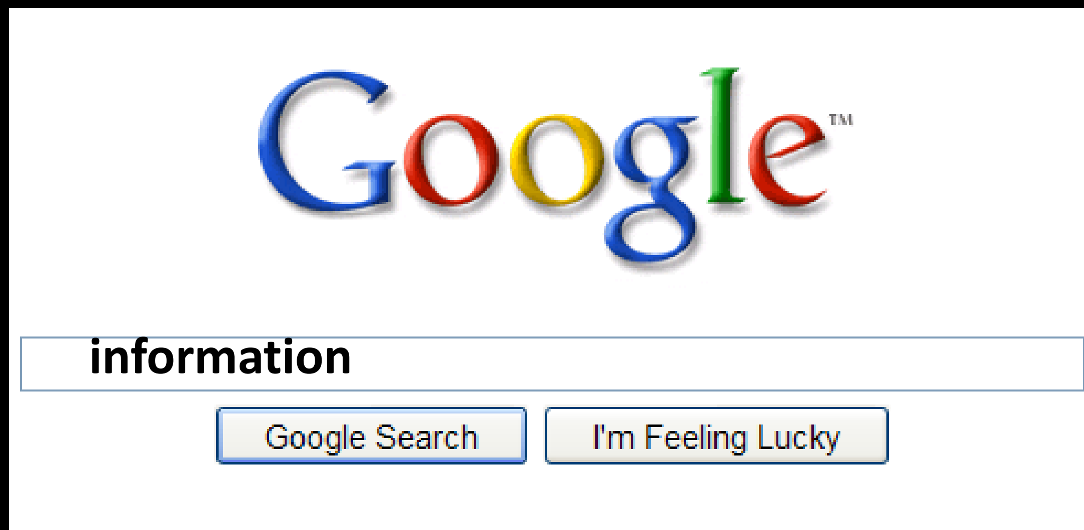
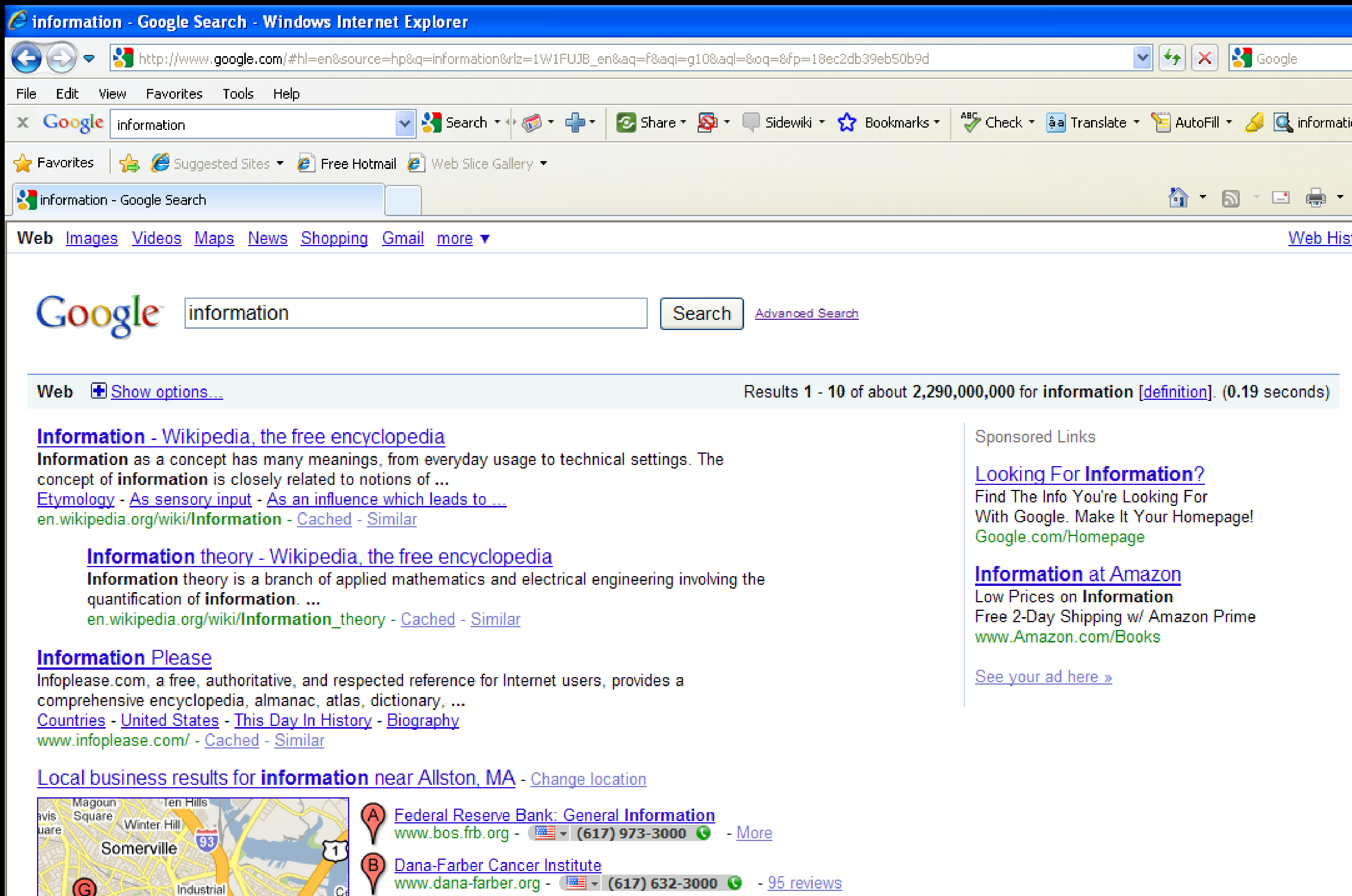
Ranking Methods in Machine Learning

Example 1: Recommendation Systems

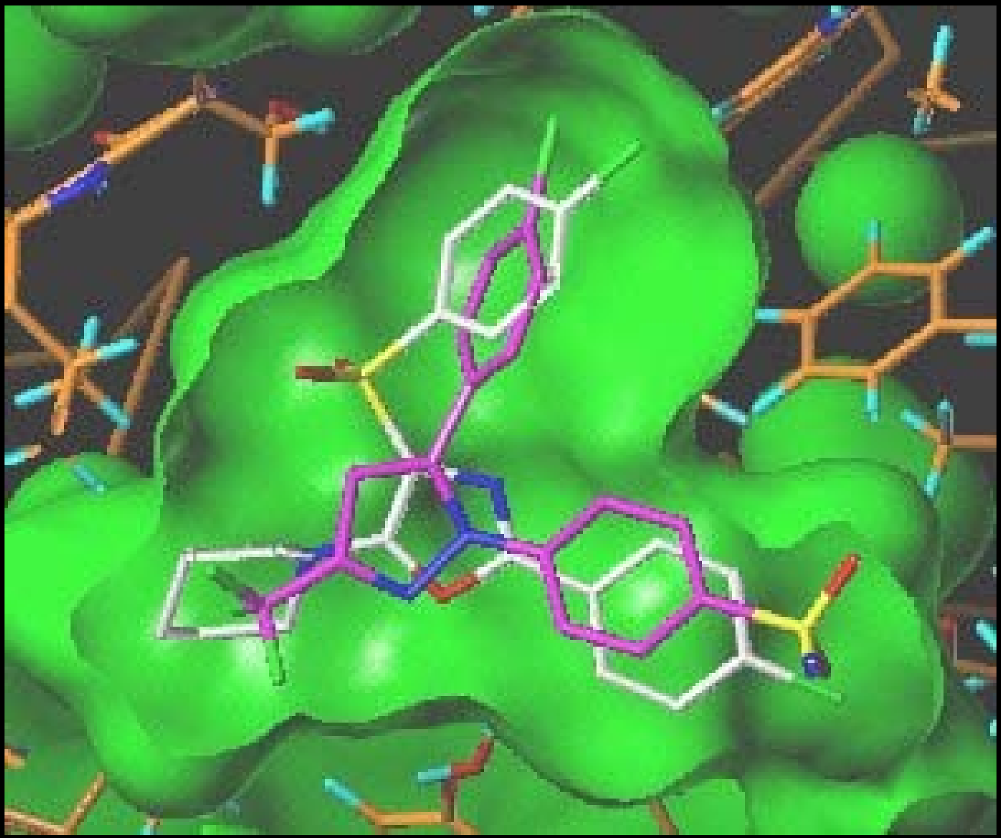
Example 2: Information Retrieval

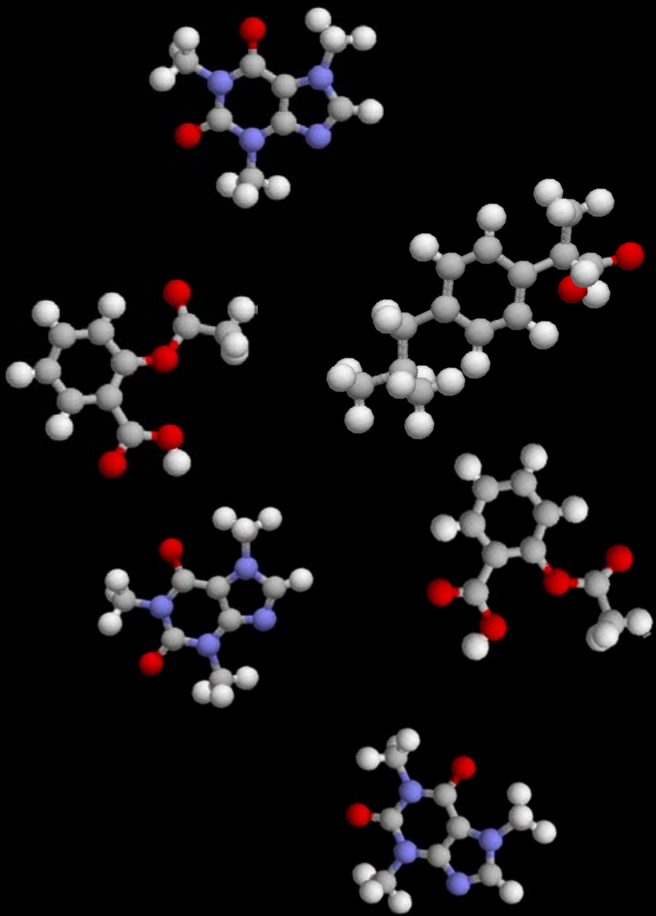


Example 2: Information Retrieval



Example 3: Drug Discovery





Problem: Millions of structures in a chemical library. How do we identify the most promising ones?

