Modeling Impression Discounting In Large-scale Recommender Systems

Pei Lee, Laks V.S. Lakshmanan

Mitul Tiwari, Sam Shah

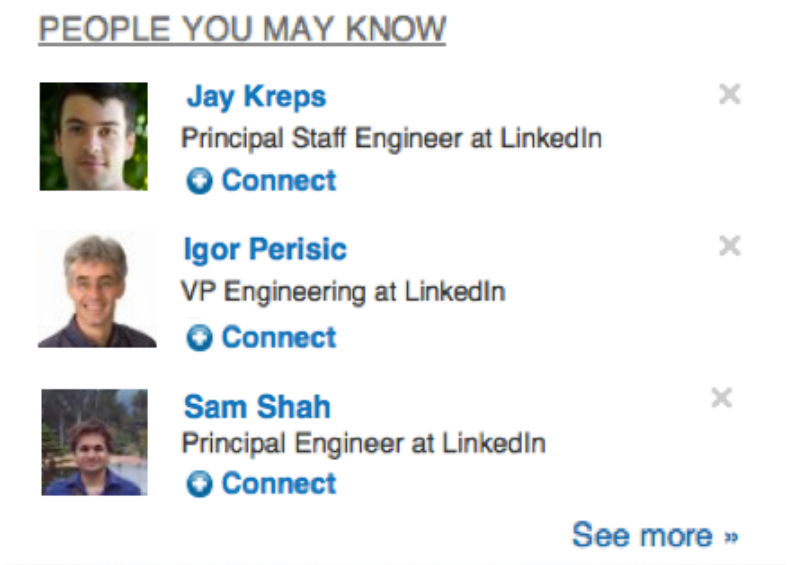
LinkedIn By The Numbers

300M members

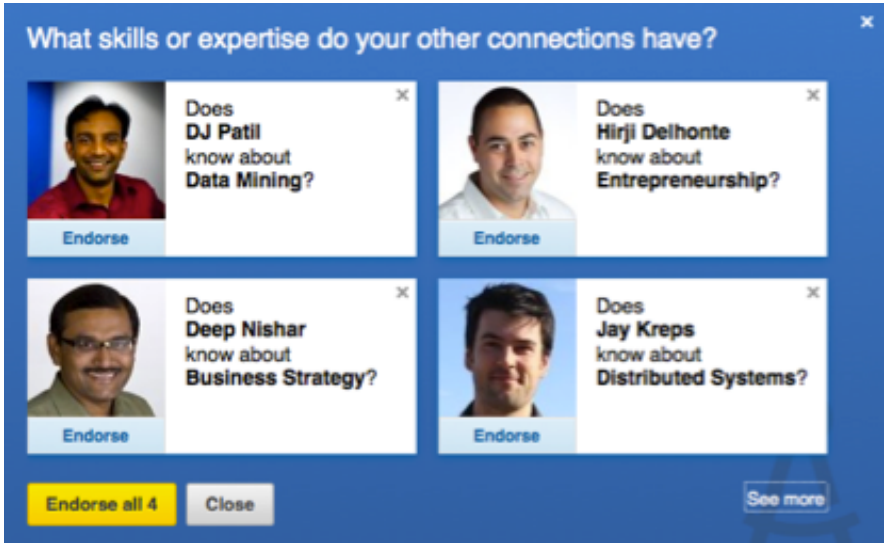
2 new members/sec

Large-scale Recommender Systems at LinkedIn

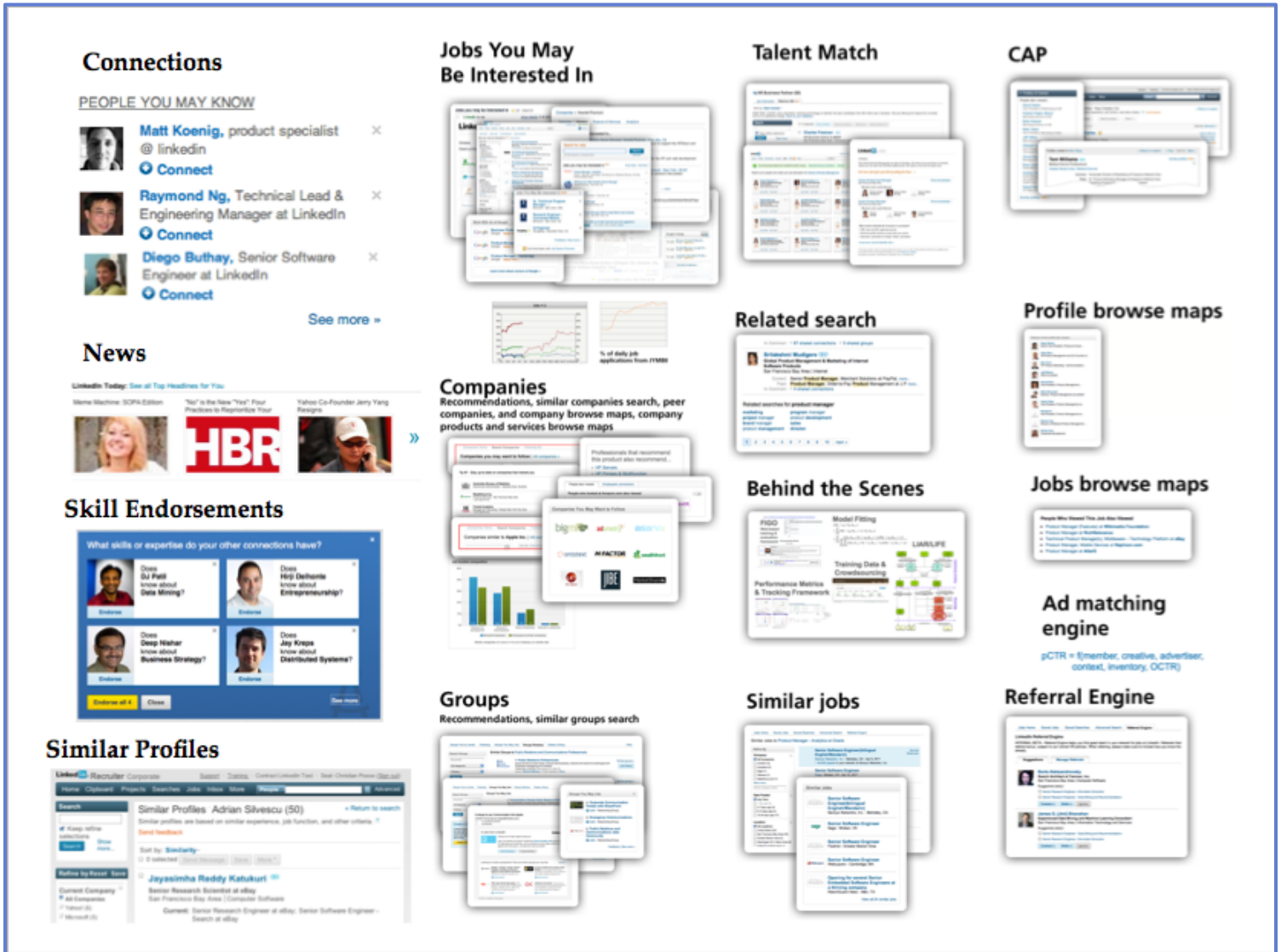
- People You May Know



- Skills Endorsements



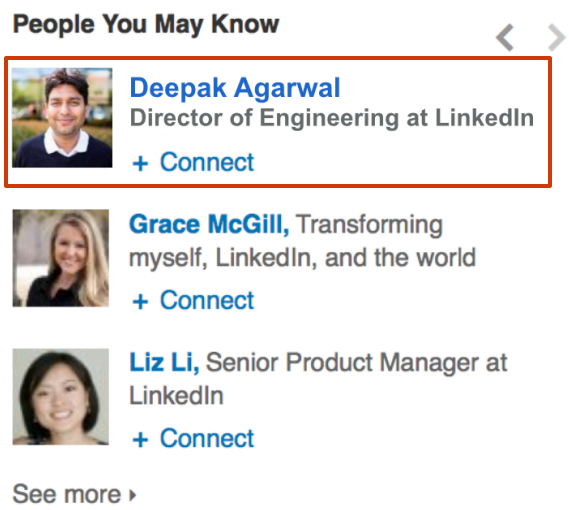
Recommender Systems at LinkedIn



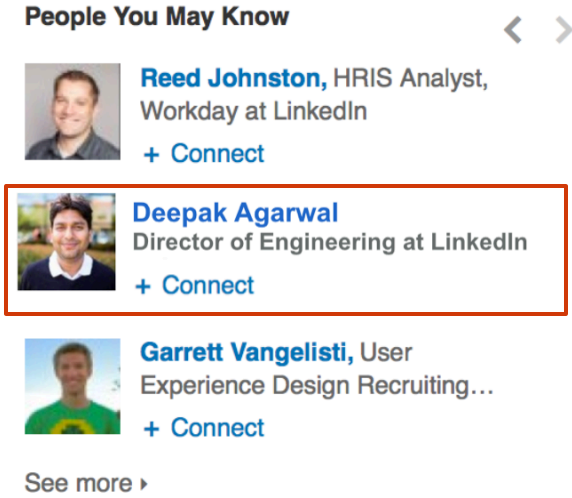
