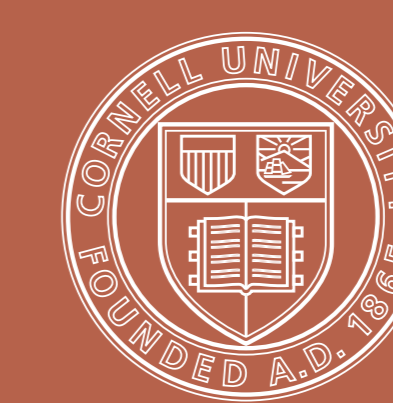


Learning Socially Optimal Information Systems from Egoistic Users

Karthik Raman, Thorsten Joachims (`{karthik,tj}@cs.cornell.edu`)

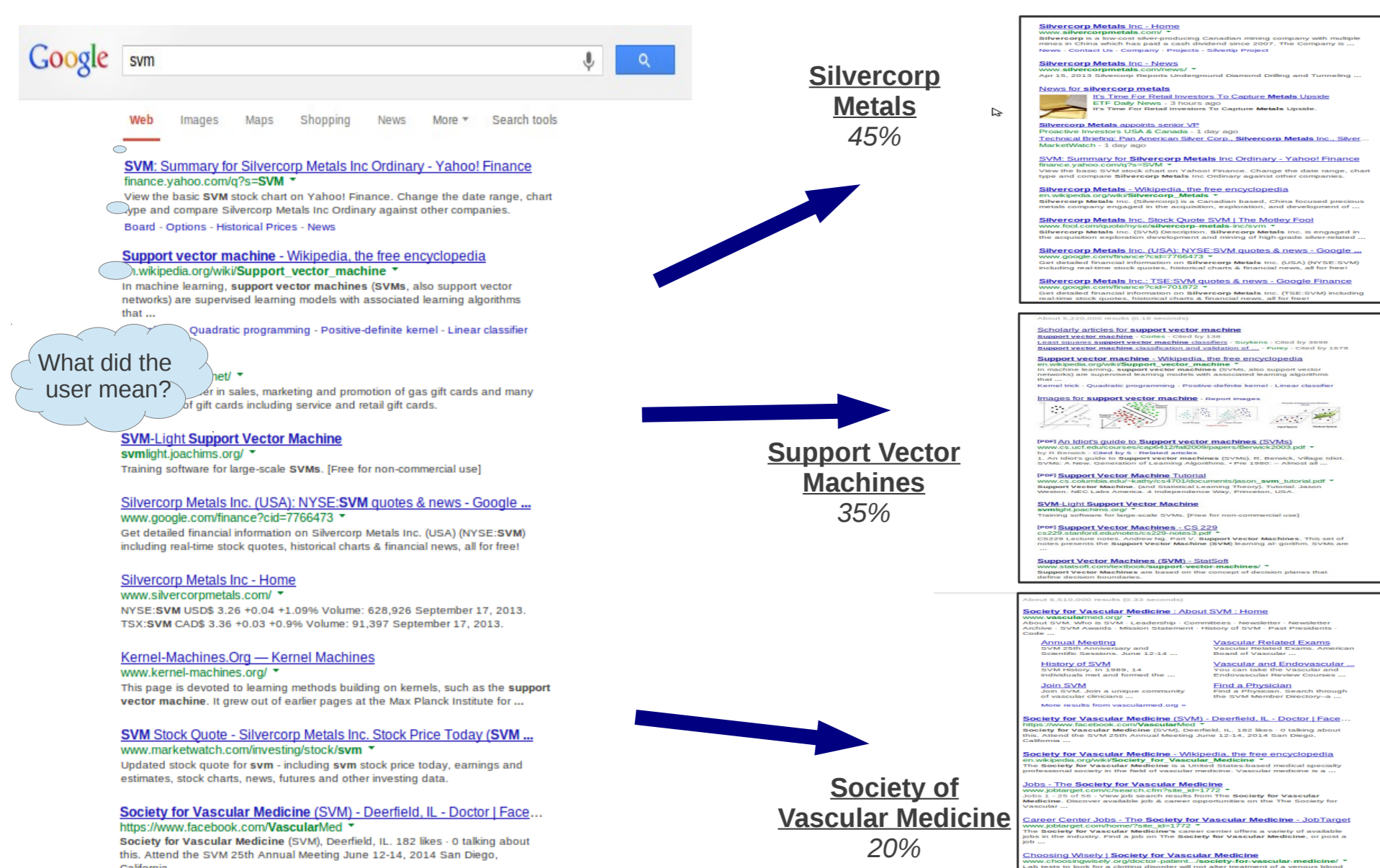


Cornell University

Problem Overview

- Learn robust systems that collectively satisfy a population of diverse users from user feedback.