Example 4: Bioinformatics

nature

Searching for genetic determinants in the new millennium

N.J. Risch

Human genetics is now at a critical juncture. The molecular methods used successfully to identify the genes underlying rare mendelian syndromes are failing to find the numerous genes causing more common, familial, non-mendelian diseases . . .

Nature **405**:847–856, 2000

Example 4: Bioinformatics

Searching for genetic determinants in the new millennium

N.J. Risch

With the human genome sequence nearing completion, new opportunities are being presented for unravelling the complex genetic basis of nonmendelian disorders based on large-scale genomewide studies . . .

Nature **405**:847–856, 2000

Types of Ranking Problems

Instance Ranking

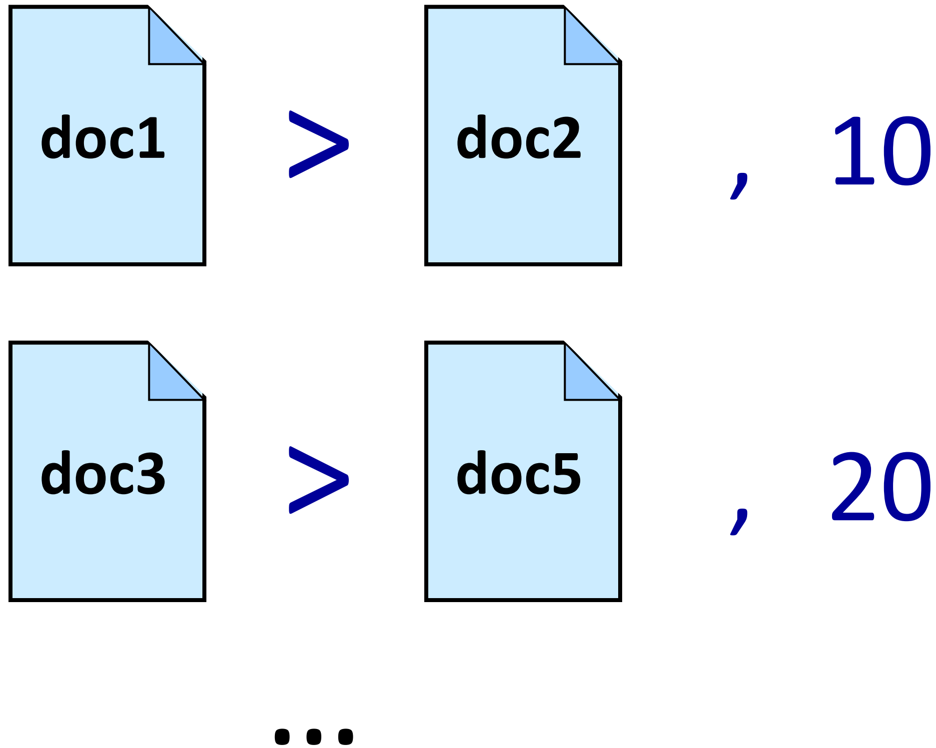
Label Ranking

Subset Ranking

Rank Aggregation

?

Instance Ranking



Label Ranking

doc1

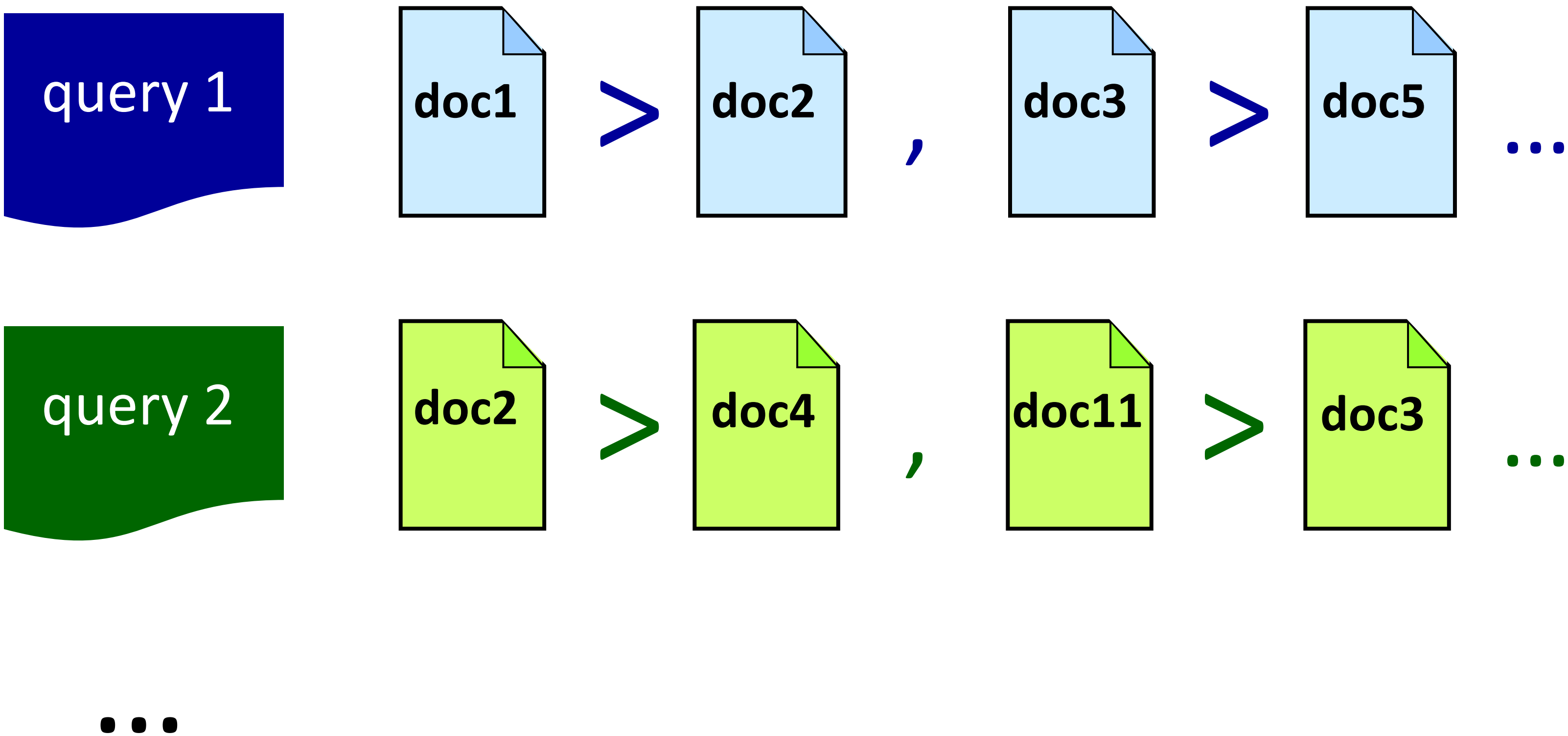
sports > politics
health > money
...

