



Do organic search results help or hurt sponsored search performance?

Evaluating the probabilistic model
by Agarwal et al. 2015

**Lisa Gotzian
Erik Schau
Ian Shulman**

A paper presented for the course
Probabilistic Modelling
instructed by
Prof. Dr. Burkhardt Funk
in the Master's Program
Management & Data Science

August 26, 2018

Contents

1	Introduction	2
1.1	The findings by Agarwal et al.	2
1.2	The framework	2
2	Simulating the data and the model	7
2.1	The data	7
2.2	The Distributions	8
2.2.1	Poisson-Distributed Variables	8
2.2.2	Binomial-Distributed Variables	10
2.2.3	Normally Distributed Variables	10
2.3	Simulating the parameters of the model	12
2.3.1	The θ^k 's	12
2.3.2	The error terms ϵ_{kt}^θ of the linear model	14
2.3.3	The final step towards Λ_{kt}^{CTR}	15
2.4	Simulation and Summary Statistics Comparison	15
3	Estimating the model using a Gibbs Sampler	18
4	Evaluation of the model	25
	Bibliography	27
5	Eidesstattliche Erklärung	28

1 Introduction

Bayesian hierarchical models have gained quite some importance in the marketing segment during the last 15 years (Rossi et al., 2006; Ghose and Yang, 2009; Agarwal et al., 2015). They are now being employed to analyze click-behavior and orders to evaluate the effectiveness of advertisements in search engines. Eventually, the factors that lead to this behavior are modeled and given a certain importance.

Among factors such as sponsored competition, an ad's position relative to other ads, or the quality score of a given keyword, organic search results have not extensively been taken into account. Following the approach of Agarwal et al. 2015., this paper rebuilds and comments on a bayesian hierarchical model that assesses the role of organic search results in particular.

The paper is broken up into the following sections: first, a brief overview of the research performed by Agarwal et al in their paper "Do organic search results help or hurt sponsored search performance?" (Agarwal et al., 2015), as well as an explanation of their development, and our interpretation of their model; then, a discussion on the data used for the model, our attempts to simulate the data based on the provided (and some noticeably absent) summary statistics; further, there is an outline of the model estimation made using MCMC sampling and related packages in R; finally, an evaluation of the model describing challenges faced in extrapolating Agarwal et al's model from the paper into a workable format.

1.1 The findings by Agarwal et al.

The authors explored how organic competition from organic listings of direct competitors impacts click and conversion performance in sponsored search, using the data collected during an advertisement campaign of a pet food online retailer. The paper's findings link an increase in organic competition with a decrease in the click performance of sponsored advertisements; nevertheless, organic competition positively impacts the conversion performance of sponsored ads, which, in turn, increases revenue. Their results state that the negative impact of organic competition on click performance appeared to be higher than that of sponsored competition.

1.2 The framework

The model targets both click-through rate (CTR) and conversion rate (CONV) as dependent variables and assumes the data to be explained by a logit function of the latent utility of clicking or conversion, respectively. All parameters denoted with kt have a value for each of 36 keywords (k) and 40 days (t), while those with k are influenced by keywords only, not time. This section explores the model in more depth to clarify details that have not explicitly been defined by the authors.

$$\begin{aligned}\Lambda_{kt}^{CTR} &= \frac{\exp(U_{kt}^{CTR})}{1+\exp(U_{kt}^{CTR})} \quad \text{or} \quad \frac{\text{Clicks}_{kt}}{\text{Impressions}_{kt}} \\ \Lambda_{kt}^{CONV} &= \frac{\exp(U_{kt}^{CONV})}{1+\exp(U_{kt}^{CONV})} \quad \text{or} \quad \frac{\text{Orders}_{kt}}{\text{Impressions}_{kt}}\end{aligned}\tag{1}$$

The latent utilities of clicking and buying are based on ad position, organic search results and the quality score by Google (LQscore) multiplied by their importance, denoted by β and θ .

$$\begin{aligned}U_{kt}^{CTR} = & \theta_0^k + \theta_1^k AdPos_{kt} + \theta_2^k OrganicComp_{kt} + \theta_3^k SponsoredComp_{kt} \\ & + \theta_4^k Organic_{kt} + \theta_5^k LQScore_{kt} + \theta_{Time}^k Time_{kt} + \epsilon_{kt}^\theta \\ U_{kt}^{CONV} = & \beta_0^k + \beta_1^k AdPos_{kt} + \beta_2^k OrganicComp_{kt} + \beta_3^k SponsoredComp_{kt} \\ & + \beta_4^k Organic_{kt} + \beta_5^k LQScore_{kt} + \beta_{Time}^k Time_{kt} + \epsilon_{kt}^\beta\end{aligned}\tag{2}$$

The model furthermore handles $AdPos_{kt}$ as well as $OrganicComp_{kt}$ as endogenous variables. They depend on another set of biases γ and α with only selected parts of the data.¹

$$\begin{aligned}\ln(AdPos_{kt}) &= \gamma_0^k + \gamma_1^k \ln(bid)_{kt} + \gamma_2^k LQScore_{kt} + \gamma_{Time}^k Time_{kt} + \epsilon_{kt}^\gamma \\ OrganicComp_{kt} &= \alpha_0^k + \alpha_1^k IVOrganic_{kt} + \alpha_{Time}^k Time_{kt} + \epsilon_{kt}^\alpha\end{aligned}\tag{3}$$

All parameters θ^k , β^k , γ^k and α^k are drawn from a multivariate normal distribution with a mean of $\Delta^\theta z_k$ and a sigma matrix V^θ for θ . The same applies to the other variables like β .²

$$\begin{aligned}\theta^k &= \Delta^\theta z_k + u_k^\theta & \beta^k &= \Delta^\beta z_k + u_k^\beta \\ u_k^\theta &\sim MVN(0, V^\theta) & u_k^\beta &\sim MVN(0, V^\beta) \\ \gamma^k &= \Delta^\gamma z_k + u_k^\gamma & \alpha^k &= \Delta^\alpha z_k + u_k^\alpha \\ u_k^\gamma &\sim MVN(0, V^\gamma) & u_k^\alpha &\sim MVN(0, V^\alpha)\end{aligned}\tag{4}$$

Four θ^k 's depend on the keyword, three θ 's are given as one value. For β , the 4/3 ratio remains, for γ , this becomes 2/2 and alpha's ratio is 2/1.³

In the following, the model is described for the click-through rate and its parameters θ_k and θ which can be mirrored for β , γ and α respectively.

First, z_k is a 2xk matrix from the data, denoting brand and specificity of the

¹It should be noted that the appendix consequently uses the natural log of Organic Competition instead of the value itself. Thus, it is likely that this formulation misses $\ln()$ given it has been used for $AdPosition$.

² N has been adjusted to the multivariate distribution (MVN).

