Stable Coactive Learning via Perturbation

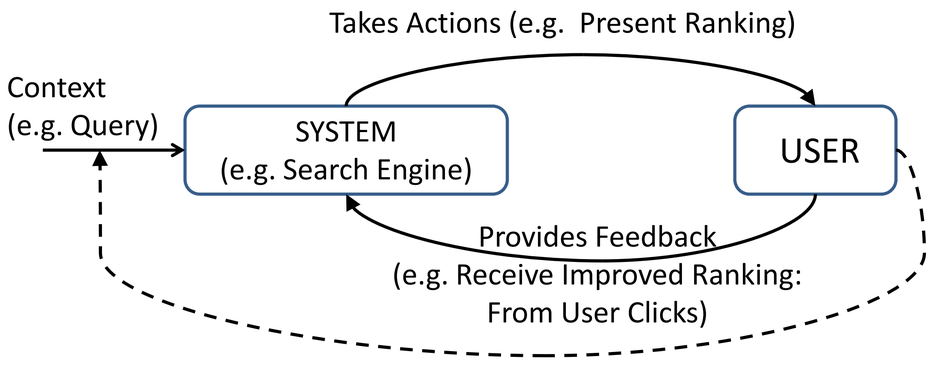
Karthik Raman, Thorsten Joachims (Cornell University)
Pannaga Shivaswamy (AT&T Research), Tobias Schnabel (University of Stuttgart)

Cornell University

Coactive Learning

Coactive Learning: [Shivaswamy and Joachims, 2012]

- Given context $\mathbf{x}$, predict object $\mathbf{y}$ to optimize utility $\mathbf{U(x, y)}$.
- Models the interaction between user(s) and learning system.



Example: Using implicit feedback for ranking:

Instability of the Preference Perceptron

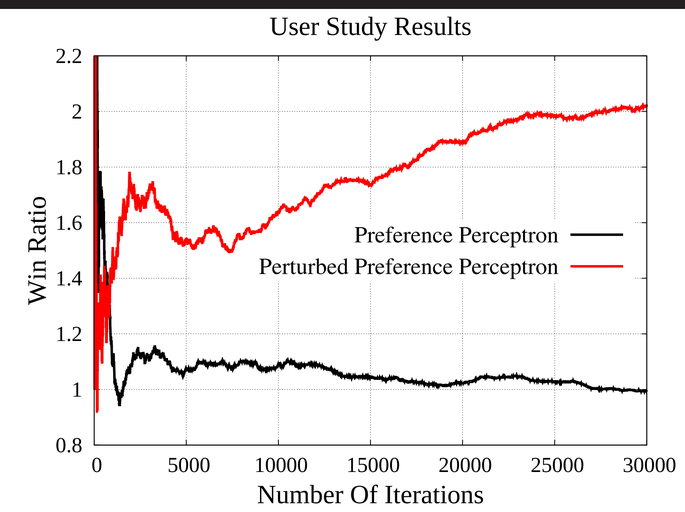
Linear Utility: $\mathbf{U(x, y) = w_*^\top \phi(x, y)}$

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

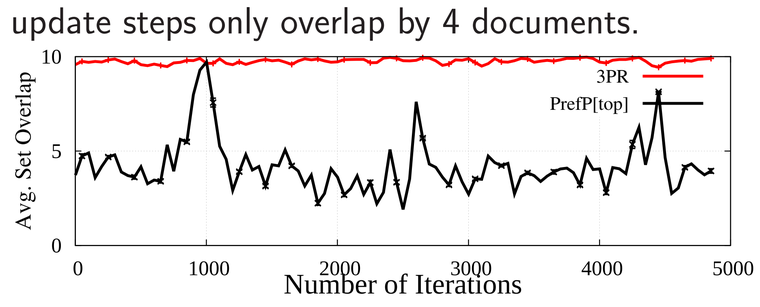
1. Initialize weight vector $\mathbf{w_1 \leftarrow 0}$.
2. Given context $\mathbf{x_t}$ present $\mathbf{y_t \leftarrow \text{argmax}_y w_t^\top \phi(x_t, y)}$.
3. Move clicked documents to top to get feedback ranking $\mathbf{\bar{y}_t}$.
4. $\mathbf{w_{t+1} \leftarrow w_t + \phi(x_t, \bar{y}_t) - \phi(x_t, y_t)}$.
5. 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.

