Beyond Ranking: Optimizing Whole-Page Presentation

Introduction

- Page presentation is broadly defined as the strategy to present a set of heterogeneous items on search result page (SERP), much more expressive than a ranked list.
- The framework first learns a scoring function that maps search results and their presentation on SERP to user satisfaction metric.
- Then, given search results of a new query, the framework computes a presentation that maximizes user satisfaction.

Problem Formulation

- Page Content: It is the set of search results to be displayed on a page. It is represented as concatenation of k item vectors:

$$x^T = (x_1^T, x_2^T, \ldots, x_k^T)$$

- Page Presentation: It defines the way in which page content *x* is displayed, such as position, vertical type, size and color. It is encoded as vector p.

Problem Formulation

- Search Result Page(SERP): When page content x is put on page according to presentation strategy p, a SERP is generated. It is represented as a tuple (x, p).
- User Response: It includes users actions on SERP. This information is encoded in a vector y.
- User Satisfaction: It refers to the degree of satisfaction user experiences when he interacts with the SERP. It is denoted by s.

Formulating Problem as Optimization problem

Page Presentation Optimization is to find the presentation $p \in P$ for a given page content $x \in X$, such that when the SERP (x, p) is presented to the user, her satisfaction score s is maximized.

If we assume that there exists a scoring function F, then the optimization problem can be formally written as

$$\max_{p \in P} F(x, p)$$

Learning Stage

- Direct Approach: Collect page-wise user satisfaction ratings and directly model the dependency between SERP and user satisfaction.
- Factorized Approach: First predict user response y on SERP, then find a function that measure user satisfaction from these responses.

$$y = f(x, p)$$

$$s = g(y) = g(f(x, p))$$

OPTIMIZATION FRAMEWORK

CONTENT FEATURES

- Global result set features
- Query features - unigram, bigram etc
- Corpus level features - ctr, user preferences etc.
- Search result features - relevance, ranking etc.

PRESENTATION FEATURES

- Binary Indicators
- Categorical, numerical, other.

USER SATISFACTION METRIC

We assume that user satisfaction metric $g(\mathbf{y})$ is in the form of weighted sum of components in $\mathbf{y}$:

$$g(\mathbf{y}) = \mathbf{c}^\top \mathbf{y}.$$

In experiments, we use the click-skip metric for k items [23]:

$$g(\mathbf{y}) = \sum_{i=1}^{k} y_i,$$

where $y_i = 1$ if item i is clicked, and $y_i = -1$ if item i is skipped *and* some item below is clicked. A skip often indicates wasted inspection, so we set it to be a unit of negative utility. This metric strongly prefers adjacent clicks at top positions.

USER RESPONSE MODEL 1

- QUADRATIC FEATURE MODEL
  - Considers interaction between **x** and **p**.

$$y_i = f_i(\phi) = \mathbf{w}_i^\top \phi = \mathbf{u}_i^\top \mathbf{x} + \mathbf{v}_i^\top \mathbf{p} + \mathbf{x}^\top \mathbf{Q}_i \mathbf{p}.$$

A more generalised version of the above equation is:

$$\mathbf{y} = f(\mathbf{x}, \mathbf{p}) = \mathbf{U}\mathbf{x} + \mathbf{V}\mathbf{p} + \mathbf{X}\mathbf{Q}\mathbf{t}.$$

Problem reduction to maximum bipartite matching

Denote user satisfaction metric as $g(\mathbf{y}) = \mathbf{c}^\top \mathbf{y}$. Then the scoring function $F = g \circ f$ is

$$\begin{aligned} F(\mathbf{x}, \mathbf{p}) &= g(f(\mathbf{x}, \mathbf{p})) \\ &= \mathbf{c}^\top \mathbf{U}\mathbf{x} + \mathbf{c}^\top \mathbf{V}\mathbf{p} + \mathbf{c}^\top \mathbf{X}\mathbf{Q}\mathbf{t} \\ &= \mathbf{c}^\top \mathbf{U}\mathbf{x} + \mathbf{a}^\top \mathbf{p} \end{aligned} \tag{2}$$

where $\mathbf{a} = \mathbf{V}^\top \mathbf{c} + \sum_{i=1}^{k} c_i \mathbf{Q}_i^\top \mathbf{x}$ is a known vector.

