Proceedings of CONF-CDS 2026 Symposium: Data-Centric AI Security: Securing Models, Learning Agents, and Autonomous Systems
DOI: 10.54254/2755-2721/2026.CH34613

Preference Drift and Short-Term Signals in LinUCB-Based Movie Recommendation

Beitian Chen

School of Computing, National University of Singapore, Singapore, Singapore
selena.cbt0102@gmail.com

Abstract. Online recommendation systems need to balance exploration with exploitation while responding to changing user preferences. Linear Upper Confidence Bound (LinUCB) provides an efficient contextual-bandit framework. However, it usually relies on a relatively stable user-interest representation. This paper examines whether combining long-term and short-term genre-preference features improves LinUCB-based movie recommendation, and when short-term signals are beneficial under preference drift. Using MovieLens-1M, experiments are conducted on the 50 most active users. Each round presents one logged movie and nine sampled distractors. LinUCB-Fusion with $\alpha = 0.85$ achieves the highest average positive-feedback hit rate of 48.81%, compared with 48.42% for LinUCB-Long and 43.06% for Random. The average improvement over LinUCB-Long is small. A clearer pattern appears after splitting users by preference drift: Fusion helps high-drift users but hurts stable users. This suggests that short-term signals are mainly useful when user interests are changing, not for all users equally.

Keywords: Multi-Armed Bandit, LinUCB, Recommendation System, Preference Drift, Long-Term and Short-Term Interest

1. Introduction

Online recommendation systems need to balance exploitation of known user preferences with exploration of new ones, a problem that Multi-Armed Bandit (MAB) algorithms are designed to handle. Silva et al. report that contextual variants account for 43% of recent MAB recommender studies, showing the central role of contextual bandits in this field [1]. Among these methods, the LinUCB algorithm is a natural baseline owing to its closed-form online updates and ease of extension [2]. However, standard LinUCB usually assumes relatively stable user preferences. In a CHI field study with 316 real users, Leqi et al. found that reward distributions can shift even within a short interaction session of less than thirty minutes, motivating bandit recommenders that account for preference dynamics [3].

Existing responses to non-stationarity include recency-aware bandit methods and long-short interest models. Discounting older observations tracks recent preferences but may weaken the stability of long-term evidence [4]. Deep sequential recommenders such as TLSRec explicitly fuse long-term and short-term embeddings, but they are less aligned with lightweight per-interaction

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.CH34613

LinUCB updates [5]. Less is known about whether a simple LinUCB model can combine long-term and short-term signals effectively.

To test this idea, this study employs LinUCB-Fusion, a weighted combination of long-term and short-term genre-preference features. Using MovieLens-1M, it compares Fusion with Random, LinUCB-Long, LinUCB-Window, and LinUCB-Decay. The study further explores whether preference drift explains when short-term information helps. This study provides reference for future drift-aware LinUCB extensions, suggests caution when using fixed fusion weights, and highlights the need for per-user adaptation.

2. Related work

2.1. Contextual bandits in recommendation

MAB algorithms are widely used for recommendation under the exploration–exploitation trade-off. A systematic review by Silva et al. reports that contextual variants account for 43% of recent works, with Upper Confidence Bound (UCB) based algorithms comprising 45.8% of the exploration strategies [1]. Within this family, Li et al. propose LinUCB for personalized news recommendation and show a 12.5% click lift over a context-free bandit [2]. Pires et al. show that offline evaluation of linear bandit recommenders can over-favour exploitative policies, motivating the controlled candidate-set protocol used in this paper [6].

2.2. Non-stationary bandits and preference dynamics

A central assumption of the standard MAB formulation, which is that reward distributions remain stationary over time, has been challenged both empirically and algorithmically. Leqi et al. find in a comics recommendation field study that reward patterns can differ when the same arms are presented in different orders, suggesting short-horizon preference dynamics [3]. The study also reports heterogeneous dynamics across user types, which supports the user-level analysis in this paper. Algorithmically, Xu et al. extend LinUCB to piecewise-stationary user interests [7]. Han et al. propose Discounted Thompson Sampling [4]. Shen et al. introduce time-aware HyperBandit [8]. Meanwhile, Loh et al. study feature-rich non-stationary contextual bandits [9].

