Bayesian analysis of subjective ranking data using Thurstonian Models: Tutorial, novel methods, and an open-source library

Oscar Giles[1,2], Richard Romano[1], Gustav Markkula[1]

[1]Institute for Transport Studies, University of Leeds, Leeds

[2]School of Psychology, University of Leeds, Leeds

Address correspondence to:

Dr Oscar T Giles,

Institute for Transport Studies,

University of Leeds,

Leeds,

LS2 9JT

Email: o.t.giles@leeds.ac.uk

GitHub: https://github.com/OscartGiles

Abstract

Subjective ranking data are ubiquitous in behavioural research, design and marketing. Subjective ranking data arise when subjects are asked to order a set of items or stimuli according to some criteria. For example, a psychologist may ask subjects to order a set of faces according to their perceived 'friendliness', while an automotive engineer may rank a set of vehicle prototypes according to their subjectively perceived stability. Thurstonian models provide a powerful framework for analysing ranking data, but have not seen widespread use in the behavioural sciences. By modelling discrete rankings as arising from a set of continuous latent variables, Thurstonian models can extend the flexibility of generalised linear models to ranking data. This allows us to estimate aggregate (or "consensus") ranks for a group of participants, incorporate covariates into the models, test for differences between conditions and populations, and assess the degree of agreement between rankings from multiple subjects. Here we provide an introduction to Thurstonian models, and illustrate how these, when coupled with Bayesian estimation approaches, can provide a powerful tool for analysing ranking data. Specifically, using three example datasets, including both simulated data and data from our own research on subjective impressions of driving simulators, we illustrate how we can fit and interpret Thurstonian models. We demonstrate a number of novel approaches for summarising the Bayesian Thurstonian model fits using distance metrics, and provide an open source code library which allows users to flexibly specify and fit Thurstonian models. This implementation uses a novel reformulation of the Thurstonian model, which admits the use of Hamiltonian Monte Carlo, a fast and efficient algorithm for fitting Bayesian models.

Introduction

Subjective ranking data are produced when people order a set of items or stimuli according to a particular criterion. Subjective rankings can provide a robust measure of preference which can often be easier to elicit than continuous estimates of magnitude, and are not subject to individual differences in understanding of scale. For example, intuitively it seems easier for a test driver to say that one vehicle handles better than another, than to quantify exactly how much better it is. Subjective ranking data are ubiquitous across the behavioural sciences, product design and marketing. Despite this, it can be challenging to analyse ranking data, not least because there is a lack of appropriate statistical tools available to enable researchers to do so.

A number of statistical models of ranking data exist. Of particular note is the Thurstonian model, developed by Thurstone (1931). Thurstonian models map discrete ranking data to a set of latent continuous variables, allowing the model to be easily extended to include covariates (Johnson & Kuhn, 2013). Thus Thurstonian models allow us to estimate an aggregate ranking from a set of rankings provided by individuals, as well as compare aggregate rankings between experimental conditions and populations, or even predict an aggregate ranking given a set of covariates.

While the Thurstonian model of ranking data have existed for almost a century, it has not seen widespread use in data analysis. This is largely because of difficulty fitting the models parameters using tradition optimisation methods, as the likelihood function cannot be analytically evaluated when there are more than two items being ranked (see Johnson & Kuhn (2013) for an overview). However, these models can be fit using Markov Chain Monte Carlo (MCMC) methods which sample from a Bayesian posterior distribution over the model parameters (Gelman, 2014). This is made possible by employing a rank censoring method, and can be readily implemented in probabilistic programming frameworks such as JAGS (Plummer, 2003). This also brings the added value of Bayesian estimation which allow us to quantify our uncertainty in the parameter estimates, as well as take account of prior knowledge about the inference problem at hand (John K. Kruschke & Liddell, 2017a). Fitting a joint posterior distribution over Thurstonian model parameters allows for uncertainty to be quantified with regards to key inferential goals. We refer to Thurstonian models fitted using Bayesian estimation approaches as Bayesian Thurstonian Models (BTMs).

As well as enabling regression analysis to be carried out with ranking data, BTMs provide a theoretically grounded method for aggregating rankings provided by multiple subjects (Lee,

Steyvers, & Miller, 2014). When a ground truth ranking exists (i.e. there is a correct ordering of the items of interest), aggregate rankings can often be closer to the ground truth than most or all of the rankings produced by individuals. This phenomenon is known as the wisdom of crowds effect, and was first described by Sir Francis Galton in relation to estimates of weight (Galton, 1907). While the wisdom of crowds effect can be observed across a wide variety of tasks, from estimating the solubility of organic compounds (Boobier, Osbourn, & Mitchell, 2017) to doctors predicting patient health outcomes (Kattan, O'Rourke, Yu, & Chagin, 2016), it has recently been demonstrated that the same phenomena can also be observed with ranking data (Lee et al., 2014). Thus BTMs are likely to be a useful tool for decision making when multiple subjects independently assess a set of items.

Given the potential broad appeal of BTMs for analysing ranking data across research fields, here we provide a comprehensive tutorial on their use. We provide example datasets from our own research, comparing subjective rankings of vehicle system prototypes on real world test tracks and high-fidelity driving simulators. We introduce a number of novel methods for interpreting BTM model fits. Specifically we demonstrate how established metrics for measuring distances between rankings can be used to (i) provide meaningful summaries of BTM fits, (ii) perform contrasts between conditions of interest, and (iii) assess the degree of agreement between multiple subjects. Finally, we demonstrate how the Thurstonian model can be reformulated to overcome common inefficiencies in model fitting. This admits the use of Hamiltonian Monte Carlo, a fast and efficient algorithm for fitting high dimensional Bayesian models. These methods are implemented in an open source Python library which allows users to flexibly specify and fit BTMs.

Ranking Notation

Rank data arise when individuals order a set of items according to some criterion. For example, a subject may rank three faces, labelled A-C, according to their perceived happiness, from least happy to most happy. For notation, we can numerically index the subject's ranking. The ranking $y = [2,3,1]$, as in Table 1, would mean that C was judged the least happy, A the intermediate and B the most happy. For shorthand we refer to this ranking using the notation 2-3-1. Note that this ranking refers to the rank of each item, and should not be confused with an "ordering" – i.e. the items listed in order of preference (in this case C-A-B). In addition, ranking data can also be generated from other data types. For example, if a subject rates items on a continuous scale we can readily convert this to a ranking.