³While the formulation in equation 4 seems to only cover the parameters denoted with k, the appendix suggests a normal distribution for those without k as well, with a mean of 0 instead of $\Delta^\theta z_k$ and a standard deviation of 100.

given keyword. Consequently, Δ^θ is a 4x2 matrix and represents the relationship of the four θ^k s with a keyword's brand and specificity. $\Delta^\theta z_k$ multiplied then is the mean of the parameters' multivariate normal distribution.

$$\theta^k = \begin{bmatrix} k=1 \\ \theta_0^{(1)} \\ \dots \\ \theta_3^{(1)} \end{bmatrix} \begin{bmatrix} \dots \end{bmatrix} \begin{bmatrix} k=36 \\ \theta_0^{(K)} \\ \dots \\ \theta_3^{(K)} \end{bmatrix} \begin{array}{l} \theta_0 \text{ for const} \\ \theta_1 \text{ for AdPos} \\ \theta_2 \text{ for OrganicComp} \\ \theta_3 \text{ for SponsoredComp} \end{array} \quad (5)$$

e.g. for $k = 5$:⁴

$$\theta^{(5)} = \begin{bmatrix} \theta_0 xBrand & \theta_0 xSpec \\ \theta_1 xBrand & \theta_1 xSpec \\ \theta_2 xBrand & \theta_2 xSpec \\ \theta_3 xBrand & \theta_3 xSpec \end{bmatrix} \cdot \begin{bmatrix} Brand_5 \\ Specificity_5 \end{bmatrix} + \begin{bmatrix} u_5^{(0)} \\ u_5^{(1)} \\ u_5^{(2)} \\ u_5^{(3)} \end{bmatrix} \quad (6)$$

$$\Delta^\theta \qquad \qquad \qquad z_5 \qquad \qquad \qquad u_5^\theta$$

To estimate the sigma matrix for the multivariate distribution of each parameter θ^k , the model proposes an intermediate step via a u vector. The u vector can be seen as the unobserved heterogeneity of each keyword. It is drawn from a multivariate normal distribution with a mean of 0 and the covariance matrix V^θ . V^θ in return is Inverse-Wishart distributed. The IW distribution is associated as a conjugate prior for covariance matrices of multivariate normal distributions (Rossi et al., 2006, p. 28ff.). $S = 10I$ is the identity matrix, scaled with 10.

$$\begin{aligned} u_k^\theta &\sim MVN(0, V^\theta) \\ V^\theta &\sim IW(\nu + N, \sum_{k=1}^K (\theta^k - \Delta^\theta z_k)^T (\theta^k - \Delta^\theta z_k) + S) \end{aligned}$$

where $N = \text{No of keywords}$, $\nu = 10$, $S = 10I$

$$V^{\theta-1} = \begin{bmatrix} var[\theta_0] & cov[\theta_0, \theta_1] & cov[\theta_0, \theta_2] & cov[\theta_0, \theta_3] \\ cov[\theta_1, \theta_0] & var[\theta_1] & cov[\theta_1, \theta_2] & cov[\theta_1, \theta_3] \\ cov[\theta_2, \theta_0] & cov[\theta_2, \theta_1] & var[\theta_2] & cov[\theta_2, \theta_3] \\ cov[\theta_3, \theta_0] & cov[\theta_3, \theta_1] & cov[\theta_3, \theta_2] & var[\theta_3] \end{bmatrix} \quad (7)$$

The model is endogenous, meaning that the errors ϵ_{kt} for U_{kt}^{CTR} , U_{kt}^{CONV} , $AdPos_{kt}$ and $OrganicComp_{kt}$ from equations 2 and 3 are correlated and have thus been drawn from a multivariate normal distribution as well. Like the other covariance matrices V , the covariance matrix Ω is estimated by the Inverse Wishart distribution. All errors then form a vector of four kxt matrices, hence in the example, ϵ^θ is a kxt matrix. "As the position of the advertisement as well as organic competition are endogenous, the unobservable time varying

⁴Table 4 on p. 41 either missed the top two rows of Δ^θ or doesn't mention that these correspond to 0. They have been interpreted as 0 for the simulation. The problem arose in a similar manner for the other parameters as well.

keyword attributes for the equations representing consumer decisions will be correlated with error terms for the equations representing position and organic competition.”(Agarwal et al., 2015, p. 26)

$$\begin{bmatrix} \epsilon_{kt}^\beta \\ \epsilon_{kt}^\theta \\ \epsilon_{kt}^\gamma \\ \epsilon_{kt}^\alpha \end{bmatrix} \sim MVN(0, \Omega) \quad (8)$$

$$\Omega \sim IW(\nu_\Omega + N, \sum_{k=1}^K \sum_{t=1}^T Y_{kt}^T Y_{kt} + S_\Omega)$$

where N = No of keywords, $\nu_\Omega = 10$, $S_\Omega = 10I$

$$\Omega^{-1} = \begin{bmatrix} var[\epsilon^\beta] & cov[\epsilon^\beta, \epsilon^\theta] & cov[\epsilon^\beta, \epsilon^\gamma] & cov[\epsilon^\beta, \epsilon^\alpha] \\ cov[\epsilon^\theta, \epsilon^\beta] & var[\epsilon^\theta] & cov[\epsilon^\theta, \epsilon^\gamma] & cov[\epsilon^\theta, \epsilon^\alpha] \\ cov[\epsilon^\gamma, \epsilon^\beta] & cov[\epsilon^\gamma, \epsilon^\theta] & var[\epsilon^\gamma] & cov[\epsilon^\gamma, \epsilon^\alpha] \\ cov[\epsilon^\alpha, \epsilon^\beta] & cov[\epsilon^\alpha, \epsilon^\theta] & cov[\epsilon^\alpha, \epsilon^\gamma] & var[\epsilon^\alpha] \end{bmatrix} \quad (9)$$

In equation 9, Y_{kt} refers to 4 kxt matrices. It is calculated by subtracting all parameters times the data (essentially all terms of each linear model except for the error) from the current estimate (Agarwal et al., 2015, appendix p. 3). To summarize the idea, the parameter θ from the CTR example, as well as β , α and γ act as ‘weights’, defining the role each factor plays for CTR, for example. These are precisely the unknown parameters the model has to define, see figure 1. Taking the errors and covariances into account, the overall triangular system then has the following dependencies:

$$\begin{aligned} U_{kt}^{CTR} &= f(AdPos_{kt}, OrganicComp_{kt}, X1, \epsilon_{kt}^\theta) \\ U_{kt}^{CONV} &= f(AdPos_{kt}, OrganicComp_{kt}, X2, \epsilon_{kt}^\beta) \\ AdPos_{kt} &= f(X3, \epsilon_{kt}^\gamma) \\ OrgComp_{kt} &= f(X4, \epsilon_{kt}^\alpha) \end{aligned} \quad (10)$$

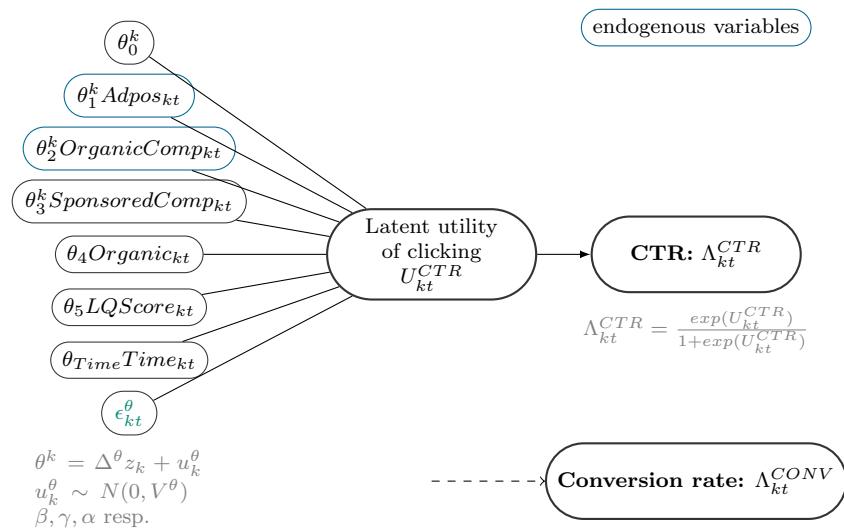


Figure 1: The full model, shown for CTR. Conversion rate and the two endogenous variables have a similar setup and are therefore only hinted at in this graph.

2 Simulating the data and the model

2.1 The data

The field experiment originally collected data on 1,440 daily impressions, clicks and orders for 36 keywords over a 40-day period from June 2009 to July 2009.⁵ To follow the model, data has been simulated based on the keyword performance summary statistics (Agarwal et al., 2015, table 2 on p. 40). Figure 2 gives an overview to the distributions assumed.

