Text Processing on the Web

Week 13
Learning to Rank / Revision

(source of LeToR slides from Tie-Yan Liu @ MSRA)
Conventional Ranking Models

• Content relevance
  – Boolean model, vector space model, probabilistic BM25 model, language model

• Page importance
  – Link analysis: HITS, PageRank, etc.
  – And by log mining
Machine Learning Can Help

• Machine learning is an effective tool
  – To automatically tune parameters
  – To combine multiple evidence
  – To avoid over-fitting (by means of regularization, etc.)

• Learning to Rank
  – Use machine learning technologies to train the ranking model
  – A hot research topic these years
Learning to Rank

Labels refer to the judgments in IR evaluation

$q_1$ queries $q_N$

$(x_1^{(1)}, 4)$, $(x_2^{(1)}, 2)$, $\ldots$, $(x_M^{(1)}, 1)$

$x_1^{(N)}, 5$, $x_2^{(N)}, 3$, $\ldots$, $x_M^{(N)}, 2$

$s$

$q$

$(x_1, ?)$, $(x_2, ?)$, $\ldots$, $(x_n, ?)$

Test data

Training Data

Learning System

Model $f(x; w)$

$\min \text{Loss}$

$\left( x_1, f(x_1; w) \right)$, $\left( x_2, f(x_2; w) \right)$, $\ldots$, $\left( x_M, f(x_M; w) \right)$

7/20/2008

Tie-Yan Liu @ Tutorial at SIGIR 2008
The general idea

- Training examples in the form of $<Q,d,\{rel,\bar{rel}\}>$
- Simple: replace $<Q,d>$ with features: $\vec{x} = \{x_0, x_1, \ldots, x_n\}$
  - Similarity of $Q,d$
  - Density of $Q$ within $d$
  - Other factors PageRank, etc.

- Train a simple learner on this data to get a probabilistic belief of
- Rank by belief on $rel$ to $\bar{rel}$
Least Squares Retrieval Function
(N. Fuhr, TOIS 1989)

• Relevance judgment for a query-document pair is represented by a vector:
  – For binary judgment: \( y = (1, 0) \) or \((0, 1)\)

• Use a polynomial function as the ranking function \( f(x) \).

• Use least square error (LSE) method to learn the regression function

\[
\min \sum_{i=1}^{N} \sum_{j=1}^{M^{(i)}} \left| y_j^{(i)} - f(x_j^{(i)}) \right|^2
\]
Discriminative Model for IR
(R. Nallapati, SIGIR 2004)

• Idea: Use discriminative modeling instead of generative model

• Generative models (i.e. via \(P(d|R) \cdot P(R)\)) include BIR and language model (in their interpretation)

• Discriminative learning algorithms (i.e. model \(P(R|d)\) directly) used:
  – Maximum Entropy
  – Support Vector Machines
Conventional ML Approach

• These are examples of a direct ML approach
• Apply regression or classification methods to solve the problem of ranking
  – Regard binary judgments or multi-valued discrete as “non-ordered” categories, or real values.
  – Although ground truths are neither “non-ordered” categories nor real values.

Serious shortcomings. What’s the problem?
Ordinal Regression

• Confusion between **relevance** with **ranking**
  – Absolute and independent relevance assumed
    • But relevance is relative and defined only among documents for the same query: a non-rel doc for a popular query may have higher TF than a rel doc for a rare query
  – Also we don’t necessary care about relevance
    • Care about ranking w.r.t other possible candidate \( d_n \), especially at top ranks
    • Relative order is important: don’t need to predict accurate category, or value of \( f(x) \).
Bridging the Gap

• Go beyond conventional ML methods
  1. Ordinal regression (a pointwise approach)
     • Target the ground truth of multi-valued discrete.
  2. Preference learning (a pairwise approach)
     • Target the ground truth of pairwise preference.
     • Also compatible with that of multi-valued discrete.
  3. Listwise ranking (a listwise approach)
     • Target the ground truth of partial / total order.
     • Also compatible with other types of ground truths.
1. Ordinal Regression: A Pointwise Approach

- **Input space**
  - Features of a single document (w.r.t. a query): \( X \in \mathbb{R}^T \)