Table 1: An example of ranking notation.

Item	A	B	C
Ranking	2	3	1

The Thurstonian model

The Thurstonian model is a generative model of ranking data. Just as we might model a continuous measure as being normally distributed around a mean, Thurstonian models assume that individuals' rankings are distributed around some aggregate ranking. In Thurstonian models the aggregate ranking is represented on a continuous latent scale. When ranking K items we can denote each of the items positions on the latent scale as the row vector $\boldsymbol{\mu} = [\mu_1, \dots, \mu_K]$, where μ_k is a real valued number.

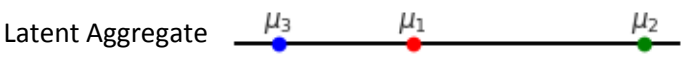

Figure 1: Latent Aggregate ranking

The latent aggregate ranking is then mapped to a discrete ranking by assigning a rank to each μ_k. For example, Figure 1 shows a case in which the latent aggregate ranking of three items is $\boldsymbol{\mu} = [\mu_1 = -5, \mu_2 = 0.2, \mu_3 = -8]$. This would have the discrete aggregate ranking of 2-3-1, as the first item is the middle latent rank, the second is the largest and the third is the smallest.

Individual's rankings

Thurstonian models assume that individuals' rankings are distributed around the aggregate ranking. Individual rankings are also represented on a continuous latent scale, and we can denote the latent ranking of an individual as the row vector $\boldsymbol{z} = [z_1, \dots, z_K]$. For a given individual we assume that the latent ranking of the kth item, z_k, is distributed by a normal distribution (the latent ranking distribution) with mean μ_k and a subject specific standard deviation parameter σ which is shared across all items. Figure 2 illustrates this for the above example when the latent aggregate rank is $\boldsymbol{\mu} = [\mu_1 = -5, \mu_2 = 0.2, \mu_3 = -8]$. We can see that the mean of the distributions for item 1 (μ_1) is lower than item 2 (μ_2) and the distributions overlap very little. Intuitively this suggests that when we repeatedly draw $\boldsymbol{z}$ from these distributions z_1 will almost always be smaller than z_2, although there may be a small number of cases where the order changes. We can also see that the mean of the distribution for item 3 (μ_3) is lower than for item 1, but the distributions overlap to a greater extent. This suggests that over repeated draws from the distribution z_3 will normally be smaller than z_1, but there will be a larger proportion of cases in which the ranking is reversed. Finally the distribution for item 3 is smaller than, and does not overlap at all with that of item 2, suggesting z_3 will always be smaller than z_2 over repeated draws of $\boldsymbol{z}$.

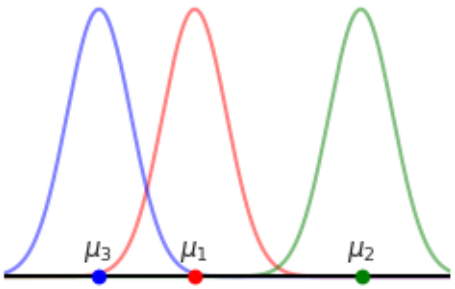

Figure 2: Latent ranking distribution for a single participant

We can again map the latent ranking for an individual $\boldsymbol{z}$ to a discrete ranking by assigning a rank to each z_k,

$$\boldsymbol{y} = rank(\boldsymbol{z}).$$

For example, if $\boldsymbol{z} = [z_1 = -4.5, z_2 = 0.23, z_3 = -6]$ then the discrete rank $\boldsymbol{y}$ would equal 2-3-1, which in this case is the same as the aggregate ranking. We can see that while the aggregate ranking is the most probable ranking of $\boldsymbol{z}$, other rankings are possible and their probabilities will depend on the position and overlap of the latent ranking distributions shown in Figure 2, which is determined by the latent aggregate ranking $\boldsymbol{\mu}$ and the subject specific standard deviation parameter σ. While we could fix this parameter to be same for all subjects, allowing it to vary between subjects means that we can model some subjects as being more likely to produce rankings closer to the aggregate than others. This is particularly useful when ranking items for which there is a *ground truth*, when the aggregate ranking can often be closer to the ground truth than the ranking of any individual (Lee et al., 2014). In this context the subject's σ parameter is often thought of as an index of their expertise (smaller σ is associated with a greater probability of ranking the items correctly).

Extending to covariates

A major benefit of modelling rankings on a latent continuous scale is that the Thurstonian model can be extended to a generalised linear model by allowing the aggregate ranking $\boldsymbol{\mu}$ to vary as a function of one or more covariates. In this case the aggregate ranking would depend on a matrix of regression coefficients $\boldsymbol{\beta}$ and a row vector $\boldsymbol{X}$ of covariates,

$$\boldsymbol{\mu} = \boldsymbol{X\beta}.$$

For example, if we wanted to allow the aggregate ranking to vary between two conditions we could set $\boldsymbol{X} = [1, 0]$ for the first condition and $\boldsymbol{X} = [0,1]$ in the second. In matrix notation the latent aggregate ranking in the first condition would then be given by,

$$\boldsymbol{\mu} = [\beta_{1,1} \quad \beta_{1,2} \quad \beta_{1,2}] = [1 \quad 0] \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \beta_{1,2} \\ \beta_{2,1} & \beta_{2,2} & \beta_{2,2} \end{bmatrix}$$

while for the second it would be given by,

$$\boldsymbol{\mu} = [\beta_{2,1} \quad \beta_{2,2} \quad \beta_{2,2}] = [0 \quad 1] \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \beta_{1,2} \\ \beta_{2,1} & \beta_{2,2} & \beta_{2,2} \end{bmatrix}$$

In practice $\boldsymbol{X}$ could contain any number of categorical or continuous predictors.