The code below shows the simulation in R. With the help of the package `rtruncnorm`, the max and min values from figure 2 could be taken into account as well.⁶

```

1 library(rtruncnorm)
2 ## instantiate the values to estimate
3 num_keywords <- 36
4 num_days <- 40
5 num_observations <- 1440
6
7 # Poisson impressions
8 impressions <- matrix(round(rpois(n=num_observations, lambda = rep
9     (72.8, num_observations)), 0), nrow=num_keywords, ncol=num_days
10    )
11 impressions [impressions > 1666] <- 1666
12 impressions [impressions < 1] <- 1
13
14 # [removed clicks and orders for this example as they were poisson-
15 # distributed as well]
16
17 # Normally distributed data with organic being binomial distributed
18 organic <- matrix(rbinom(n=num_observations, size = 1, p=0.15),
19     nrow=num_keywords, ncol=num_days)
20 iv_organic <- matrix(round(rtruncnorm(n=num_observations, a=0.1, b
21     =2.34, mean=0.42, sd=0.23 ),2), nrow=num_keywords, ncol=num_days
22    )
23 # [removed sponsored_comp, iv_sponsored and lq_score and bid as
24 # they are normally distributed and constructed the same way]
25
26 # Binomial distributions of brand and specificity
27 brand <- matrix(rbinom(n=num_keywords, size = 1, p=0.6),
28     nrow=num_keywords, ncol=1)
29
30 specificity <- matrix(rbinom(n=num_keywords, size=1, p=0.4),
31     nrow=num_keywords, ncol=1)
```

Listing 1: Simulating the data

⁵Though distinguishing between the biases of the factors and the data itself may seem straight forward (like θ_1^k times $Adpos_{kt}$), the authors didn't clearly allude in their estimations to either the Greek letters (the biases) or the roman words (the data). Interpreting the exact model with its parameters has thus been a challenge. The group has arrived at the conclusion to treat table 2 on page 40 as the data and tables 4, 5, 7 and 8 on pages 41f. as mean values for the parameter estimates.

⁶The full code can be found in the appendix.

Table 2: Keyword Performance Summary Statistics

Variable	Mean	St. Dev.	Min	Max
Impressions	72.8	159	1	1666
Clicks	1.1	2.2	0	24
Orders	0.03	0.2	0	3
AdPos	3.47	1.7	1	9.78
Organic	0.15	0.36	0	1
Organic_Comp normalized by position and Alexa Rank	0.78	0.54	0	3.08
Organic_Comp normalized by position	0.82	0.53	0	2.69
Organic_Comp normalized by Alexa Rank	1.7	1.52	0	10.1
IVOrganic	0.42	0.23	0.1	2.34
Sponsored_Comp normalized by position and Alexa Rank	1.79	0.89	0	5.59
Sponsored_Comp normalized by position	1.82	0.95	0	4.43
Sponsored_Comp normalized by Alexa Rank	3.28	2.1	0	11.4
IVSponsored	1.49	1	0.13	5.19
LQScore	8	1.5	6	10
Brand Specificity	0.6	0.5	0	1
Bid	0.5	0.3	0.08	2

Figure 2: The summary statistics for the original data, including its interpreted distributions.⁷

2.2 The Distributions

As shown in figure 2, the simulated variables follow a variety of distributions. Since the authors do not explicitly state which type of distribution each data item follows, we have assumed the most likely distribution based on the values provided in the summary statistics table. To this end, we simulated a total of 13 variables (we only focused on Sponsored Competition normalized by both position and Alexa Rank), the variable counts for each type of distribution being as follows:

- 3 Poission-distributed
- 3 Binomial-distributed
- 7 Normally distributed

The density distributions for each of these variable sets are described in figure 3.

2.2.1 Poisson-Distributed Variables

Although it may appear to be normally distributed, the distribution for Impressions is, in fact, a Poisson distribution. To be in line with the min and max values established in the summary statistics table, any values below 1 and above 1666 (min and max, respectively), if any, have been replaced by the min and max values themselves. The Clicks distribution peaks close to 1 and gently

⁷The endogenous variables $AdPos_{kt}$ and $OrganicComp_{kt}$ were listed in this list, though contradicting the notion of "data".

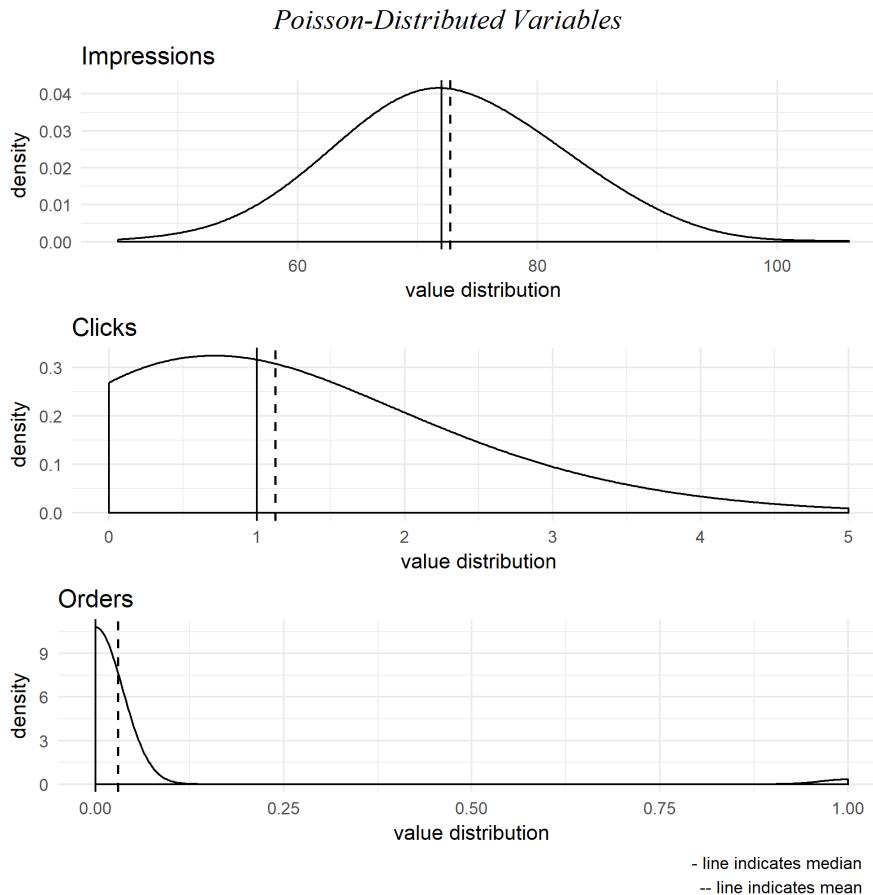


Figure 3: Distributions for the three Poisson-distributed variables

slopes downward to 5, which logically makes sense as users would most often click 0 to 1 times on any particular ad in one user session. As with Impressions, Clicks has established min and max values - 0 and 24 - and any values outside of this range were replaced with the min/max values as appropriate⁸.

Finally, the Orders distribution is bimodal, with a large peak at 0 and a much smaller peak at 1. This distribution is a bit misleading as a user can only either place an order or not, hence the 0-1 peaks. One may consider this more a binomial variable, but we chose to model it as Poisson because it is an event that occurs potentially at random and across an unknown time interval.

The Orders variable also has minimum- and maximum-allowed values, these being 0 and 3, respectively (as with clicks, a user cannot make a negative order).

⁸It is not possible, at least as far as we are aware, for a user to make -1 clicks!

