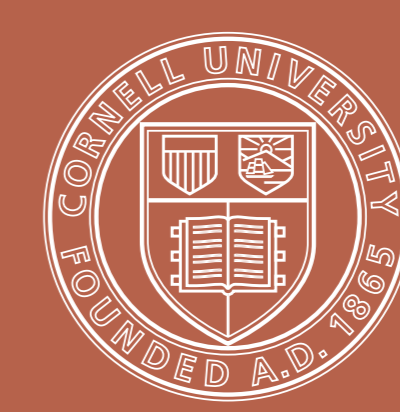


Stable Coactive Learning via Perturbation

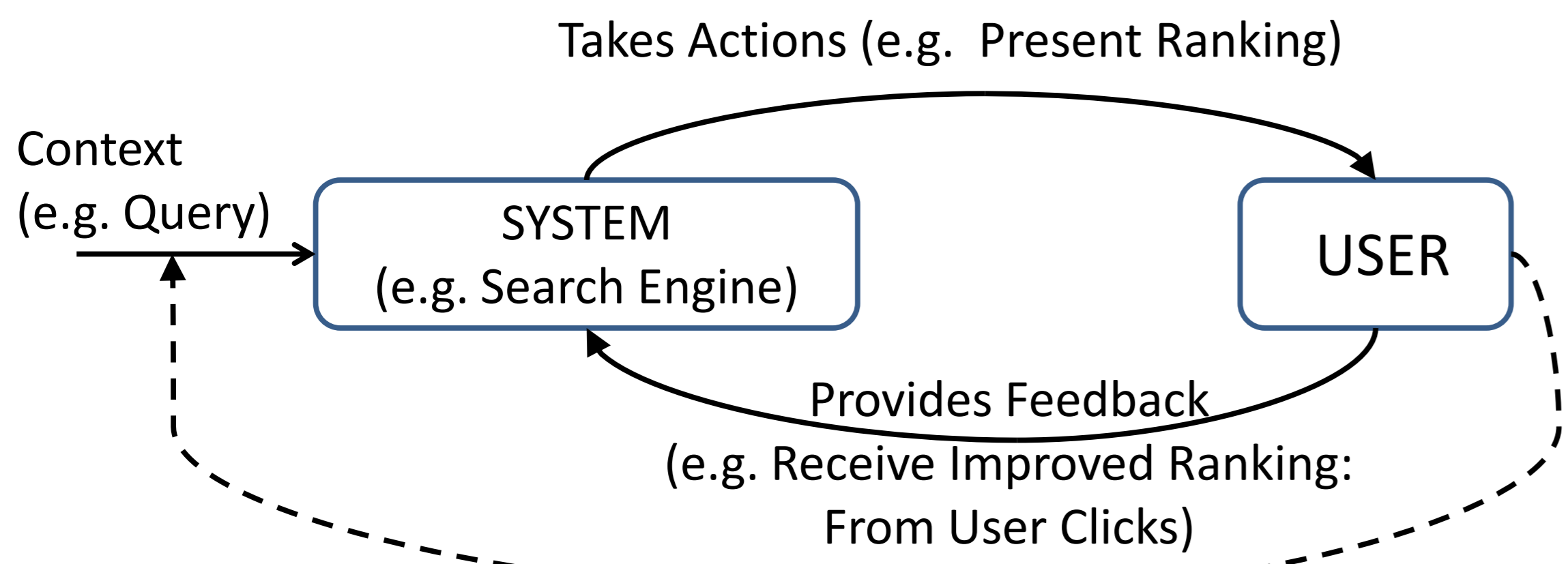
Karthik Raman, Thorsten Joachims (Cornell University)
Pannaga Shivaswamy (AT&T Research), Tobias Schnabel (University of Stuttgart)



Cornell University

Coactive Learning

- Coactive Learning:** [Shivaswamy and Joachims, 2012]
- Given context x , predict object y to optimize utility $U(x, y)$.
 - Models the interaction between user(s) and learning system.



