Feature Selection for Ranking

Gytis Tumas, Gregory Osunde, Linas Baltrunas
Outline

1. Big picture
2. Feature selection methods
   1. Filter
   2. Wrapper
3. New Feature selection method for ranking
4. Experiments
5. Conclusion
FEATURE SELECTION FOR RANKING

- Document length
- Term Frequency
- Inverse Document Frequency
  - PageRank
  - BM25
  - Relevance Propagation

MAXIMUM IMPORTANCE
MINIMUM SIMILARITY

OPTIMIZATION PROBLEM

SIMILARITY
- RANKING DISTANCE METHODS
  - Kendall’s τ

IMPORTANCE
- EVALUATION MEASURES
  - MAP
  - NDCG

GAS ALGORITHM

RANKING MODELS
- Ranking SVM
- RankNet

1. imc genova . Italy imc
   Utile features in categoria [sardigna] Ripensare 1.... AGGIORNA QUESTA FEATURE commenta ecc:
   italy.indymedia.org/features/genova/ - 98k - Copia cache

2. Google Help : Search Features - [ Traduci ]
   In addition to providing easy access to billions of useful features to help you find exactly what you're looking for:
   www.google.com/help/features.html - 68k - Copia cache

3. The Features - CONTRAST EP - [ Traduci ]
   The official site for The Features. Your new favorite website:
   www.thefeatures.com/ - 7k - Cópia cache - Flash

4. Opera browser - [ Traduci questa pagina ]
   Opera is the Web pioneer that delivered tabbed browsing to everyone. Other great tools for quick navigation:
   www.opera.com/ - 6k - Copia cache
Feature selection has its background in:

- **Information retrieval**

  Ranking is a crucial part of information retrieval. It is able to compute sorted score when given documents as objects. The score may represent (degrees of relevance, preference, and importance, etc.)
Goal

- Develop different feature selection for ranking
- Analyzes Ranking accuracy
  - Performance Measure
  - Loss function
Features

- Page rank
- Inverse document frequency
- Frequency
- Frequency document length etc.

**Importance of feature selection**

- **Machine Learning**
  It improve the efficiency in many machine learning.

- **Over fitting problem**
  Over fitting is the problem of training the machine so much that when the actual data is place it behave well to an extent and start to fail.

- **Improve Efficiency of training**
  If the performance is bad we drop it and take another feature and test. The good feature will be used to compare the similarity with the next good feature.
Feature selection methods for classification

- **Filter Method**
  - pre-process computation of score for each feature and then select feature according to the score

- **Wrapper Method**
  - The wrapper utilize learning as a black box to score subset features

- **Embedded Method**
  - Feature selection is perform within the process of training the algorithm
Feature selection methods for classification (2)

- Problems of using the same method for Ranking
  
  - Significant gap between classification and ranking
  - Categories are flat in classification
  - Categories are order in ranking

So existing feature selection methods are not suitable

- Evaluation measures
  
  - Ranking (MAP and NDCG)

Precision is more important than recall in ranking

Precision and recall are important in classification
Feature Selection Method for ranking

• Goal – select $t \ (1 \leq t \leq m)$ features from the entire feature set $\{v_1, \ldots, v_m\}$.

• Take into account:
  – Importance of the feature
  – Similarity between features
Importance of feature

Q: How to find how much each feature is important?

Answer:
1) Consider each feature as a ranking model!
2) Evaluate them (using MAP, NDCG or a loss function)
3) The result will be the importance score for the feature
Similarity between features

- It is important to remove the redundancy between features in cases where we can use only a few features.
- How to calculate similarity between 2 features?
  - Each feature - a ranking model
  - \textbf{Similarity between features} = similarity between the ranking results that they produce
  - Example:
    - \( v_i \) – PR (PageRank)
    - \( v_j \) – TF (Term Frequency)
Similarity between features

\[ \tau_q(v_i, v_j) = \frac{\# \{(d_s, d_t) \in D_q \mid d_s \prec_{v_i} d_t \text{ and } d_s \prec_{v_j} d_t\}}{\# \{(d_s, d_t) \in D_q\} } \]

\[ \tau_q(v_i, v_j) = \frac{1 + 1 + 0 + 1 + 1 + 0 + 1 + 1 + 0 + 0}{10} = 0,6 \]
Optimization formulation

\[ \begin{align*}
\max & \sum_i w_i x_i \\
\min & \sum_i \sum_{j \neq i} e_{i,j} x_i x_j \\
\text{s.t.} & \quad x_i \in \{0,1\} \quad i = 1, \ldots, m \\
\sum_i x_i &= t
\end{align*} \]

Multi-objective programming needed!