2.2.2 Binomial-Distributed Variables

The three binomial variables are relatively simple in their usage, in that they represent whether an event occurred or not. Each of these variables act as a sort of on/off switch for the inclusion of the parameters in the simulation for a particular keyword at a particular time. As mentioned in the paper, Organic is

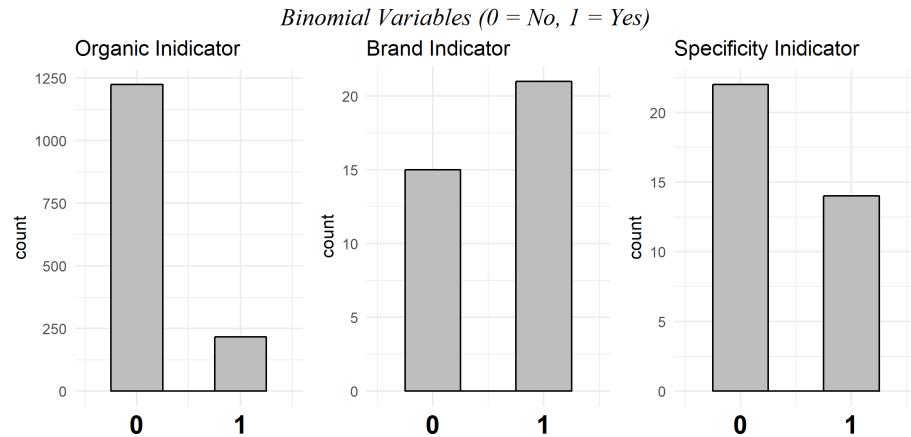


Figure 4: Distributions for the three Binomial-distributed variables

"a dummy variable which is set to one if the advertiser's organic listing appears on the first page, and is set to zero otherwise"(Agarwal et al., 2015, p. 20). Similarly, Brand is an indicator variable which is set to 1 if a known brand name is in the keyphrase used in the experiment, and is 0 if a brand name is not included.

Specificity is a bit more involved - this is actually an indicator of which level in the resulting website's site hierarchy the user reaches when routed through the search engine (i.e., how close to the actual searched-for product does the user get when they click on the link in the search engine?). For example, the specificity for a generic keyword - the authors mention 'pet food' - is 0, while a more specific keyword, such as 'dog food', is scored as a 1 (Agarwal et al., 2015). The paper mentions the possibility for values greater than 1, but the maximum value for specificity given in the summary statistics table is 1, so we chose to model this variable as binomial.

2.2.3 Normally Distributed Variables

As might be expected, the predominance of our simulated variables follow a normal distribution.

It is not necessary to get into the specifics of each variable, save it to point out a few noticeable oddities about the graphs. As can be seen in figure 5, both the mean and median are slightly right-of-center for all of the variables simulated.

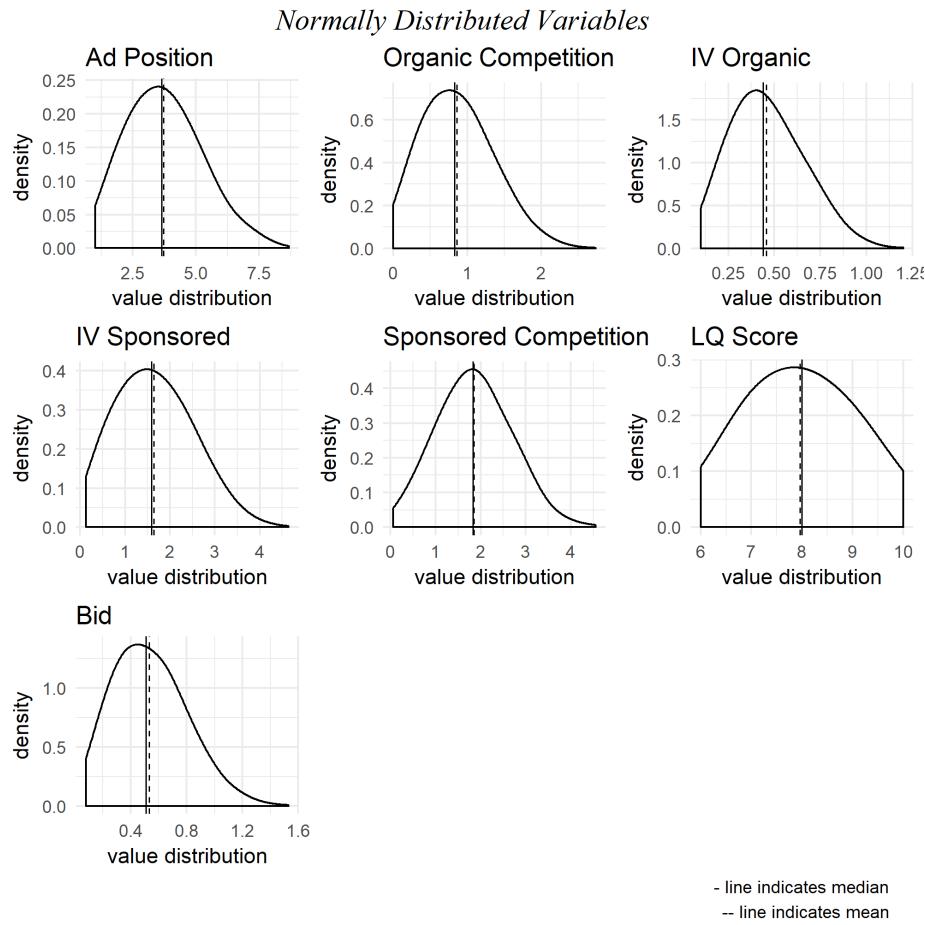


Figure 5: Distributions for the seven Normally distributed variables

This is due to the fact that all of the normal distributions are *truncated* normal distributions. for all but the LQ score, the minimum value given for each variable in the summary statistics table is 0; that is, the simulation would have produced values less than 0 had the minimum value in the code used not been set at 0. LQ score is even more truncated, with limits placed on its left and right tails at 6 and 10, respectively (6 being the minimum score found in the data statistics, and 10 being the maximum value allowed in the LQ scoring system).

Table 4: Estimates for the CTR

	Cumulative Competition normalized by position and Alexa rank	Cumulative Competition normalized by position	Cumulative Competition Normalized by Alexa Rank
Const	-4.22 (0.55)***	-4.35 (0.63)***	-4.84 (0.32)***
AdPos	-1.42 (0.18)***	-1.67 (0.16)***	-1.06 (0.21)***
Organic_Comp	-0.41 (0.15)***	-0.27 (0.14)**	-0.23 (0.11)**
Sponsored_Comp	-0.17 (0.12)	-0.01 (0.15)	-0.08 (0.08)
Organic	-0.2 (0.2)	-0.16 (0.1)	-0.2 (0.17)
LQScore	0.3 (0.06)***	0.32 (0.05)***	0.34 (0.03)***
Time	-0.01 (0.0)***	-0.011 (0.002)***	-0.013 (0.002)***
Brand	0.25 (0.51)	-0.17 (0.47)	0.24 (0.39)
Specificity	-0.51 (0.65)	-0.45 (0.66)	0.35 (0.64)
Organic_Comp x Brand	-0.09 (0.36)	0.14 (0.31)	0.02 (0.23)
Sponsored_Comp x Brand	-0.01 (0.29)	0.12 (0.27)	-0.05 (0.19)
Organic_Comp x Specificity	0.64 (0.42)	0.69 (0.37)*	0.13 (0.28)
Sponsored_Comp x Specificity	0.55 (0.37)	0.42 (0.36)	0.08 (0.24)

*, **, *** Statistically significant at 10%, 5%, and 1% respectively

Figure 6: The estimates for CTR from p. 39. The authors' estimates for the θ^k s in dark blue are only used for comparison. The values marked in dark grey served as constants for θ_4 , θ_5 and θ_{Time} . The light grey box refers to the Δ^θ matrix in the code, denoted as `theta_delta`⁹.