Example: Using implicit feedback for ranking:



Instability of the Preference Perceptron

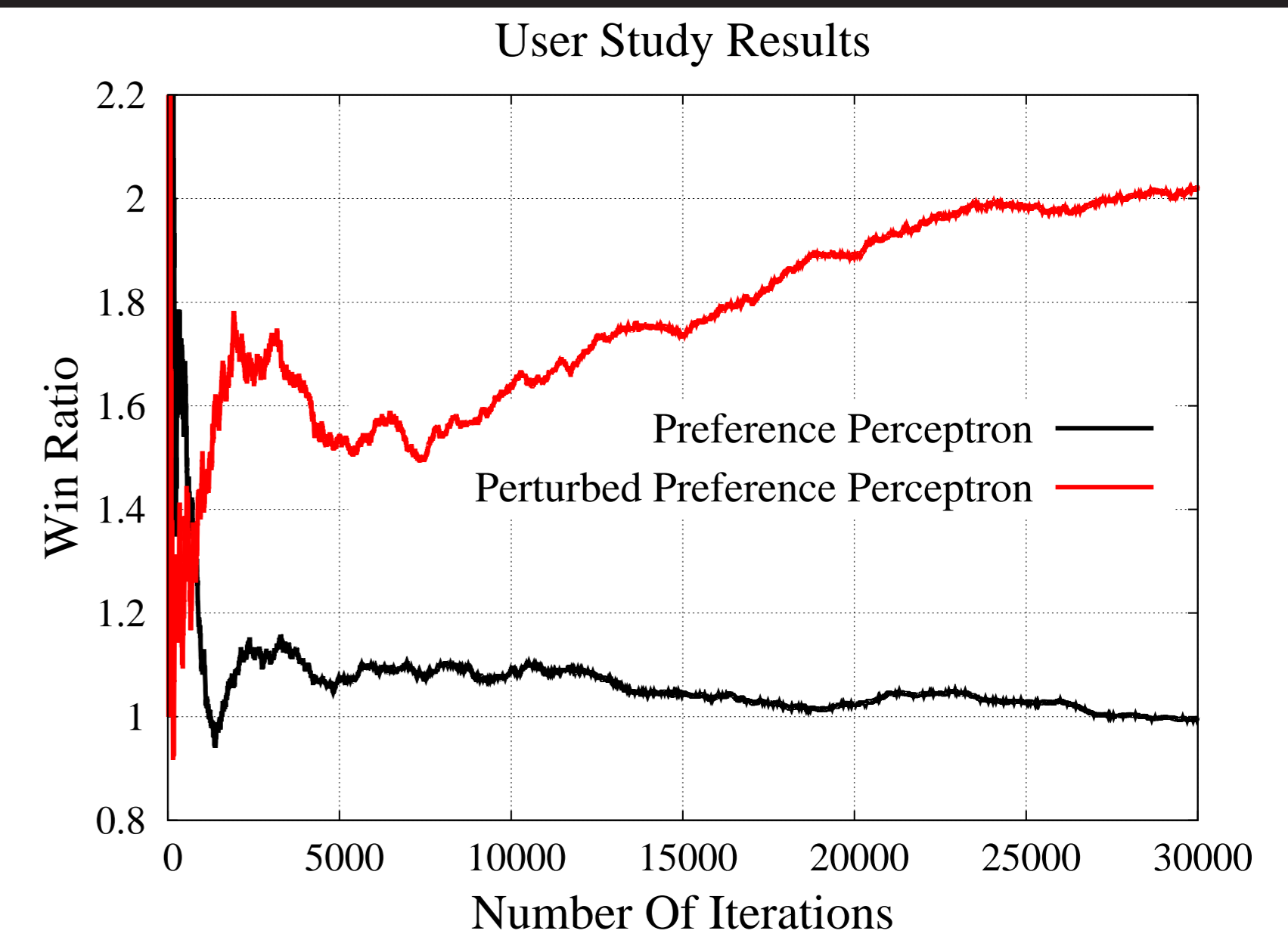
Linear Utility: $U(x, y) = w_*^T \phi(x, y)$

Preference Perceptron Algorithm: (Proposed in [Shivaswamy and Joachims, 2012])

- Initialize weight vector $w_1 \leftarrow 0$.
- Given context x_t present $y_t \leftarrow \text{argmax}_y w_t^T \phi(x_t, y)$.
- Move clicked documents to top to get feedback ranking \bar{y}_t .
- $w_{t+1} \leftarrow w_t + \phi(x_t, \bar{y}_t) - \phi(x_t, y_t)$.
- Repeat from step 2.

