MemToolbox Tutorial

Suchow, J. W., Brady, T. F., Fougnie, D., & Alvarez, G. A. (2013). Modeling visual working memory with the MemToolbox. *Journal of Vision, 13*(10):9, 1–8. doi:10.1167/13.10.9.

Qingqing Yang, qy775@nyu.edu

Department of Psychology

NYU

Visual WM

- Working memory is a storage system that actively holds information in mind and allows for its manipulation (Baddeley, 1986).
- Paradigms predominate visual WM study:

The observer sees the stimulus display, and after a delay is asked to **report the color of a single item or subset**

The observer sees the stimulus display, and after a delay is asked to report **whether the test display matches**

NYU

Suchow, J. W., Brady, T. F., Fougnie, D., & Alvarez, G. A. (2013). Modeling visual working memory with the MemToolbox. Journal of Vision, 13(10):9, 1–8, http://www.journalofvision.org/content/13/10/9, doi:10.1167/13.10.9.

Models proposed

- Models have been proposed that link performance in 2 tasks to the architecture and capacity of the WM system.
- Each model specifies the structure of visual memory and the decision process used to perform the task.
  - Item-limit model (Pashler, 1988)
  - slot model (Luck & Vogel, 1997; Cowan, 2001)
  - slots + averaging model (Figure 2; Zhang & Luck, 2008)
  - slots + resources model (Awh, Barton, & Vogel, 2007)
  - Continuous resource model (Wilken & Ma, 2004)
  - resources + swaps model (Bays, Catalao, & Husain, 2009)
  - Ensemble statistics + items model (Brady & Alvarez, 2011)
  - Variable precision model (Fougnie, Suchow, & Alvarez, 2012; van den Berg, Shin, Chou, George, & Ma, 2012;).

Intro of Memtoolbox

- A collection of MATLAB functions for modeling visual working memory. [MemToolbox (visionlab.github.io)]
- the toolbox includes:
  - Implementations of **popular models** of visual working memory
  - Real and simulated data sets
  - Bayesian and maximum likelihood estimation procedures for fitting models to data
  - Model comparison metrics
  - experiment scripts

GeneralHelpers	12/27/2021 1:48 AM	File folder
MemData	12/27/2021 1:48 AM	File folder
MemExperiments	12/27/2021 1:48 AM	File folder
MemFitting	12/27/2021 1:48 AM	File folder
MemModelComparison	12/27/2021 1:48 AM	File folder
MemModels	12/27/2021 1:48 AM	File folder
MemPlots	12/27/2021 1:48 AM	File folder
MemTests	12/27/2021 1:48 AM	File folder
MemTutorial	12/27/2021 1:48 AM	File folder
MemUtilities	12/27/2021 1:48 AM	File folder
Contents.m	12/27/2021 2:15 AM	MATLAB Code
LICENSE	12/27/2021 1:48 AM	File

NYU

Models in Memtoolbox

- >> help MemModels
- To check any model m, >>help m; to view the code, >> edit m;

```
% One component models:
%   AllGuessingModel         - only guessing
%   NoGuessingModel          - just precision, no guessing
%
% Mixture models for a single set size:
%   StandardMixtureModel     - guess rate and precision
%   SwapModel                - guess rate, precision and swaps to other items.
%   VariablePrecisionModel   - guess rate and variable precision
%   EnsembleIntegrationModel - integration with distractors shifts reports
%
% Models parameterized based on set size:
%   SlotModel                - capacity and precision (no benefit when cap.>setsize)
%   SlotsPlusResourcesModel - capacity and precision (more juice when cap.>setsize)
%   SlotsPlusAveragingModel - capacity and precision (more slots/item when cap.>setsize)
%   ContinuousResourceModel - capacity juice split among all items equally
%
% Models that depend on delay duration:
%   ExponentialDecayModel    - capacity K and sd, plus objects drop out over time
%
% Model wrappers:
%   WithBias                 - adds a bias terms (mu) to any model
%   FixParameterValue        - fix any parameter in a model to a specific value
%   Orientation              - converts a model to use a 180 degree space, for objects rotationally symmetric
%   TwoAFC                   - converts a model so that can be fit to 2afc data
%   WithLapses               - adds inattentional parameter to any model
```

Set up

- Download from [MemToolbox (visionlab.github.io)]
- Add the files to Matlab
  - Manually
  - Run setup.m

