UNIVERSITY OF BRISTOL

Crowdsourcing Experiments

by

Barry Park

Technical Report

Systems Centre
Faculty of Engineering

June 28, 2013

UNIVERSITY OF BRISTOL

ABSTRACT

SYSTEMS CENTRE
FACULTY OF ENGINEERING

by Barry Park

This work is all about ...

Contents

List of Figures

List of Tables

Chapter 1

Data Fusion

1.1 Summaries

One difficulty I have found with analysing the results, is how to summarise the results of data fusion. Upon completion of the data fusion using a certain number of crowdsourced reports, we arrive at a posterior distribution. This distribution tells us the probability of the circle being found within a certain region.

There are two difficulties. The first is how to summarise a single posterior distribution, as often the distribution is non-symmetric.

1.1.1 Expectation Values

For some function $f(x)$, under a probability distribution $p(x)$, then the average value of $f(x)$is called the Expectation Value, written as $\mathbb{E}[f(x)]$. For a discrete distribution, the expectation is given by

$$\mathbb{E}[f(x)] = \sum_x p(x)f(x) \tag{1.1}$$

and for a continuous distribution we take the integral

$$\mathbb{E}[f(x)] = \int p(x)f(x)dx \tag{1.2}$$

When a finite number of samples N are drawn from the probability distribution $p(x)$, then the expectation can be approximated by

$$\mathbb{E}[f(x)] \simeq \frac{1}{N}\sum_{n=1}^{N} f(x_n) \tag{1.3}$$

The expectation value of a function can be viewed as a weighted average.

Figure **??** shows a Gaussian and Gamma probability functions, and the corresponding expectation value $\mathbb{E}[f]$ where $x = \{0, 1, ..., N\}$ with $f(x) = x/N$ and $N = 1000$.

1.1.2 Point Estimates from Posterior Distributions

Point estiamtes are useful for reporting a single summary value of the distribution. Probably the most common is to take the expected value of the distribution, as shown in Figure **??** which gives us the mean value of the parameter that distribution describes.

Another common point estimate is the Maximum A-Posteriori of the distribution, which corresponds to a mode of the distribution.

For data fusion, the expected value is commonly reported as it is the mean estiamtion we are interested in

1.1.3 Spread of values

One standard measure for how much a distribution varies from its mean is variance. The varaince of a function $f(x)$ can be defined in terms of expectations as

$$var[f] = \mathbb{E}[(f(x) - \mathbb{E}[f(x)])^2]$$

which can be rewritten as

$$var[f] = \mathbb{E}[f(x)^2] - \mathbb{E}[f(x)]^2$$

For a posterior distribution, another standard measure of the spread around the mean of the distribution is the use of Bayesian confidence intervals, also known as credible intervals. A credible interval defines a region of a distribution where there is a defined probability that the true parameter being estimated lies within. For example, if the probability that a variable x lies between 0.1 and 0.9 is 0.95, then $0.1 \leq x \leq 0.9$ is a 95% confidence interval

More formally, we can define a credible interval of a probability distribution $p(x)$ as

$$\int_{L}^{U} p(x) = 1 - \alpha$$

where U is the upper credible bound, L is the lower credible bound, and α sets the value of the credible interval. Credible intervals in more than 1 dimension are called credible regions. Credible regions are invariant under reparametrization A common credible interval is to centre the interval on the posterior mean. If we select the credible interval with minimum size, this region is called the highest probability density (HPD) region. HPDs are not invariant under reparametrization, but an HPD will always be a credible interval afterwards.

Calculating the HPD of ran arbitrary distribution can be computationally challenging, and is often achieved through sampling the distribution

1.2 Sampling Distributions

In most real applications, it is difficult to analytically integrate a distribution, as a closed form solution may not exist. We therefore have to resort to numerical integration.

A relatively simple brute force approach is to evaluate the distribution at a series of evenly spaced points, and then interpolate between these using easy to integrate polynomials.

These methods can give accurate approximations of the integral when the spacing between points is small, but as the state space increases, the function evaluations grow exponentially.

A second approach to approximating an integral is the use of Monte Carlo (MC) sampling,or Monte Carlo integration. This family of methods estimates the distribution by sampling under a specific probability distribution, then using these samples to estimate the function.

1.2.1 MC sampling

Imagine the scenario where we have a function $f(x)$ which we wish to normalise such that

$$\int_{a}^{b} f(x) = 1 \tag{1.4}$$

This is a very common operation as the integral of a valid probability distribution must equal 1. To achieve this we can calculate

$$p(x) = \frac{f(x)}{\int_a^b f(x)} dx \tag{1.5}$$

However, we may not be able to directly calculate the integral in (1.4). In this case it may be possible to use Monte Carlo integration

As a brief introduction to sampling methods, let's look at two approaches. The first will generate samples from a uniform distribution, the second will draw samples directly from a gaussian probability distribution.

Let $f(x)$=x, which we wish to calculate the expectation value of x under a normal probability density p(x) with $\mu = 0$ and $\sigma = 1$. Also assume that we cannot calculate this expectation analytically, so we resort to sampling. One method of sampling would be to draw points from a uniform distribution, and then calculate the value of (1.1) using these samples. An alternative approach would be to sample the probability distribution $p(x)$and use these samples to calculate (1.3).

We can see in FIGURE that for a typical set of samples, the expectation value using the samples drawn from $p(x)$gives a robust estimate o the true expectation value, with little change in variance of the estimator with increasing state size. Drawing from the uniform distribution showed an increase in estimator variance with increased state size.

Here we had the option of drawing samples directly from $p(x)$. Often this is not the case, and we cannot draw samples directly from this distribution. In cases like this, we could generate samples from a distribution that is easier to sample, then make some sort of adjustment to take in to account that we are not sampling directly from $p(x)$.

It is not always best to draw samples from $p(x)$- we can at times out perform samples drawn from this distribution. Imagine the case where $f(x)$is low where $p(x)$is high, but $f(x)$has large values elsewhere. This would mean that our true expectation value is heavily influenced by high value, but low probability states of x. For example, a stock portfolio might be thought to return £1 99.9% of the time, but return -£1Million 0.1% of the time. Despite having a low probability, this negative return clearly outweighs any benefits that we can achieve. However, if we were to sample $p(x)$, and calculate our expectation, or expected return, from a finite number of samples, we may not actually get any samples from this low probability region, and therefore get a poor estimate of the expectation. We therefore wish to draw samples from the most important regions of the product $f(x)p(x)$.

One approach that is used for both these cases is importance sampling. Importance sampling lies at the core of particle filters.