- **Output space**
  - Ordered categories: \( Y \in \{ c_1 < c_2 < \ldots < c_K \} \)

\[
\begin{pmatrix}
q_i \\
\begin{pmatrix}
x_1^{(i)}, 5 \\
x_2^{(i)}, 3 \\
\vdots \\
x_M^{(i)}, 2
\end{pmatrix}
\end{pmatrix}
\xrightarrow{\text{Transform}}
\begin{pmatrix}
q_i \\
\{(x_1^{(i)}, c_4), (x_2^{(i)}, c_3), \ldots, (x_M^{(i)}, c_1)\}
\end{pmatrix} \quad c_1 < c_2 < c_3 < c_4
Ordinal Regression vs. Regression/Classification

- **Regression:** Real values
- **Classification:** Non-ordered categories
- **Ordinal regression:** Discrete values / Ordered categories

- Ordinal regression can be regarded as something between regression and classification.
2. Preference Learning: A Pairwise Approach

- Input space: two documents
  - Document pairs: \((X_u, X_v) \in R^T \times R^T\)
- Output space
  - Preference: \(Y \in \{+1,-1\}\)
  - Use pairs of features or differences between the two vectors

\[
\begin{pmatrix} q_i \\ x_1^{(i)}, 5 \\ x_2^{(i)}, 3 \\ \vdots \\ x_{n(i)}^{(i)}, 2 \end{pmatrix}
\xrightarrow{\text{Transform}}
\begin{pmatrix} q_i \\ \left\{ (x_1^{(i)}, x_2^{(i)}, +1), (x_2^{(i)}, x_1^{(i)}, -1), \ldots \right\} \\ \left\{ (x_2^{(i)}, x_{n(i)}^{(i)}, +1), (x_{n(i)}^{(i)}, x_2^{(i)}, -1) \right\} \end{pmatrix}
\]
Learning to Order Things

• Pairwise ranking function
  \[ f(x_u, x_v) = \sum w_i f_t(x_u, x_v) \]

• Important: pairwise loss function
  \[ L(f) = \sum_{i=1}^{N} \sum_{x_u^{(i)} > x_v^{(i)}} \left( 1 - f(x_u^{(i)}, x_v^{(i)}) \right) / \sum_{i=1}^{N} \sum_{x_u^{(i)} > x_v^{(i)}} 1 \]

• A weighted majority algorithm is used to learn the parameters \( w \) from the pairwise ground truth.
Learning to Order Things

• Go from pairwise preferences to a total order:
  - \[ \max_{\rho} AGREE(\rho, f) = \sum_{x_u, x_v : \rho(x_u) > \rho(x_v)} f(x_u, x_v) \]
  - Con: the optimal total order construction is proven NP hard.

• Then must approximate:
  - Use a greedy ordering
  - Proven: the agreement for the approximation algorithm is at least half the optimal agreement
Ranking SVM

- Formally discussed that ordinal regression can be solved by pairwise preference learning

\[
\begin{align*}
\min & \frac{1}{2} \left\| w \right\|^2 + C \sum_{i=1}^{N} \sum_{u,v} \xi_{uv}^{(i)} \\
& \left< w, x_u^{(i)} - x_v^{(i)} \right> \geq 1 - \xi_{uv}^{(i)}, \text{if } x_u^{(i)} > x_v^{(i)} \\
& \xi_{uv}^{(i)} \geq 0.
\end{align*}
\]

Use SVM to perform pairwise classification on these instances, to learn model parameter \( w \)

Use SVM to perform binary classification on these instances, to learn model parameter \( w \)

\( x_u - x_v \) as positive instance of learning

Use \( w \) for testing

Use SVM to perform pairwise classification

\[ f(x; \hat{w}) = < \hat{w}, x > \]
Results look … poor

- It is not clear how pairwise loss correlates with query-level IR evaluation measures.

Pairwise loss vs. (1-NDCG@5)

TREC Dataset
Possible Explanation?

