Bachelor's Thesis in Artificial Intelligence

Using Bayesian Adaptive Stimulus Selection to Estimate Generalization Curves

Mihaela Neacsu

Student number: s1002844
Supervisor: Luc Selen - *Donders Institute for Brain, Cognition and Behavior*
Second reader: Max Hinne - *Donders Institute for Brain, Cognition and Behavior*

Faculty of Social Sciences
Radboud University
Netherlands
31-01-2020

Abstract

In my research, I implement the Bayesian Adaptive Stimulus Selection algorithm proposed by Kontsevich and Tyler (Kontsevich & Tyler, 1999) and adapt it in order to estimate the parameters of generalization curves of motor learning. Using this algorithm could lead to faster and more efficient computations and, as a result, more relevant findings. In order to test the algorithm's performance, I run it against an algorithm which selects stimuli randomly. Eyeballing the resulting plots shows a considerable difference in the performance, although for 2 out of 3 of the curve parameters, the results are not statistically significant. Future research could further build on this model to improve its performance, or use it in a model comparison study between symmetric and asymmetric models of a generalization curve.

Contents

1 Introduction

1.1 Background

The human brain continuously adapts its movement strategies and motor internal models, as it is trying to make sense of both an ever changing body and environment. One tool used in this regard is generalization, an ability of great importance when it comes to learning. Generalization is the process of predicting and adapting to new situations or scenarios, based on knowledge gathered in a similar setting (Poggio and Bizzi, 2004). As stated in a paper from 2006 (Krakauer et al., 2006), generalization is considered by some "the distinguishing feature of biological learning [...] because our survival may depend on our ability to correctly extrapolate to contexts that are different from our limited experience" (p.1). However, even though the existence of such a mechanism has been well established, the answer to how exactly this process happens in the brain remains a topic of heavy debate and discussion in the scientific community, and it is yet to be understood (Castro et al., 2011).

In motor control research, one aim is to gather relevant findings on generalization. One example of a possible experiment is testing generalization using a robotic arm. Here the question of interest is, 'how does something that I have learned for one specific direction generalize to directions that have not been experienced yet?'. The participant tries to make a forward movement with their arm, which is inside a robotic manipulandum. However, its movement is distorted by a force field produced by the robotic arm, which pushes the subject's arm to the side. After a certain amount of trials, the participant adapts to the force field and learns how to compensate for it, by pushing back with an equal amount of force. The main idea of the experiment is that the subject trains to adapt to one direction and then later different directions are probed to see how much one generalizes what they have learned. However, this process is significantly time consuming for both researchers and participants. Therefore, we would like to speed it up, in order to get a feeling for how generalization happens in as few trials as possible.

1.2 Bayesian Adaptive Stimulus Selection

One efficient way to do so is by using Bayesian Adaptive Stimulus Selection. Adaptive Stimulus Selection picks stimuli (here, the stimuli are the angles of the directions we are probing) that are maximally informative given the data we have so far, in order to reduce the number of trials necessary (Bak and Pillow, 2018). A Bayesian approach was chosen, as probabilistic models are becoming more and more prevalent in research which aims to understand cognition (Chater et al., 2006). There are more ways to implement Bayesian Adaptive Stimulus Selection, but the method presented in this paper is based on the algorithm proposed by Kontsevich and Tyler (K&T), who try to approximate the parameters of a psychometric function, being their generative model (Kontsevich and Tyler, 1999). After firstly having implemented their algorithm, I adapt it in order to work on generalization curves in motor tasks and then run it against an algorithm that picks stimuli in a random manner. I hope to see that the adaptive algorithm performs more efficiently than the random one, more specifically needing less trials in order to converge towards relevant results. The aim of this project is to attest the efficiency of the Bayesian Adaptive Stimulus Selection algorithm, which in the future could lead to more complex experiments being performed and more relevant results being found. Therefore, this project could be considered a stepping stone towards more advanced findings in Neuroscience, which were limited beforehand by technological shortcomings.