1.2.2 Importance Sampling

Assume we have a probability density $p(x)$, which is difficult to sample, over a function $f(x)$. We can define a proposal distribution $q(x)$which we choose to be easy to sample. We can therefore write

$$\begin{aligned}\mathbb{E}[f(x)] &= \int f(x)p(x)dx \\ &= \int f(x)\frac{p(x)}{q(x)}q(x)dx \end{aligned} \tag{1.6}$$

If we generate L samples $x^{(l)}$ which we draw from the proposal distribution $q(x)$, we can then treat (1.6) as a finite sum of the form shown in (1.3), leading to

$$\mathbb{E}[f(x)] \simeq \frac{1}{L}\sum_{l=1}^{L}\frac{p(x^{(l)})}{q(x^{(l)})}f(x^{(l)}) \tag{1.7}$$

The quantity given by $\tilde{w}(x^{(l)}) = \frac{p(x^{(l)})}{q(x^{(l)})}$ is known as the importance weights. This weight compensates for drawing samples from the proposal density q(x), rather than the true probability density p(x). Often, we will not know the normalisation factor of the original distribution. We can correct for this by normalising our importance weights. The normalised weights are given by

$$w(x^{(l)}) = \frac{\tilde{w}(x^{(l)})}{\sum_{j=1}^{L}\tilde{w}(x^{(j)})} \tag{1.8}$$

Therefore our expectation value is given by

$$\mathbb{E}[f(x)] \simeq \sum_{l=1}^{L} f(x^{(l)})w(x^{(l)}) \tag{1.9}$$

Clearly, if we wish to get a good estimate of the expectation value $\mathbb{E}[f(x)]$, then we want to pick a proposal distribution $q(x)$ that provides good coverage of the product $f(x)p(x)$

1.3 Data fusion

Data fusion is the process of combining data from difference sources in to meaningful state estimates. There are a number of different approaches to this, including Bayesian

Probability Theory, Dempster-Shafer approaches, and fuzzy logic. The approach outlined here is based on the Bayesian approach.

1.3.1 Probability Theory

Bayesian probability theory views probability as a measurement of uncertainty in the parameters of interest. Uncertainty is a measure of the degree of beleif we have in the state of the parameter.

Imagine the scenario where we have two types of coloured balls - red and blue - mixed together in two bags. We know that in bag 1, there are 70% red, and 30% blue, and in the second bag the mix is 50-50. If we were to place our hand in to bag 1 and pull a ball out at random, clearly there is a 70% chance of it being red. Now imagine the case where we do not know which bag is which, and upon pulling a blue ball out of that bag, we wish to determine the probability of it being from bag 1 or bag 2. Clearly, we should have a greater belief that the ball is from bag 2. We can take this to a more extreme example where we now change the ratio of balls in bag 1 to 99% red, and 1% blue. In this case, we should be an even greater belief that the blue ball was drawn from bag 2. Bayesian probability theory gives us a way of measuring this belief in a robust manner.

Let X be a random variable that represents the bags we can draw from. If the probability of choosing bag 1 is 0.4, then we could write

$$p(X = x_1) = 0.4 \tag{1.10}$$

or in a more compact form as

$$p(x_1) = 0.4 \tag{1.11}$$

Intuitively, if we only have two bags, then the probability of choosing bag 2 is

$$p(x_2) = 1 - p(x_1) \tag{1.12}$$

$$= 0.6 \tag{1.13}$$

In other words, if we do not choose bag 1, then we choose bag 2.

...

Chapter 2

1D Positional Experiments - 2 Categories

2.1 Experimental Setup

2.2 2 Category Decisions

2.3 Dataset

2.4 Initial Analysis

2.1 shows a bar chart of the responses at each position. At position 0.5, the number and of left and right responses were found to be exactly equal. At points away from the centre, the number of left or right responses at each position remains near constant. This indicates that there is a sharp transition from left to right (for most workers) as the position of the circle is increased from $0 - 1$. There is little ambiguity or fuzziness in the workers' responses.

2.4.1 Non-gold responses

We can apply simple gold standard data filtering to the dataset. By setting the extreme values of the circle position at 0 and 1 to gold standard data, we can filter out some workers. A total of 7 workers, or 10% of the dataset, were filtered out. **??** shows the workers that were filtered out. In 2.2a we can see the response models of the individual workers. Some show the opposite from what we might expect, by responding 'Left-hand

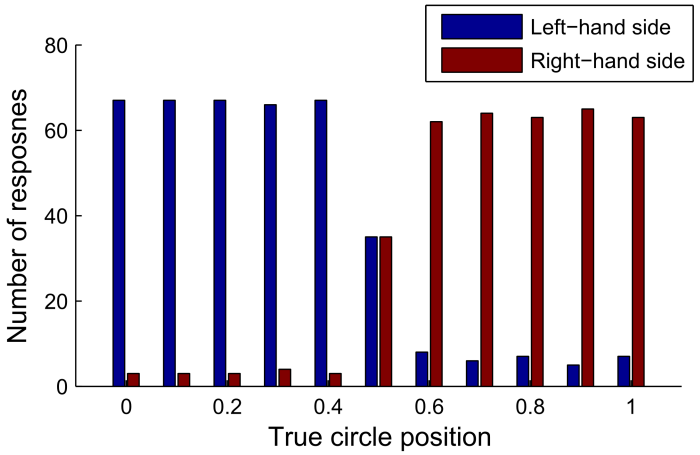
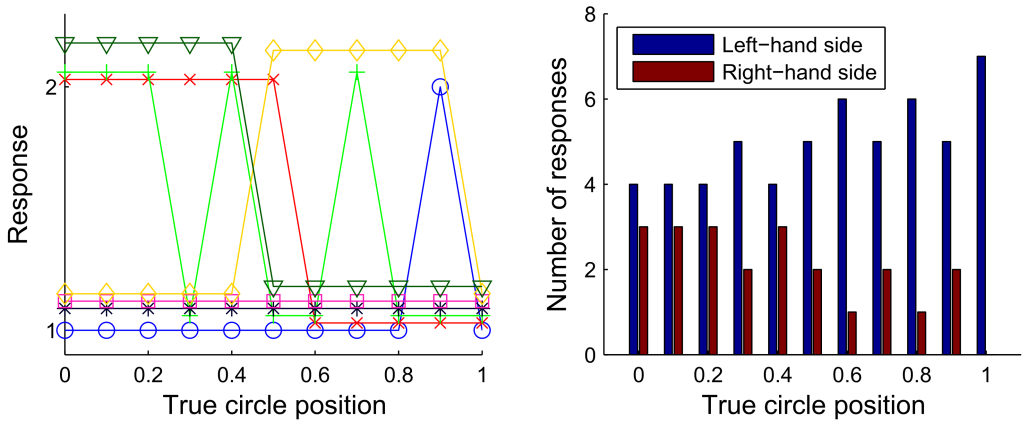


FIGURE 2.1: The number of left and right responses at each circle position. The total number of workers is 70.



(A) (B)

FIGURE 2.2: The number of left and right responses for the non-gold response workers. The total number of workers is 7.

side' when the circle was on the far right. This suggests that they have not understood the answer they are giving. 2.2b shows the responses at each circle position for

The resulting dataset after gold filtering is shown in figure 2.3

2.5 Softmax Function Fitting

FIGURE shows the softmax function fit for the full dataset and gold standard dataset.

