A Omitted expressions

Derivation of interim belief update. Given user consent decisions, the agent updates the prior μ to form interim beliefs as follows. For a user i who provided consent ($x_i = 1$), with its revealed cookie $\phi_i = \varphi$, the interim belief $\tilde{\mu}_{i,0} \in \Delta(\Theta)$ is given elementwise by

$$\begin{aligned}\tilde{\mu}_{i,0}(\vartheta) &= \mathbb{P}(\theta_i = \vartheta \mid x_i = 1, \phi_i = \varphi) \\ &= \frac{\mathbb{P}(x_i = 1 \mid \phi_i = \varphi, \theta_i = \vartheta)\mathbb{P}(\phi_i = \varphi, \theta_i = \vartheta)}{\mathbb{P}(x_i = 1, \phi_i = \varphi)} \\ &= \frac{P(x_i = 1 \mid \theta_i = \vartheta)p(\phi_i = \varphi, \theta_i = \vartheta)}{\sum_{\vartheta'} \mathbb{P}(x_i = 1 \mid \theta_i = \vartheta')\mathbb{P}(\phi_i = \varphi, \theta_i = \vartheta')} \\ &= \frac{q_\vartheta \mu(\varphi, \vartheta)}{\sum_{\vartheta'} q_{\vartheta'} \mu(\varphi, \vartheta')}.\end{aligned}$$

Similarly, for a user i who did not provide consent ($x_i = 0$), its cookie is not revealed and thus interim beliefs are given by

$$\begin{aligned}\tilde{\mu}_{i,0}(\vartheta) &= \mathbb{P}(\theta_i = \vartheta \mid x_i = 0) \\ &= \frac{\mathbb{P}(x_i = 0 \mid \theta_i = \vartheta)\mathbb{P}(\theta_i = \vartheta)}{\mathbb{P}(x_i = 0)} \\ &= \frac{\mathbb{P}(x_i = 0 \mid \theta_i = \vartheta)\sum_{\varphi} \mathbb{P}(\phi_i = \varphi, \theta_i = \vartheta)}{\sum_{\vartheta'} \mathbb{P}(x_i = 0 \mid \theta_i = \vartheta')\sum_{\varphi'} \mathbb{P}(\phi_i = \varphi', \theta_i = \vartheta')} \\ &= \frac{(1 - q_\vartheta)\sum_{\varphi} \mu(\varphi, \vartheta)}{\sum_{\vartheta'}(1 - q_{\vartheta'})\sum_{\varphi'} \mu(\varphi', \vartheta')}.\end{aligned}$$

Cohort belief update. Given a set of cohort beliefs across users $\mu = (\mu_1, \ldots, \mu_n)$ and a set of recommendation-response pairs $\mathcal{L}$, the agent forms updated beliefs $\mu' = (\mu'_1, \ldots, \mu'_n)$ via a Bayesian update, $\mu' = f(\mu, \mathcal{L})$. This update is carried out independently for each user i. Let μ_i denote the agent's current belief on user i' cohort θ_i, and denote $\mathcal{L}_i = \{(a_i, c_{i,a})\}$ as the pairs that correspond to user i. The updated belief μ'_i is given by $\mu'_i = (\mu'_i(\theta^1), \ldots, \mu'_i(\theta^d)) = f_i(\mu_i, \mathcal{L}_i)$ where each $\mu'_i(\vartheta)$ is given by