2.3 Simulating the parameters of the model

2.3.1 The θ^k s

After sampling the data, the hierarchical model's parameters needed to be estimated. Figure 6 depicts how the data given was interpreted as not all values relate to a clearly identifiable part in the introduced model.

Essentially, the parameters from the linear model dependent on k have been drawn from a multivariate distribution with a mean $\Delta^\theta z_k$ specifying its relationship with the keyword specifics and a sigma matrix V^θ . While $\Delta^\theta z_k$ has been taken from figure 6, V^θ was sampled from an Inverse Wishart distribution with 4 degrees of freedom and an identity matrix.¹⁰ In a sense, V^θ serves as a matrix for the error within the estimates of one parameter, so how the different factors come into play for the CTR. The code uses `theta_delta` as Δ^θ , `VTheta` as V^θ and `random_thetas` as the 4 θ^k s.

⁹"Brand" and "specificity" was interpreted as $\theta_0 \times$ brand or specificity. As θ_1 , the bias for $AdPos_{kt}$, is missing it is assumed to be 0. Though Δ^θ is constant for all k , the standard deviation might refer to the variation throughout the MCMC chain.

¹⁰The matrix given in equation 7 is recursively defined and has thus been neglected in this section. See the estimation, section 3, for further details.

```

1 # The theta_delta matrix is given incompletely, all missing values
2 # are treated as 0. It is all parameters depending on k in
3 # interaction with a keyword's brand and specificity, giving a 4
4 # x2 (or 2x2) matrix.
5 theta_delta <- matrix(c(c(0.25, -0.51), # const x brand and spec
6 # c(0,0), # adpos x brand and spec
7 # c(-0.09, 0.64), # organic_comp x brand and
8 # spec
9 # c(-0.01,0.55)), # sponsored_comp x brand
# and spec
10 nrow=4, ncol=2, byrow=TRUE)
11
12 # The covariance matrices: basically the error "within" a set of
13 # parameters 4 thetas depend on k, so VTheta is 4x4. All in all,
14 # there are 4 V matrices, one for each parameter set, two of size
15 # 4x4, two of size 2x2.
16 VTheta <- riwish(4, diag(1, nrow=4))
17
18 # Define the function to draw the parameters dependent on k
19 # It first simulates the u_error and then takes the delta_matrix
# times the z vector.
20 calc_coeff_matr <- function(keywords, delta_matr, z, v_matr) {
21   l = matrix(nrow= 4, ncol=keywords)
22   for (k in 1:keywords){
23     u <- matrix(rmvnorm(n=1, mean=rep(0,nrow(delta_matr)), v_matr),
24                 nrow=nrow(delta_matr))
25     d <- delta_matr %*% z[[k]] + u
26     l[,k] <- d
27   }
28   return(l)
29 }
30
31 # The thetas dependent on k for CTR
32 random_thetas <- calc_coeff_matr(num_keywords, theta_delta, z,
# VTheta)
33
34 # Took the mean values for the thetas that do not depend on k.
35 theta4 <- -0.2; theta5 <- 0.3; theta6 <- -0.01

```

Listing 2: Code for simulating Δ^θ and V^θ .

2.3.2 The error terms ϵ_{kt}^θ of the linear model

As V^θ denotes the error within the different θ^k 's, Ω accounts for correlated error terms among θ , β , α and γ . Based on the numbers in figure 7, Ω is the sigma matrix for the draw of the 4 kxt error matrices ϵ from the multivariate normal distribution, see equation 9.

```

1 # calculate the error terms #
2 r1 <- c(0.370, -0.036, 0.000, -0.005)
3 r2 <- c(-0.036, 0.232, 0.007, 0.02)
4 r3 <- c(0, 0.007, 0.064, -0.005)
5 r4 <- c(-0.005, 0.02, -0.005, 0.087)
6
7 o_mat_cols <- c('CONV', 'CTR', 'Pos', 'Organic_comp')
8 omega_matrix <- matrix(c(r1, r2, r3, r4), nrow=4, ncol=4, byrow=
  TRUE, dimnames = list(o_mat_cols))
9 colnames(omega_matrix) <- o_mat_cols
10
11 error_terms <- rmvnorm(n = num_observations, sigma=omega_matrix)
12 errors <- vector(mode="list", length=4)
13 names(errors) <- c('theta_error', 'beta_error', 'gamma_error', 'alpha_
  error')
14 errors[[1]] <- matrix(error_terms[,1], nrow=num_keywords, ncol=num_
  days)
15 # [errors[[2]] till 4 in the same manner]
16
17 theta_error <- errors$theta_error

```

Listing 3: Code for generating the error terms.

Table 9: Estimates for the Covariance Matrix Ω

Cumulative Competition normalized by position and Alexa rank

	CONV	CTR	Pos	Organic_Comp
CONV	0.37 (0.06)***	-0.036 (0.015)**	0.00 (0.009)	-0.005 (0.014)
CTR		0.232 (0.02)***	0.007 (0.008)	0.02 (0.008)**
Pos			0.064 (0.003)***	-0.005 (0.002)**
Organic_Comp				0.087 (0.004)***

Figure 7: Table 9 from the paper, giving estimates for the error terms

2.3.3 The final step towards Λ_{kt}^{CTR}

The parameters were then combined into a linear model with the data and the error term ϵ_{kt}^θ , equation 2. The resulting utility of clicking U_{kt}^{CTR} represents the latent variable for the click-through rate. The logistic function of U_{kt}^{CTR} finally leads to the dependent variable CTR, the Λ_{kt}^{CTR} from equation 1 or `CTR_estimate` in the code.

```

1 # The final linear model
2 u_ctr <- matrix(nrow=num_keywords, ncol=num_days)
3 CTR_estimate <- matrix(nrow=num_keywords, ncol=num_days)
4 for (t in 1:num_days) {
5   for (k in 1:num_keywords){
6     u_ctr[k,t] <- random_thetas[[k]][1] + random_thetas[[k]][2] *
7       adpos_eq[k,t] +
8       random_thetas[[k]][3]*orgcomp_eq[k,t] + random_thetas[[k]][4]
9       *sponsored_comp[k,t] +
10      theta4*organic[k,t] + theta5*lqscore[k,t] +
11      theta6*t + theta_error[k,t]
12
13 }
14 # [same for betas, gammas and alphas]
```

Listing 4: Code for simulating the parameters, shown for θ .

2.4 Simulation and Summary Statistics Comparison

The end result of our simulation is the estimation of the CTR and CONV, ideally to be comparable to those found by the authors themselves. To compare our simulation outcome with the estimation from the paper, we first calculated the CTR and CONV based on the summary statistics given in the paper. The mean values for these are shown in table 1.

	CTR Paper	CTR Simulation	CONV Paper	CONV Simulation
Mean	0.01553	0.67544	0.00026	0.36571
Median	0.01369	0.90372	0	0.02198

Table 1: Mean and median values of simulated and paper-estimated CTR and CONV rates.

As is apparent in 1, our simulation produced both CTR and CONV rate estimates with a much higher mean and median value than the summary statistics in the paper would suggest. Examining the distribution of the data helps to make the estimates we generated more understandable.

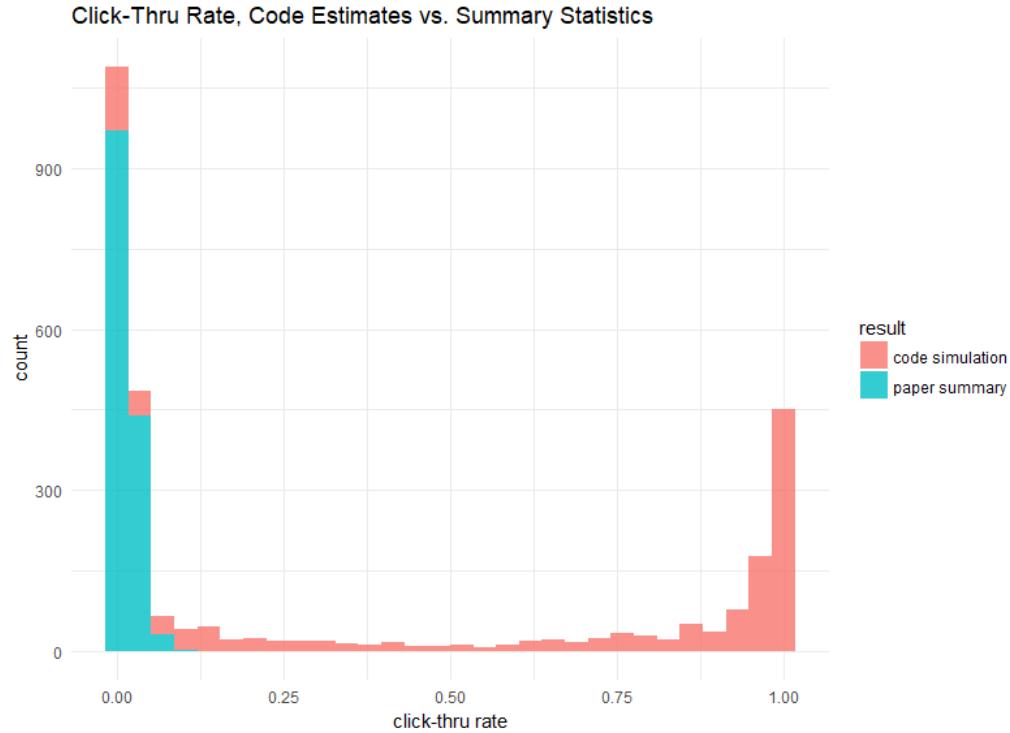


Figure 8: Count of simulated and summary statistic-based CTR estimates

Figure 8 shows that our simulation and the summary statistic-generated data do trend heavily toward 0, although our simulated data includes a much wider variety of possible click-through rates, as well as a great deal hovering around 1. While we did our best to follow the equations and formulas given in the paper, it is possible that we either mistakenly coded something, or that a piece of necessary information was missing in the paper itself¹¹.

The simulated conversion rate follows a similar pattern as that of the click-through rate, as is clear in figure 9. In keeping in line with the summary statistics data, our simulated data does trend around 0, yet, we again see a much greater distribution of data points in the simulated data. The spread does, however, seem to be less pronounced with the conversion rate than with the click-through rate.

¹¹As mentioned, there seem to have been several noticeable omissions in the data needed to fully replicate the paper.

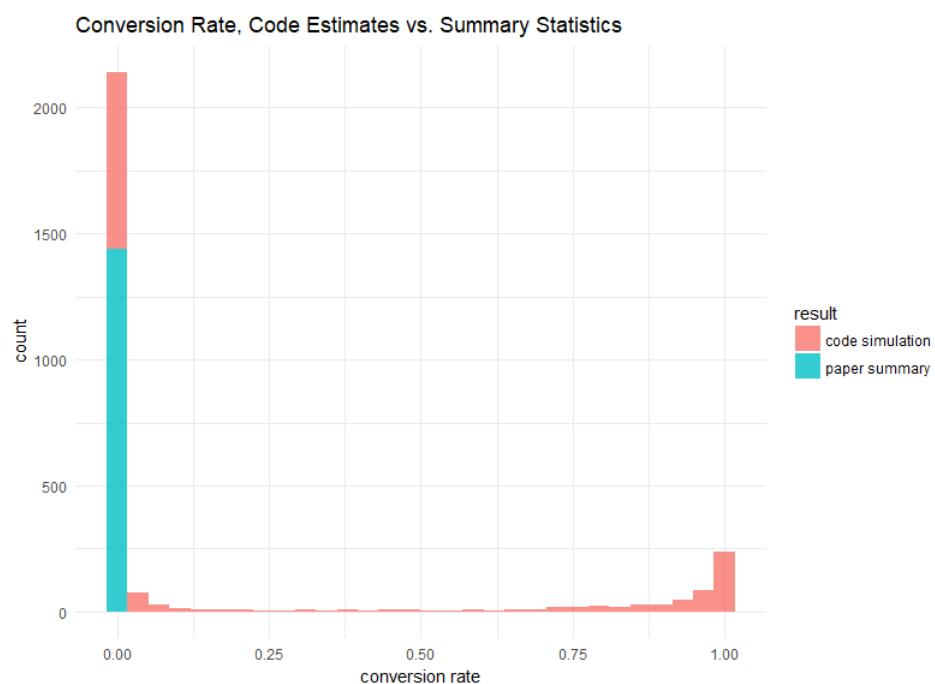


Figure 9: Count of simulated and summary statistic-based CONV estimates

3 Estimating the model using a Gibbs Sampler

While the simulation yields results, it used values that originally came from the Metropolis-Hastings Algorithm (Agarwal et al., 2015, Appendix, p. 2). To estimate the factors that drive the click-through rate and the orders, we implemented a Gibbs sampler. The original model runs 40,000 iterations as burn-in and 80,000 iterations in total.

A Bayesian model needs appropriate priors and a likelihood. The priors used in the model have been introduced as the distributions in section 1. The likelihood is a combined Bernoulli likelihood for both U_{kt}^{CTR} and U_{kt}^{CONV} (Agarwal et al., 2015, Appendix, p. 2). Yet, this hierarchical model is a triangular system, meaning it involves recursive elements which then become directed cycles. This imposed a problem when following the approach. The different steps have been summarized in table 2.

	Method	Main Focus	Solution
1	JAGS	<p>A basic implementation of the model.</p> <p>The paper's likelihood is a combined likelihood for both U_{kt}^{CTR} and U_{kt}^{CONV} and is dependent on itself. It does not match the input required by JAGS.</p> <p>When fully specified, the model seems to be recursive, it involves directed cycles, a "simultaneous model" as also stated by the authors (Agarwal et al., 2015, p. 19). In JAGS however, directed cycles are forbidden (Plummer, 2017, p. 15). It returned a compilation error and "possible directed cycle involving deltaTildeAlpha". The syntactically correct model <code>JAGS_attempt.R</code> can be found in the appendix.</p>	<p>It was replaced by a simple Bernoulli likelihood using <code>dbern()</code>.</p> <p>In contrast to JAGS, BUGS allows directed cycles at a user's own risk. BUGS was therefore installed and run by <code>R2OpenBUGS</code>.</p>

2	BUGS	<p>Though similar to JAGS, BUGS uses a slightly different syntax and generally offered fewer logical elements such as functions or conditional statements, e.g.:</p> <ul style="list-style-type: none"> - <code>%*%</code> - <code>if</code> statements - <code>dbern()</code> throws an error 	<ul style="list-style-type: none"> - matrix multiplication in loops instead of <code>%*%</code> - replaced <code>if</code> by <code>step()</code> - replaced by <code>dnorm()</code> for the sake of simplicity
		<p>In the end, BUGS reported a stack overflow which "can occur if there is a recursive definition of a logical node", so a simultaneous model. The syntactically correct model <code>BUGS_attempt.R</code> can be found in the appendix.</p>	<p>An alternative to BUGS is the package <code>bayesm</code>. The package and the accompanying book (Rossi et al., 2006) seem to match our model and allow for simultaneous models as well.</p>
3	<code>bayesm</code>	<p>Overall, the authors seemed to have followed Rossi et al.'s approach. In particular, the function <code>rhierMnlRwMixture</code> is quite similar to the model, but would need adjustments to mirror the model.</p>	<p>Based on the package and its functions, an MCMC could very well be defined by us as well. After all, the Metropolis only seems to be "another algorithm".</p>

4	Metropolis Hastings by hand	The appended file <code>Metropolis_Hastings.R</code> defined all necessary steps to run an MCMC by hand. It follows the appendix of the original paper. Though it encountered further dimension problems and notation issues, it was interpreted in a way that it matched dimension-wise. Unfortunately, the estimated values grow to ∞ after 5-7 big iterations, depending on the seed. We strongly suspect this is due to an incorrect use of the inverse, be it a notation problem or a problem in the code. See the code for more details.	To have a simple running model, JAGS was employed again.
5	JAGS	The last JAGS model implements the main four equations from equations 2 and 3 and assumes normally distributed parameters. The ϵ error terms have been taken from the simulation in 2. Parts of it can be seen below. The full model is denoted as <code>JAGS_running.R</code> .	

Table 2: A short overview to the approaches to using a Markov Chain Monte Carlo algorithm or a Gibbs sampler for this model.

```

1 modelString = "
2 model {
3   for (j in 1:N){
4     for (i in 1:M){
5
6       ctr[i, j] ~ dnorm(p[i, j], 1)
7       p[i, j] <- exp(z[i, j]) / (1 + exp(z[i, j]))
8       z[i, j] <- (-4.22 + theta0) + (-1.42 + theta1) * adpos_eq[i, j] +
9         random_thetas[3, i] * orgcomp_eq[i, j]
10      + random_thetas[4, i] * sponsored_comp[i, j] + theta4 * organic[i,
11        j]
12      + theta5 * lqscore[i, j] + theta6 * i + theta_error[i, j]
13
14      conv[i, j] ~ dnorm(r[i, j], 1)
15      r[i, j] <- exp(q[i, j]) / (1 + exp(q[i, j]))
16      q[i, j] <- (-2.75 + beta0) + (0.81 + beta1) * adpos_eq[i, j] +
17        random_betas[3, i] * orgcomp_eq[i, j] +
18      random_betas[4, i] * sponsored_comp[i, j] + beta4 * organic[i, j] +
19      beta5 * lqscore[i, j] + beta6 * i + beta_error[i, j]
20
21      adpos_eq[i, j] <- exp(random_gammas[1, i] + random_gammas[2, i] *
22        log(bid[i, j]) + gamma2 * lqscore[i, j] +
23        gamma3 * i + gamma_error[i, j])
24
25    }
26  }
27  gamma2 <- -0.049
28  gamma3 <- -0.002
29  alpha2 <- -0.004
30
31  theta0 ~ dnorm(0, 0.55)
32  theta1 ~ dnorm(0, 0.18)
33  theta4 <- -0.2
34  theta5 <- 0.3
35  theta6 <- -0.01
36
37  beta0 ~ dnorm(0, 0.68)
38  beta1 ~ dnorm(0, 0.28)
39  beta4 <- 0.17
40  beta5 <- -0.01
41  beta6 <- -0.01
42  }
43  "
44  writeLines(modelString, con="TEMPmodel.txt")
45
46  jags <- jags.model('TEMPmodel.txt',
47    data = list('N' = num_days,
48               'M' = num_keywords,
49               'adpos' = adpos,
50               'ctr' = CTR_estimate,
51               'conv' = CONV_estimate,
52               'bid' = bid,

```

```

53   'iv_organic' = iv_organic,
54   'orgcomp' = organic_comp,
55   'organic' = organic,
56   'sponsored_comp' = sponsored_comp,
57   'lqscore' = lqscore,
58   'theta_error' = theta_error,
59   'beta_error' = beta_error,
60   'random_betas' = random_betas,
61   'random_thetas' = random_thetas,
62   'random_gammas' = random_gammas,
63   'random_alphas' = random_alphas,
64   'gamma_error' = gamma_error,
65   'alpha_error' = alpha_error),
66   n.chains = 4,
67   n.adapt = 1000)
68
69 update(jags, 1000)

```

Listing 5: Estimating the model with a very basic JAGS version

Although we ran into some calculation issues with our more complicated model, we were able to achieve results using a simplified version. In this version, 1000 adaptations were followed by a 1000 burn-in period. The plots here show the previous values for CTR and CONV that were calculated by our data simulation, the summary statistics-specified CTR and CONV, and the new values for these two rates calculated by JAGS.

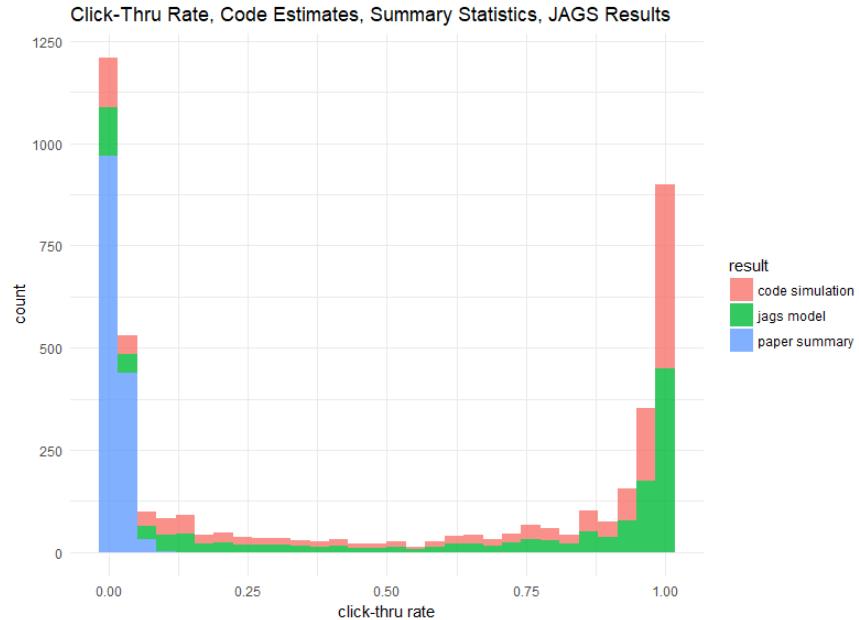


Figure 10: Count of simulated and summary statistic-based CTR estimates

The distribution of the JAGS-generated CTR values is shown as somewhere

between the data-simulated values and those from the summary statistics - this is displayed in figure 11. As seen in figure 10, the spread of the conversation rates also follows this pattern.

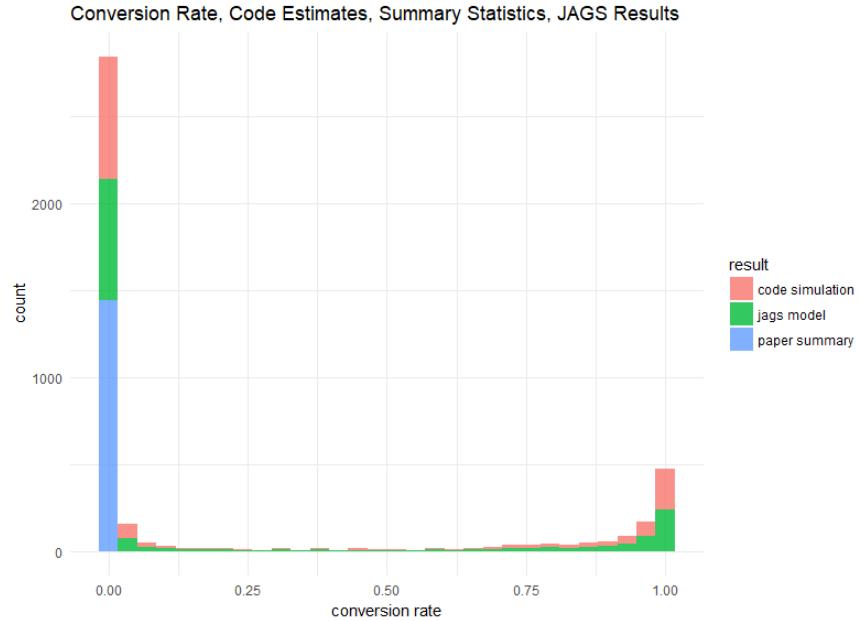


Figure 11: Count of simulated and summary statistic-based CTR estimates

As we see in the trace plots of theta0 and theta1 (figure 12), the distribution of the chain is relatively normal and the trajectory is stable over time. As we are expecting to see normal distributions, this is a good indicator that things are working properly (White, 2010).

