Beyond Exchangeability: The Chinese Voting Process

Moontae Lee
Dept. of Computer Science
Cornell University
Ithaca, NY 14853
moontae@cs.cornell.edu

Seok Hyun Jin
Dept. of Computer Science
Cornell University
Ithaca, NY 14853
sj372@cornell.edu

David Mimno
Dept. of Information Science
Cornell University
Ithaca, NY 14853
mimno@cornell.edu

Abstract

Many online communities present user-contributed responses such as reviews of products and answers to questions. User-provided helpfulness votes can highlight the most useful responses, but voting is a social process that can gain momentum based on the popularity of responses and the polarity of existing votes. We propose the Chinese Voting Process (CVP) which models the evolution of helpfulness votes as a self-reinforcing process dependent on position and presentation biases. We evaluate this model on Amazon product reviews and more than 80 StackExchange forums, measuring the intrinsic quality of individual responses and behavioral coefficients of different communities.

1 Introduction

With the expansion of online social platforms, user-generated content has become increasingly influential. Customer reviews in e-commerce like Amazon are often more helpful than editorial reviews [14], and question answers in Q&A forums such as StackOverflow and MathOverflow are highly useful for coders and researchers [9, 18]. Due to the diversity and abundance of user content, promoting better access to more useful information is critical for both users and service providers. Helpfulness voting is a powerful means to evaluate the quality of user responses (i.e., reviews/answers) by the wisdom of crowds. While these votes are generally valuable in aggregate, estimating the true quality of the responses is difficult because users are heavily influenced by previous votes. We propose a new model that is capable of learning the intrinsic quality of responses by considering their social contexts and momentum.

Previous work in self-reinforcing social behaviors shows that although inherent quality is an important factor in overall ranking, users are susceptible to *position bias* [12, 13]. Displaying items in an order affects users: top-ranked items get more popularity, while low-ranked items remain in obscurity. We find that sensitivity to orders also differs across communities: some value a range of opinions, while others prefer a single authoritative answer. Summary information displayed together can lead to *presentation bias* [19]. As the current voting scores are visibly presented with responses, users inevitably perceive the score before reading the contents of responses. Such exposure could immediately nudge user evaluations toward the majority opinion, making high-scored responses more attractive. We also find that the relative length of each response affects the polarity of future votes.

Res	Votes	Diff	Ratio	Relative Quality
1	$+++--$	0	0.5	quite negative
2	$+-+-+-$	0	0.5	moderately negative
3	$-+-+-+$	0	0.5	moderately positive
4	$---+++$	0	0.5	quite positive

Table 1: Quality interpretation for each sequence of six votes.

Standard discrete models for self-reinforcing process include the Chinese Restaurant Process and the Pólya urn model. Since these models are *exchangeable*, the order of events does not affect the probability of a sequence. However, Table 1 suggests how different *contexts* of votes cause different impacts. While the four sequences have equal numbers of positive and

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

negative votes in aggregate, the fourth votes in the first and last responses are given against a clear majority opinion. Our model treats objection as a more challenging decision, thereby deserving higher weight. In contrast, the middle two sequences receive alternating votes. As each vote is a relatively weaker disagreement, the underlying quality is moderate compared to the other two responses. Furthermore, if these are responses to one item, the order between them also matters. If the initial three votes on the fourth response pushed its display position to the next page, for example, it might not have a chance to get future votes, which recover its reputation.

The **Chinese Voting Process (CVP)** models generation of responses and votes, formalizing the evolution of helpfulness under *positional* and *presentational* reinforcement. Whereas most previous work on helpfulness prediction [7, 5, 8, 4, 11, 10, 15] has involved a single snapshot, the CVP estimates intrinsic quality of responses solely from selection and voting trajectories over multiple snapshots. The resulting model shows significant improvements in predictive probability for helpfulness votes, especially in the critical early stages of a trajectory. We find that the CVP estimated intrinsic quality ranks responses better than existing system rank, correlating orderly with the sentiment of comments associated with each response. Finally, we qualitatively compare different characteristics of self-reinforcing behavior between communities using two learned coefficients: *Trendiness* and *Conformity*. The two-dimensional embedding in Figure 1 characterizes different opinion dynamics from Judaism to Javascript (in StackOverflow).

