Cornell University
Computer Science

Information Retrieval as Structured Prediction

CS 6784 March 4th, 2010

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

$$x_{q,d} = \phi(q,d) = \begin{bmatrix} \sum_i \blacksquare_i \text{ appears in title} \\ \sum_i \blacksquare_i \text{ appears in first paragraph} \\ \sum_i \blacksquare_i \text{ appears in anchor text linking to } d \\ \cos(q,d) \\ pagerank(d) \end{bmatrix}$$

Learning to Rank

- Design a retrieval function $f(x) = w^T x$
  – (weighted average of features)
- For each query q
  – Score all $s_{q,d} = w^T x_{q,d}$
  – Sort by $s_{q,d}$ to produce ranking
- Which weight vector w is best?

Outline

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

Mean Average Precision

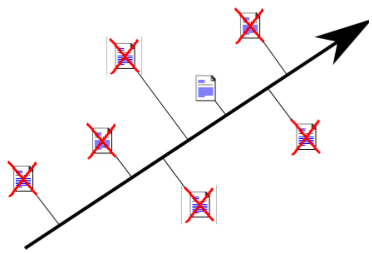
- Consider rank position of each relevance doc
  – $K_1, K_2, \ldots K_R$
- Compute Precision@K for each $K_1, K_2, \ldots K_R$
- Average precision = average of P@K
- Ex: has AvgPrec of $\frac{1}{3} \cdot \left(\frac{1}{1} + \frac{2}{3} + \frac{3}{5} \right) \approx 0.76$
- MAP is Average Precision across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

MAP vs Accuracy

Rel?	1	0	0	0	0	1	1	1	1	0	0
H1	11	10	9	8	7	6	5	4	3	2	1
H2	1	2	3	4	5	6	7	8	9	10	11

Ranking	MAP	Best Acc
H1	0.56	0.64
H2	0.51	0.73

Optimizing Pairwise Agreements



- 2 pairwise disagreements

Pairwise Preferences SVM

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

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

Rel?	1	0	0	0	0	1	1	0
H1	8	7	6	5	4	3	2	1
H2	1	2	3	4	5	6	7	8

Ranking	MAP	ROC-area
H1	0.59	0.47
H2	0.51	0.53

Linear Discriminant for Ranking

- Let **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.

$$\Psi(\mathbf{y}, \mathbf{x}) = \sum_{j:rel} \sum_{k:!rel} y_{jk} \cdot (x_j - x_k)$$

- Prediction made by sorting on document scores $w^T x_i$

$$\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} w^T \Psi(\mathbf{y}, \mathbf{x})$$

Structural SVM

- Let **x** denote a structured input (candidate documents)
- Let **y** denote a structured output (ranking)

- Standard objective function: $\frac{1}{2} w^2 + \frac{C}{N} \sum_i \xi_i$

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

$$\boxed{\forall \mathbf{y}' \neq \mathbf{y}^{(i)} : \; w^T \Psi(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) \geq w^T \Psi(\mathbf{y}', \mathbf{x}^{(i)}) + \Delta_i(\mathbf{y}') - \xi_i}$$

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

Structural SVM for MAP

- Minimize $\frac{1}{2} w^2 + \frac{C}{N} \sum_i \xi_i$

subject to $\forall \mathbf{y}' \neq \mathbf{y}^{(i)} : \; w^T \Psi(\mathbf{y}^{(i)}, \mathbf{x}^{(i)}) \geq w^T \Psi(\mathbf{y}', \mathbf{x}^{(i)}) + \Delta_i(\mathbf{y}') - \xi_i$

where $\Psi(\mathbf{y}^{(i)}, \mathbf{x}) = \sum_{j:rel} \sum_{k:!rel} y_{jk}^{(i)} \cdot (x_j^{(i)} - x_k^{(i)})$ (y_{jk} = {-1, +1})

and $\Delta(\mathbf{y}') = 1 - Avgprec(\mathbf{y}')$

- Sum of slacks $\sum \xi_i$ is **smooth** upper bound on MAP loss.

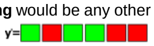
[Yue, Finley, Radlinski, Joachims; SIGIR 2007]

Too Many Constraints!