2.3. Long-term vs. short-term preference fusion

Deep sequential recommendation has also studied the separation between long-term and short-term user interests. Chen et al. propose TLSRec, a time-lag-aware model that captures global stability and local fluctuation through neural time gates [5]. Zheng et al. disentangle long- and short-term interest embeddings with contrastive learning and adaptive aggregation, while Cheng et al. model stage-wise user-interest evolution in news recommendation [10, 11]. These studies suggest that recent and historical interests should not always be treated in the same way. In contrast, this paper studies a simpler weighted fusion method as a controllable and computationally light baseline within LinUCB.

DOI: 10.54254/2755-2721/2026.CH34613

3. Methodology

3.1. Problem formulation

This paper formulates movie recommendation as a controlled K-armed contextual-bandit evaluation. At each round t, a user is assigned a candidate set A_t of K = 10 movies. The set contains the logged movie rated at that timestamp and nine distractors sampled from the same user's rated evaluation items. For each candidate $a \in A_t$, the policy observes a d-dimensional context vector $x_{t,a}$, selects one arm a_t, and receives a binary reward:

$$r_t = 1[\,\text{rating}_t \geq 4\,]. \tag{1}$$

where rating_t is the user's rating for the selected movie. The objective is to maximise cumulative reward over T evaluation rounds. Candidate sets are re-sampled at every round to avoid a fixed-distractor artefact, while the protocol remains an offline candidate-set evaluation with the usual replay limitations [6].

3.2. LinUCB with shared parameters

The policy adheres to the LinUCB upper-confidence principle of Li et al., but uses a shared d-dimensional parameter vector θ rather than the original disjoint per-arm parameterisation, since per-movie estimation would be sparse [2]. The model maintains a d × d matrix A and a d-dimensional vector b, initialised as A = I and b = 0, with the ridge-regression estimate:

$$\theta = A^{-1} b. \tag{2}$$

At each round, the upper-confidence-bound score of candidate a is computed as

$$p_{t,a} = x_{t,a}^{\top} \theta + \beta \sqrt{x_{t,a}^{\top} A^{-1} x_{t,a}}, \tag{3}$$

where the first term estimates reward and the second is the exploration bonus controlled by $\beta > 0$. The policy selects the candidate with the largest LinUCB score and updates A and b after observing the reward:

$$A \leftarrow A + x_{t,a_t} x_{t,a_t}^{\top}, \; b \leftarrow b + r_t x_{t,a_t}. \tag{4}$$

All experiments use $\beta = 1.0$.

3.3. Interest feature construction

Each context vector represents genre-level user–movie affinity. The user vector captures historical genre preference, and the movie vector is the candidate movie's normalised multi-hot genre representation. The context is defined as the Hadamard product:

$$x_{t,a} = u_t \odot m_a. \tag{5}$$

Thus the j-th component of the context is large only when the user prefers genre j and the candidate movie belongs to that genre.

DOI: 10.54254/2755-2721/2026.CH34613

The long-term feature aggregates the user's full history before round t. Let H_t denote all interactions observed before round t, where each interaction contains a movie genre vector and a rating from 1 to 5. The long-term vector is defined as the rating-weighted average:

$$u_t^{long} \propto \sum_{s=1}^{t-1} \frac{rating_s}{5} g_s, \tag{6}$$

and is then normalised to unit L1 norm. The rating divided by 5 preserves rating intensity while producing a stable estimate of long-run genre preference.

The short-term feature uses only the most recent N interactions, denoted H_t^N. Equivalently, the summation starts from $s = \max(1, t - N)$ and ends at $s = t - 1$:

$$u_t^{short} \propto \sum_{s=\max(1,t-N)}^{t-1} \frac{rating_s}{5} g_s, \tag{7}$$

again followed by unit L1 normalization. The default window is $N = 20$, with $N \in \{10, 20, 50\}$ evaluated in Section 4. A second recency baseline uses exponential decay:

$$u_t^{decay} \propto \sum_{s=1}^{t-1} \lambda^{t-1-s} \frac{rating_s}{5} g_s, \tag{8}$$

with $\lambda \in \{0.90, 0.95, 0.99\}$. This serves as the LinUCB analogue of Discounted Thompson Sampling [4].

