Journal of Neuroscience Methods 194 (2011) 402–406

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Short communication

Dynamic causal modeling with genetic algorithms

M. Pyka[a,*,1], D. Heider[b,1], S. Hauke[c], T. Kircher[d], A. Jansen[a]

[a] Section of Brain Imaging, Department of Psychiatry und Psychotherapy, Philipps-University Marburg, Rudolf-Bultmann-Str. 8, 35039 Marburg, Germany
[b] Department of Bioinformatics, Center for Medical Biotechnology, University of Duisburg-Essen, Germany
[c] FHDW, Bergisch Gladbach, Germany
[d] Department of Psychiatry und Psychotherapy, Philipps-University Marburg, Germany

ARTICLE INFO

Article history:
Received 27 May 2010
Received in revised form 13 October 2010
Accepted 10 November 2010

Keywords:
Dynamic causal modeling
Genetic algorithm
fMRI

ABSTRACT

In the last years, dynamic causal modeling has gained increased popularity in the neuroimaging community as an approach for the estimation of effective connectivity from functional magnetic resonance imaging (fMRI) data. The algorithm calls for an *a priori* defined model, whose parameter estimates are subsequently computed upon the given data. As the number of possible models increases exponentially with additional areas, it rapidly becomes inefficient to compute parameter estimates for all models in order to reveal the family of models with the highest posterior probability. In the present study, we developed a genetic algorithm for dynamic causal models and investigated whether this evolutionary approach can accelerate the model search. In this context, the configuration of the intrinsic, extrinsic and bilinear connection matrices represents the genetic code and Bayesian model selection serves as a fitness function. Using crossover and mutation, populations of models are created and compared with each other. The most probable ones survive the current generation and serve as a source for the next generation of models. Tests with artificially created data sets show that the genetic algorithm approximates the most plausible models faster than a random-driven brute-force search. The fitness landscape revealed by the genetic algorithm indicates that dynamic causal modeling has excellent properties for evolution-driven optimization techniques.

1. Introduction

Dynamic causal modeling is a general framework for inferring processes and mechanisms at the neuronal level from the measurement of brain activity with different imaging techniques, for example, functional magnetic resonance imaging (fMRI). With increasing computational power and the continuing extension of dynamic causal modeling (Stephan et al., 2009), it became affordable to test a multitude of dynamic causal models (DCMs) across subjects automatically. Thereby, the characteristics of a perceptually driven network can be described more as common properties underlying a family of DCMs with similar posterior probability rather than through a single DCM (Penny et al., 2010). However, as the number of possible dynamic causal models exponentially increases with each additional brain area and with each additional extrinsic input, a systematic brute-force search through the whole space of DCMs is computationally inefficient. Therefore, we analyzed the applicability of genetic algorithms (GAs) to accelerate the search for the most plausible DCMs.

GAs are based on the idea of evolution and mimic natural processes, such as mutation, recombination and selection of individuals in an artificial environment. The first described procedure, which included all of the essential elements of GAs, was published by Bremermann (1962). During the 1960s and 1970s, GAs became increasingly popular due to the work of Rechenberg and Schwefel, who solved complex problems by applying evolutionary approaches (Rechenberg, 1973; Schwefel, 1974).

GAs are used for optimization problems with a complex fitness landscape, for instance, NP-complete problems such as the knapsack problem (Kellerer et al., 2004). In addition, GAs are often applied in bioinformatics or physical research to get approximations in adequate time. In bioinformatics, GAs are used, for instance, in peptide and protein design (Gronwald et al., 2008; Suarez et al., 2010).

The basis of a GA is a population of solutions and a fitness function. The fitness function is used to evaluate the solutions, often called *individuals* or *chromosomes*, and the next generation is built by applying different selection methods such as stochastic universal sampling or tournament selection (Baker, 1987; Goldberg and Deb, 1991). Additionally, genetic operators, mutation and crossover, can be applied to increase the so-called *gene pool*. While

* Corresponding author. Tel.: +49 6421 58 65076; fax: +49 6421 58 68939.
E-mail addresses: martin.pyka@med.uni-marburg.de, martin.pyka@gmx.de (M. Pyka).
[1] These authors contributed equally to this study.

0165-0270/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.jneumeth.2010.11.007

mutation only changes small parts of the chromosome, crossover leads to a combination of two or even more parents to a novel offspring. The selection method decides which individuals are transferred into the next generation and which are discarded. This procedure is repeated until a termination condition, for instance, a fixed number of generations, has been reached.

In the context of dynamic causal modeling, the configuration of the connectivity matrices *A*, *B* and *C* represents the genetic code of an individual and the posterior model probability serves as a fitness function. In the current study, we propose a genetic algorithm for dynamic causal modeling and test whether the algorithm is able to find plausible DCMs faster than a brute-force search. On the basis of these results, we reveal the properties of the fitness landscape of DCMs that support the usage of GAs.

2. Methods

In fMRI studies, the search for the most plausible DCM is typically constrained by *a priori* assumptions about the effective connectivity of the selected areas. The remaining space of possible network configurations can subsequently be browsed for the best model (Penny et al., 2010). As the general search principles of GAs are the same for constrained and unconstrained DCMs, we decided to use in the present study the whole space of possible network configurations to investigate the performance of the evolutionary approach.