$$\begin{aligned}\mu'_i(\vartheta) &= \mathbb{P}(\theta_i = \vartheta \mid \mathcal{L}_i) \\ &= \frac{\prod_{(a_i, c_{i,a}) \in \mathcal{L}_i} \mathbb{P}(C_{i,a} = c_{i,a} \mid A_i = a_i, \theta_i = \vartheta)\mathbb{P}(\theta_i = \vartheta)}{\sum_{\vartheta'} \prod_{(a_i, c_{i,a}) \in \mathcal{L}_i} \mathbb{P}(C_{i,a} = c_{i,a} \mid A_i = a_i, \theta_i = \vartheta')\mathbb{P}(\theta_i = \vartheta')} \\ &= \frac{\prod_{(a_i, c_{i,a}) \in \mathcal{L}_i^1} \bar{p}_{i,a_i}(\vartheta) \prod_{(a_i, c_{i,a}) \in \mathcal{L}_i^0}(1 - \bar{p}_{i,a_i}(\vartheta))\mu_i(\vartheta)}{\sum_{\vartheta'} \prod_{(a_i, c_{i,a}) \in \mathcal{L}_i^1} \bar{p}_{i,a_i}(\vartheta') \prod_{(a_i, c_{i,a}) \in \mathcal{L}_i^0}(1 - \bar{p}_{i,a_i}(\vartheta'))\mu_i(\vartheta')}\end{aligned}$$

where $\mathcal{L}_i^1$ (resp. $\mathcal{L}_i^0$) are the responses where the user clicked (resp. did not click) on the recommendation and $\bar{p}_{i,a}(\vartheta)$ is the expected click probability given ϑ defined as

$$\bar{p}_{i,a}(\vartheta) = \mathbb{E}_{\alpha_i \sim \text{LogNormal}(\rho_\vartheta, \sigma_\vartheta^2)}[p_{i,a}(\alpha_i)]$$

where the click probability $p_{i,a}$ from (1) has been written as $p_{i,a}(\alpha_i)$ to make the dependence on user i's topic affinities explicit.

Confidence weights. The confidence weights $\bar{w}_{i,a,t}$ in (2) are computed as the expected binomial probability for seeing the current response counts $r_{i,a,t}$ given the agent's current belief on user i's cohort ϑ_i, i.e., $\bar{w}_{i,a,t} = \mathbb{E}_{\vartheta_i \sim \mu_{i,t}}[p_t(I_{i,a,t}, r_{i,a,t}, \vartheta_i)]$. The binomial probability $p_t(I_{i,a,t}, r_{i,a,t}, \vartheta_i)$ is defined as

$$p_t(I_{i,a,t}, r_{i,a,t}, \vartheta_i) = \binom{I_{i,a,t}}{r_{i,a,t}} \bar{p}_{i,a}(\vartheta_i)^{r_{i,a,t}}(1 - \bar{p}_{i,a}(\vartheta_i))^{I_{i,a,t} - r_{i,a,t}}$$

where $I_{i,a,t}$ is the current impression count for user-ad pair (i, a), $r_{i,a,t}$ is the current click click for (i, a), and $\bar{p}_{i,a}(\vartheta_i) = \mathbb{E}_{\alpha_i \sim \text{LogNormal}(\rho_{\vartheta_i}, \sigma_{\vartheta_i}^2)}[p_{i,a}(\alpha_i)]$ is the expected click probability.

B Recommendation procedure

Algorithm 1: Recommendation procedure.

Input parameters. $(n, l, m, \Theta, \Phi, \mathcal{T}, \{\rho_\theta, \sigma_\theta\}, u_0, \{q_\theta\}, \mu, k, \lambda, \varepsilon, \mathcal{L}_0, T_b, T)$

Initialize users. For each user $i \in [n]$, sample:

- cookie-cohort pairs $(\phi_i, \theta_i) \sim \mu$
- topic affinities $\alpha_i \sim \text{LogNormal}(\rho_{\theta_i}, \sigma^2_{\theta_i})$
- consent decision $x_i \sim \text{Bernoulli}(q_{\theta_i})$