Example: Diversity in Search



Goal: Find best overall (socially optimal) ranking.

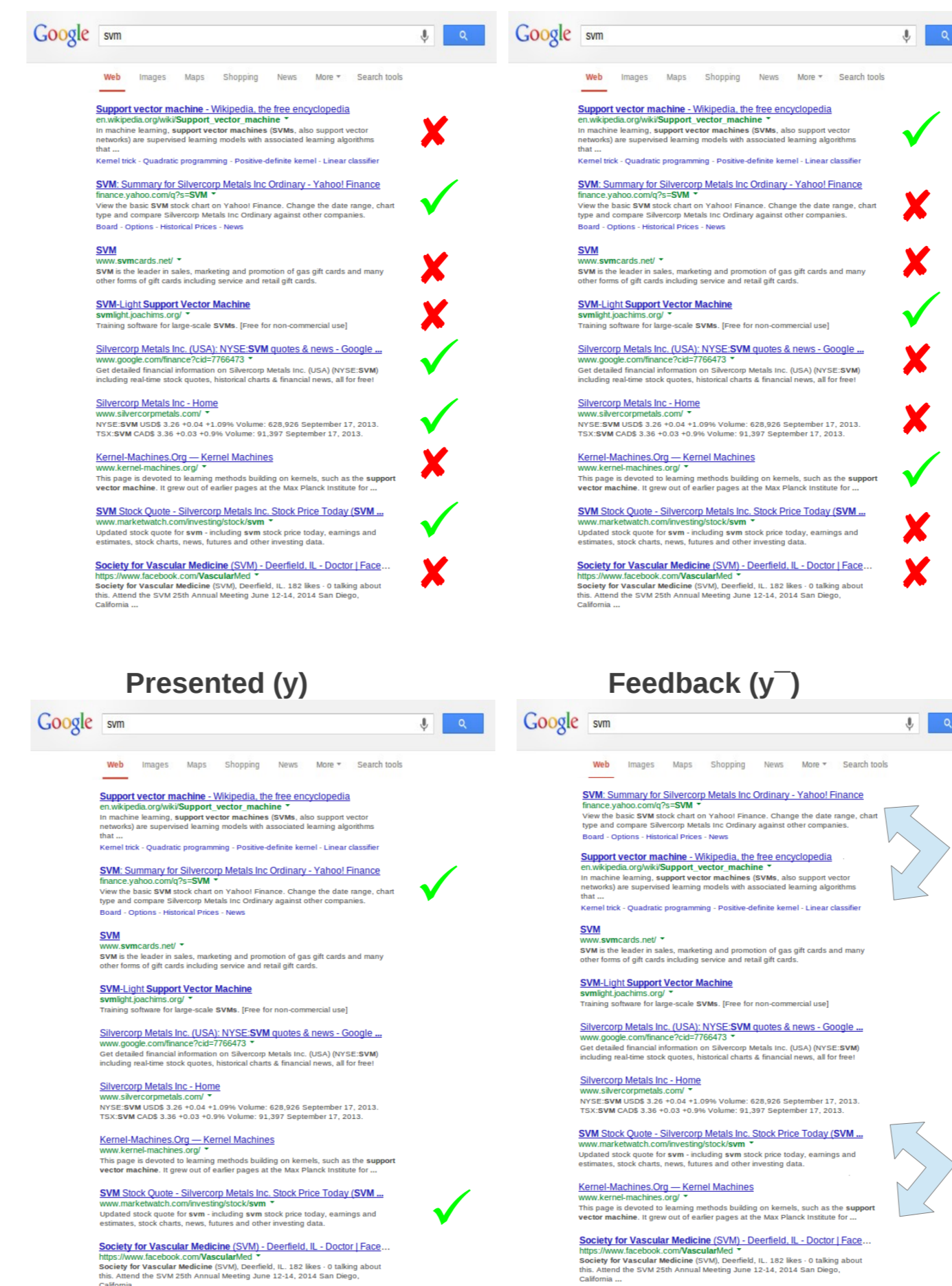
User Feedback

Egoistic user feedback

- User's choice not social.
- Conflicting choices.