- For Average Precision, the **true labeling** is a ranking where the relevant documents are all ranked in the front, e.g.,

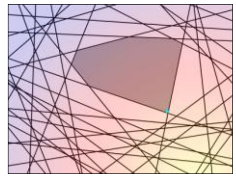
y= 

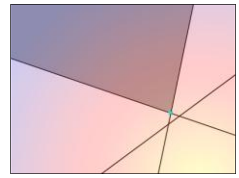
- An **incorrect labeling** would be any other ranking, e.g.,

y'= 

- This ranking has Average Precision of about 0.8 with $\Delta(\mathbf{y}') \approx 0.2$
- **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

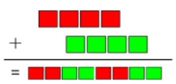
- Repeatedly finds the next most violated constraint…
- …until set of constraints is a good approximation.

Finding Most Violated Constraint

$$\arg\max_{\mathbf{y}'} \Delta(\mathbf{y}') + \sum_{j:rel}\sum_{k:!rel} y'_{jk} \cdot (w^T x_j - w^T x_k)$$

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

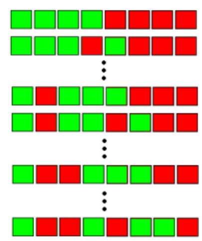


[Yue et al., SIGIR 2007]

Finding Most Violated Constraint

$$\arg\max_{\mathbf{y}'} \Delta(\mathbf{y}') + \sum_{j:rel}\sum_{k:!rel} y'_{jk} \cdot (w^T x_j - w^T x_k)$$

- 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

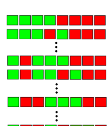


[Yue et al., SIGIR 2007]

Proof (Sketch)

$$H(\mathbf{y}) = \Delta(\mathbf{y}) + \sum_{j:rel}\sum_{k:!rel} y_{jk} \cdot (w^T x_j - w^T x_k)$$

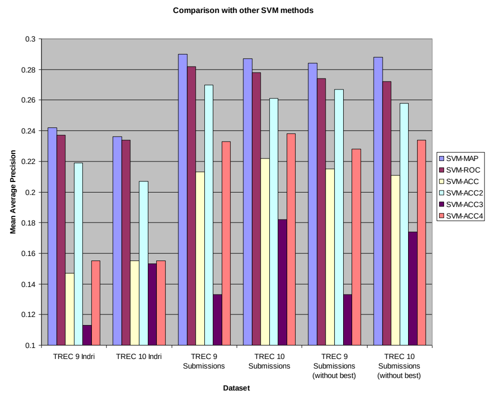
- Assume relevant and non-relevant docs are sorted
- Define $\delta_k(i,j)$ as the change in H when:
  - The highest ranked relevant document after x_k changes from x_j to x_i
  - i and j index relevant documents (i < j)
  - k indexes non-relevant document
- Need to show $\delta_{k+1}(i,i+1) \le \delta_k(i,i+1)$



[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.



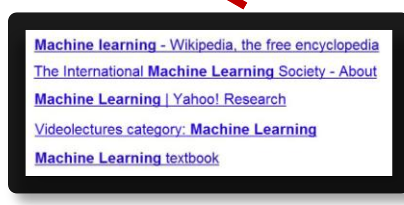
Finding Most Violated Constraint

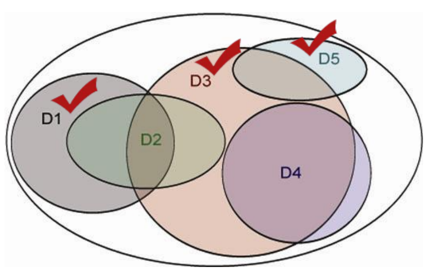
- Required for structural SVM training
  - Depends on structure of loss function
  - Depends on structure of the feature map
  - Efficient algorithms exist despite intractable number of constraints.
- More than one approach
  - [Yue et al., 2007]
  - [Chapelle et al., 2007]

Story so Far

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

Bobby Kleinberg
the curious high school student



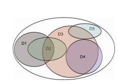


- Choose top 3 documents
  - Individual Relevance: D3 D4 D1
  - **Coverage Solution: D3 D1 D5**

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

Document Word Counts

	V1	V2	V3	V4	V5
D1			X	X	X
D2		X		X	X
D3	X	X	X	X	

Word	Benefit
V1	1
V2	2
V3	3
V4	4
V5	5

Marginal Benefit

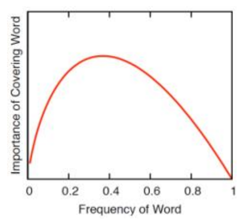
	D1	D2	D3	Best
Iter 1	12	11	10	D1
Iter 2	--	2	3	D3

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** = ($x_1, x_2, \ldots, x_n$) - candidate documents
- v – an individual word