Model fitting

Fitting Thurstonian models is challenging due to difficulties in analytically evaluating the likelihood function, $\pi(\boldsymbol{y}|\boldsymbol{\mu}, \sigma)$. Figure 2 provides an intuition that the probability of any given ranking $\boldsymbol{y}$ depends on the overlap between the latent distributions, which is determined by $\boldsymbol{\mu}$ and σ. While it is obvious that the probability of any latent ranking $\boldsymbol{z}$ is given by,

$$\pi(\boldsymbol{z}|\boldsymbol{\mu}, \sigma) = \prod_{i=1}^{K} \pi(z_i|\mu_i, \sigma),$$

where $\pi(z_i|\mu_i, \sigma)$ is given by the normal probability density function, to calculate $\pi(\boldsymbol{y}|\boldsymbol{\mu}, \sigma)$ we need to integrate $\pi(\boldsymbol{z}|\boldsymbol{\mu}, \sigma)$ over all possible $\boldsymbol{z}$ that respect $rank(\boldsymbol{z}) = \boldsymbol{y}$**.** Thus the posterior distribution over the Thurstonian model (when applied to a single ranking) is provided by,

$$\pi(\boldsymbol{y}|\theta) \propto \pi(\boldsymbol{\mu}, \sigma) \int d\boldsymbol{z}\, \pi(\boldsymbol{z}|\boldsymbol{\mu}, \sigma)\, \mathbb{1}(\mathrm{rank}(\mathbf{z}) = \boldsymbol{y})$$

It becomes prohibitively challenging to assess the likelihood function when there are more than two items being ranked.

Bayesian Estimation and Markov Chain Monte Carlo

The difficulties with evaluating the likelihood function of the Thurstonian model makes it extremely challenging to apply traditional fitting algorithms (i.e., maximum likelihood estimation). However, Bayesian estimation can be used to fit Thurstonian models (Yao & Böckenholt, 1999). Bayesian estimation specifies a joint posterior distribution over the Thurstonian model's parameters using Bayes' rule,

$$\pi(\boldsymbol{\theta}|\boldsymbol{Y}) \propto \pi(\boldsymbol{Y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}).$$

This states that the posterior distribution, which is the probability of the model parameters $\boldsymbol{\theta}$ given the data $\boldsymbol{Y}$**,** is proportional to the product of the likelihood function $\pi(\boldsymbol{Y}|\boldsymbol{\theta})$ and a prior $\pi(\boldsymbol{\theta})$. Note that here $\boldsymbol{Y}$ is a matrix where each row corresponds to an individual, empirically observed ranking.

While an introduction to Bayesian estimation is beyond the scope of this article, a number of exemplary resources exist for readers unfamiliar with this method of inference (Gelman, 2014; J. K.

Kruschke, 2010; John K. Kruschke & Liddell, 2017b; McElreath, 2015). However, in the next section we provide detailed explanations of the information provided by BTMs, so a detailed understanding is not a prerequisite.

While calculating the exact posterior distribution is generally intractable, Markov Chain Monte Carlo (MCMC) algorithms can provide a robust approximation of the posterior (J. K. Kruschke, 2015). It has been shown that MCMC algorithms can be used to fit BTMs (Yao and Böckenholt, 1999; Yu, 2000; and Johnson and Kuhn, 2013), and can be implemented in probabilistic programming frameworks such as JAGS (Johnson and Kuhn, 2013). MCMC algorithms work by indirectly taking a representative sample from the joint posterior distribution. This is achieved by generating a Markov chain through parameter space, where the Markov transitions *preserve* the posterior distribution (Betancourt, 2017). We can work directly with the samples returned by the MCMC algorithm to evaluate expectations over functions of interest. For example, to estimate the mean of the posterior we can simply calculate the mean of the MCMC samples.

MCMC algorithms have previously been applied to Thurstonian models using a "rank censoring constraint" which obviates the need to analytically evaluate the likelihood function. The rank censoring constraint essentially ensures that the Markov Chain has zero probability of transitioning to locations in parameter space where $\boldsymbol{z}$ does not have the same rank as $\boldsymbol{y}$ (see Johnson and Kuhn, 2013, for an overview). Popular MCMC algorithms, such as the Metropolis algorithm (Hastings, 1970; Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) work by first proposing a transition in parameter space, and then accepting the transition according to a probabilistic acceptance criterion The rank censoring constraint ensures that any proposed transition which does not respect the rank censoring constraint is immediately rejected. However, this is inefficient as the MCMC algorithm will often propose many transitions that have zero probability of being accepted, slowing the Markov chains progression. Here we provide a novel implementation of the Thurstonian model which removes this inefficiency and allows us to fit the models using Hamiltonian Monte Carlo (Hoffman & Gelman, 2014; Neal, 2011), a cutting edge MCMC algorithm which uses gradient information to efficiently explore the posterior distribution.

Applications of the BTM

In this section we provide worked examples of using BTMs to analyse rank data. For the analysis we developed an open source Python library, PyThurstonian

(https://github.com/OscartGiles/PyThurstonian_public), which allows for flexible specification and fitting of BTMs using Stan, while abstracting the implementation details away from the user. The library also provides a number of convenience functions for interrogating and visualising the model fits which are used to generate all the included figures. All code required to recreate the analysis is provided as part of the PyThurstonian package and all figures are generated using convenience functions included in the package.

We first provide a detailed example using a simulated dataset, introducing a number of metrics to aid inferences. We then analyse two real datasets from our research group, which seeks to test how subjective rankings regarding vehicle prototypes differ between those obtained in the real world and those obtained in high fidelity driving simulators. Our key aim is to illustrate what information is provided by the BTM and how we can use this make sensible inferences. In these datasets subjects ranked three items, although BTM and PyThrurstonian can be used to fit datasets with any number of items.

Simulated data set

When using a new statistical method for the first time it is often beneficial to work with simulated data before using the method on real world data. The benefit of this approach is that we can simulate data using our model and then ensure that we can recover the model parameters and make sensible inferences. Thus we first generated an example dataset which consisted of 10 simulated participants who ranked three items in two conditions (C1 and C2). For the simulated dataset we set the latent ranks to $\mu = [0 \quad 1 \quad 2]$ in condition C1 and $\mu = [0 \quad 0.4 \quad 0.8]$ in condition C2. The aggregate rank is therefore 1-2-3 in both conditions, but the participant's ranks will be much more variable in condition C2 due to the relative closeness of the latent distributions. The subject specific σ parameters were randomly sampled from a lognormal distribution. Figure 3 shows the latent aggregate scores, as well as the latent distribution and sampled $\boldsymbol{z}$ for two of ten participants with different σ parameters.

