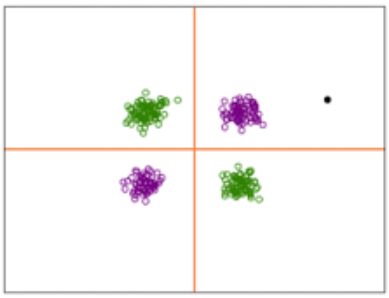
Induced Bias Behavioral Test

(The test can be accessed from here)

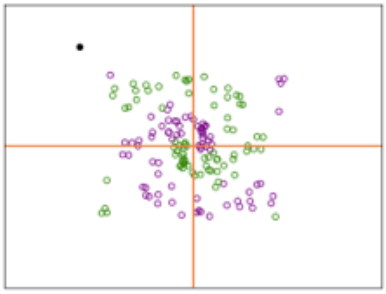
Table of Contents

Purpose

The purpose of this experiment is to assess human performance on logical inference to interpolate/extrapolate certain patterns. In this experiment, two different patterns are shown to the participants and their confidence rate on whether a test point belongs to one of two classes (purple/green) that make up these patterns are collected. For this experiment, the participants are instructed to rate purple as 1 and green as 0. Following figures are the two datasets used in this experiment.



Gaussian XOR with small variance

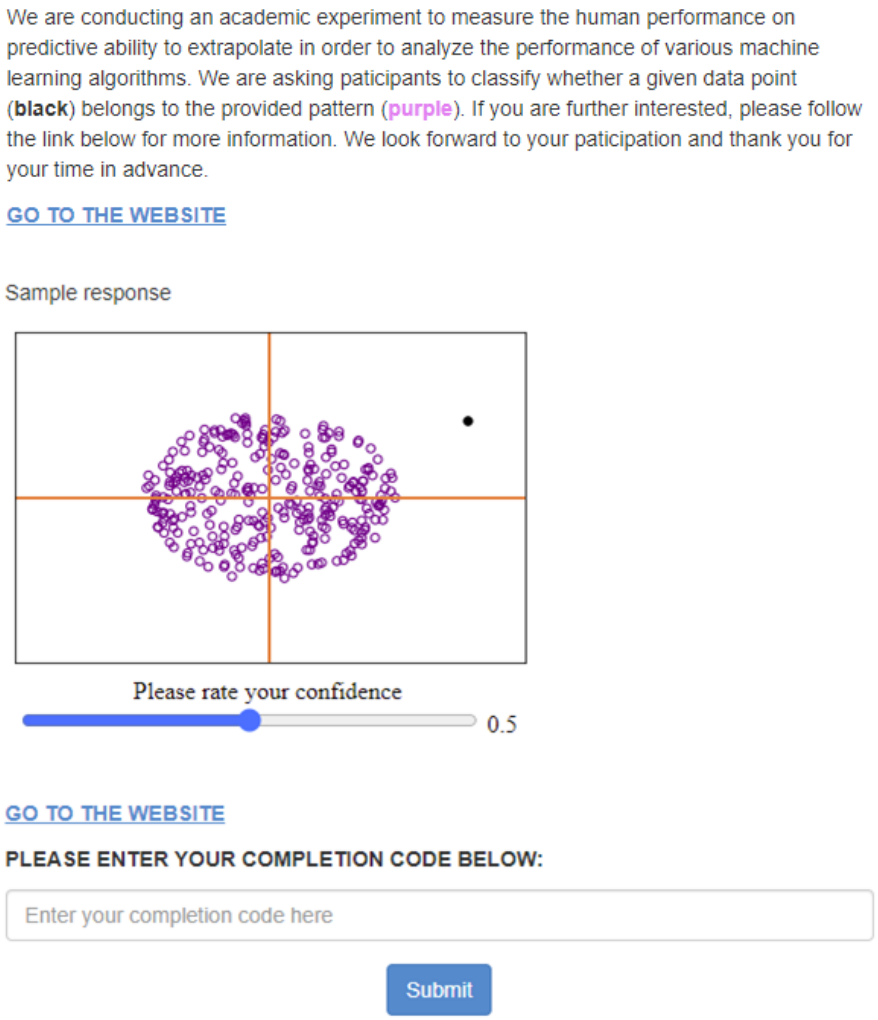


Spiral

Two datasets used in this experiment

Platform

The test will be hosted on Amazon Mechanical Turk (MTurk). It is an online platform that outsources the posted job to a qualified workforce. When the "workers" complete the job, they will be provided with a unique completion code with which they will use to request for compensation. The "requesters" will review the submission quality after which they will decide whether to accept the submission or not. Following job post page is ready to be deployed once the test is finalized.



A number of parameters are still needed to be determined (these parameters are required by MTurk):

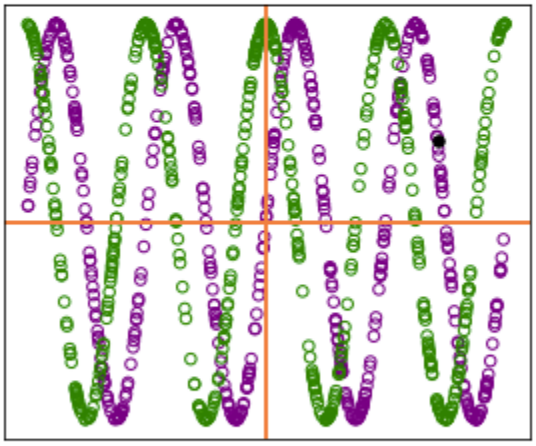
1. Number of responses (currently preset to 50-200)
2. Reward per response (e.g. if set at $0.01 per response, 200 responses will pay workers $2.00)
3. Number of respondents (limiting number of participants)
4. Time allotted for each test (this is defaulted to 1 hour if not changed)
5. Expiration (the job can be posted for a period of time, defaulted to 7 days if not changed)

Tutorial

Prior to the test, the participants are briefly introduced with the concept and provided with instructions. They are acquainted with the test in a step-by-step manner in order to simplify the concept as much as possible.

In fact, for each question, there will be two different classes in the pattern. Therefore, you are actually rating your confidence to measure whether a test point (**black**) belongs to a certain class or not. For our purpose, your task is to predict if a test point (**black**) belongs to the **purple class**.

SUMMARY: IF YOU THINK A **TEST POINT** BELONGS TO THE **PURPLE CLASS** BRING YOUR CONFIDENCE TOWARDS **[1]**. LIKEWISE, IF YOU BELIEVE A **TEST POINT** BELONGS TO THE **GREEN CLASS** LOWER YOUR CONFIDENCE TOWARDS **[0]**.



Please rate your confidence

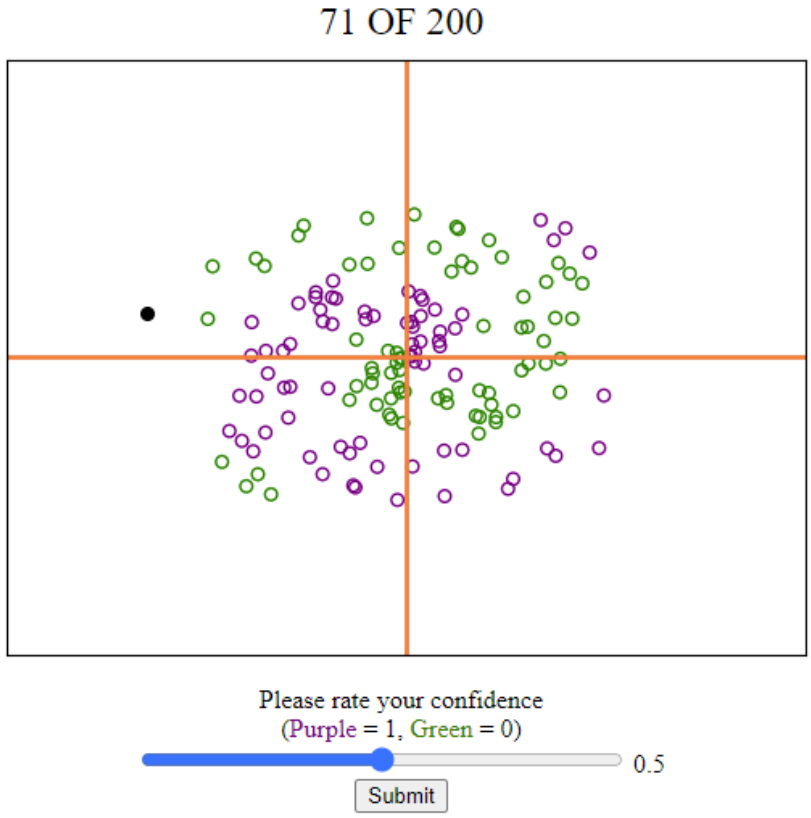
0.5

Prev Page Next Page

Sample page of the tutorial. The participants can revisit this tutorial upon completion.

Test Interface

The test is designed to be as simple as possible to maximize the completion rate. The participants will assess the image (generated in real-time using python), and rate their confidence using the slider. Once set, the participants will submit their responses by clicking the submit button after which the button and the slider will be temporarily disabled until each query is processed. After the submission, the image is regenerated from randomly chosen datasets.



The experiment interface. For a number of responses (currently, participants can choose between 50 and 200 responses), the participants are shown different patterns of two classes (purple and green). They are asked to rate their confidence on whether a test point (black) belongs to purple class or not using the slider (range: 0.0-1.0). Each response is stored in the SQL database which can be retrieved at any time.

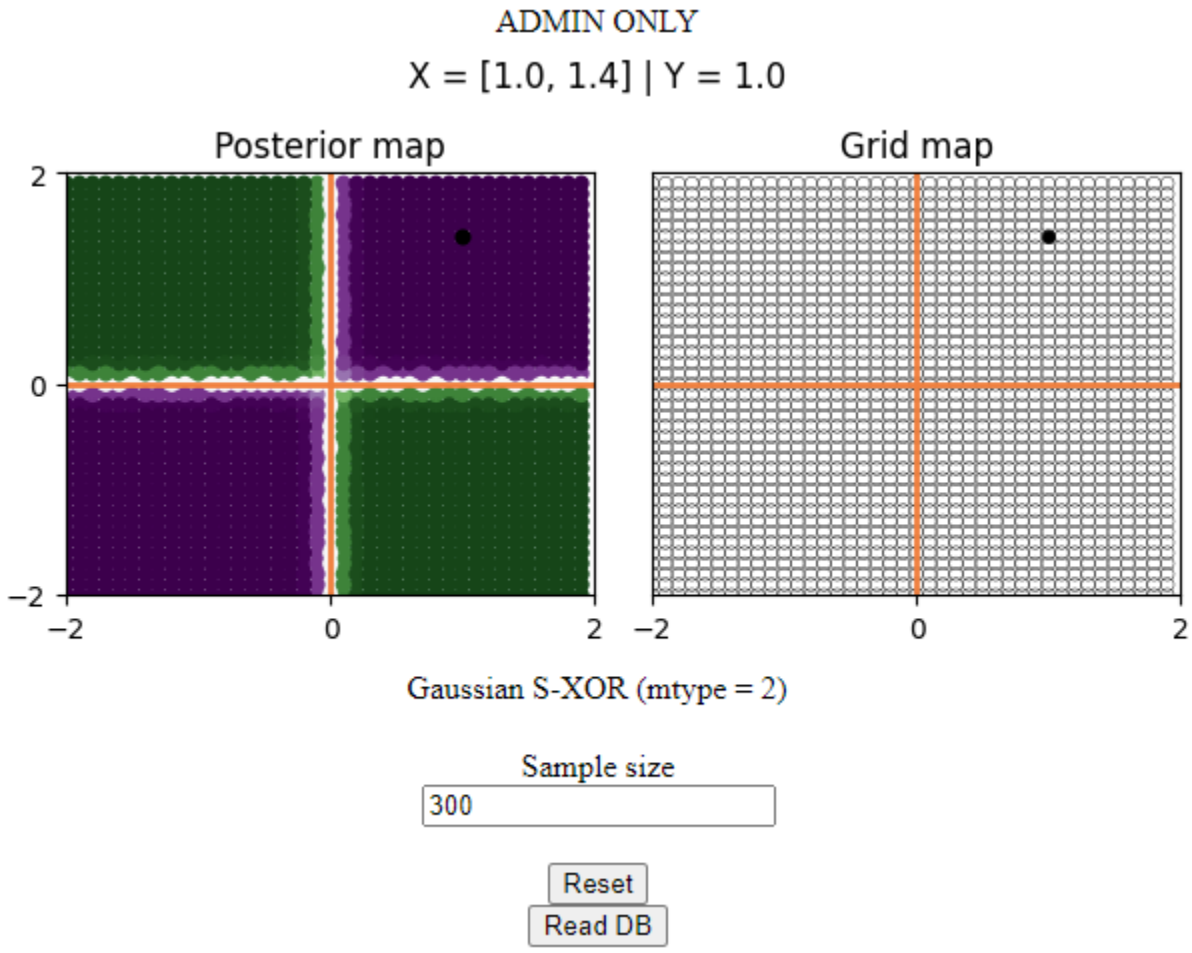
Main Page | Download | Flush DB

ID	USER	HIT	TRIAL	DATASET	PRED	TRUE	X COORDINATES	SAMPLE #	DATE
120	e983c04e-1b73-4a5a-8b83-0110b93aa7f2	50	6	2	0.54	0.5000000000000089	[0.0,0.4]	300	2020-11-22 14:13:47.172494
119	e983c04e-1b73-4a5a-8b83-0110b93aa7f2	50	5	1	0.95	0.9196909168222016	[-0.1,0.8]	300	2020-11-22 14:13:41.164719
118	e983c04e-1b73-4a5a-8b83-0110b93aa7f2	50	4	3	0.06	0.0	[-1.4,0.7]	300	2020-11-22 14:13:34.423583
117	e983c04e-1b73-4a5a-8b83-0110b93aa7f2	50	3	1	0.9	0.9897443241382888	[0.0,1.3]	300	2020-11-22 14:13:27.198725
116	e983c04e-1b73-4a5a-8b83-0110b93aa7f2	50	2	3	0.43	0.0	[-1.4,1.5]	300	2020-11-22 14:13:22.949251

Sample SQL Database

Control Panel (admin only)

This control panel provides researchers the statistics of each generated test inquiry. The left figure displays posterior probability. The title above indicates the coordinates of the test point (denoted as X; used for epsilon/distance calculation and troubleshooting) and the label (denoted as Y; used to score each response). The test point is randomly sampled from the n by n grid shown on the right-side figure. Immediately below these figures is the type of current dataset. The sample size refers to the number of sample points on the test (i.e. the total number of the purple and green hollow points shown in the testing panel). The researchers can navigate to the very first tutorial page by clicking the Reset button or access the database by clicking the Read DB button.



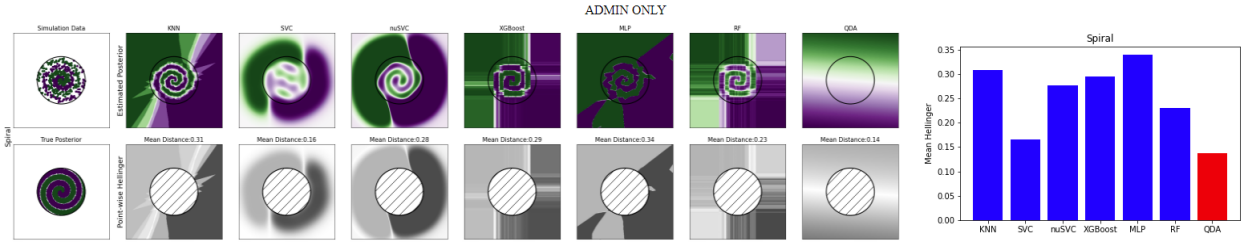
Sample page of the control panel. The posterior map for spiral dataset has not been implemented currently.

Posterior panel (admin only)

Lastly, the posterior panel provides pre-generated posterior probability estimates for 7 sklearn models trained on the simulated datasets (first row first column) which are shown in the first row spanning from 2nd to last columns. Namely, the models used here are KNN, SVC, nuSVC, XGBoost, MLP, RF, and QDA. In the second row are the point-wise Hellinger distance calculations between estimated posteriors and true posteriors (second row first column) spanning outside of $(-1,1)^2$ range (i.e. the region outside of the bounding circle). The bar graph is the mean Hellinger distance again only on the outside of $(-1,1)^2$ range where red bar indicates the lowest mean Hellinger distance among 7 algorithms.

$$H(P,Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i-1}^{k} (\sqrt{p_i} - \sqrt{q_i})^2}$$

Equation for Hellinger Distance



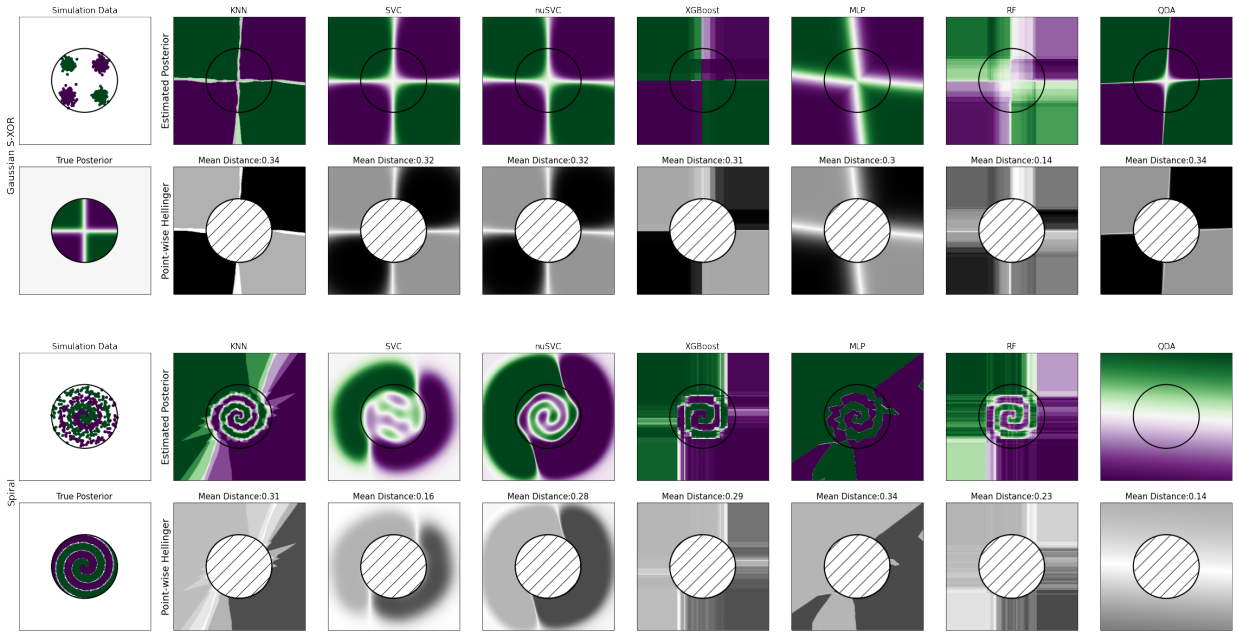
Sample posterior panel. Complete list of these figures are listed in the summary posterior plots section.

Model Hyperparameters

```
########################################################################
Gaussian S-XOR
########################################################################
KNeighborsClassifier(leaf_size=10, n_neighbors=3)
SVC(gamma='auto', probability=True)
NuSVC(gamma='auto', probability=True)
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.10526315789473684, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=128, n_jobs=-1, num_parallel_tree=1, random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
MLPClassifier(alpha=0, hidden_layer_sizes=100, learning_rate_init=0.0001,
              max_iter=7000)
RandomForestClassifier(max_depth=4, n_estimators=64, n_jobs=-1)
QuadraticDiscriminantAnalysis()
```

```
########################################################################
Spiral
########################################################################
KNeighborsClassifier(n_neighbors=7)
SVC(C=10.0, probability=True)
NuSVC(probability=True)
XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
              importance_type='gain', interaction_constraints='',
              learning_rate=0.10526315789473684, max_delta_step=0, max_depth=6,
              min_child_weight=1, missing=nan, monotone_constraints='()',
              n_estimators=256, n_jobs=-1, num_parallel_tree=1, random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None)
MLPClassifier(alpha=0, learning_rate_init=0.0001, max_iter=10000,
              solver='lbfgs')
RandomForestClassifier(max_depth=18, n_estimators=128, n_jobs=-1)
QuadraticDiscriminantAnalysis()
```

Summary posterior plots



Summary Hellinger distance

These are the means of the point-wise Hellinger distances shown in previous sections. The red bar is the lowest mean Hellinger distance among 7 algorithms. These Hellinger distances only account for the region outside of $(-1,1)^2$ range.