Looking at figure 13, the Gelman plots for theta0 and theta1, we see that they appear to be well-dispersed throughout the sampling period. This indicates that our chains are relatively similar to each other, at least in terms of values for theta0.

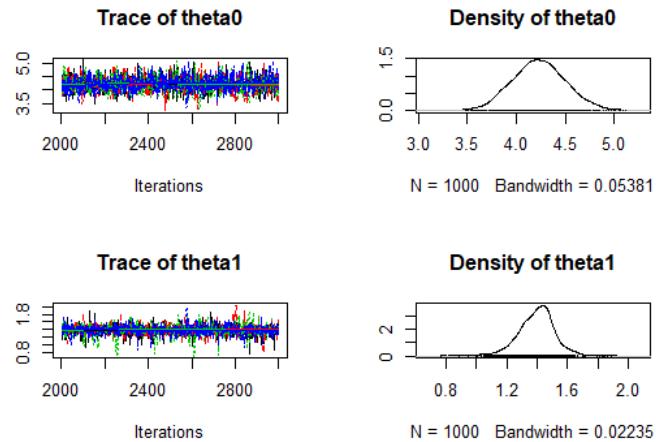


Figure 12: Trace plot of Theta0 and Theta1 from JAGS model

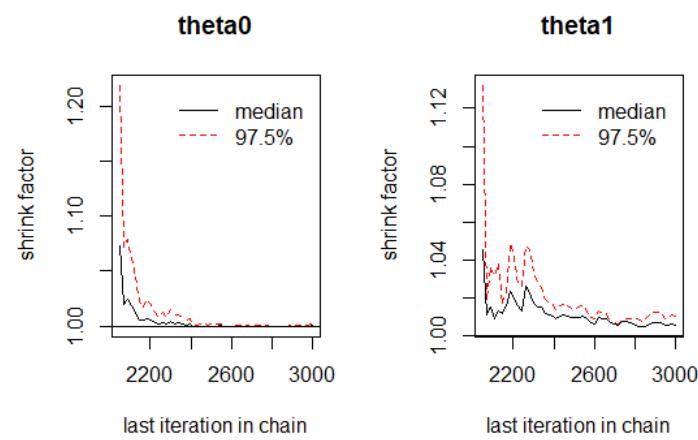


Figure 13: Gelman plots of Theta0 and Theta1



Figure 14: Google ad positions in 2009

4 Evaluation of the model

The model involves a number of short-comings regarding its notation and mathematical formulation, and many errors have been reported above. In general, following the verified approach by Rossi et al. should ensure a conclusive model. From a mathematical perspective, the chosen levels of hierarchy, the way to account for endogeneity and the chosen logit function make sense despite the level of complexity. The simultaneity however adds a certain fragility to the method, the numbers, and thus the reliability of the instrument as a whole. This system cannot be easily reproduced nor can the correctness of the results be cross-validated.

Furthermore, some parameters, such as data on competitors, are completely missing. This forced us to focus on each particular detail of the model and effectively interpret the missing data in the most probable way according to our interpretation. While this is the only approach we could have taken given the scarcity of the data, it made us question the accurateness of the estimations and whether the results can still hold.

Another thing worth mentioning is that the model described in the paper cannot be replicated entirely due to a different layout of Google search pages. The positioning of both organic and paid ads has changed a great deal since 2009 (when the experiment in the Agarwal et al. paper was conducted), which also means that the way users interact with the ads would not be the same today. Examples of ad positions then and now are provided below: Figure 14 from 2009 is taken from the paper itself, while the current one in Figure 15 is prepared by the group.

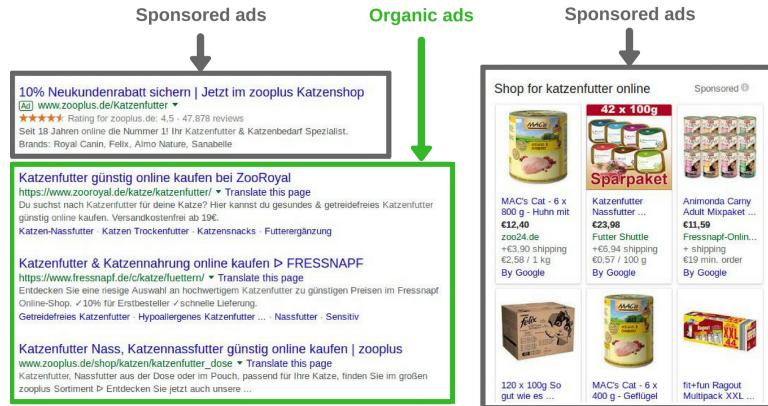
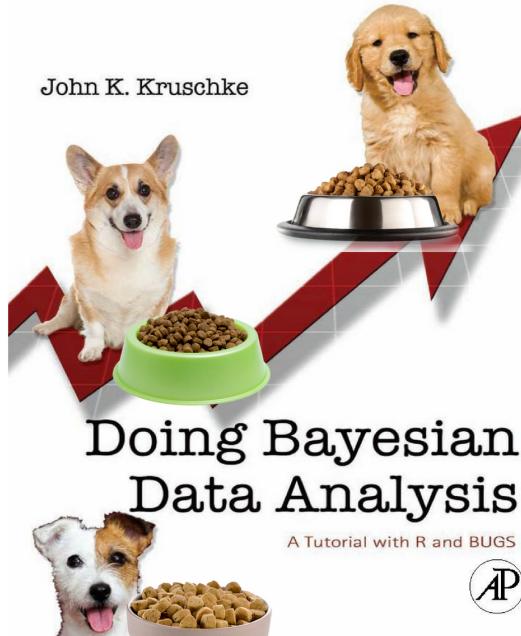


Figure 15: Google ad positions today

All in all, the model by Agarwal et al. posed quite a few challenges when trying to follow the approach. The model as such, and especially its documentation, could be improved in the future. We would like to close this paper with a small statement: if pet food were a problem, then the most simple solution might be the correct one, according to William of Ockham. Just give the dogs some petfood.



Bibliography

- Agarwal, A., Hosanagar, K., and Smith, M. D. (2015). Do Organic Results Help or Hurt Sponsored Search Performance? *Information Systems Research*, 26(4):695–713.
- Ghose, A. and Yang, S. (2009). An empirical analysis of search engine advertising: Sponsored search in electronic markets. *Management Science*, 55(10):1605–1622.
- Plummer, M. (2017). Jags version 4.3.0 user manual.
- Rossi, P., Allenby, G., and McCulloch, R. (2006). *Bayesian Statistics and Marketing*. Wiley.
- White, J. M. (2010). Mcmc diagnostics in r with the coda package.

5 Eidesstattliche Erklärung

Hiermit versichere ich, dass die Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt wurden, alle Stellen der Arbeit, die wortwörtlich oder sinngemäß aus anderen Quellen übernommen und als solche kenntlich gemacht wurden und die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Datum und Unterschrift

Datum und Unterschrift

Datum und Unterschrift

Statutory Declaration

I hereby declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Date and signature

Date and signature

Date and signature