Information Retrieval as Structured Prediction

Yisong Yue
Cornell University

Joint work with: Thorsten Joachims, Filip Radlinski, and Thomas Finley

Machine Learning for IR

- Machine learning often used (learning to rank)
- First generate features

\[
x_{e,d} = \phi(q,d) = \left[ \sum \text{appears in title} \right] - \left[ \sum \text{appears in first paragraph} \right] + \left[ \sum \text{appears in anchor text linking to } d \right] - \frac{\cos(q,d)}{\text{pagerank}(d)}
\]

Outline

- Optimizing ranking measures
  - “Learning to Rank”
  - Structured loss function
  - Mean average precision
- Diversified retrieval
  - Coverage problem
  - Structured prediction problem

Mean Average Precision

- Consider rank position of each relevance doc
  - \(K_1, K_2, \ldots, K_R\)
- Compute Precision@K for each \(K_1, K_2, \ldots, K_R\)
- Average precision = average of P@K
- Ex: \(\text{has AvgPrec of } \frac{1}{3} (1 + \frac{2}{3} + \frac{3}{5}) = 0.76\)
- MAP is Average Precision across multiple queries/rankings

MAP vs Accuracy

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>H2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ranking</th>
<th>MAP</th>
<th>Best Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>0.56</td>
<td>0.64</td>
</tr>
<tr>
<td>H2</td>
<td>0.51</td>
<td>0.73</td>
</tr>
</tbody>
</table>
Optimizing Pairwise Agreements

- 2 pairwise disagreements

Pairwise Preferences SVM

\[ \arg \min_{w, \xi} \frac{1}{2} w^T w + \frac{C}{N} \sum_{i,j} \xi_{i,j} \]

Such that:

\[ w^T x_i - w^T x_j \geq 1 - \xi_{i,j}, \quad \forall i, j : y_i > y_j \]

\[ \xi_{i,j} \geq 0, \quad \forall i, j \]

Large Margin Ordinal Regression [Herbrich et al., 1999]
Can be reduced to \( O(n \log n) \) time [Joachims, 2005]
Pairs can be reweighted to more closely model IR goals [Cao et al., 2006]

MAP vs ROC-area

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>8</td>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>H2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ranking</th>
<th>MAP</th>
<th>ROC-area</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td>0.59</td>
<td>0.47</td>
</tr>
<tr>
<td>H2</td>
<td>0.51</td>
<td>0.53</td>
</tr>
</tbody>
</table>

Linear Discriminant for Ranking

- Let \( \mathbf{x} = (x_1, \ldots, x_n) \) denote candidate documents (features)
- Let \( y_{jk} = \{+1, -1\} \) encode pairwise rank orders
- Feature map is linear combination of documents.
  \[ \Psi(y, \mathbf{x}) = \sum_{j \neq k} y_{jk} \cdot (x_j - x_k) \]
- Prediction made by sorting on document scores \( w^T x_i \)
  \[ \hat{y} = \arg \max_y w^T \Psi(y, \mathbf{x}) \]

Structural SVM

- Let \( \mathbf{x} \) denote a structured input (candidate documents)
- Let \( \mathbf{y} \) denote a structured output (ranking)
- Standard objective function:
  \[ \frac{1}{2} w^T w + \frac{C}{N} \sum \xi_i \]
- Constraints are defined for each incorrect labeling \( \mathbf{y}' \) over the set of documents \( \mathbf{x} \).
  \[ \forall \mathbf{y}' \neq \mathbf{y}^{(i)} : w^T \Psi(y^{(i)}, x^{(i)}) \geq w^T \Psi(y', x^{(i)}) + \Delta_i (y') - \xi_i \]

[Yue, Finley, Radlinski, Joachims; SIGIR 2007]

Structural SVM for MAP

- Minimize
  \[ \frac{1}{2} w^T w + \frac{C}{N} \sum \xi_i, \]
  subject to \( \forall \mathbf{y}' \neq \mathbf{y}^{(i)} : w^T \Psi(y^{(i)}, x^{(i)}) \geq w^T \Psi(y', x^{(i)}) + \Delta_i (y') - \xi_i \)
  where \( \Psi(y^{(i)}, \mathbf{x}) = \sum_{j \neq k} y_{jk} \cdot (x_j - x_k) \) \( (y_{jk} = \{-1, +1\}) \)
  and \( \Delta_i (y') = 1 - \text{Avgprec}(y') \)
- Sum of slacks: \( \sum \xi_i \) is smooth upper bound on MAP loss.