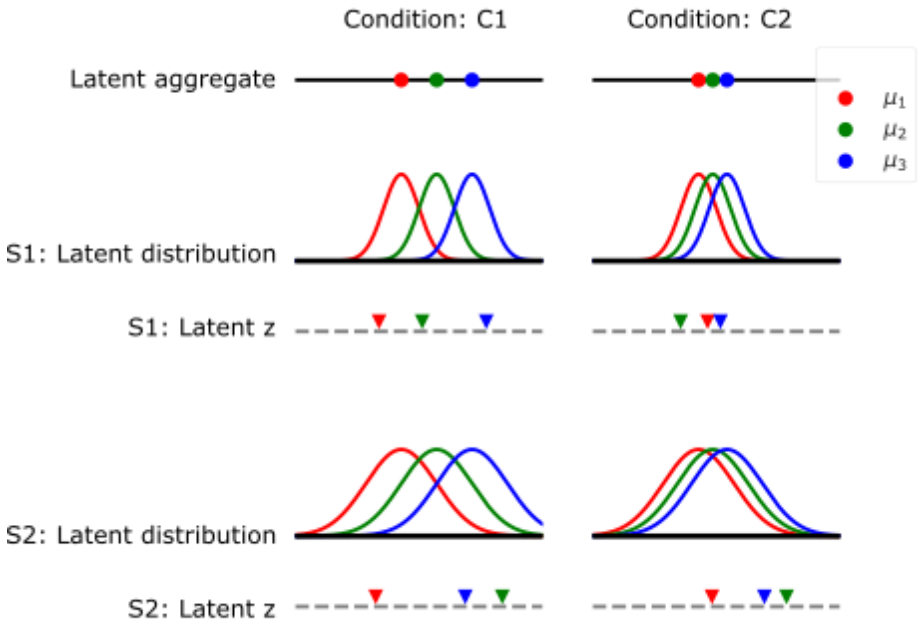

Figure 3: Latent aggregate ranking, and latent distributions with resulting draws from the distribution for two participants (S1 and S2) with different σ parameters, for conditions C1 and C2.

The simulated dataset is summarised in Figure 4. We can see that in condition C1 the ranking 1-2-3 was chosen over 50% of the time, while 2-1-3 was chosen sometimes. In condition C2 the distribution of rankings appeared to be much more variable. Next we can fit a BTM to the dataset using PyThurstonian.

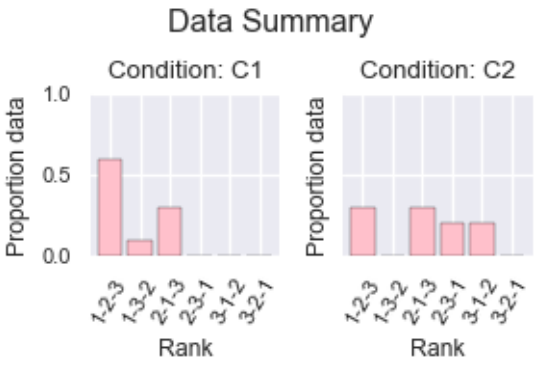

Figure 4: Propotion of subjects who chose each rank in conditions C1 and C2.

Aggregate ranking

The BTM provides us with an estimate of the aggregate rankings. However, rather than providing a single estimate of the aggregate ranking, Bayesian estimation provides us with a posterior (probability) distribution over the aggregate ranking. In other words, for every possible ranking, we assign a probability that it is the aggregate ranking. Figure 5 shows this for conditions C1 and C2. We can see that the posterior suggests that the aggregate ranking in C1 is 1-2-3 with high probability. In condition C2 the posterior assigns a larger probability to 2-1-3, with some probability mass over 1-2-3. This may be somewhat unintuitive given that 2-1-3 was not chosen more than 1-2-3. However, the aggregate ranking considers the entire distribution of rankings, which are more likely under aggregate ranking 2-1-3 than 1-2-3 . While we cannot be sure of the aggregate ranking in condition C2 we can still draw some useful inferences, specifically that the third item is always ranked last.

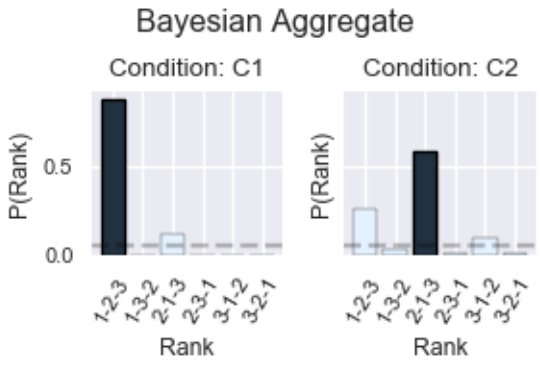

Figure 5: Posterior distribution over aggregate ranking in conditions C1 and C2.

Agreement between subjects

As well as an aggregate ranking, it is often very useful to know the extent to which participants agree with one another. It is possible that the posterior distribution may provide high certainty in the aggregate ranking despite there being considerable variability between participants (in the same way that with large sample sizes we can precisely estimate the mean of a normal distribution despite high variance). Thus quantifying the agreement between participants may be important for many applications. In the Thurstonian model this variability is represented by how much the latent distributions overlap (see Figure 3), which depends on the latent aggregate rank ($\boldsymbol{\mu}$) and all the subjects variance parameters (σ). It is therefore difficult to provide a single summary of agreement using the model parameters. However we can use a summary statistic to estimate the level of

agreement within our dataset, and we can also use the posterior distribution to estimate the values of the summary statistic that we expect to see in future datasets.

Here we use Kendall's W (also known as Kendall's coefficient of concordance; simply referred to here as W) as our measure of agreement. W is a non-parametric statistic which ranges from 0 (no agreement) to 1 (unanimous agreement). We calculate W for a dataset $\boldsymbol{Y}$ (where $\boldsymbol{Y}$ is an $J \times K$ matrix of rankings where J is the number of subjects and K is the number of items), as follows:

$$R_k = \sum_{j=1}^{J} Y_{i,j}\,, k \in [1, \dots, K]$$

$$S = Var(R)$$

$$W = \frac{12S}{J^2(K^3 - K)}$$

First we can calculate this for the dataset (red line in Figure 6). In condition C1 we can see that $W = 0.67$, indicating a reasonable degree of agreement. In condition C2 however $W = 0.13$ which suggests a small level of agreement.