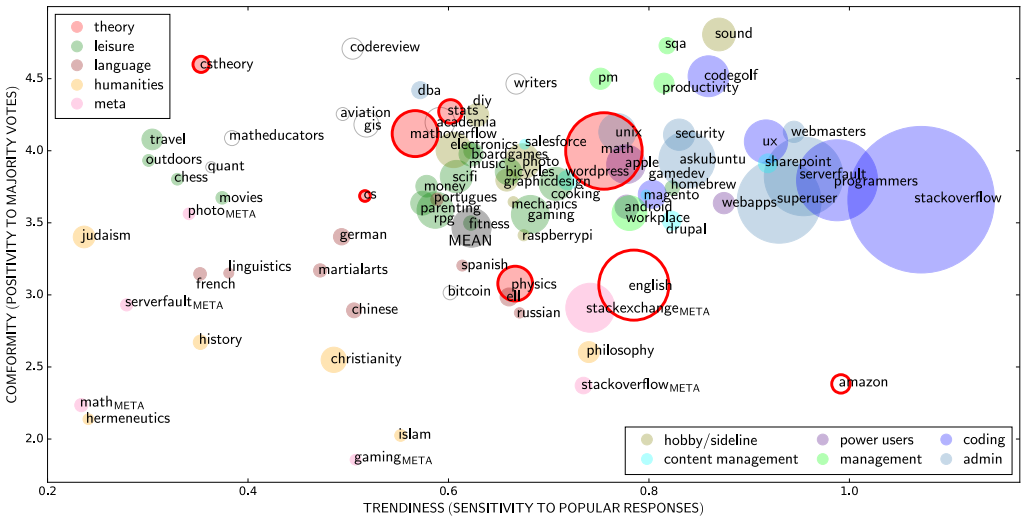


Figure 1: 2D Community embedding. Each of 83 communities is represented by two behavioral coefficients (*Trendiness*, *Conformity*). Eleven clusters are grouped based on their common focus. The MEAN community is synthesized by sampling 20 questions from every community (except Amazon due to the different user interface).

Related work. There is strong evidence that helpfulness voting is socially influenced. Helpfulness ratings on Amazon product reviews differ significantly from independent human annotators [8]. Votes are generally more positive, and the number of votes decreases exponentially based on displayed page position. Review polarity is biased towards matching the consensus opinion [4]: when two reviews contain essentially the same text but differ in star rating, the review closer to the consensus star rating is considered more helpful. There is also evidence that users vote strategically to correct perceived mismatches in review rank [16]. Many studies have attempted to predict helpfulness given review-content features [7, 5, 11, 10, 15]. Each of these examples predicts helpfulness based on text, star-ratings, sales, and badges, but only at a single snapshot. Our work differs in two ways. First, we combine data on Amazon helpfulness votes from [16] with a much larger collection of helpfulness votes from 82 StackExchange forums. Second, instead of considering text-based features (which we hold out for evaluation) within a single snapshot, we attempt to predict the next vote at each stage based on the previous voting trajectory over multiple snapshots without considering textual contents.

2 The Chinese Voting Process

Our goal is to model helpfulness voting as a two-phase self-reinforcing stochastic process. In the *selection phase*, each user either selects an existing response based on their positions or writes a new

response. The positional reinforcement is inspired by the Chinese Restaurant Process (CRP) and Distance Dependent Chinese Restaurant Process (ddCRP). In the *voting phase*, when one response is selected, the user chooses one of the two feedback options: a positive or negative vote based on the intrinsic quality and the presentational factors. The presentational reinforcement is modeled by a log-linear model with time-varying features based on the Pólya urn model. The CVP implements *the-rich-get-richer* dynamics as an interplay of these two preferential reinforcements, learning latent qualities of individual responses as inspired in Table 1. Specifically, each user at time t interested in the item i follows the generative story in Table 2.

Generative process	**Sample parametrization (Amazon)**
1. Evaluate j-th response: $p(z_i^{(t)} = j \mid z_i^{(1:t-1)}; \alpha) \propto f_i^{(t-1)}(j)$	$f_i^{(t)}(j) = \Big(\frac{1}{1 + \textit{the-display-rank}_i^{(t)}(j)}\Big)^{\tau}$
(a) 'Yes': $p(v_i^{(t)} = 1 \mid \boldsymbol{\theta}) = logit^{-1}(q_{ij} + g_i^{(t-1)}(j))$	$g_i^{(t)}(j) = \lambda r_{ij}^{(t)} + \mu s_{ij}^{(t)} + \nu_i u_{ij}^{(t)}$
(b) 'No' : $p(v_i^{(t)} = 0 \mid \boldsymbol{\theta}) = 1 - p(v_i^{(t)} = 1 \mid \boldsymbol{\theta})$	$\boldsymbol{\theta} = \{\{q_{ij}\}, \lambda, \mu, \{\nu_i\}\}$
2. Or write a new response: $p(z_i^{(t)} = J_i + 1 \mid z_i^{(1:t-1)}; \alpha) \propto \alpha$	$J_i = J_i^{(t-1)}$ (abbreviated notation)
(a) Sample $q_{i(J+1)}$ from $\mathcal{N}(0, \sigma^2)$.	

Table 2: The generative story and the parametrization of the Chinese Voting Process (CVP).

2.1 Selection phase

The CRP [1, 2] is a self-reinforcing decision process over an infinite discrete set. For each item (product/question) i, the first user writes a new response (review/answer). The t-th subsequent user can choose an existing response j out of $J_i^{(t-1)}$ possible responses with probability proportional to the number of votes $n_j^{(t-1)}$ given to the response j by time $t-1$, whereas the probability of writing a new response $J_i^{(t-1)} + 1$ is proportional to a constant α. While the CRP models self-reinforcement — each vote for a response makes that response more likely to be selected later — there is evidence that the actual selection rate in an ordered list decays with display rank [6]. Since such rankings are mechanism-specific and not always clearly known in advance, we need a more flexible model that can specify various degrees of positional preference. The ddCRP [3] introduces a function f that decays with respect to some distance measure. In our formulation, the distance function varies over time and is further configurable with respect to the specific interface of service providers.