2.6 Fusion Results

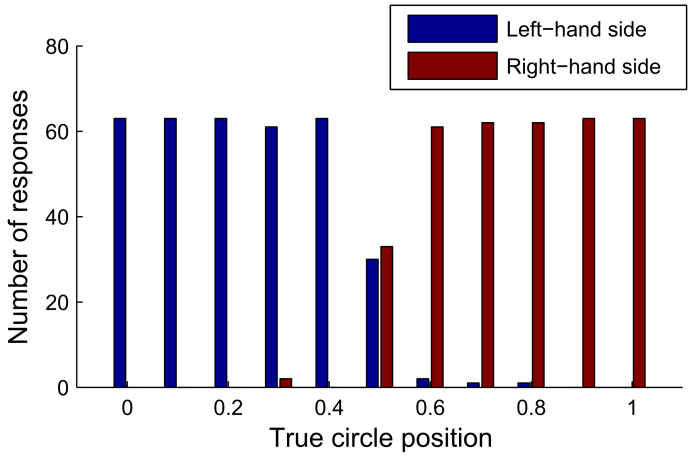
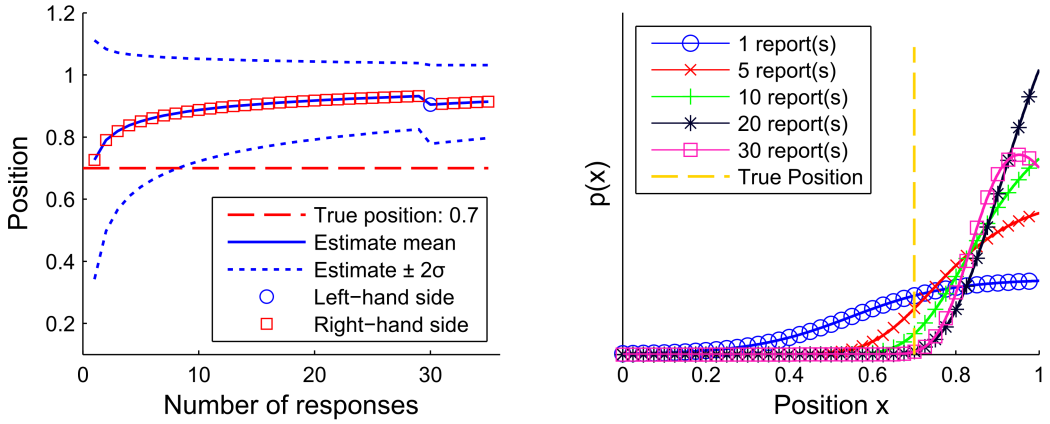


FIGURE 2.3: The number of left and right responses at each circle position with gold filtering. The total number of workers is 63.



(A) (B)

FIGURE 2.4: Estimating the position of a circle as new reports are received a) The cahnge in posterior mean and standard deviation with responses b) Examples of the posterior distribution at different numbers of reports

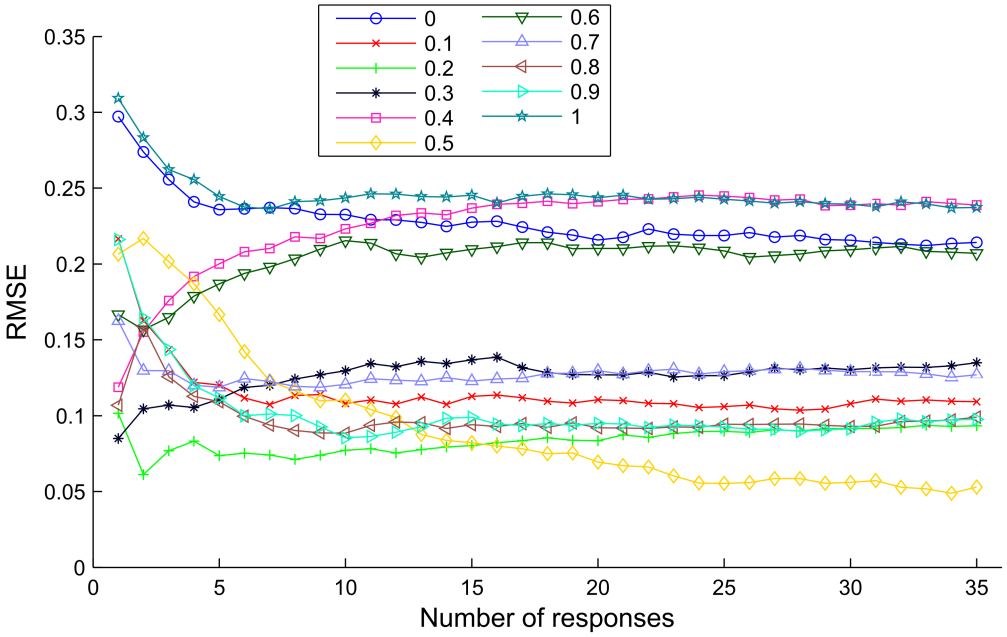
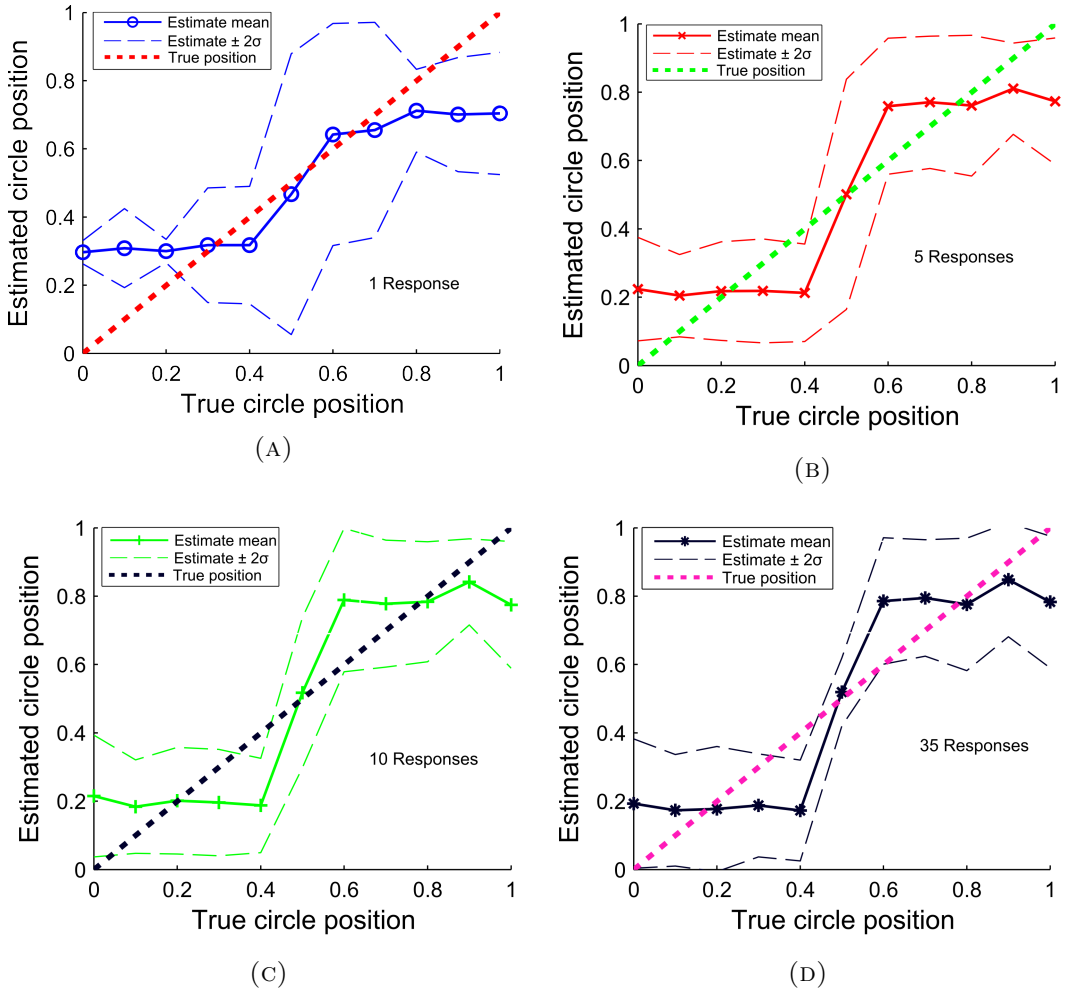