[Yue, Finley, Radlinski, Joachims; SIGIR 2007]
Too Many Constraints!

- For Average Precision, the true labeling is a ranking where the relevant documents are all ranked in the front, e.g., $y = [1, 1, 1, 0, 0]$
- An incorrect labeling would be any other ranking, e.g., $y = [1, 1, 0, 1, 0]$
- This ranking has Average Precision of about 0.8 with $\Delta(y') = 0.2$
- Intractable number of rankings, thus an intractable number of constraints!

Cutting Plane Training

Original SVM Problem

- Exponential constraints
- Most are dominated by a small set of “important” constraints

Structural SVM Approach

- Repeatedly finds the next most violated constraint...
- ...until set of constraints is a good approximation.

Finding Most Violated Constraint

$$\arg \max_y \Delta(y') + \sum_{j \in [M]} y'_{j} (w^T x_j - w^T x_k)$$

Observations

- MAP is invariant on the order of documents within a relevance class
  - Swapping two relevant or non-relevant documents does not change MAP.
- Joint SVM score is optimized by sorting by document score, $w^T x_j$
- Reduces to finding an interleaving between two sorted lists of documents

[Yue et al., SIGIR 2007]

Finding Most Violated Constraint

$$\arg \max_y \Delta(y') + \sum_{j \in [M]} y'_{j} (w^T x_j - w^T x_k)$$

- Start with perfect ranking
- Consider swapping adjacent relevant/non-relevant documents
- Find the best feasible ranking of the non-relevant document
- Repeat for next non-relevant document
- Never want to swap past previous non-relevant document
- Repeat until all non-relevant documents have been considered

[Yue et al., SIGIR 2007]

Proof (Sketch)

$$H(y) = \Delta(y) + \sum_{j \in [M]} y_{j} (w^T x_j - w^T x_k)$$

- Assume relevant and non-relevant docs are sorted
- Define $\delta_{j}(i, j)$ as the change in $H$ when:
  - The highest ranked relevant document after $x_k$ changes from $x_i$ to $x_j$
  - $i$ and $j$ index relevant documents ($i < j$)
  - $k$ indexes non-relevant document
- Need to show $\delta_{i,j}(i, i+1) \leq \delta_{j}(i, i+1)$

[Yue et al., SIGIR 2007]

Experiments

- Used TREC 9 & 10 Web Track corpus.
- Features of document/query pairs computed from outputs of existing retrieval functions.
  (Indri Retrieval Functions & TREC Submissions)
- Goal is to learn a recombination of outputs which improves mean average precision.
Finding Most Violated Constraint

- Required for structural SVM training
  - Depends on structure of loss function
  - Depends on structure of the feature map
  - Efficient algorithms exist despite intractable number of constraints.
- More than one approach
  - [Yue et al., 2007]
  - [Chapelle et al., 2007]

Story so Far

- Optimizing ranking measures
  - “Learning to Rank”
  - Structured loss function
  - Mean average precision
- Diversified retrieval
  - Coverage problem
  - Structured prediction problem

Submodular Functions

- For set S, \( F : 2^S \rightarrow R \) is submodular if
  \[ F(A \cup B) + F(A \cap B) \leq F(A) + F(B) \]
- Budgeted Maximum Coverage Problem
  - Documents cover some amount of information
  - Documents overlap in information covered
  - Documents have uniform “cost”
  - Select K docs that collectively maximize information
  - Greedy has \((1-1/e)\) approximation bound
Diversity as Coverage Problem

- Given a good representation of information
  - Retrieve documents to maximize coverage
- Learning approach to automatically learn coverage representation
  - Used to make predictions on new test examples
  - Structural SVMs

How to Represent Information?

- All the words
  - (title words, anchor text, etc)
- Cluster memberships
  - (topic models / dim reduction)
- Taxonomy memberships (ODP)

Weighted Word Coverage

- More distinct words = more information
- Weight word importance
- **Goal**: select K documents which collectively cover as many distinct (weighted) words as possible
  - Greedy algorithm
  - \((1-1/e)\) – approximation bound (submodular)
  - Need good weighting function (learning problem).

[Yue & Joachims, ICML 2008]

Example

<table>
<thead>
<tr>
<th>Document Word Counts</th>
<th>Document Word Count</th>
<th>Word Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>V1</td>
<td>V2</td>
</tr>
<tr>
<td>D1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D2</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>D3</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Marginal Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>D1</td>
</tr>
<tr>
<td>D2</td>
</tr>
<tr>
<td>D3</td>
</tr>
<tr>
<td>Best</td>
</tr>
</tbody>
</table>

How to Weight Words?