1.3 Research Question

"Can we use Bayesian Adaptive Stimulus Selection in order to estimate generalization curves in a reliable and efficient manner?"

2 Methods

2.1 Kontsevich and Tyler (1999)

Kontsevich and Tyler (Kontsevich and Tyler, 1999) introduce a Bayesian adaptive algorithm that approximates the parameters of their model (threshold and slope) such that the psychometric function best fits the given data. The algorithm takes a list of possible parameter sets, and returns a posterior probability distribution over all of them. In order to do so, they use entropy as a utility function and try to pick the stimulus that minimizes it, in order to maximize information gain. Based on that stimulus they obtain a response for which they compute the new probability distribution. The new probability distribution becomes the prior for the next iteration. The algorithm functions under the assumption that the right model is included in the sample space of parameter sets.

Before implementing their algorithm, certain steps had to be taken. Firstly, the stimulus array had to be defined. I chose 200 evenly distributed values in the range [-5, 5]. The reason for choosing such a small interval is that, because we are dealing with psychometric functions, a lot of information is contained in the very small interval around the threshold. Also, choosing too big an interval would significantly affect model running time whilst not producing more relevant results. Secondly, the generative model had to be created. Here, the generative model is the parameter set that the algorithm needs to approximate back. Furthermore, all the parameter sets had to be generated. For the means I chose 61 evenly distributed numbers in the range [-3, 3], while for the standard deviation I chose 50 evenly distributed values in the range [0.01, 5]. The parameters sets were created by combining each mean value with each standard deviation value, thus I had 3050 possible parameter sets.

The next step was creating a lookup table, containing the conditional probabilities $p(r|\lambda, x)$, where r is the response, λ is the parameter set and x is the stimulus. As Kontsevich and Tyler use the two-alternative forced-choice method, the response is either a success or a failure. Therefore, in the lookup table, I only stored the probabilities of success, as the probabilities of failures can be derived from those (1 - probability of success). In order to get the success probabilities, I used the norm.cdf() function from the SciPy/NumPy library in Python, which fits a psychometric curve with a given mean and standard deviation to the stimuli array and returns the resulting probabilities, which I stored in a 2D lookup table. Finally, the last step was initializing a uniform prior probability over the parameter sets such that there is no bias in the starting off point for the algorithm.

As stated before, the algorithm tries to maximize expected gain, by minimizing expected entropy. Firstly, entropy is calculated for each and every stimulus and the one with minimal entropy is chosen. Secondly, the stimulus is fed to the generative model, in order to receive a success (“1”) or failure (“0”) response. Lastly, for every parameter set λ, the probability of it given that specific response r and stimulus x is calculated by using the Bayes formula:

$$P_t(\lambda|x,r) = \frac{P_t(\lambda)P(r|\lambda,x)}{\sum_\lambda P_t(\lambda)P(r|\lambda,x)} \tag{1}$$

Here, $p(\lambda)$ is the prior probability of each parameter set (which is uniform in the beginning) and $p(r|\lambda,x)$ is the conditional probability from the lookup table. The posterior probability given by one iteration becomes the prior probability for the next one. When the termination criterion is reached, which Kontsevich and Tyler defined as a particular number of trials, the algorithm stops and returns both a probability distribution over all the parameter sets, as well as an expected value for the parameters of the generative model, by using the following formula:

$$\lambda_{t+1} = \sum_\lambda \lambda P_{t+1}(\lambda) \tag{2}$$

2.2 My Implementation of Kontsevich & Tyler

I implemented the Kontsevich and Tyler Bayesian adaptive algorithm in Python 3, as the multiple built-in functions and matrix manipulation libraries made it for an easy implementation. I used the NumPy/SciPy libraries for many of its mathematical tools, Pandas for its usage of DataFrames for easy handling of the data, and MatPlotLib for the plotting of results. I followed the steps of the algorithm meticulously up until the last step, calculating the expected parameters of the generative model, which I have chosen to leave out, in order to focus solely on the posterior probability distribution that the algorithm returned. My termination criterion was iterating through 100 trials.

