Learning Socially Optimal Information Systems from Egoistic Users

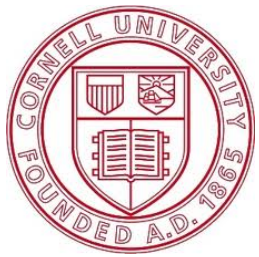
Karthik Raman Thorsten Joachims

Department of Computer Science
Cornell University, Ithaca NY

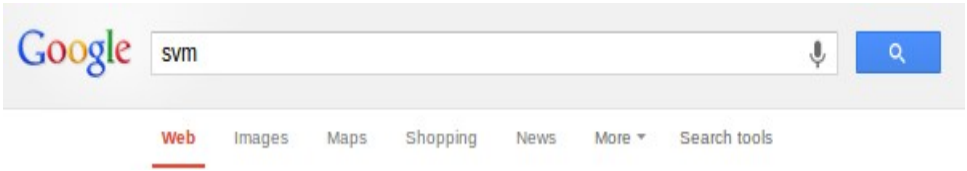
karthik@cs.cornell.edu
www.cs.cornell.edu/∼karthik

tj@cs.cornell.edu
www.cs.cornell.edu/People/tj

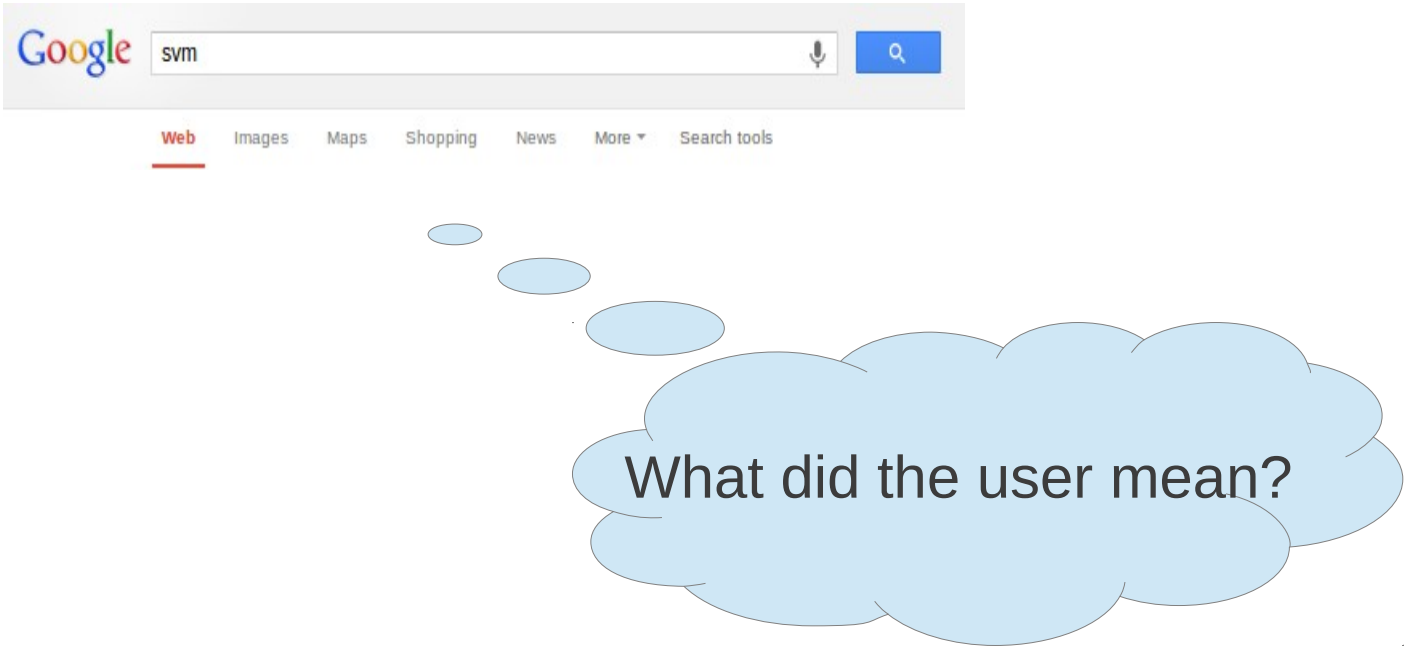
Sept 23, 2013



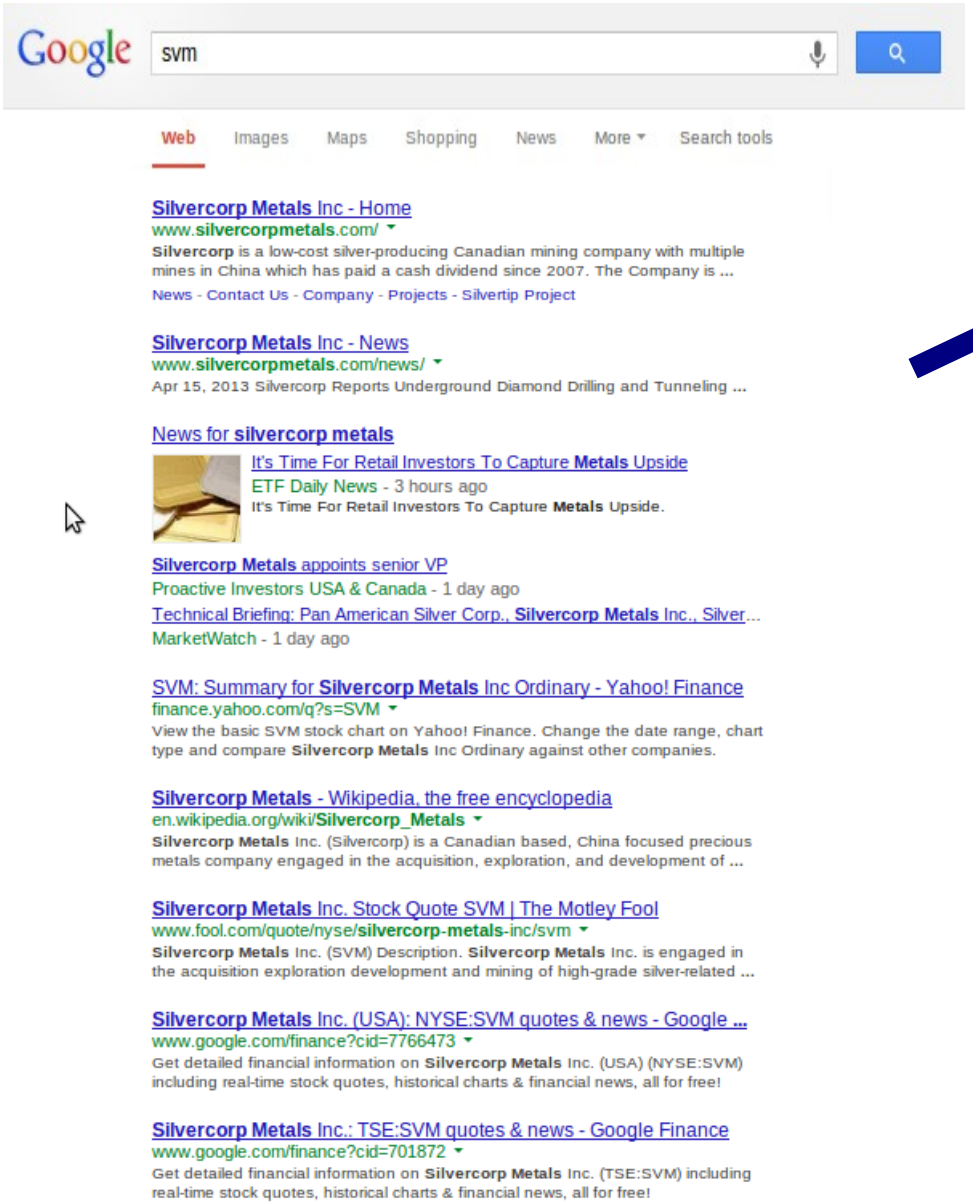
Example: (Extrinsic) Diversity in Web Search