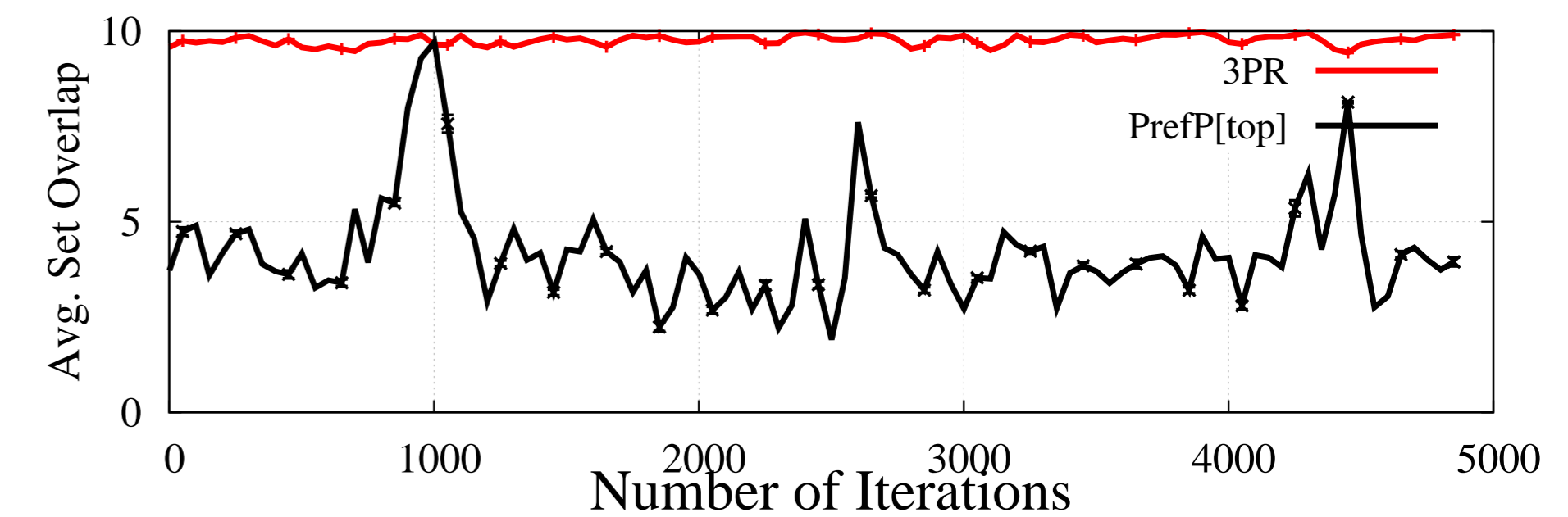
User Study:

- Experimented using live full-text search engine at arxiv.org.
- Goal: Learning a ranking function from implicit feedback *i.e.*, user clicks.
- Interleaved evaluation against hand-tuned baseline ranker.
- Win ratio of 1 indicates being no better than the baseline. Higher win ratio is better.