doc2

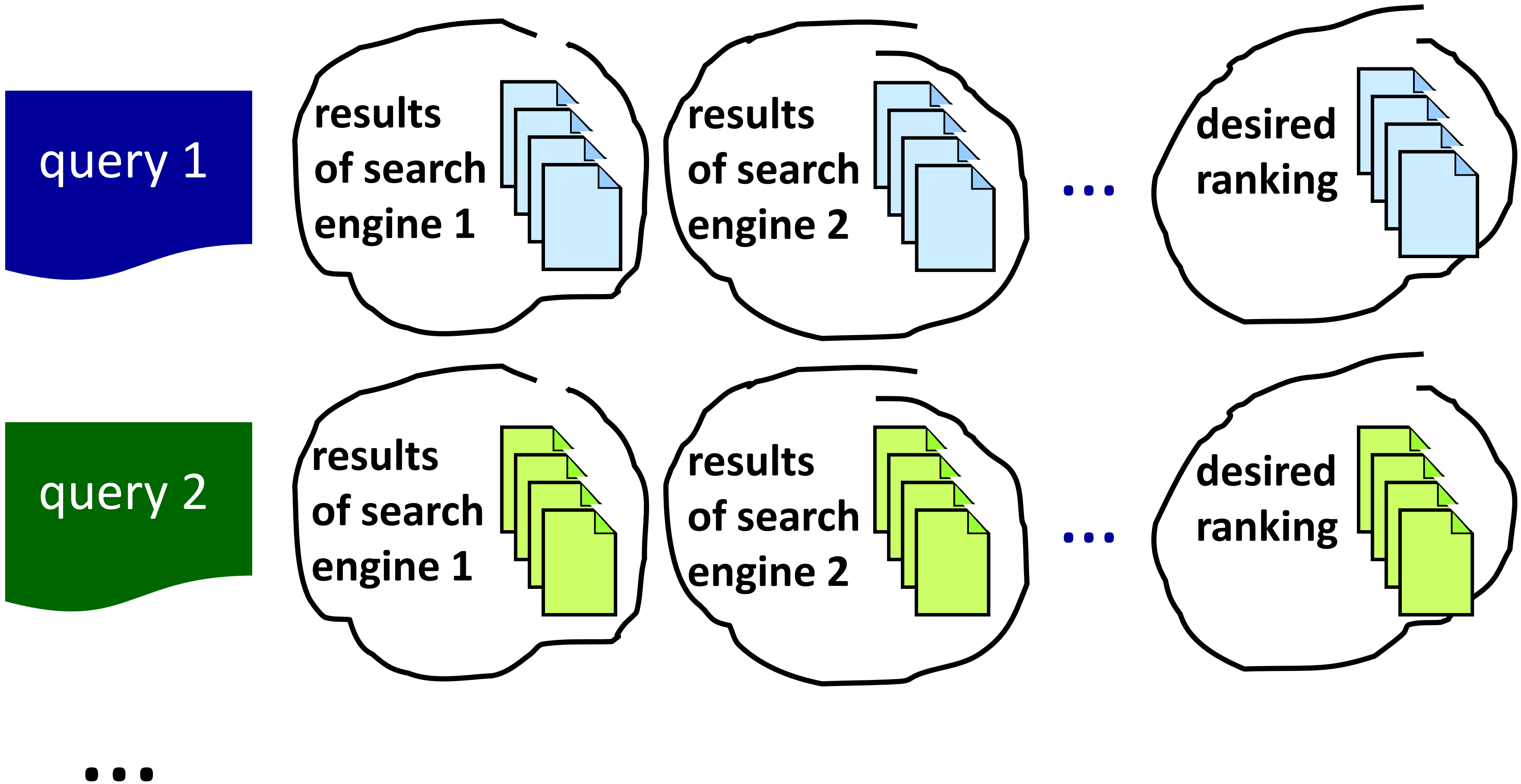
science > sports
money > politics
...

...

Subset Ranking



Rank Aggregation



Types of Ranking Problems

Instance Ranking

Label Ranking

Subset Ranking

Rank Aggregation

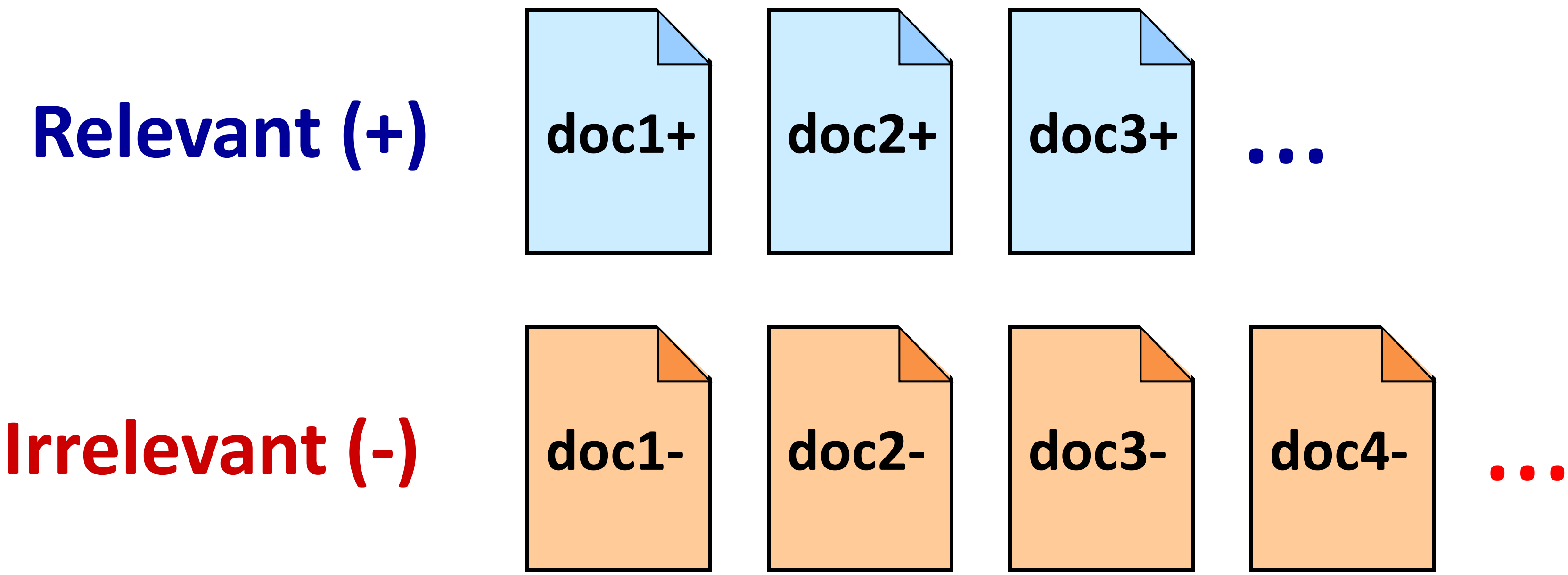
?

This tutorial

Part I
Theory & Algorithms

[for Instance Ranking]

Bipartite Ranking

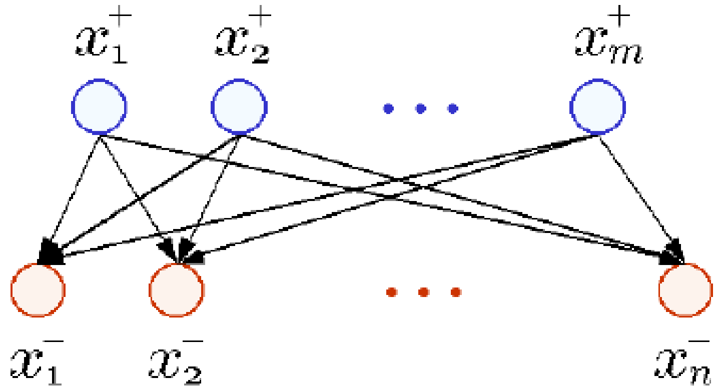


Bipartite Ranking

- Instance space X
- Input: Training sample $S = (S_+, S_-)$:

$$S_+ = (x_1^+, \ldots, x_m^+) \in X^m \quad \text{(positive examples)}$$
$$S_- = (x_1^-, \ldots, x_n^-) \in X^n \quad \text{(negative examples)}$$

- Output: Ranking function $f : X \rightarrow \mathbb{R}$



Bipartite Ranking

- Instance space X
- Input: Training sample $S = (S_+, S_-)$:

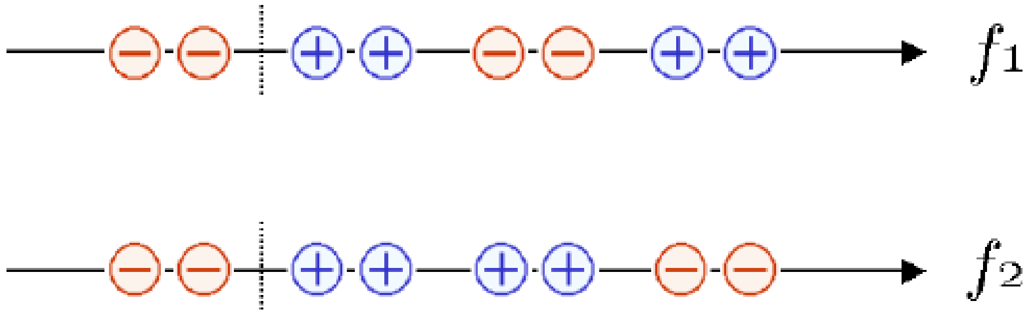
$$S_+ = (x_1^+, \ldots, x_m^+) \in X^m \quad \text{(positive examples)}$$
$$S_- = (x_1^-, \ldots, x_n^-) \in X^n \quad \text{(negative examples)}$$

- Output: Ranking function $f : X \rightarrow \mathbb{R}$

- Expected error: $\mathbf{er}(f) = \mathbf{P}_{(x,x') \sim \mathcal{D}_+ \times \mathcal{D}_-} \left[f(x) < f(x') \right]$
- Empirical error: $\widehat{\mathbf{er}}_S(f) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \mathbf{1}(f(x_i^+) < f(x_j^-))$

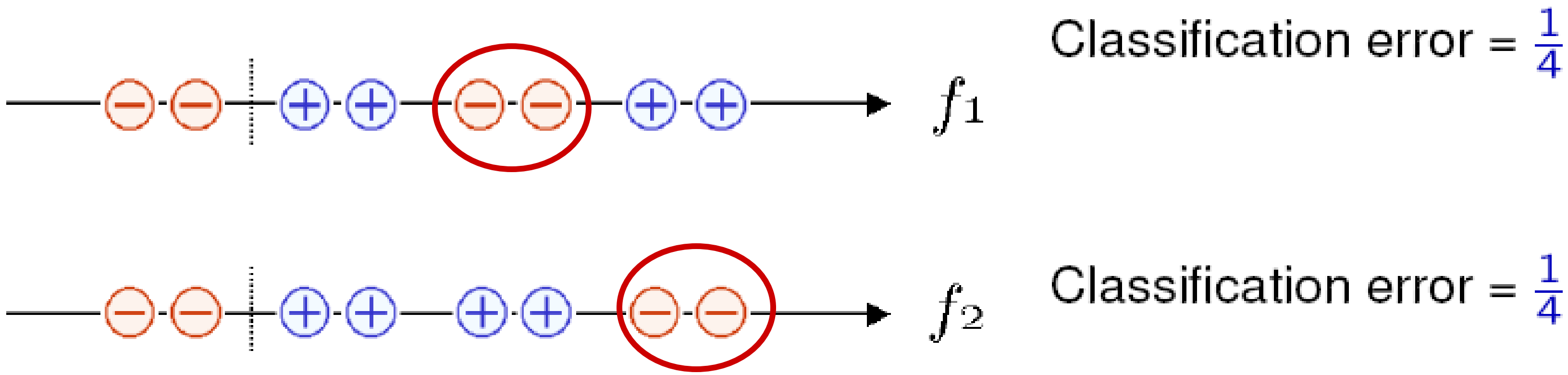
Is Bipartite Ranking Different from Binary Classification?

Example 1



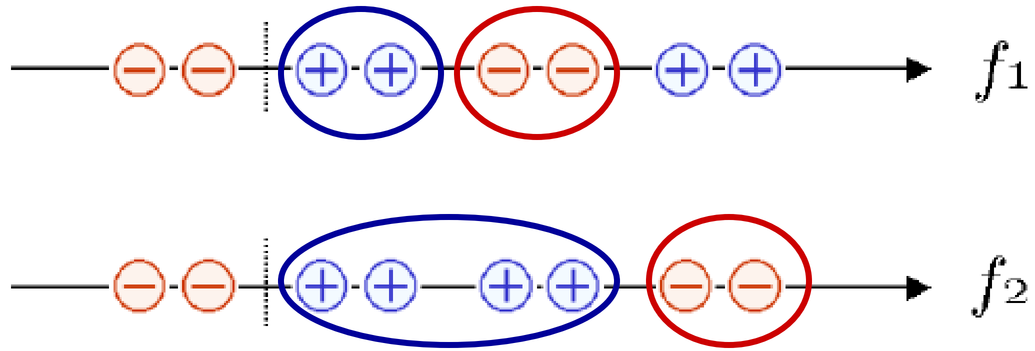
Is Bipartite Ranking Different from Binary Classification?

Example 1



Is Bipartite Ranking Different from Binary Classification?

Example 1



Classification error = $\frac{1}{4}$

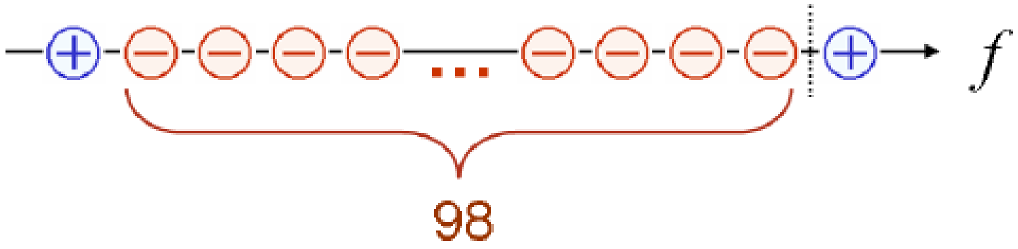
Ranking error = $\frac{1}{4}$

Classification error = $\frac{1}{4}$

Ranking error = $\frac{1}{2}$

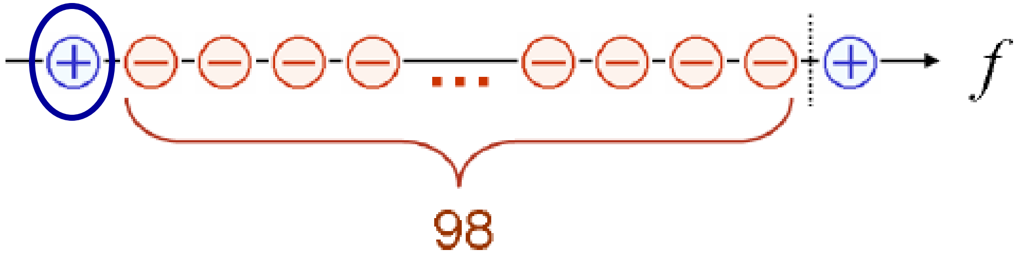
Is Bipartite Ranking Different from Binary Classification?

Example 2



Is Bipartite Ranking Different from Binary Classification?

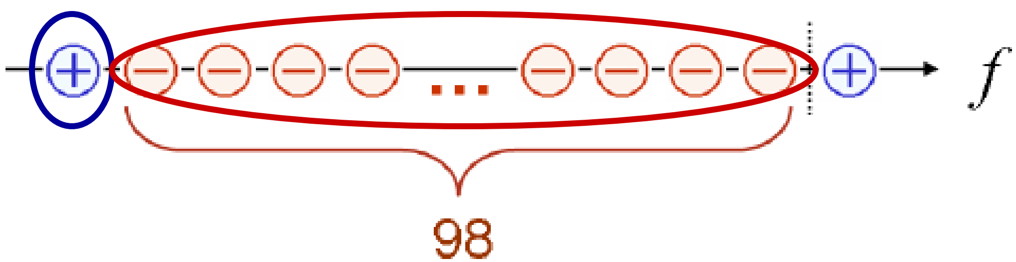
Example 2



Classification error = $\frac{1}{100}$

Is Bipartite Ranking Different from Binary Classification?

Example 2



Classification error = $\frac{1}{100}$

Ranking error = $\frac{1}{2}$

Bipartite Ranking: Basic Algorithmic Framework

Minimize a convex upper bound on the empirical ranking error, possibly with some regularization, over some class of ranking functions:

$$\min_{f \in \mathcal{F}} \left[\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell(f, x_i^+, x_j^-) + \lambda N(f) \right]$$

where

$\ell(f, x_i^+, x_j^-)$	:	convex upper bound on $\mathbf{1}(f(x_i^+) < f(x_j^-))$
$N(f)$	:	regularizer
$\lambda > 0$	:	regularization parameter
$\mathcal{F}$	:	class of ranking functions

Bipartite RankSVM Algorithm

$$\min_{f \in \mathcal{F}_K} \left[\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell_{\text{hinge}}(f, x_i^+, x_j^-) + \frac{\lambda}{2} \|f\|_K^2 \right]$$

$$\ell_{\text{hinge}}(f, x_i^+, x_j^-) = \Big(1 - \big(f(x_i^+) - f(x_j^-)\big)\Big)_+ \quad \big[u_+ = \max(u, 0)\big]$$

$\mathcal{F}_K$ = reproducing kernel Hilbert space (RKHS) with kernel function K

$$N(f) = \frac{\|f\|_K^2}{2}$$

[Herbrich et al, 2000; Joachims, 2002; Rakotomamonjy, 2004]

Bipartite RankSVM Algorithm

Introducing slack variables and taking the Lagrangian dual results in the following convex quadratic program (QP) over mn variables $\{\alpha_{ij} : 1 \leq i \leq m, 1 \leq j \leq n\}$:

$$\min_{\alpha} \left[\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{m} \sum_{l=1}^{n} \alpha_{ij}\alpha_{kl}\, \phi(x_i^+, x_j^-, x_k^+, x_l^-) - \sum_{i=1}^{m} \sum_{j=1}^{n} \alpha_{ij} \right]$$

subject to $0 \leq \alpha_{ij} \leq C \quad \forall\, i, j$

where

$$\phi(x_i^+, x_j^-, x_k^+, x_l^-) = \left(K(x_i^+, x_k^+) - K(x_i^+, x_l^-) - K(x_j^-, x_k^+) + K(x_j^-, x_l^-) \right)$$

$$C = \frac{1}{\lambda mn}$$

Can be solved using a standard QP solver, or more efficient methods (e.g. Chapelle & Keerthi, 2010).

Bipartite RankBoost Algorithm

$$\min_{f \in \mathcal{L}(\mathcal{F}_{\text{base}})} \left[\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell_{\text{exp}}(f, x_i^+, x_j^-) \right]$$

$$\ell_{\text{exp}}(f, x_i^+, x_j^-) = \exp\left(-\left(f(x_i^+) - f(x_j^-)\right)\right)$$

$\mathcal{L}(\mathcal{F}_{\text{base}})$ = linear combinations of functions in some base class $\mathcal{F}_{\text{base}}$

[Freund et al, 2003]

Bipartite RankBoost Algorithm

Input: $(S_+, S_-) \in X^m \times X^n$.

