Rule-Based Active Sampling for Learning to Rank

Rodrigo Silva, Marcos A. Gonçalves and Adriano Veloso

Department of Computer Science - Federal University of Minas Gerais, Brazil

ECML PKDD, Athens, 2011/09

Outline

- Active Sampling motivation
- Rule-based supervised Learning to Rank (L2R)
- Active Sampling using association rules
- Experimental setup and results
- Discussion

Active Learning Motivation

- Many applications can benefit from Learning to Rank methods: document search, product recommendation, etc.
- Creating training sets can be expensive, as labels must be assessed by human annotators
- Is it possible to select only very "informative" instances to be labeled and obtain good results?
- Do carefully selected sets improve learned models by avoiding noise in the training data?
- Is it possible to iteratively create an actively sampled training set without an initial (labeled) seed set?

Learning to Rank using Association Rules

- Training set $\mathcal{D}$ composed of records $< q, d, r >$
- Documents represented as a list of m feature-values $\{f_1, f_2, \ldots, f_m\}$ (e.g. PageRank, BM25, etc.)
- Relevance of d to q from a discrete set of possibilities $\{r_0, r_1, \ldots, r_k\}$ (e.g. $r_0 = 0$, $r_1 = 1$ and $r_2 = 2$)
- Test set $\mathcal{T}$ with records $< q, d, ? >$
- Ranking functions obtained from $\mathcal{D}$ are used to estimate the relevance
- $\mathcal{R}$ a rule-set composed of rules of the form $\{f_j \wedge \ldots \wedge f_l \xrightarrow{\theta} r_i\}$
- θ is the conditional probability of the consequent given the antecedent

Learning to Rank using Association Rules

- The search space for rules is potentially huge
- A support threshold σ_{min} may be used to limit rule extraction
  - But if σ_{min} is set too low we may have too many rules
  - Most of which are useless for estimating the relevance of documents in $\mathcal{T}$ (a rule $\{\mathcal{X} \rightarrow r_i\}$ is only useful for $d \in \mathcal{T}$ if $\mathcal{X} \subseteq d$)
  - If σ_{min} is set too high, important rules may not be included in $\mathcal{R}$
- Instead, we can do on-demand rule extraction at query time
- Once a set of documents is retrieved for a query in $\mathcal{T}$, each document d is used to filter $\mathcal{D}$, creating a projected training set $\mathcal{D}_d$
- Then, a specific rule-set, $\mathcal{R}_d$ extracted from $\mathcal{D}_d$, is produced for each document d in $\mathcal{T}$

Relevance Estimation

- Each rule $\{\mathcal{X} \xrightarrow{\theta} r_i\} \in \mathcal{R}_d$ is a vote given by a set of features $\mathcal{X}$ for relevance level r_i where each vote has a weight θ
- The score of a document regarding a relevance level is

$$s(d, r_i) = \frac{\sum \theta(\mathcal{X} \rightarrow r_i)}{\mid \mathcal{R}_d \mid}, \text{ where } \mathcal{X} \subseteq d \tag{1}$$

- We normalize the scores, obtaining the likelihood that a document has relevance r_i

$$\hat{p}(r_i|d) = \frac{s(d, r_i)}{\sum_{j=0}^{k} s(d, r_j)} \tag{2}$$

- Finally, the rank of document is estimated by the linear combination of the likelihoods of each r_i

$$rank(d) = \sum_{i=0}^{k} \left(r_i \times \hat{p}(r_i|d)\right) \tag{3}$$

SSAR - Selective Sampling using Association Rules

- We want to select from an unlabeled set $\mathcal{U} = \{u_1, u_2, \ldots, u_n\}$ a set of informative documents to compose our training set $\mathcal{D}$
- If a document $u_i \in \mathcal{U}$ is inserted into $\mathcal{D}$, the number of rules for documents in $\mathcal{U}$ that share feature-values with u_i will either increase or remain unchanged
- The number of rules for documents that *do not share* feature-values with u_i will remain unchanged
- Therefore, the number of rules extracted for each document in $\mathcal{U}$ can be used as an approximation of the amount of redundant information between documents already in $\mathcal{D}$ and documents in $\mathcal{U}$
- SSAR selects from $\mathcal{U}$ documents that contribute non-redundant information: i.e. documents that demand the fewer number of rules from $\mathcal{D}$

SSAR - Selective Sampling using Association Rules

- The sampling function $\gamma(\mathcal{U})$ returns a document in $\mathcal{U}$:

$$\gamma(\mathcal{U}) = \{u_i \text{ such that } \forall u_j : \ |\mathcal{R}_{u_i}| \ \leq \ |\mathcal{R}_{u_j}|\} \tag{4}$$