USER RESPONSE MODEL 2

- Gradient Boosted Decision Tree Model

Our feature vector is

$$\boldsymbol{\phi}^\top = (\mathbf{x}^\top, \mathbf{p}^\top),$$

and each user response y_i in $\mathbf{y}$ is predicted by a GBDT model:

$$y_i = h_i^{\mathrm{GBDT}}(\mathbf{x}, \mathbf{p}).$$

The user satisfaction metric is $g(\mathbf{y}) = \mathbf{c}^\top \mathbf{y} = \sum_{i=1}^{k} c_i y_i$.

SPECIAL CASE : Learning to rank

When page presentation is restricted to be a ranked list, and if it is assumed that users are more satisfied if more relevant results are places at top ranks, then presentation optimisation reduces to the traditional ranking problem.

Simulation of dataset

Generate Synthetic Datasets to.

- Show that the framework enables general definition of page presentation
- Show that framework can adapt to the position bias and item specific bias

Position Bias: More attention paid to certain region of the page then elsewhere.[Usually top left.

Vertical bias/Item specific bias: More attention is paid to specific tye of item [May be video or a picture than the traditional blue link.

A user has position or vertical bias.

Data Generation

Page content has k items, each item having a specific reward.

Page presentation is a random permutation of k items.

User examines the page with attention bias.

User interaction is simulated by drawing k binary values, $y=1$ if item i is clicked, 0 otherwise.

User satisfaction towards the page is the sum of the rewards of examined items.

100000 page-interactions are generated to train the model.

Page content, presentation, examined items and positions are used for learning.

Discussion

The algorithm learns to put “better content” to positions where it draws more user attention.

It captured complicated distribution of position bias for a 2D canvas

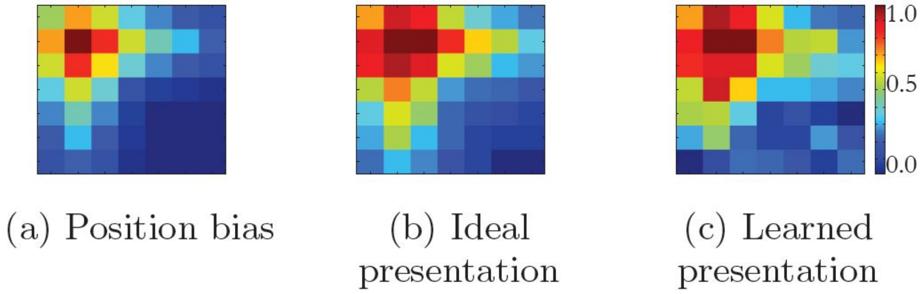

Figure 4: Top-left position bias and presentation on 2-D canvas.

Real Data collection

Small portion of search traffic was used.

News, shopping and local listings which had vertical presentation were included.

SERPs are not influenced by any ranking algorithm[??]

Users response to the page are logged.

Methods

LOGIT-RANK : Train a logistic regression model with the features and the response as the target class(click/no click)

GBDT-RANK : Uses Gradient Boost Decision Tree instead of Logistic Regression.

Each search result and the action(click/no click) is an instance for the above methods.

QUAD-PRES and GBDT-PRES: In these models the given page(vector) and the presentation(vector) is used to predict the clicks vector(A binary vector of k predicting if the ith link is clicked or not). Page content and the presentation are the features, click vector is predicted.

Evaluation

Use 50% for training and 50% for test.

Evaluation focussed on vertical results shown above the first, second and third webpage result. [Why ?]

Results

Table 1: Match rate between random exploration presentation p and predicted optimal presentation p^*. "Until Web_1 " means that p and p^* encode the same presentation above the 1st webpage result.

	Until WEB_1	Until WEB_2	Until WEB_3
LOGIT-RANK	68.68%	46.76%	30.85%
QUAD-PRES	71.63%	50.68%	33.42%

Related work

Document Ranking - A special case when presentation is a ranked list

Federated Search - Merge vertical results with general webpage results on the same page.

Search-Behaviour Modeling - Models can be created to estimate user response to a specific search content.

Conclusions - Takeaways

- Novel way to rank search results
- Novel way to learn how to present the search results and improve user satisfaction.
- When you don’t have real data, try to synthesise the data
- Discuss: Why did the author not use SVMs or Neural Networks or go down the deep learning route ?
    - Probably they have sufficient discriminative features ?
- Discuss: Can we use similar techniques for recommendations - with this mapping
    - Query = Article which the user has read
    - Query results - List of recommended articles
    - Presentation - The ranking of recommended algorithm
    - Action/Clicks - Action/User reading the article

QUESTIONS/DISCUSSIONS