What is Impression Discounting?

- Examples
  - People You May Know
  - Skills Endorsement
- Problem Definition

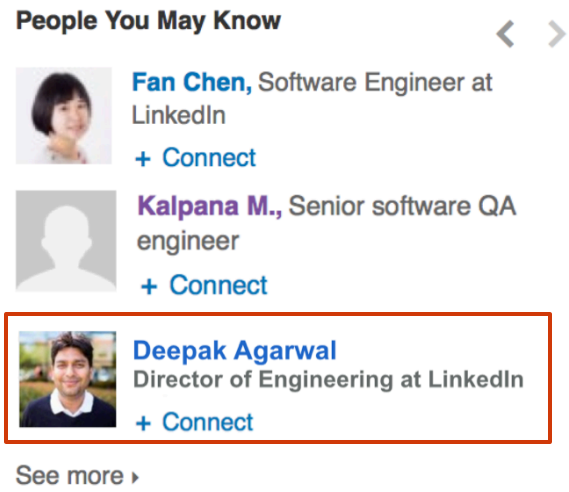
Example: People You May Know



5 days ago:
Recommended, but not invited



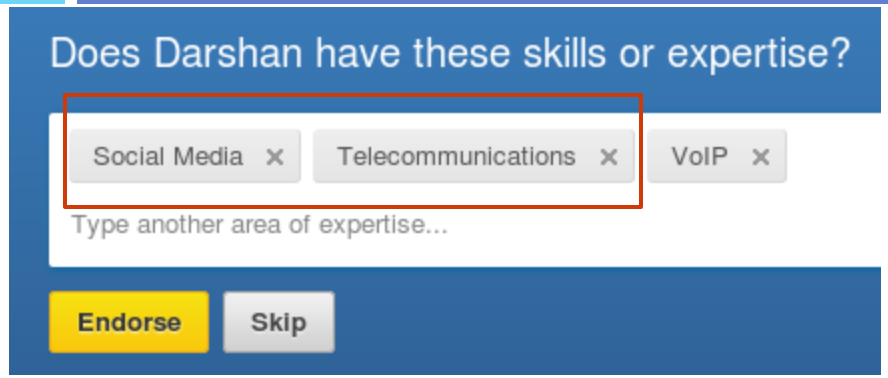
1 day ago:
Recommended again, but not invited



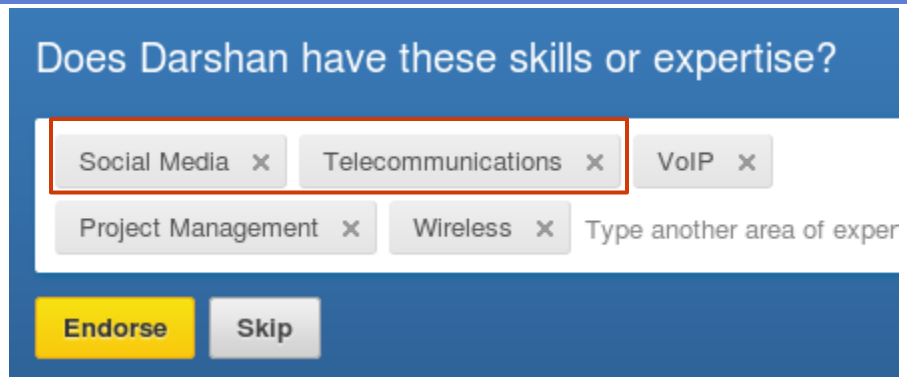
Today:
If we recommend again, would you invite or not?

If you are not likely to invite, we better discount this recommendation.