Initialize: $D_1(x_i^+, x_j^-) = \frac{1}{mn}$ for all $i \in \{1, \ldots, m\},\ j \in \{1, \ldots, n\}$.

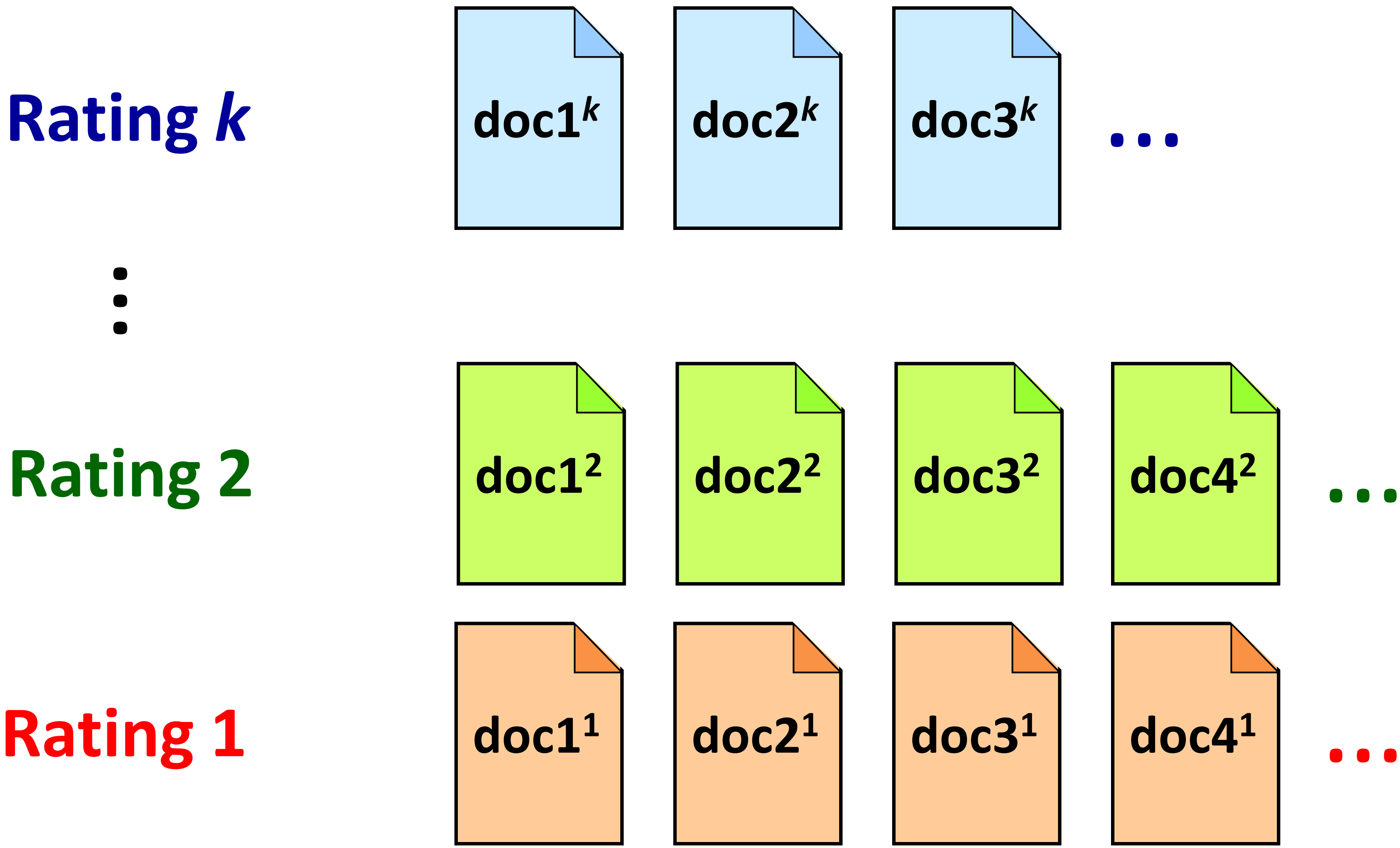
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t; get weak ranker $f_t \in \mathcal{F}_{\text{base}}$.
- Choose $\alpha_t \in \mathbb{R}$.
- Update: $D_{t+1}(x_i^+, x_j^-) = \frac{1}{Z_t} D_t(x_i^+, x_j^-) \exp\Big(-\alpha_t \Big(f_t(x_i^+) - f_t(x_j^-)\Big)\Big)$

  where $Z_t = \sum_{i=1}^{m} \sum_{j=1}^{n} D_t(x_i^+, x_j^-) \exp\Big(-\alpha_t \Big(f_t(x_i^+) - f_t(x_j^-)\Big)\Big)$.

Output final ranking: $f(x) = \sum_{t=1}^{T} \alpha_t f_t(x)$.

k-partite Ranking



k-partite Ranking

- Instance space X
- Input: Training sample $S = (S_1, S_2, \ldots, S_k)$:

$$S_k = (x_1^k, \ldots, x_{n_k}^k) \in X^{n_k} \quad \text{(examples of rating } k\text{)}$$
$$\vdots$$
$$S_2 = (x_1^2, \ldots, x_{n_2}^2) \in X^{n_2} \quad \text{(examples of rating } 2\text{)}$$
$$S_1 = (x_1^1, \ldots, x_{n_1}^1) \in X^{n_1} \quad \text{(examples of rating } 1\text{)}$$

- Output: Ranking function $f : X \rightarrow \mathbb{R}$
- Empirical error:

$$\widehat{\mathbf{er}}_S(f) = \left(\frac{1}{\sum_{1 \le a < b \le k} n_a n_b} \right) \sum_{1 \le a < b \le k} \sum_{i=1}^{n_b} \sum_{j=1}^{n_a} (b - a) \, \mathbf{1}\left(f(x_i^b) < f(x_j^a)\right)$$

k-partite Ranking: Basic Algorithmic Framework

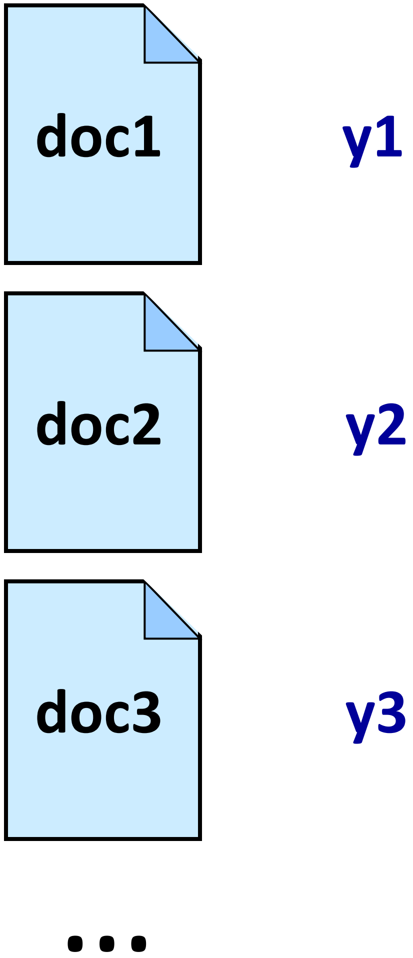
Minimize a convex upper bound on the empirical ranking error, possibly with some regularization, over some class of ranking functions:

$$\min_{f \in \mathcal{F}} \left[\left(\frac{1}{\sum_{1 \le a < b \le k} n_a n_b} \right) \sum_{1 \le a < b \le k} \sum_{i=1}^{n_b} \sum_{j=1}^{n_a} \ell(f, x_i^b, x_j^a, (b-a)) + \lambda N(f) \right]$$

where

$$\begin{aligned} \ell(f, x_i^b, x_j^a, (b-a)) &: \text{ convex upper bound on } (b-a)\,\mathbf{1}(f(x_i^b) < f(x_j^a)) \\ N(f) &: \text{ regularizer} \\ \lambda > 0 &: \text{ regularization parameter} \\ \mathcal{F} &: \text{ class of ranking functions} \end{aligned}$$

Ranking with Real-Valued Labels



Ranking with Real-Valued Labels

- Instance space X
- Real-valued labels $Y = \mathbb{R}$
- Input: Training sample $S = ((x_1, y_1), \ldots, (x_m, y_m)) \in (X \times \mathbb{R})^m$
- Output: Ranking function $f : X \to \mathbb{R}$
- Empirical error:

$$\widehat{\mathbf{er}}_S(f) = \frac{1}{\binom{m}{2}} \sum_{1 \le i < j \le m} |y_i - y_j| \, \mathbf{1}\left((y_i - y_j)(f(x_i) - f(x_j)) < 0\right)$$

Ranking with Real-Valued Labels: Basic Algorithmic Framework

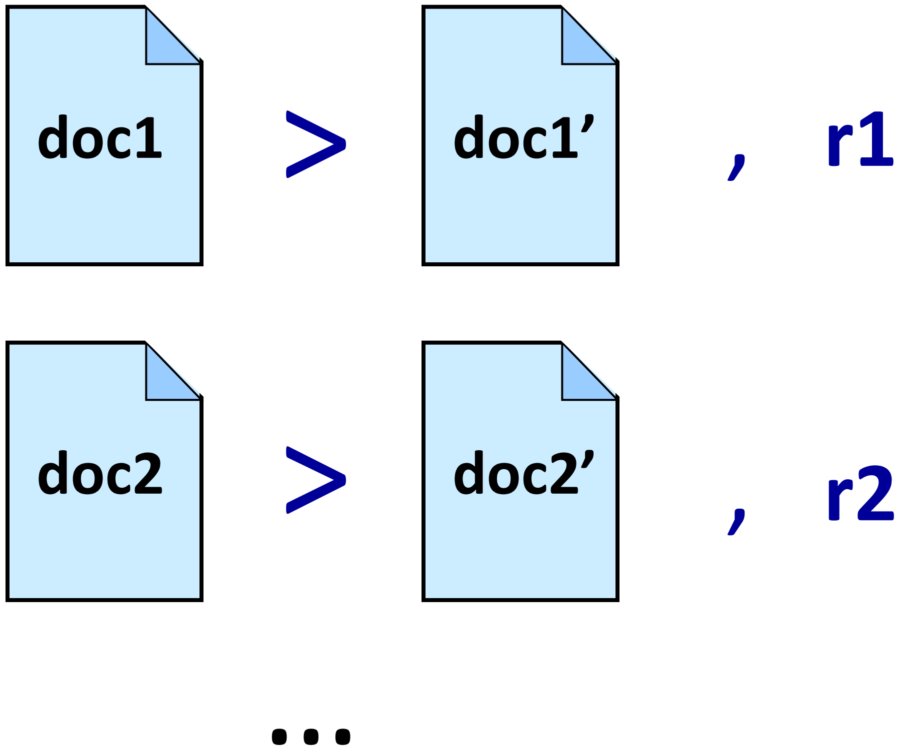
Minimize a convex upper bound on the empirical ranking error, possibly with some regularization, over some class of ranking functions:

$$\min_{f \in \mathcal{F}} \left[\frac{1}{\binom{m}{2}} \sum_{1 \leq i < j \leq m} \ell(f, (x_i, y_i), (x_j, y_j)) + \lambda N(f) \right]$$

where

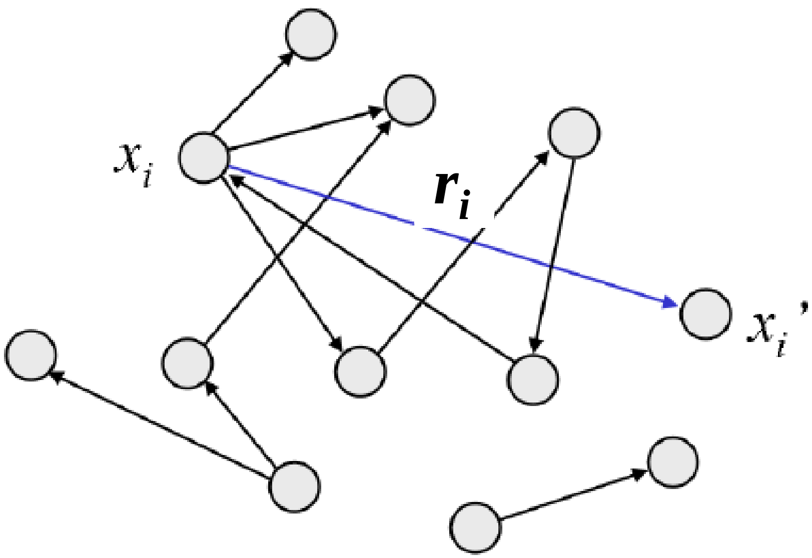
$$\begin{aligned}
\ell(f, (x_i, y_i), (x_j, y_j)) &: \text{ convex upper bound on } \\
& \quad |y_i - y_j| \, \mathbf{1}\left((y_i - y_j)(f(x_i) - f(x_j)) < 0 \right) \\
N(f) &: \text{ regularizer} \\
\lambda > 0 &: \text{ regularization parameter} \\
\mathcal{F} &: \text{ class of ranking functions}
\end{aligned}$$

General Instance Ranking



General Instance Ranking

- Instance space X
- Input: Training sample $S = ((x_1, x'_1, r_1), \ldots, (x_m, x'_m, r_m)) \in (X^2 \times \mathbb{R}_+)^m$
- Output: Ranking function $f : X \rightarrow \mathbb{R}$



General Instance Ranking

- Instance space X
- Input: Training sample $S = ((x_1, x'_1, r_1), \ldots, (x_m, x'_m, r_m)) \in (X^2 \times \mathbb{R}_+)^m$
- Output: Ranking function $f : X \rightarrow \mathbb{R}$
- Empirical error: $\widehat{\mathbf{er}}_S(f) = \frac{1}{m} \sum_{i=1}^{m} r_i \, \mathbf{1}(f(x_i) < f(x'_i))$

General Instance Ranking: Basic Algorithmic Framework

Minimize a convex upper bound on the empirical ranking error, possibly with some regularization, over some class of ranking functions:

$$\min_{f \in \mathcal{F}} \left[\frac{1}{m} \sum_{i=1}^{m} \ell(f, x_i, x_i', r_i) + \lambda N(f) \right]$$

where

$\ell(f, x_i, x_i', r_i)$	:	convex upper bound on $r_i \mathbf{1}(f(x_i) < f(x_i'))$
$N(f)$	:	regularizer
$\lambda > 0$	:	regularization parameter
$\mathcal{F}$	:	class of ranking functions

General RankSVM Algorithm

$$\min_{f \in \mathcal{F}_K} \left[\frac{1}{m} \sum_{i=1}^{m} \ell_{\text{hinge}}(f, x_i, x_i', r_i) + \frac{\lambda}{2} \|f\|_K^2 \right]$$

$$\ell_{\text{hinge}}(f, x_i, x_i', r_i) = \Big(r_i - \big(f(x_i) - f(x_i') \big) \Big)_+ \quad \left[u_+ = \max(u, 0) \right]$$

$\mathcal{F}_K$ = reproducing kernel Hilbert space (RKHS) with kernel function K

$$N(f) = \frac{\|f\|_K^2}{2}$$

[Herbrich et al, 2000; Joachims, 2002]

General RankBoost Algorithm

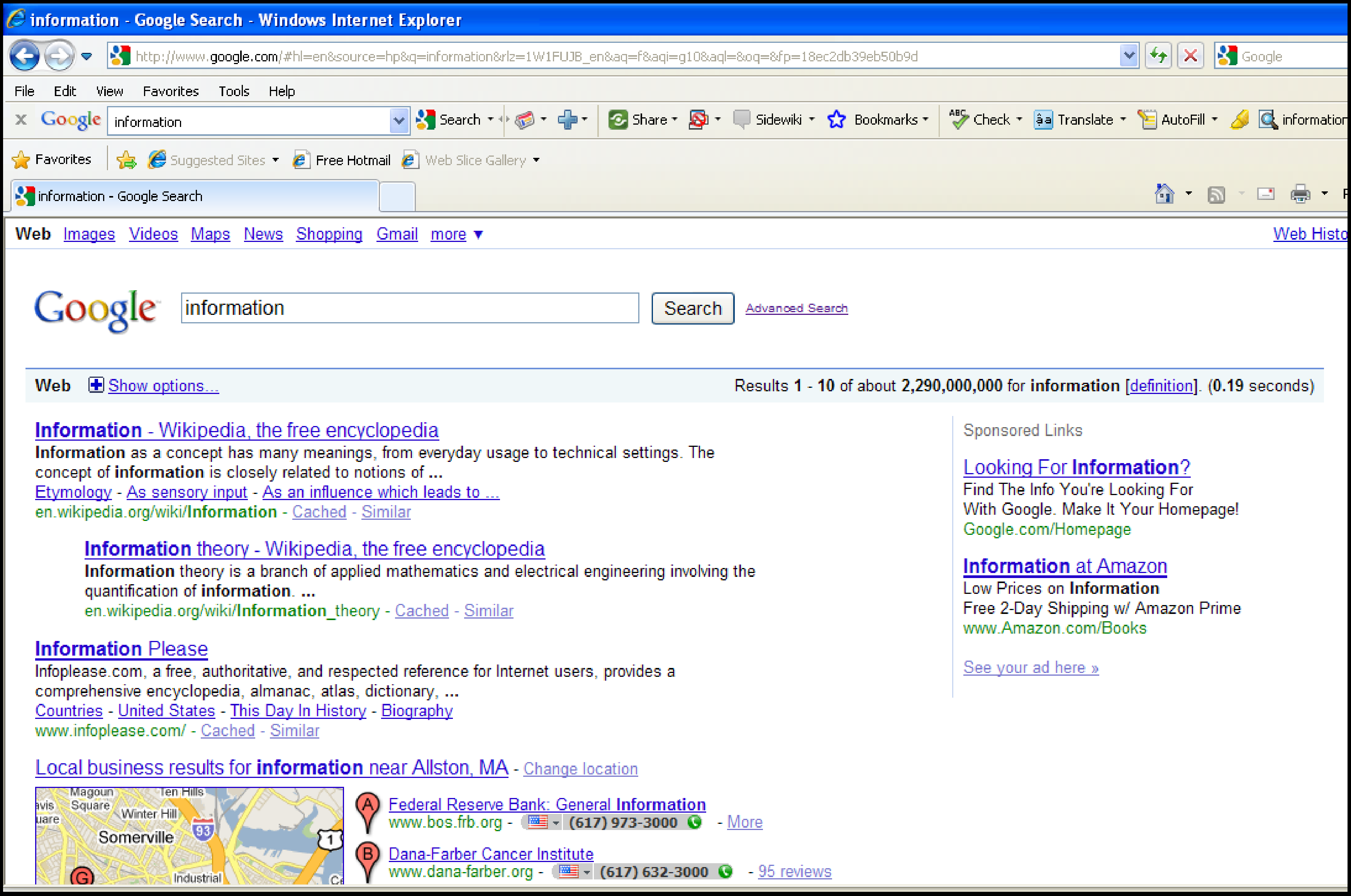
$$\min_{f \in \mathcal{L}(\mathcal{F}_{\text{base}})} \left[\frac{1}{m} \sum_{i=1}^{m} \ell_{\text{exp}}(f, x_i, x_i', r_i) \right]$$