Example: (Extrinsic) Diversity in Web Search

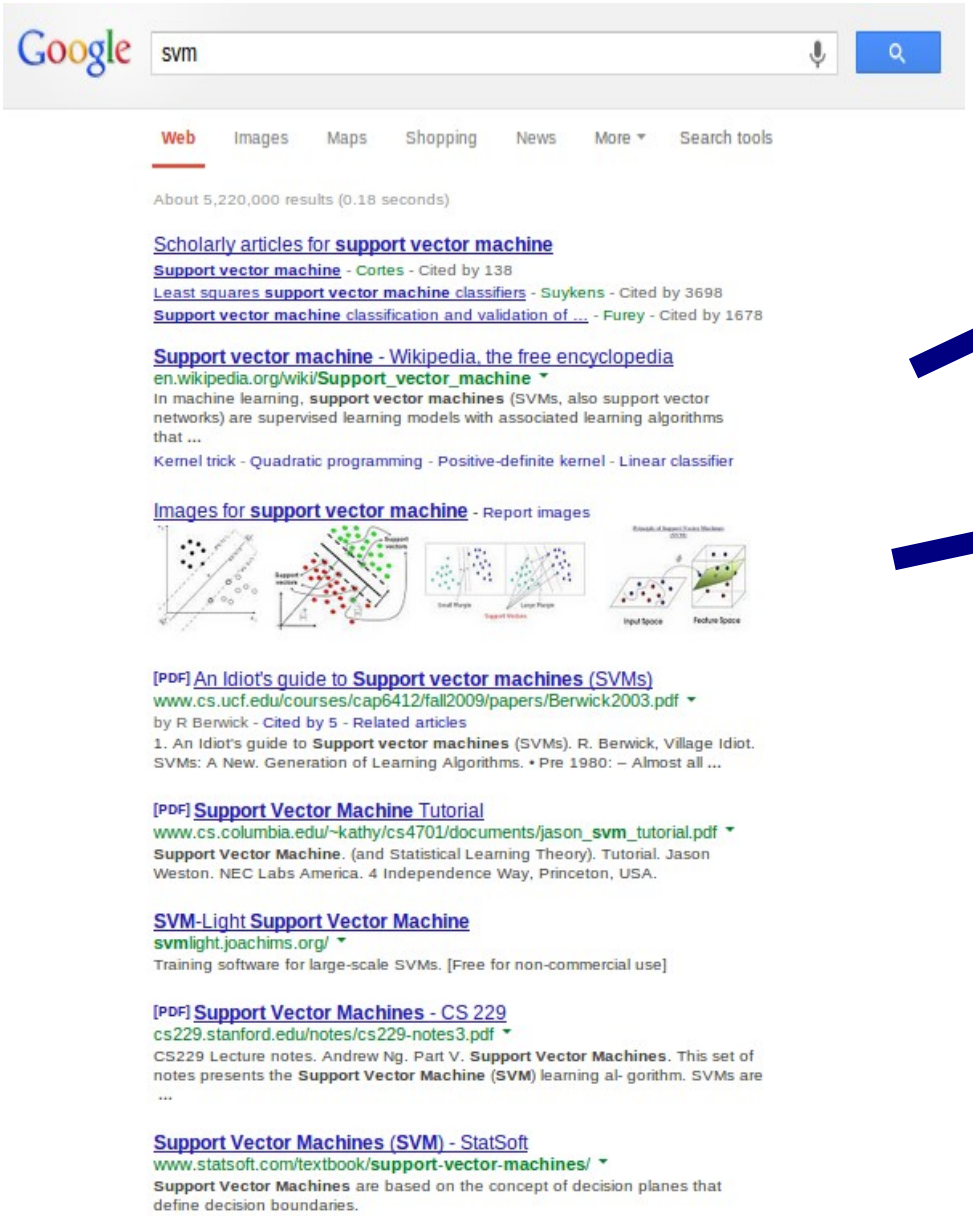


Example: (Extrinsic) Diversity in Web Search



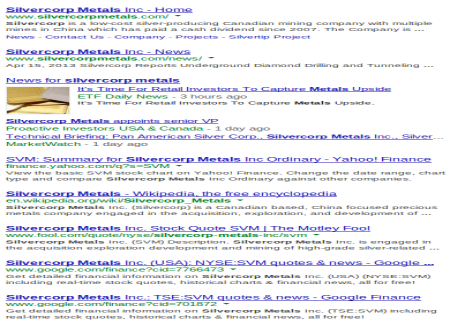
Silvercorp Metals
40%

Example: (Extrinsic) Diversity in Web Search

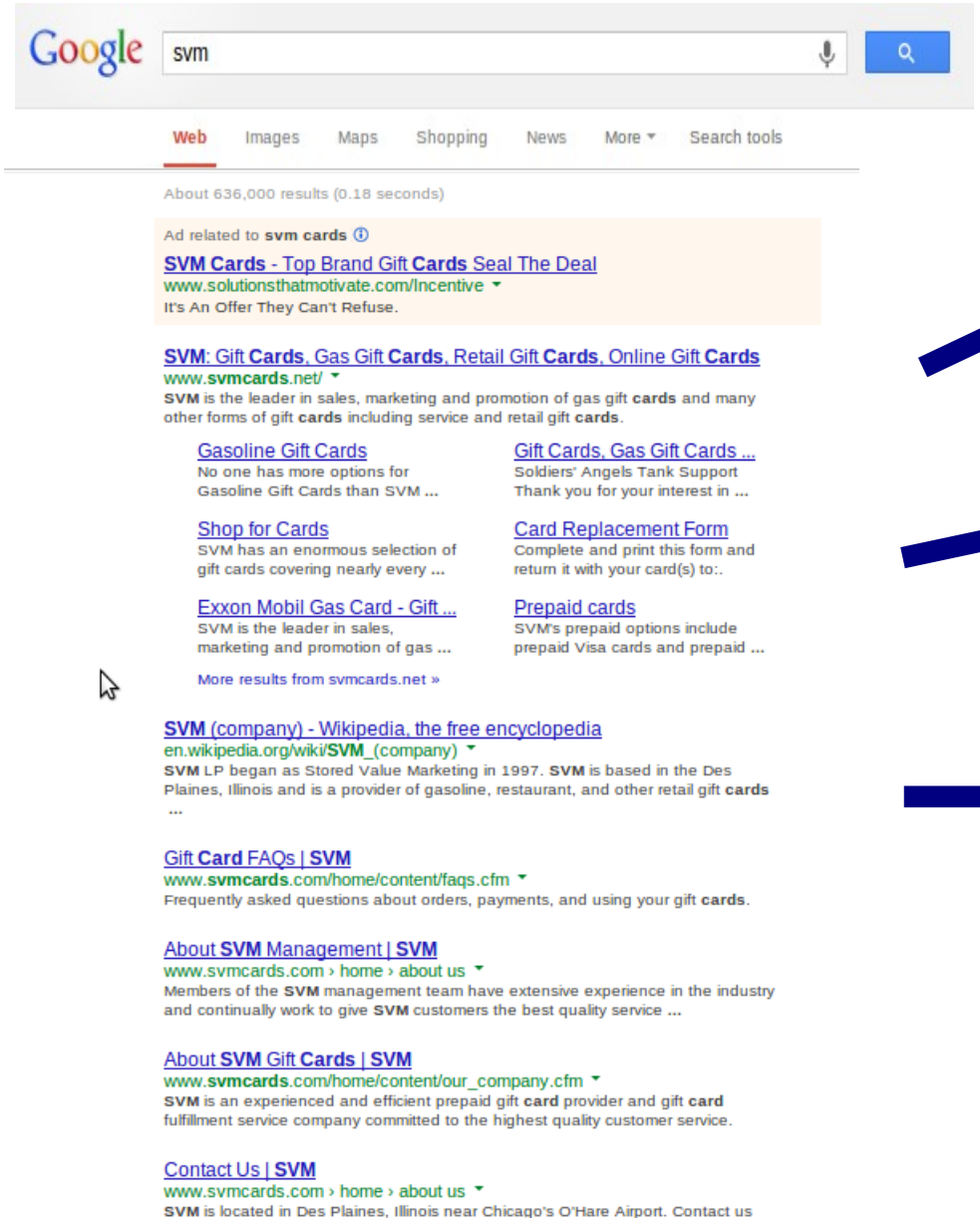


Silvercorp Metals
40%

Support Vector Machines
35%



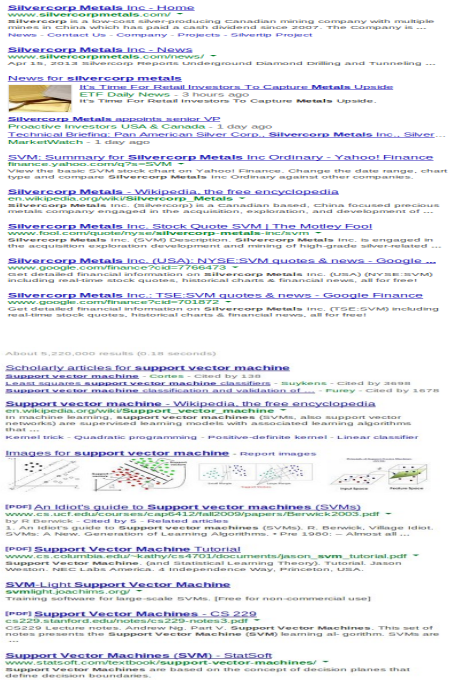
Example: (Extrinsic) Diversity in Web Search