Example: Skills Endorsement



Recommended



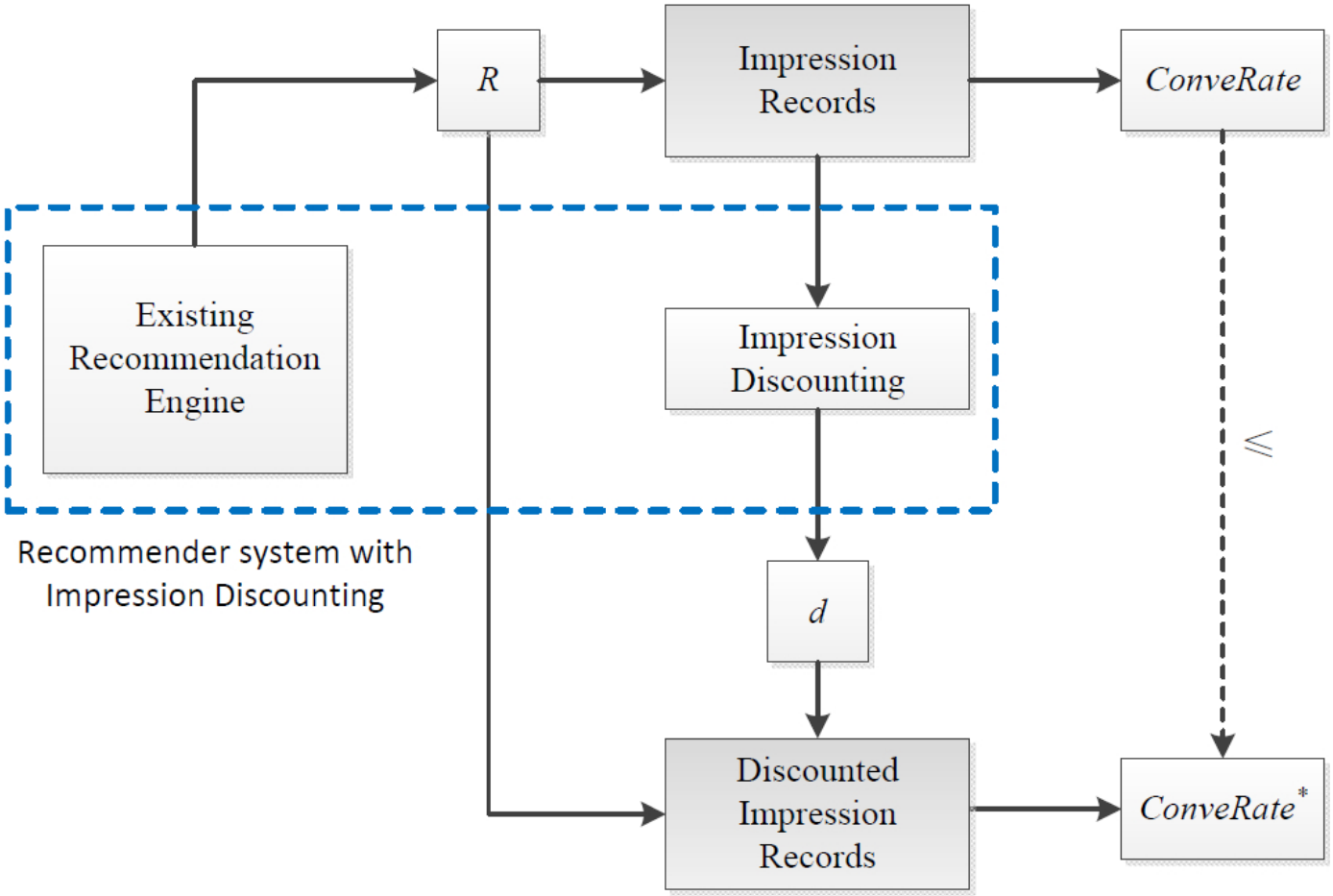
Recommended Again

- If you are likely to endorse, we will show a skills suggestion again;
- Otherwise, we should leave this space for other skills suggestions

Problem Definition

- Impressions: recommendations shown to user
- Conversion: positive action - invite, endorse, etc.
- In natural language:
  - We already impressed an item several times with no conversion. How much should we discount that item?
- More formally:
  - For an user *u* and item *i*, given an impression history (T_1, T_2 , …, T_n) between *u* and *i*, can we predict the conversion rate for *u* on *i*?

Impression Discounting Framework



Outline

- Define an impression
  - User, Item, Conversion Rate
  - Features
    - How many times the user saw this item?
    - When is the last time the user saw this item? …
- Data analysis: impression features and conversion relationship
- Impression discounting model fitting
- Experimental evaluation

Define an Impression

$$\mathcal{T} = (user, item, conversion, [behavior1, behavior2, \cdots], t, R)$$

where:

- $user$ is a user ID;
- $item$ is a recommendation item ID;
- $conversion$ is a boolean type to describe whether or not $user$ takes an action on $item$ in this impression;
- $behavior$ is an observed feature of interaction;
- t is the time stamp of impression;
- R is the recommendation score of $item$ to $user$, which is provided by the recommendation engine.

Features on Impressed Items

- LastSeen (recency):
  - the day difference between the last impression and the current impression, associated with the same (user, item)
- ImpCount (frequency):
  - the number of historical impressions before the current impression, associated with the same (user, item)
- Position:
  - the offset of item in the recommendation list of user
- UserFreq:
  - the interaction frequency of user in a recommender system

PYMK Dataset

- Data: 1.09 B impression tuples
- Training dataset: 80% of data
  - 0.55 billion unique impressions
  - 0.87 billion total impressions
  - 20 millions invitations
- Testing dataset: 20% of data
  - 0.14 billion unique impressions (3.7%)
  - 0.22 billion total impressions (2.4%)
  - 5.2 millions invitations

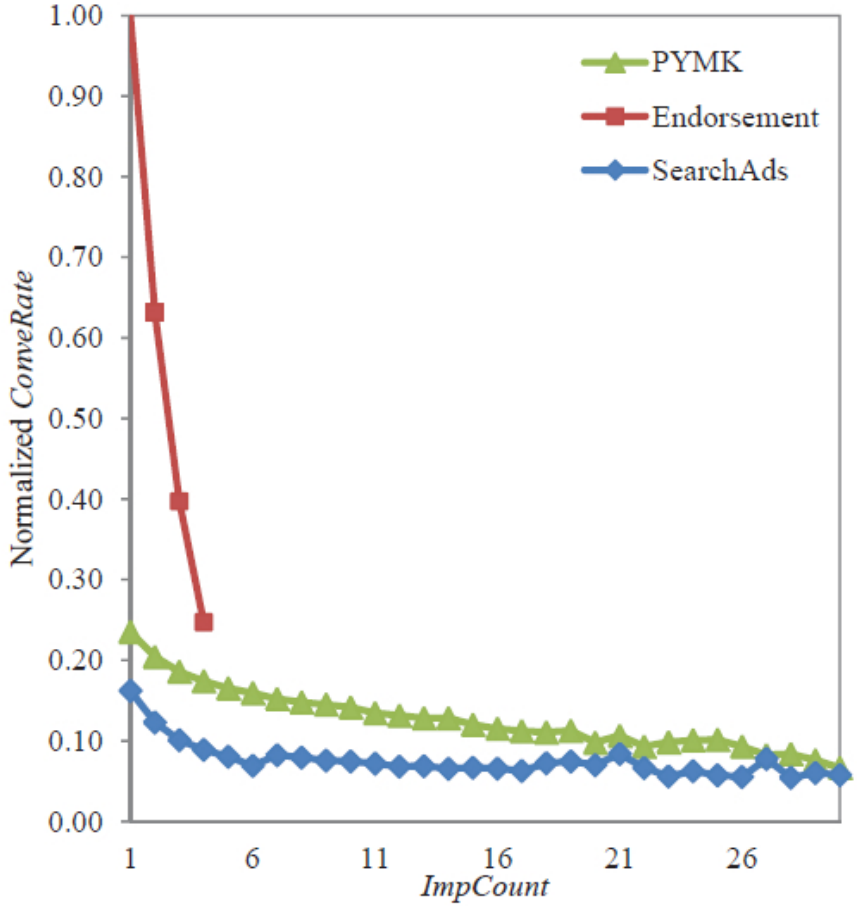
Skills Endorsement Dataset

- Total dataset size: 190 million impression tuples
- Training dataset: 80% of data
- Testing dataset: 20% of data

Tencent SearchAds Dataset

- Publicly available for KDD Cup 2012 by the Tencent search engine
- Total Size: 150 million impression sequences
- CTR of the Ad at the 1st, 2nd and 3rd position: 4.8%, 2.7%, and 1.4%

Conversion Rate Changes with Impression Count

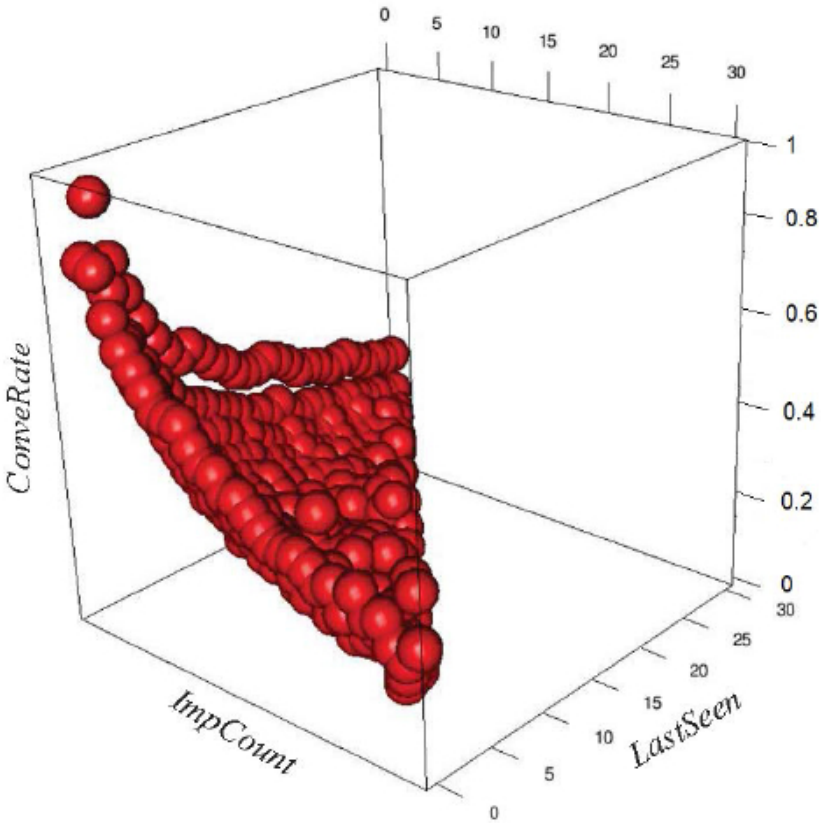


If we show a item to a user repeatedly, the conversion rate decreases

Figure 3: The change of Normalized $ConveRate$ with increasing *ImpCount* on three real-world impression data sets.

PYMK Invitation Rate Changes with Impression Count and Last Seen

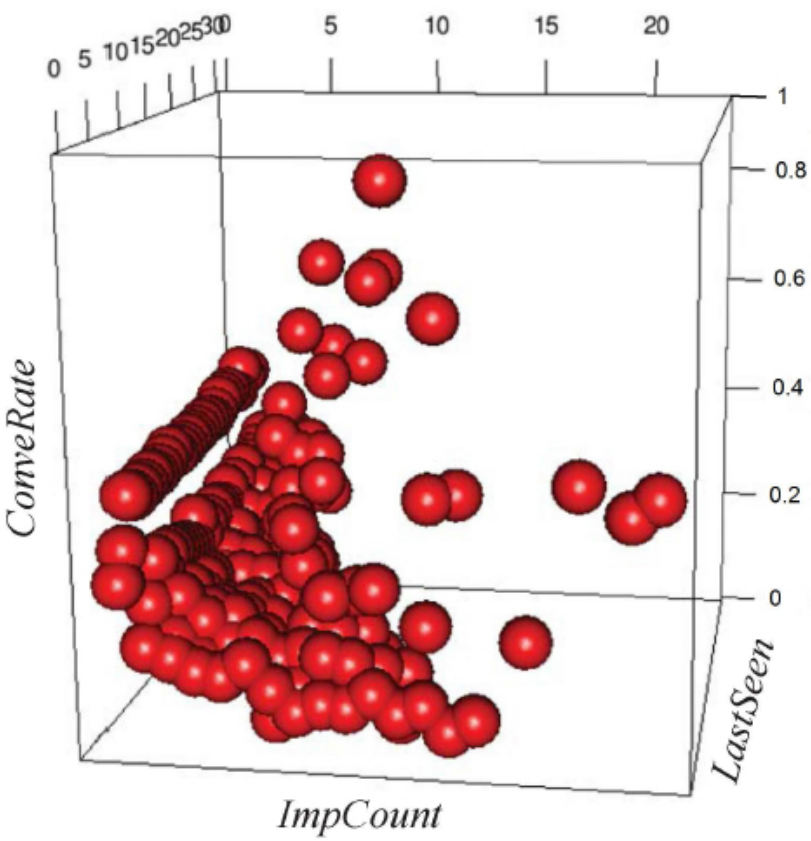
- The conversion rate decreases with both ImpCount and LastSeen



(a) PYMK 3-D

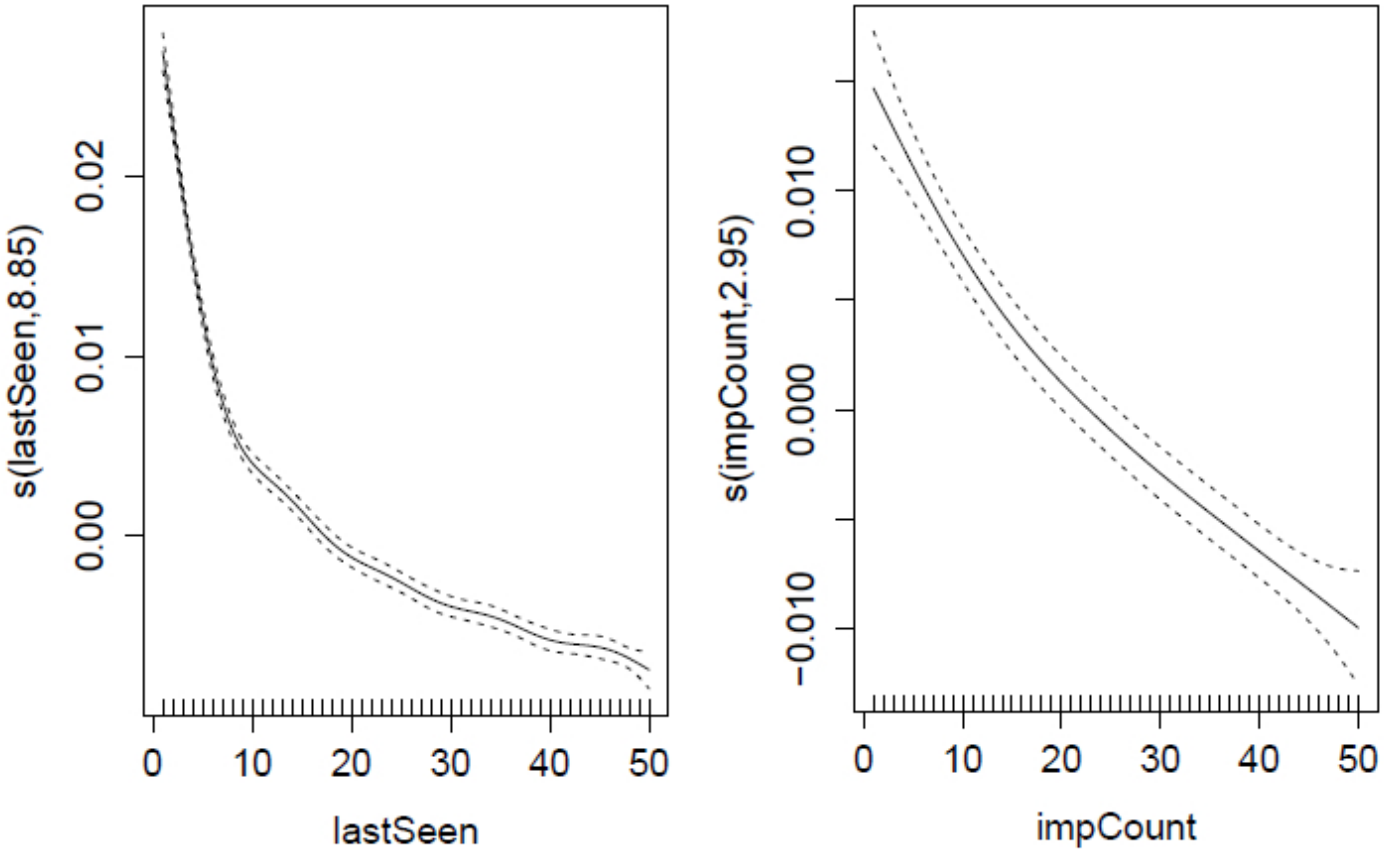
Endorsement Rate Changes with Impression Count and Last Seen

- Similar observation: The conversion rate also decreases with both ImpCount and LastSeen



(b) Endorsement 3-D

Correlation Confidence Analysis

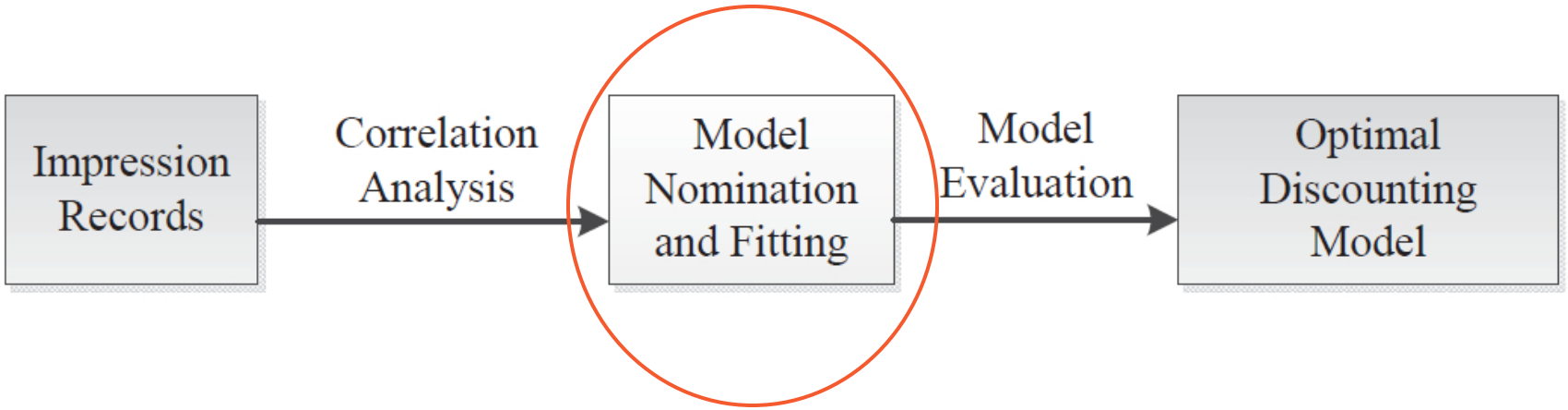


(a) *LastSeen* Correlation (b) *ImpCount* Correlation

Conversion rate has a strong, negative correlation with LastSeen and ImpCount

Impression Discounting Model Fitting

- Workflow:

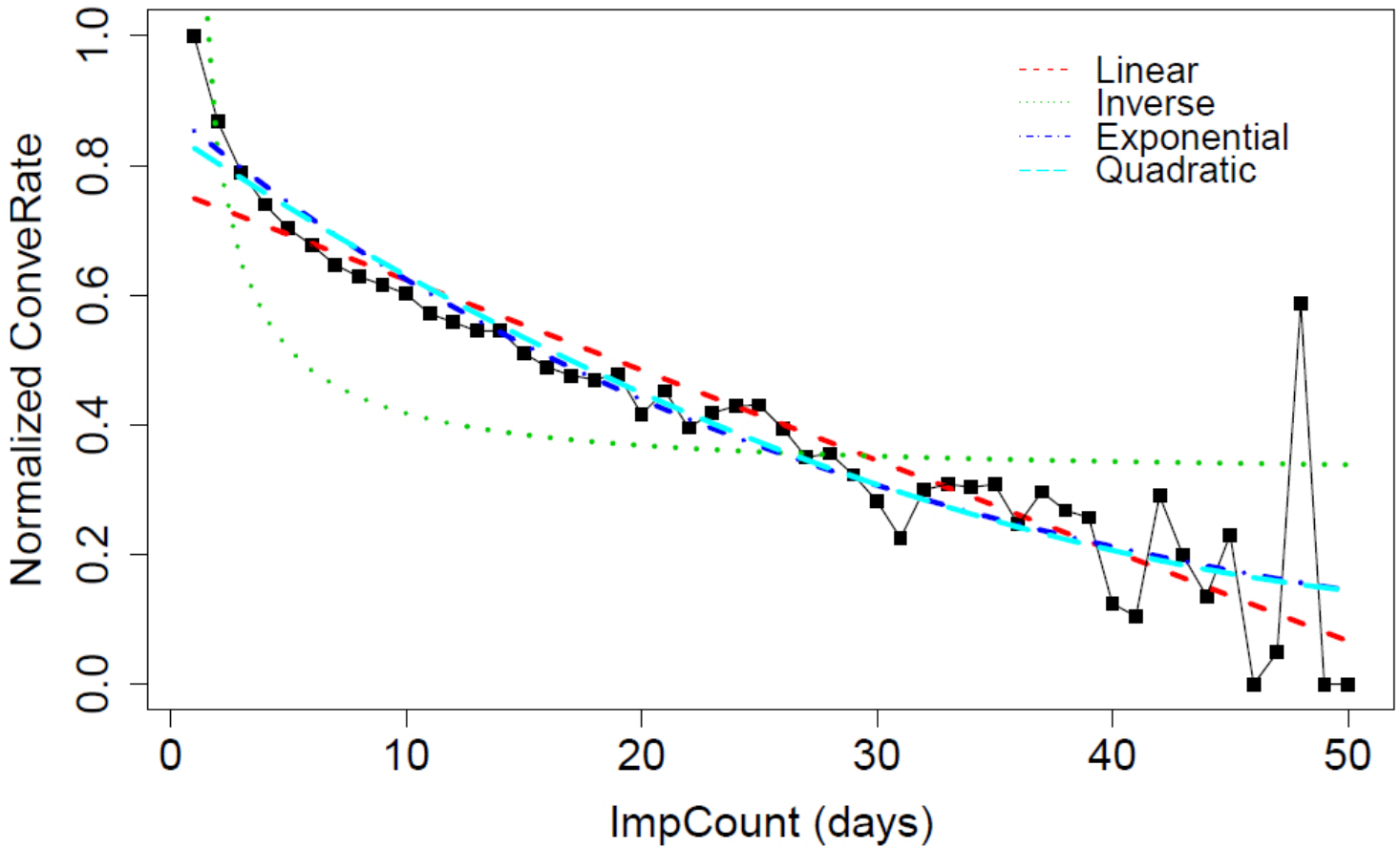


- Model fitting process:
  - Fundamental discounting functions
  - Aggregation Model
  - Offline and online

Discounting Functions

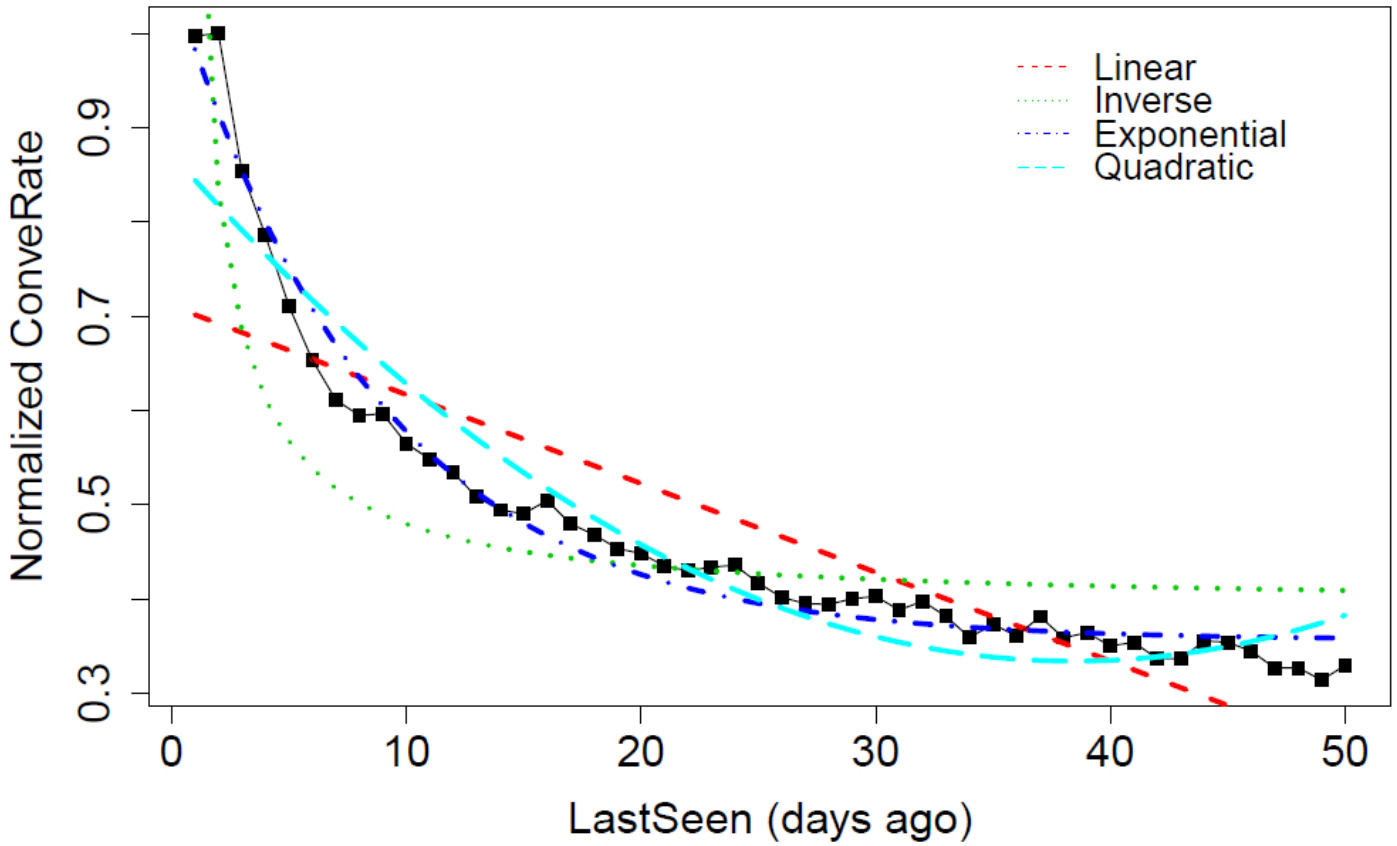
- Linear Discounting: $f_L(x) = \alpha_1 \cdot x + \alpha_2$;
- Inverse Discounting: $f_I(x) = \frac{\alpha_1}{x} + \alpha_2$;
- Exponential Discounting: $f_E(x) = e^{\alpha_1 \cdot x + \alpha_2}$;
- Quadratic Discounting: $f_Q(x) = \alpha_1 (x - \alpha_2)^2 + \alpha_3$.

