



Classification

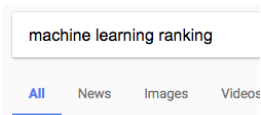
Jordan Boyd-Graber
University of Maryland

RANKING

Roadmap

- Combining rankings: taking advantage of multiple weak rankers
- Maximum margin ranking: support vector machines
- Reduction to classification: optimizing

Ranking



About 75,800,000 results (0.58 seconds)

Learning to rank - Wikipedia

https://en.wikipedia.org/wiki/Learning_to_rank
Learning to rank or machine-learned ranking is a supervised, semi-supervised or reinforcement learning problem. Applications · Feature vectors · Evaluation

Result ranking by machine learning

<https://nlp.stanford.edu/IR-book/html/sect04.1.html>
Result ranking by machine learning. There are more than two variables. There are lots of

[PDF] Ranking Methods in Machine Learning

www.shivani-agarwal.net/Events/SD
by S Agarwal - Cited by 1 - Related articles
C.J.C. Burges, T. Shaked, E. Renshaw, J. Bennett, K. El-Nehry, S. S. Choudhry, using gradient descent, ICML 2005. S. Agarwal, Generalization bounds for the area under the ROC curve, 2005.

Given input $x_1 \dots x_r$, return permutation of $[r]$. Permutation often parameterized by vector of scalars $y_1 \dots y_r$.

- Web search (Google used > 200 features)
- Movie rankings
- Dating

What's the goal? Loss functions ...

- Kendall- τ

$$L(y', y) = \frac{2}{r(r-1)} \sum_i \sum_j \mathbb{1}[\text{sign}(y'_i - y'_j) \neq (y_i - y_j)] \quad (1)$$

What's the goal? Loss functions ...

- Kendall- τ

$$L(y', y) = \frac{2}{r(r-1)} \sum_i \sum_j \mathbb{1}[\text{sign}(y'_i - y'_j) \neq (y_i - y_j)] \quad (1)$$

- Normalized Discounted Cumulative Gain:

$$D(i) = \frac{1}{\lg(r-i+2)} \text{ if } i \in \{r-k+1, \dots, r\} \quad (2)$$

$$G(y', y) = \sum_i D(\pi(y')_i) y_i \quad (3)$$

$$(4)$$

Discount function: focus on top k elements in list

What's the goal? Loss functions ...

- Kendall- τ

$$L(y', y) = \frac{2}{r(r-1)} \sum_i \sum_j \mathbb{1}[\text{sign}(y'_i - y'_j) \neq (y_i - y_j)] \quad (1)$$

- Normalized Discounted Cumulative Gain:

$$D(i) = \frac{1}{\lg(r-i+2)} \text{ if } i \in \{r-k+1, \dots, r\} \quad (2)$$

$$G(y', y) = \sum_i D(\pi(y')_i) y_i \quad (3)$$

(4)

Gain function: weight examples based on whether they are in important part of list, defined by permutation π . For example, for $r = 5$, the vector $y = (2, 1, 6, 1, 0.5)$ induces the permutation $\pi(y) = (4, 3, 5, 1, 2)$

What's the goal? Loss functions ...

- Kendall- τ

$$L(y', y) = \frac{2}{r(r-1)} \sum_i \sum_j \mathbb{1}[\text{sign}(y'_i - y'_j) \neq (y_i - y_j)] \quad (1)$$

- Normalized Discounted Cumulative Gain:

$$D(i) = \frac{1}{\lg(r-i+2)} \text{ if } i \in \{r-k+1, \dots, r\} \quad (2)$$

$$G(y', y) = \sum_i D(\pi(y')_i) y_i \quad (3)$$

$$L(y', y) = \sum_i \frac{1}{G(y, y)} \sum_i (D(\pi(y)_i) - D(\pi(y')_i)) y_i \quad (4)$$

$$(5)$$

Loss function focuses on how wrong top of list is

Examples as feature vectors

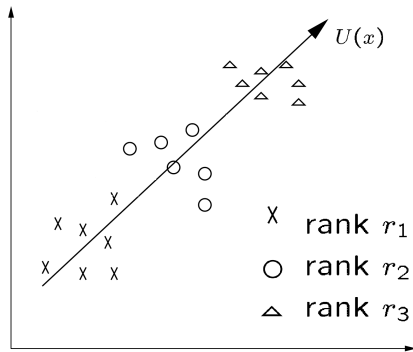
Every example has a feature vector $f(x)$

example	docID	query	cosine score	ω	judgment
Φ_1	37	linux operating system	0.032	3	<i>relevant</i>
Φ_2	37	penguin logo	0.02	4	<i>nonrelevant</i>
Φ_3	238	operating system	0.043	2	<i>relevant</i>
Φ_4	238	runtime environment	0.004	2	<i>nonrelevant</i>
Φ_5	1741	kernel layer	0.022	3	<i>relevant</i>
Φ_6	2094	device driver	0.03	2	<i>relevant</i>
Φ_7	3191	device driver	0.027	5	<i>nonrelevant</i>

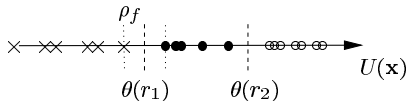
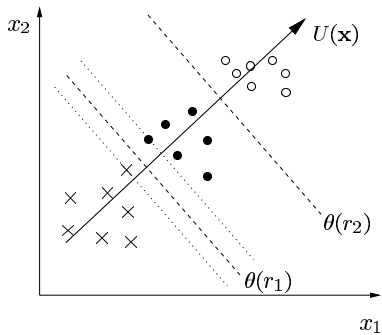
Turning features to rank

- Have a series of “levels” or ranks $y = 1 \dots$
- We want to find a function to separate examples

$$f(x) \equiv \langle w \cdot \phi(x) \rangle \quad (6)$$



Maximizing the margin



Recap

- Ranking is an important problem
- Different objective function
- Implementation similar to regression