Form interim beliefs. For each user $i \in [n]$:

```
if x_i = 1 then                                        // user provided consent
    μ̃_i(ϑ) ← q_ϑ μ(φ_i, ϑ) / Σ_ϑ' q_ϑ' μ(φ_i, ϑ'),  ϑ ∈ Θ
else                                                   // user withheld consent
    μ̃_i(ϑ) ← (1 − q_ϑ) Σ_φ μ(φ, ϑ) / Σ_ϑ' (1 − q_ϑ') Σ_φ' μ(φ', ϑ'),  ϑ ∈ Θ
end
```

if $x_i = 1$ **then** // user provided consent

$\quad \tilde{\mu}_i(\vartheta) \leftarrow \frac{q_\vartheta \mu(\phi_i, \vartheta)}{\sum_{\vartheta'} q_{\vartheta'} \mu(\phi_i, \vartheta')}, \ \vartheta \in \Theta$

else // user withheld consent

$\quad \tilde{\mu}_i(\vartheta) \leftarrow \frac{(1-q_\vartheta) \sum_\varphi \mu(\varphi, \vartheta)}{\sum_{\vartheta'} (1-q_{\vartheta'}) \sum_{\varphi'} \mu(\varphi', \vartheta')}, \ \vartheta \in \Theta$

end

Offline responses. Form priors using $\mathcal{L}_0$: $\mu_{i,0} \leftarrow f(\tilde{\mu}_i, \mathcal{L}_0)$, $i \in [n]$

Online recommendations.

for $t = 1, \ldots, T$ **do**

Define current ad pool $\mathcal{A}_t$ by sampling l items uniformly without replacement from $\mathcal{T}$

if $\mod (t, T_b) = 0$ **then** // retraining step

Update cohort beliefs: $\mu_{i,t} \leftarrow f(\mu_{i,t-1}, \mathcal{L}_t \setminus \mathcal{L}_{t-T_b})$, $i \in [n]$

Compute weights: $\bar{w}_{i,a,t} \leftarrow \mathbb{E}_{\vartheta_i \sim \mu_{i,t}}[p_t(r_{i,a,t}, \vartheta_i)]$, $i \in [n]$

Update factor estimates:

$$(\hat{\boldsymbol{u}}_t, \hat{\boldsymbol{v}}_t) \leftarrow \text{argmin}_{(u,v) \in \mathcal{U} \times \mathcal{V}} \Bigg[\sum_{(a_i, c_{i,a}) \in \mathcal{L}_t} \bar{w}_{i,a,t} (u_i^\top v_a - r_{i,a,t})^2 + \lambda \Bigg(\sum_{i \in [n]} \|u_i\|_2 + \sum_{a \in [m]} \|v_a\|_2 \Bigg) \Bigg]$$

else

Propagate cohort beliefs: $\mu_t \leftarrow \mu_{t-1}$

Propagate factor estimates: $(\hat{\boldsymbol{u}}_t, \hat{\boldsymbol{v}}_t) \leftarrow (\hat{\boldsymbol{u}}_{t-1}, \hat{\boldsymbol{v}}_{t-1})$

end

Recommend ads: for each $i \in [n]$, recommend at $a_{i,t}$ via

$$a_{i,t} = \begin{cases} \text{argmax}_{a \in \mathcal{A}_t} \hat{u}_{i,t}^\top \hat{v}_{a,t} & \text{w.p. } 1 - \varepsilon \\ a \sim \text{U}(\mathcal{A}_t) & \text{w.p. } \varepsilon \end{cases}$$

Append responses: $\mathcal{L}_{t+1} \leftarrow \mathcal{L}_t \cup \{a_{i,t}, c_{a,i,t}\}$

end

C Simulator and experiments

Our simulator was built upon RecSim (21) (source code at https://github.com/emiehling/cookie-consent/). The high-level architecture of our simulator is illustrated in Fig. 1 of Section 4. Additional details (with references to objects in the source code) are provided below.

Advertisement and user samplers. The advertisement sampler object (AdvertisementSampler) defines the distribution of each ad feature, here assumed to simply be the ad's topic. Similarly, the user sampler object (UserStateSampler) defines the distribution of each user feature, described by the joint cookie-cohort prior, the opt-in distribution, and the statistics of the user's topic affinities. The ad sampler and user sampler objects are used to define a gym environment for the recommendation procedure (via MultiUserEnvironment and RecSimGymEnv). Ads are resampled in each round whereas users remain fixed for the duration of the episode.