2.3 Adapting Kontsevich & Tyler to Generalization Curves

In order to go one step forward, I chose to apply the Bayesian adaptive method to Gaussian-shaped generalization curves. In order to achieve this, the algorithm had to be altered significantly: the models are no longer cumulative distributions but Gaussian-shaped curves, and the possible outcomes are not only success or failure but different values for a variable called the Adaptation Index (AI). Furthermore, the utility function is different, switching from entropy to mutual information. A list of the changes made is shown in the table below:

Kontsevich & Tyler (1999)	**Adaptation**
2AFC Task	Motor Task
Psychometric Function	Gaussian curve
Utility function: Entropy	Utility function: Mutual Information
Parameters: - Slope - Threshold	Parameters: - Mean - Standard Deviation - Amplitude

2.4 The Adaptation Index

When one moves their arm from point A to point B, normally one does that in a straight line. For the motor experiment described in the Introduction, however, a force field is imposed by a robotic manipulandum that pushes a subject's arm to the right. Therefore, instead of going straight, their hand will be driven to the right. When the participant learns to adapt against the force field, they start to compensate with a force in the opposite direction. If the subject perfectly compensates for the force field, then they use forces that are the complete opposite of the ones applied to them. If the force is perfectly matched, then it gets an adaptation index (AI) of 1, but if it is not matched at all, the AI is 0. If one does only half the compensation, then the AI is 0.5.

In order to calculate the AI and to make sure the researchers have an independent measure of what is happening, on some trials not known to the subject they build two virtual walls using the robotic arm. These walls ensure movement in a straight trajectory, while the subject expects a force to the right, so they push to the left. That means they will produce a force on the left virtual wall, which can be measured. This is measured against the force the force field would have produced, which is a function of the velocity of the movement of the participant (the higher the speed, the bigger the force). Therefore, an analysis can be done on the two values and the AI can be calculated.

Despite the fact that the values of adaptation indices are in the interval [0, 1], because there is noise, the range I am looking at is slightly bigger than that. I start with the value of -0.5, which could happen both because of noise or because the participant compensates in the opposite direction. Furthermore, if the subject perfectly compensates the AI is 1, but in some trials participants can overcompensate a little, hence going up to an AI of 1.5. Also, I will model the AIs as discrete values, such that the probability of each can be calculated, even though in reality it is a continuum.

2.5 Mutual Information

As a utility function I followed Pillow's (Pillow, 2016) suggestion of utilising mutual information (MI), as he states it is a beneficial method to choose stimuli that are maximally informative in Bayesian adaptive methods. MI shows the dependency between two variables, more specifically how much information one observed random variable conveys about another random variable. In our algorithm we are looking to maximize MI, such that we have as much information as possible from the data gathered so far. MI can be calculated using the following formula:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{3}$$

Here, $H(X)$ and $H(Y)$ are the entropies for each variable, and $H(X \mid Y)$ and $H(Y \mid X)$ are the conditional entropy.

2.6 Generalization curves

To model the process of generalization to new conditions or a new environment, it is often assumed that the data falls in a Gaussian-like pattern (Sarwary et al., 2015). Therefore, I used Gaussian curves in the second part of the experiment to model the generalization process. In order to define them, I took into account 3 parameters: the mean (the center of the curve), the width of the curve and the amplitude (height of the curve). The mathematical formula used to define generalization curves can be found below:

$$Gaussian(x) = A * e^{-(x-mean)^2/(2*width^2)} \tag{4}$$

Here, x is the stimuli array and A is the amplitude of the curve.

