Information Storage and Retrieval

CSCE 670
Texas A&M University
Department of Computer Science & Engineering
Instructor: Prof. James Caverlee

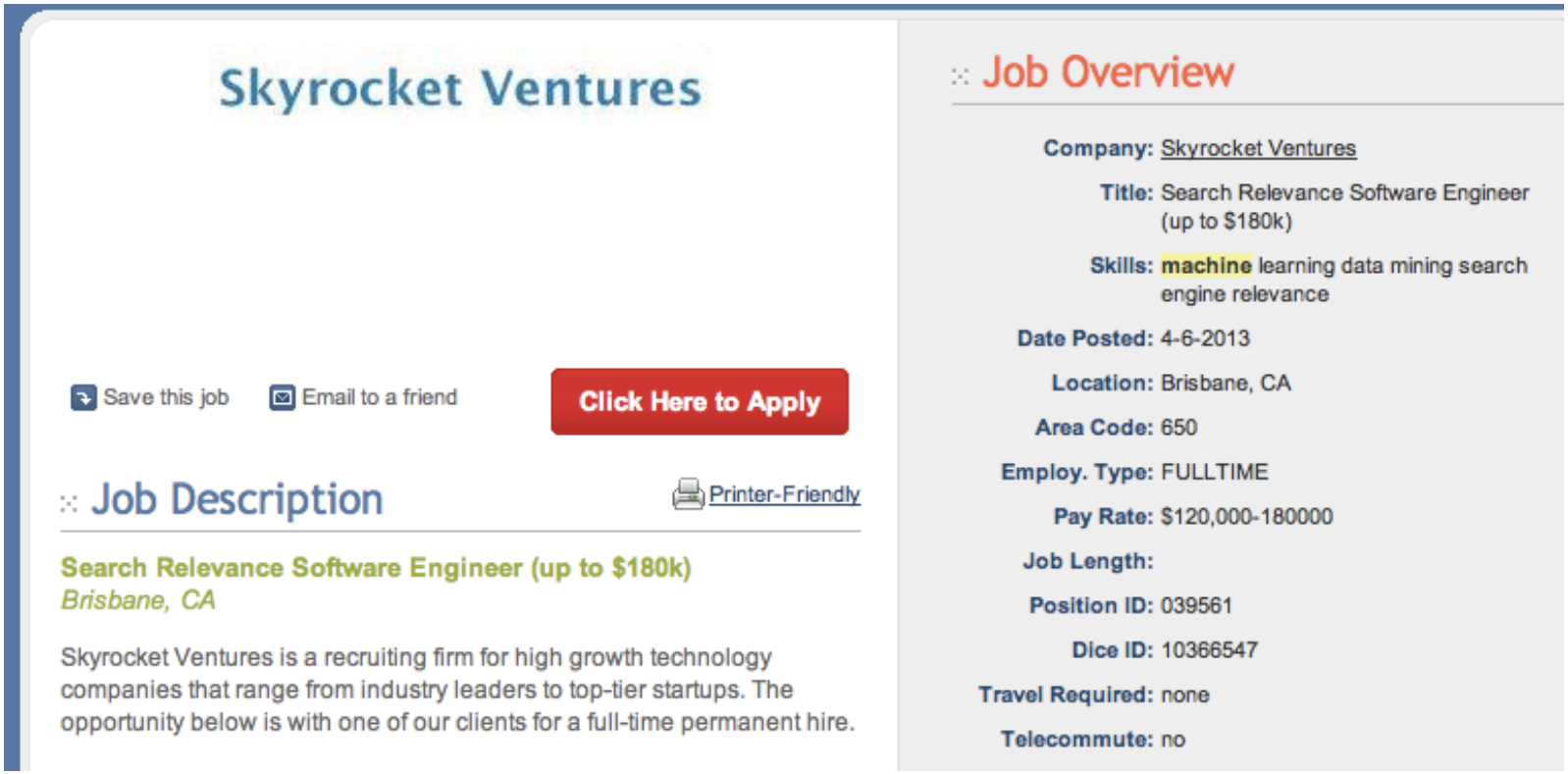
Learning to Rank
14 February 2017

Conventional Ranking Models

- Content relevance
  - Boolean, vector space, probabilistic, language model, ...
- Page importance
  - Link analysis: PageRank, HITS, ...
  - Query log mining, clickthroughs, ...

Machine learning for IR ranking?