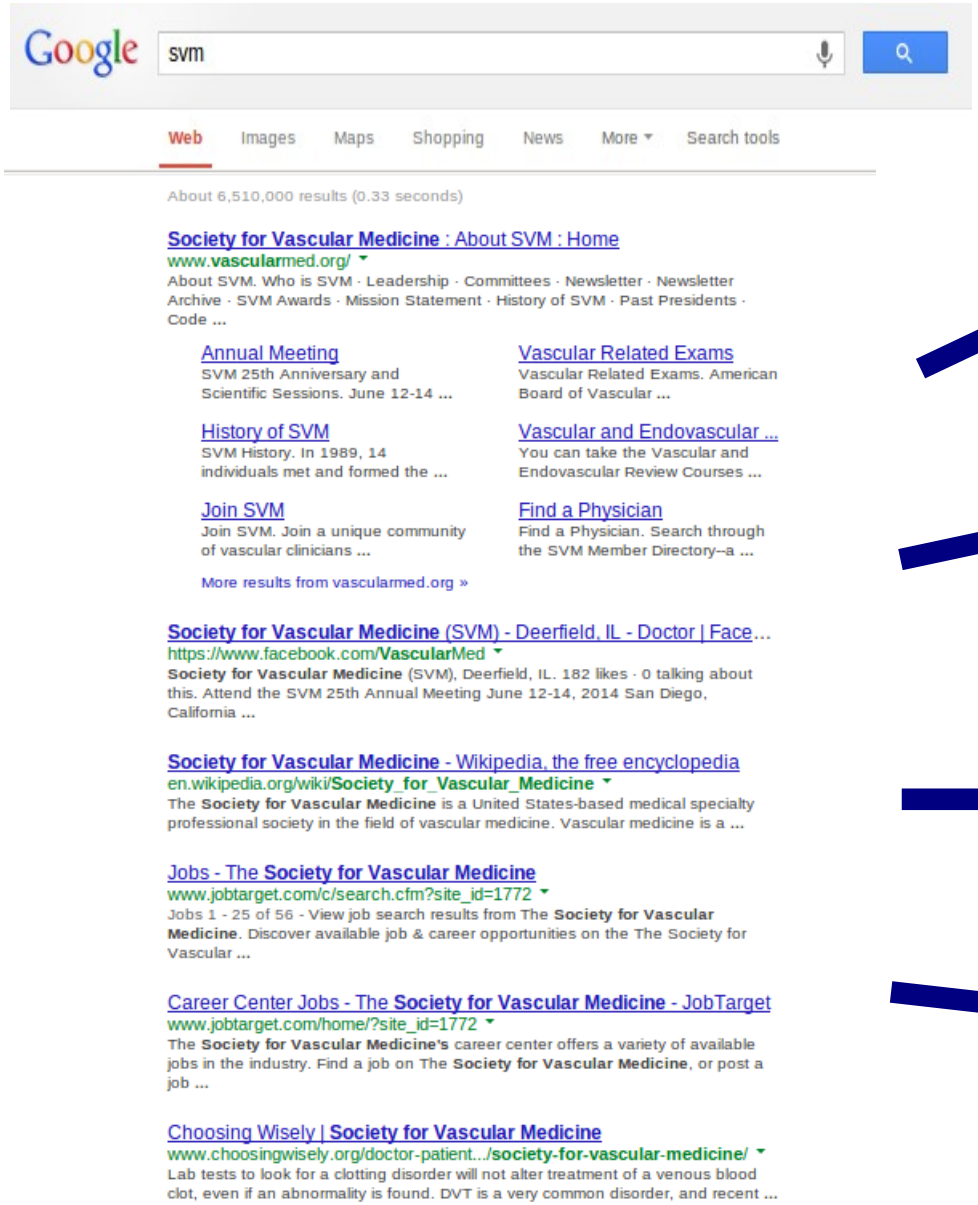
Silvercorp Metals
40%

Support Vector Machines
35%

SVM Gift Cards
15%



Example: (Extrinsic) Diversity in Web Search



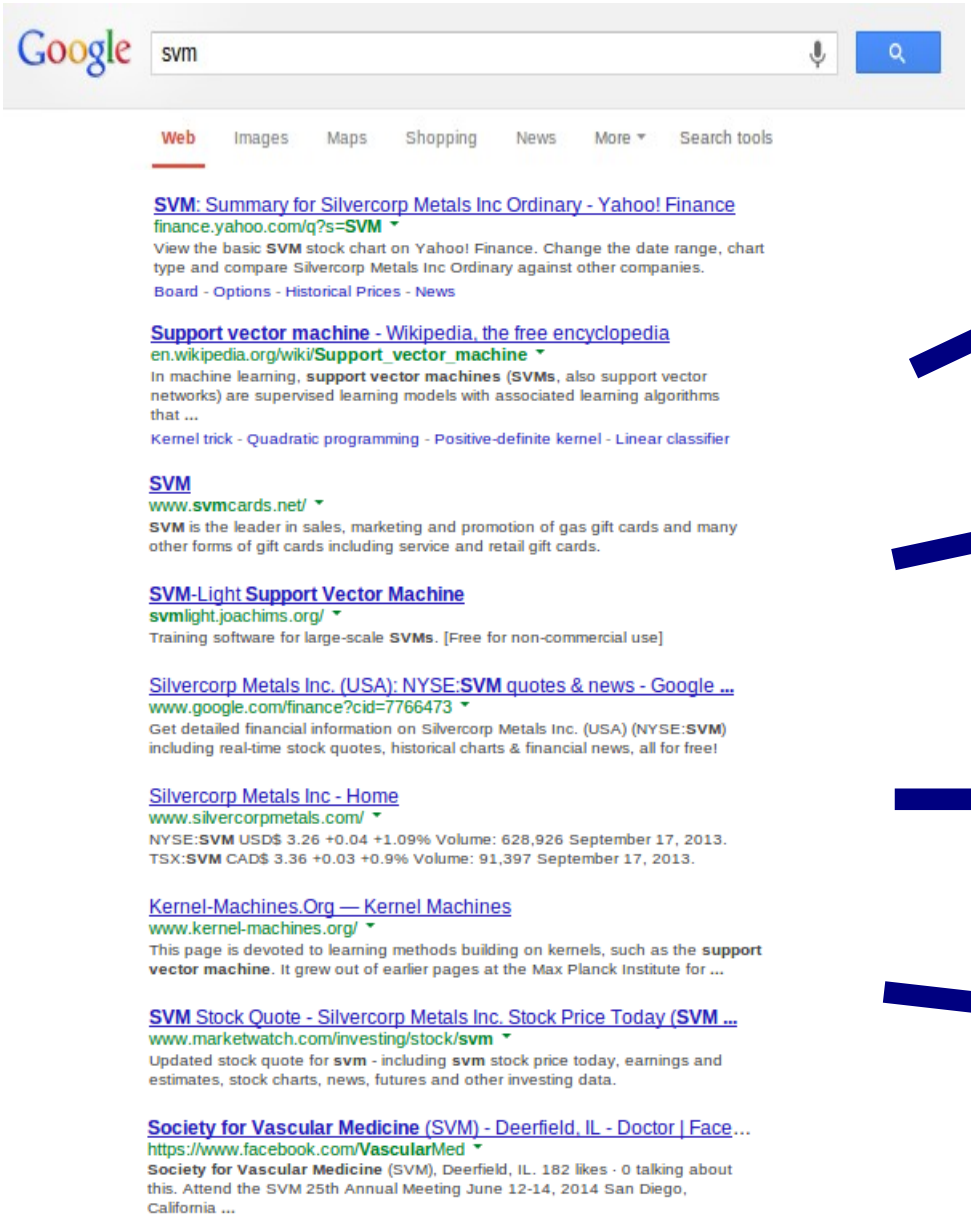
Silvercorp Metals
40%

Support Vector Machines
35%

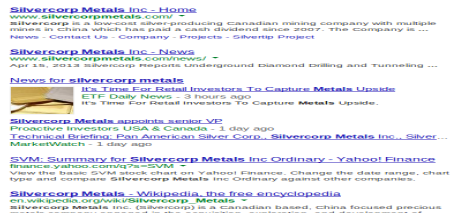
SVM Gift Cards
15%

Society of Vascular Medicine
10%

Example: (Extrinsic) Diversity in Web Search

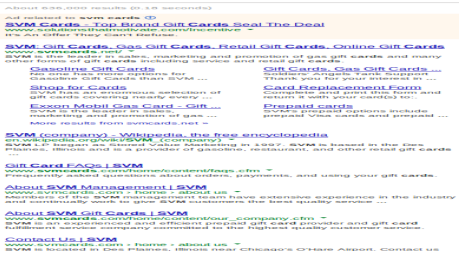


Silvercorp Metals
40%

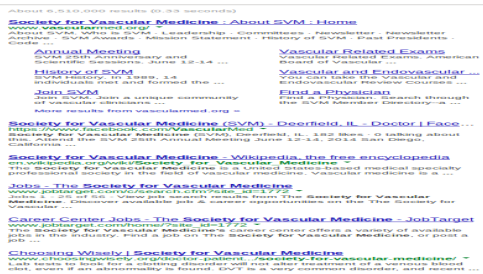


Support Vector Machines
35%

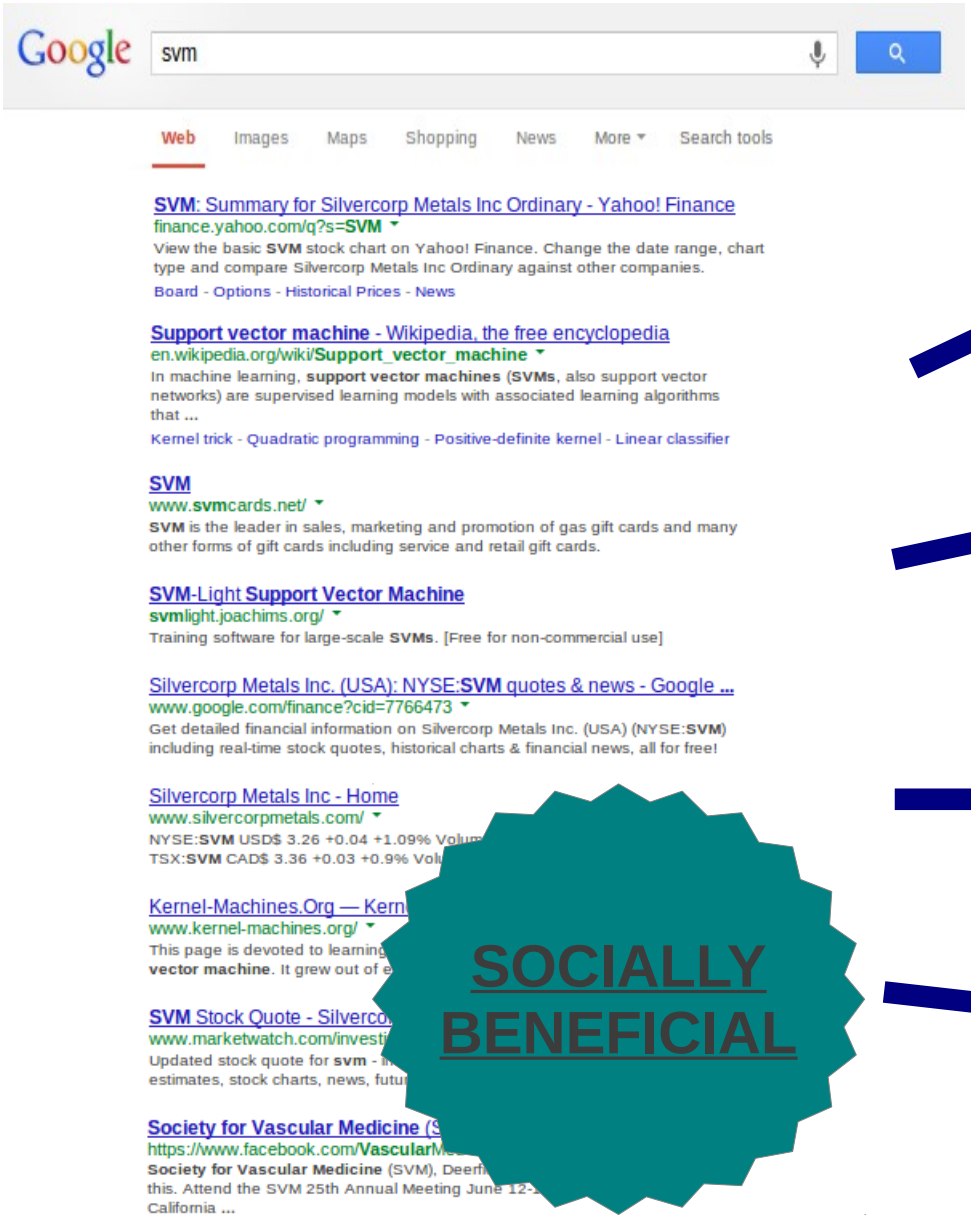
SVM Gift Cards
15%



Society of Vascular Medicine
10%

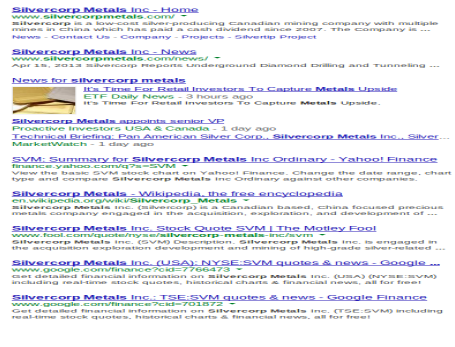


Example: (Extrinsic) Diversity in Web Search

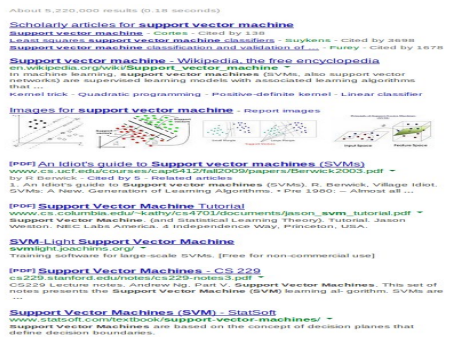


SOCIALLY BENEFICIAL

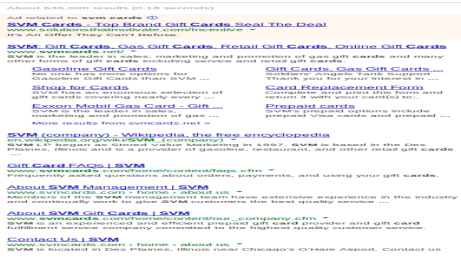
Silvercorp Metals *40%*



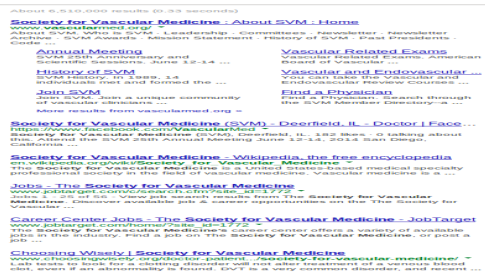
Support Vector Machines *35%*



SVM Gift Cards *15%*



Society of Vascular Medicine *10%*



Socially optimal solutions in Information systems

- **Problem:** Common content for users with different tastes.
  - Hedge against uncertainty in user's preferences.

Socially optimal solutions in Information systems

- **Problem:** Common content for users with different tastes.
  - Hedge against uncertainty in user's preferences.
  - **Ex:** News Sites, Product & Media Recommendation, Frontpages.

Socially optimal solutions in Information systems

- **Problem:** Common content for users with different tastes.
  - Hedge against uncertainty in user's preferences.
  - **Ex:** News Sites, Product & Media Recommendation, Frontpages.
- Diverse User population: N user types.
- User type i has probability p_i.

Socially optimal solutions in Information systems

- **Problem:** Common content for users with different tastes.
  - Hedge against uncertainty in user's preferences.
  - **Ex:** News Sites, Product & Media Recommendation, Frontpages.
- Diverse User population: N user types.
- User type i has probability p_i.
- **Personal** utility for object (e.g. ranking) $\mathbf{y}$: $U_i(\cdot, \mathbf{y})$.

Socially optimal solutions in Information systems

- **Problem:** Common content for users with different tastes.
  - Hedge against uncertainty in user's preferences.
  - **Ex:** News Sites, Product & Media Recommendation, Frontpages.
- Diverse User population: N user types.
- User type i has probability p_i.
- **Personal** utility for object (e.g. ranking) $\mathbf{y}$: $U_i(\cdot, \mathbf{y})$.
- **Social** utility is the expected utility (over all users):

$$U(\cdot, \mathbf{y}) = \mathbf{E}[U_i(\cdot, \mathbf{y})] = \sum_{i=1}^{N} p_i U_i(\cdot, \mathbf{y})$$

Socially optimal solutions in Information systems

- **Problem:** Common content for users with different tastes.
  - Hedge against uncertainty in user's preferences.
  - **Ex:** News Sites, Product & Media Recommendation, Frontpages.
- Diverse User population: N user types.
- User type i has probability p_i.
- **Personal** utility for object (e.g. ranking) $\mathbf{y}$: $U_i(\cdot, \mathbf{y})$.
- **Social** utility is the expected utility (over all users):

$$U(\cdot, \mathbf{y}) = \mathbf{E}[U_i(\cdot, \mathbf{y})] = \sum_{i=1}^{N} p_i U_i(\cdot, \mathbf{y})$$

- **Goal:** Find Socially Optimal Solution $\mathbf{y}^* = \text{argmax}_{\mathbf{y}} U(\cdot, \mathbf{y})$.

Socially optimal solutions in Information systems

- **Problem:** Common content for users with different tastes.
  - ▷ Hedge against uncertainty in user's preferences.
  - ▷ **Ex:** News Sites, Product & Media Recommendation, Frontpages.
- Diverse User population: N user types.
- User type i has probability p_i.
- **Personal** utility for object (e.g. ranking) $\mathbf{y}$: $U_i(\cdot, \mathbf{y})$.
- **Social** utility is the expected utility (over all users):

$$U(\cdot, \mathbf{y}) = \mathbf{E}[U_i(\cdot, \mathbf{y})] = \sum_{i=1}^{N} p_i U_i(\cdot, \mathbf{y})$$

- **Goal:** Find Socially Optimal Solution $\mathbf{y}^* = \text{argmax}_{\mathbf{y}} U(\cdot, \mathbf{y})$.
- **Challenge:** Learn from egoistic, weak, noisy user feedback.

Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.

Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.
- User i's feedback reflects them behaving as per personal utility U_i.

Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.
- User i's feedback reflects them behaving as per personal utility U_i.
- **Not** social utility U.
  - ▷ As in the case of prior work [RSJ12] for the **intrinsic** diversity problem.