Specifically, the function $f_i^{(t)}(j)$ in the CVP evaluates the popularity of the j-th response in the item i at time t. Since we assume that popularity of responses is decided by their positional accessibility, we can parametrize f to be inversely proportional to their display ranks. The exponent τ determines sensitivity to popularity in the selection phase by controlling the degree of harmonic penalization over ranks. Larger $\tau > 0$ indicates that users are more sensitive to trendy responses displayed near the top. If $\tau < 0$, users often select low-ranked responses over high-ranked ones for some reasons.[1] Note that even if the user at time t does not vote on the j-th response, $f_i^{(t)}(j)$ could be different from $f_i^{(t-1)}(j)$ in the CVP,[2] whereas $n_{ij}^{(t)} = n_{ij}^{(t-1)}$ in the CRP. Thus one can view the selection phase of the CVP as a *non-exchangeable* extension of the CRP via a time-varying function f.

2.2 Voting phase

We next construct a self-reinforcing process for the inner voting phase. The Pólya urn model is a self-reinforcing decision process over a finite discrete set, but because it is exchangeable, it is unable to capture contextual information encoded in each a sequence of votes. We instead use a log-linear formulation with the urn-based features, allowing other presentational features to be flexibly incorporated based on the modeler's observations.

Each response initially has $x = x^{(0)}$ positive and $y = y^{(0)}$ negative votes, which could be fractional pseudo-votes. For each draw of a vote, we return $w + 1$ votes with the same polarity, thus self-reinforcing when $w > 0$. The following Table 3 shows time-evolving positive/negative ratios $r_j^{(t)} = x_j^{(t)}/(x_j^{(t)} + y_j^{(t)})$ and $s_j^{(t)} = y_j^{(t)}/(x_j^{(t)} + y_j^{(t)})$ of the first two responses: $j \in \{1, 2\}$ in Table 1 with the corresponding ratio gain $\Delta_j^{(t)} = r_j^{(t)} - r_j^{(t-1)}$ (if $v_j^{(t)} = 1$ or $+$) or $s_j^{(t)} - s_j^{(t-1)}$ (if $v_j^{(t)} = 0$ or $-$).

[1] This sometimes happens especially in the early stage when only a few responses exist.

[2] Say the rank of another response j' was lower than j's at time $t-1$. If t-th vote given to the response j' raises its rank higher than the rank of the response j, then $f_i^{(t)}(j) < f_i^{(t-1)}(j)$ assuming $\tau > 0$.

t or T	$v_1^{(t)}$	$r_1^{(t)}$	$s_1^{(t)}$	$\Delta_1^{(t)}$	q_1^T	$v_2^{(t)}$	$r_2^{(t)}$	$s_2^{(t)}$	$\Delta_2^{(t)}$	q_2^T
0		1/2	1/2				1/2	1/2		
1	+	2/3	1/3	0.167	−	+	2/3	1/3	0.167	−
2	+	3/4	1/4	0.083	0.363	−	2/4	2/4	0.167	-0.363
3	+	4/5	1/5	0.050	0.574	+	3/5	2/5	0.100	0.004
4	−	4/6	2/6	0.133	0.237	−	3/6	3/6	0.100	-0.230
5	−	4/7	3/7	0.095	0.004	+	4/7	3/7	0.071	0.007
6	−	4/8	4/8	0.071	**-0.175**	−	4/8	4/8	0.071	**-0.166**

Table 3: Change of quality estimation q_j over times for the first two example responses in Table 1 with the initial pseudo-votes $(x, y, w) = (1, 1, 1)$. The estimated quality at the first response sharply decreases when receiving the first majority-against vote at $t = 4$. The first response ends up being more negative than the second, even if they receive the same number of votes in aggregate. These *non-exchangeable* behaviors cannot be modeled with a simple exchangeable process.

In this toy setting, the polarity of a vote to a response is an outcome of its intrinsic quality as well as presentational factors: positive and negative votes. Thus we model each sequence of votes by ℓ_2-regularized logistic regression with the latent intrinsic quality and the Pólya urn ratios.[3]

$$\max_{\boldsymbol{\theta}} \ \log \prod_{t=2}^{T} logit^{-1}\big(q_j^T + \lambda r_j^{(t-1)} + \mu s_j^{(t-1)}\big) - \frac{1}{2}\|\boldsymbol{\theta}\|_2^2 \quad \text{where } \boldsymbol{\theta} = \big(q_j^T, \lambda, \mu\big) \tag{1}$$

The $\{q_j^T\}$ in the Table 3 shows the result from solving (1) up to T-th votes for each $j \in \{1, 2\}$. The initial vote given at $t = 1$ is disregarded in the training due to its arbitrariness from the uniform prior ($x_0 = y_0$). Since it is quite possible to have only positive or only negative votes, Gaussian regularization is necessary. Note that using the urn-based ratio features is essential to encode contextual information. If we instead use raw count features (only the numerators of r_j and s_j), for example in the first response, the estimated quality q_1^T keeps increasing even after getting negative votes from time 4 to 6. Log raw count features are unable to infer the negative quality.

In the first response, $\Delta_1^{(t)}$ shows the decreasing gain in positive ratios from $t = 1$ to 3 and in negative ratios from $t = 4$ to 6, whereas it gains a relatively large momentum at the first negative vote when $t = 4$. $\Delta_2^{(t)}$ converges to 0 in the 2nd response, implying that future votes have less effect than earlier votes for alternating $+/-$ votes. q_2^T also converges to 0 as we expect neutral quality in the limit. Overall the model is capable of learning intrinsic quality as desired in Table 1 where relative gains can be further controlled by tuning the initial pseudo-votes (x, y).