In contrast to K&T, where the psychometric function encompasses the probabilities for a success or a failure, a Gaussian curve does not show the probabilities of a certain value, but the mean AI for every stimulus angle which in turn has a Gaussian distribution around it. In order to determine the AI we obtain when given a stimulus, one does not take the mean value, but samples from the Gaussian distribution around it with the mean mAI and the standard deviation considered known and set at $\sigma = 0.1$. The graphical representation is depicted in Figure 1:

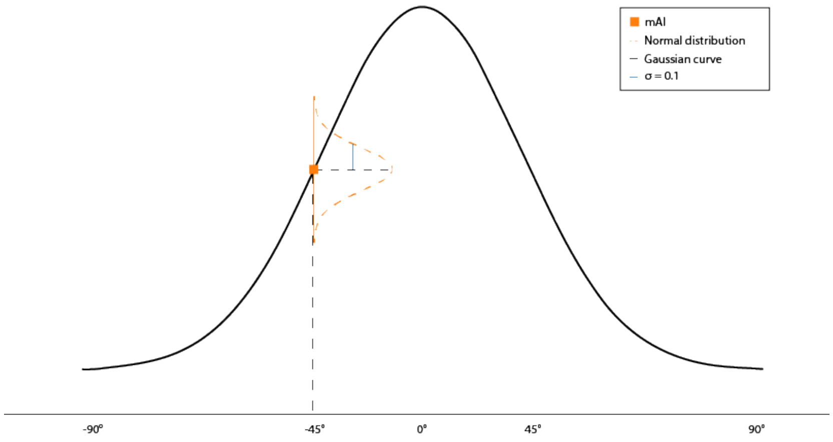

Figure 1: *Visual representation of a generalization curve: the angles are the stimuli which correspond not to a probability, but to a mean value mAI of the Adaptation Indeces. In order to obtain the value for the AI, one needs to sample from a distribution with the mean mAI and the standard deviation set to 0.1.*

Therefore, in essence, there are an infinite number of possible outcomes, thus the response is replaced with the probability of an AI given a stimulus. As mentioned before, the AIs are theoretically a continuous variable, but we represent them as discrete values.

2.7 Implementation of the Adaptation

For the stimulus array, I chose 51 evenly distributed values in the range [-120, 120], which represent the different angles generalization is probed for. The generative model was a Gaussian curve with a mean of -20, a width of 27.5 and an amplitude of 0.75. These are the parameters that the algorithm had to approximate. Also, in order to define the parameter sets, I chose 38 values for the mean (range [-90, 90]), 15 for the width (range [5, 45]) and 11 for the amplitudes (range [0.5, 1.5]) and defined all the possible combinations of these values. I thus ended up with 6270 parameters sets from which the algorithm needs to estimate back the correct one.

The changes mentioned above lead to the lookup table becoming more complex: we switch from a 2D table to a 3D table in order to accommodate for the different discretized values of AI. To generate the lookup table I used the norm.pdf() function from the SciPy/NumPy Python library, which fits a Gaussian curve with a given mean (which is the average AI value given a certain parameter set and a certain stimulus) and standard deviation (which is set to 0.1) to the array containing the AIs and returns the resulting probabilities.

Following the K&T method, the algorithm starts with a uniform prior over all parameter sets. Similarly, it uses mutual information as the utility function in order to choose the most informative stimulus for every trial. Therefore, it picks that stimulus that has the highest mutual information and then feeds it to the generative model, which gives back a response. Finally, in order to update the probabilities, we apply Bayes formula. Based on the updated probability distribution, the algorithm can estimate the 3 parameters of the generalization curve.

3 Analysis & Results

In order to test the adaptive algorithm's performance in estimating generalization curves, I ran it against an algorithm that samples the stimuli in a random manner. This way, I could analyse whether the adaptive algorithm is indeed more efficient, and more specifically if it needs fewer iterations to obtain relevant results.

