Query-Dependent Image Re-Ranking Using Click Data

Vidit Jain
Yahoo Labs India

Manik Varma
Microsoft Research India
Distribution of Image Search Queries

Zipf's Law

Query Frequency


“300 workout”, “24 inch rims”, “ninja metal gear solid”
Re-Ranking Using Click Data

query → Ranker → Original ranked list → Re-ranked list
In This Talk

- We mitigate 3 problems of existing search engines
- Leverage user click data to perform re-ranking
- Use Gaussian Process regression to predict click counts for unclicked images
Limitations of Existing Rankers

- Ignore image content
Limitations of Existing Rankers

• Ignore image content
Limitations of Existing Rankers

• A single prediction model is learnt for all queries
• Score(x) = \( w^t x = \sum_j w_j x_j \) with query-independent \( w \)
• Query: “tom cruise”

\( \text{tom cruise.jpg} \)

• Query: “delhi”

\( \text{delhi.jpg} \)
## Obtaining Annotated Training Data

<table>
<thead>
<tr>
<th>Query</th>
<th>Thumbnail</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>“night train”</td>
<td><img src="image" alt="Motorcycle" /></td>
<td>1 (Not Relevant)</td>
</tr>
<tr>
<td>“fracture”</td>
<td><img src="image" alt="Movie Poster" /></td>
<td>5 (Excellent Match)</td>
</tr>
<tr>
<td>“she who must be obeyed”</td>
<td><img src="image" alt="Image" /></td>
<td>5 (Excellent Match)</td>
</tr>
</tbody>
</table>
Limitations of Existing Rankers

• Query: “night train”
• Training labels generated by human experts
Limitations of Existing Rankers

• Query: “fracture”
• Training labels generated by human experts
Leveraging User Click Data

- We use clicks as surrogate training data
  - We avoid problems due to “expert” labels
  - We train a query-dependent re-ranker
  - We can compute visual features from the clicked images.

- Key assumption: user clicks are highly correlated with relevance
  - Documents: 2 line snippet
  - Videos: 30 second clip
  - Images: Thumbnails
Clicks and Relevance

• Query: “fracture”

• Query: “child drinking water”

• Query: “Spring Break 2007”
Evidence for clicks-relevance relationship

![Bar chart showing the percentage of clicked items that are relevant in document search and image search.]

- **Document search** (short snippet)
- **Image search** (thumbnail)

[Agichtein06]
Naïve solution – ClickBoosting

- Disadvantages
  - Self reinforcement loop
  - Distracter images promoted to the top
  - Relevant, un-clicked images will never get shown
Overview of our solution

query → Ranker → Original ranked list

Re-ranked list

Pseudo-click estimation

\( f^*_{\text{text}} \)

\( f^*_{\text{visual}} \)

clicked
GP Regression on Click Data

- Given a query, obtain the baseline image search results and the associated click data
- Train a regression model on the click data to predict the number of clicks for a novel image
- Re-rank the top 1000 images according to a linear combination of the predicted number of clicks and the original ranking score
Re-scoring Function

- Re-scoring function for image $\mathbf{x}$
  
  $$
  s_R(\mathbf{x}) = a_1 s_O(\mathbf{x}) + a_2 y_{Text}(\mathbf{x}) + a_3 y_{Visual}(\mathbf{x})
  $$

- where
  
  - $s_R$ and $s_O$ are the re-ranked and original scores
  - $y_{Text}$ and $y_{Visual}$ are the predicted number of clicks estimated using text and visual features
  - $a_1$, $a_2$ and $a_3$ are global weighting constants
Measuring Search Performance – nDCG

- Given a ranked list of relevance judgments $\mathbf{R}$

- Cumulative Gain at $P$
  \[
  CG_P(\mathbf{R}) = \sum_{i=1..P} 2^{R_i} - 1
  \]

- Discounted Cumulative Gain
  \[
  DCG_P(\mathbf{R}) = \sum_{i=1..P} (2^{R_i} - 1) / \log_2 (i+1)
  \]

- Normalized Discounted Cumulative Gain
  \[
  nDCG_P(\mathbf{R}) = DCG_P(\mathbf{R}) / DCG_P(I)
  \]
  where $I$ is the judgment for the ideal ranked list
Measuring Search Performance – nDCG

• Query: “Andrew Zisserman”

\[
\begin{align*}
R_1 &= 0 & R_2 &= 5 & R_3 &= 0 & R_4 &= 0 & R_5 &= 5 \\
\text{CG} &= 62 & \text{DCG} &= 46 \\
\text{CG} &= 62 & \text{DCG} &= 73 \\
R_1 &= 5 & R_2 &= 5 & R_3 &= 0 & R_4 &= 0 & R_5 &= 0
\end{align*}
\]

