

# Classification

Jordan Boyd-Graber University of Maryland RANKING

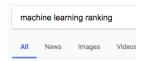
lordan Boyd-Graber | UMD Classification | 1 / 1

#### Roadmap

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

Jordan Boyd-Graber | UMD Classification | 2/1

#### Ranking



About 75,800,000 results (0.58 seconds

#### Learning to rank - Wikipedia

https://en.wikipedia.org/wiki/Learnir Learning to rank or machine-learned rai supervised, semi-supervised or reinforc Applications · Feature vectors · Evaluati

# Result ranking by machine le

https://nlp.stanford.edu/IR-book/htn Result ranking by machine learning. Th more than two variables. There are lots

#### [PDF] Ranking Methods in Mac www.shivani-agarwal.net/Events/SD

www.shivani-agarwal.net/Events/SD by S Agarwal - Cited by 1 - Related artic C.J.C. Burges, T. Shaked, E. Renshaw, & using gradient descent, ICML 2005. S. & Generalization bounds for the area und 425, 2005. Given input  $x_1 ldots x_r$ , return permutation of [r]. Permutation often parameterized by vector of scalars  $y_1 ldots y_r$ .

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

Kendall-τ

$$L(y',y) = \frac{2}{r(r-1)} \sum_{i} \sum_{j} \mathbb{1} \left[ sign(y'_i - y'_j) \neq (y_i - y_j) \right]$$
 (1)

Jordan Boyd-Graber | UMD Classification | 4 / 1

Kendall-τ

$$L(y',y) = \frac{2}{r(r-1)} \sum_{i} \sum_{j} \mathbb{1} \left[ sign(y'_i - y'_j) \neq (y_i - y_j) \right]$$
 (1)

Normalized Discounted Cumulative Gain:

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

$$G(y',y) = \sum_{i} D(\pi(y')_i) y_i$$
(3)

(4)

Discount function: focus on top k elements in list

Jordan Boyd-Graber | UMD Classification

Kendall-τ

$$L(y', y) = \frac{2}{r(r-1)} \sum_{i} \sum_{j} \mathbb{1} \left[ sign(y'_i - y'_j) \neq (y_i - y_j) \right]$$
 (1)

Normalized Discounted Cumulative Gain:

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

$$G(y',y) = \sum_{i} D(\pi(y')_i) y_i$$
(3)

(4)

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

Kendall- $\tau$ 

$$L(y',y) = \frac{2}{r(r-1)} \sum_{i} \sum_{j} \mathbb{1} \left[ sign(y_i' - y_j') \neq (y_i - y_j) \right]$$
 (1)

Normalized Discounted Cumulative Gain:

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

$$G(y', y) = \sum_{i} D(\pi(y')_{i}) y_{i}$$
 (3)

$$L(y', y) = \sum_{i} \frac{1}{G(y, y)} \sum_{i} (D(\pi(y)_{i}) - D(\pi(y')_{i})) y_{i}$$
 (4)

(5)

Loss function focuses on how wrong top of list is

# **Examples as feature vectors**

# Every example has a feature vector f(x)

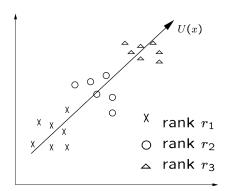
| example        | docID | query                  | cosine score | ω | judgment    |
|----------------|-------|------------------------|--------------|---|-------------|
| Φ <sub>1</sub> | 37    | linux operating system | 0.032        | 3 | relevant    |
| $\Phi_2$       | 37    | penguin logo           | 0.02         | 4 | nonrelevant |
| $\Phi_3$       | 238   | operating system       | 0.043        | 2 | relevant    |
| $\Phi_4$       | 238   | runtime environment    | 0.004        | 2 | nonrelevant |
| $\Phi_5$       | 1741  | kernel layer           | 0.022        | 3 | relevant    |
| $\Phi_6$       | 2094  | device driver          | 0.03         | 2 | relevant    |
| $\Phi_7$       | 3191  | device driver          | 0.027        | 5 | nonrelevant |

Jordan Boyd-Graber | UMD Classification | 5/1

#### Turning features to rank

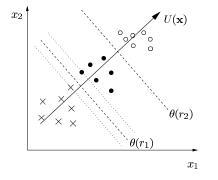
- Have a series of "levels" or ranks y = 1...
- We want to find a function to separate examples

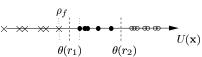
$$f(x) \equiv \langle w \cdot \phi(x) \rangle \tag{6}$$



Jordan Boyd-Graber | UMD Classification |

# Maximizing the margin





lordan Boyd-Graher | LIMD Classification |

#### Recap

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

Jordan Boyd-Graber | UMD Classification | 8 / 1