\[ \begin{align*}
\max & \sum_i w_i x_i - c \sum_i \sum_{j \neq i} e_{i,j} x_i x_j \\
\text{s.t.} & \quad x_i \in \{0,1\} \quad i = 1, \ldots, m \\
\sum_i x_i &= t
\end{align*} \]

Convert into single-objective programming!
Solution to optimization problem

- There’s no efficient solution to such kind of problem!
- An option: perform exhaustive search!
  - The time complexity would be $O(C^t_m)$ – too high to make it applicable in real applications.
- We need a more practical solution!
Algorithm GAS (Greedy search algorithm of feature selection)

Undirected graph $G_0$

Set $S$ containing all selected features
Algorithm GAS (Greedy search algorithm of feature selection)
Algorithm GAS (Greedy search algorithm of feature selection)

Undirected graph $G_0$

Set $S$ containing all selected features
Algorithm GAS (Greedy search algorithm of feature selection)

Undirected graph $G_0$

Set $S$ containing all selected features
Algorithm GAS (Greedy search algorithm of feature selection)

Undirected graph $G_0$

Set $S$ containing all selected features
Algorithm GAS (Greedy search algorithm of feature selection)

Undirected graph $G_0$

Set $S$ containing all selected features
Algorithm GAS (Greedy search algorithm of feature selection)

- Initialize a set $S$ to contain nothing.
- For $i = 1 \ldots t$
  - Select the node with the largest weight $v_{k_i}$
  - Punish all other nodes (update their weights):
    \[ w_j \leftarrow w_j - e_{k_i,j} \times 2c, \quad j \neq k_i \]
  - Add $v_{k_i}$ to the set $S$ and remove it from graph together with all edges connected to it
    \[ S_{i+1} = S_i \cup \{ v_{k_i} \}, \quad G_{i+1} = G_i \setminus \{ v_{k_i} \} \]
- Output $S$
Experiment settings

Experiments with two datasets

.feature selection and training the model

Validate the efficiency of the model

Testing to find out how one model outperformed another
Use in web track of TREC 2004 to monitor the web details

Total of 1,053,110 documents and 75 queries

BM model was used to retrieve the top 1000 document for each queries

Extracted 44 features for each document
Experiment settings (3)

OHSUMED Data

- Used extensively in many data in information retrieval systems
- It is a bibliographical document collection
- Total of 16,140 documents

Query were made to analyzes
- Definitely relevant
- Possibly relevant
- Not relevant

Result
- 26 features taken from each document
### Measures use to evaluate ranking methods for information retrieval

<table>
<thead>
<tr>
<th>Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean Average Precision (MAP)</td>
</tr>
<tr>
<td>Normalized Discount Cumulative Gain (NDCG)</td>
</tr>
</tbody>
</table>
Evaluation Measures (2)

Mean Average Precision (MAP)

MAP is a Boolean measure on ranking result

- Assumption (Two documents)
  - Positive (relevant)
  - Negative (irrelevant)

- Calculate Precision at top n and measures the result on the top n query

\[ P(n) = \frac{# \text{ of positive instances within top } n}{n} \]

Sample example:

- \( P(1) = \frac{1}{1} = 1 \)
- \( P(2) = \frac{2}{2} = 1 \)
- \( P(3) = \frac{2}{3} = 0.666666667 \)
- \( P(4) = \frac{3}{4} = 0.75 \)
- \( P(5) = \frac{3}{5} = 0.6 \)
- \( P(6) = \frac{4}{6} = 0.666666667 \)
Evaluation Measures (3)

Precision at top n

- Number of positive instances
- $\text{pos}(n)$ – binary function
- $n =$ number of documents

**Sample example**

- $P(1) = \frac{1}{1} = 1$
- $P(2) = \frac{2}{2} = 1$
- $P(3) = \frac{2}{3} = 0.666666667$
- $P(4) = \frac{3}{4} = 0.75$
- $P(5) = \frac{3}{5} = 0.6$
- $P(6) = \frac{4}{6} = 0.666666667$

$$AP = \frac{\sum_{n=1}^{N} P(n) \times \text{pos}(n)}{\text{number of positive instances}}$$

**MAP Sample example**

$$AP = \frac{(\sum_{1}^{6}[(1\times 1) + (1\times 1) + (0.66\times 0) + (0.75\times 1) + (0.6\times 0) + (0.66\times 1)])}{4} = 0.85$$
Evaluation Measures (4)

- Measure ranking accuracies
- Evaluate feature selection
- Analyzes multiple of relevance

Normalized Discount Cumulative Gain (NDCG)

\[
N(n) = Z_n \sum_{j=1}^{n} \frac{2^{R(j)} - 1}{\log(1+j)}
\]

