Instructions for MATLAB sequential experimental design for functional response code

This document outlines instructions to use the MATLAB code provided for sequential experimental design for functional response experiments.

Description of MATLAB code

The main MATLAB script that is used to run the sequential experimental design (and the best starting point) is called “SMC.m”. The entire sequential experimental design algorithm can be run from this script and it calls the other scripts and functions. In the first section of “SMC.m”, labelled “Problem and SMC Set-Up”, input regarding the SMC algorithm and the functional response problem is required. Some of the inputs include specifying the method for selecting the next design point in the experimental design, which of the 8 possible models are to be the candidate models and other information such as the number of particles, ESS threshold and the number of observations that are to be collected. For methods that require random generation of observations you also have to select “true values” of the parameters to generate the data. In addition, you define the hyperparameters of the log normal priors of the parameters.

The second section of the script, labelled “Conducting the SMC”, incorporates all the code required to run the SMC algorithm. No input is required in this section. The following table outlines the different scripts and a description of each.

SCRIPT NAME	DESCRIPTION
SMC.m	The main script to run the sequential experimental design process. Requires input and contains calls to other scripts and functions.
move_step.m	This script contains code to construct a Markov Chain Monte Carlo (MCMC) proposal distribution and to conduct a MCMC move step on the particles after the resampling step. We compute the sample covariance matrix and use it in a multivariate normal random walk proposal. It is called in the “SMC.m” script.
generate_data.m	This script uses a selected design point to generate a random observation from either the beta-binomial distribution or the binomial distribution depending on the model of interest. It is called in the “SMC.m” script.
prior_sampling.m	Samples N particles from the prior distribution of each model specified. In this case the prior distributions of the parameters are log prior. It is called in the “SMC.m” script.
plotSMC.m	Plots the posterior for all estimated parameters after SMC is run. It is called at the conclusion of the “SMC.m” script.
parameter_estimation_utility.m	This script determines the design point from the discrete design space which maximises the parameter estimation utility. It is called in the “SMC.m” script.
model_discrimination_utility.m	This script determines the design point from the discrete design space which maximises the model discrimination utility. It is called in the “SMC.m” script.
total_entropy_utility.m	This script determines the design point from the discrete design space which maximises the total entropy (dual purpose) utility. It is called in the “SMC.m” script.
plot_data.m	This script plots an example dataset. The sample means for each value N are also shown.

The following table outlines the different functions and a description of each.

FUNCTION NAME	DESCRIPTION
find_parameters.m	For each particle in a given particle set and an initial condition, this function determines the parameter for the expected proportion of prey eaten (mu) and over-dispersion parameter (lambda) for a specified model. It is called in the “parameter_estimation_utility.m”, “model_discrimination_utility.m”, and total_entropy_utility.m” scripts.
log_betabinpdf.m	Performs a stable calculation of the logarithm of the beta--binomial pdf. It is used in the “log_likelihood.m” and “log_lik.m” functions.
log_binpdf.m	Performs a stable calculation of the logarithm of the binomial pdf. It is used in the “log_likelihood.m” and “log_lik.m” functions.
log_lik.m	Given the expected proportion of prey eaten (mu) and over-dispersion parameter (lambda) for each particle, this function determines the log likelihood of observing data. It is called in the “parameter_estimation_utility.m”, “model_discrimination_utility.m”, and total_entropy_utility.m” scripts.
log_likelihood.m	This function determines the log likelihood of observing the first t data points for each particle in the particle set. It is called in the “SMC.m” and “move_step.m” scripts.
log_prior.m	Determines the log prior probability of each particle in the particle set. It is used in the “SMC.m” and “move_step.m” scripts.
logsumexp.m	Performs a stable calculation of log(sum(exp(x))).
multinomial_resampling	This function conducts multinomial resampling for when ESS becomes too low. It is used in the “residual_resampling.m” function.
ode_M2.m	Equation for Holling's Type II model. Used in the “ode_solver.m” function.
ode_M3.m	Equation for Holling's Type III model. Used in the “ode_solver.m” function.
ode_solver.m	This function solves the ode for a range of parameters and initial conditions for a given model and time.
residual_resampling.m	This function conducts residual resampling for when ESS becomes too low. It is used in the “SMC.m” script.

The dataset “papanikolau_data.mat” is also included in the file. This dataset incorporates data from an example experiment. The first column represents the initial prey population and the second column represents the number of prey eaten in a given time period. The dataset “hassell_data.mat” is also included as an alternative.