• nDCG@5 = 46/73 = 0.63
Click Estimation

• Features
  • Query independent text features (Pagerank)
  • Query dependent text features (filename match)
  • Visual features based on shape, colour and texture (HOG, SIFT, LBP, colour histograms, etc)

• We have very few training images (approximately 20 – 100) and more than 3000 features

• It is therefore essential to perform dimensionality reduction to avoid over fitting
Click Estimation - Dimensionality Reduction

• We only have “positive” training data so discriminative methods did not work well (generating negative training data is non-trivial)

• Simple methods did work well

<table>
<thead>
<tr>
<th>Approach</th>
<th>Mean nDCG at 20</th>
<th>Relative Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average click rank</td>
<td>0.6266</td>
<td>− 8.6 %</td>
</tr>
<tr>
<td>Correlation with score</td>
<td>0.7209</td>
<td>+5.2 %</td>
</tr>
<tr>
<td>Correlation with clicks</td>
<td>0.7409</td>
<td>+8.1 %</td>
</tr>
<tr>
<td>PCA</td>
<td>0.7692</td>
<td>+12.2 %</td>
</tr>
</tbody>
</table>
Click Estimation – Regression

Query: “night train”

(#clicks) 50 20 5 2 35
Visual features are not enough

Query: “night train”

Need both visual and text features
Re-ranking function

\[ s_R(x) = \alpha_1 s_O(x) + \alpha_2 y_{text}(x) + \alpha_3 y_{visual}(x) \]
Click Estimation - Regression

• Gaussian Process Regression

\[ y(x) = k(x, x_{\text{Train}}) \left[ k(x_{\text{Train}}, x_{\text{Train}}) + \sigma^2 I \right]^{-1} y_{\text{Train}} \]

\[ = d^t(x, x_{\text{Train}}) y_{\text{Train}} \]

\[ = w^t \phi(x) \]

• where

• \( y \) is the predicted number of clicks and \( y_{\text{Train}} \) the number of clicks for the set of training images
• \( x \) are the features extracted from a novel image
• \( x_{\text{Train}} \) are the training set features
• \( \sigma \) is a noise parameter
• \( k \) is a Gaussian kernel function
## Click Estimation - Regression

<table>
<thead>
<tr>
<th>Approach</th>
<th>Mean nDCG at 20</th>
<th>Relative Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Regression</td>
<td>0.6871</td>
<td>− 0.2%</td>
</tr>
<tr>
<td>Support Vector Regression</td>
<td>0.6997</td>
<td>+2.1%</td>
</tr>
<tr>
<td>Nearest Neighbour</td>
<td>0.7428</td>
<td>+8.3%</td>
</tr>
<tr>
<td>GP Regression</td>
<td>0.7692</td>
<td>+12.2%</td>
</tr>
</tbody>
</table>
Re-scoring Function

- Re-scoring function for image $x$
  $$s_R(x) = a_1 s_O(x) + a_2 y_{Text}(x) + a_3 y_{Visual}(x)$$

<table>
<thead>
<tr>
<th>Approach</th>
<th>Mean nDCG at 20</th>
<th>Relative Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline ($a_2 = a_3 = 0$)</td>
<td>0.6854</td>
<td>–</td>
</tr>
<tr>
<td>Baseline + $y_{Text}$ ($a_3 = 0$)</td>
<td>0.7077</td>
<td>+3.3 %</td>
</tr>
<tr>
<td>Baseline + $y_{Visual}$ ($a_2 = 0$)</td>
<td>0.6136</td>
<td>−10.5%</td>
</tr>
<tr>
<td>Baseline + $y_{Text}$ + $y_{Visual}$</td>
<td>0.7692</td>
<td>+12.2%</td>
</tr>
</tbody>
</table>
Evaluation on 193 Queries
Evaluation on 193 Queries

Avg gain in mean(nDCG@20) (%) vs. # of clicked images per query
Bing Results – “fracture”
GP Regression Results – “fracture”
GP Regression Results – “pacific ocean”
GP Regression Results – “camel caravan”
GP Regression Results – “24 inch rims”
Query: “turkey”

Multiple interpretations are retained if manifested by clicks
Bing Results – “Stargate (1994)"
GP Regression Results – “Stargate (1994)”
Acknowledgements

• Deepak Agarwal
• James Allan
• Mark Bolin
• Michael Bendersky,
• Qifa Ke