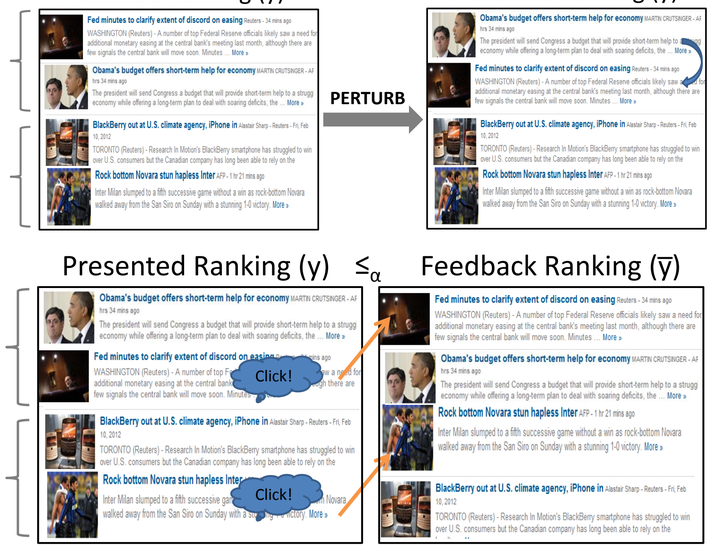


Preference Perceptron is unstable:
The rankings learned by PrefP never stabilize: Even after thousands of updates, the top 10 documents of the same query before and after 100 update steps only overlap by 4 documents.

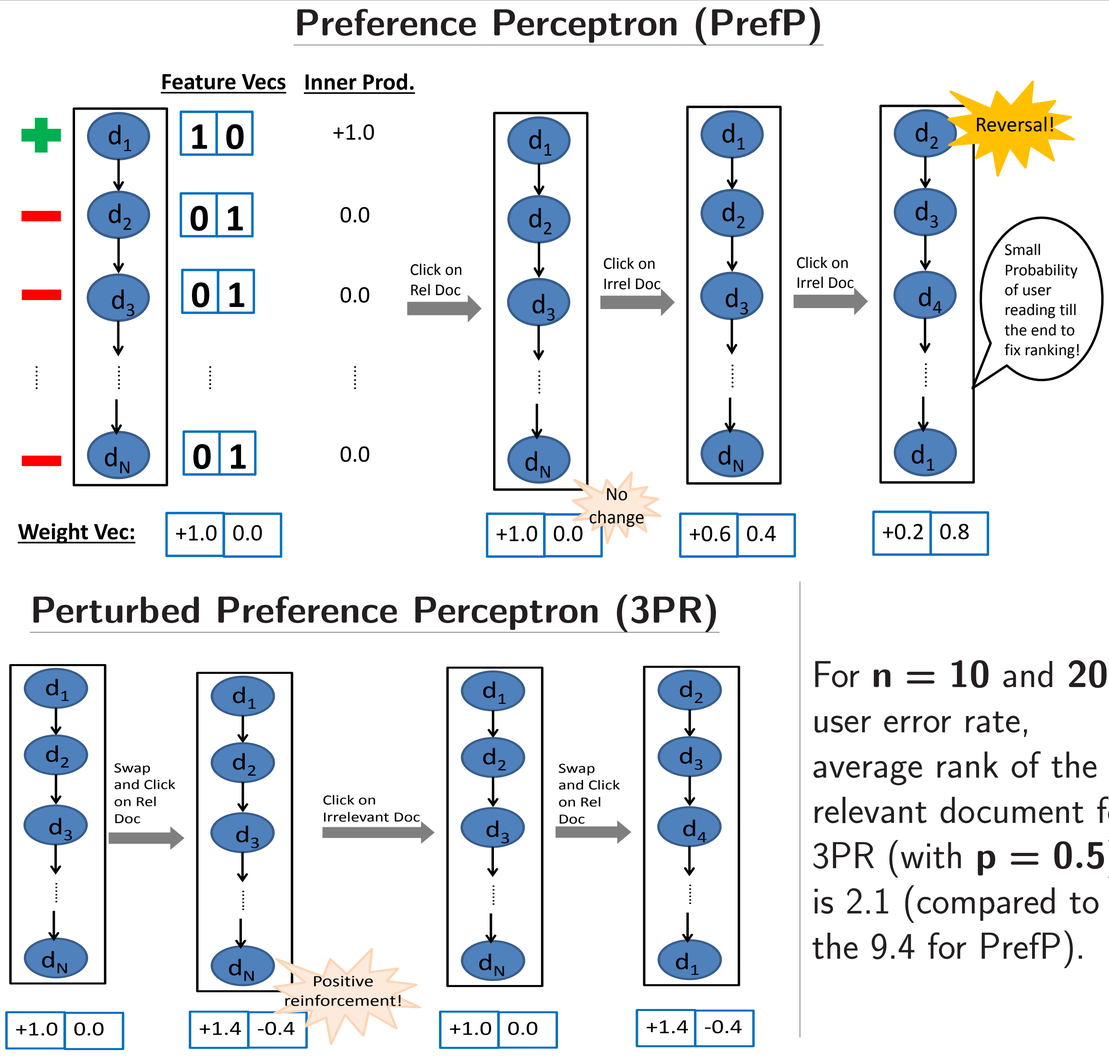


Perturbed Preference Perceptron

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



Illustrative Example



For $\mathbf{n = 10}$ and **20%** user error rate, average rank of the relevant document for 3PR (with $\mathbf{p = 0.5}$) is 2.1 (compared to the 9.4 for PrefP).