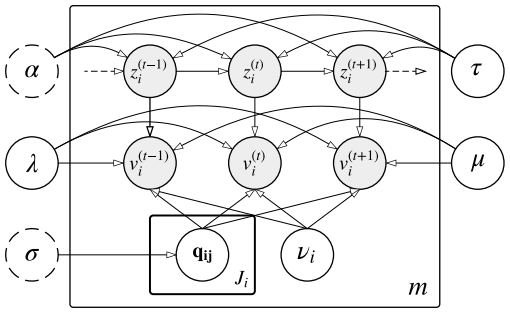
In the real setting, the polarity score function $g_i^{(t)}(j)$ in the CVP evaluates presentational factors of the j-th response in the item i at time t. Because we adopt a log-linear formulation, one can easily add additional information about responses. In addition to the positive ratio $r_{ij}^{(t)}$ and the negative ratio $s_{ij}^{(t)}$, g also contains a length feature $u_{ij}^{(t)}$ (as given in Table 2), which is the relative length of the response j against the average length of responses in the item i at particular time t. Users in some items may prefer shorter responses than longer ones for brevity, whereas users in other items may blindly believe that longer responses are more credible before reading their contents. The parameter ν_i explains length-wise preferential idiosyncrasy as a per-item bias: $\nu_i < 0$ means a preference toward the shorter responses. Note that $g_i^{(t)}(j)$ could be different from $g_i^{(t-1)}(j)$ even if the user at time t does not choose to vote.[4] All together, the voting phase of the CVP generates *non-exchangeable* votes.

3 Inference

Each phase of the CVP depends on the result of all previous stages, so decoupling these related problems is crucial for efficient inference. We need to estimate community-level parameters, item-level length preferences, and response-level intrinsic qualities. The graphical model of the CVP and corresponding parameters to estimate are illustrated in Table 4. We further compute two community-level behavioral coefficients: *Trendiness* and *Conformity*, which are useful summary statistics for exploring different voting patterns and explaining macro characteristics across different communities.

[3] One might think (1) can be equivalently achievable with only two parameters because of $r_j^{(t)} + s_j^{(t)} = 1$ for all t. However, such reparametrization adds inconsistent translations to q_j^T and makes it difficult to interpret different inclinations between positive and negative votes for various communities.

[4] If a new response is written at time t, $u_{ij}^{(t)} \neq u_{ij}^{(t-1)}$ as the new response changes the average length.



α: hyper-parameter for response growth
σ^2: hyper-parameter for quality variance
τ: community-level sensitivity to popularity
λ: community-level preference for positive ratio
μ: community-level preference for negative ratio
ν_i: item-level preference for response length
q_{ij}: response-level hidden intrinsic quality
m: # of items (e.g., products/questions)
J_i: # of responses of item i (e.g., reviews/answers)

Table 4: Graphical model and parameters for the CVP. Only three time steps are unrolled for visualization.

Parameter inference. The goal is to infer parameters $\boldsymbol{\theta} = \{\{q_{ij}\}, \lambda, \mu, \{\nu_i\}\}$. We sometimes use f and g instead to compactly indicate parameters associated to each function. The likelihood of one CVP step in the item i at time t is $L_i^{(t)}(\tau, \boldsymbol{\theta}; \alpha, \sigma) =$

$$\Big\{\frac{\alpha}{\alpha+\sum_{j=1}^{J_i^{(t-1)}} f_i^{(t-1)}(j)}\mathcal{N}(q_{i,z_i^{(t)}}; 0, \sigma^2)\Big\}^{\mathbb{1}[z_i^{(t)}=J_i^{(t-1)}+1]}\Big\{\frac{f_i^{(t-1)}(z_i^{(t)})}{\alpha+\sum_{j=1}^{J_i^{(t-1)}} f_i^{(t-1)}(j)}p(v_i^{(t)}|q_{i,z_i^{(t)}}, g_i^{(t-1)}(j))\Big\}^{\mathbb{1}[z_i^{(t)}\le J_i^{(t-1)}]}$$

where the two terms correspond to writing a new response and selecting an existing response to vote. The fractions in each term respectively indicate the probability of writing a new response and choosing existing responses in the selection phase. The other two probability expression in each term describe quality sampling from a normal distribution and the logistic regression in the voting phase.

It is important to note that our setting differs from many CRP-based models. The CRP is typically used to represent a non-parametric prior over the choice of latent cluster assignments that must themselves be inferred from noisy observations. In our case, the result of each choice is directly observable because we have the *complete trajectory* of helpfulness votes. As a result, we only need to infer the continuous parameters of the process, and not combinatorial configurations of discrete variables. Since we know the complete trajectory where the rank inside the function f is a part of the true observations, we can view each vote as an independent sample. Denoting the last timestamp of the item i by T_i, the log-likelihood becomes $\ell(\tau, \boldsymbol{\theta}; \alpha, \sigma) = \sum_{i=1}^{m}\sum_{t=1}^{T_i}\log L_i^{(t)}$ and is further separated into two pieces:

$$\ell_v(\boldsymbol{\theta}; \sigma) = \sum_{i=1}^{m}\sum_{t=1}^{T_i}\Big\{\mathbb{1}[write]\cdot\log\mathcal{N}(q_{i,z_i^{(t)}}; 0, \sigma^2) + \mathbb{1}[choose]\cdot\log p(v_i^{(t)}|q_{i,z_i^{(t)}}, g_i^{(t-1)}(j))\Big\}, \tag{2}$$

$$\ell_s(\tau; \alpha) = \sum_{i=1}^{m}\sum_{t=1}^{T_i}\Big\{\mathbb{1}[write]\cdot\log\frac{\alpha}{\alpha+\sum_{j=1}^{J_i^{(t-1)}} f_i^{(t-1)}(j)} + \mathbb{1}[choose]\cdot\log\frac{f_i^{(t-1)}(z_i^{(t)})}{\alpha+\sum_{j=1}^{J_i^{(t-1)}} f_i^{(t-1)}(j)}\Big\}.$$

Inferring a whole trajectory based only on the final snapshots would likely be intractable for a non-exchangeable model. Due to the continuous interaction between f and g for every time step, small mis-predictions in the earlier stages will cause entirely different configurations. Moreover the rank function inside f is in many cases site-specific.[5] It is therefore vital to observe all trajectories of random variables $\{z_i^{(t)}, v_i^{(t)}\}$: decoupling f and g reduces the inference problem into estimating parameters separately for the selection phase and the voting phase. Maximizing ℓ_v can be efficiently solved by ℓ_2-regularized logistic regression as demonstrated for (1). If the hyper-parameter α is fixed, maximizing ℓ_s becomes a convex optimization because τ appears in both the numerator and the denominator. Since the gradient for each parameter in $\boldsymbol{\theta}$ is obvious, we only include the gradient of $\ell_{s,i}^{(t)}$ for the particular item i at time t with respect to τ. Then $\frac{\partial\ell_s}{\partial\tau} = \sum_{i=1}^{m}\sum_{t=1}^{T_i}\partial\ell_{s,i}^{(t)}/\partial\tau$.

$$\frac{\partial\ell_{s,i}^{(t)}}{\partial\tau} = \frac{1}{\tau}\left\{\mathbb{1}[z_i^{(t)}\le J_i^{(t-1)}]\cdot\frac{f_i^{(t-1)}(z_i^{(t)})\log f_i^{(t-1)}(z_i^{(t)})}{f_i^{(t-1)}(z_i^{(t)})} - \frac{\sum_{j=1}^{J_i^{(t-1)}} f_i^{(t-1)}(j)\log f_{if}^{(t-1)}(j)}{\alpha+\sum_{j=1}^{J_i^{(t-1)}} f_i^{(t-1)}(j)}\right\} \tag{3}$$

[5]We generally know that Amazon decides the display order by the portion of positive votes and the total number of votes on each response, but the relative weights between them are not known. We do not know how StackExchange forums break ties, which affects highly in the early stages of voting.

Community	Selection		Voting							Residual		Bumpiness	
	CRP	*CVP*	q_{ij}	λ	ν_i	q_{ij}, λ	q_{ij}, ν_i	λ, ν_i	*Full*	*Rank*	*Qual*	*Rank*	*Qual*
SOF$_{(22925)}$	2.152	**1.989**	.107	.103	.108	.100	.106	.100	**.096**	.005	**.003**	.080	**.038**
math$_{(6245)}$	**1.841**	1.876	.071	.064	.067	.062	.066	.060	**.059**	.014	**.008**	.280	**.139**
english$_{(5242)}$	1.969	**1.924**	.160	.146	.152	.141	.147	.137	**.135**	.018	**.007**	.285	**.149**
mathOF$_{(2255)}$	1.992	**1.910**	.049	.046	.049	.045	.047	.046	**.045**	.009	**.007**	.185	**.119**
physics$_{(1288)}$	1.824	**1.801**	.174	.155	.166	.150	.156	.146	**.142**	.032	**.014**	.497	**.273**
stats$_{(598)}$	1.889	**1.822**	.051	.044	.048	.043	.046	.042	**.042**	.030	**.019**	.613	**.347**
judaism$_{(504)}$	2.039	**1.859**	.135	.124	.132	.121	.125	.118	**.116**	.046	**.018**	.875	**.403**
amazon$_{(363)}$	2.597	**2.261**	.266	.270	.262	.254	.243	.253	**.240**	.023	**.016**	.392	**.345**
meta.SOF$_{(294)}$	**1.411**	1.575	.261	.241	.270	.229	.243	.232	**.225**	.018	**.013**	.281	**.255**
cstheory$_{(279)}$	1.893	**1.795**	.052	.040	.053	.039	.049	.039	**.038**	.032	**.029**	**.485**	.553
cs$_{(123)}$	1.825	**1.780**	.128	.100	.118	.099	.113	.097	**.096**	.069	**.040**	.725	**.673**
linguistics$_{(107)}$	1.993	**1.789**	.133	.127	.130	.122	.123	.120	**.116**	.074	**.038**	.778	**.656**
AVERAGE	2.050	**1.945**	.109	.103	.108	.099	.105	.098	**.095**	.011	**.006**	.186	**.101**