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

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

[Yue & Joachims, ICML 2008]

Structured Prediction for Maximizing Coverage

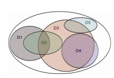
- **x** = ($x_1, x_2, \ldots, x_n$) - candidate documents
- **y** – subset of **x** of size K (the prediction)
- *V*(**y**) – union of words from **y**
- Discriminant Function: $w^T\Psi(\mathbf{x},\mathbf{y}) = \sum_{v \in V(\mathbf{y})} w^T\phi(v,\mathbf{x})$
- Benefit of covering word *v* is $w^T\phi(v,\mathbf{x})$

$$\boxed{\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} w^T\Psi(\mathbf{x},\mathbf{y})}$$

[Yue & Joachims, ICML 2008]

Structured Prediction for Maximizing Coverage

$$\boxed{w^T\Psi(\mathbf{x},\mathbf{y}) = \sum_{v \in V(\mathbf{y})} w^T\phi(v,\mathbf{x})}$$



- 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

- Documents “cover” words to different degrees
  - A document with 5 copies of “Microsoft” might cover it better than another document with only 2 copies.
- Use multiple word sets, $V_1(\mathbf{y})$, $V_2(\mathbf{y})$, … , $V_L(\mathbf{y})$
- Each $V_i(\mathbf{y})$ contains only words satisfying certain importance criteria.

[Y, Joachims; ICML 2008]

More Sophisticated Discriminant

$$\Psi(\mathbf{y},\mathbf{x}) = \begin{bmatrix} \sum_{v \in V_1(\mathbf{y})} \phi_1(v,\mathbf{x}) \\ \vdots \\ \sum_{v \in V_L(\mathbf{y})} \phi_L(v,\mathbf{x}) \end{bmatrix}$$

- Separate ϕ_i for each importance level *i*.
- Joint feature map Ψ is vector composition of all ϕ_i

$$\boxed{\hat{\mathbf{y}} = \arg\max_{\mathbf{y}} w^T\Psi(\mathbf{y},\mathbf{x})}$$

- 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)
- Standard SVM objective function: $\frac{1}{2}w^2 + \frac{C}{N}\sum_i \xi_i$
- Constraints are defined for each incorrect labeling y’ over the set of documents x.

$$\boxed{\forall \mathbf{y}' \neq \mathbf{y}^{(i)}: \; w^T\Psi(\mathbf{x}^{(i)},\mathbf{y}^{(i)}) \geq w^T\Psi(\mathbf{x}^{(i)},\mathbf{y}') + \Delta_i(\mathbf{y}') - \xi_i}$$

[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

	# Docs	Loss
t_1	3	1/2
t_2	1	1/6
t_3	2	1/3

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

[Yue & Joachims, ICML 2008]

Finding Most Violated Constraint

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

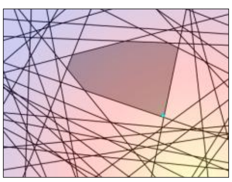
- Encode each subtopic as an additional "word" to be covered.

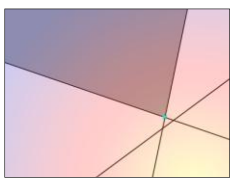
$$\Psi'(\mathbf{x},\mathbf{y}') = \begin{bmatrix} \sum_{v\in V_1(\mathbf{y}')} \phi_1(v,\mathbf{x}) \\ \vdots \\ \sum_{v\in V_L(\mathbf{y}')} \phi_L(v,\mathbf{x}) \\ \sum_{T\notin \mathrm{T}(\mathbf{y}')} \Delta_T \end{bmatrix}$$

- Use greedy prediction to find approximate most violated constraint.

$$\boxed{\hat{\mathbf{y}} = \arg\max_{\mathbf{y}'} w^T \Psi'(\mathbf{x},\mathbf{y}')}$$

Approximate Constraint Generation



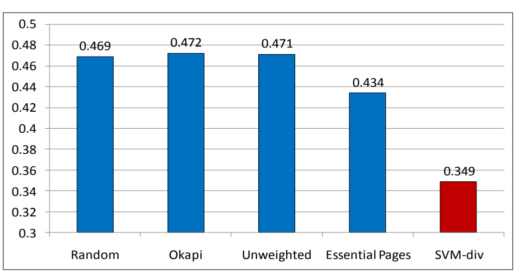


- Theoretical guarantees still hold.
  - Constant factor approximation to finding optimal cutting plane
  - (1-1/e) approximation for solving coverage problems
- Performs well in practice.

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