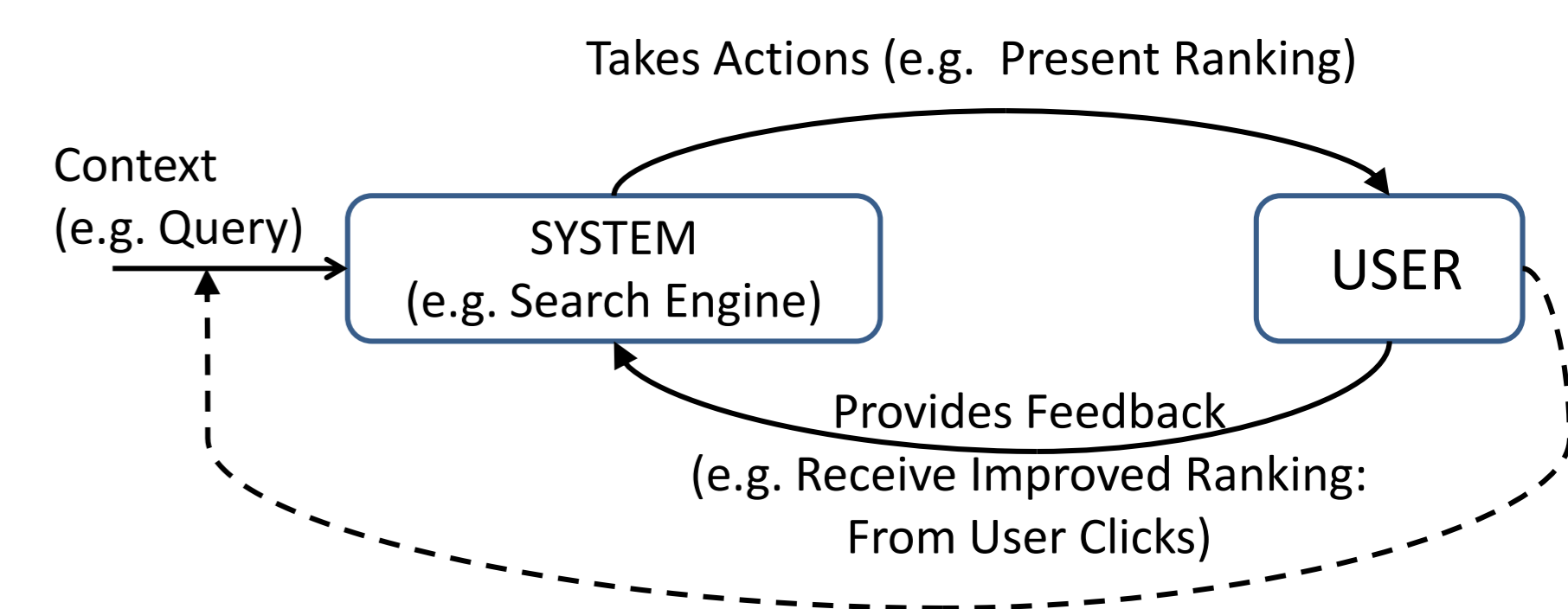
Weak, noisy & biased feedback.

- Cannot regard as cardinal labels.
- Treat as preferences.



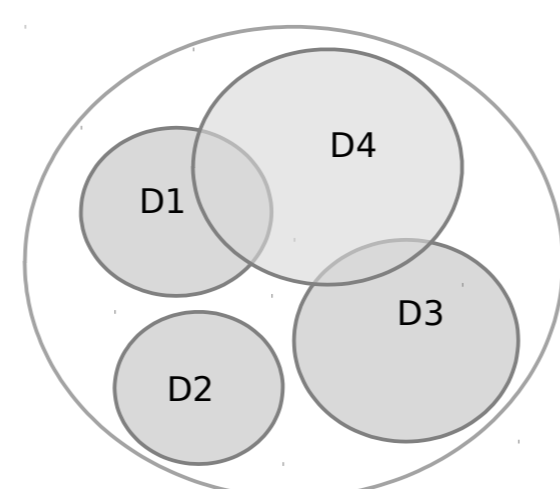
Learning from User Preferences

- Builds on **Coactive Learning** [SJ12,RJSS13].
- Given context x , predict object y .
- Goal: Optimize social utility $U(x, y) = E[U_i(x, y)]$.
 - U_i is personal utility of user type i (w.p. p_i).
- User preferences: Feedback tends to improve personal utility: $U_i(x, \bar{y}) \geq_{\alpha, \delta} U_i(x, y)$.
- Not social utility.