Firstly, I wanted the see the performance of the algorithm when choosing the most informative stimulus, thus I visualized the performance of both algorithms in picking stimuli over 100 trials. The results are depicted in Figure 2:

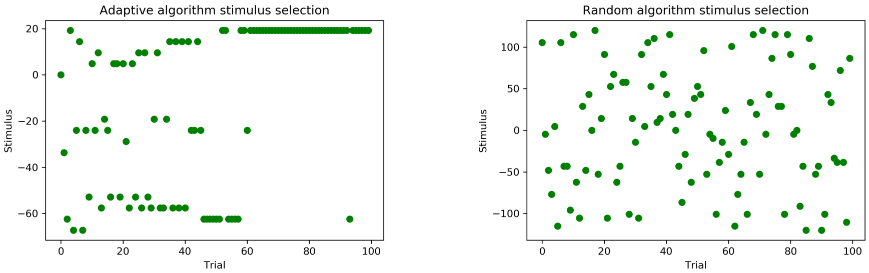

Figure 2: *Stimulus choice for adaptive (left) and random (right) sampling over 100 trials*

In the left panel it can be observed that the adaptive algorithm has a more structured approached and needs less trials in order to find the stimuli that are maximally informative, as well as having a smaller searching pool. In comparison, the random algorithm shown in the right panel performs randomly for all the 100 trials.

Secondly, I wanted to test the performance of the 2 algorithms in estimating the values of the parameters. The error measure I used in my analysis was the squared difference between the real value of the parameter and the value estimated by the algorithm (adaptive or random). During the testing phase, I did 10 runs for both algorithms, each one consisting of 100 trials. I thus ended up with 2000 data points, 1000 for the adaptive algorithm, and 1000 for the random one. Afterwards, for each trial, I used the mean squared error to calculate the mean over all the runs. The mean squared error has the following formula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{5}$$

I repeated the process for both the adaptive and random algorithm and for each parameter of the Gaussian curve: the mean, the standard deviation and the amplitude and plotted the results. I also plotted the standard deviation of the error points for every trial, which is the difference between the most extreme values. This shows the consistency over runs for both algorithms. In total, I ended up with 6 different graphs that show the disparities between the performances of the adaptive algorithm and the random one. The results are depicted in Figure 3:

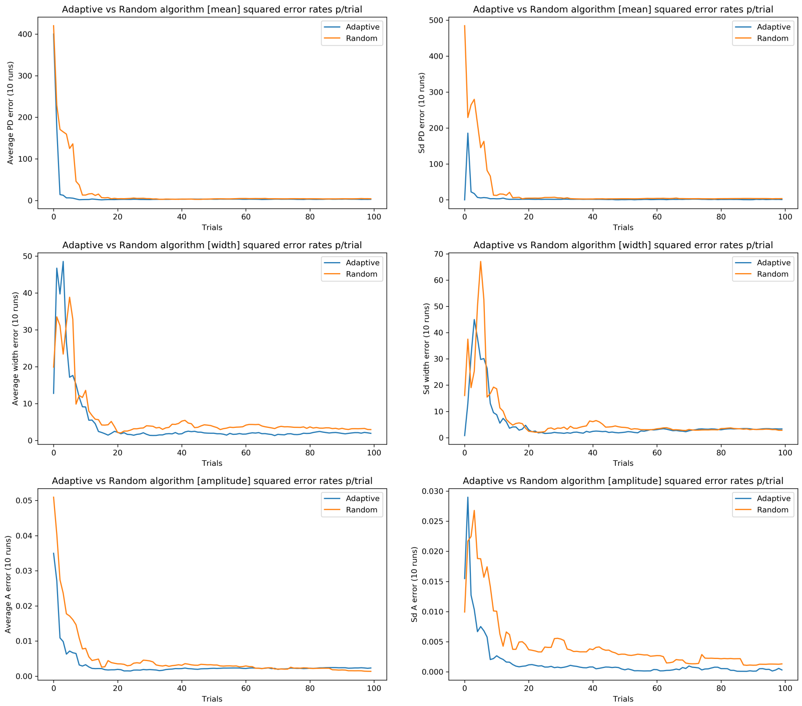

Figure 3: *Performance of the Adaptive algorithm (blue) vs the Random algorithm (orange) for each of the 3 parameters (mean, width, amplitude). On the left, for each algorithm, for each trial, the average error (calculated using MSE) over runs is plotted. On the right, for each algorithm, for each trial, the average standard deviation over runs is plotted.*

It can be observed that, when choosing stimuli adaptively, it takes fewer trials for the curve to come down, which confirms the idea that the adaptive algorithm needs less time in order to reach relevant results. Furthermore, the adaptive algorithm generally shows lower deviations in error rates than the random one. This means that there is less divergence in between runs, so the adaptive algorithm does show more consistency in its performance.