Our genetic algorithm works as follows: at first, randomly chosen connectivity matrices that define the intrinsic connections, the bilinear modulatory connections, and the driving inputs are transformed into a single vector representing the binary genetic code of a DCM. Three derivatives of this code were generated by changing eight randomly chosen bits in the vector. These four genetic codes are used to start the iterative GA. First, a population of 20 DCM codes is created, comprising of the four original chromosomes and 16 chromosomes that are the result of mutation and crossover between the four initially created chromosomes. Crossover is accomplished by randomly choosing two codes and two crossover points that determine the part of the code that is replaced by the information of the second code. Mutation is applied on each new code with a probability of 50%, which changes two to eight randomly selected bits within the chromosome. The configuration of matrix *B* depends on the selected connections of matrix *A*. Therefore, all impossible bilinear connections in matrix *B* are deleted. Furthermore, a routine checks whether a created model exists already in the current or in any generation before. If this is the case, the algorithm generates another model until a unique one has been found. Subsequently, all codes are transformed back to the DCM format and are estimated with SPM8 software (http://www.fil.ion.ucl.ac.uk/spm/). Bayesian model selection, as implemented in SPM8, serves as fitness function to create a ranking for all estimated DCMs (Penny et al., 2004). Third, the best four models enter the GA again. After 50 iterations the GA stops.

2.1. Test with artificial data sets

To analyze whether the genetic approach can optimize the search for the most plausible DCMs, we compared it to a random driven brute-force search. The brute-force algorithm generates three integer values, whose binary representation is converted to DCM matrices *A*, *B* and *C*, respectively, to guarantee that each DCM configuration can occur with the same probability. For example, if a DCM with 4 brain areas and 3 external stimuli has to be created, matrix *A* consists of 12 configurable binaries (the main diagonal elements in this matrix represent self-referential inhibitory connections and are therefore fixed). That leads to 2^{12} distinct models for matrix *A*. In general, the selection of matrix *A* is constrained by *a priori* assumptions, and thus the real number of configurations is lower. But, as mentioned before, for a methodical comparison of two DCM search algorithms, this constraint can be neglected.

Since the bilinear influence of stimuli depends on the previously defined connectivity matrix *A*, the second randomly generated number ranges to the number of connections that are determined in matrix *A* times the number of modeled regressors. *C* is a $n \times s$-matrix, where *n* is the number of brain areas and *s* is the number of modeled regressors. Therefore, the binary structure of this matrix is created by a randomly generated number between zero and 2^{n+s}. Thus, we ensured that each model can occur with the same probability. Again, a control routine assures that no DCM occurs more than one time.

The brute-force algorithm creates two random models, as described above, and compares them using Bayesian model selection. Subsequently, the better model is transferred to the next iteration, in which another model is randomly created and compared with the previously selected model using Bayesian model selection. The algorithm terminates when the number of generated models exceeds the number of models generated by the GA for the same data set.

To evaluate the efficiency of the GA, we created synthetic data from two predefined DCMs. The first model consisted of three areas and two driving inputs and contained the same intrinsic properties and similar driving inputs as the bilinear model used for the generation of synthetic data in Stephan et al. (Stephan et al., 2008). Driving input u_1 was created as a set of delta-functions randomly distributed over 100 time points. A box-car function with two blocks (25 s duration) served as second driving input u_2 (see Fig. 3b in Stephan et al., 2008). A second model was derived from this configuration by adding a fourth region to the system with reciprocal connections to area three and a directed connection from area two to area four. For both models, synthetic data were generated with the SPM build-in function spm_dcm_create. As suggested in Stephan et al. (2008), we created synthetic data with a low and a high signal-to-noise ratio (SNR = 2 and SNR = 5, respectively), reflecting a relatively good signal quality, as it is usually the case in real data due to the computation of the first eigenvariate of the time series in a volume of interest. Overall, we generated four synthetic data sets: for the three area model one data set with SNR = 2 and one with SNR = 5, and for the four area model one data set with SNR = 2 and one with SNR = 5. The binary vector derived from the model with three areas included 24 bins (6 parameters for matrix *A*, 2×6 parameters for matrix *B* and 2×3 parameters for matrix *C*) leading to 2^24 theoretical possible models. The binary vector for the model with four areas included 44 bins (12 parameter for matrix *A*, 2×12 parameters for matrix *B* and 2×4 parameters for matrix *C*) leading to 2^44 theoretical possible models.

For each data set, 10 runs with the GA (each with 50 generations) and with the brute-force search were performed starting from the same random model for each data set, resulting in 64320 created and estimated models. We compared both algorithms using three different criteria: First, we determined after each run the number of better models found by the GA and the brute-force algorithm, respectively, compared to the initial model. Second, after all runs on one data set, we computed the posterior model probability of the best models found by the genetic algorithm and the brute-force search algorithm using Bayesian model selection. Third, we checked how many times the GA approximated to the same model after each run.

A successful genetic algorithm has to head for the global optimum and does not have to be caught in a local optimum of the fitness landscape. To reveal the landscape of the fitness function, we therefore determined the Hamming distance between each DCM model and the best DCM model of each run and compared it with its posterior-log evidence, using the negative free-energy value *F*