*The more the number varies, the more pairwise is different from query-level.*
A case for query-specific loss

- Consider two queries with 40 and 5 document results. Say a system gets 780 of the 790 possible pairs correct
  - Sys 1: gets all of the $5\times4/2 = 10$ pairs from Q2 wrong
  - Sys 2: gets a random 10 of the $40\times39/2 = 780$ pairs wrong
- Clearly, we prefer Sys 2. How to cater for this?
- Change the loss function (evaluation function)
A Possible Solution

- Introduce a per-query normalization to the pairwise loss function.

\[
\min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{N} \mu^{(i)}(i) \sum_{u,v} \xi_{uv}(i)
\]

**Query-level normalizer**

\[
\max_i \frac{\# \{ \text{instance pairs associated with } q_i \}}{\# \{ \text{instance pairs associated with } q_i \}}
\]

**Loss function desiderata:**

1) Insensitive to number of document pairs.
2) Top ranks should be more important.
3) Upper bound on loss. Difficult queries shouldn’t have more importance.
Pairwise Summary

Pros:
• No longer assume absolute relevance
• Use pairwise relationship to represent relative ranking.

Cons
• Minimizing document pairs classification error and not errors in ranking of documents.
• # of generated document pairs can vary
  – Need to fix loss, otherwise model can be biased
3. A Listwise Approach

- **Input space**
  - Document collection w.r.t. a query
    \[
    (X_{1}^{(q)}, \ldots, X_{M(q)}^{(q)}) \in (R^{T})^{M(q)}
    \]

- **Output space**
  - Permutation of these documents: \( Y \in \prod_{M(q)} \)

- By treating the list of documents associated with the same query as a learning instance, one can naturally obtain
  - The rank (position) information,
  - The query-level information.

- Opportunity to model more of the unique properties of IR ranking in the learning process.
Direct Optimization of IR Measures

• Let’s try to directly optimize the ranking results.

• But this is difficult:
  – Evaluation functions such as NDCG are non-smooth and non-differentiable, since they depend on ranks
  – Most optimization was developed to handle smooth and differentiable functions

• Two methods:
  1. Smooth out the evaluation function with a surrogate;
  2. Use other optimization routines (e.g., genetic algorithms).
ListNet
(Z. Cao, T. Qin, T. Liu, et al. ICML 2007)

- Loss function = KL-divergence between two permutation probability distributions

\[ L(f) \propto D(P(\pi | e^{\psi(y)}) || P(\pi | e^{f(x)}) \right) \]

- Model = Neural Network
- Algorithm = Gradient Descent
Experimental Results

Training Performance on TD2003 Dataset

Pairwise (RankNet) vs Listwise (ListNet): Better correlation
Selected References

• N. Fuhr. Optimum polynomial retrieval functions based on the probability ranking principle, TOIS, 1989.
• R. Herbrich, T. Graepel, et al. Support Vector Learning for Ordinal Regression, ICANN1999
• R. Herbrich, T. Graepel, et al. Large Margin Rank Boundaries for Ordinal Regression, Advances in large margin classifiers, 2000
• T. Joachims, Optimizing Search Engines Using Clickthrough Data, KDD 2002.
• R. Nallapati, Discriminative model for information retrieval, SIGIR 2004.
• A. Trotman, Learning to rank, Information Retrieval, 2005
• D. Metzler, W. B. Croft, et al. Direct maximization of rank-based metrics for information retrieval, CIIR, 2005
• H. Yu, SVM Selective sampling for ranking with application to data retrieval, KDD 2005.
Revision
Announcements

• I will be away right before the exam (17-22 Nov), so please come ask questions earlier
• Send me anonymous mail (via IVLE) about what you liked about the course, what you disliked
  – Criticisms always more helpful
  – You can also save it for the “official feedback” if you’d like
Final Exam

• 2 hours, 26 Nov, in the evening
• Open book

• 3 multi-part questions, no calculation needed
  – But that doesn’t mean there’s no math
• Similar to other past year exams and more open-ended tutorial questions
Course in a nutshell

W0: Math
W1: Web basics and models
W2: Basic IR
W3: Probabilistic IR
W4: Dimensionality Reduction
W5: Link Structure
W6: Passage Retrieval

W7: Question Answering
W8: Summarization
W9: Intro to Machine Learning
W9: Text Categorization
W11: Sequence Labeling
W12: CRF + Info. Extraction
W13: Learning to Rank
Text Analysis Example

Singapore Flyer

**Singapore Flyer Pte Ltd** 30 Raffles Avenue, #01-07
Singapore 039803
Telephone: (65) 6854 5200 Fax: (65) 6339 9167

Singapore Flyer is the world's largest observation wheel. Standing at a stunning 165m from the ground, the Flyer offers you breathtaking, panoramic views of the Marina Bay, our island city and beyond. There's also a wide range of shops, restaurants, activities and facilities.