In addition to calculating W for the dataset we can use the posterior distribution to estimate values of W that we are likely to see in future data sets when sampling from the same population. To do this we use the posterior predictive distribution (Gelman, 2014) which is a distribution over future (or replicated) datasets. To do this we simulate a dataset for every draw from the posterior distribution (obtained by the MCMC algorithm) and calculate W for each dataset. We plot this as the red histogram in Figure 6 and provide the mean value of the W from the posterior predictive distribution (W_{ppd}; black line in Figure 6). This captures both the uncertainty in the models parameter values as well as the expected sampling variance of W. We can see that for condition C1 $W_{ppd} = 0.75$ and the distribution suggests that we will see large values of W in future datasets. For condition C2 $W_{ppd} = 0.32$ and the distribution has most of its mass below values of 0.5, but with a tail that covers larger values. This suggests that while it is most likely that we will see low levels of agreement, there is quite large uncertainty in the expected values of W.

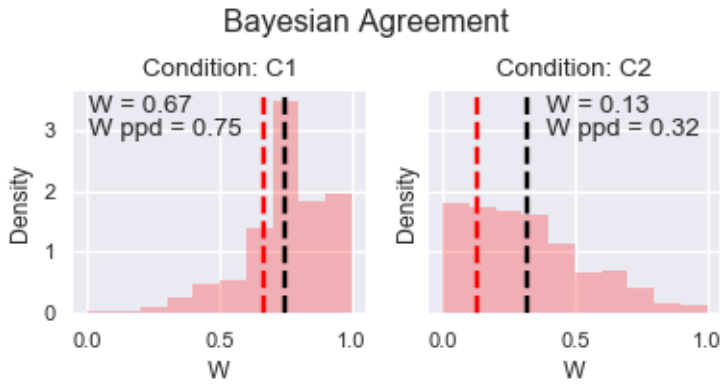

Figure 6: Agreement in conditions C1 and C2. The red line shows W for the data sample. The black line shows the mean of W for the posterior predictive distribution, while the red histogram shows the posterior predictive distribution over W.

Contrasts

In addition to identifying an aggregate ranking and the level of agreement between participants, when we have multiple conditions we may wish to test whether the aggregate ranking is the same in the two conditions. For example, a researcher may want to know whether people will provide the same rankings of a set of items when seeing them in virtual reality compared to the real world, as a test of the usefulness of the virtual reality method. To estimate such contrasts we can make use of distance metrics, which provide a useful measure of the difference between two rankings. Here we use Kendall's tau distance (τ), which is normalised between 0 and 1, with 0 indicating that the rankings are identical and 1 indicating that the rankings are opposite. As we are using Bayesian estimation we can calculate a probability distribution over the τ contrast between two conditions of interest.

This is shown in Figure 7, which shows that the τ distance between the aggregate rankings in C1 and C2 is most likely to be 0.33 (similar but not identical aggregate rankings), or 0.0 (identical aggregate rankings). The contrast takes into account the uncertainty in the aggregate rankings for each condition, as quantified by the posterior distribution. Thus rather than simply providing a point estimate for τ we are able to report full distributional information.

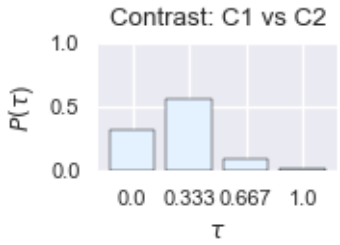

Figure 7: Bayesian contrast between condition C1 and C2.

Application to real datasets

Experiment overview

Now we will consider real world applications of the Thurstonian model to two datasets in which expert test drivers assessed vehicle prototypes. The aim of both experiments was to examine whether test drivers would rank vehicle prototypes in the same way as in the real world, if the testing was performed in a high fidelity driving simulator. The University of Leeds driving simulator (UoLDS) was used for testing, which consists of a large hexapod mounted on a rail system which can translate the hexapod both laterally and longitudinally. This motion system accelerates the test driver allowing motion cues to be provided. We were interested in how the size of the motion platform would affect the driver's rankings, as larger motion bases are able to accelerate for longer, potentially providing higher fidelity motion cues. Thus the test drivers completed the vehicle prototype evaluations in the real world and on a small, medium and large simulator motion base.

Study 1 - Stability control systems (SCS)

In the first study drivers assessed three stability control systems (SCS). SCS applies the brakes on the vehicle's wheels individually during losses of traction, in such a way as to try and keep the vehicle following the driver's intended path. During development, such systems need extensive testing and tuning by test drivers and engineers driving on low friction test tracks, often in remote northern locations. Thus there is growing interest in whether some of the driver assessments can be conducted in advanced driving simulators.

Eight test drivers drove a Jaguar XF on a winter test track in northern Sweden. They assessed three different configurations of the SCS system on a circular curve test track, and the drivers were not given any information on what configuration changes were being made. The drivers were asked to maintain a constant radius turn and complete the track as fast as possible. The drivers completed a

questionnaire in which they rated the SCS systems on a number of criteria. Here we only report the response to the first questionnaire item, in which they were asked to rate the "overall stability (how well the vehicle followed your intended path)". The drivers then completed the same task in the UoLDS, using a high fidelity simulation of the test track, vehicle dynamics and SCS systems.

Study 2 - Suspension configurations (SUSP)

In the second study nine test drivers drove a Land Rover prototype on a dry tarmac public road in the UK. The drivers assessed three configurations of the vehicle's suspension system, in which the height of the suspension was varied (high, medium and low), although drivers were not informed that the change in suspension configuration was a change in height. The drivers completed a questionnaire rating multiple aspects of the vehicle's ride qualities under the different suspension configurations. Here we report the first questionnaire item which asked drivers to rate the "primary ride of the vehicle: These are low frequency vertical movements of the vehicle. Evaluate large amplitude undulations that cause suspension movement over a moderate range up to the full range of the suspension". The drivers ranked the suspension configurations from constrained/ trapped to free/floaty.