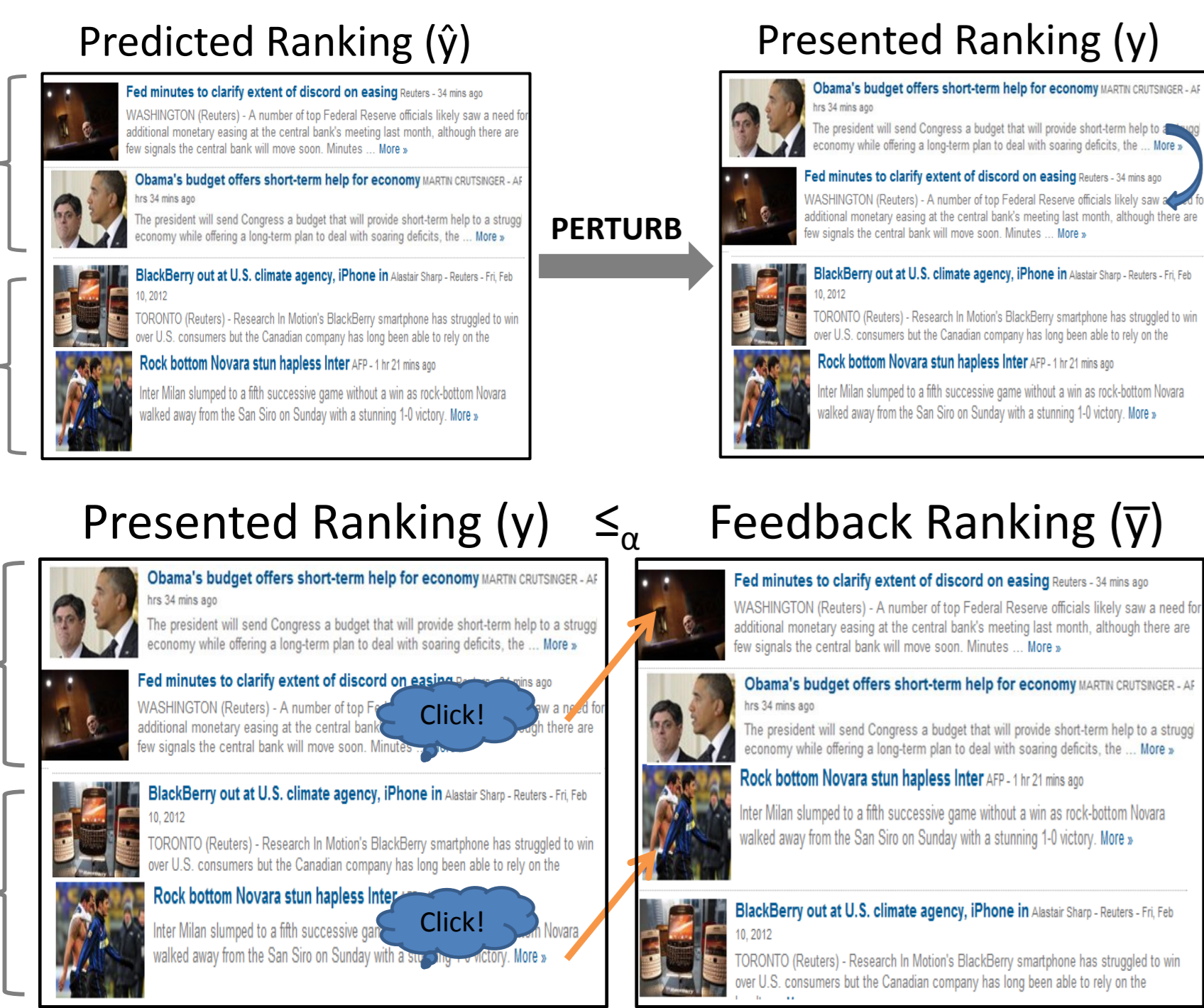
Preference Perceptron is unstable:

The rankings learned by PrefP never stabilize: Even after thousands of updates, the top 10 documents of the same query before and after 100 update steps only overlap by 4 documents.