3.4. Fixed-weight fusion

The proposed fusion strategy combines the long- and short-term features via a convex combination,

$$u_t^{fusion}(\alpha) \propto \alpha\, u_t^{long} + (1 - \alpha)\, u_t^{short}, \tag{9}$$

followed by L1 normalisation. The scalar $\alpha \in [0, 1]$ controls the long-term fraction, while β denotes the LinUCB exploration coefficient. Setting $\alpha = 1$ recovers LinUCB-Long, whereas $\alpha = 0$ recovers the short-term feature. A grid search over $\alpha \in \{0.0, 0.1, \ldots, 0.8, 0.85, 0.9, 1.0\}$ finds $\alpha = 0.85$ as the best value. Consequently, it is used in the main results and user-level analysis. Since the sweep and main comparison use the same population, this value is treated as exploratory.

4. Experiments

4.1. Dataset and preprocessing

Experiments are conducted on MovieLens-1M, which contains 1,000,209 five-point ratings from 6,040 users on 3,883 movies collected between 2000 and 2003. Movies are represented by $d = 18$ normalised genre tags. After merging metadata, 3,706 rated movies appear in the processed records. Users with fewer than 20 ratings are filtered, with no MovieLens-1M users removed by this threshold. Each user's timeline is sorted by timestamp and split 50/50 into warm-up and evaluation segments. The 50 most active users are used for all experiments, yielding 596.6 evaluation rounds per user on average and a shared minimum horizon of 478 rounds.

DOI: 10.54254/2755-2721/2026.CH34613

4.2. Evaluation protocol and baselines

At each evaluation round, the candidate set contains the logged movie and nine distractors sampled from the same user's rated evaluation items. Ratings of four or five stars are treated as positive feedback (reward 1), and lower ratings as reward 0. Hit rate is therefore the positive-feedback rate of selected candidates, not exact logged-item recovery. User-interest features use logged interactions before round t, while LinUCB parameters are updated only with the selected candidate's reward. Each method is evaluated over three random seeds (42, 123, 2024), producing 150 matched user-seed trajectories for paired tests.

Five methods are compared. Random selects uniformly from the candidate set. LinUCB-Long uses the full observed history, LinUCB-Window uses the best-performing sliding window ($N = 50$), LinUCB-Decay uses the best-performing discount factor ($\lambda = 0.99$), and LinUCB-Fusion uses the best fixed fusion weight ($\alpha = 0.85$). All LinUCB variants share the same context construction, $\beta = 1.0$, and matched candidate sets.

4.3. Main results

Table 1 reports average hit rate across the 150 trajectories. LinUCB-Fusion obtains the highest mean hit rate, 48.81%, improving over Random by +5.74 percentage points. LinUCB-Long is close behind at 48.42%, while LinUCB-Decay and LinUCB-Window reach 46.83% and 45.33%. Against LinUCB-Long, the recency-only baselines are worse ($p < 10^{-3}$), suggesting that short-term evidence alone is noisy under this genre-level representation. The Fusion-versus-Long difference is much smaller (+0.39 pp), so the average gain should be interpreted with caution.

Table 1. Mean hit rate of the five evaluated methods across 50 users × 3 random seeds

Method	Hit rate	Std	Δ vs Random	p (vs Long)
Random	43.06%	18.52%	—	—
LinUCB-Long	48.42%	20.12%	+5.36 pp	—
LinUCB-Window (N=50)	45.33%	18.86%	+2.27 pp	1.5×10^{-12}
LinUCB-Decay (λ=0.99)	46.83%	19.50%	+3.77 pp	2.0×10^{-4}
LinUCB-Fusion (α=0.85)	48.81%	19.86%	+5.74 pp	4.6×10^{-2}

4.4. Ablation studies

The window and decay ablations show that more memory helps the recency baselines. Window performance rises from 44.69% ($N = 10$) to 45.33% ($N = 50$), while decay improves from 45.16% ($\lambda = 0.90$) to 46.83% ($\lambda = 0.99$). This suggests that short-term signals alone are weaker than long-term preferences when using genre-level MovieLens features. The fusion sweep in Figure 1 presents a more mixed story.