Modeling Utility: Submodularity

- Model personal utility of users as submodular in individual elements.
- Diminishing returns:** Marginal benefit diminishes.
- Example: Coverage Function
- Given ranking/set $y = (d_1, \dots, d_n)$ and position-discount factors $\gamma_1 \geq \gamma_2 \geq \dots \geq \gamma_n \geq 0$, aggregate features using submodular function F :



$$\phi_F^j(x, y) = F(\gamma_1 \phi^j(x, d_1), \gamma_2 \phi^j(x, d_2), \dots, \gamma_n \phi^j(x, d_n))$$

- $\phi^j(x, d_i)$ is j^{th} feature of d_i .
- Model personal utility as linear in submodular aggregate: $U_i(x, y) = w_{*,i}^T \phi_F(x, y)$
- Submodular aggregation leads to diversity.
- Computing ranking \approx Submodular maximization
- Use simple, efficient greedy algorithm.
- Approximation guarantee of $\frac{1}{2}$ (under partition matroid constraint).
- Example of Diversity:

Doc	Words	Word	Weight	Posn	Doc	ma	le	me	si	Doc	Marginal Benefit
d_1	ma:3 le:3	machine	5	1	d_3	2	5	0	0	d_1	$3*5 + 3*7$
d_2	ma:5 le:2	learning	7	2	d_4	0	0	3	5	d_2	$5*5 + 2*7$
d_3	ma:2 le:5	metal	4	3	d_2	5	2	0	0	d_3	$2*5 + 5*7$
d_4	me:3 si:5	silver	6	MAX of Col	5	5	3	5		d_4	$3*4 + 5*6$
d_5	me:6 si:2									d_5	$6*4 + 2*6$
d_6	me:3 si:1									d_6	$3*4 + 1*6$

Social Perceptron for Ranking

- Initialize weight vector $w_1 \leftarrow 0$.
- Given context x_t present user with $y_t \leftarrow \text{argmax}_y w_t^T \phi(x_t, y)$.
- Observe user clicks \mathcal{D} .
- Construct preference feedback: $\bar{y}_t \leftarrow \text{PairedFeedback}(y_t, \mathcal{D})$.
- Update weight vector: $\bar{w}_{t+1} \leftarrow w_t + \phi(x_t, \bar{y}_t) - \phi(x_t, y_t)$
- Clip to be non-negative: $w_{t+1}^j \leftarrow \max(\bar{w}_{t+1}^j, 0)$
- Repeat from step 2.



PairedFeedback: Form pairs and swap if only lower element is clicked.

Referred to as the **SoPer-R** algorithm.

Also provide an algorithm for learning diverse sets called the **SoPer-S** algorithm.

- See paper for more details

Theoretical Analysis

α_i, δ_i -Informative Feedback:

Characterize feedback \bar{y} in terms of α_i, δ_i, ξ as:

$$E_{\bar{y}}[U_i(x, \bar{y})] \geq (1 + \delta_i)U_i(x, y) + \alpha_i (U_i(x, y^{*,i}) - U_i(x, y)) - \xi$$

- where $y^{*,i}$ is optimal for user i
- and y is the presented object.
- Note that this is a characterization (not an assumption).
- Does not assume anything about social utility.
- Used to prove regret bounds.

Regret: Define the regret after T iterations as:

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

- Note: In terms of social utility and social optimal.

Regret Bound

If $\delta_i \geq \left(\Gamma_F \cdot \frac{1-p_i}{p_i} \right)$, average regret of the SoPer-R is:

$$\leq \frac{1}{\eta T} \sum_{t=0}^{T-1} E_i[p_i \xi_t] + \frac{R \|w_{*}\|}{2\eta} + \frac{\sqrt{15R} \|w_{*}\|}{\eta \sqrt{2T}}$$

with $\eta = \min_i p_i \alpha_i$.

Understanding the bound:

- Does not depend on number of dimensions only radius of ball R .
- Decays gracefully with weak feedback: $\alpha_i \xi$.
- Need not converge to optimal (due to NP-hardness of submodular maximization).
- Bound is loose as solution improves.

