Computational Neuroimaging
Andreea Diaconescu
Methods & Models 2017
What is it all about?

- Why do we use functional magnetic resonance imaging?
  - To measure brain activity

- When does the brain become active?
  - When it learns
    i.e., when its predictions have to be adjusted

- Where do these predictions come from?
  - A model
How to build a model

Sensory Input

Predictions

Inferred Hidden States

True Hidden States

“Prediction Error”
Computational Neuroimaging

1. Computational learning model:
   \[ \mu_2^{(k+1)} = \mu_2^{(k)} + \psi_2^{(k)} \delta_1^{(k)} = \varepsilon_2^{(k)} \]

2. Computational trajectories:
   - Canonical and derivative
   - Trials

3. GLM with basic and parametric regressors:
   - Scans

4. Statistical parametric map and time series from SFG:
   - Brain image with highlighted areas
   - Time series for different scans

Iglesias et al., 2016
Advantages of computational neuroimaging

- Computational neuroimaging permits us to:
  - **Infer** the computational mechanisms underlying brain function
  - **Localize** such mechanisms
  - **Compare** different models
Explanatory Gap

Biological
- Molecular
- Neurochemical

Cognitive
- Computational
- “cognitive/computational phenotyping”

Phenomenological
- Performance Accuracy
- Reaction Time
- Choices, preferences

Computational Models
Three Levels of Inference

- **Computational Level**: predictions, prediction errors
- **Algorithmic Level**: reinforcement learning, hierarchical Bayesian inference, predictive coding
- **Implementational Level**: Brain activity, neuromodulation

**3 ingredients:**

1. Experimental paradigm:
   - Adviser
   - Player

2. Computational model of learning:

3. Model-based fMRI analysis:

---

David Marr, 1982
Outline

1. Computational
2. Algorithmic
3. Implementational
4. Application to Psychiatry
Example of a simple model

Rescorla-Wagner Learning:

\[ \mu^{(k)} = \mu^{(k-1)} + \alpha (u^{(k)} - \mu^{(k-1)}) \]
Example of a simple model

Rescorla-Wagner Learning:

$\Delta \mu^{(k)} \propto \alpha \delta$

Belief Update

Learning Rate
Computational Variables

Methods & Models: Computational Neuroimaging
Example of a simple model

Rescorla-Wagner Learning:

\[ \Delta \mu^{(k)} \propto \alpha \delta \]
Example of a hierarchical model

Hierarchical Gaussian Filter:

\[ \Delta \mu^{(k)} \propto \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta \]

Belief Update

Weight

\[ \text{Weight} = \frac{\text{how much we're learning here}}{\text{how much we already know}} \]
Inference is Hierarchical

Mathys et al., *Frontiers Human Neurosci* 2011

- Learning rate as precision-weighted prediction errors

\[
\Delta \mu_i^{(k)} \propto \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} PE
\]
Inference is Hierarchical

Mathys et al., *Frontiers Human Neurosci* 2011

\[ \Delta \mu_i^{(k)} \propto \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \]

- Learning rate as precision-weighted prediction errors
Outline

1. Computational
2. Algorithmic
3. Implementational
4. Application to Psychiatry
Perception (learning) via hierarchical interactions

Top-down; Predictions

Filter

Update

Bottom-up; Sensations/Prediction Error
From perception to action

Agent \quad World

Perceptual model with parameters $\chi$

Inferred hidden states

Response model with parameters $\zeta$

Sensory input

Response

True hidden states

$x$

$\lambda$

$\gamma$

$\chi$

$u$
From perception to action

Inversion of Perceptual Model

Perceptual model with parameters $\chi$

Inferred hidden states $\lambda$

Response model with parameters $\zeta$

Response $\gamma$

Sensory input

Agent

World

Generative Model

True hidden states $\chi$

Methods & Models: Computational Neuroimaging
From perception to action to observation

Inversion of Perceptual Model

Perceptual model with parameters $\chi$

Inferred hidden states

Experimenter

Generative Model

True hidden states

Inversion of Response Model

Response model with parameters $\zeta$

Response
Observing the observer

• The observer obtains input from the world via the sensory systems

• He/she has prior beliefs about the state of the world and how it is changing.

• Based on these prior beliefs and the sensory inputs, he makes predictions.

Daunizeau et al., PONE, 2011
Observing the observer

- The observer obtains input from the world via the sensory systems.
- He/she has prior beliefs about the state of the world and how it is changing.
- Based on these prior beliefs and the sensory inputs, he makes predictions.
- As the experimenter, we want to infer on what the observer is thinking ...
- But all we can observe is his/her behaviour.
- We invert the observer’s beliefs from his/her behaviour: computational model.

Daunizeau et al., PONE, 2011
Bayesian Models

The Bayesian Brain

- The brain is an inference machine
- Conceptualise beliefs as probability distributions
- Updates via Bayes’ rule:

\[
p(\Theta | y, m) = \frac{p(\Theta | m)p(y | \Theta, m)}{\int p(\Theta | m)p(y | \Theta, m)d\Theta}
\]

Prediction Errors

Sensory Data

Prior Belief

Posterior Belief

Evidence
Hierarchical Gaussian Filter

• Updates as precision-weighted prediction errors

\[ \Delta \mu_i^{(k)} \propto \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_i^{(k)} \]

Sensory Precision

Belief Precision

Predictions

Mathys et al., *Frontiers Human Neurosci* 2011
Mathys et al., *Frontiers Human Neurosci* 2014
Dark Room Experiment
The hierarchical Gaussian filter (HGF): a computationally tractable model for individual learning under uncertainty

Level 3: Phasic volatility
\[ p \left( x_3^{(k)} \right) \sim \mathcal{N} \left( x_3^{(k-1)}, \vartheta \right) \]

Level 2: Tendency towards category 1
\[ p \left( x_2^{(k)} \right) \sim \mathcal{N} \left( x_2^{(k-1)}, e^{(\kappa x_3^{(k-1)} + \omega)} \right) \]

Level 1: Stimulus category
\[ p(x_1 = 1) = \frac{1}{1 + e^{-x_2}} \]

Mathys et al., *Front Hum Neurosci*, 2011
The hierarchical Gaussian filter (HGF): a computationally tractable model for individual learning under uncertainty

Mathys et al., *Front Hum Neurosci*, 2011
HGF: Variational inversion and update equations

- Inversion proceeds by introducing a mean field approximation and fitting quadratic approximations to the resulting variational energies.

- This leads to simple one-step update equations for the sufficient statistics (mean and precision) of the approximate Gaussian posteriors of the hidden states $x_i$.

- The updates of the means have the same structure as value updates in Rescorla-Wagner learning:

$$
\Delta \mu_i = \frac{\hat{\pi}_{i-1}}{\pi_i} \delta_{i-1}
$$

Precisions determine the learning rate
Hierarchical Learning

Simulations: $\theta = 0.5$, $\omega = -2.2$, $\kappa = 1.4$
From perception to action

- In behavioural tasks, we observe actions $a$
- How do we use them to infer on beliefs $\lambda$?
- Answer: we invert (estimate) a response model
Example of a simple response model

- Options A, B and C have values: \( v_A = 8, v_B = 4, v_C = 2 \)

- We translate these values into action probabilities via a Softmax function:

\[
p(a = A) = \frac{e^{\beta v_A}}{e^{\beta v_A} + e^{\beta v_B} + e^{\beta v_C}}
\]

- Parameter \( \beta \) determines sensitivity to value differences:

low \( \beta \)  

<table>
<thead>
<tr>
<th>( \beta = 0.1 )</th>
<th>( \beta = 0.6 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>low ( \beta )</td>
<td>high ( \beta )</td>
</tr>
</tbody>
</table>
All the necessary ingredients

- Perceptual model (updates based on prediction errors)
- Value function (inferred state to action value)
- Response model (action value to response probability)
Outline

1. Computational
2. Algorithmic
3. Implementational
4. Application to Psychiatry
Computational fMRI: The advantage

The question event-related/block designs answer:
- Where in the brain do particular experimental conditions elicit BOLD responses?

The question model-based fMRI answers:
- How (i.e., by activation of which areas) does the brain implement a particular cognitive process?

It is able to do so because its regressors correspond to particular cognitive processes instead of experimental conditions.
Computational fMRI analyses of neuromodulation

Iglesias et al., 2016
Computational fMRI analyses of neuromodulation

Iglesias et al., 2016

Dopamine

Acetylcholine

Serotonin

Noradrenaline

Iglesias et al., 2016
Application of the HGF: Two types of PEs

1. Outcome PE

2. Cue-Outcome Contingency PE

Iglesias et al., *Neuron*, 2013
Application of the HGF: Sensory Learning

Changes in cue strength (black), and posterior expectation of visual category (red)

Iglesias et al., Neuron, 2013
Application of the HGF: Representation of precision-weighted PEs

1. Outcome PE

\[ z = -18 \]

- right VTA
- Dopamine

2. Probability PE

- left basal forebrain
- Acetycholine

Iglesias et al., *Neuron*, 2013
Application of the HGF: Social Learning

Changes in advice accuracy (black), and posterior expectation of adviser fidelity (red)