Summary statistics

Figure 8 shows the proportion of participants who chose each possible ranking for each test track and motion base. The top row shows the data for the SCS study, while the second row is for the SUSP study. In the real world SCS testing the ranking 1-2-3 was the most commonly chosen. This pattern was also seen in the medium (Sim_M) and large and (Sim_L) conditions, while the pattern was less clear the small condition (Sim_S).

For the SUSP study the rankings were distributed far more uniformly in the real world than in the SCS study (bottom left panel). Similar levels of disagreement was seen in the Sim_S and Sim_M conditions. However, over 50% of drives selected 1-2-3 in the Sim_L condition, suggesting that there may have been a perceptible difference between the suspension configurations in this condition.

Figure 8: Proportion of subjects who chose each rank as a function of simulator configuration (columns) in the SCS (top row) and SUSP studies (bottom row).

Aggregate rankings

We fit a Thurstonian model separately for both the SCS and SUSP studies, allowing us to apply all of the analysis from the previous section to our real world datasets. Figure 9 shows the Bayesian estimates of the aggregate ranking for all the conditions in both the SCS (top row) and SUSP (bottom row) studies. The posterior distribution over the aggregate ranking revealed that there is reasonably high certainty that 1-2-3 is the aggregate rank in the SCS real condition (>90%), while the estimate in the SUSP real condition is less certain (60%) that the aggregate is 3-1-2. However, there was fairly high certainty that either 3-1-2 or 3-2-1 is the aggregate ranking.

In the SCS study the Thurstonian model suggested that either 1-2-3 or 2-3-1 was the aggregate ranking in the Sim_S and Sim_M condition, while 1-2-3 was almost certainly the aggregate ranking in the Sim_L condition. Thus in this case the large motion base appeared to give the closest agreement to the real world in terms of the aggregate ranking. For the SUSP study 1-2-3 was always the most probable aggregate ranking in the simulator conditions, but the posterior only had high certainty in this for the Sim_L condition and this did not appear to match the aggregate ranking in the real world.

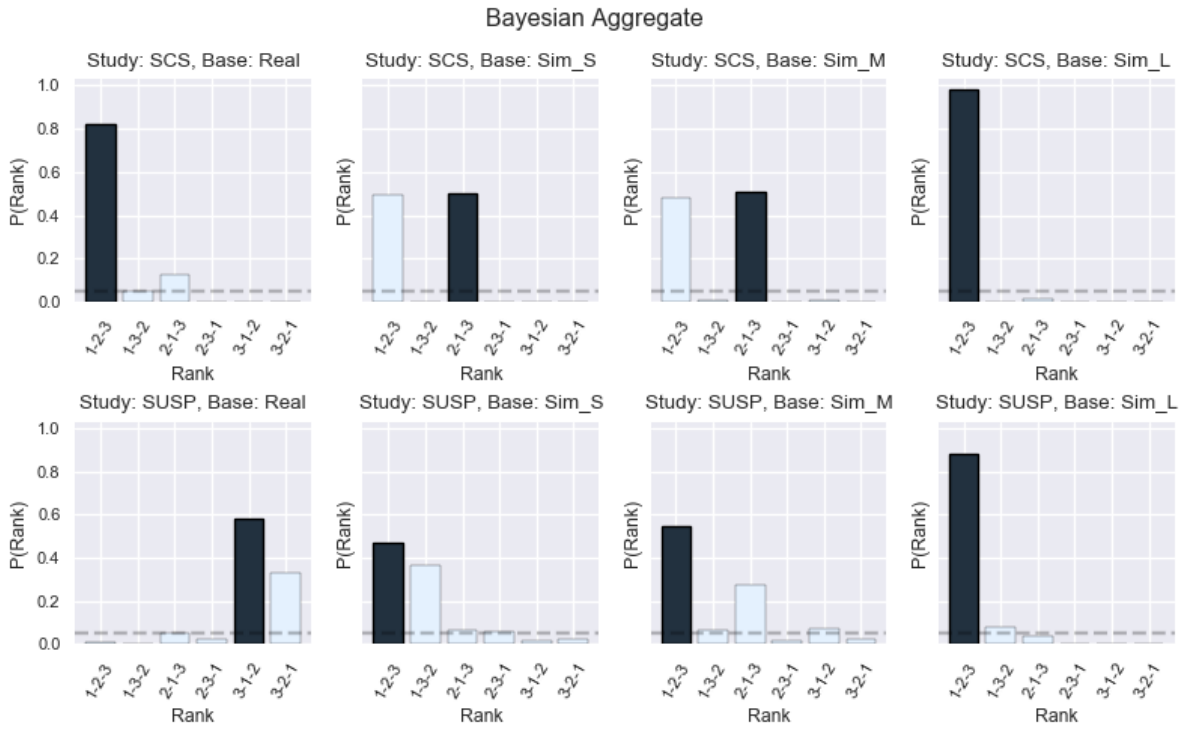

Figure 9: Bayesian aggregate rankings as a function of simulator configuration (columns) for both the SCS (top row) and SUSP studies (bottom row).

Agreement

Figure 10 shows the level of agreement between drivers for both the SCS and SUSP studies. In the SCS study we can see that in the real world $W = 0.33$ in the dataset, while the posterior predictive distribution (red histogram) spanned a large range of values. This suggests that there is a high level of uncertainty in values of W in future datasets. A similar pattern was seen in the Sim_S and Sim_M conditions, suggesting that agreement was similar to in the real world. However, there much larger levels of agreement were seen in the Sim_L condition ($W = 0.63$) and the posterior predictive distribution suggests that we should expect to see large levels of agreement in future datasets. For the SUSP study values of W were very low across all conditions, except the Sim_L where there seemed to be some level of agreement. The posterior predictive distributions were very broad and suggest high uncertainty in the values of W we expect to observe in future datasets.