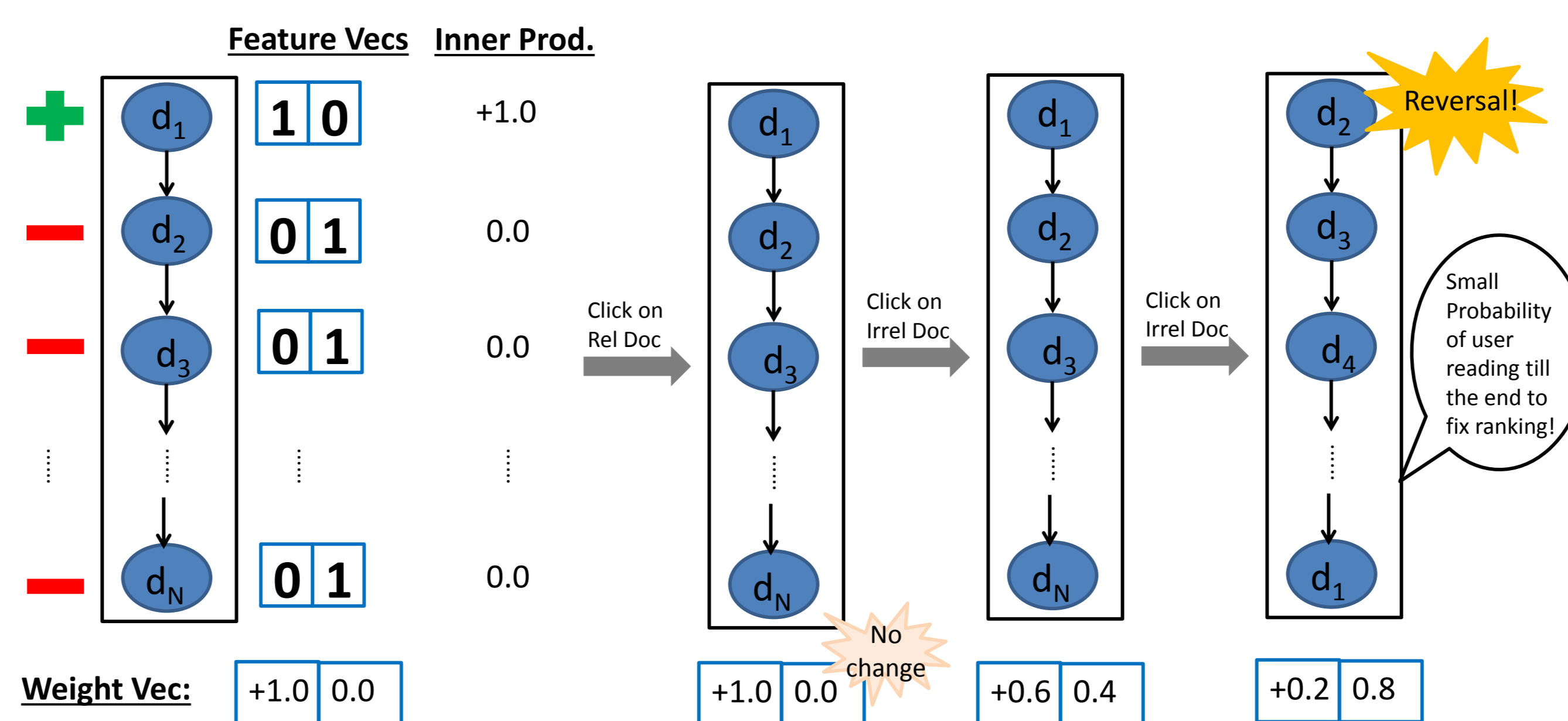
Perturbed Preference Perceptron

- Initialize weight vector $w_1 \leftarrow 0$.
- Given context x_t predict $\hat{y}_t \leftarrow \text{argmax}_y w_t^T \phi(x_t, y)$.
- Present y_t : Obtained by randomly swapping adjacent pairs in \hat{y}_t with probability p_t .
- Observe clicks. If clicked document is lower element of pair, move it up by one to get \bar{y}_t .
- $w_{t+1} \leftarrow w_t + \phi(x_t, \bar{y}_t) - \phi(x_t, y_t)$.
- Repeat from step 2.