3.1 Significance Testing

In order to test for significance, a couple of steps had to be taken. To begin with, I defined an exponential function to be fitted on error data as: a*exp(-b*x)+c. Next, I used the curve_fit() function from the SciPy optimize library per run to fit the defined exponential function to the squared error data from the random and adaptive algorithm's performance on predicting the 3 Gaussian curve parameters. Therefore, for each parameter, 20 exponential curves were obtained. The results are depicted in Figure 4:

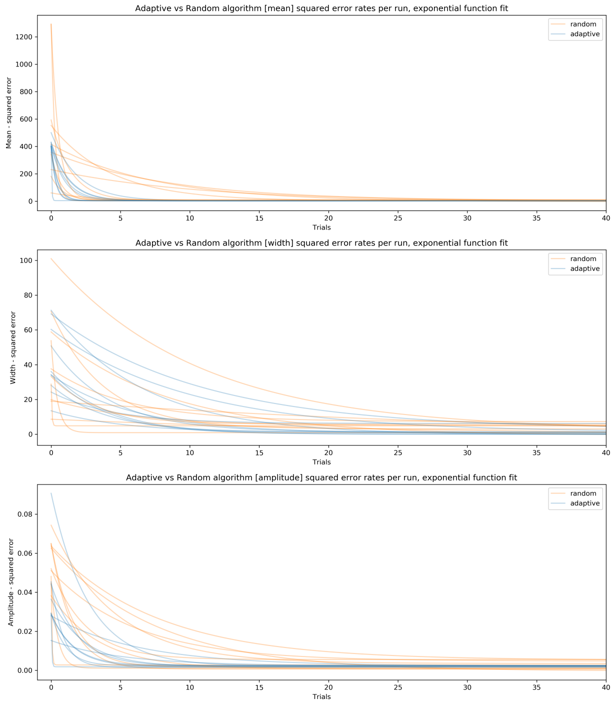

Figure 4: *Significance Testing for the difference in the performance of the Adaptive algorithm (blue) vs the Random algorithm (orange) for each of the 3 pa-*

rameters (mean, width, amplitude). For every algorithm, for every run, an exponential curve was fitted to the squared error data and the slope of the curve was used in a student t-test.

From these curves, the slopes b of both the random and adaptive algorithm were tested for significance, using a student t-test. The slopes had to be transformed using a log transform, in order to be normally distributed for the t-test. Out of all the 3 cases, only the values for the mean showed a significant difference (p = 0.284). The width (p = 0.5753) and the amplitude (p = 0.6555) were not significant.

4 Discussion

We have seen that the adaptive algorithm outperforms the random one in estimating the parameters of generalization curves, as it needs fewer trials to reach relevant results. This could have a positive impact on future research, as limiting experiment duration can lead to faster and more advanced findings in the future. Therefore, this experiment could be considered a step towards answering the question of how the human brain performs generalization (Yokoi et al., 2011).

The current study presents a couple of limitations, however. Firstly, there was no statistical significance shown in the t-test done on the data for 2 out of the 3 parameters. This could be because 10 runs are not enough in order to gather enough data for the significance testing, thus more data could be gathered in hope to see more significant results. Despite the lack of statistical significance, eyeballing already reveals a positive trend in the data from the adaptive algorithm compared to the random one, thus better methods should be devised to truly test for significance as the curve's slope is not always indicative of better performance.