FIGURE 2.5: The RMSE at each circle position as the number of responses increases. The RMSE was generated over 50 simulation runs



(A) (B)

(C) (D)

FIGURE 2.6: The mean performance of estimating the circle's position. The mean line is the expectation value of the posterior distribution taken from 50 simulations. The standard deviation is for the spread of expectation values over the simulations - not the spread in individual posteriors. The data was collected for varying numbers of responses with a) 1 response b) 5 responses c) 10 responses d)35 responses

Chapter 3

1D Positional experiments - 3 Categories

3.1 Experimental setup

EDIT IMAGES Crowdflower [1] is an online service that allows for simple tasks to be posted to a number of crowdsourcing worker markets. By providing a wrapper and common API for several crowdsourcing provides, Crowdflower boasts the largest crowd of workers out of any platform. Due to limitations in its API, only relatively simple tasks can be posted. Crowdflower was used to post a series of simple images to the crowd, getting feedback from hundreds of workers.

All the following experiments consisted of showing a static image of a circle to workers who were required to pick the best 1 of 2 or more avaiable options which described the location of the circle in the image.

Each worker was shown a series of images, where the position of the circle changed from image to image, with each worker providing a single response for each image. A total of 11 images were shown to each worker, with the position of the circle ranging from 0 to 1 [2], as shown in 3.1 . For each experiment, a simple html page was created with the images stored in a public Dropbox folder.

3.2 3 Category Decisions

Let D be an m valued discrete variable that represents the possible worker responses, where $D \in \mathbb{N}$. Workers were asked to describe the position of the circle using one of

[1] http://crowdflower.com/

[2] The units are arbitrary, but the values $0 : 0.1 : 1$ are used throughout

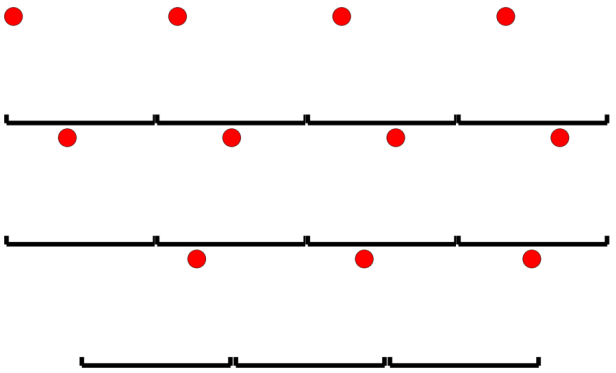


FIGURE 3.1: The images posted to the crowd. These show the circle at positions $0 : 0.1 : 1$ giving a total of 11 images

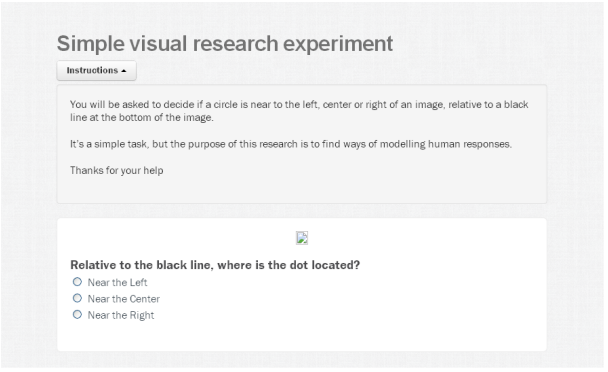


FIGURE 3.2: The task window for a worker, requiring one of three responses

three responses $R = \{'NearTheLeft', 'NearTheCentre', 'NearTheRight'\}$, which were mapped to $D = \{1, 2, 3\}$, setting $m = 3$. An example of the worker GUI is shown in 3.2 (current screenshot does not show image as dropbox is not accessible from BAE systems - get another screenshot). Each worker was shown a total of 11 images.

A total of 100 workers were asked to provide feedback, leading to a dataset of 1100 responses. 1 worker set, for some reason, was provided by two workers (with one worker providing 10 responses, and another 1 response), so the total number of usable workers was 99

The aim is to estimate the ground truth position of the circle given a response from 1 or more workers.

3.2.1 Dataset

A large amount of data is returned by crowdflower, including worker responses, their declared location, their IP address, the crowdsourcing platfrom that they provided responses through, and the time each response was received. Table 3.1 shows a sample of three columns of the returned data which will be useful for the following analysis.

worker id	**image description**	**responses**
16800977	dot located at 0.3	Near the Left
17519077	dot located at 0.0	Near the Left
2666559	dot located at 0.7	Near the Right
16800977	dot located at 0.5	Near the Centre
16362086	dot located at 0.4	Near the Centre

TABLE 3.1: Example of relevant data collected

workers	**ground truth**	**responses**
16800977	0.3	1
17519077	0.0	1
2666559	0.7	3
16800977	0.5	2
16362086	0.4	2

TABLE 3.2: Translation of text responses in to numerical ordinal responses

In order to make analysis simpler, the strings of data in Table 3.1 were transformed in to numerical data as shown in Table 3.2.

Let the random state variable $X \in \mathbb{R}$ be the circles true position. For this experiment the circle can take on one of $n = 11$ possible positions given by $X = \{0.0, 0.1, ..., 0.9, 1.0\}$. Each worker i provides a single response for each circle location. We can define the set of responses for a single worker as the vector Z_i where each element $Z_{i,j} \in D$, and $Z_{i,j}$ is the response of worker i for true circle position X_j

3.3 Initial Analysis

We can visualise the responses from workers as a bar chart as shown in **??**. This chart shows the number of left, centre and right responses from all the workers at each of the true circle positions. From this chart we can see that the majority of the responses appear to be reasonable.