$$\ell_{\text{exp}}(f, x_i, x_i', r_i) = r_i \exp\left(- \left(f(x_i) - f(x_i') \right) \right)$$

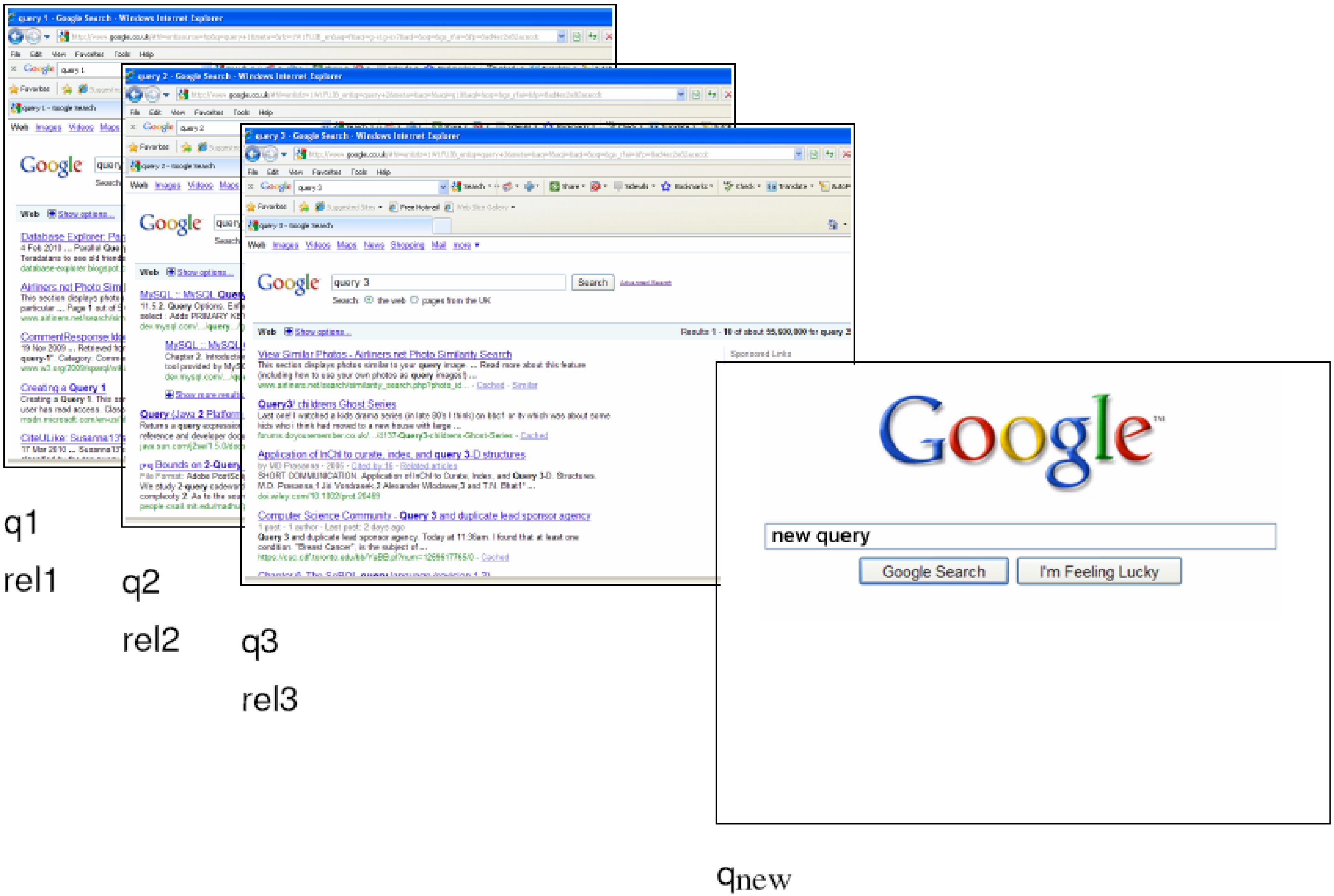
$\mathcal{L}(\mathcal{F}_{\text{base}})$ = linear combinations of functions in some base class $\mathcal{F}_{\text{base}}$

[Freund et al, 2003]

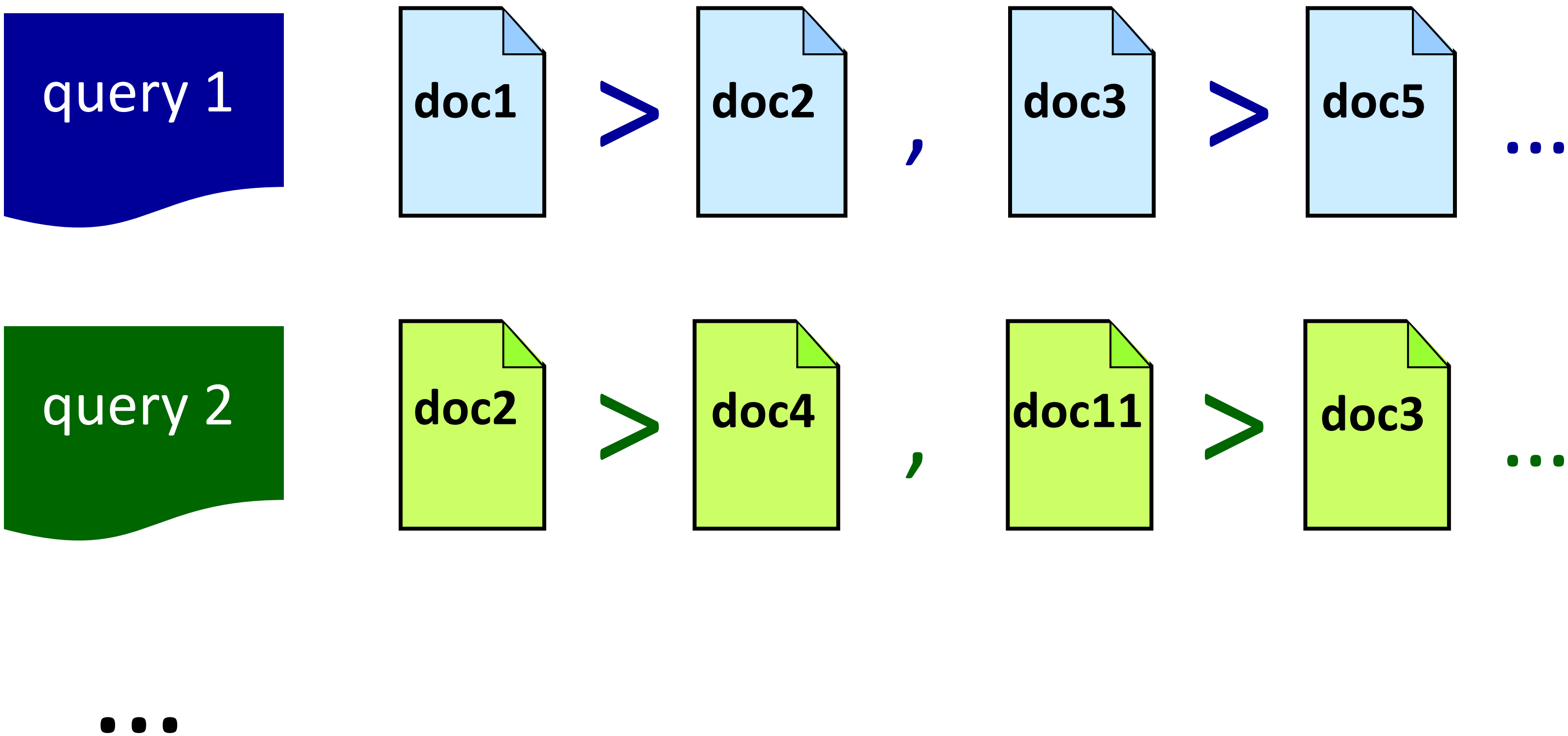
Application to Information Retrieval (IR)