---

**Information Units**

- **IR:** terms: raffles x 1; Singapore x 3; pte x 1 ...
- **IE:** info units: Singapore Flyer, Raffles Avenue, Marina Bay, (65) 6854-5200 ...
  
  and their relations
- **QA:** Which is the nearest MRT to Singapore Flyer?
  
  Answer: City Hall MRT
  
- **NLP:** understanding the contents
W0-W1: Math and Web basics

• Size and growth of the web
  – Size: an instance of Bayesian estimation
  – Growth: instances of temporal graph modeling
    new nodes and edges added/changed over timesteps

• Compare these to other instances in the course

• Math:
  – Prior and posterior probabilities
  – Parameter estimation: EM (the chicken and egg problem)
W2-W3: Models of IR

• Heuristic systems
  – TF.IDF (compare IDF to RF in text classification)

• Prob IR
  – Model how a query is an representation of a document
  – A mathematical basis for IDF

• Language Modeling
  – Putting word order dependencies in the retrieval model
  – First look at Hidden Markov Models and n-grams
W4: Dimensionality Reduction

Link to machine learning and text classification

- Upwards of 30K dimensions, sparse vectors
- Reduce to save space, and help both recall and precision

- LSI: apply singular value decomposition to find best orthogonal axes to represent doc-term matrix
- pLSI: view this from a probabilistic interpretation, using a unigram LM and using a latent topic variable in modeling

- Both have problems determining k, # of topics/dimensions, similar to text clustering
W5: Link structure

• Dealing with hyperlinks. Can be generalized to recommendation frameworks.

• PageRank: Random Walk + Teleportation
  – Topic sensitive teleportation

• HITS: Hubs and authorities
  – Salsa: SVD

Still needs work integrating within standard IR
W6-W7: Passage retrieval and QA

Information
- Query
- Typed Query
- Expanded Query
- Documents
- Passages
- Exact Answers

System
- Query Analysis
- Query Expansion
- Document Retriever
- Passage Retriever
- Answer Extractor

IR
Passage
QA
W6-W7: Passage retrieval and QA

- From document to exact answer retrieval
- Need heavier duty processing for smaller fragments
  - Query Expansion (from external websites, from lexicons, from logs)
  - Density based retrieval towards syntactic analysis
    - Carefully targeted NLP analysis helps
  - Question Typing
    - When questions are in NL form or when we can infer more about the user’s context
W8-W12: Applying machine learning to NLP/IR tasks

- Many NLP/IR tasks can be framed as learning problems

- **Supervised**: have labeled training data; learn a function
- **Unsupervised**: have training data, no label; learn a clustering/pattern
- **Semi supervised**:
  - Small amounts of labeled data, lots of unlabeled data: text classification, named entity recognition
  - Labeled data but not at the fine-grained answer level: IE, summarization
Feature Engineering

- Domain independent
- Task independent
- Order independent
- Language independent
- Shallow NLP
- Local context statistics (TF, position)
- Orthographic
- Domain dependent
- Task dependent
- Context sensitive
- Language dependent
- Deep NLP
- Corpus wide statistics (IDF, RF)

Text problems: Dealing with 10K+ features, skewed datasets, finding an appropriate learning algorithm (not just SVMs)
W8-W12: Application areas

- Summarization
  - Selecting sentences or text units
- Text Classification
  - Selecting one or more categories for a text unit
- Sequence Labeling / Information Extraction
  - Identifying a chunk
  - Selecting a chunk tag
  - Managing co-reference
W13: Learning to Rank

• BUG
Three lessons learned

• Probabilistic analyses of text processing
  – Bayesian Analysis
• Feature/vector creation
  – Latent variables
  – Aspects of the problem and setting
• Dealing with aspects of text processing
  – Size of number of features

• Still very much open ended research topics
  – Heuristic IR still scales better
  – Adversarial IR is a real issue
  – Integration of better knowledge sources and scalability continues to be a problem
That’s it!

Thanks for learning about Text processing!