- At each round, the document returned by $\gamma(\mathcal{U})$ is inserted into $\mathcal{D}$ (but remains in $\mathcal{U}$)
- The document which demands the fewest rules is the one which shares the least possible number of feature-values with documents already in $\mathcal{D}$
- Initially, $\mathcal{D}$ is empty and SSAR selects the document that shares more feature-values with the other documents of the collection and can be considered as the best representative of it
- Eventually, $\gamma(\mathcal{U})$ selects a document already in $\mathcal{D}$

Experimental Setup and Results

- LETOR 3.0 datasets
- Greedy non-parametric density estimation algorithm that uses the log-likelihood to discretize each attribute into 10 bins
- Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain (NDCG)

Table: SSAR MAP Results and Statistics

	SSAR	LRAR	Sel	Utot	Sel%
TD2003	0.2032	0.2459	157	29,435	0.53
TD2004	0.1792	0.2463	141	44,488	0.32
NP2003	0.7202	0.6373	207	89,194	0.23
NP2004	0.4993	0.5155	181	44,300	0.41
HP2003	0.6487	0.7083	218	88,564	0.25
HP2004	0.6332	0.5443	222	44,645	0.50

Delaying Convergence - SSARP

- Vertically partitioning unlabeled set into groups of features, selecting instances from each partition and then using the full instances
- The objective is to increase the number of selected instances AND their diversity
- How many partitions?
  - Few features per partition: selected instances may not be informative
  - Few features per partition: SSAR may converge too fast
  - Too many features: SSAR may converge too fast
  - From 8 to 12 features per partition provided a good balance for 2 datasets tested
- How to select which features to put into each partition?
- As a simple measure of the informativeness of each feature, we calculate the χ^2 of each attribute in relation to the others. From this $n \times n$ matrix we calculate a score for each feature and rank them in descending order of rank. Finally, we "spread" the features into the partitions.

SSARP Results

- Selective sampling using 5 partitions (12 features per partition)

Table: SSARP MAP Results and Statistics

	SSARP	LRAR	Sel	Utot	Sel%
TD2003	0.2689	0.2459	642	29,435	2.18
TD2004	0.2006	0.2463	633	44,488	1.42
NP2003	0.6960	0.6373	995	89,194	1.12
NP2004	0.5499	0.5155	860	44,300	1.94
HP2003	0.7411	0.7083	1091	88,564	1.23
HP2004	0.6168	0.5443	855	44,645	1.91

Using selected instances with other L2R methods

- Running SVMRank with the instances selected by SSARP
- Selecting the same amount of instances by their BM25 value
- Randomly selecting the same amount of instances

Table: MAP for SVM using selected samples, BM25 and Random Baselines

	SSARP	SVMS	SBM25	Random	G%
TD2003	0.2689	0.2194	0.1568	0.1417±0.0285	39.95
TD2004	0.2006	0.1957	0.1335	0.1687±0.0145	16.01
NP2003	0.6960	0.6428	0.6587	0.5739±0.0237	-2.41
NP2004	0.5499	0.5929	0.5811	0.5787±0.0329	2.04
HP2003	0.7411	0.6747	0.7090	0.5798±0.0592	-4.84
HP2004	0.6168	0.6734	0.6731	0.5406±0.0357	0.05

Comparing the results to LETOR baselines

- LETOR 3.0 published baselines for 12 L2R algorithms using the complete training set
- We select RankBoost, FRank and Regression for comparison

Table: MAP for SSARP and LETOR Baselines

	SSARP	**RBoost**	**FRank**	**REG**
TD2003	**0.2689**	*0.2274*	*0.2031*	*0.2409*
TD2004	0.2006	0.2614	0.2388	0.2078
NP2003	**0.6960**	0.7074	*0.6640*	*0.5644*
NP2004	*0.5499*	0.5640	0.6008	*0.5142*
HP2003	**0.7411**	*0.7330*	*0.7095*	*0.4968*
HP2004	*0.6168*	0.6251	0.6817	*0.5256*
Avg.	*0.5122*	0.5197	0.5163	*0.4250*

Comparison with other Active Learning methods

- Previous work propose methods that require a labeled seed set to train the initial learner
- Published results for TD2003 and TD2004 report selecting from 11 to 15% of the original training sets
- In contrast, our method does not rely on an initial set and provides competitive results selecting only 2.2% of the original training sets

Questions?