Users and the recommender agent. Each user is described by the class RSUserModel. This class contains the user's choice model, i.e., the logit model dictating the binary click decision given the recommended ad (see the method simulate_response).

Given the collections of ads and users, the recommender agent makes recommendations according to an ε-greedy bandit (see the pseudocode in Section B).

Retraining consists of first updating the cohort beliefs (via the methods update_cohort_beliefs and get_click_probabilities, see Appendix A for the expressions), updating weights $\bar{w}_{i,a,t}$, and recomputing the matrix factor estimates (via get_estimated_factors, see (2)). Estimation is carried out via stochastic gradient descent with a learning rate of 0.01, regularization weight of 0.01, and a stopping threshold on the mean-squared error of $\varepsilon_{\text{thresh}} = 0.001$. Latent factors are assumed to be of dimension $k = 50$.

Experimental setup. Simulations were run in Python 3.8 on an Intel(R) Xeon(R) CPU E5-2667 v2 (3.30GHz). Unless otherwise stated, baseline parameters of the simulation environment were as follows: $n = 1000$ users, $m = 200$ ads, ad candidate size $l = 50$, batch size $T_b = 1$, offline response set $\mathcal{L}_0 = \varnothing$, exploration probability $\varepsilon = 0.1$, and binary cookie and cohort spaces. Simulations were averaged over 500 runs/episodes.

The experimental setup can be extended in a variety of directions to investigate additional interesting questions. One direction is to extend the feature description of the ads (beyond topic) to include features that reflect ad quality and location. This would enable studying how the recommender system treats minority populations (compared to majority populations). Additionally, augmenting the simulator with the ability to handle a changing user pool would allow for analysis of the cold start problem.

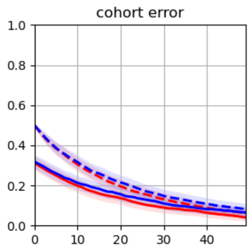
D Sensitivity analyses

Sensitivity analyses on cohort errors were carried out by independently varying the prior μ, batch size T_b, offline response set $\mathcal{L}_0$, and topic affinity means (ρ_0, ρ_1) with fixed parameters $n = 200$ users, $m = 200$ topics, ad pool size $l = 50$, latent factor dimension $k = 50$, and $|\Theta| = |\Phi| = 2$. Plots illustrate error means and standard deviations (red: $\theta = 0$, blue $\theta = 1$, solid: consent group, dashed: non-consent group) over 50 episodes. Baseline parameters are:

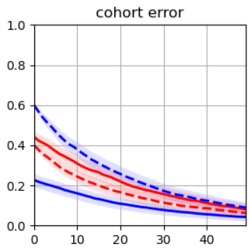
$$\mu = \begin{bmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{bmatrix}, \qquad T_b = 1, \qquad \mathcal{L}_0 = \varnothing, \qquad (\rho_0/m, \rho_1/m) = (0.3, 0.7)$$

Prior. The impact of the prior μ was studied in three cases dictated by the degree of informativeness of the cookie for inferring the cohort.

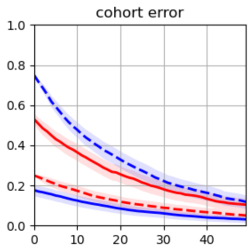
Partially informative: $\mu = \begin{bmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{bmatrix}$



$(q_0, q_1) = (0.25, 0.25)$

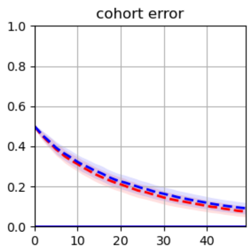


$(q_0, q_1) = (0.25, 0.5)$

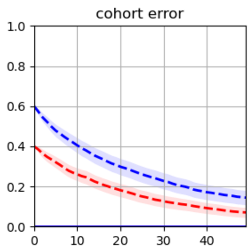


$(q_0, q_1) = (0.25, 0.75)$

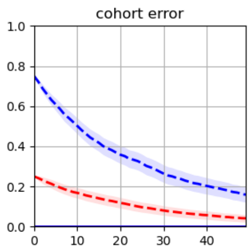
Fully informative: $\mu = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$