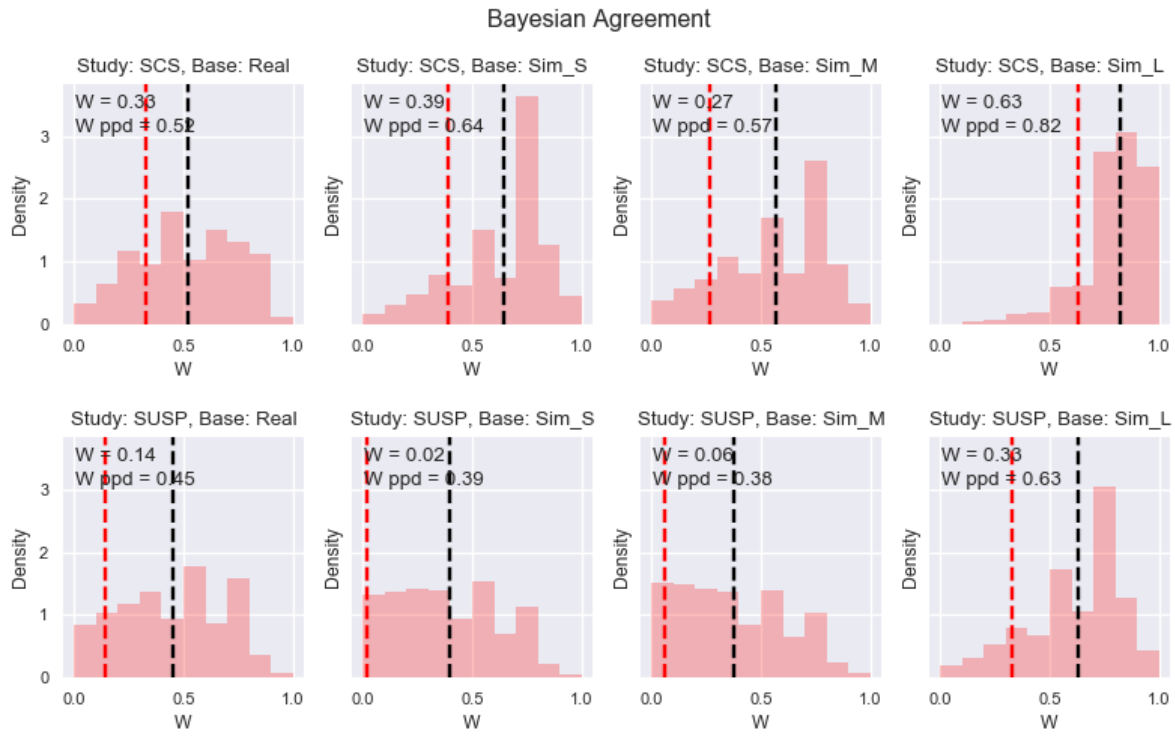

Figure 10: Agreement as a function of simulator configuration (columns) for both the SCS (top row) and SUSP studies (bottom row). The red lines shows W for the data samples. The black line shows the mean of W for the posterior predictive distributions, while the red histograms shows the posterior predictive distribution over W.

Contrasts

Finally, for both the SCS and SUSP studies we perform contrasts between the real world condition and each of the simulator conditions to assess how close the aggregate ranking was in the simulator to the real world. It is clear from Figure 11 that for the SCS study the contrasts suggested that for the the Sim_S and Sim_M τ was either 0, or 0.333 (which in the K=3 case corresponds to it being one permutation away from the real world condition). For the Sim_L however there was fairly high certainty that τ was 0. Conversely in the SUSP condition the contrasts suggested that the simulator conditions had a τ of at least 0.667 (2 permutations away) suggesting that it was very unlikely that the aggregate rankings we the same in the simulator as in the real world.

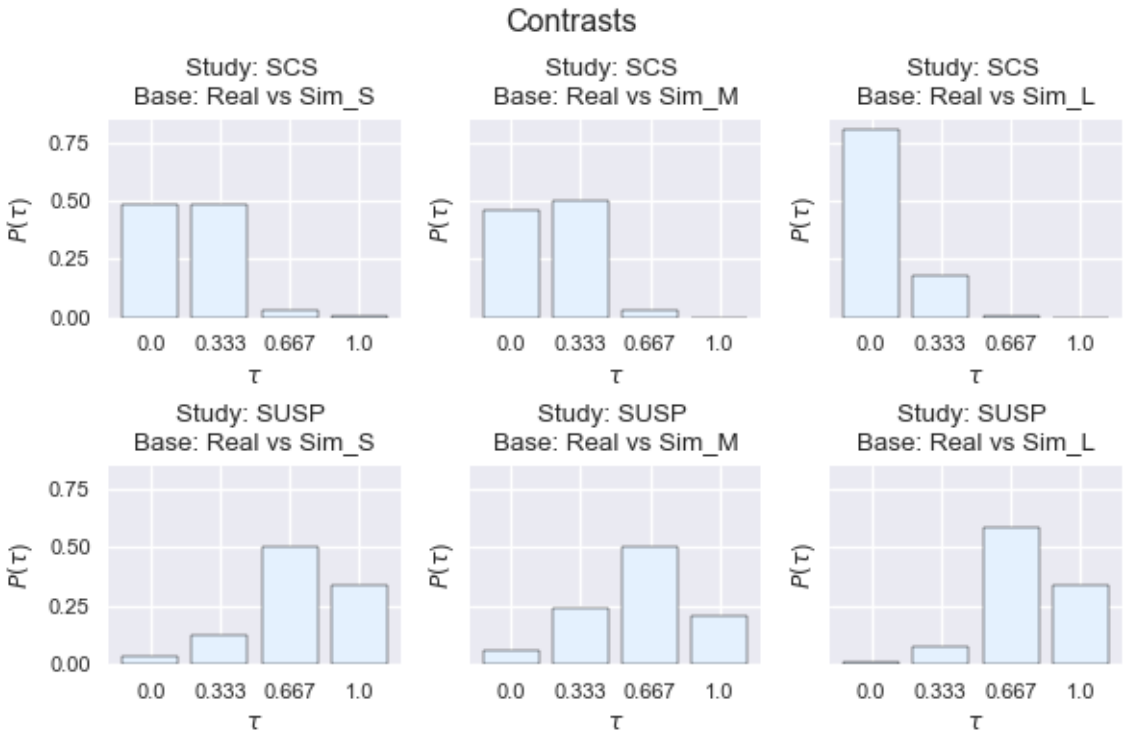

Figure 11: Contrasts between the aggregate ranking in the real world and each of the simulator configurations for the SCS study (top row) and SUSP study (bottom row)

SCS and SUSP conclusions

The Thurstonian analyses provided above allow us to draw a number of conclusions regarding our datasets. For the SCS study it was clear that 1-2-3 was the aggregate ranking in the real world, but the contrasts revealed that only the Sim_L condition clearly reproduced this. While the Sim_S and Sim_M configurations may have had the same ranking (there was roughly 50% probability of sharing the same aggregate ranking as in the real world), this level of certainty suggests that different design decisions may have been made from the simulator testing compared to the real world. In addition, the level of agreement between subjects in the real world was fairly low suggesting that test drivers found it quite difficult to reliably assess the SCS systems. While similar levels of agreement were seen in Sim_S and Sim_M, the Sim_L configuration showed a much higher level of agreement than the real world. This suggests that this simulator configuration may lead to an overestimate of the extent to which test drivers will agree in the real world. Thus it appears that while the simulators were able to capture aspects of the real world testing, no configuration was able to fully reproduce the pattern of results seen in the real world.