Table 5: Predictive analysis on the first 50 votes: In the selection phase, the CVP shows better negative log-likelihood in almost all forums. In the voting phase, the full model shows better negative log-likelihood than all subsets of features. Quality analysis at the final snapshot: Smaller residuals and bumpiness show that the order based on the estimated quality q_{ij} more coherently correlates with the average sentiments of the associated comments than the order by display rank. (SOF=StackOverflow, OF=Overflow, rest=Exchange, Blue: $p \leq 0.001$, Green: $p \leq 0.01$, Red: $p \leq 0.05$)

Behavioral coefficients. To succinctly measure overall voting behaviors across different communities, we propose two community-level coefficients. *Trendiness* indicates the sensitivity to positional popularity in the selection phase. While the community-level τ parameter renders Trendiness simply to avoid overly-complicated models, one can easily extend the CVP to have per-item τ_i to better fit the data. In that case, Trendiness would be a summary statistics for $\{\tau_i\}$. *Conformity* captures users' receptiveness to prevailing polarity in the voting phase. To count every single vote, we define Conformity to be a geometric mean of odds ratios between majority-following votes and majority-disagreeing votes. Let V_i be the set of time steps when users vote rather than writing responses in the item i. Say n is the total number of votes across all items in the target community. Then Conformity is defined as

$$\kappa = \left\{ \prod_{i=1}^{m} \prod_{t \in V_i} \left(\frac{P(v_i^{(t+1)} = 1 | q^t_{i,z_i^{(t+1)}}, \lambda^t, \mu^t, \nu_i^t)}{P(v_i^{(t+1)} = 0 | q^t_{i,z_i^{(t+1)}}, \lambda^t, \mu^t, \nu_i^t)} \right)^{h_i^{(t)}} \right\}^{1/n} \quad \text{where} \quad h_i^{(t)} = \begin{cases} 1 & (n_{ij}^{+(t)} \geq n_{ij}^{-(t)}) \\ -1 & (n_{ij}^{+(t)} < n_{ij}^{-(t)}) \end{cases}.$$

To compute Conformity κ, we need to learn $\boldsymbol{\theta}^t = \{q_{ij}^t, \lambda^t, \mu^t, \nu_i^t\}$ for each t, which is a set of parameters learned on the data only up to the time t. This is because the user at time t cannot see any future which will be given later than the time t. Note that $\boldsymbol{\theta}^{t+1}$ can be efficiently learned by *warm-starting* at $\boldsymbol{\theta}^t$. In addition, while positive votes are mostly dominant in the end, the dominant mood up to time t could be negative, exactly when the user at time $t+1$ tries to vote. In this case, $h_i^{(t)}$ becomes -1, inverting the fraction to be the ratio of following the majority against the minority. By summarizing learned parameters in terms of two coefficients (τ, κ), we can compare different selection/voting behaviors for various communities.

4 Experiments

We evaluate the CVP on product reviews from Amazon and 82 issue-specific forums from the StackExchange network. The Amazon dataset [16] originally consisted of 595 products with daily snapshots of writing/voting trajectories from Oct 2012 to Mar 2013. After eliminating duplicate products[6] and products with fewer than five reviews or fragmented trajectories,[7] 363 products are left. For the StackExchange dataset[8], we filter out questions from each community with fewer than five answers besides the answer chosen by the question owner.[9] We drop communities with fewer than 100 questions after pre-processing. Many of these are "Meta" forums where users discuss policies and logistics for their original forums.

[6]Different seasons of the same TV shows have different ASIN codes but share the same reviews.

[7]If the number of total votes between the last snapshot of the early fragment and the first snapshot of the later fragment is less than 3, we fill in the missing information simply with the last snapshot of the earlier fragment.

[8]Dataset and statistics are available at `https://archive.org/details/stackexchange`.

[9]The answer selected by the question owner is displayed first regardless of voting scores.

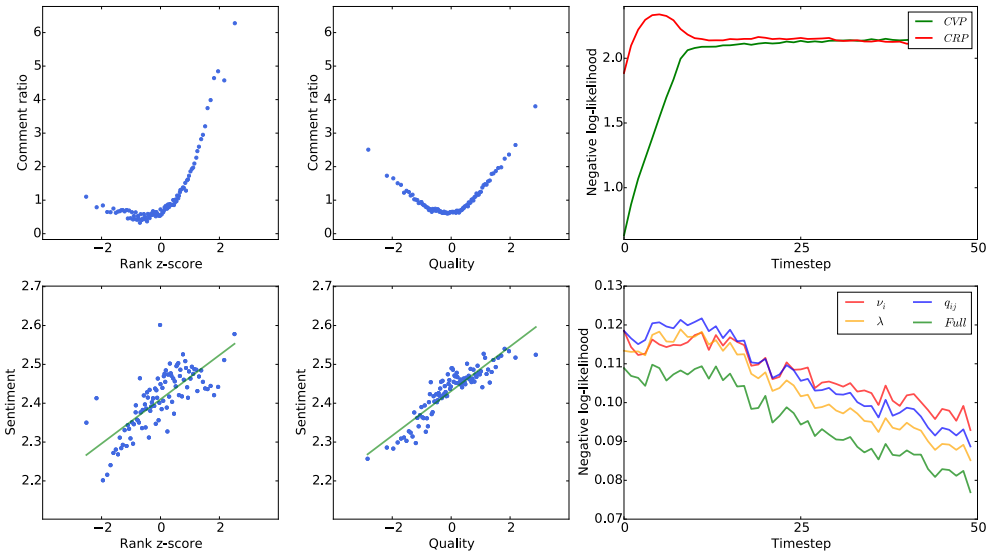


Figure 2: Comment and likelihood analysis on the StackOverflow forum. The left panels show that responses with higher ranks tend to have more comments (top) and more positive sentiments (bottom). The middle panels show responses have more comments at both high and low intrinsic quality q_{ij} (top). The corresponding sentiment correlates more cohesively with the quality score (bottom). Each blue dot is approximately an average over 1k responses, and we parse 337k comments given on 104k responses in total. The right panels show predictive power for the selection phase (top) and the voting phase (bottom) up to $t < 50$ (lower is better).