Secondly, my research functions under the assumption that generalization curves are Gaussian-shaped and symmetrical: the left side of the curve has the same width as the right side. However, this might not be the case. At the moment, there are debates in the field of psychophysics regarding the way motor learning takes place. The majority agrees that the process is plan based: actions are relying on current knowledge (Todorov and Jordan, 2002) and, therefore, use exploitation of this knowledge to maximize the performance (Castro et al., 2011). However, Gonzales-Castro presents evidence in his studies supporting a theory that motor learning is motion based: motor variability supports the learning process and therefore, is based on exploration mechanisms (Castro et al., 2011). This disparity can also lead to different models of motor generalization. On one hand, the model conventionally used shows the generalization as a symmetrical Gaussian-like pattern (Sarwary et al., 2015), with the same width for both the left and the right half. On the other hand, Gonzales-Castro models it as two shifted Gaussians, with different widths (Castro et al., 2011). Furthermore, Selen also shows evidence of skewed generalization in his research (Selen, 2013).

Future research could take the next step and further adapt the algorithm to perform model comparison. Frequentist Statistics is limitative, as model comparison is not transitive: if one model has a lower p-value than another model for the same null hypothesis, it does not mean that the former model is better than the latter (Wagenmakers, 2007). However, in Bayesian Statistics, we can use the Bayes Factor to infer which model predicts the data best. This is why the Bayesian Adaptive Stimulus Selection algorithm proposed by Cooke (Cooke et al., 2018) could be used, in order to dissociate between a symmetric and asymmetric generalization model of force expression in motor learning. I believe my code is stable enough to actually expand it in that direction and implement Cooke's algorithm, in order to see which model best fits the generalization data while performing the analysis in a fast, efficient manner.

References

Bak, J. H. & Pillow, J. W. (2018). Adaptive stimulus selection for multi-alternative psychometric functions with lapses. *Journal of Vision*, *18*(12), 1–25. doi:10.1167/18.12.4

Castro, G., Nicolas, L., Monsen, C. B., & Smith, M. A. (2011). The Binding of Learning to Action in Motor Adaptation. *7*(6). doi:10.1371/journal.pcbi.1002052

Chater, N., Tenenbaum, J. B., & Yuille, A. (2006). Probabilistic models of cognition : Conceptual foundations. *10*(7).

Cooke, J. R., Selen, L. P., van Beers, R. J., & Medendorp, W. P. (2018). Bayesian adaptive stimulus selection for dissociating models of psychophysical data. *Journal of Vision*, *18*(8), 1–20. doi:10.1167/18.8.12

Kontsevich, L. L. & Tyler, C. W. (1999). Bayesian adaptive estimation of psychometric slope and threshold. *39*(August 1998), 2729–2737.

Krakauer, J. W., Mazzoni, P., Ghazizadeh, A., Ravindran, R., & Shadmehr, R. (2006). Generalization of motor learning depends on the history of prior action. *PLoS Biology*, *4*(10), 1798–1808. doi:10.1371/journal.pbio.0040316

Poggio, T. & Bizzi, E. (2004). Generalization in visual and motor control. *431*(October).

Sarwary, A. M., Stegeman, D. F., Selen, L. P., & Medendorp, W. P. (2015). Generalization and transfer of contextual cues in motor learning. *Journal of Neurophysiology*, *114*(3), 1565–1576. doi:10.1152/jn.00217.2015

Selen, L. (2013). *Generalization of context-dependent internal models*. Presentation at the Neuroscience conference of the SfN.

Todorov, E. & Jordan, M. I. (2002). Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, *5*(11), 1226–1235. doi:10.1038/nn963

Wagenmakers, E. J. (2007). A practical solution to the pervasive problems of p values. *Psychonomic Bulletin and Review*, *14*(5), 779–804. doi:10.3758/BF03194105

Yokoi, A., Hirashima, M., & Nozaki, D. (2011). Gain field encoding of the kinematics of both arms in the internal model enables flexible bimanual action. *Journal of Neuroscience*, *31*(47), 17058–17068. doi:10.1523/JNEUROSCI.2982-11.2011