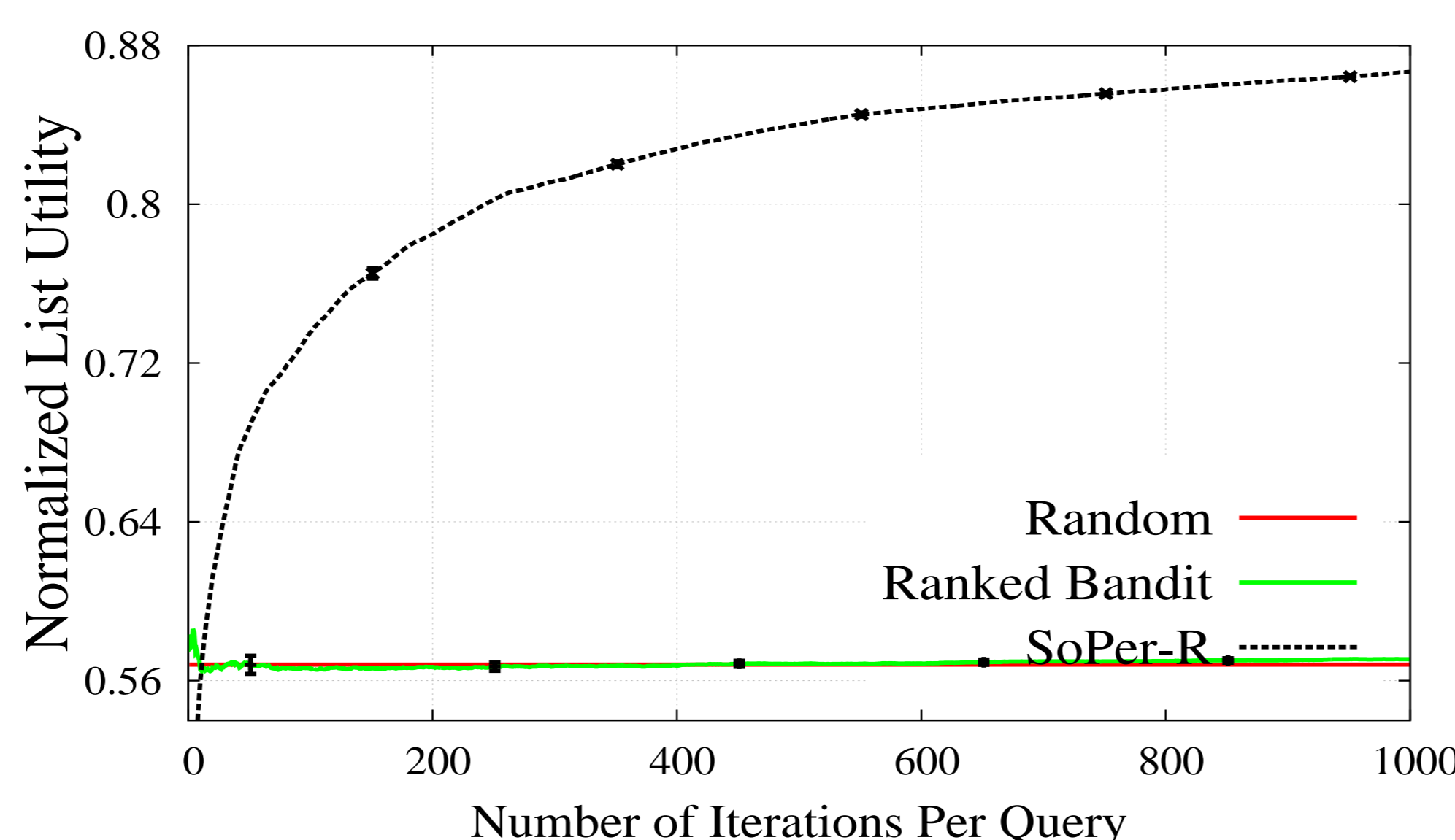
Similar bound for **SoPer-S** algorithm as well.

Experimental Results

- Offline experiments on standard **TREC 6-8 Interactive** search diversification dataset.
 - Queries have 7-56 user types with binary relevance labels.
- Simulated user behavior: Scan rankings top to bottom. Click on first document relevant to them (with some error).
- Utility: Normalized DCG-Coverage function upto rank 5.