Predictive analysis. In each community, our prediction task is to learn the model up to time t and predict the action at $t + 1$. We align all items at their initial time steps and compute the average negative log-likelihood of the next actions based on the current model. Since the complete trajectory enables us to separate the selection and voting phases in inference, we also measure the predictive power of these two tasks separately against their own baselines. For the selection phase, the baseline is the CRP, which selects responses proportional to the number of accumulated votes or writes a new response with the probability proportional to α.[10] When $t < 50$, as shown in the first column of Table 5, the CVP significantly outperforms the CRP based on paired t-tests (two-tailed). Using the function f based on display rank and Trendiness parameter τ is indeed a more precise representation of positional accessibility. Especially in the early stages, users often select responses displayed at lower ranks with fewer votes. While the CRP has no ability to give high scores in these cases, the CVP properly models it by decreasing τ. The comparative advantage of the CVP declines as more votes become available and the correlation between display rank and the number of votes increases. For items with $t \geq 50$, there is no significant difference between the two models as exemplified in the third column of Figure 2. These results are coherent across other communities ($p > 0.07$).

Improving predictive power on the voting phase is difficult because positive votes dominate in every community. We compare the fully parametrized model to simpler partial models in which certain parameters are set to zero. For example, a model with all parameters but λ knocked out is comparable to a plain Pólya Urn. As illustrated in the second column of Table 5, we verify that every sub-model is significantly different from the full model in all major communities based on one-way ANOVA test, implying that each feature adds distinctive and meaningful information. Having the item-specific length bias ν_i provides significant improvements as well as having intrinsic quality q_{ij} and current opinion counts λ. While we omit the log-likelihood results with $t \geq 50$, all model better predicts true polarity when $t \geq 50$, because the log-linear model obtains a more robust estimate of community-level parameters as the model acquires more training samples.

Quality analysis. The primary advantage of the CVP is its ability to learn "intrinsic quality" for each response that filters out noise from self-reinforcing voting processes. We validate these scores by comparing them to another source of user feedback: both StackExchange and Amazon allow users to attach comments to responses along with votes. For each response, we record the number of comments and the average sentiment of those comments as estimated by [17]. As a baseline, we

[10] We fix α to 0.5 after searching over a wide range of values.

also calculate the final display rank of each response, which we convert to a z-score to make it more comparable to the quality scores q_{ij}. After sorting responses based on display rank and quality rank, we measure the association between the two rankings and comment sentiment with linear regression. Results are shown for StackOverflow in Figure 2. As expected, highly-ranked responses have more comments, but we also find that there are more comments for both high *and* low values of intrinsic quality. Both better display rank and higher quality score q_{ij} are clearly associated with more positive comments (slope $\in$ [0.47, 0.64]), but the residuals of quality rank 0.012 are on average less than the half the residuals of display rank 0.028. In addition, we also calculate the "bumpiness" of these plots by computing the mean variation of two consecutive slopes between each adjacent pair of data points. Quality rank reduces bumpiness of display rank from 0.391 to 0.226 in average, implying the estimated intrinsic quality yields locally consistent ranking as well as globally consistent.[11]

Community analysis. The 2D embedding in Figure 1 shows that we can compare and contrast the different evaluation cultures of communities using two inferred behavioral coefficients: Trendiness τ and Conformity κ. Communities are sized according to the number of items and colored based on a manual clustering. Related communities collocate in the same neighborhood. Religion, scholarship, and meta-discussions cluster towards the bottom left, where users are interested in many different opinions, and are happy to disagree with each other. Going from left to right, communities become more trendy: users in trendier communities tend to select and vote mostly on already highly-ranked responses. Going from bottom to top, users become increasingly likely to conform to the majority opinion on any given response. By comparing related communities we can observe that characteristics of user communities determine voting behavior more than technical similarity. Highly theoretical and abstract communities (*cstheory*) have low Trendiness but high Conformity. More applied, but still graduate-level, communities in similar fields (*cs, mathoverflow, stats*) show less Conformity but greater Trendiness. Finally, more practical homework-oriented forums (*physics, math*) are even more trendy. In contrast, users in *english* are trendy and debatable. Users in *Amazon* are most sensitive to trendy reviews and least afraid of voicing minority opinion.