Diaconescu et al., SCAN, 2017
Representation of precision-weighted PEs in the social domain

1. Outcome PE

Diaconescu et al., *SCAN*, 2017
Representation of precision-weighted PEs in the social domain

2. Probability PE

A. first fMRI study

\[ x = -5, \ y = 12, \ z = -7 \]

B. second fMRI study

\[ x = -5, \ y = 12, \ z = -7 \]

C. conjunction

\[ x = -5, \ y = 12, \ z = -7 \]

Diaconescu et al., SCAN, 2017
Computational hierarchy & its neural signature

A.

vi. Volatility Precision

v. Volatility PE

iv. Advice Precision

iii. Outcome PE

ii. Advice PE

i. Cue-Related PE

Hierarchy

B.

Diaconescu et al., Under Revision

Diaconescu et al., Under Revision
Analogy to electrophysiological data

1. data volume — 3d

2. specify regressors — general linear model

3. parameter estimates — SPM

Diaconescu et al., Under Revision
Temporal hierarchy

Temporal hierarchy

Time (ms)

Scalp Location

left
right

anterior
posterior

i. Cue-related PE
ii. Advice PE
iii. Outcome PE
iv. Advice Precision
v. Volatility PE
vi. Volatility Precision

Diaconescu et al., Under Revision
Outline

1. Computational

2. Algorithmic

3. Implementational

4. Application to Psychiatry
Hierarchical Gaussian Filter

\[ \Delta \mu_i^{(k)} \propto \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \]

Mathys et al., *Frontiers Human Neurosci* 2011
Mathys et al., *Frontiers Human Neurosci* 2014
Lawson et al., *Nature Neuroscience*, 2017
Psychosis & Hallucinations

Powers et al., *Science*, 2017
Psychosis & Hallucinations

Powers et al., *Science*, 2017
Conclusion

**Autism**

\[ \Delta \mu_i^{(k)} \propto \frac{\hat{\pi}_i^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \]

Belief Update

3\textsuperscript{rd} level of the hierarchy

**Psychosis**

\[ \Delta \mu_i^{(k)} \propto \frac{\hat{\pi}_{i-1}^{(k)}}{\pi_i^{(k)}} \delta_{i-1}^{(k)} \]

Belief Update

2\textsuperscript{nd} level of the hierarchy
Take-Home Message

Bayes' Theorem

\[ p(x|y,m) = \frac{p(y|x,m)p(x|m)}{p(y|m)} \]

Generative models as computational assays

Perception as the inversion of a generative model

Stephan et al., *Frontiers Human Neurosci* 2016
References


How do we construct regressors that correspond to cognitive processes and use them in SPM?

1. Pass individual subject trial history into SPM:

Response \( y \) (orange = 1 advice was taken), input \( u \) (green = 1 advice was accurate)
How do we construct regressors that correspond to cognitive processes and use them in SPM?

2. Estimated subject-by-subject model parameters:
   - Model Inversion:

```
runnning model/param combination 4 of 546
Irregular trials: none
Ignored trials: none
Irregular trials: none

Optimizing...
Calculating the negative free energy...

Results:
mu2_0: 1.0665
sa2_0: 1.4966
mu3_0: 1
sa3_0: 1
ka: 0
om: -10
th: 1.0000e-18
p: [1.0665 1.4966 1 1 0 -10 1.0000e-18]
ptrans: [1.0665 0.4032 1 0 -22.3327 -10 -34.5388]
ze1: 0.8816
ze2: 48.0000
p: [0.8816 48.0000]
ptrans: [2.0073 3.8712]

Negative free energy F: -82.9603
```
How do we construct regressors that correspond to cognitive processes and use them in SPM?

3. Generate model-based time-series:

3. Convolve them with HRF:

Adapted from O’Doherty et al., 2007
How do we construct regressors that correspond to cognitive processes and use them in SPM?

5. Construct your GLM:

\[ \mu_1^k = \mu_1^{(k-1)} + \sigma \cdot \delta \]

Adapted from Behrens et al., 2010
6. **First-level analysis:**
   - Load your regressors:

   ```
   reg1 =
   [1x189 double]
   [1x189 double]
   [1x189 double]
   ```

   ```
   mu1hat <1x189 double>
   positive_PE <1x189 double>
   ```
6. First-level analysis:
   - Open SPM: Specify first level analysis
Estimate: single subject

6. First-level analysis:
   - Load Design matrix into Batch editor
Estimate: single subject

6. First-level analysis:
   - Examine results:
     - PE