Modeling in standard workflow

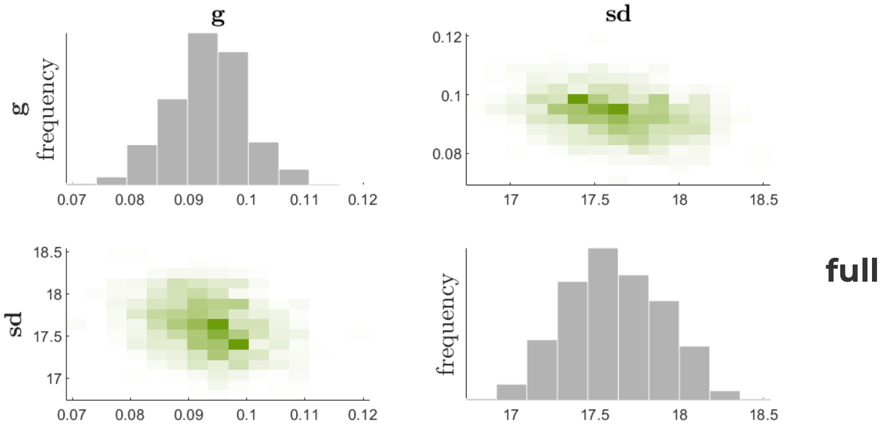
- picks a model
- fit the model to data using probabilistic methods
  - Given a model M with free parameters θ, the model's likelihood function specifies a **probability distribution P(θ|D) (integral = 1)** over likelihood P(D|θ).
- a likelihood function is defined, describes the model's predictions for each possible setting of its parameters.
  - an **estimator** is used to **pick the parameter settings that provide the best fit to the data.** maximum likelihood estimator or maximum a posterior estimate is often used.

Modeling in standard workflow

- 2 structs to organize the info:
  - the model and its likelihood function
  - the data to be fit
- Fitting is simply call a build in function:
  - MemFit(), this will return the MAP estimate of parameters
  - MLE(), this will return the MLE parameters

```
>>model = StandardMixtureModel();
>>data.errors = [-89, 29, -2, 6, -16, 65, 43, -12, 10, 0, 178, -42, 52, 1, -2];
>>fit = MLE(data, model) Or  MemFit(data.errors)
```

Bayesian Workflow



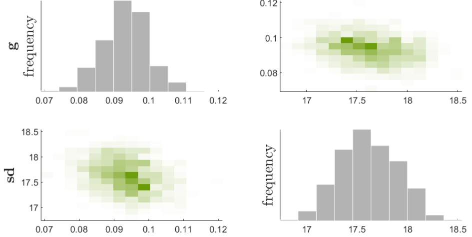
- Instead of just returning the MLE/MAP estimate **probability distribution over parameter value** setting after considering the observed data as p
  - >> fit = MemFit(data, model)
  - Or >> get posterior sample from MCMC(data, model);

- The prior, P(θ) conveys which parameter values are reasonable.
- Once specified the P(θ), beliefs are updated based on the experimental data. After observing data D, the posterior beliefs about parameters ("the posterior") are given by

$$\mathrm{P}(\theta|D) \propto \mathrm{P}(D|\theta) \cdot P(\theta),$$

NYU

Bayesian Workflow

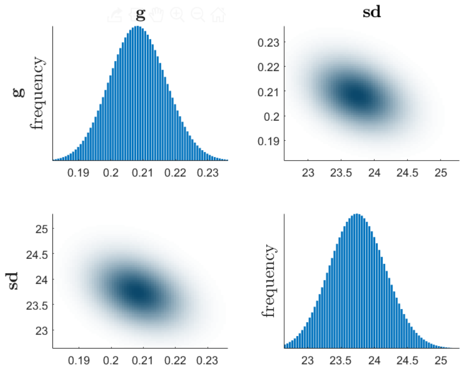


- To estimate the full posterior distribution, **sampling-based algorithms** are used.
- The Metropolis-Hastings variant of Markov Chain Monte Carlo **(MCMC):**

The algorithm chooses an initial set, over many iterations, proposes small moves to these parameter values, accepting or rejectiof model parametersng them based on how possible the new parameter values are in both the prior and the likelihood function.