- There's a large body of work in machine learning
- Surely we can also use machine learning to rank the documents displayed in search results?
  - Sounds like a good idea
  - => "machine-learned relevance" or "learning to rank"



* Design and implement systems and features for improving the relevance of the company's next-generation search engine
* Apply creativity and insight into the development of algorithms and tools for content classification and **machine-learned ranking** in close cooperation with the research team

Learning to rank algorithms

Least Square Retrieval Function (TOIS 1989)

Query refinement (WWW 2008)

SVM-MAP (SIGIR 2007)

Nested Ranker (SIGIR 2006)

ListNet (ICML 2007)

Pranking (NIPS 2002)

MPRank (ICML 2007)

LambdaRank (NIPS 2006)

Frank (SIGIR 2007)

Learning to retrieval info (SCC 1995)

MHR (SIGIR 2007)

RankBoost (JMLR 2003)

LDM (SIGIR 2005)

Large margin ranker (NIPS 2002)

Ranking SVM (ICANN 1999)

IRSVM (SIGIR 2006)

RankNet (ICML 2005)

Discriminative model for IR (SIGIR 2004)

SVM Structure (JMLR 2005)

OAP-BPM (ICML 2003)

Subset Ranking (COLT 2006)

GPRank (LR4IR 2007)

QBRank (NIPS 2007)

GBRank (SIGIR 2007)

Constraint Ordinal Regression (ICML 2005)

McRank (NIPS 2007)

SoftRank (LR4IR 2007)

AdaRank (SIGIR 2007)

CCA (SIGIR 2007)

ListMLE (ICML 2008)

RankCosine (IP&M 2007)

Supervised Rank Aggregation (WWW 2007)

Yahoo! Learning to Rank Challenge

Yahoo! Learning to Rank Challenge

LEARNING TO RANK CHALLENGE from YAHOO! LABS

Home | Datasets | Instructions | Registration | Submission | Leaderboard | FAQs

Benchmark your ranking algorithm against the best in industry

Though over 100 papers have been published in the learning to rank (LTR) field, most of the large-scale, real-world datasets are not publicly available. This makes drawing comparisons between algorithms difficult.

In the spirit of changing this, Yahoo! is hosting the Learning to Rank Challenge. We'll offer up two of our never before released actual datasets. These datasets—used for learning our search ranking function—can only be accessed through participation in the competition.

This exciting machine learning challenge will consist of two tracks: the first is a standard LTR track and the second is a transfer-learning track. Both tracks are open to all external research groups in academia and industry.

IMPORTANT DATES

March 1st 2010: Competition begins

May 31st 2010: Competition ends

June 2010: Winners announced

June 25th 2010: LTRC workshop at ICML conference Haifa, Israel

Machine learning for IR ranking

- This "good idea" has been actively researched and actively deployed at major web search engines in the last 5 years
- Why didn't it happen earlier?
  - Modern supervised ML has been around for about 15 years
  - Naive Bayes has been around for about 45 years!

Machine learning for IR ranking

- There's some truth to the fact that the IR community wasn't very connected to the ML community
- But there were a whole bunch of precursors:
  - Wong, S.K. et al. 1988. Linear structure in information retrieval. SIGIR 1988.
  - Fuhr, N. 1992. Probabilistic methods in information retrieval. Computer Journal.
  - Gey, F. C. 1994. Inferring probability of relevance using the method of logistic regression. SIGIR 1994.
  - Herbrich, R. et al. 2000. Large Margin Rank Boundaries for Ordinal Regression. Advances in Large Margin Classifiers.

Why weren't early attempts very successful/influential?

- Sometimes an idea just takes time to be appreciated…
- Limited training data
  - Especially for real world use (as opposed to writing academic papers), it was very hard to gather test collection queries and relevance judgments that are representative of real user needs and judgments on documents returned
    - This has changed, both in academia and industry
- Poor machine learning techniques
- Insufficient customization to IR problem
- Not enough features for ML to show value

- Microsoft LETOR

Why wasn't ML much needed?

- Traditional ranking functions in IR used a very small number of features, e.g.,
  - Term frequency
  - Inverse document frequency
  - Document length
- It was easy to tune weighting coefficients by hand
  - And people did

Why is ML needed now

- Modern systems – especially on the Web – use a great number of features:
  - Arbitrary useful features – not a single unified model
  - Log frequency of query word in anchor text?
  - Query word in color on page?
  - # of images on page?
  - # of (out) links on page?
  - PageRank of page?
  - URL length?
  - URL contains "~"?
  - Page edit recency?
  - Page length?
- The New York Times (2008-06-03) quoted Amit Singhal as saying Google was using **over 200 such features.**

134 Features released from Microsoft Research on 16 June 2010

http://research.microsoft.com/en-us/projects/mslr/feature.aspx

Zones: body, anchor, title, url, whole document

Features: query term number, query term ratio, stream length, idf, sum of term frequency, min of term frequency, max of term frequency, mean of term frequency, variance of term frequency, sum of stream length normalized term frequency, min of stream length normalized term frequency, max of stream length normalized term frequency, mean of stream length normalized term
frequency, variance of stream length normalized term frequency, sum of tf*idf, min of tf*idf, max of tf*idf, mean of tf*idf, variance of tf*idf, boolean model, vector space model, BM25, LMIR.ABS, LMIR.DIR, LMIR.JM, number of slash in url, length of url, inlink number, outlink number, PageRank, SiteRank, QualityScore, QualityScore2, query-url click count, url click count, url dwell time.