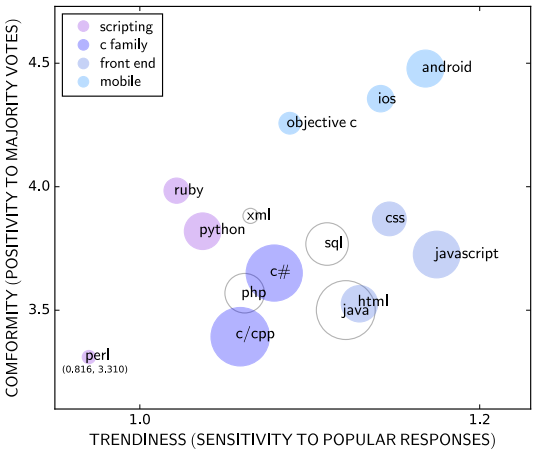


Figure 3: Sub-community embedding for StackOverflow.

StackOverflow is by far the largest community, and it is reasonable to wonder whether the Trendiness parameter is simply a proxy for size. When we subdivide StackOverflow by programming languages however (see Figure 3), individual community averages can be distinguished, but they all remain in the same region. *Javascript* programmers are more satisfied with trendy responses than those using *c/c++*. Mobile developers tend to be more conformist, while *Perl* hackers are more likely to argue.

5 Conclusions

Helpfulness voting is a powerful tool to evaluate user-generated responses such as product reviews and question answers. However such votes can be socially reinforced by positional accessibility and existing evaluations by other users. In contrast to many exchangeable random processes, the CVP takes into account sequences of votes, assigning different weights based on the *context that each vote was cast*. Instead of trying to model the response ordering function f, which is mechanism-specific and often changes based on service providers' strategies, we leverage the fully observed trajectories of votes, estimating the hidden intrinsic quality of each response and inferring two behavioral coefficients for community-level exploration. The proposed log-linear urn model is capable of generating non-exchangeable votes with great scalability to incorporate other factors such as length bias or other textual features. As we are more able to observe social interactions *as they are occurring* and not just summarized after the fact, we will increasingly be able to use models beyond exchangeability.

[11] All numbers and p-values in paragraphs are weighted averages on all 83 communities, whereas Table 5 only includes results for the major communities and their own weighted averages due to space limits.

References

[1] D. J. Aldous. Exchangeability and related topics. In *École d'Été St Flour 1983*, pages 1–198. Springer-Verlag, 1985.

[2] D. Blei, T. Griffiths, M. Jordan, and J. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing System*, NIPS '03, 2003.

[3] D. M. Blei and P. I. Frazier. Distance dependent chinese restaurant processes. *Journal of Machine Learning Learning Research*, pages 2461–2488, 2011.

[4] C. Danescu-Niculescu-Mizil, G. Kossinets, J. Kleinberg, and L. Lee. How opinions are received by online communities: A case study on Amazon.Com helpfulness votes. In *Proceedings of World Wide Web*, WWW '09, pages 141–150, 2009.

[5] A. Ghose and P. G. Ipeirotis. Designing novel review ranking systems: Predicting the usefulness and impact of reviews. In *Proceedings of the Ninth International Conference on Electronic Commerce*, ICEC '07, pages 303–310, 2007.

[6] T. Joachims, L. Granka, B. Pan, H. Hembrooke, F. Radlinski, and G. Gay. Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search. *ACM Transactions on Information Systems*, 25(2), 2007.

[7] S.-M. Kim, P. Pantel, T. Chklovski, and M. Pennacchiotti. Automatically assessing review helpfulness. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, EMNLP '06, 2006.

[8] J. Liu, Y. Cao, C.-Y. Lin, Y. Huang, and M. Zhou. Low-quality product review detection in opinion summarization. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, EMNLP-CoNLL '07, pages 334–342, 2007.

[9] L. Mamykina, B. Manoim, M. Mittal, G. Hripcsak, and B. Hartmann. Design lessons from the fastest q&a site in the west. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, 2011.

[10] L. Martin and P. Pu. Prediction of helpful reviews using emotion extraction. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI '14, pages 1551–1557, 2014.

[11] J. Otterbacher. 'helpfulness' in online communities: A measure of message quality. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '09, pages 955–964, 2009.

[12] M. J. Salganik, P. S. Dodds, and D. J. Watts. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311:854–856, 2006.

[13] M. J. Salganik and D. J. Watts. Leading the herd astray: An experimental study of self-fulfilling prophecies in an artificial cultural mmrket. *Social Psychology Quarterly*, 71:338–355, 2008.

[14] W. Shandwick. Buy it, try it, rate it: Study of consumer electronics purchase deicisions in the engagement era. *KRC Research*, 2012.

[15] S. Siersdorfer, S. Chelaru, J. S. Pedro, I. S. Altingovde, and W. Nejdl. Analyzing and mining comments and comment ratings on the social web. *ACM Trans. Web*, pages 17:1–17:39, 2014.

[16] R. Sipos, A. Ghosh, and T. Joachims. Was this review helpful to you?: It depends! context and voting patterns in online content. In *International Conference on World Wide Web*, WWW '14, pages 337–348, 2014.

[17] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, EMNLP, pages 1631–1642. Association for Computational Linguistics, 2013.

[18] Y. R. Tausczik, A. Kittur, and R. E. Kraut. Collaborative problem solving: A study of mathoverflow. In *Computer-Supported Cooperative Work and Social Computing*, CSCW' 14, 2014.

[19] Y. Yue, R. Patel, and H. Roehrig. Beyond position bias: Examining result attractiveness as a source of presentation bias in clickthrough data. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 2010.