Conversion Rate with ImpCount



(b) Conversion rate vs. *ImpCount*

Conversion Rate with LastSeen



(a) Conversion rate vs. *LastSeen*

Aggregation Models

- Linear Aggregation

$$\hat{y} = \sum_{i=1}^{m} w_i f(X_i)$$

- Multiplicative Aggregation

$$\hat{y} = w \prod_{i=1}^{m} f(X_i)$$

$f(X_i)$ is one of discounting function given earlier

Aggregation Models

- Linear Aggregation

$$\hat{y} = \sum_{i=1}^{m} w_i f(X_i)$$

- Multiplicative Aggregation

$$\hat{y} = w \prod_{i=1}^{m} f(X_i)$$

$f(X_i)$ is one of discounting function given earlier

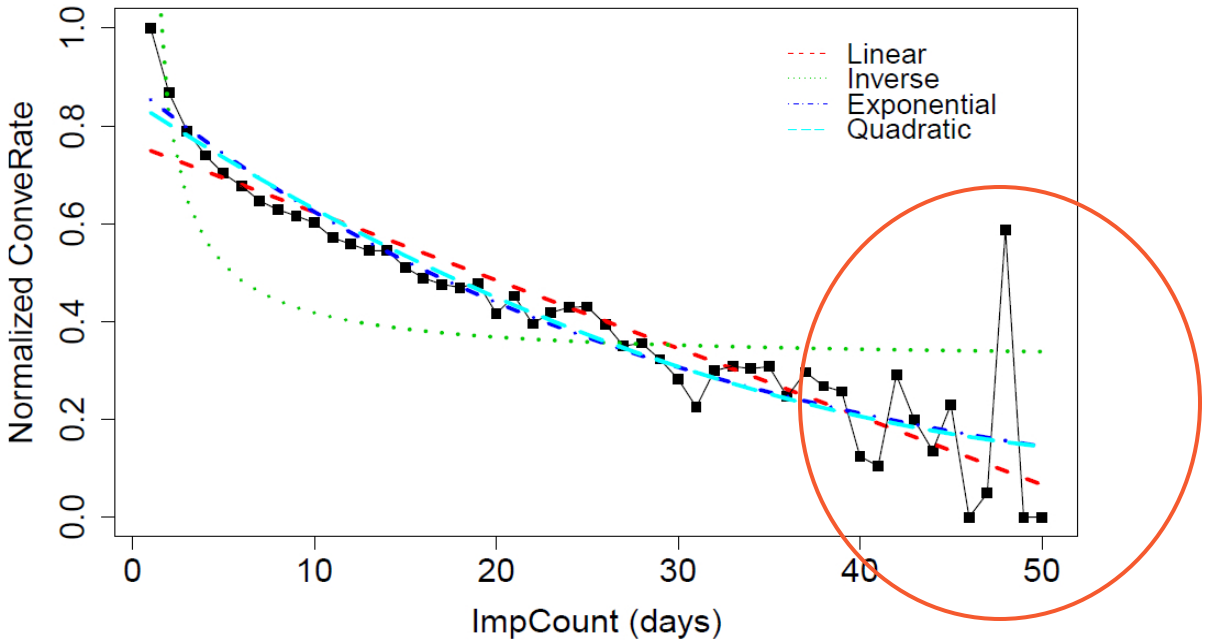
- Impression Discounting Factor

$$d = \frac{\tilde{y}}{\max \tilde{y}}$$

$0<d\leq 1$; will be applied to discount the recommendation

Anti-Noise Regression Model

- The sparsity of observations increases variance in the conversion rate.
- Typically happens when the number of observations are relatively low.



(b) Conversion rate vs. *ImpCount*

Density Weighted Regression

- Add a weight with square error
- Original RMSE:

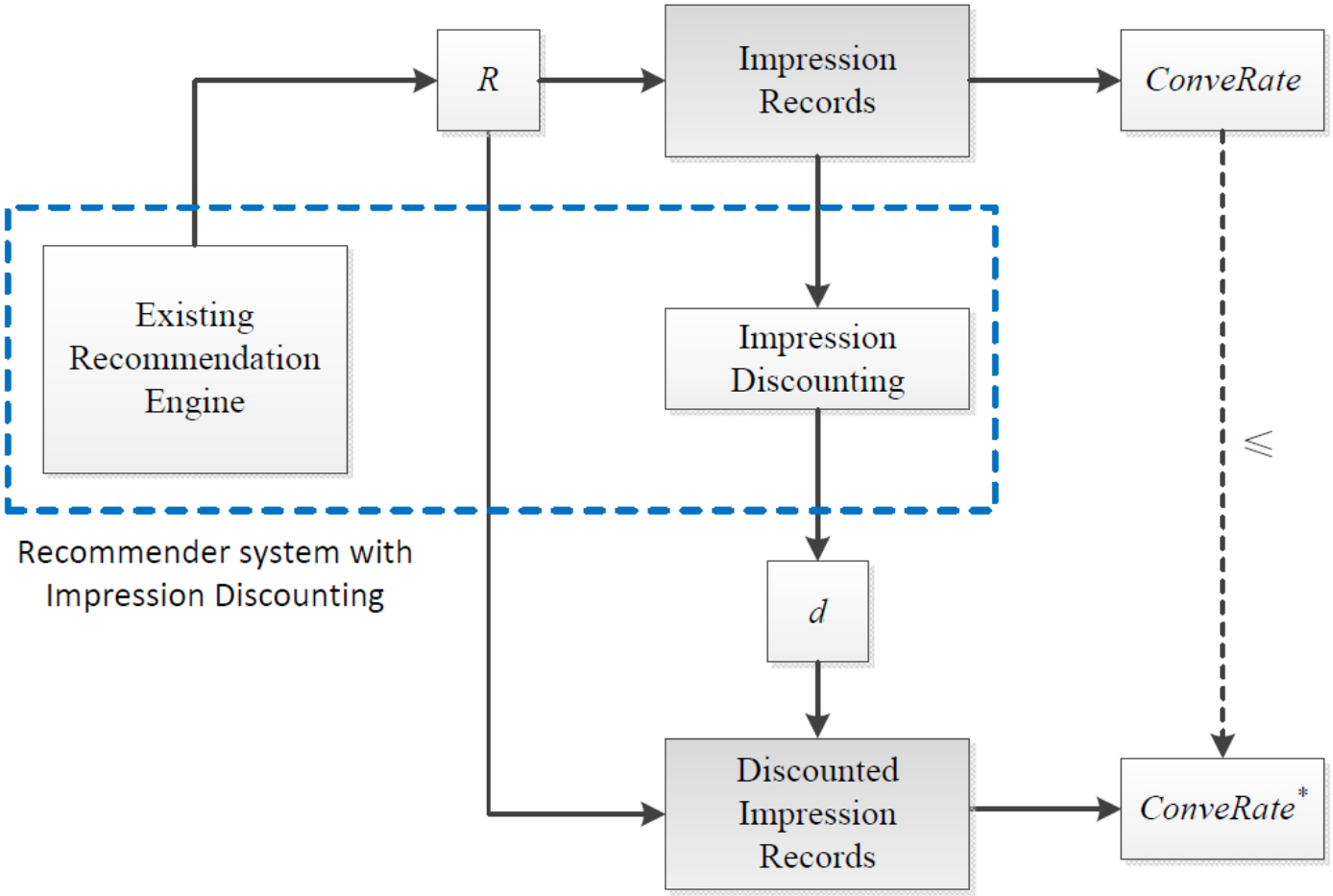
$$RMSE^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y})^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{X}_i{}^T u)^2$$

- Density weighted RMSE:

$$RMSE_v^2 = \frac{1}{n}\sum_{i=1}^{n}(v_i(y_i - \tilde{X}_i{}^T u))^2$$

v_i is assessed by the number of observations in a small neighborhood of (X_i, y_i)

Impression Discounting Framework



Quality Measure: Precision @k

- Precision at top k

$$P@k = \frac{seqSize(conversion = true)}{k}$$

- Precision improvement for different behavior sets:

Behavior Set	Precision Improvement
PYMK ($P@10$)	
LastSeen, ImpCount	13.7%
LastSeen, ImpCount Position, UserFreq	31.3%
Endorsement ($P@10$)	
LastSeen, ImpCount	1.3%
LastSeen, ImpCount Position, UserFreq	3.4%
SearchAds ($P@5, 10$)	
ImpCount	0.53%($P@10$)
ImpCount, Position	3.2%($P@10$)
ImpCount, Position	6.87%($P@5$)

More features yield higher precision

Bucket Test on People You May Know

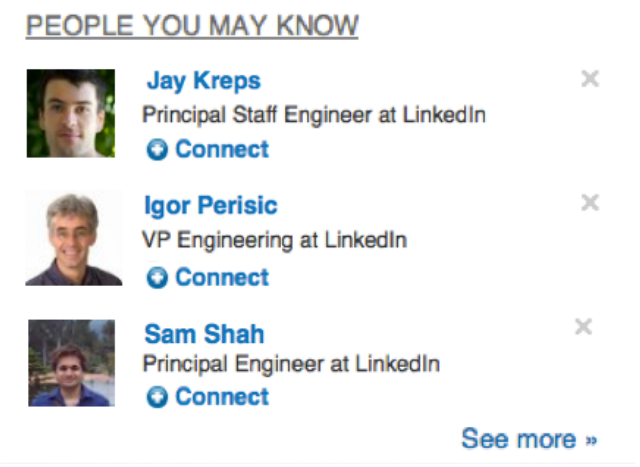
- Use behavior set (LastSeen, ImpCount)
- On LinkedIn PYMK online system

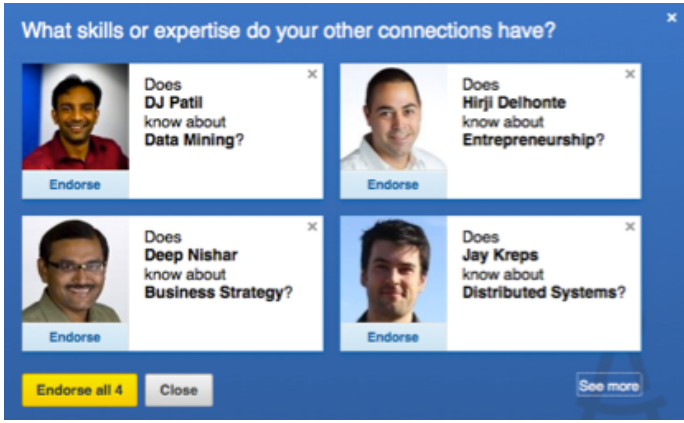
X_1: *LastSeen*; X_2: *ImpCount*	
Group A: Without Impression Discounting	
Group B:	Improvement
$\alpha_1 \cdot \alpha_2^{X_1} \cdot \alpha_3^{X_2}$	$11.97\% \pm 0.2\%$
$(\frac{\alpha_1}{X_1} + \alpha_2) \cdot \alpha_3^{X_2}$	$13.26\% \pm 0.2\%$
$(\alpha_1 \cdot X_1 + \alpha_2) \cdot \alpha_3^{X_2}$	$12.18\% \pm 0.2\%$

Table 3: The A/B test results of precision at top 10 of different impression discounting models on PYMK data set, with X =(*LastSeen*, *ImpCount*).

Conclusion

- Impression Discounting Model
  - Learnt a discounting function to capture ignored impressions
  - Linear or multiplicative aggregation model
  - Anti-noise regression model
- Offline and Online evaluation (Bucket testing)





Questions?

- Related work in the paper
- Want to know more contact: mtiwari@linkedin.com

Questions?

- Related work in the paper
- Want to know more contact: mtiwari@linkedin.com

- We're hiring!