Learning to Rank in IR



General Subset Ranking



General Subset Ranking

- Query space Q
- Document space D
- Query-document feature mapping $\phi : Q \times D \rightarrow \mathbb{R}^d$
- Input: Training sample $S = (S^1, \ldots, S^m)$:

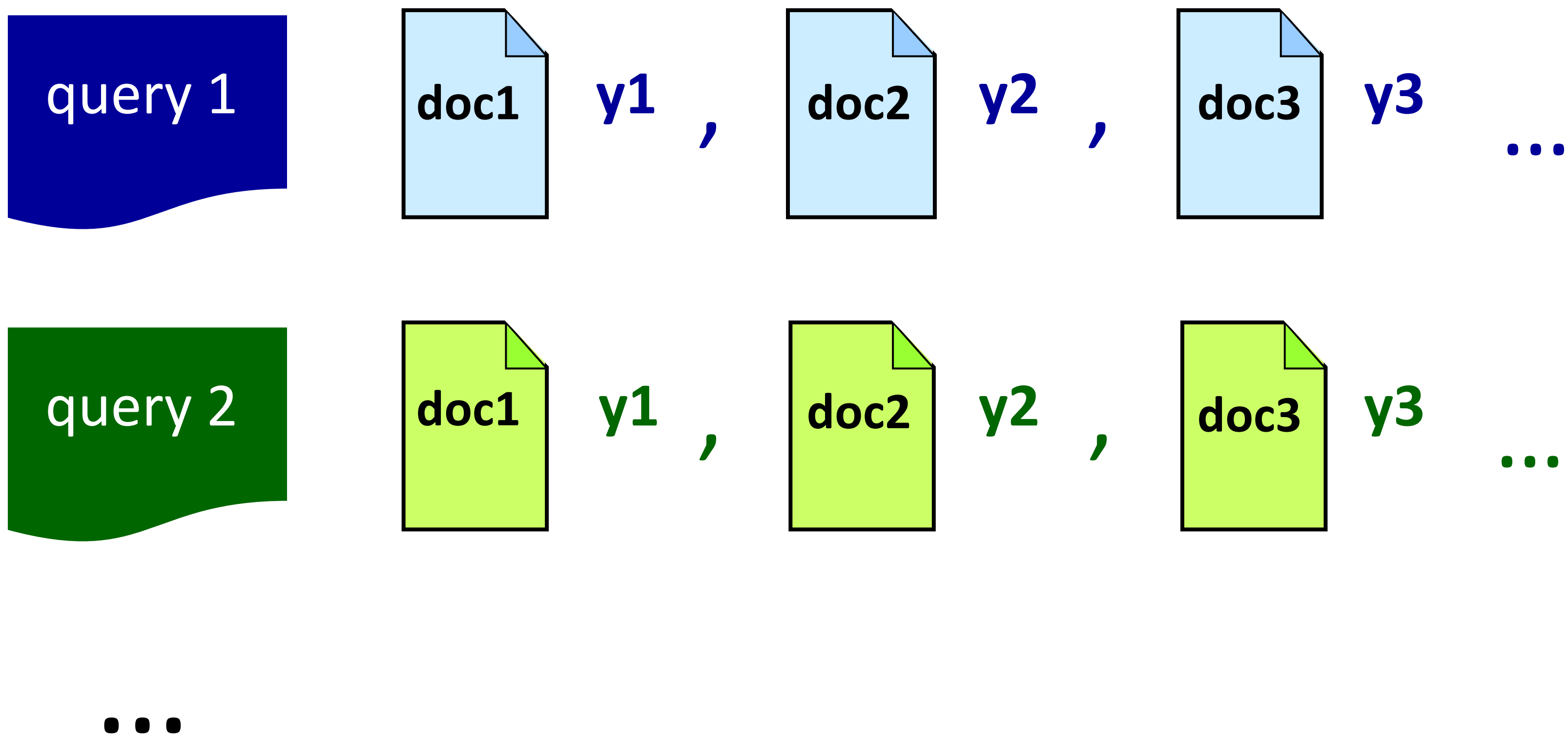
  $$S^i = \left((\phi^i_1, {\phi^i_1}'), \ldots, (\phi^i_{n_i}, {\phi^i_{n_i}}')\right) \in (\mathbb{R}^d \times \mathbb{R}^d)^{n_i}$$

  where

  $$\phi^i_j = \phi(q^i, d^i_j), \quad {\phi^i_j}' = \phi(q^i, {d^i_j}')$$

- Output: Ranking function $f : \mathbb{R}^d \rightarrow \mathbb{R}$

Subset Ranking with Real-Valued Relevance Labels



Subset Ranking with Real-Valued Relevance Labels

- Query space Q
- Document space D
- Query-document feature mapping $\phi : Q \times D \to \mathbb{R}^d$
- Input: Training sample $S = (S^1, \ldots, S^m)$:

  $$S^i = \left((\phi_1^i, y_1^i), \ldots, (\phi_{n_i}^i, y_{n_i}^i)\right) \in (\mathbb{R}^d \times \mathbb{R})^{n_i}$$

  where

  $$\phi_j^i = \phi(q^i, d_j^i), \quad y_j^i = \text{relevance of } d_j^i \text{ to } q^i$$

- Output: Ranking function $f : \mathbb{R}^d \to \mathbb{R}$

RankSVM Applied to IR/Subset Ranking

Standard RankSVM

$$\min_{f \in \mathcal{F}_K} \left[\left(\frac{1}{\sum_{i=1}^{m} \binom{n_i}{2}} \right) \sum_{i=1}^{m} \sum_{1 \le j < k \le n_i} \ell_{\text{hinge}} \left(f, (\phi_j^i, y_j^i), (\phi_k^i, y_k^i) \right) + \frac{\lambda}{2} \|f\|_K^2 \right]$$

$$\ell_{\text{hinge}} \left(f, (\phi_j^i, y_j^i), (\phi_k^i, y_k^i) \right) = \left(1 - \left(\text{sign}(y_j^i - y_k^i) \cdot (f(\phi_j^i) - f(\phi_k^i)) \right) \right)_+ ,$$

convex upper bound on

$$\mathbf{1}\left((y_j^i - y_k^i)(f(\phi_j^i) - f(\phi_k^i)) < 0 \right)$$

[Joachims, 2002]

RankSVM Applied to IR/Subset Ranking

RankSVM with Query Normalization & Relevance Weighting

$$\min_{f \in \mathcal{F}_K} \left[\frac{1}{m} \sum_{i=1}^{m} \left[\frac{1}{\binom{n_i}{2}} \sum_{1 \le j < k \le n_i} \ell^{\text{rel}}_{\text{hinge}} \left(f, (\phi^i_j, y^i_j), (\phi^i_k, y^i_k) \right) + \frac{\lambda}{2} \|f\|^2_K \right] \right]$$

$$\ell^{\text{rel}}_{\text{hinge}} \left(f, (\phi^i_j, y^i_j), (\phi^i_k, y^i_k) \right) = \left(|y^i_j - y^i_k| - \left(\text{sign}(y^i_j - y^i_k) \cdot (f(\phi^i_j) - f(\phi^i_k)) \right) \right)_+ ,$$

convex upper bound on

$$|y^i_j - y^i_k| \, \mathbf{1} \left((y^i_j - y^i_k)(f(\phi^i_j) - f(\phi^i_k)) < 0 \right)$$

[Agarwal & Collins, 2010; also Cao et al, 2006]

Ranking Performance Measures in IR

Mean Average Precision (MAP)

Binary Labels: $y_j \in \{0, 1\}$

$$\text{MAP}_S(f) \;=\; \frac{1}{m}\sum_{i=1}^{m}\left[\frac{1}{\left|\{j : y_j^i = 1\}\right|}\sum_{j:y_j^i=1}\text{prec}^i_{r_j^i}(f)\right]$$

$$r_j^i \;=\; \text{rank of document } d_j^i \text{ for query } q^i$$

$$\text{prec}^i_r(f) \;=\; \text{fraction of positives in top } r \text{ documents for query } q^i$$

Ranking Performance Measures in IR

Normalized Discounted Cumulative Gain (NDCG)

General Real-Valued Labels: $y_j \in \mathbb{R}$

$$\text{NDCG}_S(f) = \frac{1}{m} \sum_{i=1}^{m} \left[\frac{1}{Z_i} \sum_{r=1}^{n_i} \frac{2^{y^i_{\pi^i_r}} - 1}{\log_2(r+1)} \right]$$

$$\pi^i_r = \text{index of document ranked at position } r \text{ for query } q^i$$

$$Z_i = \text{normalization constant}$$

$$\text{NDCG@}k_S(f) = \frac{1}{m} \sum_{i=1}^{m} \left[\frac{1}{Z_i} \sum_{r=1}^{k} \frac{2^{y^i_{\pi^i_r}} - 1}{\log_2(r+1)} \right]$$

Ranking Algorithms for Optimizing MAP/NDCG

- SVMMAP [Yue et al. 2007]
- SVMNDCG [Chapelle et al. 2007]
- LambdaRank [Burges et al. 2007]
- AdaRank [Xu & Li 2007]
- Regression-based algorithm [Cossock & Zhang 2008]
- SoftRank [Taylor et al. 2008]
- SmoothRank [Chapelle & Wu 2010]