For the SUSP study the Thurstonian analysis suggested that the simulator conditions did not reproduce the real world testing results well. From the contrasts it was clear that the aggregate rankings in the simulator conditions were almost certainly different than in the real world, and the level of agreement between subjects was extremely low (trending towards zero). The low levels of agreement in the real world suggest that the test drivers were unable to reliably discriminate between the suspension configurations. This was also found to be the case in the Sim_S and Sim_M configurations. However, agreement was higher in the Sim_L configuration, suggesting that we may overestimate the level of agreement between subjects compared to the real world.

Conclusion

Here we provided an introductory tutorial on the use of Thurstonian models to analyse ranking data. We illustrated what information can be gleaned from BTMs and how we can use Bayesian estimation to provide probabilistic inferences. Despite Thurstonian models existing for decades, they have not seen widespread use. We see two major barriers which have limited their application. The first stem from difficulties in model fitting. These are largely eliminated by MCMC methods. While we are not the first to demonstrate that MCMC methods can be applied to Thurstonian models, we provide the first implementation which makes use of Hamiltonian Monte Carlo (Betancourt, 2017) implemented in Stan (Carpenter et al., 2017). The second potential barrier comes from a lack of simple statistical tools which allow users to fit Thurstonian models without requiring substantial knowledge of implementing models in probabilistic programming languages. We hope that by providing a high level open-source code library will encourage the use of BTMs.

We have introduced how BTMs can be used to analyse subjective ranking data. Here we provided examples in with individuals ranked 3 items, and comparisons were made between conditions. However, the model can be applied to datasets in which any number of items are ranked, and with any number of continuous or categorical covariates. PyThurstonian allows users to easily do this using a simple model specification syntax. Finally, we introduced a number of novel approaches to drawing inferences from Thurstonian model parameters. Specifically, we demonstrated how established distance metrics can be applied to Bayesian model fits to provide an intuitive understanding of the model parameters, allowing the study of aggregate rankings, contrasts between different conditions, the level of agreement between individuals.

References

Betancourt, M. (2017). A Conceptual Introduction to Hamiltonian Monte Carlo. Retrieved from http://arxiv.org/abs/1701.02434

Boobier, S., Osbourn, A., & Mitchell, J. B. O. (2017). Can human experts predict solubility better than computers? *Journal of Cheminformatics*, *9*(1), 1–14. https://doi.org/10.1186/s13321-017-0250-y

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., … Riddell, A. (2017). Stan : A Probabilistic Programming Language. *Journal of Statistical Software*, *76*(1). https://doi.org/10.18637/jss.v076.i01

Galton, F. (1907). Vox Populi. *Nature*, *75*(1949), 450–451. https://doi.org/10.1038/075450a0

Gelman, A. (2014). *Bayesian Data Analysis* (3rd ed.). Boca Raton: Taylor & Francis Group.

Hastings, W. K. (1970). Monte Carlo Sampling Methods Using Markov Chains and Their Applications Author (s): W . K . Hastings Published by : Oxford University Press on behalf of Biometrika Trust Stable URL : http://www.jstor.org/stable/2334940 Accessed : 03-05-. *Biometrika*, *57*(1), 97–109.

Hoffman, M., & Gelman, A. (2014). The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, *15*, 30.

Johnson, T. R., & Kuhn, K. M. (2013). Bayesian Thurstonian models for ranking data using JAGS. *Behavior Research Methods*, *45*(3), 857–872. https://doi.org/10.3758/s13428-012-0300-3

Kattan, M. W., O'Rourke, C., Yu, C., & Chagin, K. (2016). The Wisdom of Crowds of Doctors. *Medical Decision Making*, *36*(4), 536–540. https://doi.org/10.1177/0272989X15581615

Kruschke, J. K. (2010). Bayesian data analysis. *Wiley Interdisciplinary Reviews: Cognitive Science*, *1*(5), 658–676. https://doi.org/10.1002/wcs.72

Kruschke, J. K. (2015). *Doing Bayesian Data Analysis* (2nd ed.). London: Elsevier.

Kruschke, J. K., & Liddell, T. M. (2017a). The Bayesian New Statistics: Hypothesis testing, estimation, meta-analysis, and power analysis from a Bayesian perspective. *Psychonomic Bulletin & Review*, 1–28. https://doi.org/10.3758/s13423-016-1221-4

Kruschke, J. K., & Liddell, T. M. (2017b). The Bayesian New Statistics: Hypothesis testing, estimation,

meta-analysis, and power analysis from a Bayesian perspective. *Psychonomic Bulletin & Review*, 1–28. https://doi.org/10.3758/s13423-016-1221-4

Lee, M. D., Steyvers, M., & Miller, B. (2014). A cognitive model for aggregating people's rankings. *PLoS ONE*, *9*(5). https://doi.org/10.1371/journal.pone.0096431

McElreath, R. (2015). *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Chapman and Hall/CRC.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, *21*(6), 1087–1092. https://doi.org/10.1063/1.1699114

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 113–162. https://doi.org/doi:10.1201/b10905-6

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing*, (Dsc), 1–10. https://doi.org/ISSN 1609-395X

Thurstone, L. L. (1931). Rank Order As A Psychophysical Method. *Journal of Experimental Psychology: General*, *114*(3).

Yao, G., & Böckenholt, U. (1999). Bayesian estimation of Thurstonian ranking models based on the Gibbs sampler. *British Journal of Mathematical and Statistical Psychology*, *52*(1), 79–92. https://doi.org/10.1348/000711099158973