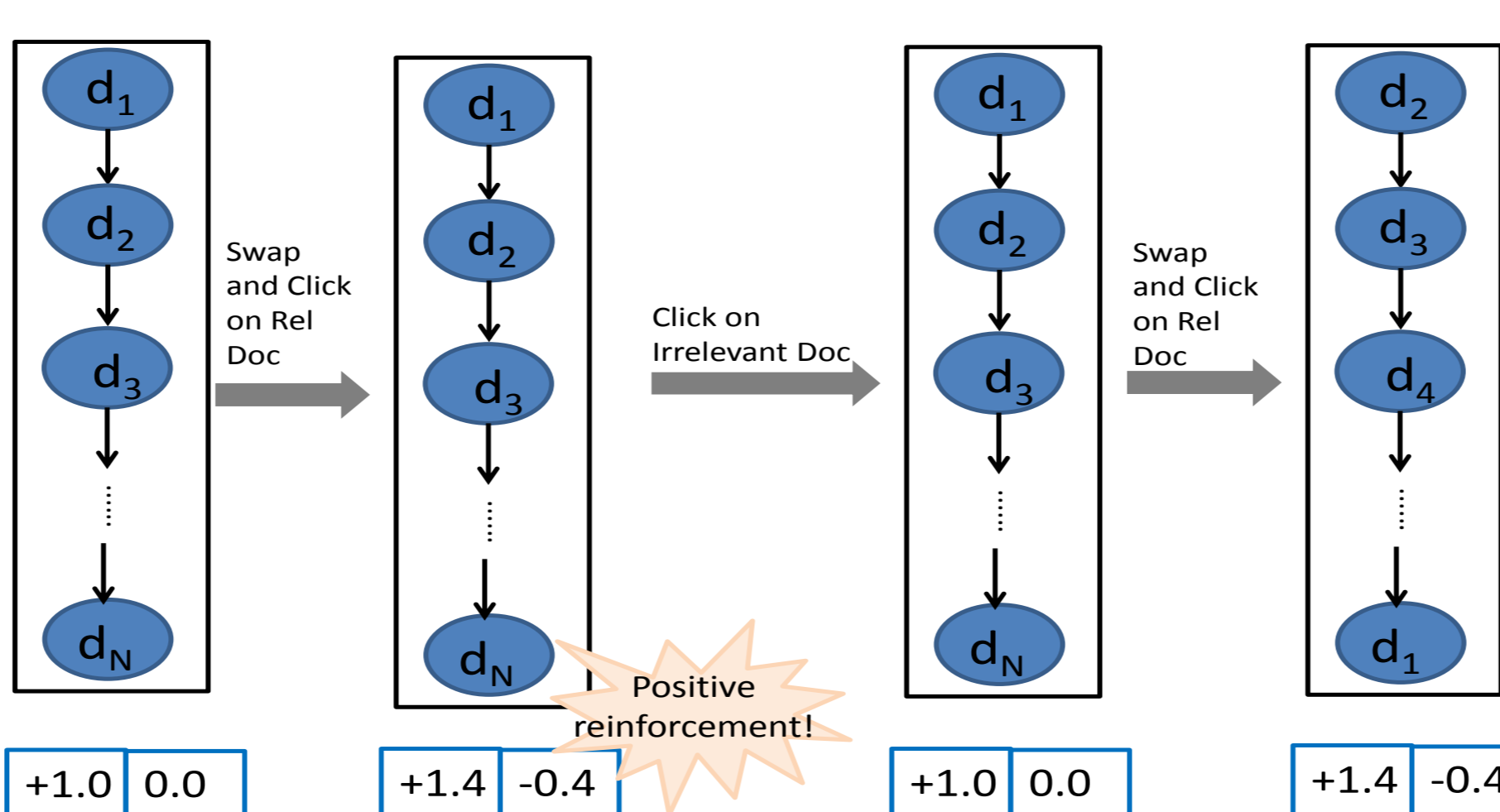


Illustrative Example

Preference Perceptron (PrefP)



Perturbed Preference Perceptron (3PR)



Theoretical Analysis

α -Informative Feedback:

We characterize the utility of the feedback received \bar{y}_t as:

$$E_{\bar{y}}[U(x, \bar{y})] \geq U(x, y) + \alpha(U(x, y^*) - U(x, y)) - \xi$$

- where y^* is the optimal and y is the presented object.
- Note that this is just a characterization (not an assumption).
- Used to prove regret bounds.

Regret:

We define the regret after T iterations as:

$$\frac{1}{T} \sum_{t=1}^T (U(x_t, y_t^*) - E[U(x_t, y_t)])$$

Fixed Probability 3PR

The regret of 3PR with fixed swap probability p (*i.e.* $\forall t: p_t = p$) is:

$$\leq \frac{\sum_{t=1}^T \xi_t}{\alpha T} + \frac{p(1 - \frac{22}{71})R \|w_*\|}{\alpha} + \sqrt{\frac{2(4 - p^2(1 - \frac{22}{71})^2)R \|w_*\|}{\alpha \sqrt{T}}}$$

Dynamic 3PR

For any $\Delta \geq 0$, dynamically setting the swap prob p_t of 3PR has regret:

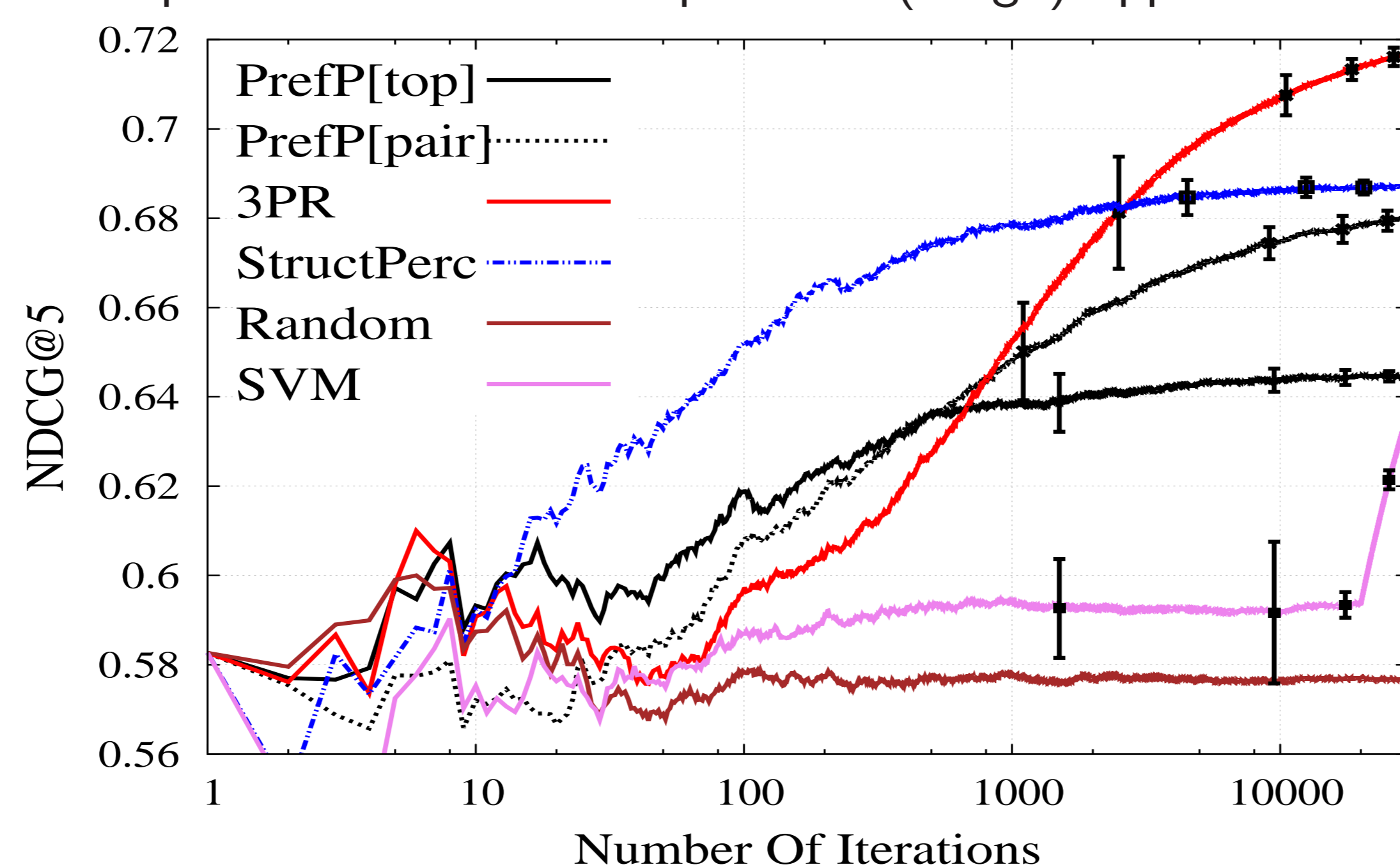
$$\leq \frac{1}{\alpha T} \sum_{t=1}^T \xi_t + \frac{\|w_*\|}{\alpha \sqrt{T}} \times \sqrt{4R^2 + 2\Delta + 2R \sqrt{\frac{4R^2 + 2\Delta}{T}}}$$

Experimental Results

- Performed offline experiments on a search dataset (Yahoo! LTR) and two news recommendation datasets: RCV1 and News.
- Simulated user behavior with and without noise.
- NDCG@5 was the utility for all three datasets.

Comparison with other methods:

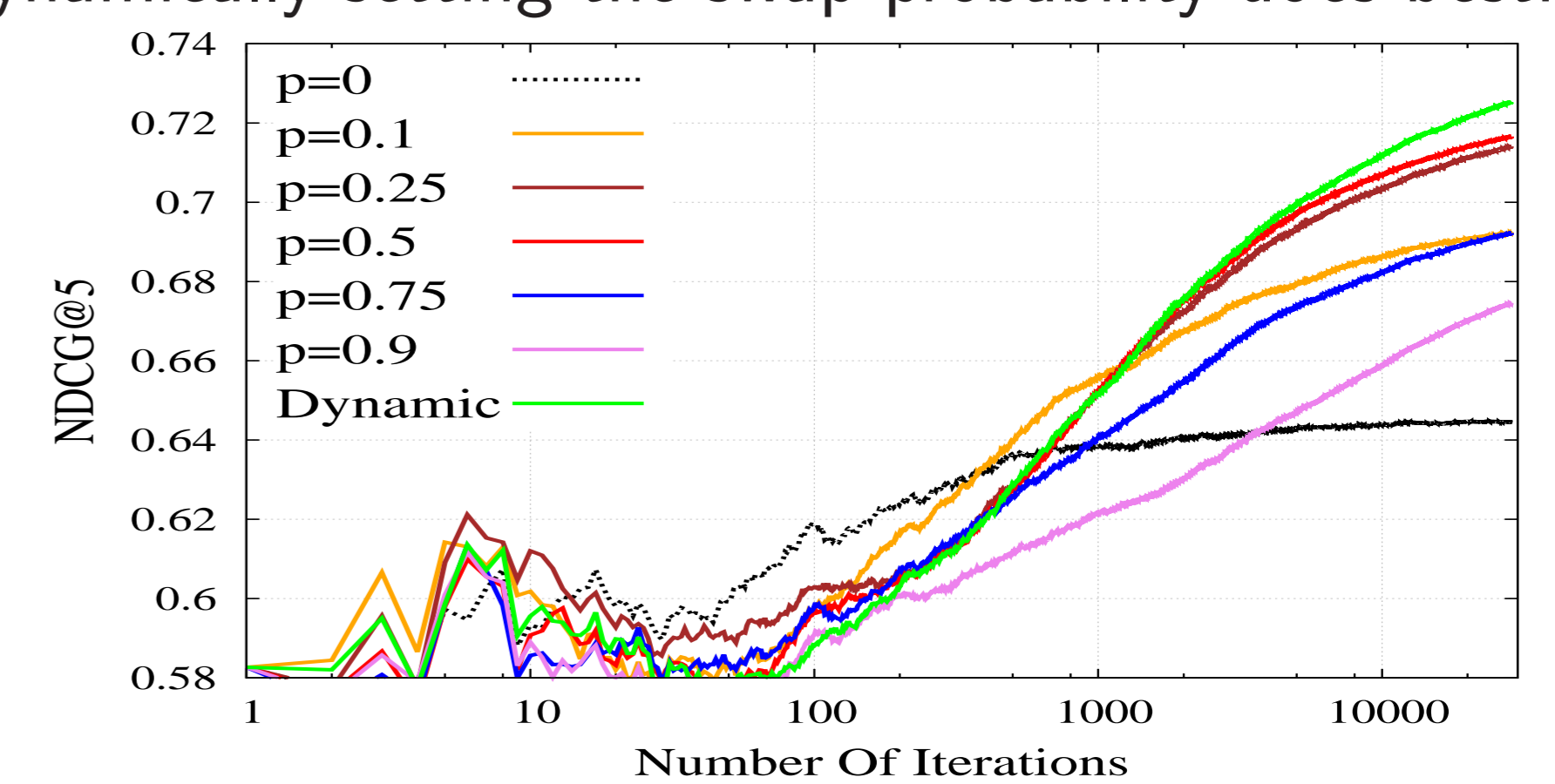
- PrefP[top]: Preference Perceptron with move-to-top feedback.
- PrefP[pair]: PrefP with pairwise feedback *i.e.*, 3PR with $p_t = 0$.
- SVM: Ranking SVM with move-to-top feedback.
- Perceptron which receives optimal is (rough) upper bound.



- Small loss in performance due to perturbation, but large gain overall (as seen from PrefP[pair] comparison).

	Websearch	RCV1	News
Presented y	.717 ± .002	.286 ± .028	.386 ± .035
Predicted \hat{y}	.723 ± .002	.291 ± .028	.397 ± .035

- Dynamically setting the swap probability does best:



- Far more robust to noise:

