A Dynamic Bayesian Network Click Model For Web Search Ranking

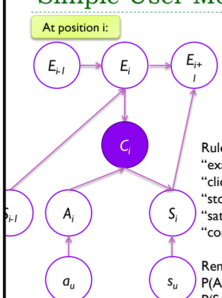
Chapelle and Zhang WWW 2009
Presented by: Michaela Götz on April 6, 2010

Simple User Model

- Idea: Understand clicking behavior of a user (how it relates to relevance of the urls) and infer relevance
- Model: User poses query, reviews results as follows:

For i=1…10:
    examine result at rank I
    determine attractiveness of abstract
    if (attractive):
        click on result
        determine satisfaction of page
        if (satisfactory):
            break

Simple User Model – Bayesian Network

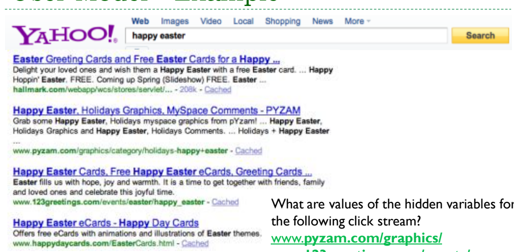


E_i	Did user examine url?
A_i	Was user attracted by url?
C_i	Did user click on url?
S_i	Was user satisfied by linked page?

Rules:
"examine in rank order" $E_i=0 \rightarrow E_{i+1}=0$
"click iff examined and attracted" $E_i=1 \wedge A_i=1 \leftrightarrow C_i=1$
"stop examining when satisfied" $S_i=1 \rightarrow E_{i+1}=0$
"satisfaction only upon click" $C_i=0 \rightarrow S_i=0$
"continue examining when not satisfied" $E_i=1, S_i=0 \rightarrow E_{i+1}=1$

Remaining Probabilities:
$P(A_i=1) = a_u$
$P(S_i=1 \mid C_i=1) = s_u$

User Model - Example



What are values of the hidden variables for the following click stream?
www.pyzam.com/graphics/
www.123greetings.com/events/
What about?
www.123greetings.com/events/
www.pyzam.com/graphics/

User Model - Inference

- Given observations (clicks)
- Infer latent variables (a_u, s_u)
  - Assume beta prior $a_u \sim \text{Beta}(\alpha_a,\beta_a)$, $s_u \sim \text{Beta}(\alpha_s,\beta_s)$
  - Update belief given clicks:
    - Let a_u^- = # of $(E_i=1), (C_i=0)$ for u in session in click stream
    - Let a_u^+ = # of $(E_i=1), (C_i=1)$ for u in session in click stream
    - Let s_u^- = # of $(C_i=1), (S_i=0)$ for u in session in click stream
    - Let s_u^+ = # of $(C_i=1), (S_i=1)$ for u in session in click stream
    - $a_u \sim \text{Beta}(\alpha_a+a_u^+, \beta_a+a_u^-)$
    - $s_u \sim \text{Beta}(\alpha_s+s_u^+, \beta_s+s_u^-)$
  - Determine values of maximum likelihood
    - $a_u = (\alpha_a+a_u^+)/(\alpha_a+\beta_a+a_u^++a_u^-)$
    - $s_u = (\alpha_s+s_u^+)/(\alpha_s+\beta_s+s_u^++s_u^-)$

User Model - Inference

- Given observations (clicks)
- Infer latent variables (a_u, s_u)
- Determine relevance r_u of url u:
  - $r_u = P(S_i=1 \mid E_i=1)$
    $= P(S_i=1, E_i=1) / P(E_i=1)$
    $= P(S_i=1, E_i=1, C_i=0) / P(E_i=1) + P(S_i=1, E_i=1, C_i=1) / P(E_i=1)$
    $= 0 + P(S_i=1, C_i=1 \mid E_i=1)$ ["satisfaction only upon click"]
    $= P(S_i=1 \mid C_i=1)\, P(C_i=1 \mid E_i=1)$ [conditional independence]
    $= a_u * s_u$

General User Model

- Idea: Understand clicking behavior of a user (how it relates to relevance of the urls) and infer relevance
- Model: User poses query, reviews results as follows:

For i=1…10:
    examine result at rank I
    determine attractiveness of abstract
    if (attractive):
        click on result
        determine satisfaction of page
        if (satisfactory):
            break
        else:
            determine frustration
            if (frustrated): break

Inference now more complicated

Remaining Probabilities +=

$P(E_{i+1}=1 \mid E_i=1, S_i=0) = \gamma$

Other User Models

- Cascade Model – special case of DBN: $\Upsilon=1$, $s_u=1$
  - Exactly one click per session
- Position Model:
  - P(clicking on url u at position p) $= \beta(p) * \alpha(u)$
- Logistic Regression
  - P(clicking on url u at position p) $= 1 / (1+\exp(\alpha'(u) * \beta'(p))$
- And more

Experiments

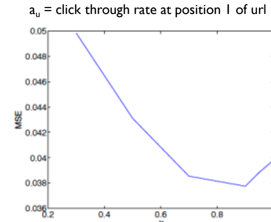
1. How accurately can the model in predict…
   - … the attractiveness of a url a_u
2. Ranking-Oriented Evaluation:
   - (a) How good is a ranking based on predicted relevance r_u?
   - (b) Can we learn some a ranking function with these predictions?
3. Model-focused Evaluation:
   - (a) Do we need the general model with Υ?
   - (b) Do we gain anything from distinguishing between a_u and s_u?

In comparison to previous work

1. Predicting the Attractiveness of a url a_u

Experimental Design
Select urls with CTR-data on various positions
Train Model based on sessions where position ≠ 1
Predict CTR at position 1
Compare to true CTR at position 1 test data

a_u = click through rate at position 1 of url u



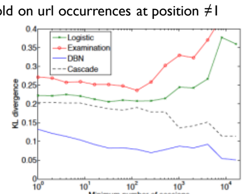
3. (a) Do we need the general model?

Simple DBN Model

1. Predicting the Attractiveness of a url a_u

Comparison to Previous Work:

- Compare accuracy of a_u
  - Examination (MLE of position model), Logistic Regression, Cascade, DBN
- Vary min threshold on url occurrences at position ≠1



- Observations:
  - Not all methods improve with more training data
  - Logistic and Examination do badly
    - Fail to consider click distribution in session

2.(a) Ranking based on Predicted Relevance

- Create ranking using DBN:
  - Train model (using clicks)
  - Sort urls according to predicted relevance r_u
- Data:
  - urls occurring in at least 10 sessions
  - queries with at least 10 such urls among results
- Report average NDCG of top-5 urls
- Compare to:
  - Logistic Regression,
  - Cascade Model,
  - "typical" ranking function as baseline

$$\mathrm{NDCG(ranking)} = \frac{\mathrm{DCG(ranking)}}{\mathrm{DCG(optimal\ ranking)}}$$

2.(a) Ranking based on Predicted Relevance

Method	NDCG
Baseline	0.795
DBN (r_u)	0.748
DBN (a_u only)	0.744
Cascade	0.73
Logistic	0.705
Baseline + DBN	0.875

- Observations:
  - DBN close to baseline.
  - DBN-feature improves baseline

Do we gain anything from distinguishing between a_u and s_u?

2.(b) Learning a Ranking Function

- How to learn a ranking function based on pairwise preference pairs $(x_i > x_j) \in \mathcal{P}$
- Goal: Learn f such that $f(x_i) > f(x_j) + \tau$

$$\operatorname{argmin}_f \sum_{(x_i > x_j) \in \mathcal{P}} \begin{cases} 0 & \text{if } f(x_i) > f(x_j) + \tau \\ (f(x_j) + \tau - f(x_i))^2 & \text{o.w.} \end{cases}$$

- How? [Zheng, Zha, Zhang, Chapelle, Chen, Sun at NIPS 07]
  - Start with initial guess f_0
  - For $k = 1, 2, \ldots$
  - Training set for each pair $(x_i > x_j) \in \mathcal{P}$
    - add two training pairs $(x_i, \max(0, f_{k-1}(x_j) + \tau - f_{k-1}(x_i)))$, $(x_j, -\max(0, f_{k-1}(x_j) + \tau - f_{k-1}(x_i)))$
  - Fit g_k using a base regressor
  - Update $f_k \leftarrow f_{k-1} + s_k \cdot g_k$ [s_k is found to minimize objective]

2.(b) Learning a Ranking Function

- Learn a ranking function using a combination of
  - DBN pairwise preference (1M pairs) $\mathcal{P}_E$
  - Editorial pairwise preference (2M pairs) $\mathcal{P}_C$

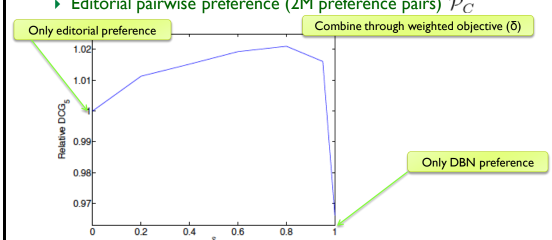
Combine through weighted objective (δ)

$$\operatorname{argmin}_f \frac{1-\delta}{|\mathcal{P}_E|} \sum_{(x_i > x_j) \in \mathcal{P}_E} \begin{cases} 0 & \text{if } f(x_i) > f(x_j) + \tau \\ (f(x_j) + \tau - f(x_i))^2 & \text{o.w.} \end{cases}$$
$$+ \frac{\delta}{|\mathcal{P}_C|} \sum_{(x_i > x_j) \in \mathcal{P}_C} \begin{cases} 0 & \text{if } f(x_i) > f(x_j) + \tau \\ (f(x_j) + \tau - f(x_i))^2 & \text{o.w.} \end{cases}$$

- Evaluate using DCG at top 5

2.(b) Learning a Ranking Function

- Learn a ranking function using a combination of
  - DBN pairwise preference (1M preference pairs) $\mathcal{P}_E$
  - Editorial pairwise preference (2M preference pairs) $\mathcal{P}_C$



Summary

- User model
  - User scans through results, keeps clicking on interesting results until a satisfactory answer is found or the user gives up
- Inference of model parameters
  - Training data: click streams
  - Infer a_u, s_u (use EM if $\Upsilon \neq 1$)
- Use model to:
  - Predict Click-Through-Rate
  - Predict relevance ($= a_u * s_u$)
    - Compute ranking
    - Compute pairwise preference and learn ranking function

Questions

- User model vs. "skip above"
  - "skip above" = hack? model = principled approach?
  - Similar accuracy in pairwise predictions
  - User model allows more general predictions
  - "skip above" easier to train (do not need url for a given query at different positions)
- User model vs. interleaved rankings
  - User model allows more general predictions
  - Interleaved rankings require active manipulation of search engine's result (but fewer clicks)
- User model vs. ΔDCG prediction [Carterette, Jones]
  - ΔDCG requires editorial relevance data for training
  - Both consider dependence of other clicks on relevance

Learn Ranking Function

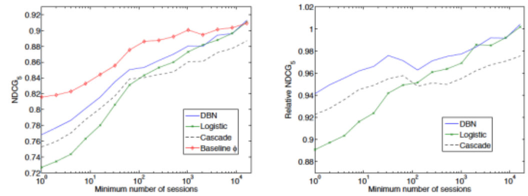
- Learn a ranking function using a combination of
  - DBN pairwise preference (1M preference pairs)
  - Editorial pairwise preference (2M preference pairs)

$$\operatorname{argmin}_f \sum_{(x_i>x_j)\in\mathcal{P}} \begin{cases}0 & \text{if } f(x_i)>f(x_j)+\tau\\ (f(x_j)+\tau-f(x_i))^2 & \text{o.w.}\end{cases}$$

$$\operatorname{argmin}_f \frac{1-\delta}{|\mathcal{P}_E|} \sum_{(x_i>x_j)\in\mathcal{P}_E} \begin{cases}0 & \text{if } f(x_i)>f(x_j)+\tau\\ (f(x_j)+\tau-f(x_i))^2 & \text{o.w.}\end{cases}$$
$$+\frac{\delta}{|\mathcal{P}_C|} \sum_{(x_i>x_j)\in\mathcal{P}_C} \begin{cases}0 & \text{if } f(x_i)>f(x_j)+\tau\\ (f(x_j)+\tau-f(x_i))^2 & \text{o.w.}\end{cases}$$

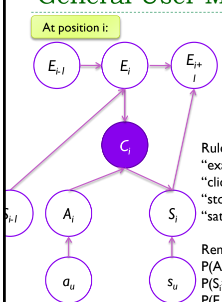
Experiment - Ranking

- Create rankings, measure NDCG



- min threshold on url occurrences increases session ➔ fewer urls to rank higher NDCG increases

General User Model – Bayesian Network



E_i	Did user examine url?
A_i	Was user attracted by url?
C_i	Did user click on url?
S_i	Was user satisfied by linked page?

Rules:
"examine in rank order" $E_i=0$ ➔ $E_{i+1}=0$
"click iff examined and attracted" $E_i=1 \wedge A_i=1$ ⟷ $C_i=1$
"stop examining when satisfied" $S_i=1$ ➔ $E_{i+1}=0$
"satisfaction only upon click" $C_i=0$ ➔ $S_i=0$

Remaining Probabilities:
$P(A_i=1) = a_u$
$P(S_i=1 \mid C_i=1) = s_u$
$P(E_{i+1}=1 \mid E_i=1, S_i=0) = \gamma$

Inference now more complicated

$P(E_{i+1}=1 \mid E_i=1, S_i=0) = \gamma$

Experiments – Pairwise Preference

- Given a pair of urls *u, u'* which one is more relevant for a query?
- Compare pairwise preferences of explicit editorial judgments and DBN model
- *Result:* **20% disagreement**
  - similar to "LastClick>SkipAbove"
- *Remark:* DBN not worse but different
  - Example: Query "bank of america"
    - Editorial judgments: www.bankofamerica.com (most relevant)
    - DBN: www.bankofamerica.com/onlinebanking/ (most relevant)