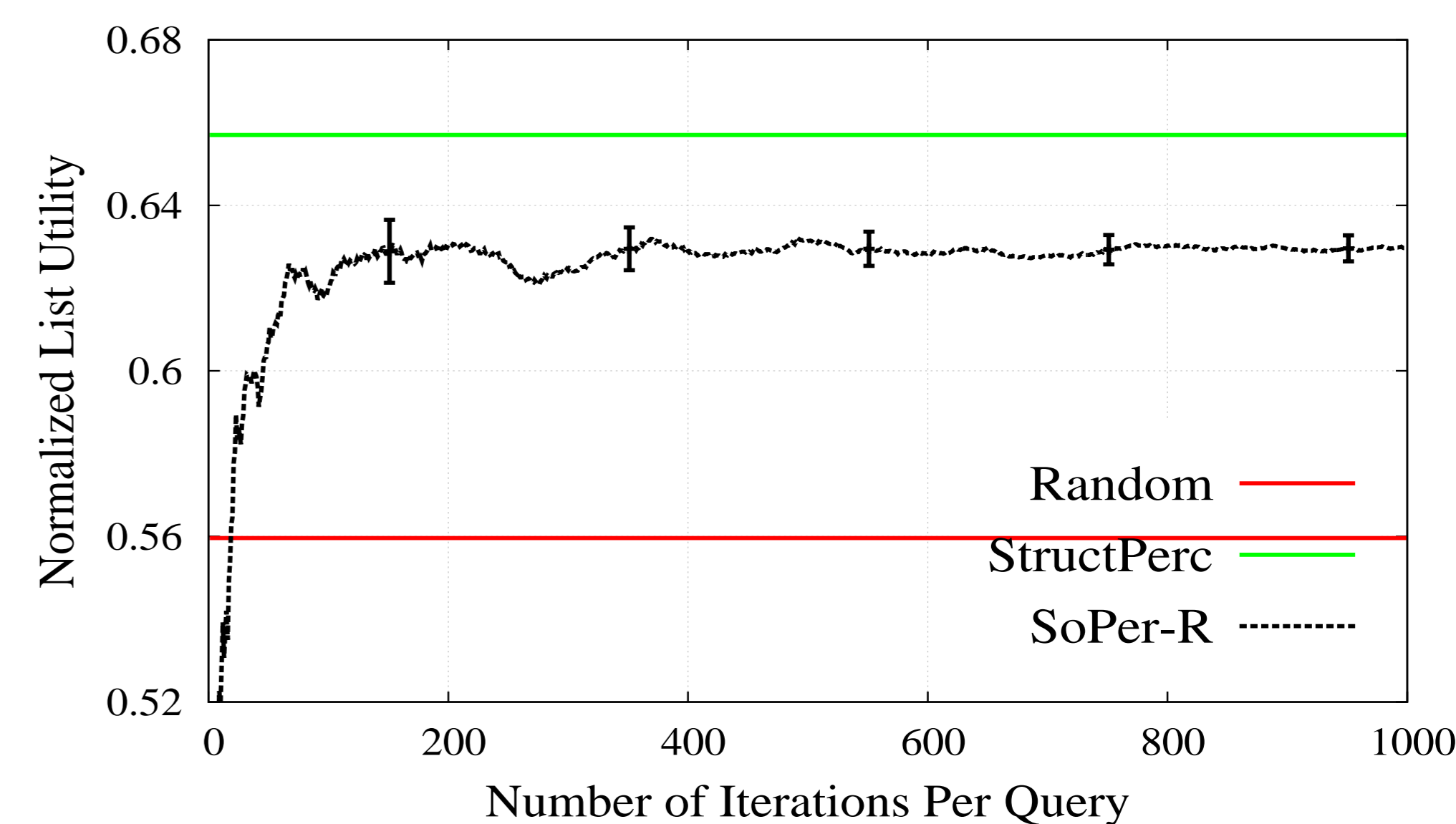
Single Query Diversification:

- Learning to diversify for single query.
- Compare with RankedBandit (Array of coupled MABs).



Cross-Query Diversification:

- Learning to diversify given any query.
- Structured Perceptron receives social-optimal as feedback.



- First method to diversify across queries from preferences.
- Robust to model mis-specification.

	TrueSocialIF	SoPer-R (Varying Submodular Function)			Rand
		MAX	SQRT	LIN	
MAX	.630 ± .007	.620 ± .006	.618 ± .006	.557 ± .006	
SQRT	.656 ± .007	.654 ± .007	.684 ± .006	.610 ± .007	
LIN	.500 ± .006	.504 ± .006	.566 ± .007	.474 ± .007	

- Robust to feedback noise (.631 vs .630).