Theoretical Analysis

α-Informative Feedback:
We characterize the utility of the feedback received $\mathbf{\bar{y}_t}$ as:

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

- where $\mathbf{y^*}$ is the optimal and $\mathbf{y}$ 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 $\mathbf{T}$ iterations as:

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

Fixed Probability 3PR

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

$$\mathbf{\leq \frac{\sum_{t=1}^{T}\xi_t}{\alpha T} + \frac{p(1 - \frac{\gamma_2}{\gamma_1})R\|w_*\|}{\alpha} + \frac{\sqrt{2(4 - p^2(1 - \frac{\gamma_2}{\gamma_1})^2)}R\|w_*\|}{\alpha\sqrt{T}}.}$$

Dynamic 3PR

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

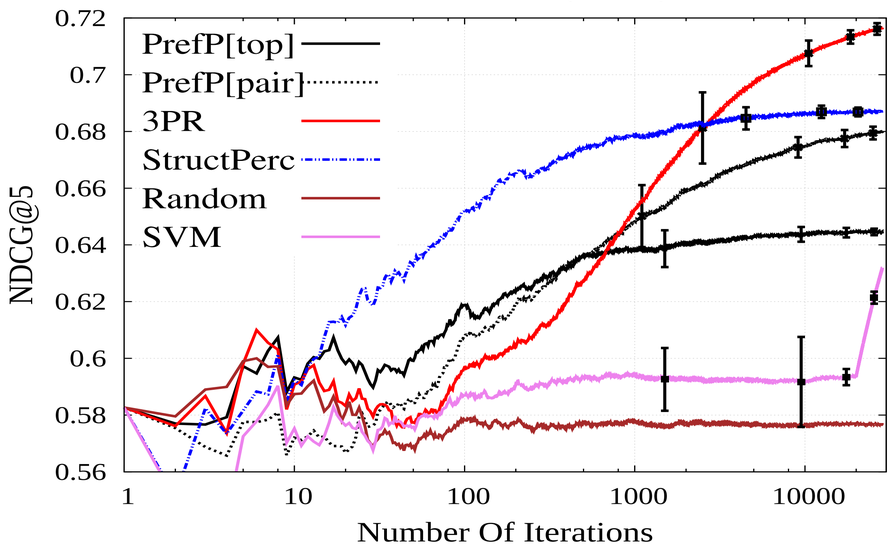
$$\mathbf{\leq \frac{1}{\alpha T}\sum_{t=1}^{T}\xi_t + \frac{\|w_*\|}{\alpha\sqrt{T}} \times \sqrt{4R^2 + 2\Delta + 2R\sqrt{\frac{4R^2 + 2\Delta}{T}}}.}$$

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:

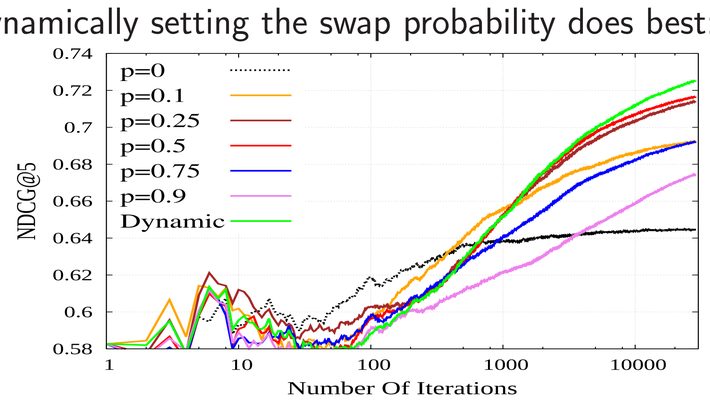
- PrefP[top]: Preference Perceptron with move-to-top feedback.
- PrefP[pair]: PrefP with pairwise feedback *i.e.*, 3PR with $\mathbf{p_t = 0}$.
- SVM: Ranking SVM with move-to-top feedback.
- Perceptron which receives optimal is (rough) upper bound.



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

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

- Dynamically setting the swap probability does best:



- Far more robust to noise: