t)$. The data window limits the amount of high dimensional data used and mediates the computation resources (time and space) required. In this setting, consider the update rule: $b(t) = \arg\min \|X(t,h)b - y(t,h)\|^2$. Note that this is a quadratic problem. Standard conjugate gradient (CG) solves it iteratively in *k* steps by finding conjugate search directions $d(i)$ with $d(i)^T (X(t,h))^T X(t,h) d(j) = 0$, $i \neq j$. In each iteration, CG first updates the current estimate $s(i)$ by minimizing the objective in direction $d(i)$. Then CG updates the residual $r(i)$ and finds a new conjugate direction $d(i+1)$ based on $r(i)$. After *k* iterations, CG solves the problem with $b(t) = s(k)$. To transform CG to an online version, we make two improvements. First, we note that the change from $X(t-1,h)$ to $X(t,h)$ involves removing one and adding one column. So we expect $b(t-1)$ to be close to $b(t)$. Hence $b(t-1)$ is used to initialize the CG iterations at time *t*, i.e. $s(0) = b(t-1)$ at time *t*. Second, we limit the number of iterations of CG to $I(max)$ << *k* since CG solves the quadratic problem in *k* steps but for fMRI data *k* is very large ($\approx 10^5$). Algorithm 1 summarizes the rtCG algorithm.

Experiments: We compare the performance of three algorithms on two real fMRI data tests: 1) real-time CG (rtCG); 2) real-time PLS (rtPLS); this is PLS applied to $X(t,h)$ and $y(t,h)$; 3) real-time BPLS (rtBPLS); this is iSB-PLS with its sparse setting part removed, applied to $X(t,h)$ and $y(t,h)$. The first fMRI test is a visual perception test in which the subject is shown a flashing checker board either on the left or right side of the visual field. Right/left side of primary visual cortex in the brain deals with the left/right half of the field of view from both eyes. Each flashing checker board is on for 15s and each TR is 3s, so we get 5 examples for each side alternatingly. Each fMRI data example has dimension 30*64*64 = 122,880. The second fMRI experiment is the same, except that the location of the flashing checkerboard (the label) is presented randomly.

For both tests, we set *h*=20, $I(\max) = 10$ and convergence tolerance to1 ($\approx 10^{-10}$ of the average example norm). Table 1 shows that rtCG outperforms rtPLS and rtBPLS in both prediction accuracy and average computation time. We then evaluate the impact of window size parameter *h* and iteration parameter $I(max)$ on rtCG. To do so, we vary *h* from 10 to 100 and test the accuracy and time with other variables fixed as before. Fig 1 shows that the accuracy doesn't improve as *h* increases, instead it reached an optimum at approximately *h*=30. It also shows the time increases almost linearly as *h* increases. We also vary $I(max)$ from 10 to 100 with other variables fixed. Fig 1 shows that the accuracy remains the same and time increases linearly as $I(max)$ increases. These suggest that *h* and $I(max)$ can be reduced considerably without impacting the prediction accuracy. We further evaluate the spatial significance of the rtCG classifier $b(t)$ in brain image. Fig. 2(1) shows the covariance between each voxel's value in all examples and sequence of labels in fMRI 1 test. This measures how each voxel changes with the label. Large covariance indicates the voxel is associated with the task. Fig.2(2) shows the average value of the classifiers $[b(1)+...+b(n)]/n$. Large coefficient in the classifier indicates more weight is given to the corresponding voxel. Comparing these, we can see that they are consistent, which means the rtCG classifier captures the correct activated voxels during the task.

Conclusion: We propose a new online learning algorithm rtCG for rtfMRI system. Our tests show that rtCG can process high dimensional fMRI data within one TR with a high accuracy. More complex experiments are planned to test rtCG and evalute its performance.

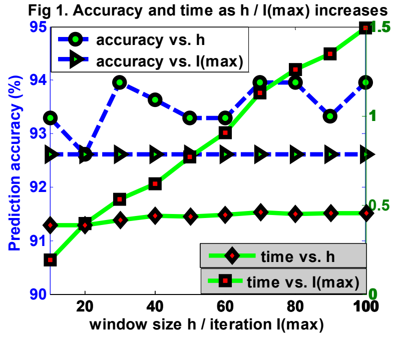
Algorithm 1 Real-time CG (rtCG) Algorithm

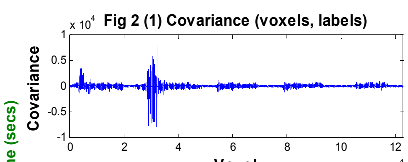
1. Initialize: $b(0)^{rtCG} = 0$
2. at time t: Let $\beta = X(t,h)^T y(t,h)$, $A = X(t,h)^T X(t,h)$
   $r(0) = d(0) = \beta - As(0)$, $s(0) = b(t-1)^{rtCG}$
   while $i \leq I(max)$ & residual $\geq$ tolerance do
   $s(i+1) = s(i) + \alpha(i)d(i)$, where $\alpha(i) = r(i)^T r(i)/d(i)^T Ad(i)$
   $r(i+1) = r(i) - \alpha(i)Ad(i)$
   $d(i+1) = r(i+1) + \theta(i+1)d(i)$, where $\theta(i+1) = r(i+1)^T r(i+1)/r(i)^T r(i)$
   end while
3. Output: $b(t)^{rtCG} = s(i+1)$

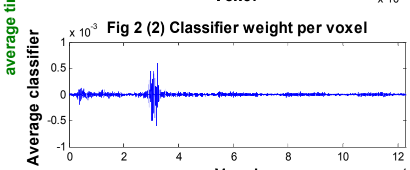
Table 1. Results on rtCG, rtPLS and rtBPLS

	rtCG	rtPLS	rtBPLS
	prediction accuracy (%) / average time (secs)		
fMRI 1	92.67 / 0.4149	74 / 1.4903	73.33 / 0.8675
fMRI 2	91.05 / 0.4082	75.26 / 1.472	73.16 / 0.9045

Fig 1. Accuracy and time as h / I(max) increases







Reference: 1. S. Wold, A. Ruhe et al. The Collinearity Problem in Linear Regression. The Partial Least Squares (PLS) Approach to Generalized Inverses. SIAM J. Sci. and Stat. Comput. 1984; 5: 735-743.

2. Brian McWilliams, Giovanni Montana. Sparse partial least squares for on-line variable selection in multivariate data streams. arXiv:0902.1323v1. 2009.

3. Aloke Phatak, Frank de Hoog. Exploiting the connection between PLS, Lanczos methods and conjugate gradients: alternative proofs of some properties of PLS. J. Chemom. 2002; 16 :361-367.

4. Magnus R. Hestenes, Eduard Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. J. Res. Natl. Bur. Stand. 1952; 49