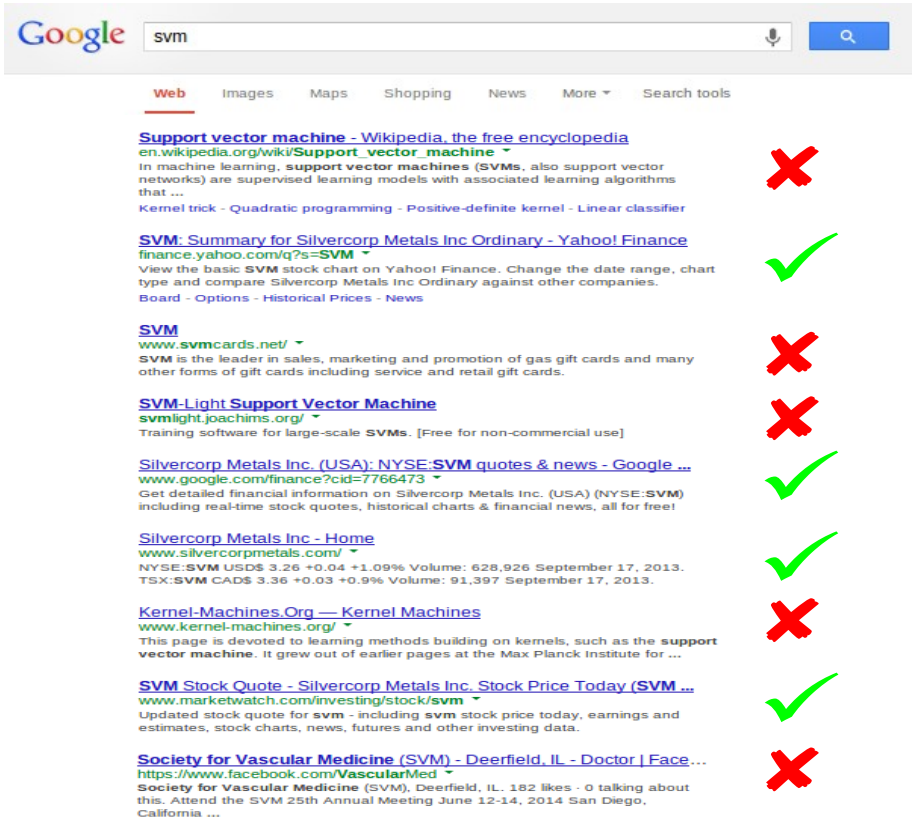
Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.
- User i's feedback reflects them behaving as per personal utility U_i.
- **Not** social utility U.
  - ▷ As in the case of prior work [RSJ12] for the **intrinsic** diversity problem.
- Need to infer social utility from such **conflicting, individual** feedback.

Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.
- User i's feedback reflects them behaving as per personal utility U_i.
- **Not** social utility U.
  - ▷ As in the case of prior work [RSJ12] for the **intrinsic** diversity problem.
- Need to infer social utility from such **conflicting, individual** feedback.

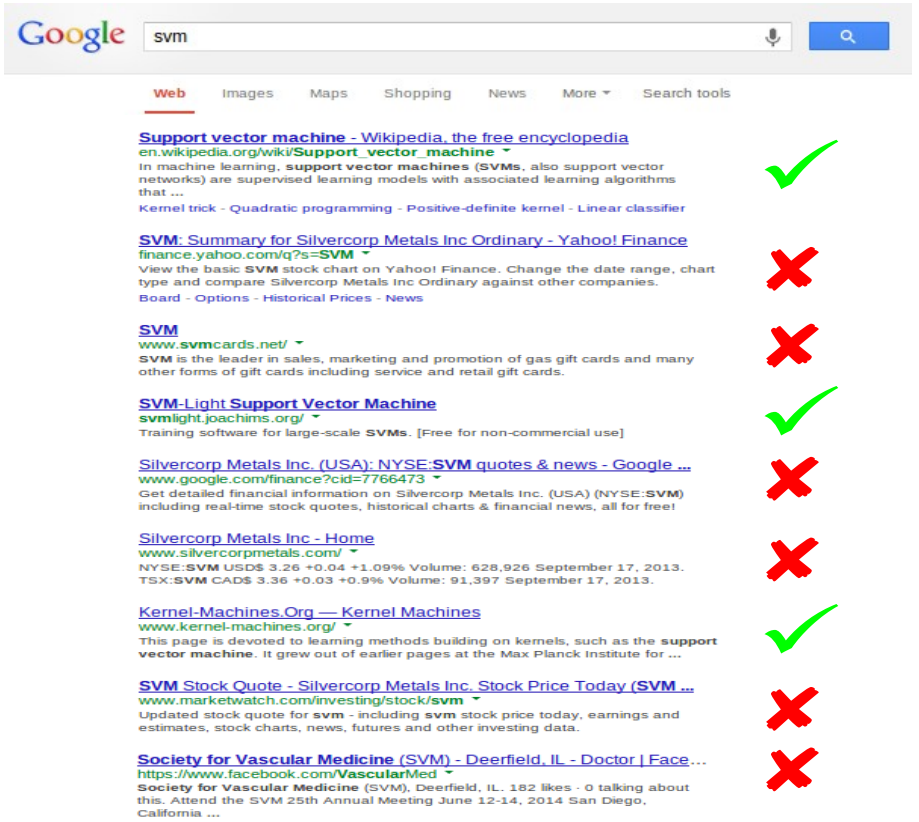
User Interest	Pref Ranking
Company	$a_1, a_2, a_3, \ldots$
ML	
Gift Cards	
Social Opt	



Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.
- User i's feedback reflects them behaving as per personal utility U_i.
- **Not** social utility U.
  - ▷ As in the case of prior work [RSJ12] for the **intrinsic** diversity problem.
- Need to infer social utility from such **conflicting, individual** feedback.

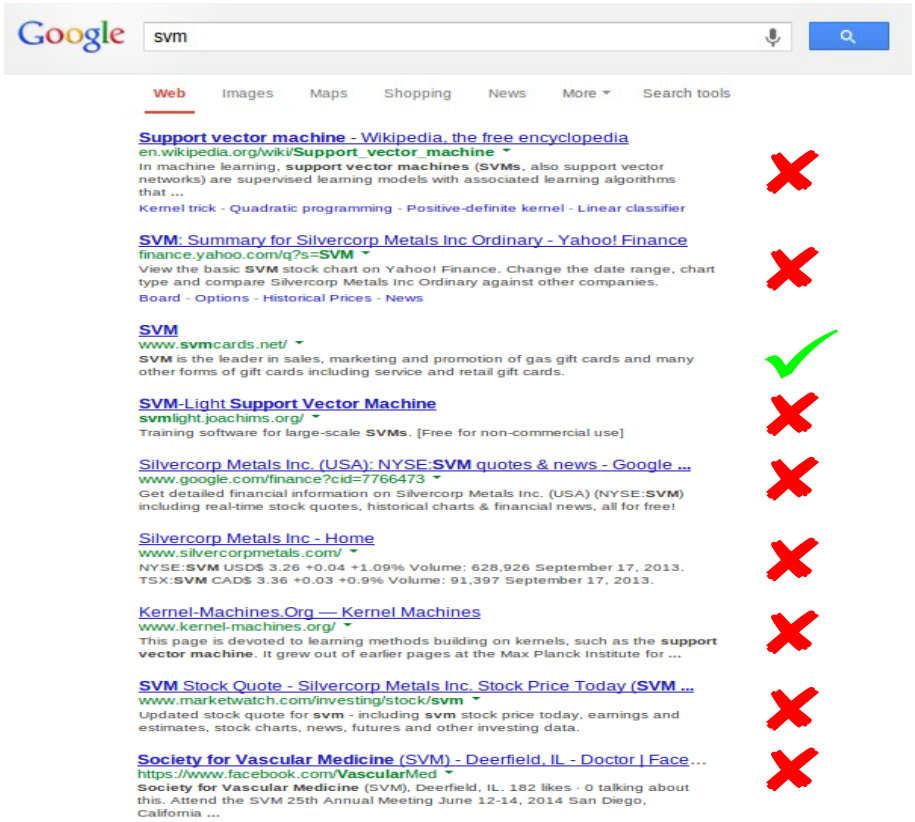
User Interest	Pref Ranking
Company	$a_1, a_2, a_3, \ldots$
ML	$b_1, b_2, b_3, \ldots$
Gift Cards	
Social Opt	



Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.
- User i's feedback reflects them behaving as per personal utility U_i.
- **Not** social utility U.
  - ▷ As in the case of prior work [RSJ12] for the **intrinsic** diversity problem.
- Need to infer social utility from such **conflicting, individual** feedback.

User Interest	Pref Ranking
Company	$a_1, a_2, a_3, \ldots$
ML	$b_1, b_2, b_3, \ldots$
Gift Cards	$c_1, c_2, c_3, \ldots$
Social Opt	

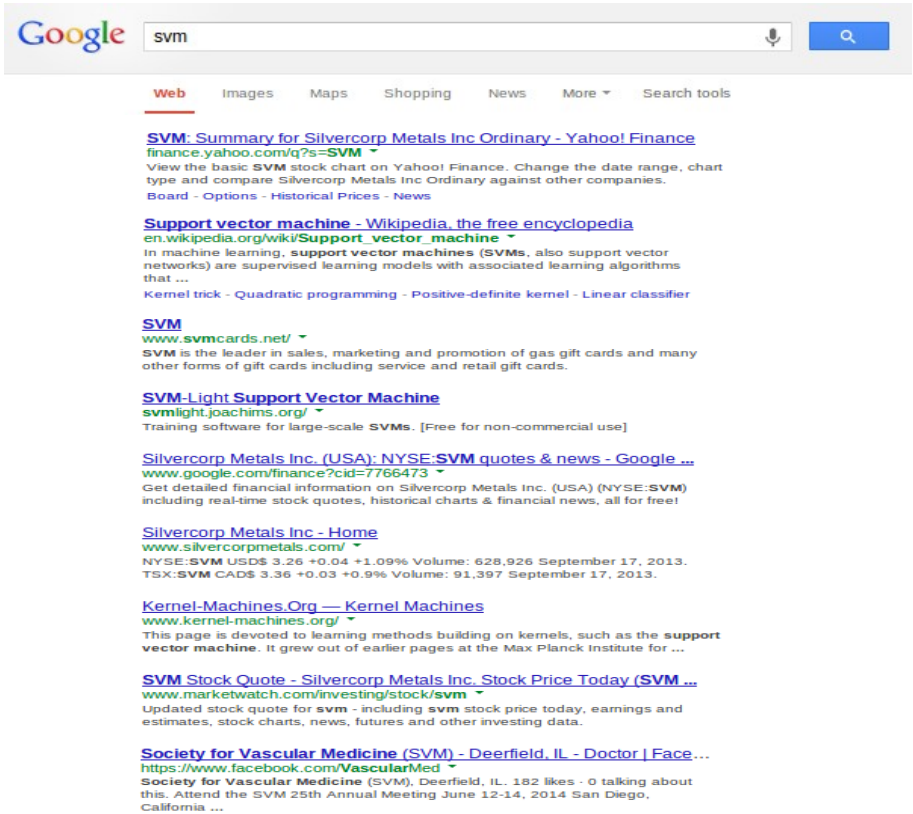


Challenge of egoistic feedback

- **Challenge:** Learn from egoistic, weak, noisy user feedback.
- User i's feedback reflects them behaving as per personal utility U_i.
- **Not** social utility U.
  - ▷ As in the case of prior work [RSJ12] for the **intrinsic** diversity problem.
- Need to infer social utility from such **conflicting, individual** feedback.

User Interest	Pref Ranking
Company	$a_1, a_2, a_3, \ldots$
ML	$b_1, b_2, b_3, \ldots$
Gift Cards	$c_1, c_2, c_3, \ldots$
Social Opt	$a_1, b_1, c_1, \ldots$

- Even if social optimal is presented, users may indicate preferences for other rankings.



Related work

- [CG98, ZCL03, CK06]: Address Extrinsic Diversity.
  - Do not use learning.
- [YJ08, SMO10, RJS11]: Use learning for diversity.
  - Require relevance labels for all user-document pairs.
- [RKJ08]: Uses online learning: Array of (decoupled) MA Bandits.
  - Learns very slowly. Does not generalize across queries.
- [SRG13]: Couples the arms together.
  - Does not generalize across queries. Hard-coded notion of diversity.
- [YG12]: Generalizes across queries.
  - Requires cardinal utilities.
- [RSJ12]: Learns from user preferences.
  - Requires all users directly optimize social utility U.

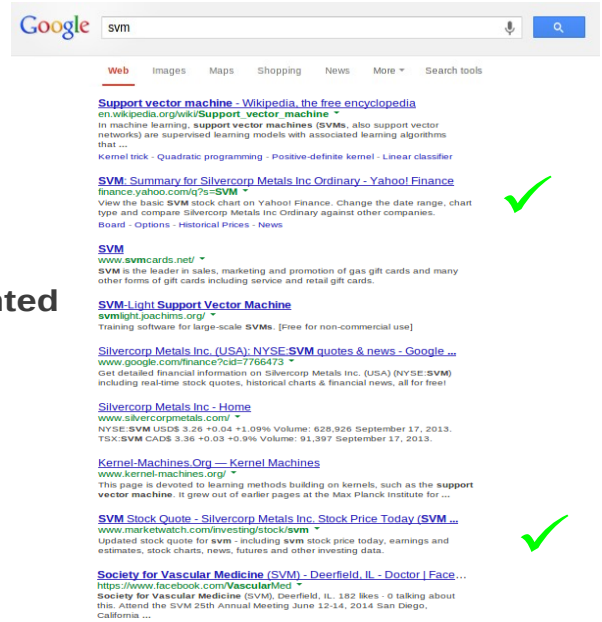
Preferential Feedback

- *What feedback do we obtain from users?*

Preferential Feedback

- *What feedback do we obtain from users?*
  - Implicit feedback (e.g. clicks) is timely and easily available.

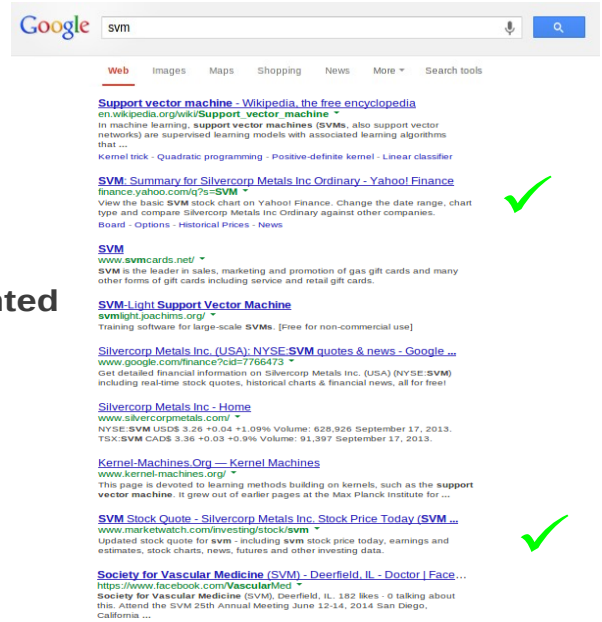
Presented (y)



Preferential Feedback

- *What feedback do we obtain from users?*
  - Implicit feedback (e.g. clicks) is timely and easily available.
- User feedback does not reflect cardinal utilities.
  - Shown in user studies [JGP+07].

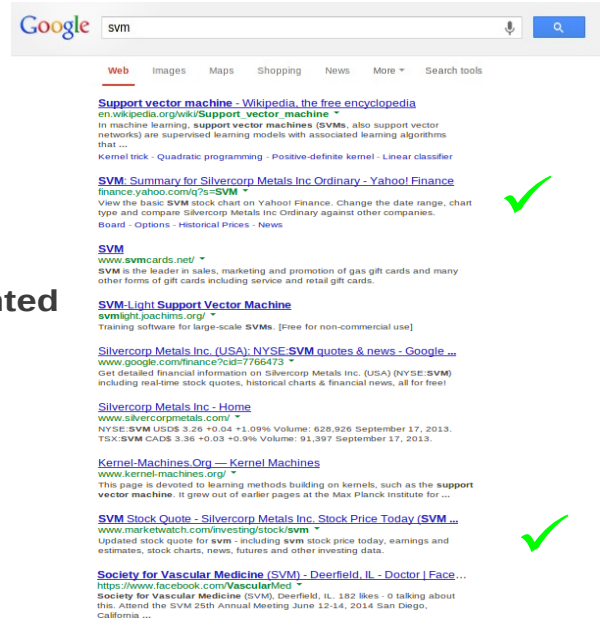
Presented (y)



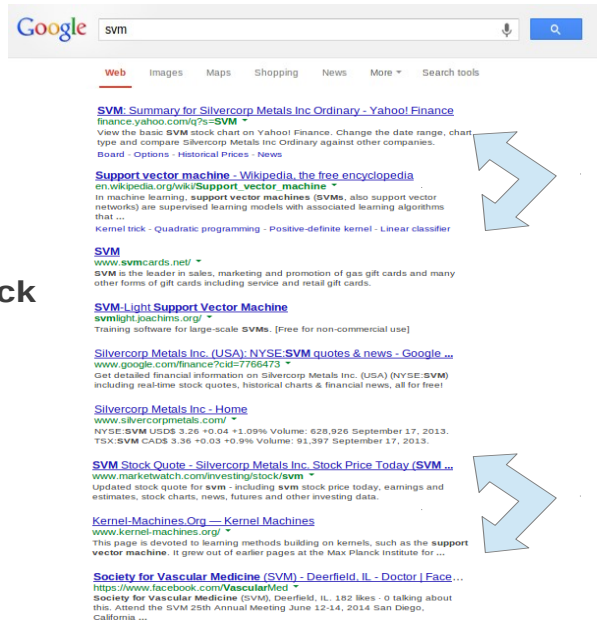
Preferential Feedback

- *What feedback do we obtain from users?*
  - ▷ Implicit feedback (e.g. clicks) is timely and easily available.
- User feedback does not reflect cardinal utilities.
  - ▷ Shown in user studies [JGP+07].
- **KEY:** Treat user feedback as preferences.

Presented (y)



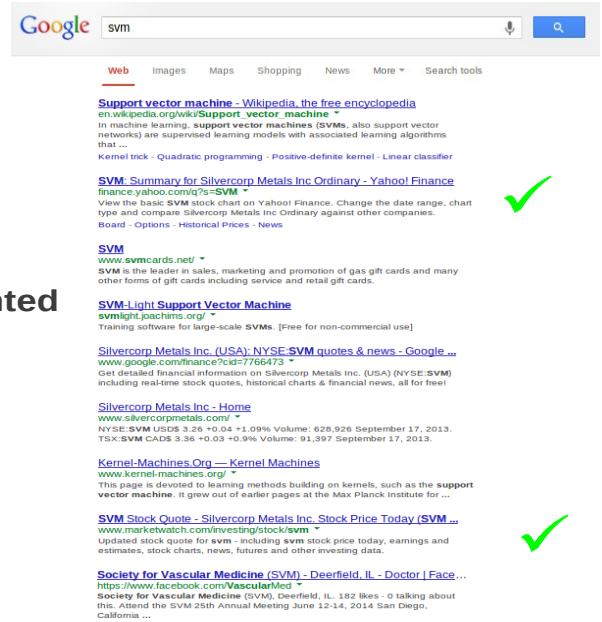
Feedback ($\bar{y}$)



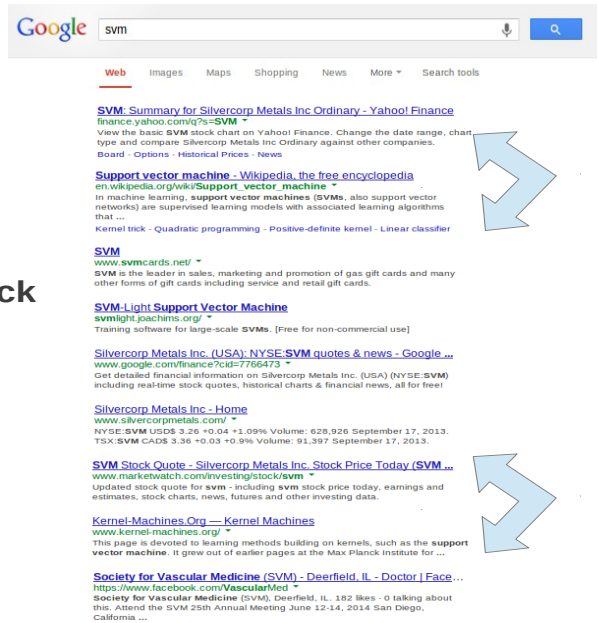
Preferential Feedback

- *What feedback do we obtain from users?*
  - ▷ Implicit feedback (e.g. clicks) is timely and easily available.
- User feedback does not reflect cardinal utilities.
  - ▷ Shown in user studies [JGP+07].
- **KEY:** Treat user feedback as preferences.
- How do we learn from such preferential feedback?

Presented (y)



Feedback ($\bar{y}$)



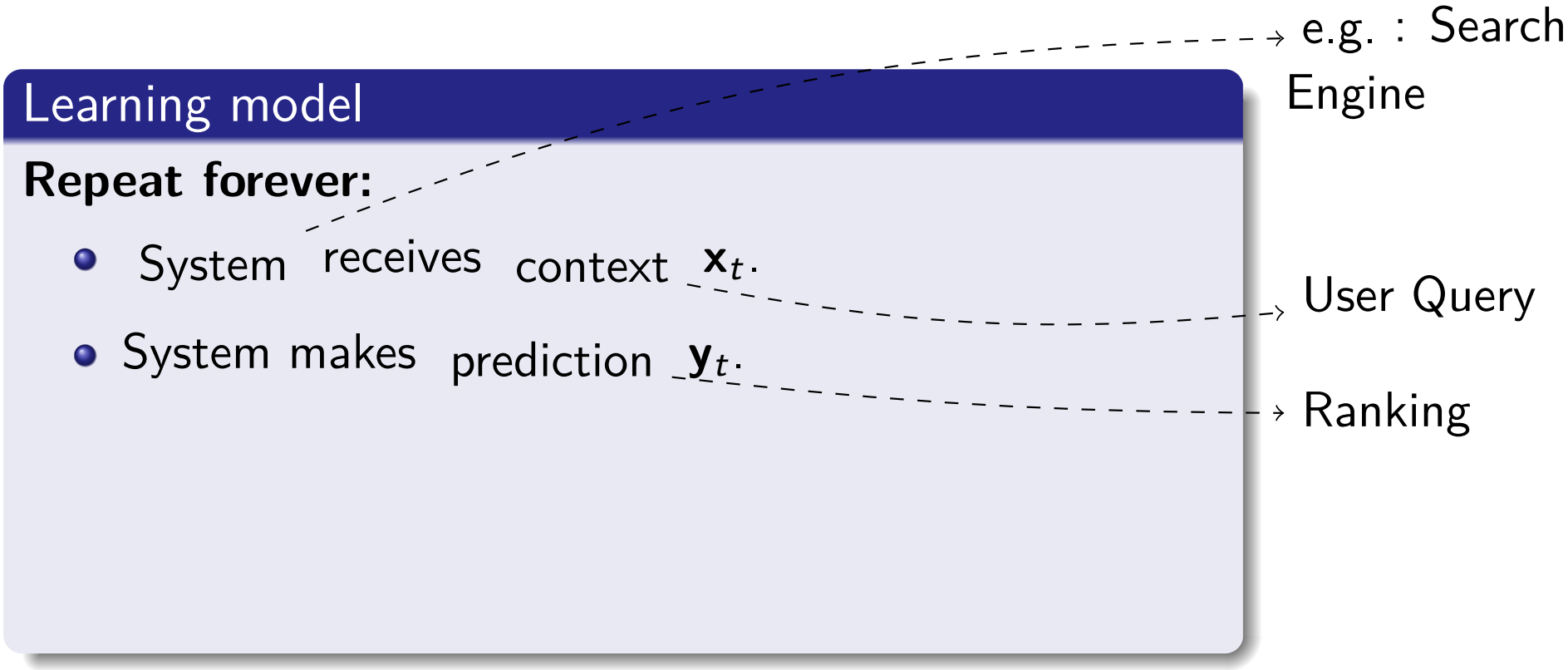
Learning from Preferences: Coactive Learning [SJ12, RJSS13]

Learning model

Repeat forever:

- System receives context $\mathbf{x}_t$.
- System makes prediction $\mathbf{y}_t$.

Learning from Preferences: Coactive Learning [SJ12, RJSS13]



Learning from Preferences: Coactive Learning [SJ12, RJSS13]

Learning model

Repeat forever:

- System receives context $\mathbf{x}_t$.
- System makes prediction $\mathbf{y}_t$.
- Regret = Regret + $U(\mathbf{x}_t, \mathbf{y}_t^*) - U(\mathbf{x}_t, \mathbf{y}_t)$

e.g. : Search Engine

User Query

Ranking

Social utility

Learning from Preferences: Coactive Learning [SJ12, RJSS13]

Learning model

Repeat forever:

- System receives context $\mathbf{x}_t$.
- System makes prediction $\mathbf{y}_t$.
- Regret = Regret + $U(\mathbf{x}_t, \mathbf{y}_t^*) - U(\mathbf{x}_t, \mathbf{y}_t)$
- System gets (preference) feedback: $U_i(\mathbf{x}_t, \bar{\mathbf{y}}_t) \geq_{\alpha,\delta} U_i(\mathbf{x}_t, \mathbf{y}_t)$

e.g. : Search Engine

User Query

Ranking

Social utility

Learning from Preferences: Coactive Learning [SJ12, RJSS13]

Learning model

Repeat forever:

- System receives context $\mathbf{x}_t$.
- System makes prediction $\mathbf{y}_t$.
- Regret = Regret + $U(\mathbf{x}_t, \mathbf{y}_t^*) - U(\mathbf{x}_t, \mathbf{y}_t)$
- System gets (preference) feedback: $U_i(\mathbf{x}_t, \bar{\mathbf{y}}_t) \geq_{\alpha,\delta} U_i(\mathbf{x}_t, \mathbf{y}_t)$

e.g. : Search Engine

User Query

Ranking

Social utility

Feedback received in terms of personal utilities U_i.
But regret is in terms of social utility U.

Learning from Preferences: Coactive Learning [SJ12, RJSS13]

e.g. : Search Engine

Learning model

Repeat forever:

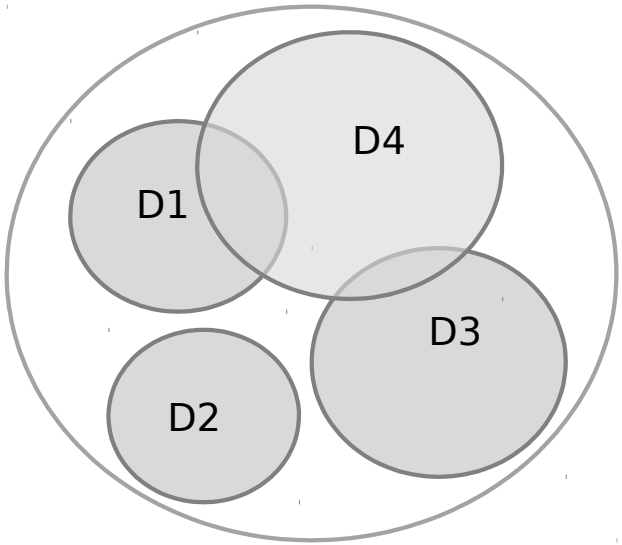
- System receives context $\mathbf{x}_t$. → User Query
- System makes prediction $\mathbf{y}_t$. → Ranking
- Regret = Regret + $U(\mathbf{x}_t, \mathbf{y}_t^*) - U(\mathbf{x}_t, \mathbf{y}_t)$ → Social utility
- System gets (preference) feedback: $U_i(\mathbf{x}_t, \bar{\mathbf{y}}_t) \geq_{\alpha,\delta} U_i(\mathbf{x}_t, \mathbf{y}_t)$

Feedback received in terms of personal utilities U_i.
But regret is in terms of social utility U.

How does we model utilities?

Modeling User Utility: Submodularity

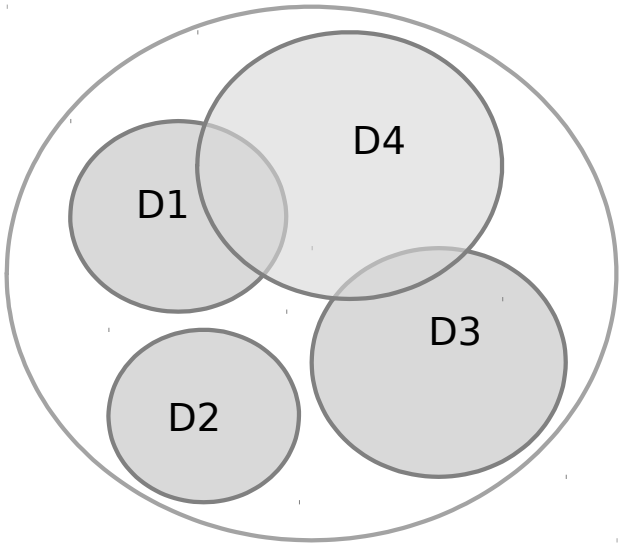
- Assume personal utilities are submodular.
- **Diminishing returns:** Marginal benefit of additional document on ML diminishes if 10 docs already shown *vs* only 1 previous doc.



- Computing ranking $\approx$ Submodular maximization
- Use simple, efficient greedy algorithm.
- Approximation guarantee of $\frac{1}{2}$ (under partition matroid constraint).

Modeling User Utility: Submodularity

- Assume personal utilities are submodular.
- **Diminishing returns:** Marginal benefit of additional document on ML diminishes if 10 docs already shown *vs* only 1 previous doc.



- Computing ranking $\approx$ Submodular maximization
- Use simple, efficient greedy algorithm.
- Approximation guarantee of $\frac{1}{2}$ (under partition matroid constraint).

- *How does this lead to diversity?*

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1					
2					
3					
4					
MAX of Col					

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1		
d_2		
d_3		
d_4		
d_5		
d_6		
d_7		
d_8		

Word	Weight
machine	5
learning	7
metal	4
silver	6

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1					
2					
3					
4					
MAX of Col					

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1	3*5 + 3*7	36
d_2		
d_3		
d_4		
d_5		
d_6		
d_7		
d_8		

Word	Weight
machine	5
learning	7
metal	4
silver	6

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1					
2					
3					
4					
MAX of Col					

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1	3*5 + 3*7	36
d_2	5*5 + 2*7	39
d_3	2*5 + 5*7	**45**
d_4	2*5 + 3*7	31
d_5	3*4 + 5*6	42
d_6	6*4 + 2*6	36
d_7	1*5 + 4*4 + 2*6	33
d_8	1*5 + 3*4 + 1*6	23

Word	Weight
machine	5
learning	7
metal	4
silver	6

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1	d_3	2	5	0	0
2					
3					
4					
MAX of Col		2	5	0	0

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1	3*5 + 3*7	36
d_2	5*5 + 2*7	39
d_3	2*5 + 5*7	**45**
d_4	2*5 + 3*7	31
d_5	3*4 + 5*6	42
d_6	6*4 + 2*6	36
d_7	1*5 + 4*4 + 2*6	33
d_8	1*5 + 3*4 + 1*6	23

Word	Weight
machine	5
learning	7
metal	4
silver	6

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1	d_3	2	5	0	0
2					
3					
4					
MAX of Col		2	5	0	0

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1	(3-2)*5	5
d_2	(5-2)*5	15
d_3	-	-
d_4	0	0
d_5	3*4 + 5*6	**42**
d_6	6*4 + 2*6	36
d_7	4*4 + 2*6	28
d_8	3*4 + 1*6	18

Word	Weight
machine	5
learning	7
metal	4
silver	6

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1	d_3	2	5	0	0
2	d_5	0	0	3	5
3					
4					
MAX of Col		2	5	3	5

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1	(3-2)*5	5
d_2	(5-2)*5	15
d_3	-	-
d_4	0	0
d_5	3*4 + 5*6	**42**
d_6	6*4 + 2*6	36
d_7	4*4 + 2*6	28
d_8	3*4 + 1*6	18

Word	Weight
machine	5
learning	7
metal	4
silver	6

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1	d_3	2	5	0	0
2	d_5	0	0	3	5
3	d_2	5	2	0	0
4					
MAX of Col		5	5	3	5

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1	(3-2)*5	5
d_2	(5-2)*5	**15**
d_3	-	-
d_4	0	0
d_5	-	-
d_6	(6-3)*4	12
d_7	(4-3)*4	4
d_8	0	0

Word	Weight
machine	5
learning	7
metal	4
silver	6

Diversity via Submodularity: An example

Posn	Doc	machine	learning	metal	silver
1	d_3	2	5	0	0
2	d_5	0	0	3	5
3	d_2	5	2	0	0
4	d_6	0	0	6	2
MAX of Col		5	5	6	5

Doc	Words
d_1	ma:3 le:3
d_2	ma:5 le:2
d_3	ma:2 le:5
d_4	ma:2 le:3
d_5	me:3 si:5
d_6	me:6 si:2
d_7	me:4 si:2 ma:1
d_8	me:3 si:1 ma:1

Doc	Marginal Benefit	
d_1	0	0
d_2	-	-
d_3	-	-
d_4	0	0
d_5	-	-
d_6	(6-3)*4	**12**
d_7	0	0
d_8	0	0

Word	Weight
machine	5
learning	7
metal	4
silver	6

More General Submodular Utility

- *Can we use other submodular functions?*

More General Submodular Utility

- *Can we use other submodular functions?*
  - ✓ **Yes.**

- Given ranking/set $\mathbf{y} = (d_{i_1}, \ldots, d_{i_n})$, aggregate features as:

$$\phi_F^j(\mathbf{x}, \mathbf{y}) = F(\gamma_1 \phi^j(\mathbf{x}, d_{i_1}), \gamma_2 \phi^j(\mathbf{x}, d_{i_2}), \ldots\ldots, \gamma_n \phi^j(\mathbf{x}, d_{i_n}))$$

  - $\phi^j(\mathbf{x}, d_i)$ is j^{th} feature of d_i.
  - F is a submodular function (modeling decision).
  - $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_n \geq 0$ are position-discount factors

More General Submodular Utility

- *Can we use other submodular functions?*
  - ✓ **Yes.**

- Given ranking/set $\mathbf{y} = (d_{i_1}, \ldots, d_{i_n})$, aggregate features as:

$$\phi_F^j(\mathbf{x}, \mathbf{y}) = F(\gamma_1 \phi^j(\mathbf{x}, d_{i_1}), \gamma_2 \phi^j(\mathbf{x}, d_{i_2}), \ldots\ldots, \gamma_n \phi^j(\mathbf{x}, d_{i_n}))$$

  - $\phi^j(\mathbf{x}, d_i)$ is j^{th} feature of d_i.
  - F is a submodular function (modeling decision).
  - $\gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_n \geq 0$ are position-discount factors

- Utility modeled as linear in aggregate features: $U(\mathbf{x}, \mathbf{y}) = \mathbf{w}_*^T \phi_F(\mathbf{x}, \mathbf{y})$
  - Submodular aggregation leads to diversity.