3.3.1 Individual Worker Response Models

Each worker provides a single response at each true circle position. We can create a *response model* for each worker which simply describes that workers response at each position. A model for an individual is given by the workers response vector Z_i. An example of a response model is shown in **??**

3.3.1.1 Unique Response Model Groups

Workers can be grouped by their respective models. Each group is formed by workers who have identical response models i.e. at every true circle position X_j, they reported the same position category. By grouping workers together, we are able to see the number of different response models there are. The top three most common response models are shown in **??**. There was found to be 24 unique response model groups, 10 of which had more than 1 worker with that model, as shown in **??**

3.3.1.2 Model Consistency

We can define a response model as consistent if the difference between adjacent values in a workers response vector are always $\geqslant 0$, or $\leqslant 0$. We can define the response change, the change between adjacent values in a workers response vector Z_i, as

$$RC(Z_{i,j}) = Z_{i,k+1} - Z_{i,k}$$

where $k = \{1, 2, ..., n-1\}$

We can define three types of consistent model

$$M_{\text{type}} = \begin{cases} M_I & \text{if } \forall j, RC(Z_{i,j}) \geq 0 \text{ and } > 0 \text{ for some } i \\ M_D & \text{if } \forall j, RC(Z_{i,j}) \leq 0 \text{ and } < 0 \text{ for some } i \\ M_C & \text{if } \forall j, RC(Z_{i,j}) = 0 \end{cases}$$

where M_I are increasing models, M_D are decreasing models and M_C are constant models. We expect the majority of consistent models to be increasing models, as the true position of the circle increases from left to right for each element of Z_i. A constant consistent model represents a worker who supplied the same response no matter what the circles position. This points towards a worker who either did not understand the question, or more likely did not attempt the question at all - they simply filled in the submission form as quickly as possible. A consistent decreasing model would imply a user who either did not understand the question, or someone who is intentionally providing poor responses. **??** shows all the increasing response models. The number of workers with each increasing response model is shown in **??**

3.3.1.3 Model Symmetry

Some of the models showed *symmetry*,(Is it symmetry? How do we define it?). Let $f_s(Z)$ be a function which transforms a workers response vector, so that each 1 is transformed to 3, and a 3 is transformed to 1. Two types of symmetry can be defined

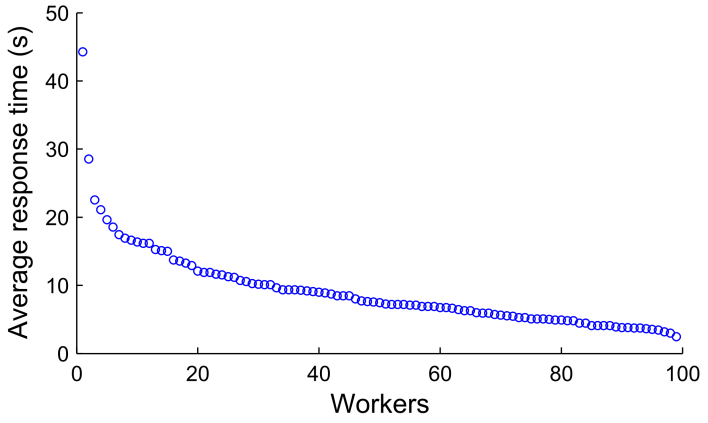


FIGURE 3.3: The ranked average response time for each worker.

$$M_{\text{type}} = \begin{cases} M_{RS} & \text{if } \forall j, f_s(Z_{i,j}) = Z_{i,(n+1-j)} \text{ and } Z_{i,j} \neq Z_{i,(n+1-j)} \text{ for some } j \\ M_{TS} & \text{if } \forall j, Z_{i,j} = Z_{i,(n+1-j)} \end{cases}$$

where M_{RS} is response symmetry, and M_{TS} is true symmetry. A model has true symmetry if, at locations $0.5+a$ and $0.5-a$ the worker gives the same response. This is an undesirable type of model, as it means that using only this type of model, we would be unable to distinguish between positions such as 0.4 and 0.6.

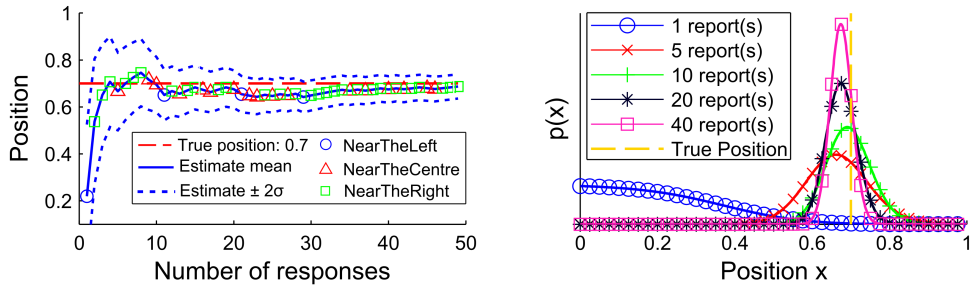
Response symmetry is where at locations $0.5+a$ and $0.5-a$, a worker gives the *opposite* response. For example, if they responded 'NearTheRight' at 0.3, they would responded 'NearTheLeft' at position 0.7. A response symmetric model looks the same when rotated through 180°. A worker who responds only 'NearTheCentre' for every true position of the circle is defined to have a model type M_{TS}. **??** shows the symmetric models. The majority of symmetric models are of the type M_{RS}

3.3.2 Time Analysis

Other than the responses of each worker, the start and finish time of each worker was available. This data only tells us when each worker began the task, and when she finished it - it does not tell us the amount of time they spent on each response.

FIGURE shows the average response time for each worker. The average response time is simply given by

$$\text{Average response time} = \frac{\text{Total task time}}{\text{Number of responses}}$$



(A) (B)

FIGURE 3.4: Estimating the position of the circle as new reports are received a) The change in posterior mean and variance with responses b) Examples of posterior distribution

3.4 Softmax model fitting

3.5 Initial Fusion Results

FIGURE shows the typical impact on the posterior distribution as reports arrive, in the case of unfiltered data, with all of the dataset having been used to train the softmax model. In general the posterior becomes more peaked as more reports arrive. We can see in FIGURE that the distribution tends towards the true value as more reports arrive. In FIGURE we can see that as we increase the number of reports, the estimate becomes more stable. In FIGURE, we see an example of the reports leading to a poor estimate of circle position. In this case, there is a bias in the model that leads to the mean to be over estimated.

The main process used for this analysis was to train the softmax model using approximately 50% of the data, and then test the performance using the remaining data. This is repeated through several simulated runs, where the training data is randomly sampled from the full dataset.

First, lets look at one typical simulation run. In this example, the circle is placed at the value 0.7. As we receive responses from workers, we can update our estimate of where the circle is. 3.4a shows the impact of reports coming in on the mean and variance of the posterior distribution. Initially, we receive a report of 'NearTheLeft' which in this case leads to an inital poor estimate of the circles position. The next three reports all state 'NearTheRight' which has the effect of shifting the posterior mean towards the right and leading to an improved estimate of circle position. The complete posterior distribution is shown in 3.4b, which shows the reduction in variance as more reports arrive.

?? shows the RMSE for each circle position over 50 simulation runs, as the number of responses is increased. The RMSE was measured between the mean of the posterior distribution and the true circle position. The worst performance is achieved at the outer

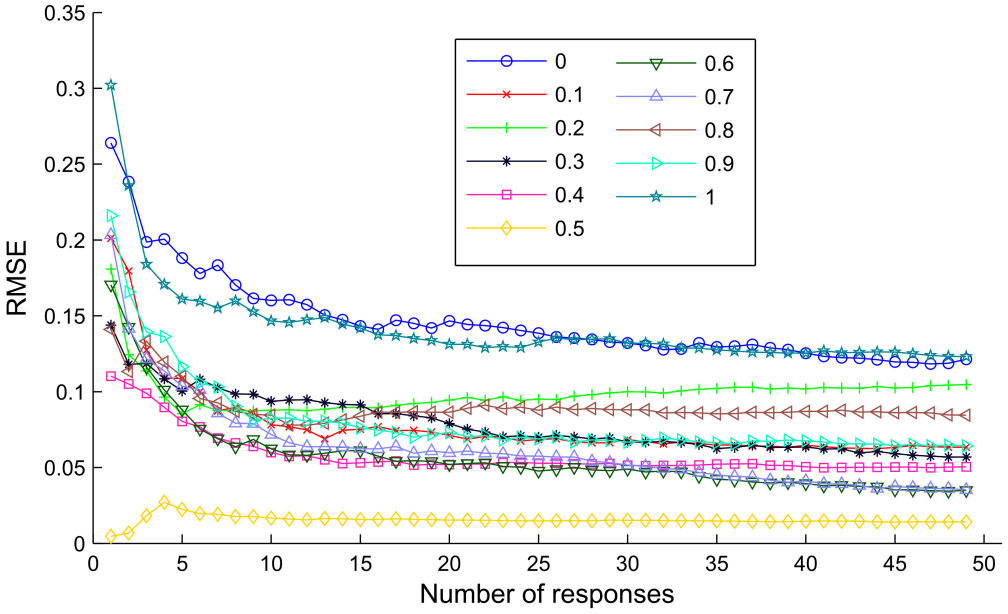


FIGURE 3.5: The Root-Mean-Square-Error of the posterior with the circle at varying positions, as the number of responses is increased. The RMSE was generated over 50 simulation runs

points 0.0 and 1.0. This can be attributed to the performance metric of the mean, or expectation value, of the distribution being used. At the outer regions of the experiment, the posterior distribution becomes heavily skewed, making the mean a poor summary statistic in this case. It is difficult to summarise multimodal or skewed distributions using a single metric.

The best performing position is 0.5. Again, this is to be expected as the mean of the 'NearTheCentre' likelihood function is close to 0.5, so if the majority of reports are of this type, we would expect a posterior with mean close to 0.5 even after only a small number of reports.

We can view the performance of the fusion by viewing the true circle position against the predicted circle position. This is shown in 3.6

3.6 Limiting the worker variety/models

It might seem intuitive to think that in an ideal world, for the purposes of data fusion, everyone would have the same response model i.e. the number of unique response models is one. That way we would know precisely the internal model that a person was using, and we could fit an accurate function to this. To test this intuition, a series of experiments were carried out in order to see the impact of limiting the variety of models found in the data.

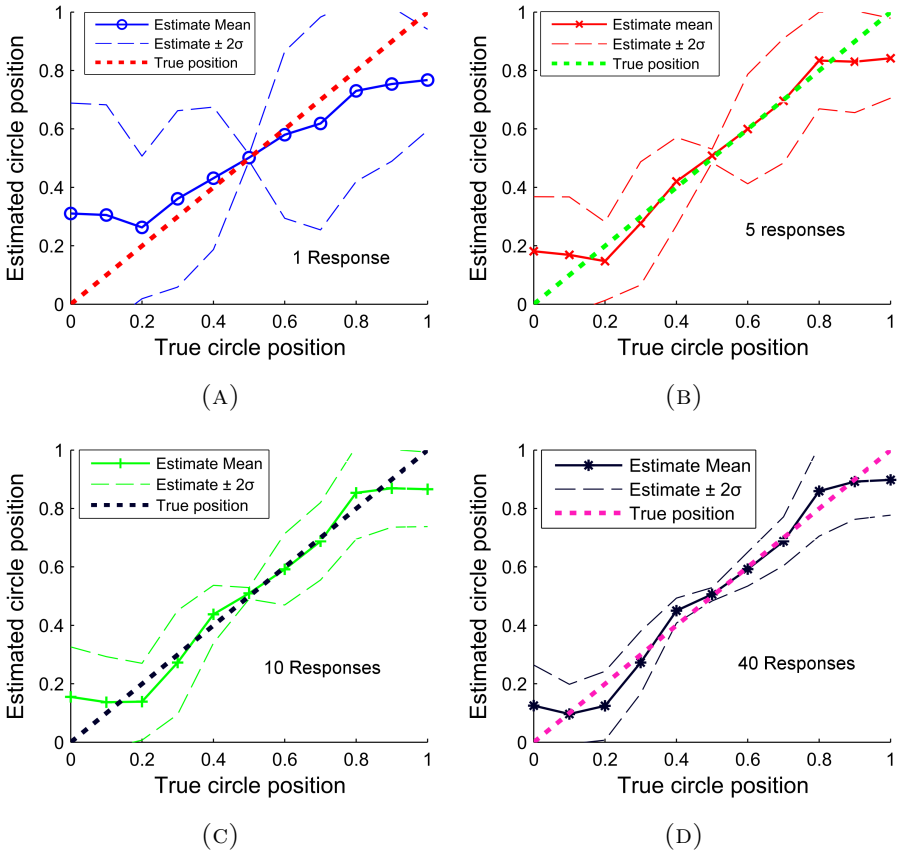


FIGURE 3.6: The mean performance of estimating the circle's position. The mean line is the average expectation value taken from 50 simulations runs. The standard deviations are for the deviations in the expectation values (not to be confused with the variance in the posterior distribution). The mean lines were generated using varying numbers of responses for each circle positon, with: a) 1 response b) 5 responses c)10 responses d) 40 responses

3.6.1 Approach

As shown in **??**, the number of unique models in the dataset with more than 1 worker associated to it was found to be 10. To investigate the impact of response model variety, these 10 models were used to create a series of training and testing datasets. 10 datasets were created in total.

The first dataset contained only the responses from the workers with the most popular response model, the model shown in **??**. This dataset contained the responses from 16 workers in total. The second dataset contained only the responses from the first and second most popular response models, which totalled 21 workers. This process was continued, with the tenth dataset containing the top 10 response models.

These datasets were then used to train the softmax model. As the datasets were now different sizes, only 8 randomly selected workers (half the number of workers in the first dataset) from each dataset were used to train the model, and then 8 remaining workers

were randomly selected used to test the model. This process was carried out for 100 experimental runs for each dataset.

3.6.2 Model fitting

3.6.3 State Estimation Results

?? shows the results of fusing different numbers of models. In **??**, we can see that using 1 response model, then our mean estimate of circle position after fusion essentially follows the response model - the shape is qualitatively the same. As the number of models is increased, the mean estimate is improved. Increasing from 6-10 models shows little improvement in mean estimate. This is likely due to the small training and testing sets used. As response models 6-10 have relatively few workers in them compared to models 1-5, then the chances of them being sampled for training or testing purposes is relatively small.

In order to investigate the impact of the less popular response models, the same process was carried out using models 3-10. So for each dataset, the first two models were included. This increased the size of the dataset from 16, to 42 workers. The results are shown in **??**. These results show the same trend as the smaller dataset, with the increase in worker response models improving the data fusion results. The conclusion that can be drawn from this is that an increase in the variety of response models improves the overall fusion. We do not want all workers to behave the same, as is the case with 1 response model in **??**. We want some spread of results, that gives each position of the circle as unique a input signature as possible. However, too much variety, and we would have an input that approximates noise.

3.7 Data Filtering

This section looks at the impact on the whole dataset using different data filtering approaches.

3.7.1 Common Models

3.7.2 Consistent Models

3.7.3 Symmetric Models

3.7.3.1

3.7.4 Gold Data Filtering

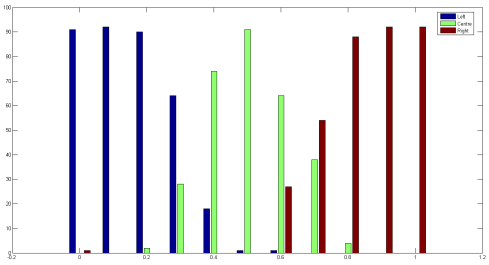


FIGURE 3.7: The number of Left(Blue),Centre(Green) and Right(Red) worker responses at ground truth circle position after filtering the data using gold data and a threshold of 2 wrong answers

Gold data filtering is a heuristic approach for filtering out poor performing workers. It relies on the creation of 'gold questions' ; questions where there is a known acceptable answer. In this experiment, there is ambiguity about where the classes NearToLeft, NearToCentre, and NearToRight start and finish. However, we would expect all responders to answer NearToLeft when the circle was located at 0, and NearToRight when the circle is located at 1. We can use these locations as gold questions, setting an expected response for each. If a worker gets a threshold value (or greater) of gold questions wrong, then we not only ignore their responses for the gold question, but also all other questions as they are deemed to be unreliable or of poor quality.

Using the ground truth locations of 0 and 1 as gold questions, the data was filtered. Setting the threshold at 2 (workers who got both gold questions wrong were ignored), the filtered responses are shown in 3.7.

The result of setting the gold data threshold to 1 is shown in **??**.

3.7.5 Filtering comparisons

Chapter 4

Circular Experiment

Workers were asked where the circle was compared to a point in the image. The circle was placed in 32 positions, with the angle between the dot and circle being varied. Each worker was asked to provide 16 responses.

Originally the workers were asked to provide all 32 responses, but no workers decided to do this. It is unclear if this was due to the number of questions being asked, which was relatively high for each worker, or if it was another issue such as the time of posting the task.

4.1 Data fusion

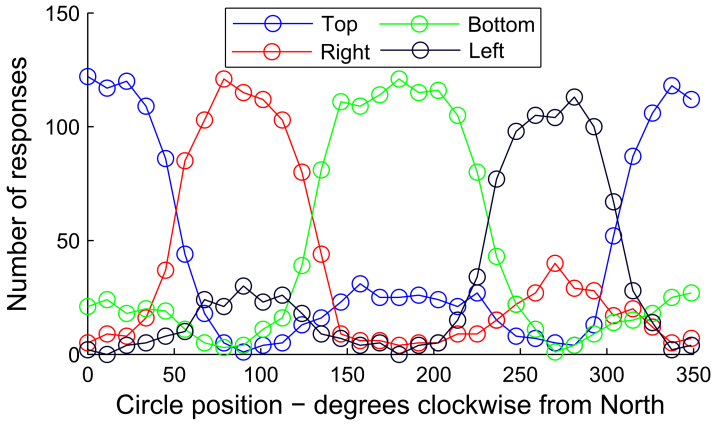


FIGURE 4.1: The number of responses at each of the 32 circle positions

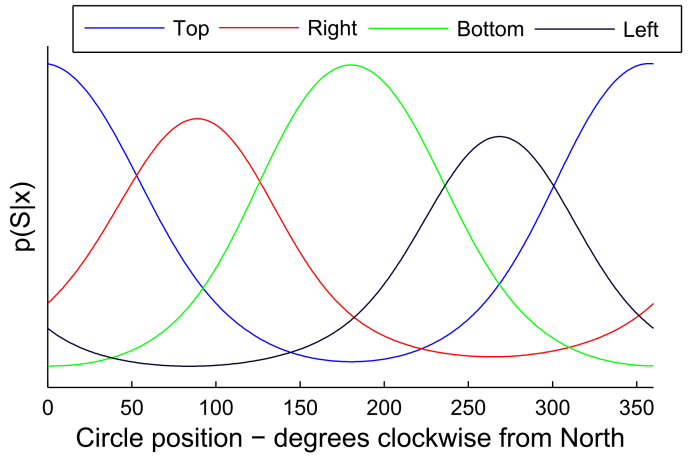


FIGURE 4.2: The maximum likelihood softmax model for the raw data

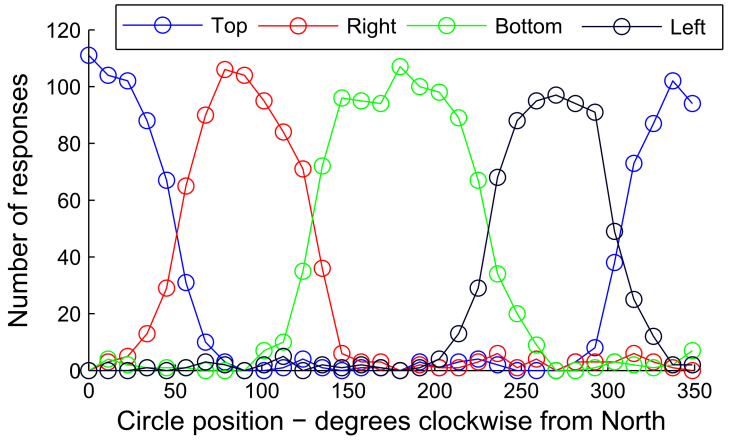


FIGURE 4.3: The gold data filtered responses. Gold responses were set at 0, 90, 180 and 270. If a worker got a single gold response incorrect, they were filtered out.

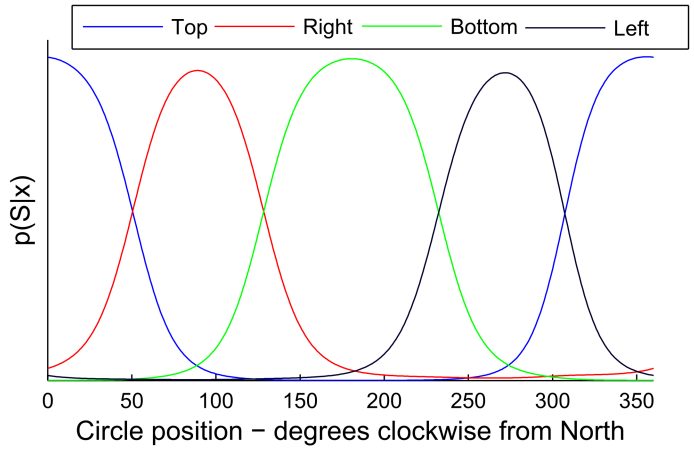


FIGURE 4.4: The maximum likelihood softmax model for the gold filtered data.