DOI: 10.54254/2755-2721/2026.CH34613

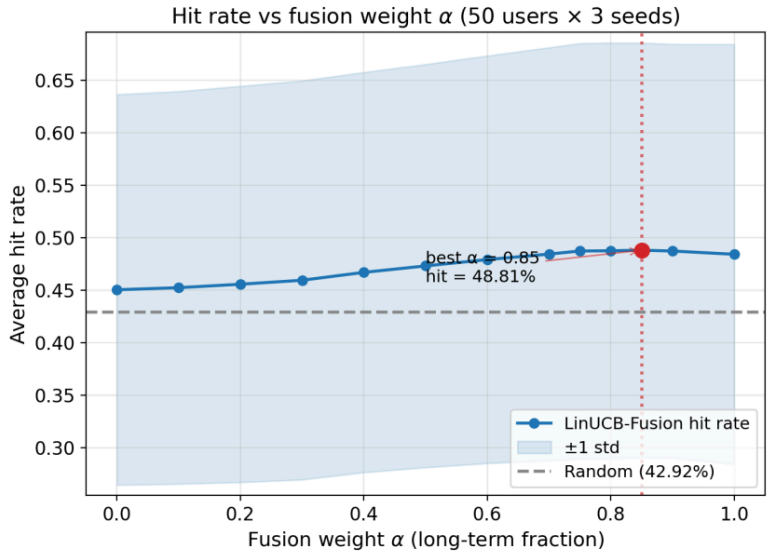

Figure 1. Average hit rate of LinUCB-Fusion as a function of the scalar fusion weight α (long-term fraction)

The shaded band shows ±1 standard deviation across the 50 user-level averages. The Random baseline from the same sweep (42.92%) is included as a visual reference. The main experiment reports a matched Random baseline of 43.06%.

Figure 1 shows that the α curve peaks at α = 0.85, with a 48.81% hit rate. Performance rises from 45.04% at α = 0 and falls slightly to 48.42% at α = 1. Overall, long-term preference remains the stronger signal, and adding a small short-term component helps slightly in this sweep. The aggregate result should be interpreted cautiously because the Fusion-versus-Long gap is small and α is tuned on the same assessment population. This motivates testing whether preference drift explains the mixed effect.

4.5. User-level heterogeneity

To examine these user-level differences, the analysis uses a simple drift score. For each user, the score is computed as the cosine distance between the rating-weighted genre vector of the first half and second half of the interaction sequence. The 50 users are then split at the median into Stable and Drifting groups of 25 users each. Since the score is computed after observing the full sequence, this study aims to explain observed heterogeneity rather than to claim that the same group label would be directly available before recommendation begins.

Figure 2 reveals that the population-average Fusion gain combines two opposite effects. In the Stable group, Fusion underperforms Long by about −0.52 percentage points ($p < 10^{-3}$, win rate 32%). In the Drifting group, Fusion outperforms Long about +1.29 percentage points ($p < 10^{-3}$, win rate 60%). The between-group interaction is approximately +1.81 percentage points and highly significant ($p < 10^{-6}$). This constitutes the key empirical finding: preference drift helps explain why the same short-term component has opposite effects across users.

DOI: 10.54254/2755-2721/2026.CH34613

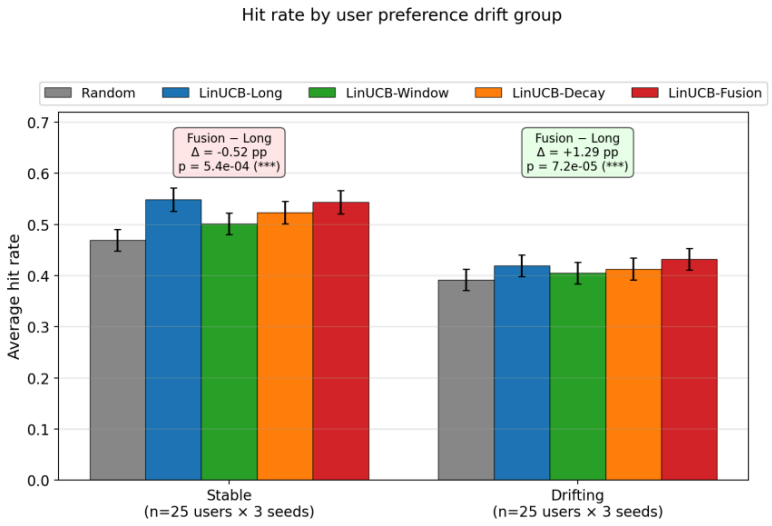

Figure 2. Hit rate by preference drift group. Annotations show the paired Fusion − Long difference and the corresponding paired t-test p-value within each group

5. Conclusion

This paper studies whether short-term genre-preference features add useful information to LinUCB-based movie recommendation. On MovieLens-1M, LinUCB-Fusion with $\alpha = 0.85$ achieved the highest mean hit rate at 48.81%. This is slightly higher than LinUCB-Long at 48.42%, Decay at 46.83%, and Window at 45.33%. However, the average gain over LinUCB-Long is small. Given that α is selected by an in-sample sweep and the paired tests are not fully consistent, this result should be interpreted cautiously. The main conclusion is therefore conditional: fixed-weight fusion is most promising for users with stronger preference drift.