- Not all words created equal
  - “the”
- Conditional on the query
  - “computer” is normally fairly informative…
  - …but not for the query “ACM”
- Weighting function based on the candidate set
  - (for a query)

Prior Work

- **Essential Pages** [Swaminathan et al., 2008]
  - Uses fixed function of word benefit
  - Depends on word frequency in candidate set
  - Local version of TF-IDF
  - Frequent words low weight (not important for diversity)
  - Rare words low weight (not representative)
### Word Frequency Features

- \( x = (x_1, x_2, ..., x_d) \) - candidate documents
- \( v \) – an individual word

\[
\phi(v, x) = \begin{cases} 
[v \text{ appears in } >10\% \text{ of } x] \\
[v \text{ appears in } >20\% \text{ of titles in } x] \\
[v \text{ appears in } >15\% \text{ of anchors in } x] \\
[v \text{ appears in } >25\% \text{ of meta in } x]
\end{cases}
\]

- We will use thousands of such features
- Benefit of covering word \( v \) is \( w^T \phi(v, x) \)

[Yue & Joachims, ICML 2008]

### Structured Prediction for Maximizing Coverage

\[
\rho^T \Psi(x, y) = \sum_{v \in y} \rho^T \phi(v, x)
\]

- Does NOT reward redundancy
  - Benefit of each word only counted once
- Greedy has \( (1-1/e) \)-approximation bound
- More sophisticated structure in experiments
- Train \( w \) using structural SVM approach
  - Optimizes empirical risk & generalization bound

[Yue & Joachims, ICML 2008]

### More Sophisticated Discriminant

\[
\Psi(y, x) = \sum_{i=1}^{N} \sum_{v \in y_i} \phi(v, x)
\]

- Separate \( \phi \) for each importance level \( i \)
- Joint feature map \( \Psi \) is vector composition of all \( \phi \)

\[
\hat{y} = \arg \max_y w^T \Psi(x, y)
\]

- Greedy has \( (1-1/e) \)-approximation bound.
- Still uses linear feature space.

[Y, Joachims; ICML 2008]

### Structural Support Vector Machine

- Let \( x \) denote a structured input (candidate documents)
- Let \( y \) denote a structured output (subset of size \( K \))

\[
\frac{1}{2} w^T + \frac{C}{N} \sum \xi
\]

- Constraints are defined for each incorrect labeling \( y' \) over the set of documents \( x \).

\[
\forall y' \neq y^{(i)}: w^T \Psi(x^{(i)}, y'^{(i)}) \geq w^T \Psi(x^{(i)}, y^{(i)}) + \Delta(y') - \xi
\]

[Tsochantaridis et al., 2005]
Weighted Subtopic Loss

- Example:
  - $x_1$ covers $t_1$
  - $x_2$ covers $t_1, t_2, t_3$
  - $x_3$ covers $t_1, t_3$

- Motivation
  - Higher penalty for not covering popular subtopics
  - Mitigates label noise in the tail

Yue & Joachims, ICML 2008

Finding Most Violated Constraint

$\hat{y} = \arg \max_y w^T \Psi(x, y') + \Delta(y')$

- Encode each subtopic as an additional "word" to be covered.
  $\Psi(x, y') = \sum_{t_i \in \Phi(y)} \phi_i(v, x) - \sum_{t_i \notin \Phi(y)} \Delta_i$

- Use greedy prediction to find approximate most violated constraint.

Approximate Constraint Generation

- Theoretical guarantees still hold.
  - Constant factor approximation to finding optimal cutting plane
  - $(1-1/e)$ approximation for solving coverage problems
- Performs well in practice.

Diversity Training Data

- TREC 6-8 Interactive Track
  - Queries with explicitly labeled subtopics
  - E.g., “Use of robots in the world today”
    - Nanorobots
    - Space mission robots
    - Underwater robots
  - Manual partitioning of the total information regarding a query

Missing Subtopic Error Rate

- Trained & tested via cross validation
- Retrieving 5 documents

Learning Coverage Representations

- Training set with gold standard labels
- Learn automatic representation
  - Does not require gold standard labels
  - Maximize coverage on new problem instances
- “Inverse” of prediction problem
  - Given gold standard, can predict a good covering
  - Learn automatic representation that agrees with gold standard solution