$(q_0, q_1) = (0.25, 0.25)$

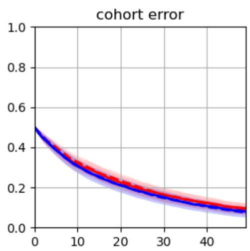


$(q_0, q_1) = (0.25, 0.5)$

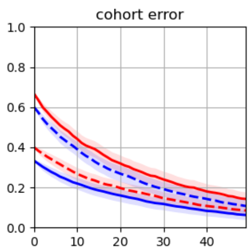


$(q_0, q_1) = (0.25, 0.75)$

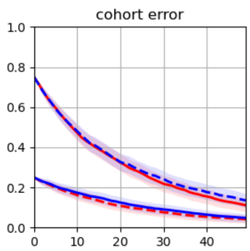
Uninformative: $\mu = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix}$



$(q_0, q_1) = (0.25, 0.25)$



$(q_0, q_1) = (0.25, 0.5)$

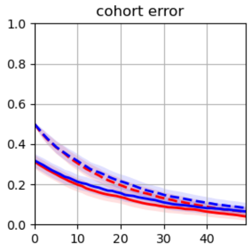


$(q_0, q_1) = (0.25, 0.75)$

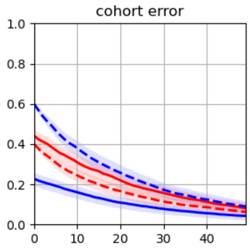
For a partially informative prior (any μ that does not have identical rows), knowledge of a user's cookie partially reveals the user's cohort. The resulting cohort errors are consistent with Fig. 2 of Section 5.1. For the fully informative prior, the agent is completely certain of users' cohorts for users who opted-in (the agree group), but still possesses uncertainty for users who did not opt-in (the disagree group). Lastly, for the uninformative prior, revelation of a user's cookie does not inform the user's cohort (as the likelihoods of seeing cookie values are identical across cohorts) and the agent must infer cohorts solely from differences in response behavior.

Batch size. The batch size, T_b, dictates how many responses to collect from each user before retraining. Values for the batch size were varied in the range $T_b \in \{1, 2, 10\}$.

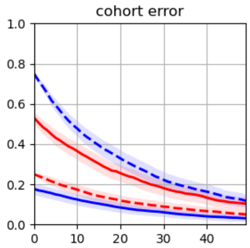
$T_b = 1$:



$(q_0, q_1) = (0.25, 0.25)$

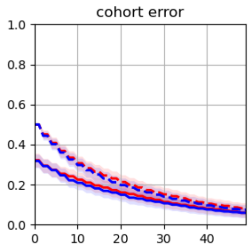


$(q_0, q_1) = (0.25, 0.5)$

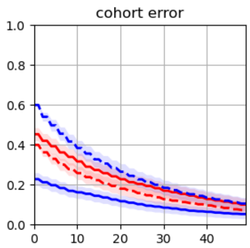


$(q_0, q_1) = (0.25, 0.75)$

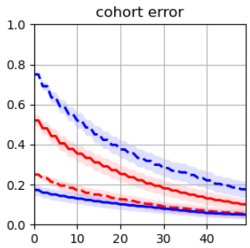
$T_b = 2$:



$(q_0, q_1) = (0.25, 0.25)$

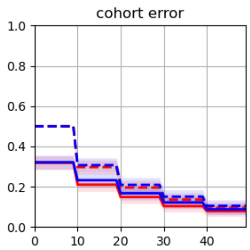


$(q_0, q_1) = (0.25, 0.5)$

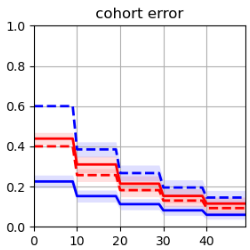


$(q_0, q_1) = (0.25, 0.75)$

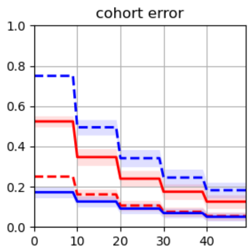
$T_b = 10$:



$(q_0, q_1) = (0.25, 0.25)$



$(q_0, q_1) = (0.25, 0.5)$

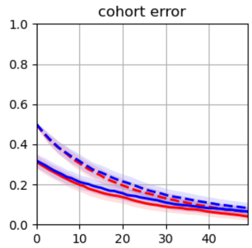


$(q_0, q_1) = (0.25, 0.75)$

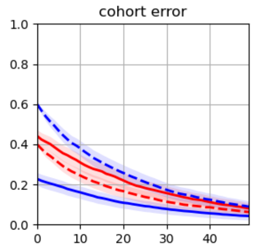
Intuitively, for the same number of observations, cohort estimation errors are the same across various batch sizes (e.g., updates may be less frequent but they contain more data). However, waiting until a batch update ($T_b > 1$) results in more interactions where the users face greater disparate estimation errors (compared to the $T_b = 1$ case).

Size of offline response set. The offline response set, $\mathcal{L}_0$, is a set of recommendation-responses that are available before the online recommendation process. Recommendations in the offline set, $\mathcal{L}_0$, were generated uniformly at random with responses generated by the users' choice models. Simulations were run for $|\mathcal{L}_0| = \{0, 1, 5\}$, differing in the number of offline responses assumed available from each user.

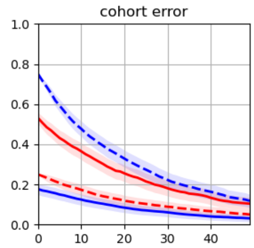
$|\mathcal{L}_0| = 0$:



$(q_0, q_1) = (0.25, 0.25)$

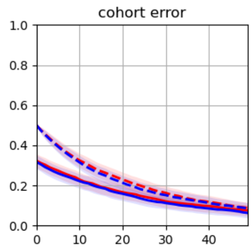


$(q_0, q_1) = (0.25, 0.5)$

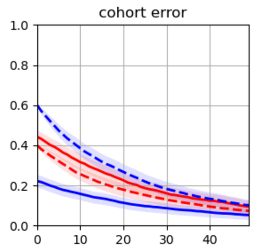


$(q_0, q_1) = (0.25, 0.75)$

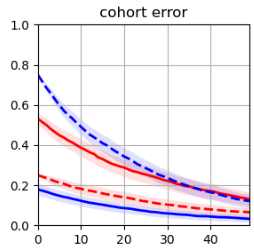
$|\mathcal{L}_0| = 1$:



$(q_0, q_1) = (0.25, 0.25)$

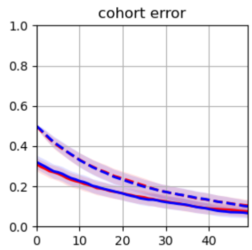


$(q_0, q_1) = (0.25, 0.5)$

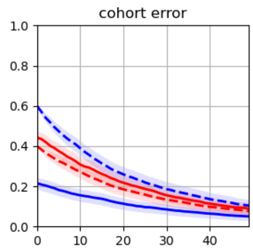


$(q_0, q_1) = (0.25, 0.75)$

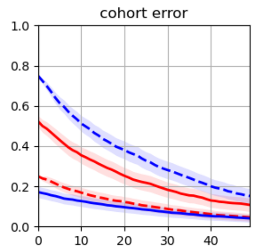
$|\mathcal{L}_0| = 5$:



$(q_0, q_1) = (0.25, 0.25)$



$(q_0, q_1) = (0.25, 0.5)$

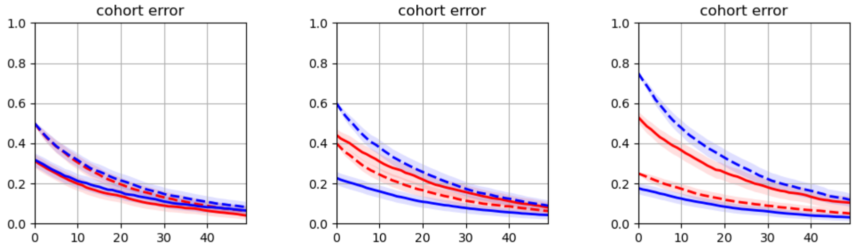


$(q_0, q_1) = (0.25, 0.75)$

The impact of the offline response set does not appear to have a significant effect on the evolution of the cohort errors.

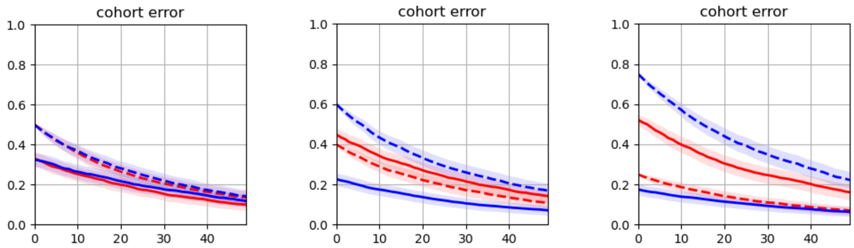
Similarity of topic affinities. The sensitivity to similarity of topic affinities was studied by increasing similarities of the synthetic affinities via means (normalized by topic count m) $(\rho_0/m, \rho_1/m) \in \{(0.3, 0.7), (0.4, 0.6), (0.5, 0.5)\}$.

$(\rho_0/m, \rho_1/m) = (0.3, 0.7)$:



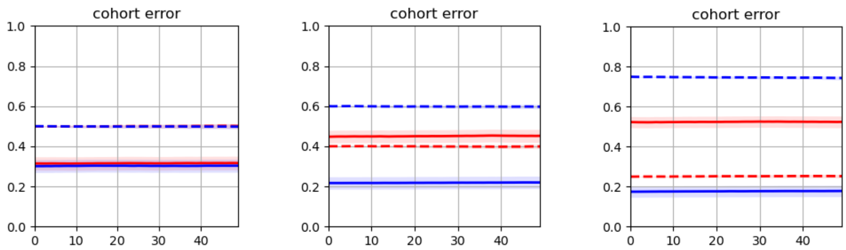
$(q_0, q_1) = (0.25, 0.25)$ $(q_0, q_1) = (0.25, 0.5)$ $(q_0, q_1) = (0.25, 0.75)$

$(\rho_0/m, \rho_1/m) = (0.4, 0.6)$:



$(q_0, q_1) = (0.25, 0.25)$ $(q_0, q_1) = (0.25, 0.5)$ $(q_0, q_1) = (0.25, 0.75)$

$(\rho_0/m, \rho_1/m) = (0.5, 0.5)$:



$(q_0, q_1) = (0.25, 0.25)$ $(q_0, q_1) = (0.25, 0.5)$ $(q_0, q_1) = (0.25, 0.75)$

The agent's ability to distinguish users based on their responses depends on the similarities of affinities across users in different cohorts. Intuitively, as the topic affinities across cohorts become more similar, the agent requires more responses to reach the same level of estimation error (since users from different cohorts behave more similarity as topic similarity grows). The extreme case of identical statistics of users' affinities across cohorts ($(\rho_0/m, \rho_1/m) = (0.5, 0.5)$) results in the agent being unable to resolve any uncertainty over users' cohorts (since user responses are uninformative for their cohort).