The user-level analysis clarifies why the average result is modest. Short-term information is not equally useful for all users. For stable users, adding a short-term component can introduce unnecessary noise and weaken the reliable long-term preference signal. For drifting users, however, recent interactions provide information that the long-term vector alone cannot capture quickly enough. This suggests that preference drift should be treated as a user-level condition rather than as a uniform property of the whole population.

The results should also be considered within the scope of the study design. The fusion weight α is fixed at the population level, and drift grouping is computed after observing the full user sequence. The protocol also relies on logged MovieLens candidate sets rather than a fully closed-loop user simulation, and the genre-level features may miss richer session-level dynamics. Future work could investigate adaptive per-user α with an online drift estimator, use a held-out tuning/evaluation split, and explore neural contextual bandits with dimension-level gating similar to TLSRec.

References

[1] Silva, N., Werneck, H., Silva, T., Pereira, A. C. M., & Rocha, L. (2022). Multi-armed bandits in recommendation systems: A survey of the state-of-the-art and future directions. Expert Systems with Applications, 197, 116669. https: //doi.org/10.1016/j.eswa.2022.116669

[2] Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In Proceedings of the 19th International Conference on World Wide Web (pp. 661–670). https: //doi.org/10.1145/1772690.1772758

DOI: 10.54254/2755-2721/2026.CH34613

[3] Leqi, L., Zhou, G., Kilinc-Karzan, F., Lipton, Z. C., & Montgomery, A. L. (2023). A field test of bandit algorithms for recommendations: Understanding the validity of assumptions on human preferences in multi-armed bandits. In Proceedings of the CHI Conference on Human Factors in Computing Systems (pp. 1–16). Article 504. https: //doi.org/10.1145/3544548.3580670

[4] Han, Q., Wang, Y., & Zhu, L. (2023). Discounted Thompson sampling for non-stationary bandit problems. arXiv preprint arXiv: 2305.10718.

[5] Chen, L., Yang, N., & Yu, P. S. (2022). Time lag aware sequential recommendation. In Proceedings of the 31st ACM International Conference on Information and Knowledge Management (pp. 212–221). https: //doi.org/10.1145/3511808.3557473

[6] Pires, P. R., Azevedo, G. F., Campos, P. L., Sereicikas, R. T., & Almeida, T. A. (2025). Exploitation over exploration: Unmasking the bias in linear bandit recommender offline evaluation. In Proceedings of the 19th ACM Conference on Recommender Systems (pp. 1–10). https: //doi.org/10.1145/3705328.3748166

[7] Xu, X., Dong, F., Li, Y., He, S., & Li, X. (2020). Contextual-bandit based personalized recommendation with time-varying user interests. Proceedings of the AAAI Conference on Artificial Intelligence, 34(4), 6518–6525. https: //doi.org/10.1609/aaai.v34i04.6125

[8] Shen, C., Zhang, X., Wei, W., & Xu, J. (2023). HyperBandit: Contextual bandit with hypernetwork for time-varying user preferences in streaming recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (pp. 2239–2248). https: //doi.org/10.1145/3583780.3614921

[9] Loh, W. M., Sinha, S. K., Agarwal, A., & Poupart, P. (2026). A practical algorithm for feature-rich, non-stationary bandit problems. Transactions on Machine Learning Research.

[10] Zheng, Y., Gao, C., Chang, J., Niu, Y., Song, Y., Jin, D., & Li, Y. (2022). Disentangling long and short-term interests for recommendation. In Proceedings of the ACM Web Conference (pp. 2256–2267). https: //doi.org/10.1145/3485447.3512098

[11] Cheng, Z., Jin, Y., Zhang, Z., Chen, H., Duan, Z., & Wang, M. (2026). Modeling stage-wise evolution of user interests for news recommendation. arXiv preprint arXiv: 2603.10471.