- This method is used to compute the importance of scores of features
- \( n = \text{denote position} \)
- \( R(j) = \text{Score of rank } j \)
- \( Z_n = \text{guarantee that perfect ranking is NDCG at position } n = 1 \)

NDCG is average over all queries
Cumulated Gain (CG)

\[ CG[i] = \begin{cases} 
G[1] & i = 1 \\
CG[i-1] + G[i] & \text{otherwise}
\end{cases} \]

- CG = \{1, 2, 2, 3, 3, 4\}

\[ CG = \{1, 2, 2, 3, 3, 4\} \]
Discount Cumulated Gain (DCG)

\[
DCG[i] = \begin{cases} 
CG[1] , & i < b \\
DCG[i - 1] + G[i] / \log_b(i) , & i \geq b 
\end{cases}
\]

- Lets say b=2
- DCG={1,2,2,2.5,2.5,(2.5+1/log_2(6))}

\[
N(n) = \sum_{j=1}^{n} \frac{2^{R(j)} - 1}{\log(1 + j)}, R(j) \in \{0,1\}
\]
Normalized Discount Cumulated Gain (NDCG)

- \( \text{DCG} = \{1, 2, 2.5, 2.5, (2.5 + 1 / \log_2(6))\} \)
- \( \text{DCG}_{\text{Ideal}} = \{1, 2, (2 + 1 / \log_2(3)), (2 + 1 / \log_2(3) + 0.5)\} \)
- \( \text{NDCG}(i) = \frac{\text{DCG}(i)}{\text{DCG}_{\text{Ideal}}(i)} \)

\[
N(n) = Z_n \sum_{j=1}^{n} \frac{2^{R(j)} - 1}{\log(1 + j)}, R(j) \in \{0, 1\}
\]
Ranking Model

**Ranking SVM**

- Prove to be an effective algorithms for Ranking from past analysis
- Ranking SVM utilizes instances pairs and their preferences in training (against traditional SVM that work on instance)

\[
\min \frac{1}{2} w^T w + C \sum_{i,j,q} \varepsilon_{q,i,j} \\
\text{s.t. } \forall (d_i, d_j) \in r_q^*: \omega \phi(q, d_i) \geq \omega \phi(q, d_j) + 1 - \varepsilon_{q,i,j}
\]
**Ranking Model (2)**

**Ranking Net**

- Also uses instance pair in training
- Uses a neural network as the ranking function
- Relative entropy as a loss function
- Effective in large datasets
- The Loss for an instance pair is computed as

\[
L_{q,i,j} \equiv L(\phi_{q,i,j}) = -\bar{P}_{ij} \phi_{q,i,j} + \log(1 + e^{\phi_{q,i,j}})
\]

Evaluate the result with MAP and NDCG
### Comparing Algorithm

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GAS-E</td>
<td>In GAS-E Evaluation measures (e.g. NDCG, MAP) to calculate the importance score of each feature. But may not be good for training the model</td>
</tr>
<tr>
<td>GAS-L</td>
<td>In GAS-L The empirical loss of ranking model was used to measure the importance of feature.</td>
</tr>
<tr>
<td></td>
<td>- Ranking SVM uses Pair wise ranking error</td>
</tr>
<tr>
<td></td>
<td>- RankNet perform analysis using cross entropy loss</td>
</tr>
</tbody>
</table>

**IG and CHI as baselines**

- IG measure the reduction in uncertainty (entropy) in classification prediction
- CHI measure degree of independence b/w feature and categories.
Experimental Results (with .gov data, MAP)

(a) MAP of Ranking SVM
Experimental Results (with .gov data, NDCG)

(b) NDCG@10 of Ranking SVM
Experimental Results (with OHSUMED data, MAP)

(a) MAP of Ranking SVM
Experimental Results (with OHSUMED data, NDCG)
Observations of the results

- Feature selection can improve the ranking performance more significantly for the .gov data than for the OHSUMED data.
  - .gov data: up to 10% improvement over WAF
  - OHSUMED: 1-2% improvement over WAF

- Proposed algorithm outperforms IG and CHI more significantly for the OHSUMED dataset than .gov data set.

- Why is there such a difference between the results of the different datasets?
The reason

MAP of individual features in 2 datasets:

(a) The .gov dataset

(b) The OHSUMED dataset
Conclusions

1) We discussed the differences between classification and ranking
2) Novel method proposed – optimization problem
3) We showed the evaluation of the method using 2 public datasets
Future work

- It would be meaningful to work out an efficient algorithm that solves the original optimization problem directly.
- It is necessary to further conduct experiments on larger datasets with more features.
Thank you for your attention!