Social Perceptron for Ranking

1. Initialize weight vector $\mathbf{w}_1 \leftarrow \mathbf{0}$.
2. Given context $\mathbf{x}_t$ compute $\mathbf{y}_t \leftarrow \text{argmax}_{\mathbf{y}} \mathbf{w}_t^\top \phi(\mathbf{x}_t, \mathbf{y})$.
3. Observe user clicks $\mathcal{D}$.
4. Construct preference feedback $\bar{\mathbf{y}}_t \leftarrow \mathit{PrefFeedback}(\mathbf{y}_t, \mathcal{D})$.
5. $\bar{\mathbf{w}}_{t+1} \leftarrow \mathbf{w}_t + \phi(\mathbf{x}_t, \bar{\mathbf{y}}_t) - \phi(\mathbf{x}_t, \mathbf{y}_t)$
6. Clip: $\mathbf{w}^j_{t+1} \leftarrow \max(\bar{\mathbf{w}}^j_{t+1}, 0)$
7. Repeat from step 2.

Social Perceptron for Ranking

1. Initialize weight vector $\mathbf{w}_1 \leftarrow \mathbf{0}$.
2. Given context $\mathbf{x}_t$ compute $\mathbf{y}_t \leftarrow \text{argmax}_{\mathbf{y}} \mathbf{w}_t^\top \phi(\mathbf{x}_t, \mathbf{y})$.
   - ▷ Using greedy algorithm.
3. Observe user clicks $\mathcal{D}$.
4. Construct preference feedback $\bar{\mathbf{y}}_t \leftarrow \mathit{PrefFeedback}(\mathbf{y}_t, \mathcal{D})$.
5. $\bar{\mathbf{w}}_{t+1} \leftarrow \mathbf{w}_t + \phi(\mathbf{x}_t, \bar{\mathbf{y}}_t) - \phi(\mathbf{x}_t, \mathbf{y}_t)$
6. Clip: $\mathbf{w}^j_{t+1} \leftarrow \max(\bar{\mathbf{w}}^j_{t+1}, 0)$
7. Repeat from step 2.

Social Perceptron for Ranking

1. Initialize weight vector $\mathbf{w}_1 \leftarrow \mathbf{0}$.
2. Given context $\mathbf{x}_t$ compute $\mathbf{y}_t \leftarrow \text{argmax}_{\mathbf{y}} \mathbf{w}_t^\top \phi(\mathbf{x}_t, \mathbf{y})$.
   - ▷ Using greedy algorithm.
3. Observe user clicks $\mathcal{D}$.
4. Construct preference feedback $\bar{\mathbf{y}}_t \leftarrow \mathit{PrefFeedback}(\mathbf{y}_t, \mathcal{D})$.
   - ▷ Pairwise feedback.
5. $\bar{\mathbf{w}}_{t+1} \leftarrow \mathbf{w}_t + \phi(\mathbf{x}_t, \bar{\mathbf{y}}_t) - \phi(\mathbf{x}_t, \mathbf{y}_t)$
6. Clip: $\mathbf{w}^j_{t+1} \leftarrow \max(\bar{\mathbf{w}}^j_{t+1}, 0)$
7. Repeat from step 2.

Social Perceptron for Ranking

1. Initialize weight vector $\mathbf{w}_1 \leftarrow \mathbf{0}$.
2. Given context $\mathbf{x}_t$ compute $\mathbf{y}_t \leftarrow \text{argmax}_{\mathbf{y}} \mathbf{w}_t^\top \phi(\mathbf{x}_t, \mathbf{y})$.
   - ▷ Using greedy algorithm.
3. Observe user clicks $\mathcal{D}$.
4. Construct preference feedback $\bar{\mathbf{y}}_t \leftarrow \mathit{PrefFeedback}(\mathbf{y}_t, \mathcal{D})$.
   - ▷ Pairwise feedback.
5. $\bar{\mathbf{w}}_{t+1} \leftarrow \mathbf{w}_t + \phi(\mathbf{x}_t, \bar{\mathbf{y}}_t) - \phi(\mathbf{x}_t, \mathbf{y}_t)$
   - ▷ Perceptron update.
6. Clip: $\mathbf{w}^j_{t+1} \leftarrow \max(\bar{\mathbf{w}}^j_{t+1}, 0)$
   - ▷ To ensure submodularity.
7. Repeat from step 2.

Regret Bound

Definition

User feedback is **expected** α_i, δ_i-**informative** if $\bar{\xi}_t \in \Re$ is chosen s.t. :

$$\mathbf{E}_{\bar{\mathbf{y}}_t}[U_i(\mathbf{x}_t, \bar{\mathbf{y}}_t)] \geq (1+\delta_i)U_i(\mathbf{x}_t, \mathbf{y}_t) + \alpha_i \left(U_i(\mathbf{x}_t, \mathbf{y}_t^{*,i}) - U_i(\mathbf{x}_t, \mathbf{y}_t)\right) - \bar{\xi}_t.$$

Regret Bound

Definition

User feedback is **expected** α_i, δ_i**-informative** if $\bar{\xi}_t \in \Re$ is chosen s.t. :

$$\mathbf{E}_{\bar{\mathbf{y}}_t}[U_i(\mathbf{x}_t, \bar{\mathbf{y}}_t)] \geq (1+\delta_i)U_i(\mathbf{x}_t, \mathbf{y}_t) + \alpha_i \left(U_i(\mathbf{x}_t, \mathbf{y}_t^{*,i}) - U_i(\mathbf{x}_t, \mathbf{y}_t)\right) - \bar{\xi}_t.$$

Theorem

For any $\mathbf{w}_* \in \mathbf{R}^m$ *and* $\|\phi(\mathbf{x}, \mathbf{y})\|_{\ell_2} \leq R$ *the average regret of the SoPer-R algorithm can be upper bounded as:*

$$REG_T \leq \frac{1}{\eta T}\sum_{t=0}^{T-1} \mathbf{E}_i[p_i \bar{\xi}_t] + \frac{R\|\mathbf{w}_*\|}{2\eta} + \frac{\sqrt{15}R\|\mathbf{w}_*\|}{\eta\sqrt{2T}}.$$

with: $\delta_i \geq \left(\Gamma_F \cdot \frac{1-p_i}{p_i}\right)$, $\eta = \min_i \ p_i\alpha_i$.

Regret Bound: Analysis

Theorem

Average regret of the SoPer-R algorithm can be upper bounded as:

$$REG_T \leq \frac{1}{\eta T} \sum_{t=0}^{T-1} \mathbf{E}_i[p_i \bar{\xi}_t] + \frac{R\|\mathbf{w}_*\|}{2\eta} + \frac{\sqrt{15}R\|\mathbf{w}_*\|}{\eta\sqrt{2T}}.$$

with: $\delta_i \geq \left(\Gamma_F \cdot \frac{1-p_i}{p_i}\right)$, $\eta = \min_i \ p_i\alpha_i$.

- Does not depend on number of dimensions.
  - Only on feature ball radius R.
- Decays gracefully with noisy feedback (the α_is and η).
- Need not converge to optimal.
  - Partly due to NP-hardness of submodular maximization.

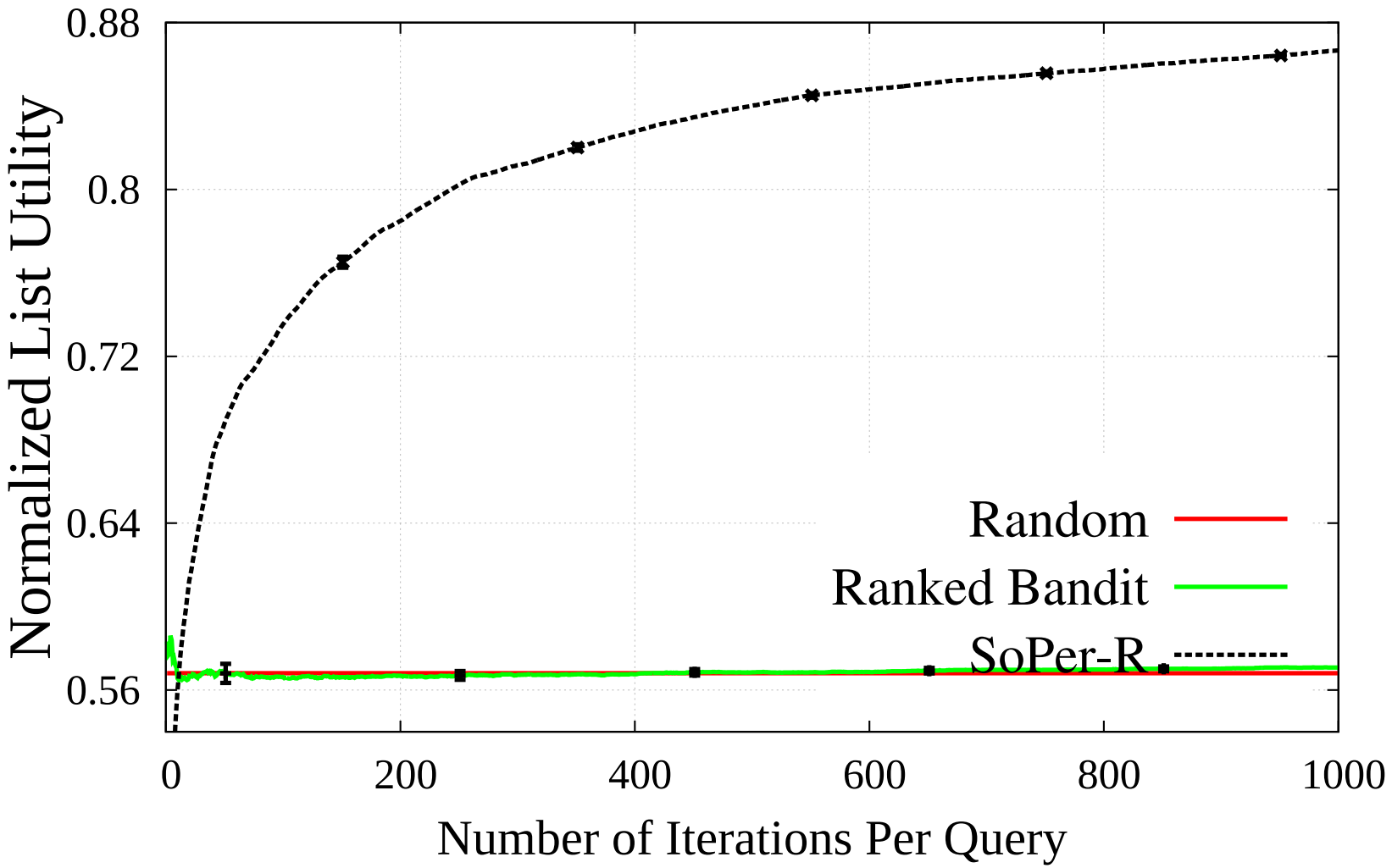
Social Perceptron for Sets

- **SoPer-S** Algorithm for predicting diverse sets.
- See paper for more details.