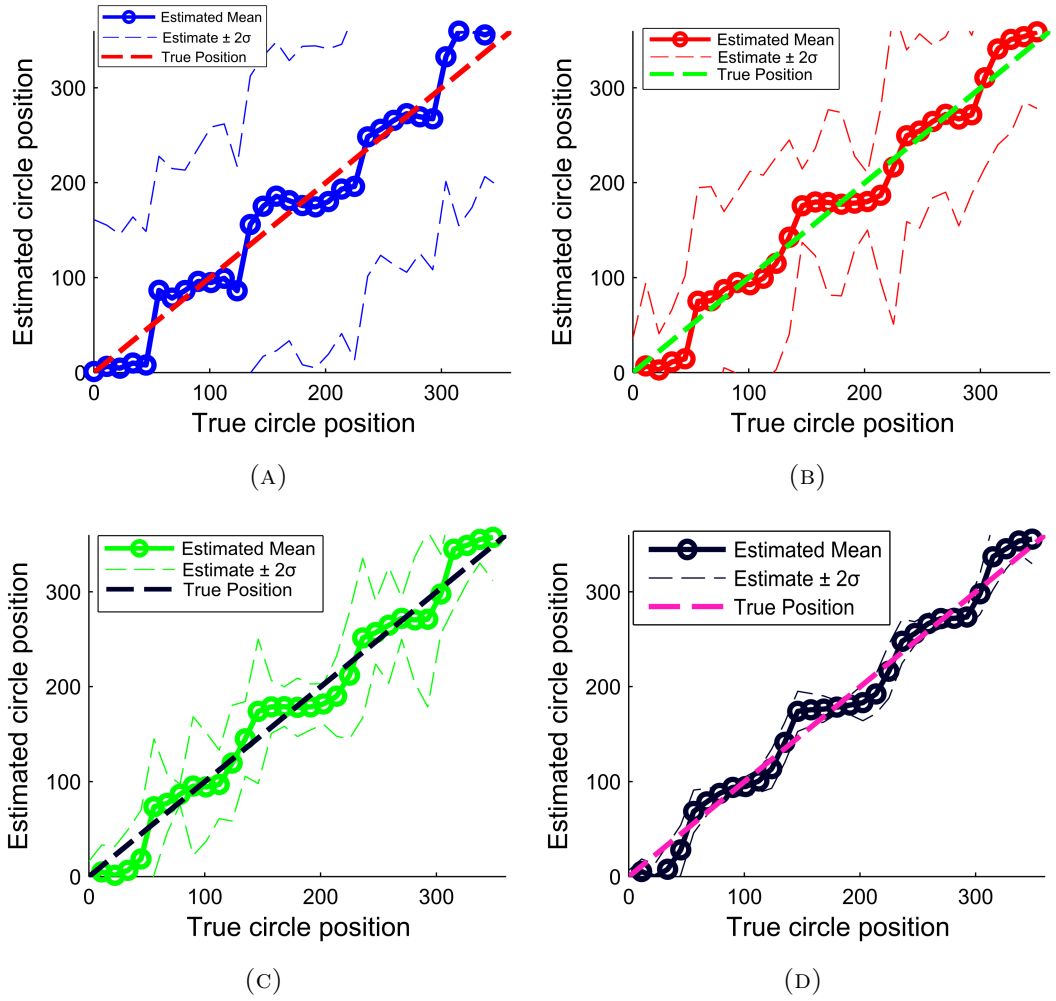
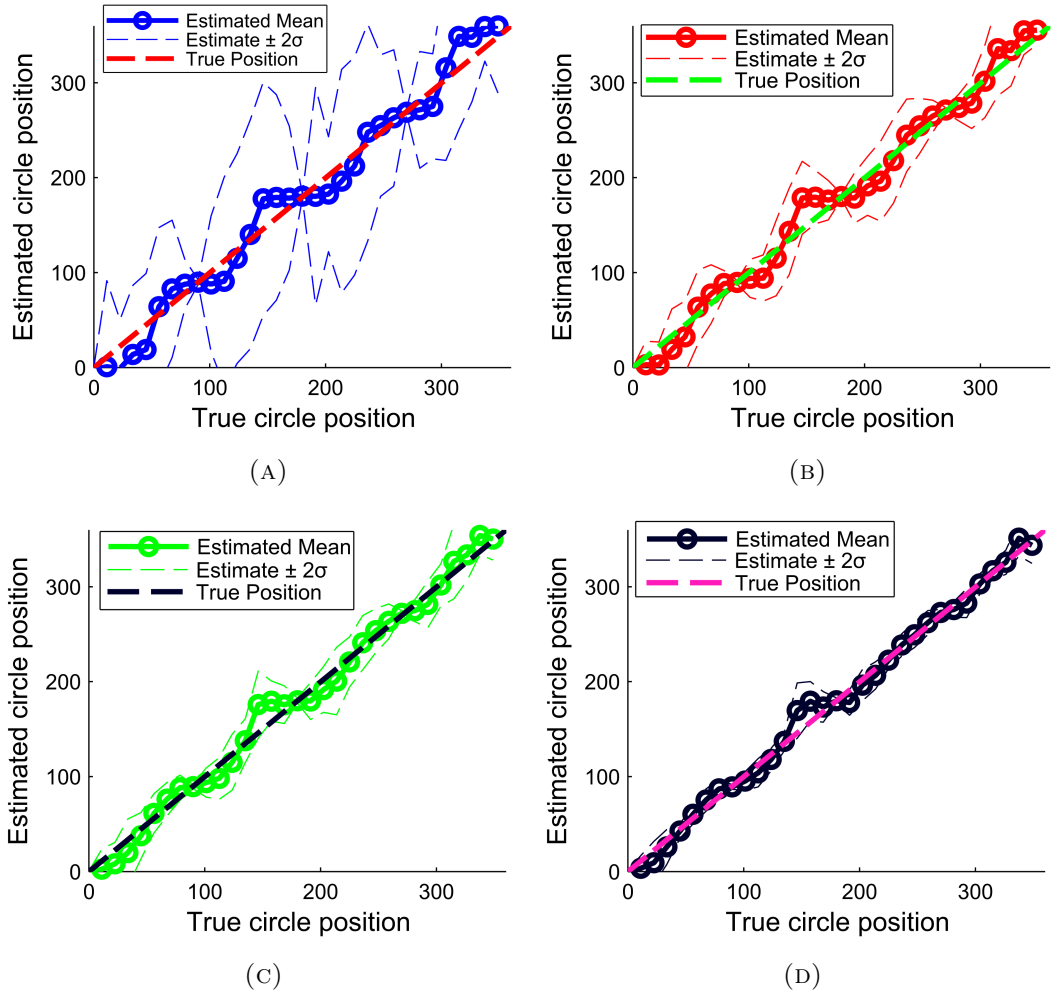


FIGURE 4.5: The mean performance of estimating the circle's position from the circular response. The mean line is the expectation value of the posterior distribution taken from 50 simulations. The standard deviation is for the spread of expectation values over the simulations - not the spread in individual posteriors. The data was collected for varying numbers of responses with a) 1 response b) 5 responses c) 10 responses d)40 responses



(A) (B)

(C) (D)

FIGURE 4.6: The mean performance of estimating the circle's position from the circular response gold data. The mean line is the expectation value of the posterior distribution taken from 50 simulations. The standard deviation is for the spread of expectation values over the simulations - not the spread in individual posteriors. The data was collected for varying numbers of responses with a) 1 response b) 5 responses c) 10 responses d)40 responses