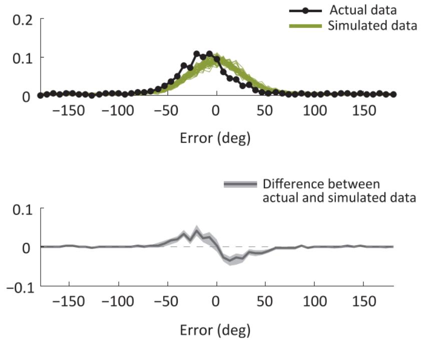
$$\mathrm{P}(\theta|D) \propto \mathrm{P}(D|\theta) \cdot P(\theta),$$

- **GridSearch** function: evaluating the model's likeliho ›le setting of the parameters
- The full posterior distribution also helps determine tł parameters.



Posterior predictive checks

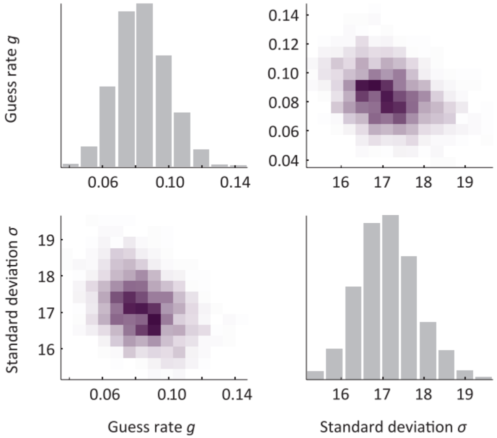
- Sometimes a whole class of models performs poorly, such that there are no parameter settings that will produce a good fit.
- Simulates new data from the posterior fit and then compares the actual and simulated data
    - A good fit does not necessarily indicate a good model. An extremely flexible model
    - models that systematically deviate in a posterior predictive check nearly always need improvement!




```
>>PlotPosteriorPredictiveData(model, posteriorSamples,...
                              data, [optionalParameters])
```

Hierarchical modeling for population

- Traditional method for a population:
  - Fit each participant separately;
  - Combine parameter by median or mean;
  - Difference between condition – t-test, ANOVAs
- Discards information about the reliability of each participant's parameter estimates.
  - Trade-off between parameters



Hierarchical modeling

- Better method: Fit **a single hierarchical model of all participants**
- This treats each participant's parameters as samples from a normally distributed population, and then infer the population mean and sd of each parameter.
    - More weight to participants whose data give more reliable parameter estimates

```
>> data1 = MemDataset(1);
>> data2 = MemDataset(2);
>> model = StandardMixtureModel();
>> fit = MemFit({data1,data2}, model,'UseHierarchical', true)
```

NYU

Model comparison

- Two elements to be considered:
  - the resemblance between the model and the data
  - the model's flexibility
- Approaches penalize more flexible models:
  - Akaike Information Criterion (**AIC**)) (with correction for finite data (**AICc**))
  - Bayesian Information Criterion (**BIC**)
  - Deviance Information Criterion (**DIC**) *for hierarchical setting*
  - Bayes factor
  - Log likelihood

```
>> model1 = StandardMixtureModel();
>> model2 = SwapModel();
>> modelComparison = MemFit(data, {model1, model2})
```

NYU

Model comparison

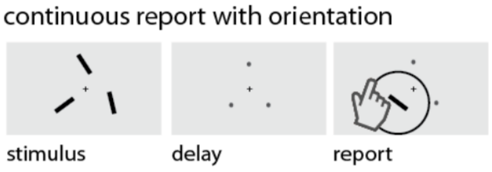
- Computing the comparison metric for each participant independently and looking at consistency to make inferences about the best-fitting models

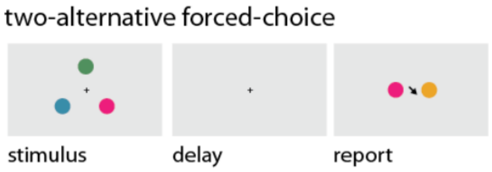
- Take participants' variance into account, therefore generalized to population, (like ANOVA over model likelihood)

- However, computing a single AICC or BIC value across all participants does not allow generalization to the population

Advanced Modeling

- Modeling with different type of data
  - Orientation data
  - 2AFC





- Modeling with different Set sizes
- Comparing parameters between conditions
- Fit model with 1 data but examine the model with another
- Create ur own new model

Thank you!