Experimental Setup

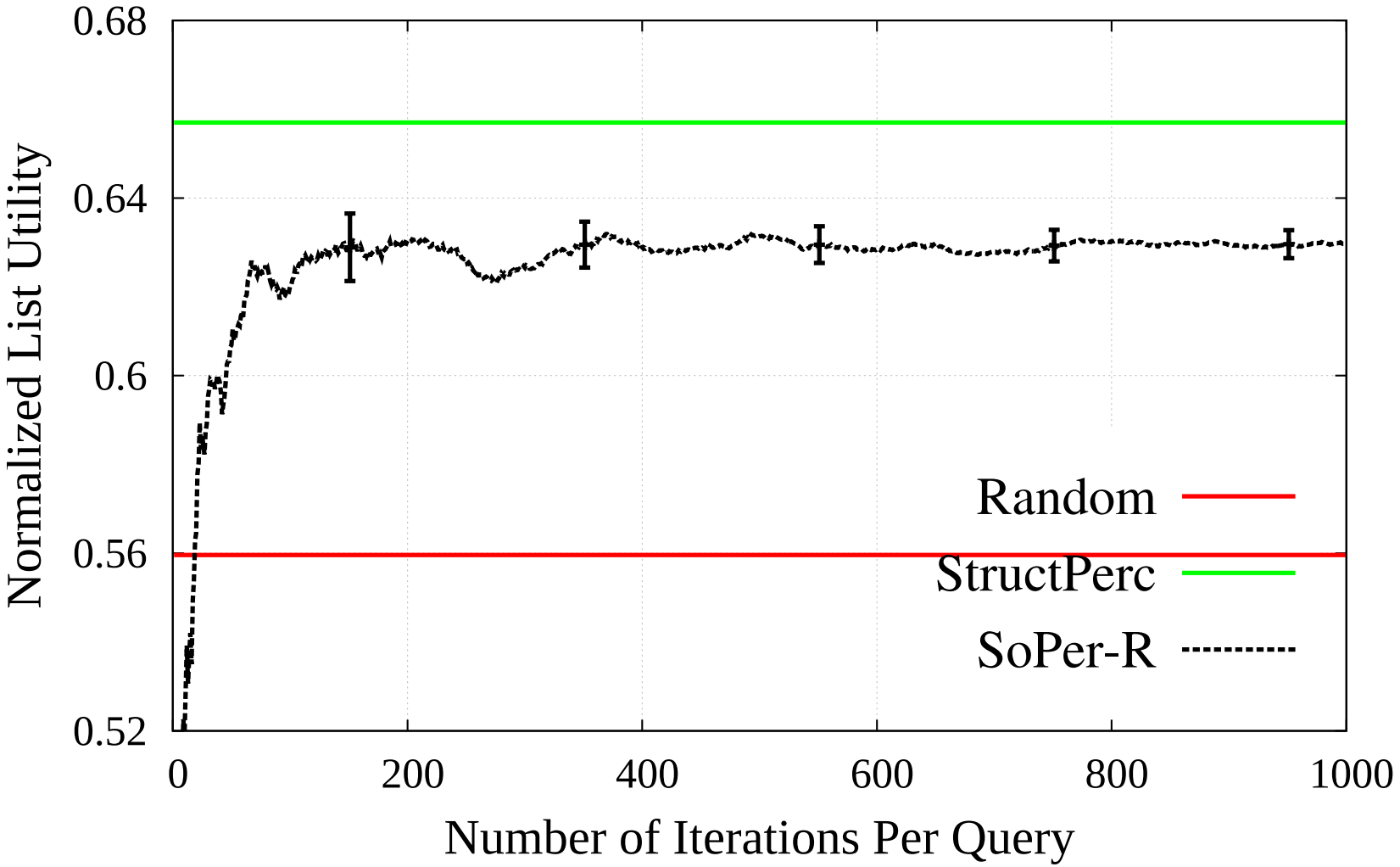
- Used standard **TREC 6-8 Interactive** search-diversification dataset:
  - Each query has 7-56 user types.
- Setup as in previous work [BJYB11, RJS11].
- Simulated user behavior:
  - Users scan rankings top to bottom.
  - Click on first document relevant to them (with small error chance).
- Utility function: Normalized DCG-coverage function (*i.e.* $F(x_1, \ldots, x_n) = \max_i \gamma_i x_i$) upto rank 5.

Learning to Diversify: Single Query



- Improved learning for single-query diversification.

Learning to Diversify: Cross-Query



- *StructPerc* is (rough) skyline: Uses optimal for training.
- **First method** to learn cross-query diversity from implicit feedback.

Robustness

User Fncn	SoPer-R Function		Rand
	MAX	SQRT	
MAX	.630 $\pm$.007	.620 $\pm$.006	.557 $\pm$.006
SQRT	.656 $\pm$.007	.654 $\pm$.007	.610 $\pm$.007

- Robust to difference between submodular functions used in *User's* utility and *Algorithm's* utility.

Robustness

User Fncn	SoPer-R Function		Rand
	MAX	SQRT	
MAX	.630 $\pm$.007	.620 $\pm$.006	.557 $\pm$.006
SQRT	.656 $\pm$.007	.654 $\pm$.007	.610 $\pm$.007

- Robust to difference between submodular functions used in *User's* utility and *Algorithm's* utility.

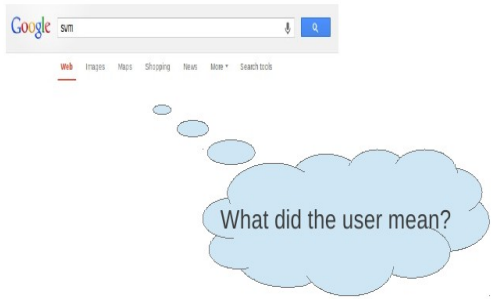
Random	No Noise	Noise
.557 $\pm$.006	.630 $\pm$.007	.631 $\pm$.007

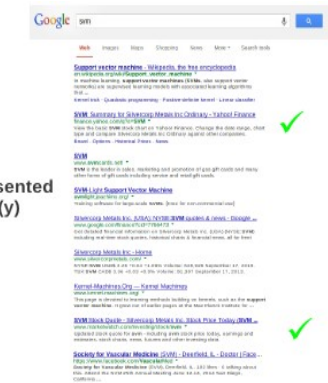
- Works even if user feedback is noisy

Conclusions

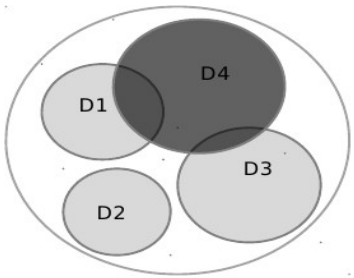
- Proposed online-learning algorithms for aggregating conflicting user preferences of a diverse population.
  - Utilizes the coactive learning model.
- Modeled user utilities as submodular.
- Provided regret bounds for algorithms.
- Works well empirically and is robust.

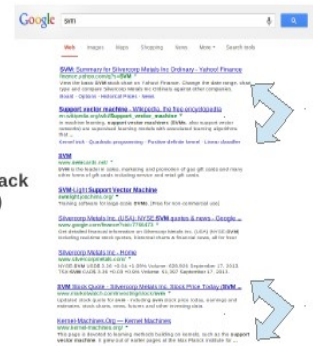
THANKS





QUESTIONS?





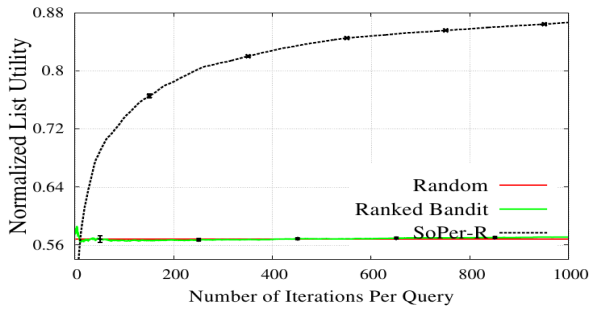
Poster #15

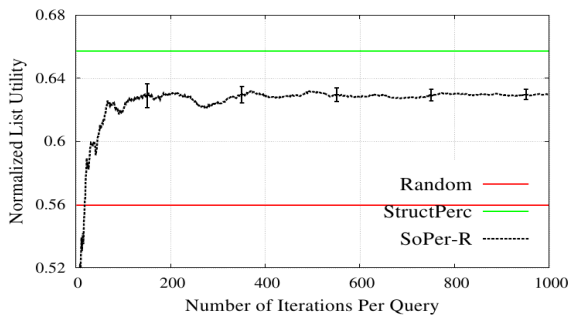
Theorem

Average regret of the SoPer-R algorithm can be upper bounded as:

$$REG_T \leq \frac{1}{\eta T}\sum_{t=0}^{T-1}\mathbf{E}_i[p_i\tilde{\xi}_t] + \frac{R\|\mathbf{w}_*\|}{2\eta} + \frac{\sqrt{15}R\|\mathbf{w}_*\|}{\eta\sqrt{2T}}.$$

with: $\delta_i \geq \left(\Gamma_F \cdot \frac{1-p_i}{p_i}\right)$, $\eta = \min_i \; p_i\alpha_i$.





C. Brandt, T. Joachims, Yisong Yue, and J. Bank.

Dynamic ranked retrieval.

In *WSDM*, 2011.

Jaime Carbonell and Jade Goldstein.

The use of MMR, diversity-based reranking for reordering documents and producing summaries.

In *SIGIR*, pages 335–336, 1998.

Harr Chen and David R. Karger.

Less is more: probabilistic models for retrieving fewer relevant documents.

In *SIGIR*, pages 429–436, 2006.

T. Joachims, L. Granka, Bing Pan, H. Hembrooke, F. Radlinski, and G. Gay.

Evaluating the accuracy of implicit feedback from clicks and query reformulations in web search.

TOIS, 25(2), April 2007.

Karthik Raman, Thorsten Joachims, and Pannagadatta Shivaswamy.

Structured learning of two-level dynamic rankings.

In *CIKM*, pages 291–296, 2011.

Karthik Raman, Thorsten Joachims, Pannaga Shivaswamy, and Tobias Schnabel.

Stable coactive learning via perturbation.
In *ICML*, pages 837–845, 2013.


Filip Radlinski, Robert Kleinberg, and Thorsten Joachims.
Learning diverse rankings with multi-armed bandits.
In *ICML*, pages 784–791, 2008.


Karthik Raman, Pannaga Shivaswamy, and Thorsten Joachims.
Online learning to diversify from implicit feedback.
In *KDD*, pages 705–713, 2012.


Pannaga Shivaswamy and Thorsten Joachims.
Online structured prediction via coactive learning.
In *ICML*, 2012.


Rodrygo L.T. Santos, Craig Macdonald, and Iadh Ounis.
Selectively diversifying web search results.
In *CIKM*, pages 1179–1188, 2010.


Aleksandrs Slivkins, Filip Radlinski, and Sreenivas Gollapudi.
Ranked bandits in metric spaces: learning optimally diverse rankings over large document collections.
JMLR, 14:399–436, 2013.


Yisong Yue and Carlos Guestrin.

Linear submodular bandits and their application to diversified retrieval.

In *NIPS*, pages 2483–2491, 2012.


Yisong Yue and Thorsten Joachims.

Predicting diverse subsets using structural SVMs.

In *ICML*, pages 1224–1231, 2008.


Cheng Xiang Zhai, William W. Cohen, and John Lafferty.

Beyond independent relevance: methods and evaluation metrics for subtopic retrieval.

In *SIGIR*, pages 10–17, 2003.

Experimental Setup Details

- **TREC 6-8 Interactive** diversification dataset:
  - Contains 17 queries. Each has 7-56 user types. Binary relevance labels.
  - Similar results observed for WEB diversification dataset.
- Setup details:
  - Re-ranking documents relevant to ≥ 1 user.
  - Probability of user type $\propto$ # of documents relevant to user.
- DCG-position discounting: $\gamma_i = \frac{1}{\log_2(1+i)}$.

Regret Bound

Definition

User feedback is **expected** α_i, δ_i**-informative** for user with personal utility function U_i, if $\bar{\xi}_t \in \Re$ is chosen s.t. for some given $\alpha_i \in [0,1]$ and $\delta_i > 0$:

$$\mathbf{E}_{\bar{\mathbf{y}}_t}[U_i(\mathbf{x}_t, \bar{\mathbf{y}}_t)] \geq (1+\delta_i)U_i(\mathbf{x}_t, \mathbf{y}_t) + \alpha_i \left(U_i(\mathbf{x}_t, \mathbf{y}_t^{*,i}) - U_i(\mathbf{x}_t, \mathbf{y}_t)\right) - \bar{\xi}_t.$$

Theorem

For any $\mathbf{w}_* \in \mathbf{R}^m$ *and* $\|\phi(\mathbf{x}, \mathbf{y})\|_{\ell_2} \leq R$ *the average regret of the SoPer-R algorithm can be upper bounded as:*

$$REG_T \leq \frac{1}{\eta T}\sum_{t=0}^{T-1} \mathbf{E}_i[p_i \bar{\xi}_t] + \frac{\beta R\|\mathbf{w}_*\|}{\eta} + \frac{\sqrt{2}\sqrt{4-\beta^2}R\|\mathbf{w}_*\|}{\eta\sqrt{T}}.$$

with: $\delta_i \geq \left(\Gamma_F \cdot \frac{1-p_i}{p_i}\right)$, $\eta = \min_i \ p_i\alpha_i$ *and* $\beta = (1-\beta_{gr}) = \frac{1}{2}$.