排序学习与语义匹配
Learning to Rank and Semantic Match

徐 君

上节课内容回顾

- 互联网搜索引擎核心问题：网页排序
  - 文档处理与倒排索引
  - 布尔模型（非排序）
  - 相关性排序
    - VSM、BM25、LMIR
    - 排序评价
  - 搜索结果多样化
    - MMR

提纲

- 排序学习
  - 问题定义
  - 排序学习算法
- 语义匹配
  - 问题定义
  - 匹配学习算法
- 总结

为何要使用机器学习进行排序?

- 可以用来进行网页排序的信息多种多样
  - 文本相关性
    - 如VSM, LM, BM25的分值
    - 网页多个域：标题、文本内容、锚文本、clicked-query
  - 临近度(Proximity)
    - data mining => … **mining** the big **data** …
  - 基于链接的网页重要性信息(如PageRank分值)
  - ULR深度
    - 顶级目录网页 vs 叶子网页
  - 是否是垃圾网页
  - 域名重要度(.com, .org; host-level PageRank)
  - ……

传统排序模型的缺点

- 只能够考虑有限的排序因素
  - 词频(tf)
  - 词的重要性(idf)
  - 文档长度(document length)
- 传统考虑多种排序因素的方式
  - 1. 将各个因素归一化: 如以0为中心，方差为1
  - 2. 设计因素聚合函数: 通常为带权线性和
  - 3. 决定权重: 通常手动设置或者暴力调试

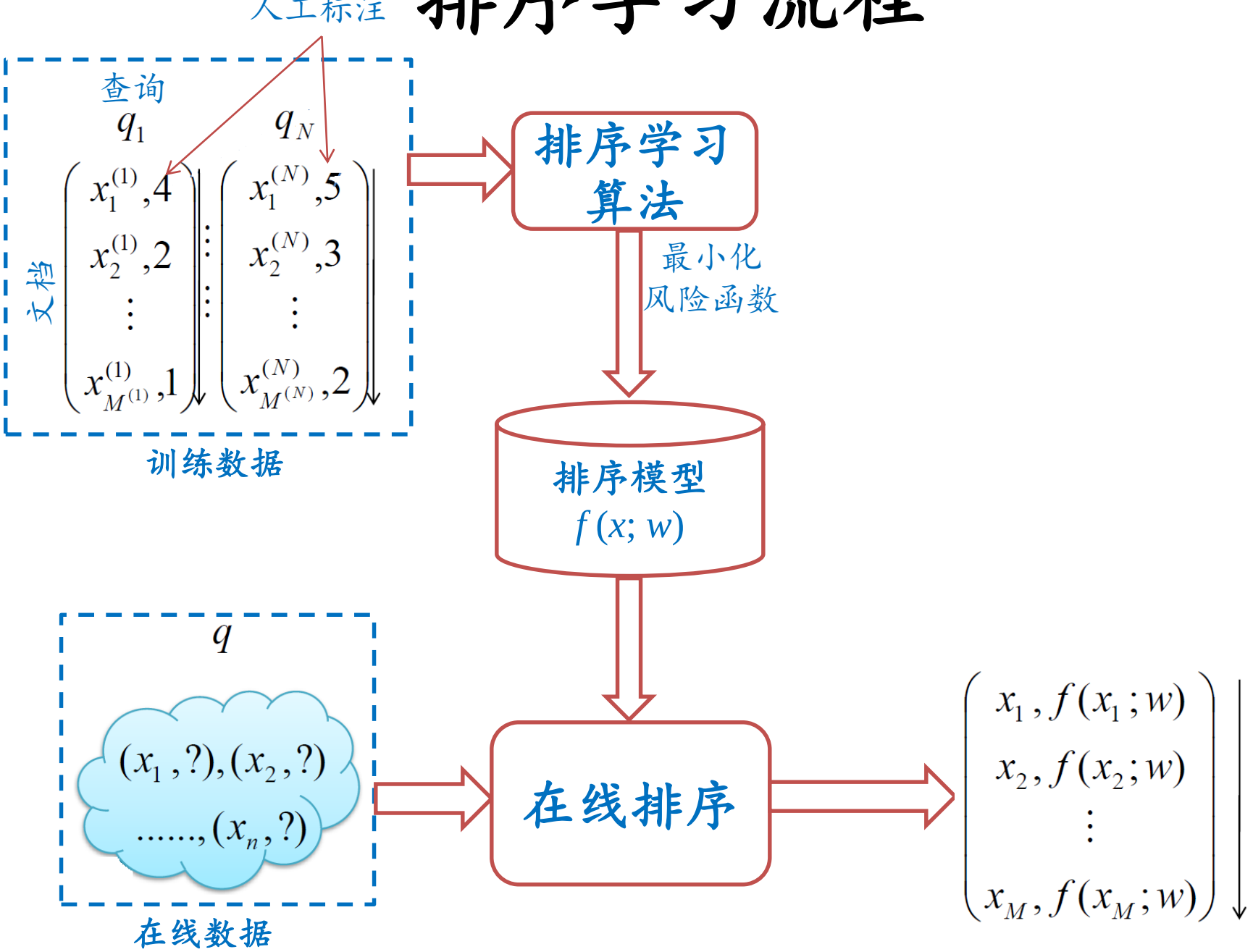
希望融合多个影响排序的因素

- 启发式规则：根据经验，构造决策流程给文档的排序（如给每一个文档打分）
  - 优点：容易理解
  - 缺点：对开发人员要求高，流程构造困难，排序规则更新困难
- 排序学习(learning to rank)：从(标注)数据中学习排序模型
  - 优点：数据驱动，排序模型构造相对容易
  - 缺点：排序模型相对不容易理解

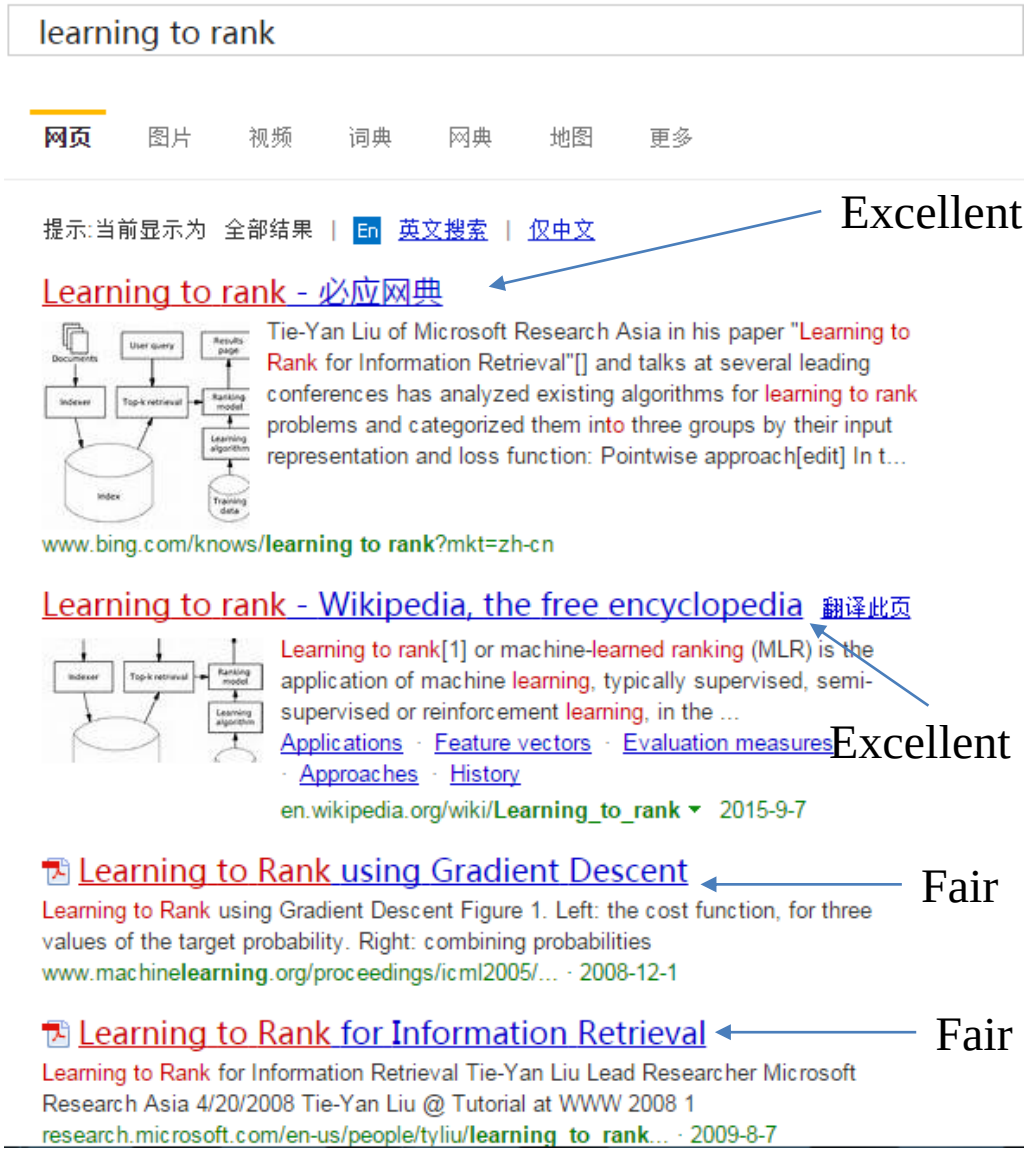
排序学习已经广泛被业界采用

- 互联网搜索引擎
  - Bing
  - Yahoo!
  - Baidu
  - Sogou
- 企业搜索引擎
  - Microsoft SharePoint Search
  - 华为GTS搜索

排序学习流程


人工标注
查询
q_1
q_N
文档
$x_1^{(1)},4$
$x_2^{(1)},2$
$x_{M^{(1)}}^{(1)},1$
$x_1^{(N)},5$
$x_2^{(N)},3$
$x_{M^{(N)}}^{(N)},2$
训练数据
排序学习算法
最小化风险函数
排序模型
$f(x;w)$
q
$(x_1,?),(x_2,?)$
$\ldots\ldots,(x_n,?)$
在线数据
在线排序
$x_1, f(x_1;w)$
$x_2, f(x_2;w)$
$x_M, f(x_M;w)$

排序学习问题定义——标注数据集合



- 给定训练数据集合$D = \{(q_i, \mathbf{d}_i, \mathbf{y}_i)\}_{i=1}^{N}$，其中$q_i$为查询，$\mathbf{d}_i$为候选文档集合，$\mathbf{y}_i$为对候选文档集合中文档的标注集合
- $\mathbf{d}_i = \{d_{i1}, d_{i2}, \cdots, d_{iM}\}$
  - 通过查询从索引中搜索出来的文档（例如通过BM25）
- $\mathbf{y}_i = \{y_{i1}, y_{i2}, \cdots, y_{iM}\}$
  - 人工标注，通常标注为

  2级：Y={相关，不相关}

  3级：Y={相关，部分相关，不相关}

  或5级：Y={Perfect, Excellent, Good, Fair, Bad}

排序函数

- 排序函数逐个对每查询-文档对进行打分 $f(q,d)$，多个文档按照此打分从大到小进行排序
  - 对每一个q-d对抽取特征
  - 对每一个文档独立进行打分
- 线性排序函数$f(q,d) = \langle \mathbf{w}, \boldsymbol{\phi}(q,d) \rangle$
  - $\mathbf{w}$：权重向量（模型参数）
  - $\boldsymbol{\phi}(q,d)$：人工定义的特征
    - 查询相关特征(dynamic features)，反映查询与文档匹配程度
    - 查询无关特征(static features)，反映文档重要性

动态特征(Dynamic Features)

- 反映查询q与文档d的匹配程度
- 单个词匹配，例如
  - VSM作用于title/body/anchor/URL等
  - BM25作用于title/body/anchor/URL等
  - LMIR作用于title/body/anchor/URL等
- 临近度匹配(可作用于title/body/anchor/URL等)，例如
  - Bigram: $\sum_{w_i w_j \in \mathbf{q}} tf\left(w_i w_j, \mathbf{d}\right)$
  - Unordered biterm: $\sum_{(w_i, w_j) \in \mathbf{q}} tf_{t-window}\left((w_i, w_j), \mathbf{d}\right)$
  - Span: d中出现所有q中词的最小窗口大小
  - ……

静态特征(Static Features)

- 反映文档*d*的特征(与查询无关)，例如
  - PageRank值
  - 文档长度
  - URL深度
  - 文档是否为Wikipedia网页
  - 文档是否为垃圾网页

  ……

提纲

- 排序学习
  - 问题定义
  - 排序学习算法
- 语义匹配
  - 问题定义
  - 匹配学习算法
- 总结

排序学习算法分类

Categorization of the Algorithms

Category	Algorithms
Pointwise Approach	**Regression**: Least Square Retrieval Function (TOIS 1989), Regression Tree for Ordinal Class Prediction (Fundamenta Informaticae, 2000), Subset Ranking using Regression (COLT 2006), … **Classification**: Discriminative model for IR (SIGIR 2004), McRank (NIPS 2007), … **Ordinal regression**: Pranking (NIPS 2002), OAP-BPM (EMCL 2003), Ranking with Large Margin Principles (NIPS 2002), Constraint Ordinal Regression (ICML 2005), …
Pairwise Approach	Learning to Retrieve Information (SCC 1995), Learning to Order Things (NIPS 1998), Ranking SVM (ICANN 1999), RankBoost (JMLR 2003), LDM (SIGIR 2005), RankNet (ICML 2005), Frank (SIGIR 2007), MHR(SIGIR 2007), GBRank (SIGIR 2007), QBRank (NIPS 2007), MPRank (ICML 2007), IRSVM (SIGIR 2006), …
Listwise Approach	**Listwise loss minimization**: RankCosine (IP&M 2008), ListNet (ICML 2007), ListMLE (ICML 2008), … **Direct optimization of IR measure**: LambdaRank (NIPS 2006), AdaRank (SIGIR 2007), SVM-MAP (SIGIR 2007), SoftRank (LR4IR 2007), GPRank (LR4IR 2007), CCA (SIGIR 2007), …

Tie-Yan Liu: WWW 2009 Tutorial

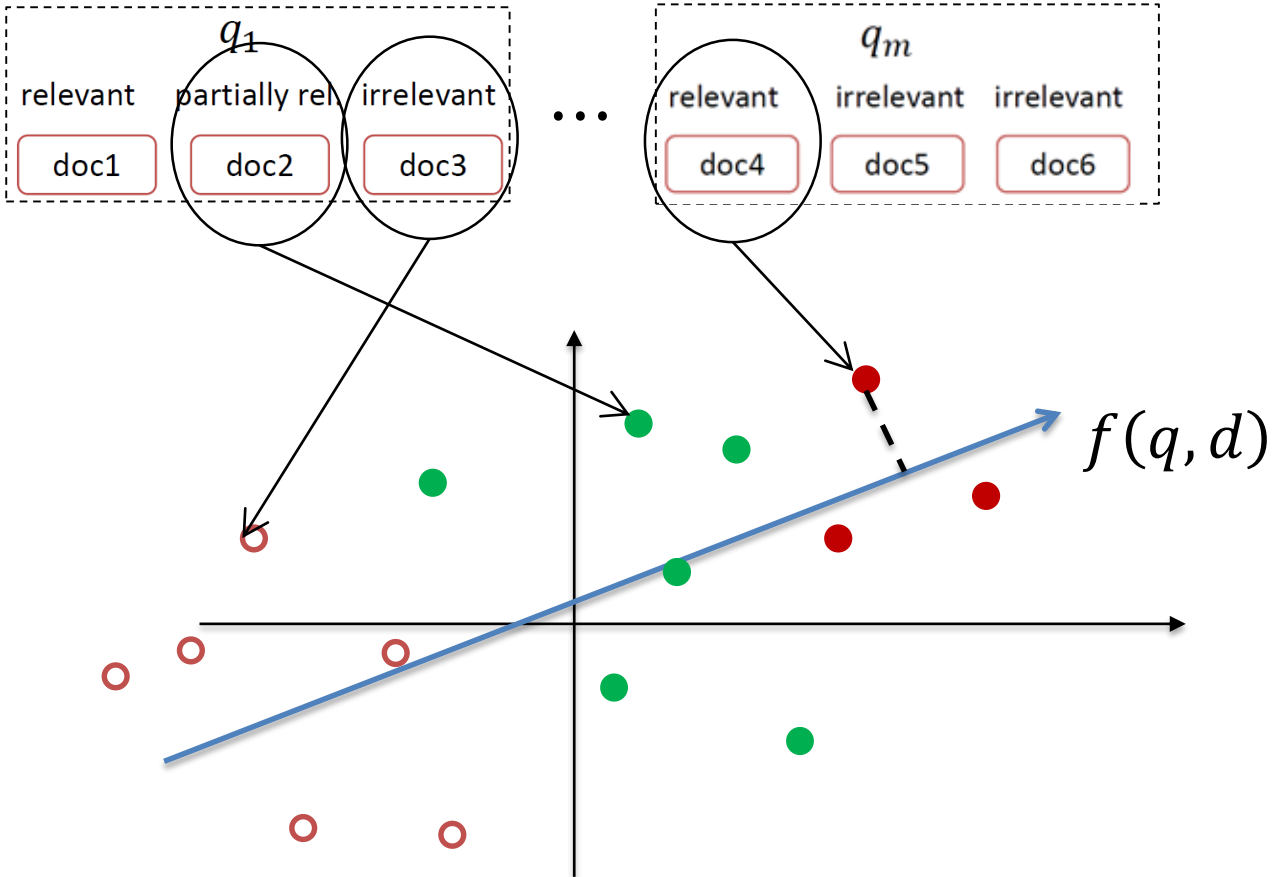
Point-wise Learning to Rank

Point-wise排序学习：直接应用分类/回归算法

- Point-wise核心想法：直接应用机器学习中的分类/回归算法
  - 将查询-网页对形成的特征向量看成空间中的点
  - 将相关性标签看成类别标签(或则回归值)
  - 直接应用分类(或者回归)算法，训练得到模型
  - 在线排序：给定一个查询-网页对，首先抽取特征，然后应用分类(回归)函数，将函数返回值直接作为其相关性打分

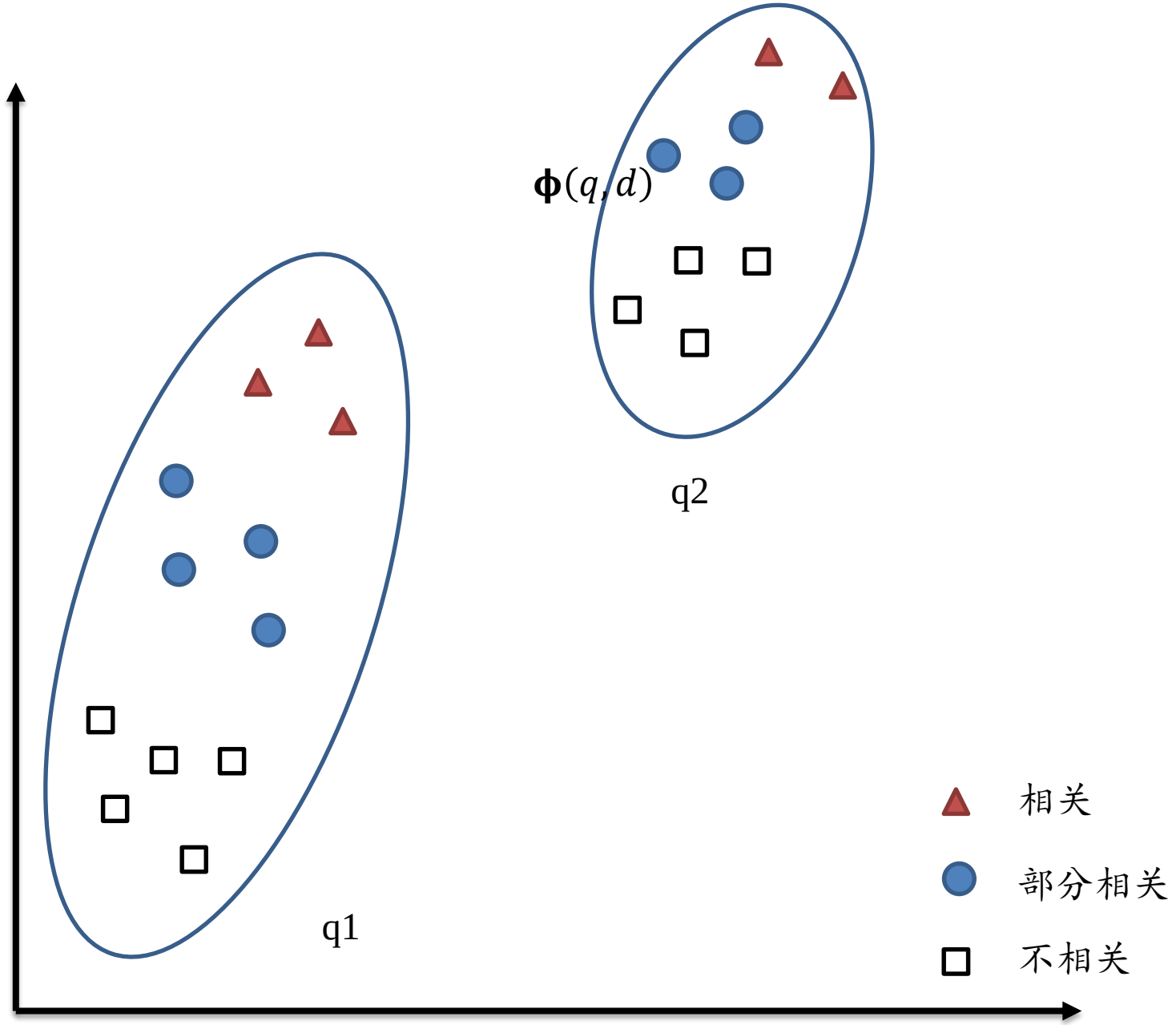
Pointwise 排序学习

- 排序问题→ 查询-文档作为训练样本，形式化为分类或者回归问题[R. Nallapati, SIGIR ’04]



直接应用分类/回归算法的不足之处

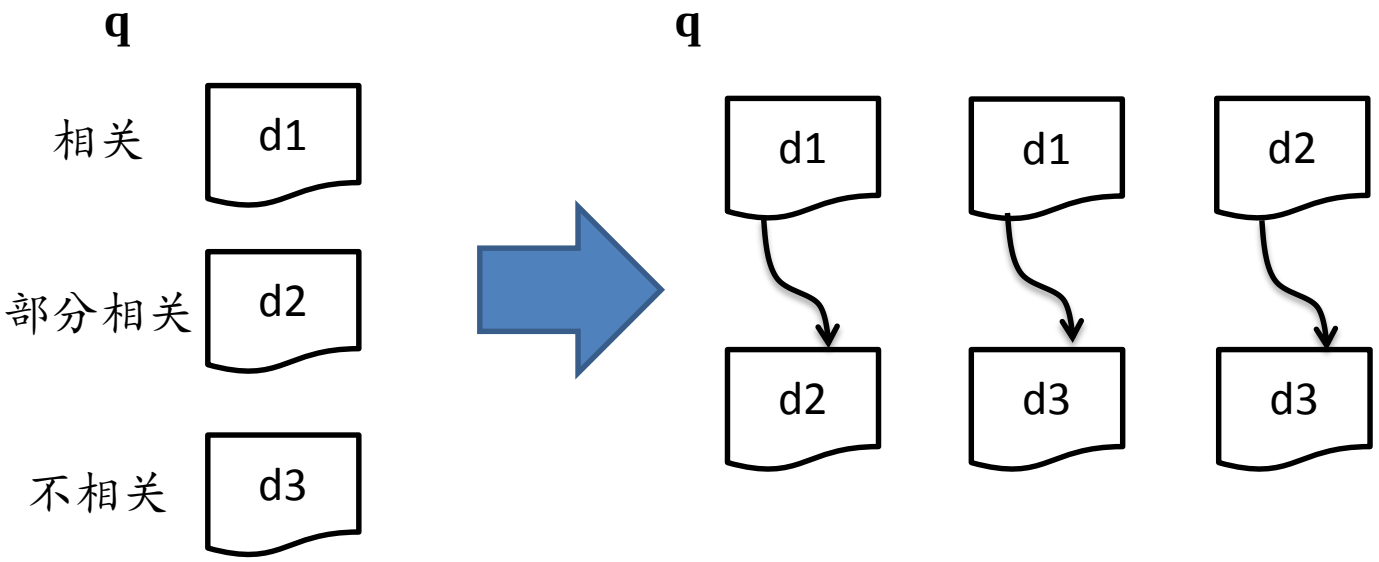
- 未考虑信息检索的特殊性
  - 检索只考虑文档的相对分值进行排序，不在乎其绝对打分大小
  - 文档的分数值比较只在查询内部进行，跨查询的分值比较没有意义
  - 不同的查询长度不同，不同的查询词的文档频率等也不一样，因此对于不同查询，其标注为同级别的文档，在特征值的分布上也会有较大差距
    - 查询q1：不相关文档0~10分，相关文档10~50分
    - 查询q2：不相关文档0~1分，相关文档1~3分
    - 难以有一个分类模型，同时把q1和q2的文档都正确排序


$\boldsymbol{\phi}(q,d)$
q2
q1
相关
部分相关
不相关

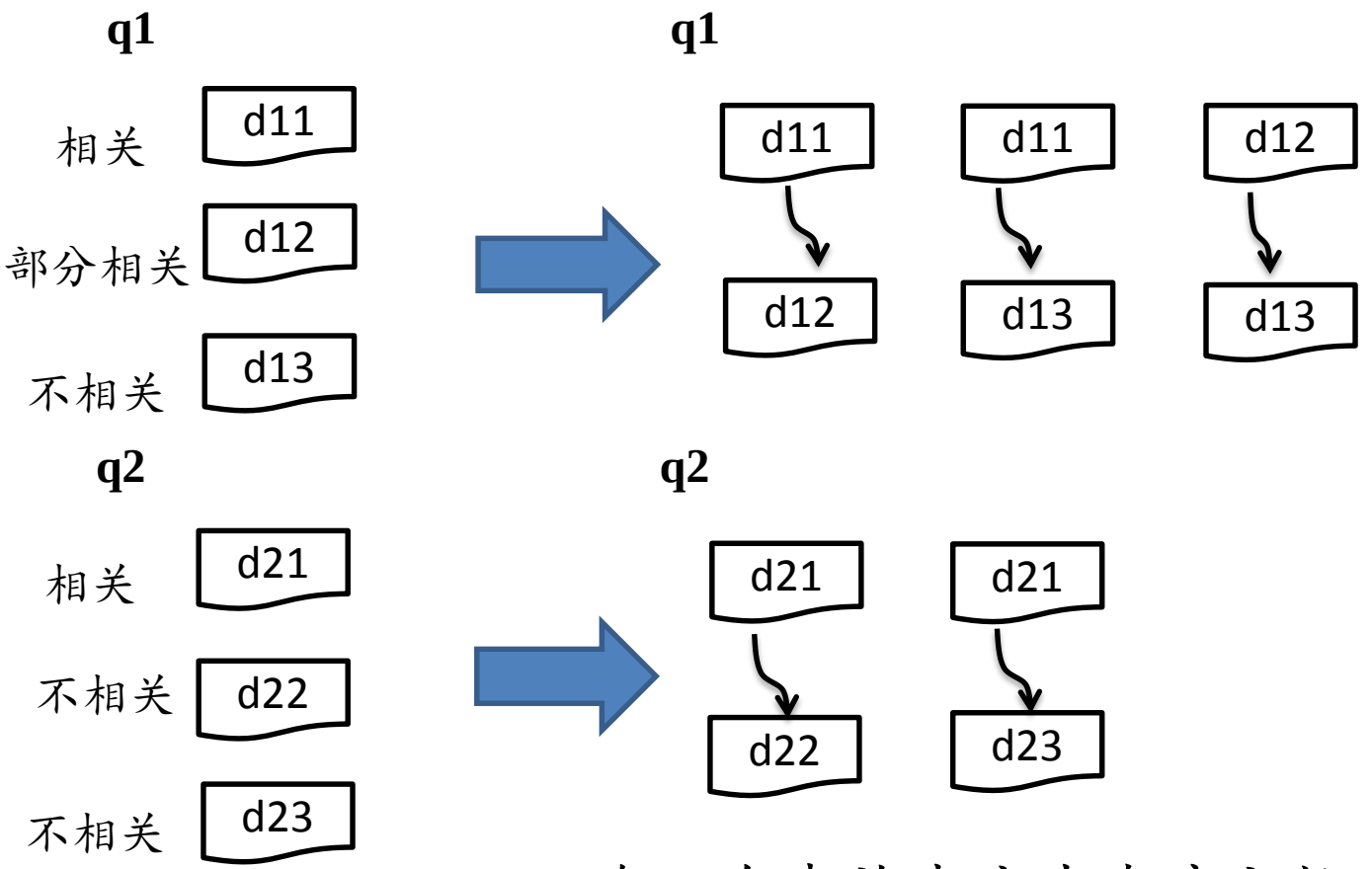
Pair-wise Learning to Rank

Pair-wise排序学习:区分文档间差异

- Pair-wise核心思想: 模型只需要区分同一个查询内部标注为不同相关度文档间的差异
- 方法：将排序问题转化为文档对上的二值分类问题



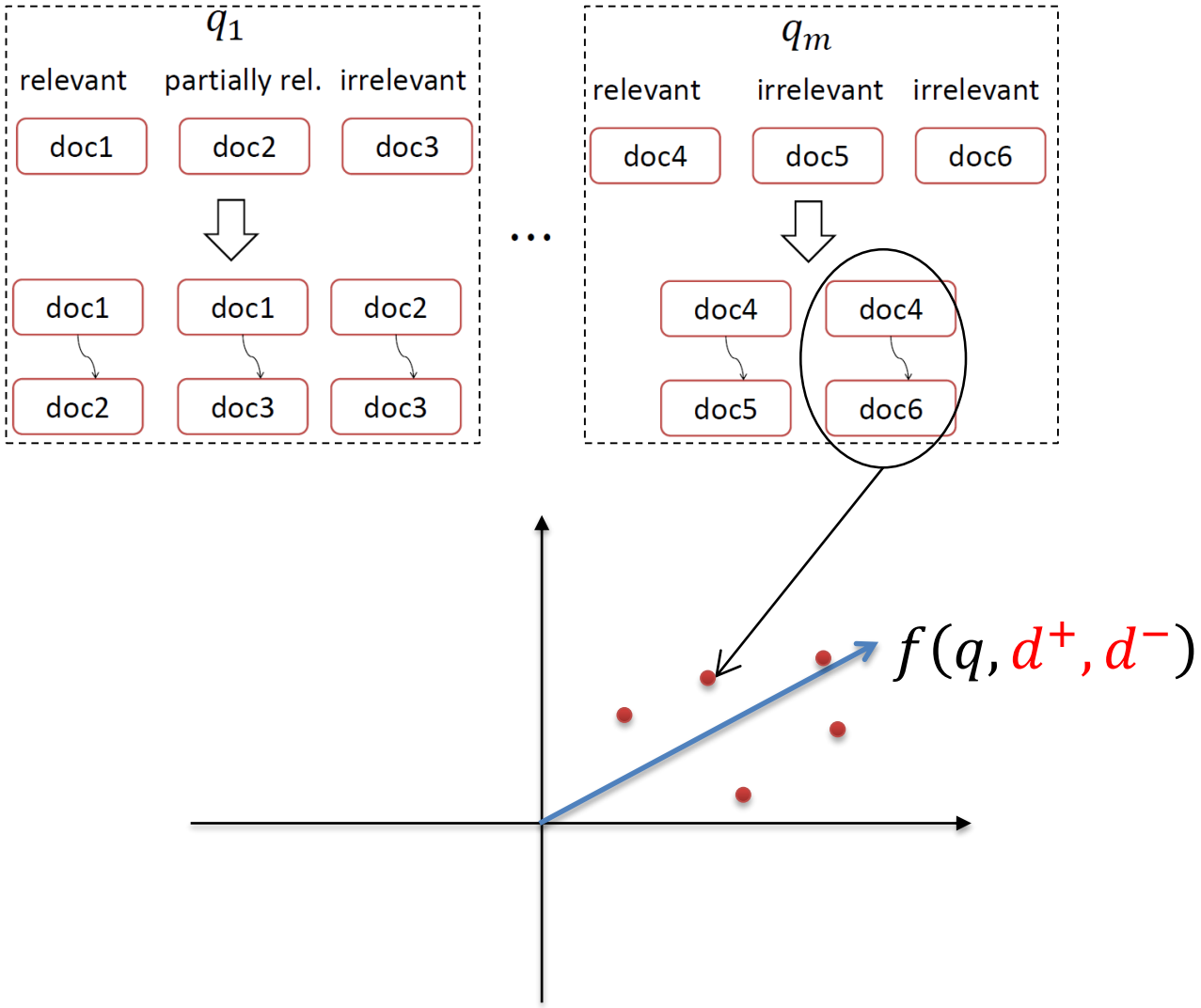
多个训练查询的情况



1. 只在一个查询内产生有序文档对(preference pair)，其原因在于只有在一个查询下，其检索出文档才可以比较。
2. 标注为同一个级别的文档之间不产生有序对

Pairwise排序学习

- 排序问题 → 文档有序对上的二值分类问题
  [Joachims, KDD '02; Freund et al., JMLR '03; Cao et al., SIGIR ' 06]



问题描述

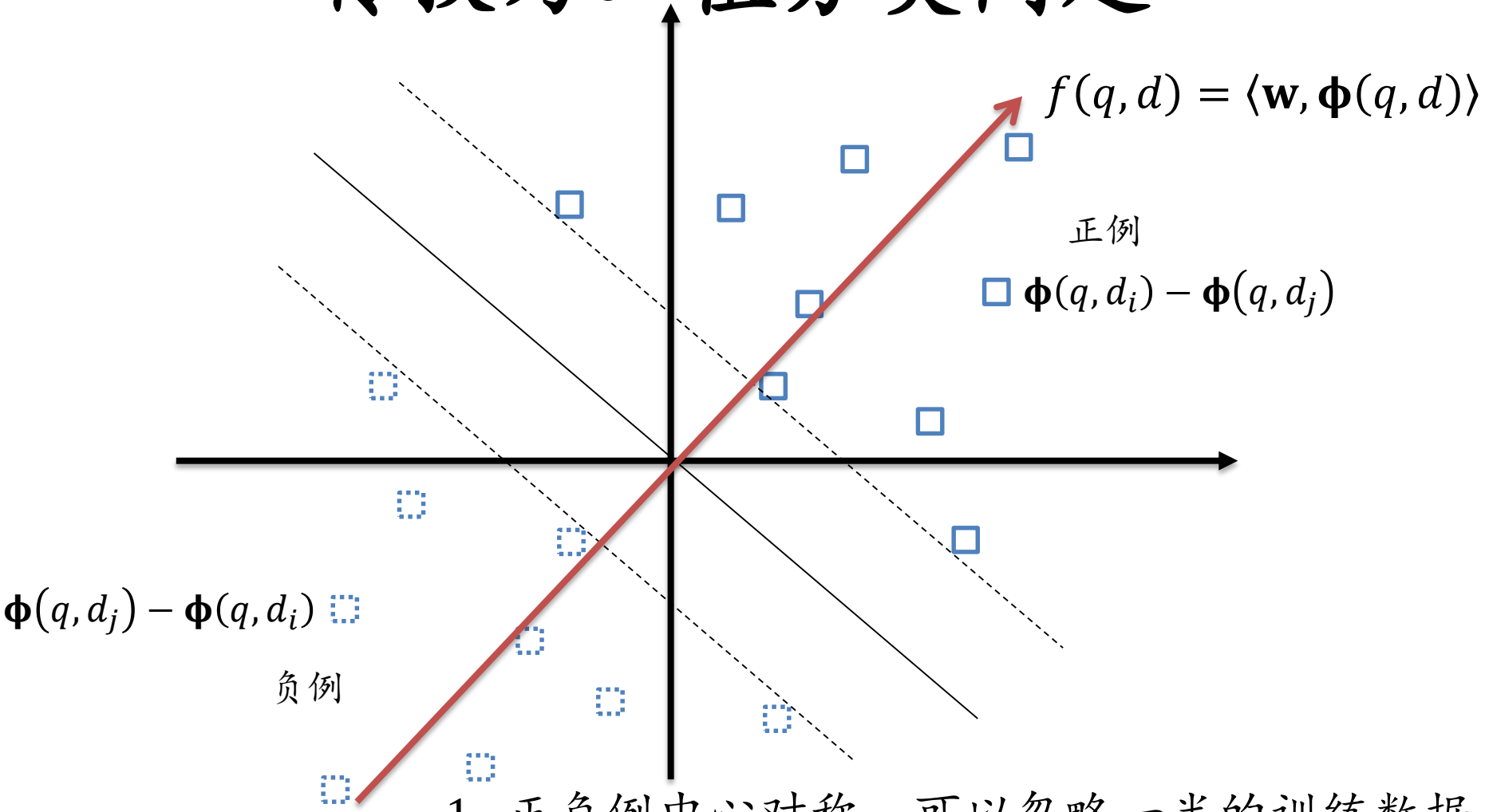
- 训练数据$\mathrm{D} = \{D_{q_i}\}_{i=1}^{N}, D_{q_i} = \{(d_{ij}, y_{ij})\}_{j=1}^{M_i}$
- 每一个查询-文档对(q-d)被表达为特征向量$\boldsymbol{\Phi}(q,d) \in X$
- 排序函数$f: X \to R$
- 排序准则:对于查询q,$d_i \succ d_j \Leftrightarrow f(q,d_i) > f(q,d_j)$
- 如为线性排序函数: $f(q,d) = \langle \mathbf{w}, \boldsymbol{\Phi}(q,d) \rangle$

  $f(q,d_i) > f(q,d_j) \Leftrightarrow \langle \mathbf{w}, \boldsymbol{\Phi}(q,d_i) - \boldsymbol{\Phi}(q,d_j) \rangle$
- 构造新的训练样本集合

$$(\boldsymbol{\Phi}(q,d_i) - \boldsymbol{\Phi}(q,d_j), z), z = \begin{cases} +1 & d_i \succ d_j \\ -1 & d_j \succ d_i \end{cases}$$

可以忽略$\mathrm{z} = -1$的样本

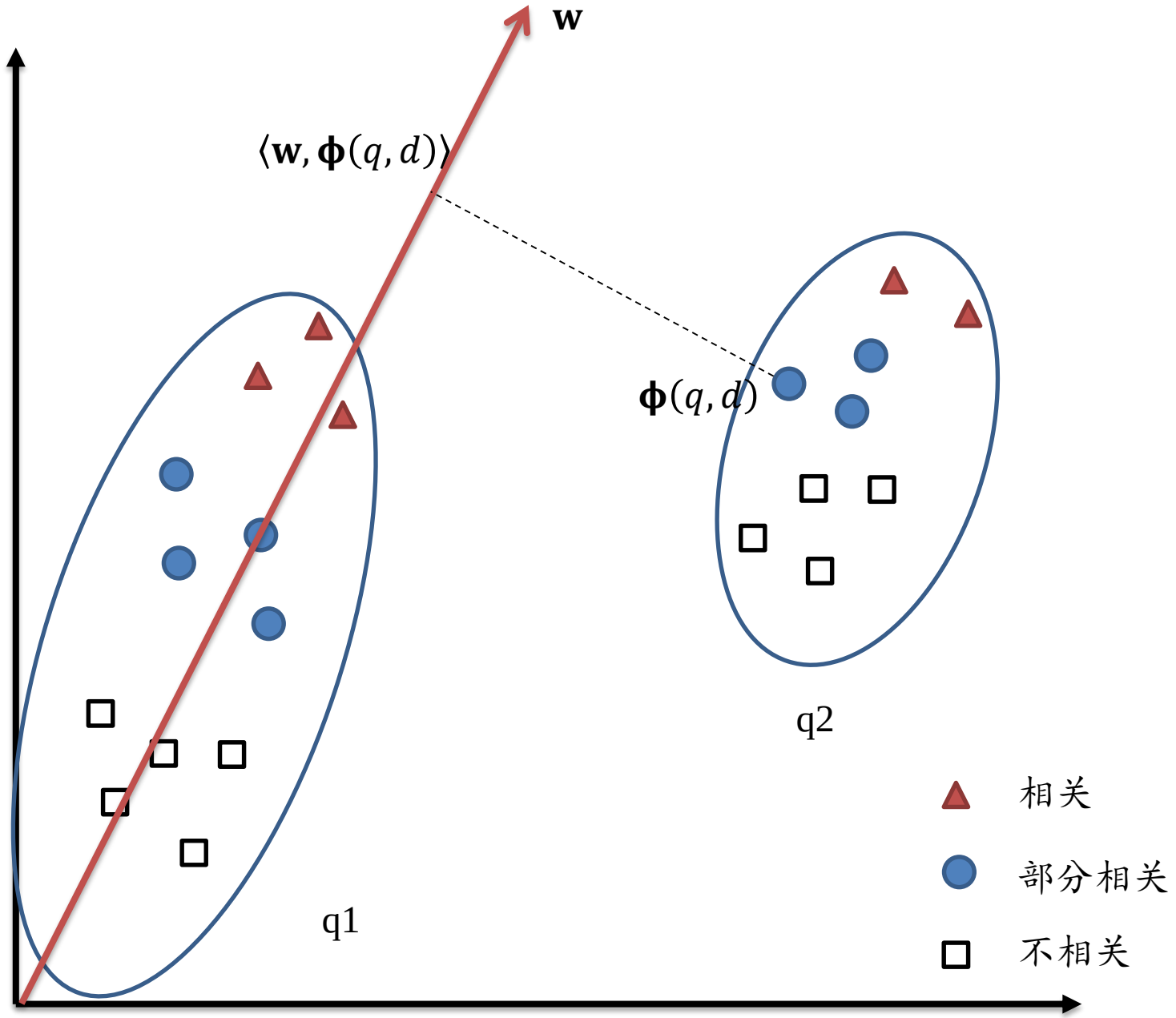
转换为二值分类问题



1. 正负例中心对称，可以忽略一半的训练数据
2. 分类平面过原点
3. 排序函数过原点(无偏置值b)

排序支持向量机(Ranking SVM)

- 用SVM解上述二值分类问题（注意：正负例中心对称，分类函数b=0）
- Ranking SVM训练
  - 第一步:构造训练数据集合$\{(\boldsymbol{\phi}(q_k, d_{ki}) - \boldsymbol{\phi}(q_k, d_{kj}), +1)\}$
  - 第二步：训练二值分类SVM模型，得到打分函数 $f(q,d) = \langle \mathbf{w}, \boldsymbol{\phi}(q,d) \rangle$
- Ranking SVM在线应用
  - 给定一个查询q和检索出的文档集合$C = \{d_i\}$
  - 用$f(q,d)$对每一个C中的文档进行打分
  - 将C中的文档按照$f(q,d)$从大到小排序


w
⟨w, ϕ(q, d)⟩
ϕ(q, d)
q1
q2
相关
部分相关
不相关

Ranking SVM优化问题

N ← 有序对的总数目

$$\min_{\mathbf{w},b} \frac{1}{2}|\mathbf{w}|^2 + C \cdot \sum_{i=1}^{N} \xi_i$$

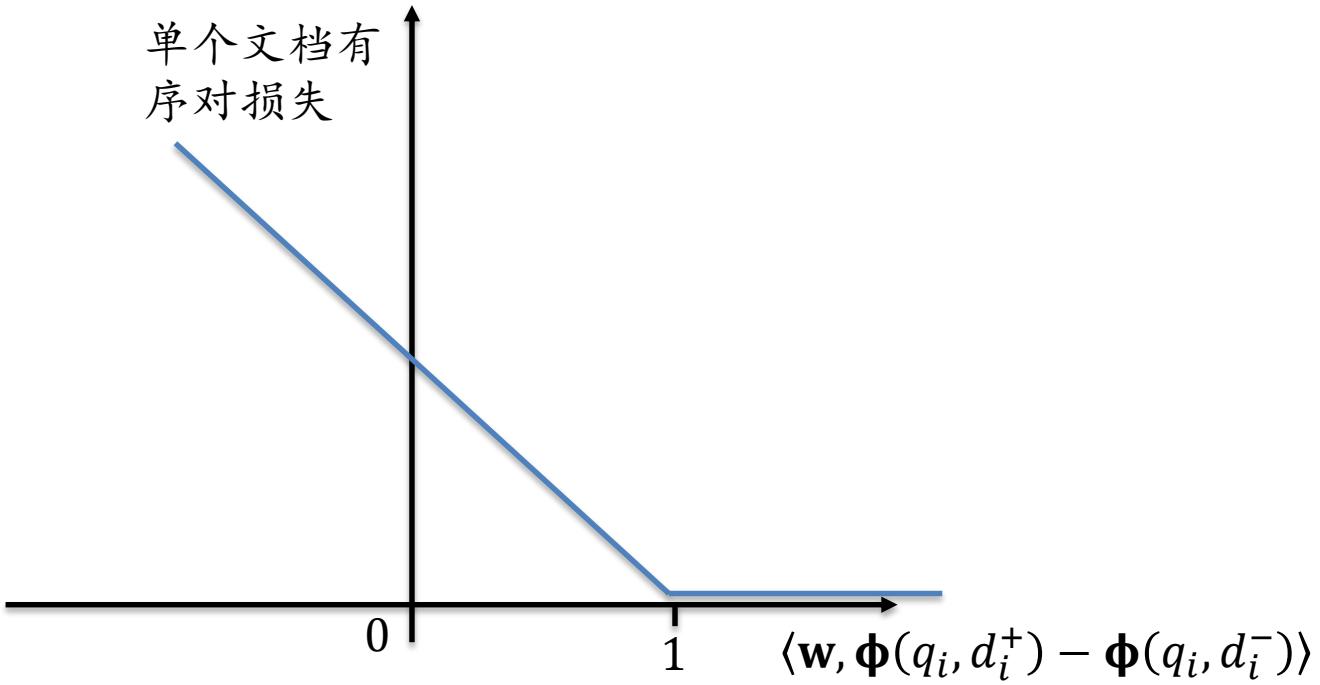
s.t. $z_i\langle \mathbf{w}, \boldsymbol{\phi}(q_i, d_i^+) - \boldsymbol{\phi}(q_i, d_i^-)\rangle \geq 1 - \xi_i,$

$\xi_i \geq 0$, for $i = 1, \ldots, N$

$$\max_{\alpha_1,\cdots,\alpha_N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{k=1}^{N}\sum_{l=1}^{N} \alpha_k \alpha_l z_k z_l \langle \boldsymbol{\phi}(q_k, d_k^+) - \boldsymbol{\phi}(q_k, d_k^-), \boldsymbol{\phi}(q_l, d_l^+)$$

从损失函数看Ranking SVM

$$L(\mathbf{w}) = \sum_{i=1}^{N} \max\{0, 1 - \langle \mathbf{w}, \boldsymbol{\phi}(q_i, d_i^+) - \boldsymbol{\phi}(q_i, d_i^-) \rangle\} + \lambda |\mathbf{w}|^2$$



Ranking SVM小结

- 将排序问题转化为文档有序对上的二值分类问题，直接应用二值SVM模型
  - 也可以应用其他所有二值分类模型，如AdaBoost
- 优点
  - 训练过程中只关注文档间的差异性，较好解决了查询间差异问题
  - 实际应用中效果良好，性能稳定
- 缺点(针对所有的Pairwise排序学习算法)
  - N个文档将产生$O(N^2)$文档有序对，时间/空间复杂度高
  - 所有有序对同等重要，未充分考虑用户从上往下浏览文档的特性
  - 违背分类训练数据独立同分布(I.I.D.)假设：$d_i \succ d_j, d_j \succ d_k$
- 源代码实现

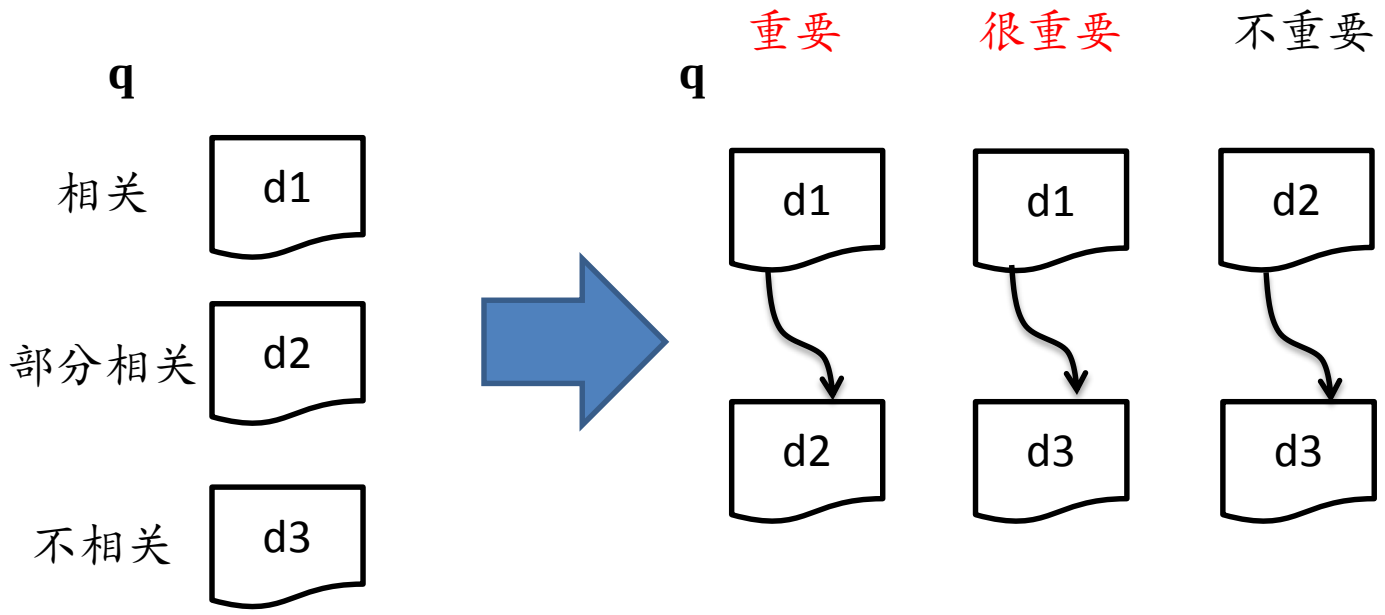
  SVM light代码：http://svmlight.joachims.org/

  Ranking SVM训练：svm_learn [options] example_file model_file

  Ranking SVM测试：svm_classify [options] example_file model_file output_file

有序对重要度

- 由于用户自顶向下浏览文档，不同的文档有序对其重要程度也不同



用不同标签间的差异来近似表示不同位置的重要度

原始Ranking SVM的问题

- 没有强调排序中序列顶部的重要性
  - 标注：相关(d)、部分相关(p)、不相关(n)
  - 排序1:p d p n n n n
  - 排序2:d p n p n n n
  - 排序2优于排序1
  - 从Ranking SVM角度，排序1和排序2错误程度相同(都只有一个有序对出错)
- 没有考虑不同查询产生有序对个数的差异
  - 查询1的标注：d p p n n n n
  - 查询2的标注：d d p p p n n n n n
  - 查询1有序对数目：2*(d→p)+4*(d→n)+8*(p→n)=14
  - 查询2有序对数目：6*(d→p)+10*(d→n)+15*(p→n)=31
  - Ranking SVM训练将偏向于查询2

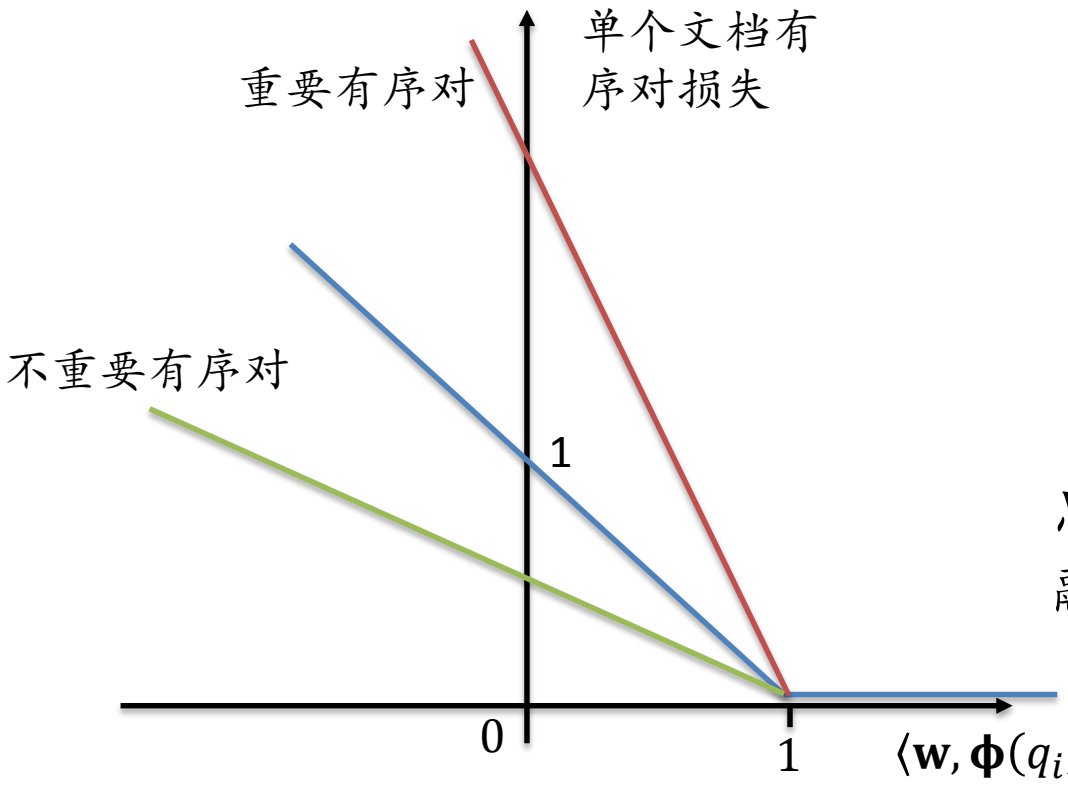
IR-SVM

Xu et al., Cost-sensitive Learning of SVM for Ranking. ECML 2006.

- 解决Ranking SVM上述两个缺陷：对重要的有序对加大权重
  - $\tau_{k(i)}$:第i个有序对通过其标签取得的权重，标签相关度越高，$\tau_{k(i)}$越大
  - $\mu_{q(i)}$:第i个有序对通过其查询取得的权重，查询下生成的有序对越多，$\mu_{q(i)}$越小
  - $\tau_{k(i)}$和$\mu_{q(i)}$通过启发式规则进行确定
- IR-SVM = 代价敏感Ranking SVM

从损失函数角度看IR-SVM

$$L(\mathbf{w}) = \sum_{i=1}^{N} \tau_{k(i)} \cdot \mu_{q(i)} \max\{0, 1 - \langle \mathbf{w}, \boldsymbol{\phi}(q_i, d_i^+) - \boldsymbol{\phi}(q_i, d_i^-) \rangle\} + \lambda |\mathbf{w}|^2$$



思考：是否还有其它方式融入代价敏感权重？

优化问题

$$\min_{\mathbf{w},b} \frac{1}{2}|\mathbf{w}|^2 + \sum_{i=1}^{N} C_i \xi_i \qquad C_i = \frac{\tau_{k(i)} \cdot \mu_{q(i)}}{2\lambda}$$

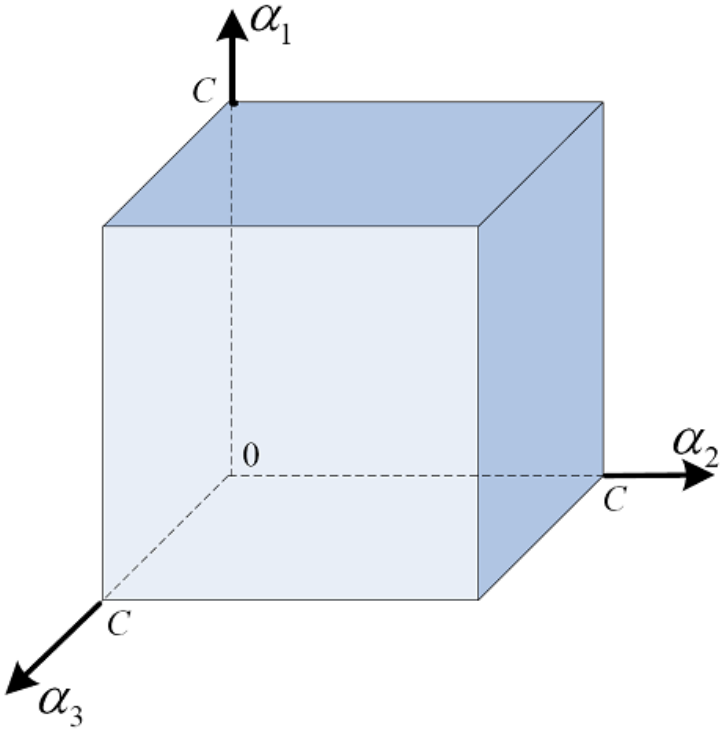
s.t. $z_i \langle \mathbf{w}, \boldsymbol{\phi}(q_i, d_i^+) - \boldsymbol{\phi}(q_i, d_i^-) \rangle \geq 1 - \xi_i,$

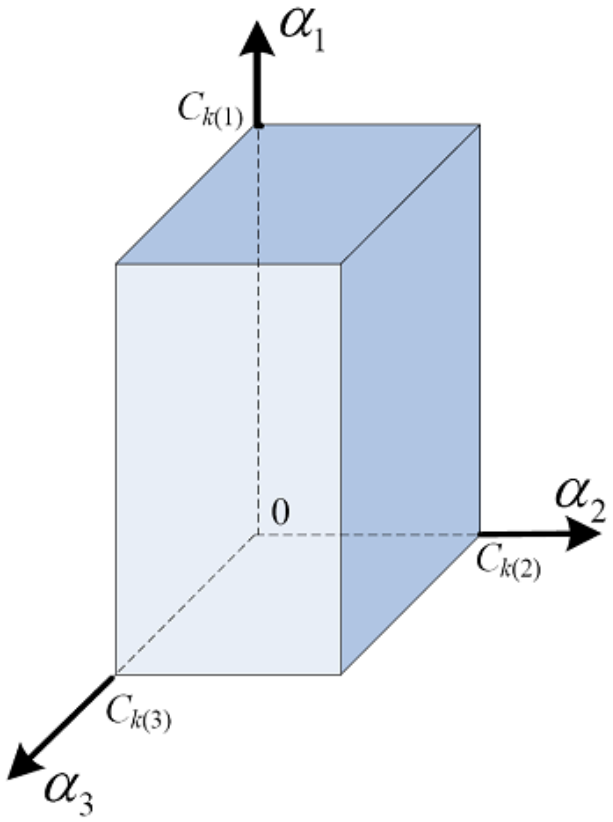
$\xi_i \geq 0$, for $i = 1, \ldots, N$

$$\max_{\alpha_1,\cdots,\alpha_N} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{k=1}^{N} \sum_{l=1}^{N} \alpha_k \alpha_l z_k z_l \langle \boldsymbol{\phi}(q_k, d_k^+) - \boldsymbol{\phi}(q_k, d_k^-), \boldsymbol{\phi}(q_l, d_l^+) \rangle$$

与Ranking SVM优化相比较

- 改变了α_i的取值上界：立方体盒子约束变为长方体盒子约束





实际效果

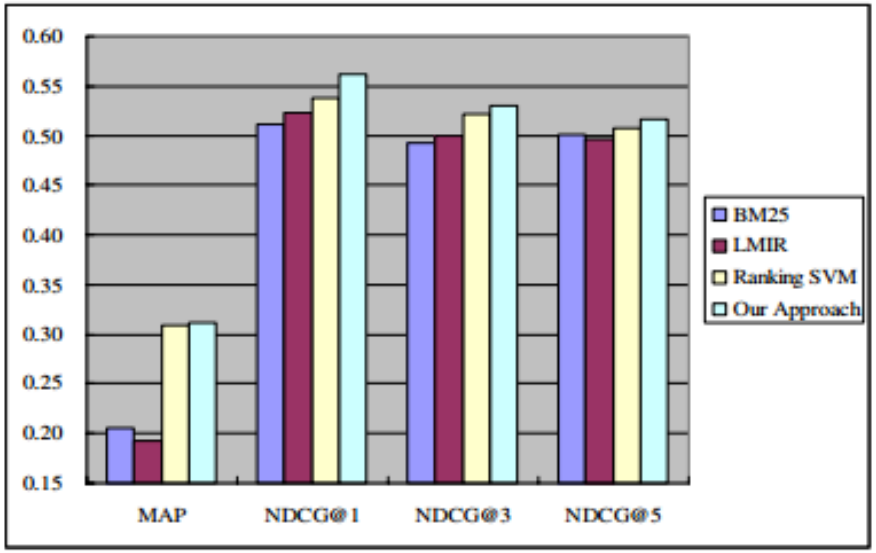


Fig. 4. Ranking accuracies in document search

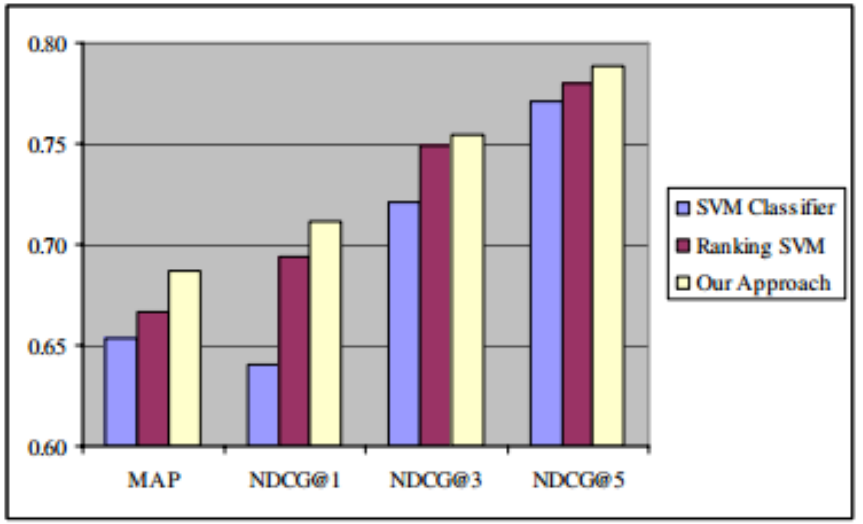


Fig. 5. Ranking accuracies in definition search

List-wise Learning to Rank

为什么要研究Listwise排序学习

- Pointwise和pairwise模型学习
  - 数据：基于网页或者网页对构造
  - 学习优化目标：减少对网页或者网页对分类的错误
- 模型的应用(在线排序)
  - 数据：用户输入的查询和检索出来的**所有文档**
  - 展示：将文档按照打分从大到小排序
  - 评价：对整个**文档序列**进行评价
- Listwise排序学习：解决模型训练和模型应用不一致问题

排序评价准则与Pairwise损失函数不一致

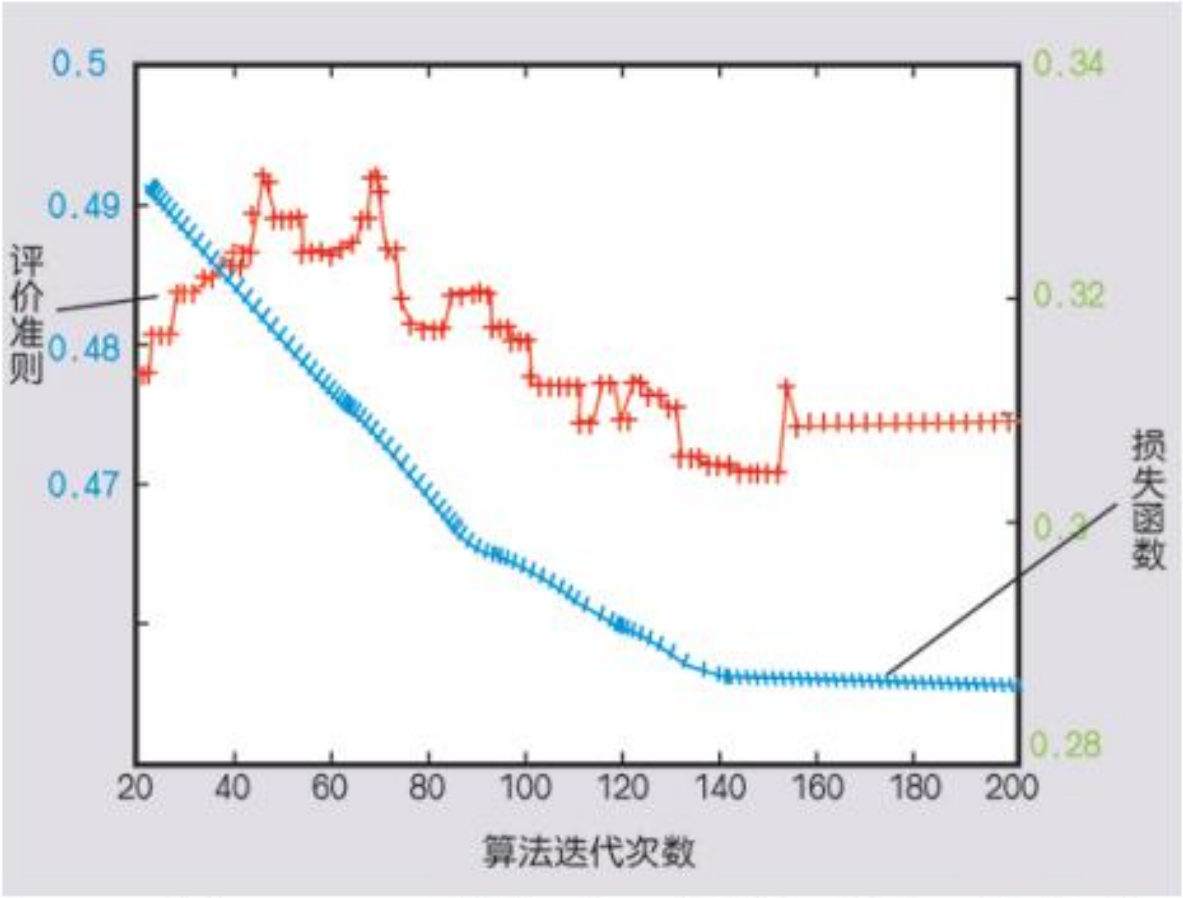


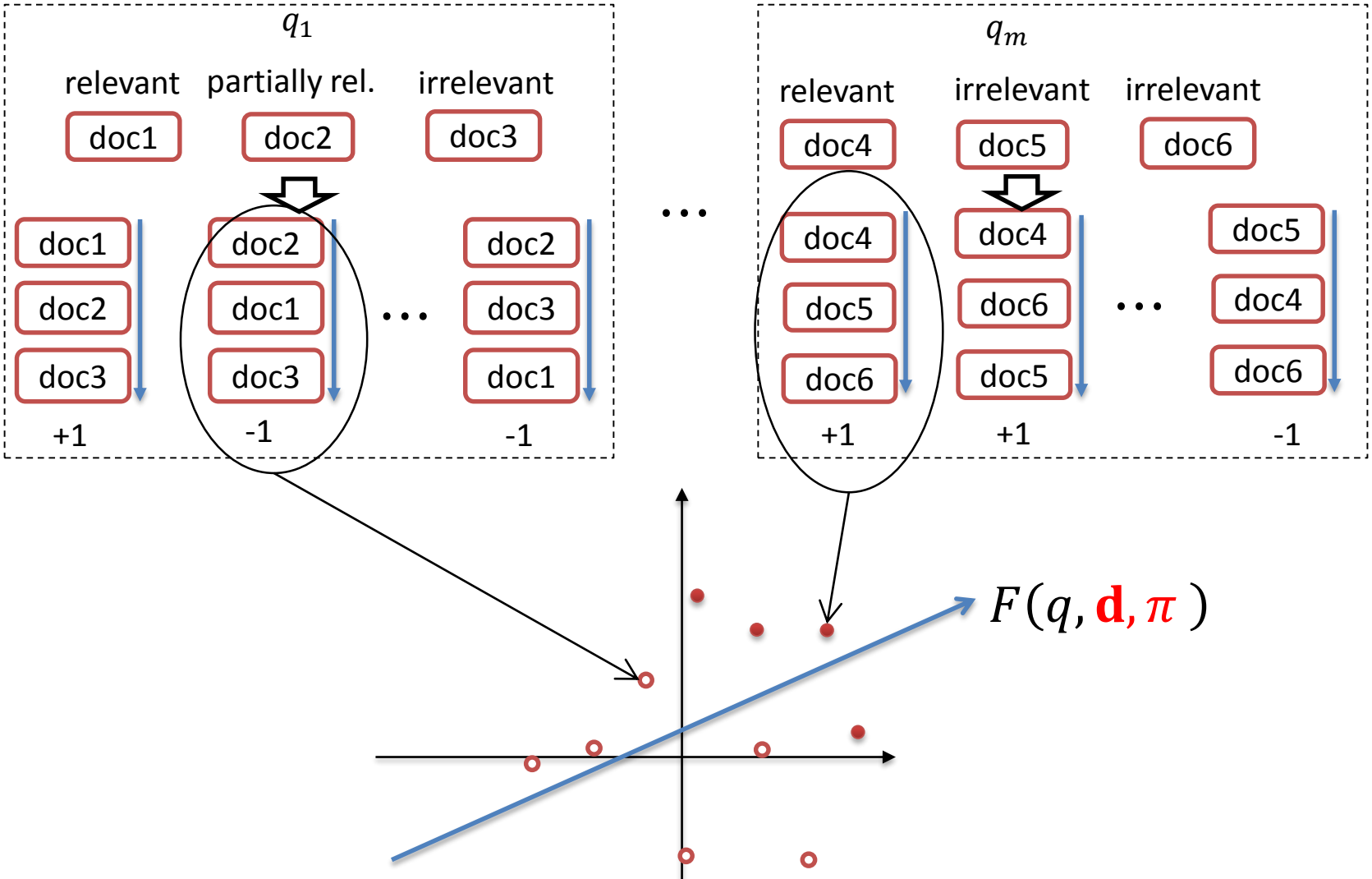
图2　随着RankNet的损失函数被极小化，评价准则NDCG并不相应提升

Listwise排序学习算法设计

- 在给定查询q和其检索出的文档集合$\mathbf{d}$，对文档的排序可以分成两大类
  - “正确”的排序：$\Pi^+ = \{\pi^+\}$，其中任意π^+中，排序评价准则对其评分均为最高分，如$\mathrm{MAP}(\pi^+)=1$
  - “不正确”的排序：$\Pi^- = \{\pi^-\}$，其中任意π^-中，排序评价准则对其评分低于最高分，如$\mathrm{MAP}(\pi^-)<1$
- 学习问题：构造一个二值分类器，学习排序模型$F(q, \mathbf{d}, \pi)$, 使得$F(q, \mathbf{d}, \pi^+) > F(q, \mathbf{d}, \pi^-)$, $\forall(\pi^+, \pi^-) \in \Pi^+ \times \Pi^-$

Listwise 排序学习

- 排序问题→在文档列表级别上的分类问题



在线排序

- 在给定查询q和其检索出的文档集合$\mathbf{d}$，对于一般形式的排序模型$F(q, \mathbf{d}, \pi)$，最优排序为

$$\pi^* = \text{argmax}_{\pi \in \Pi} F(q, \mathbf{d}, \pi)$$

  - 在线排序不能再看成简单的文档赋值+排序(sort)
- 时间复杂度可能会比较高
  - N个文档有N!个排列
- 精心选取$F(q, \mathbf{d}, \pi)$的形式可以降低在线排序复杂度

对$F(q,\mathbf{d},\pi)$的设计——非概率模型

- 基于对单个文档的打分$f(q,d)$设计$F(q,\mathbf{d},\pi)$
  - 文档打分函数：$f(q,d)=w^T\phi(q,d)$
  - 排列打分函数：$F(q,\mathbf{d},\pi)=w^T\Phi(q,\mathbf{d},\pi)$，其中

$$\Phi(q,\mathbf{d},\pi)=\frac{1}{N(N-1)}\sum_{k,l:k<l}\left[z_{kl}\left(\phi(q,d_k)-\phi(q,d_l)\right)\right]$$

  如$\pi(k)<\pi(l), z_{kl}=+1$；否则$z_{kl}=-1$

- 在线排序

$$\pi^*=\text{argmax}_\pi w^T\Phi(q,\mathbf{d},\pi)$$

  等价于

  将$\mathbf{d}$中的文档按照$f(q,d)=w^T\phi(q,d)$进行排序

证明：Jun Xu et al., Directly Optimizing Evaluation Measures in Learning to Rank. SIGIR 2008.

Listwise排序学习算法举例：SVM^{map} [Yue SIGIR 2007]

- 模型：非概率模型$F(q,\mathbf{d},\pi)=w^T\Phi(q,\mathbf{d},\pi)$
- 基于标注数据构造出正例集合Π^*/负例$\Pi\backslash\Pi^*$
- 利用SVM解决Listwise排序学习问题

$$\min_{\vec{w};\xi\geq 0}\frac{1}{2}\|\vec{w}\|^2+\frac{C}{m}\sum_{i=1}^{m}\xi_i$$

一个查询对应一个松弛变量

$$s.t.\quad \forall i,\forall\pi_i^*\in\Pi_i^*,\forall\pi_i\in\Pi_i\setminus\Pi_i^*:$$

$$F(q_i,\mathbf{d}_i,\pi_i^*)-F(q_i,\mathbf{d}_i,\pi_i)\geq E(\pi_i^*,\mathbf{y}_i)-E(\pi_i,\mathbf{y}_i)-\xi_i,$$

对正例的打分

对负例的打分

不同的正负例对应不同的边界，用MAP差异进行定义

SVM^{map}

- 损失函数

$$\sum_{i=1}^{m}\left[\max_{\pi_i^*\in\Pi_i^*;\pi_i\in\Pi_i\setminus\Pi_i^*}((E(\pi_i^*,\mathbf{y}_i)-E(\pi_i,\mathbf{y}_i))\right.$$
$$\left.-(F(q_i,\mathbf{d}_i,\pi_i^*)-F(q_i,\mathbf{d}_i,\pi_i)))\right]_+ + \lambda\parallel\vec{w}\parallel^2 .$$

- 可以证明，上述损失函数是以下函数的上界

$$R(F)=\sum_{i=1}^{m}(E(\pi_i^*,\mathbf{y}_i)-E(\pi_i,\mathbf{y}_i))=\sum_{i=1}^{m}(1-E(\pi_i,\mathbf{y}_i))$$

直接优化（任意）评价准则

AdaRank: [Xu and Li, SIGIR 2007] 利用Boosting解决Listwise排序学习

- 优化(任意)排序评价准则

$$R(F) = \sum_{i=1}^{m} (E(\pi_i^*, \mathbf{y}_i) - E(\pi_i, \mathbf{y}_i)) = \sum_{i=1}^{m} (1 - E(\pi_i, \mathbf{y}_i))$$

- 思路来源于AdaBoost算法
  - 通过组合多个“弱”排序函数，得到“强”的排序函数
  - 在查询级别考虑弱排序函数的构造和组合

AdaRank优化目标推导

任意取值在0~1之间
排序评价准则函数

依据排序模型 f 得到的对文档的排序

$$\max_{f \in \mathcal{F}} \sum_{i=1}^{m} E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i)$$

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} (1 - E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i))$$

$$e^{-x} \geq 1 - x$$

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} \exp\{-E(\pi(q_i, \mathbf{d}_i, f), \mathbf{y}_i)\}$$

$$f(\vec{x}) = \sum_{t=1}^{T} \alpha_t h_t(\vec{x})$$

$$\min_{h_t \in \mathcal{H}, \alpha_t \in \Re^+} L(h_t, \alpha_t) = \sum_{i=1}^{m} \exp\{-E(\pi(q_i, \mathbf{d}_i, f_{t-1} + \alpha_t h_t), \mathbf{y}_i)\}$$

算法流程

AdaRank

Input: $S = \{(q_i, \mathbf{d}_i, \mathbf{y}_i)\}_{i=1}^{m}$, and parameters E and T
Initialize $P_1(i) = 1/m$.
For $t = 1, \cdots, T$

- Create weak ranker h_t with weighted distribution $\boldsymbol{P}_t$ on training data S.
- Choose α_t

THEOREM 1. *The following bound holds on the ranking accuracy of the AdaRank algorithm on training data:*

$$\frac{1}{m}\sum_{i=1}^{m} E(\pi(q_i, \mathbf{d}_i, f_T), \mathbf{y}_i) \geq 1 - \prod_{t=1}^{T} e^{-\delta^t_{min}} \sqrt{1 - \varphi(t)^2},$$

where $\varphi(t) = \sum_{i=1}^{m} P_t(i) E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i)$, $\delta^t_{min} = \min_{i=1,\cdots,m} \delta^t_i$, *and*

$$\delta^t_i = E(\pi(q_i, \mathbf{d}_i, f_{t-1} + \alpha_t h_t), \mathbf{y}_i) - E(\pi(q_i, \mathbf{d}_i, f_{t-1}), \mathbf{y}_i) - \alpha_t E(\pi(q_i, \mathbf{d}_i, h_t), \mathbf{y}_i),$$

for all $i = 1, 2, \cdots, m$ *and* $t = 1, 2, \cdots, T$.

End For
Output ranking model: $f(\vec{x}) = f_T(\vec{x})$.

AdaBoost

Initialization...
For $t = 1, ..., T$:

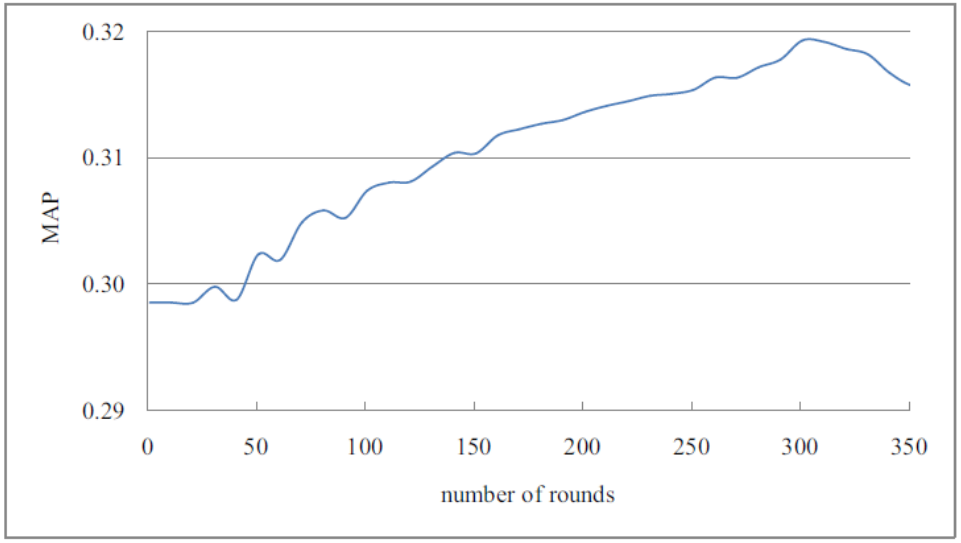
- Find $h_t = \arg\min_{h_j \in \mathcal{H}} \epsilon_j = \sum_{i=1}^{m} D_t(i)[y_i \neq h_j(x_i)]$
- If $\epsilon_t \geq 1/2$ then stop
- Set $\alpha_t = \frac{1}{2}\log(\frac{1+r_t}{1-r_t})$
- Update

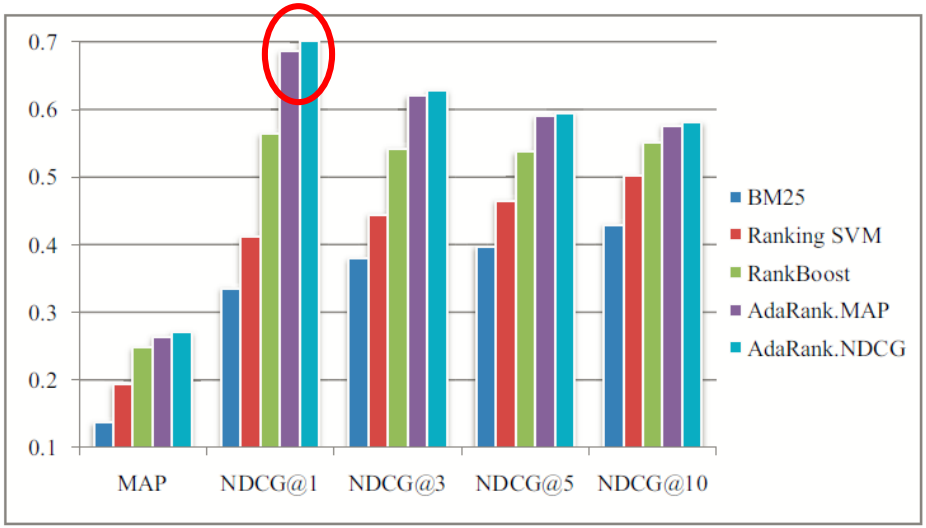
$$D_{t+1}(i) = \frac{D_t(i) exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Output the final classifier:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

实验效果





收敛性

排序效果

AdaRank可执行代码

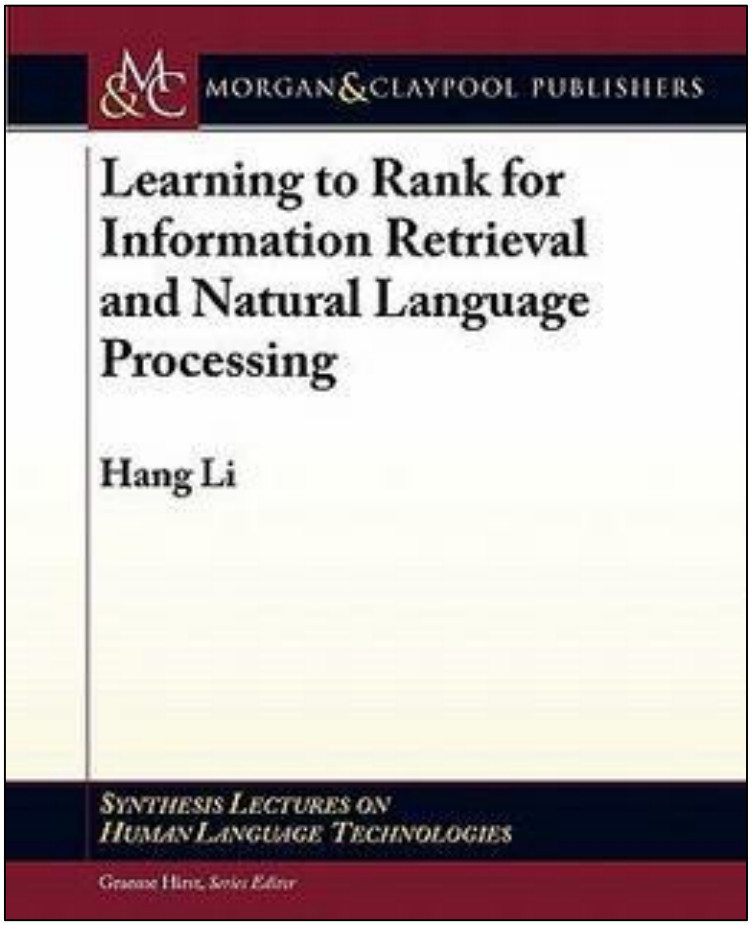
http://research.microsoft.com/en-us/downloads/0eae7224-8c9b-4f1e-b515-515c71675d5c/

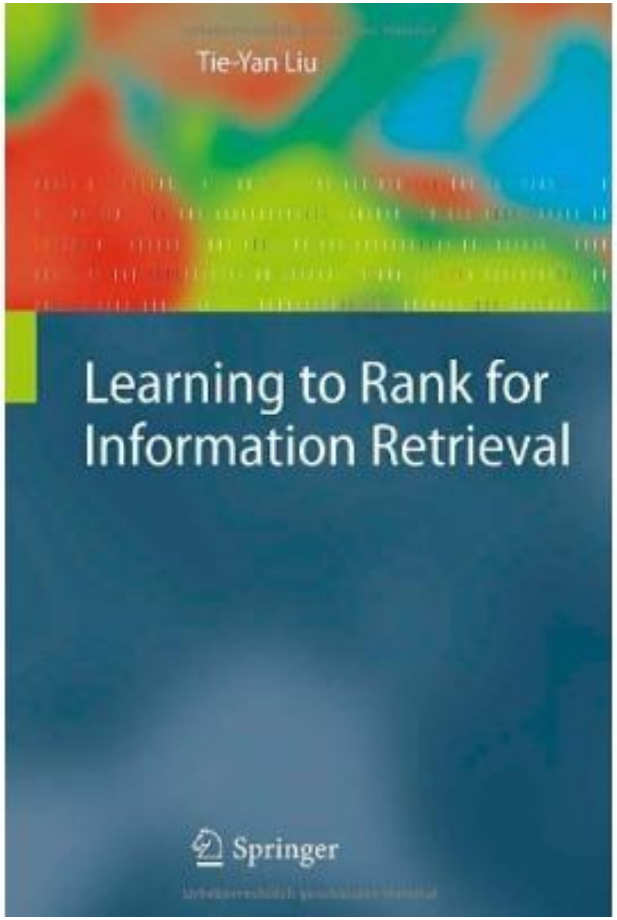
更多排序学习算法

- Pairwise排序算法
  - RankBoost
- Listwise排序算法
  - ListNet, ListMLE, LambdaMART等
- 多样化排序学习
  - R-LTR, PAMM
- 排序学习数据集合Letor

  http://research.microsoft.com/en-us/um/beijing/projects/letor/

关于排序学习的资料


MORGAN&CLAYPOOL PUBLISHERS
Learning to Rank for Information Retrieval and Natural Language Processing
Hang Li
SYNTHESIS LECTURES ON HUMAN LANGUAGE TECHNOLOGIES
Graeme Hirst, Series Editor


Tie-Yan Liu
Learning to Rank for Information Retrieval
Springer

排序学习小结

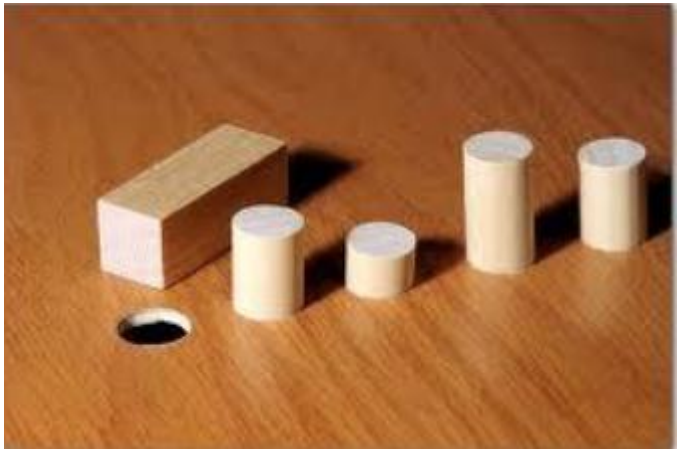
- 通过有监督机器学习算法构造排序模型
  - 可以同时考虑多个排序因素
  - 算法适应性强，其应用不限于相关性排序
- 排序学习算法
  - Point-wise:直接应用分类/回归模型，未考虑排序的特点
  - Pair-wise:将排序问题转化为文档有序对上的二值分类问题，取得了较好的效果
  - List-wise:在文档列表(查询)层面构造排序损失函数

提纲

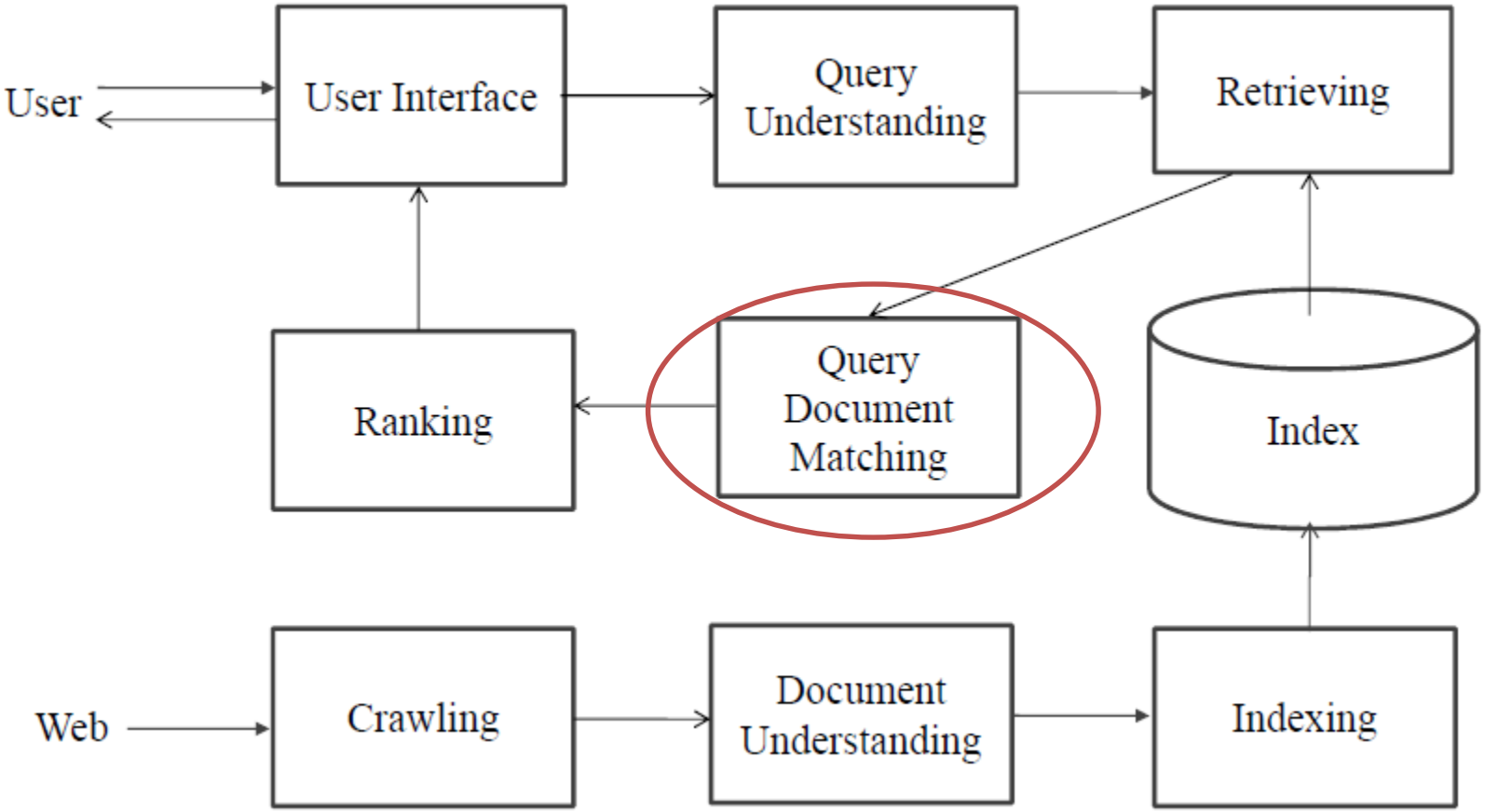
- 排序学习
  - 问题定义
  - 排序学习算法
- 语义匹配
  - 问题定义
  - 匹配学习算法
- 总结

匹配(matching)

- 搜索：查询⇔文档
- 问答：问题⇔答案
- 推荐：用户⇔商品
- 翻译：源语言⇔目标语言



搜索引擎主要模块

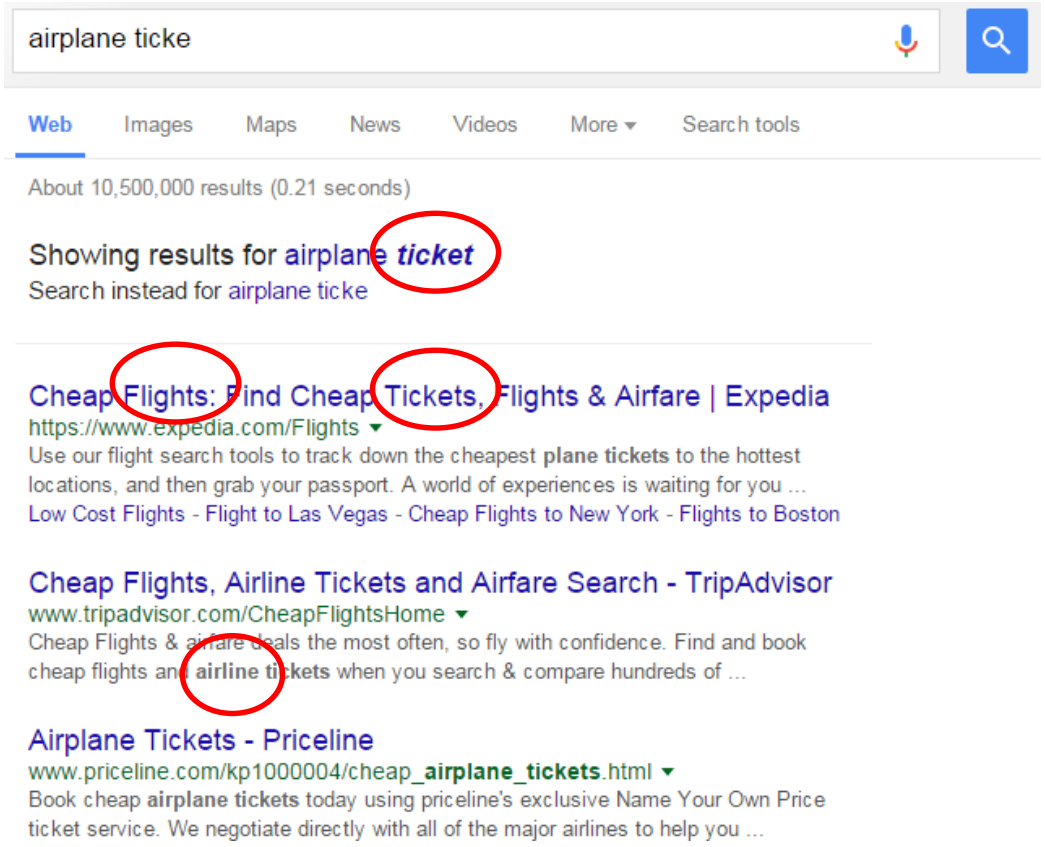

User
User Interface
Query Understanding
Retrieving
Ranking
Query Document Matching
Index
Web
Crawling
Document Understanding
Indexing

排序与匹配的关系

- 排序:关注与一个查询与多个文档
  - 给定查询和检索出的文档
  - 抽取查询-文档关系特征
  - 目标:对文档进行最优的排序
- 匹配:关注与一个查询与一个文档
  - 给定一个查询与一个文档
  - 目标:计算查询与文档的匹配值
- 匹配可以为排序学习提供强有力的特征

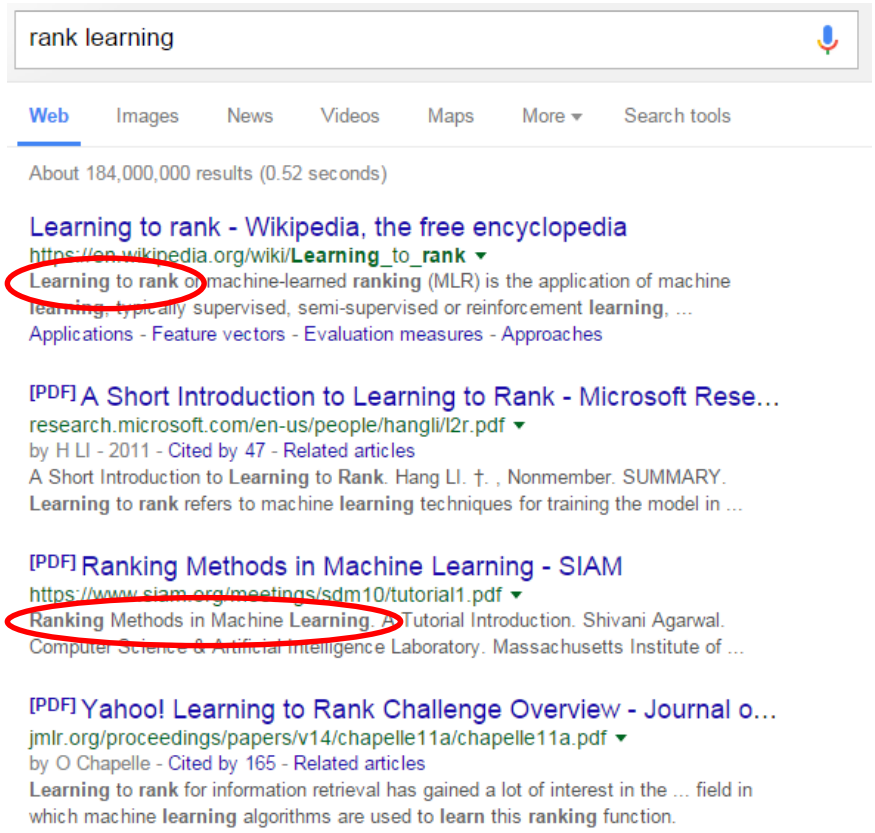
匹配:让搜索更人性化

- 模糊匹配
  - airplane → flights, airline



匹配:让搜索更人性化

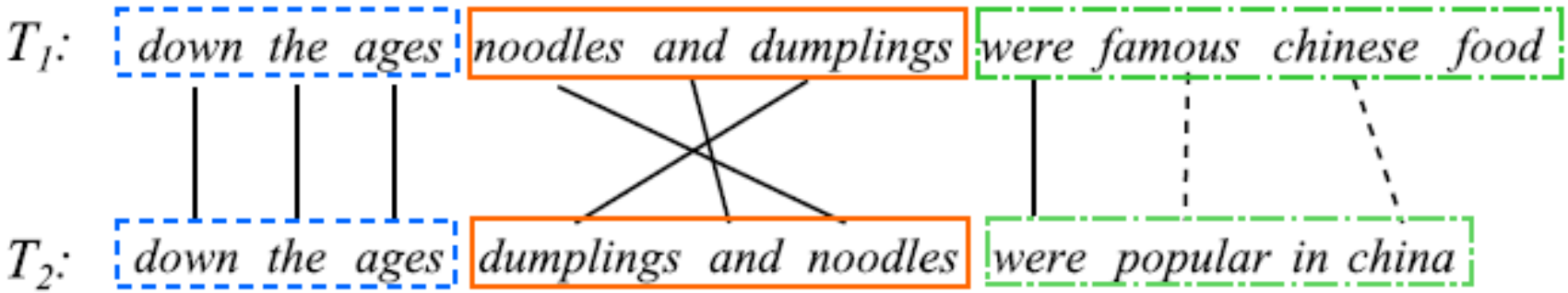
- 临近度(proximity)
  - data mining → the data mining course ☺
  - data mining → big data … web mining … ☹



更多的匹配举例

query	document	term match	semantic match
seattle best hotel	seattle best hotels	partial	yes
pool schedule	swimming pool schedule	partial	yes
natural logarithm transform	logarithm transform	partial	yes
china kong	china hong kong	partial	no
why are windows so expensive	why are macs so expensive	partial	no

举例：Paraphrase identification



- 匹配级别
  - 单词(word level)
  - 词组(phrase level)
  - 句子(sentence level)
  - 篇章(paragraph level)

互联网搜索中的语义匹配

互联网搜索中的语义匹配

- 查询⇔文档
- 查询
  - 用户即时输入，用词随意，可能含有错误
  - 反映用户搜索意图，focus
  - 短文本，关键词的组合/自然语言
- 文档
  - 用词相对严谨，含有错误可能性相对较小
  - 可能包含多个话题
  - 自然语言篇章，通常有多个域

一样的搜索意图，不同的查询

Table 1.2: Queries about "distance between sun and earth".

"how far" earth sun	average distance from the earth to the sun
"how far" sun	how far away is the sun from earth
average distance earth sun	average distance from earth to sun
how far from earth to sun	distance from earth to the sun
distance from sun to earth	distance between earth and the sun
distance between earth & sun	distance between earth and sun
how far earth is from the sun	distance from the earth to the sun
distance between earth sun	distance from the sun to the earth
distance of earth from sun	distance from the sun to earth
"how far" sun earth	how far away is the sun from the earth
how far earth from sun	distance between sun and earth
how far from earth is the sun	how far from the earth to the sun
distance from sun to the earth	

一样的搜索意图，不同的查询

Table 1.3: Queries about "Youtube".

yutube	yuotube	yuo tube
ytube	youtubr	yu tube
youtubo	youtuber	youtubecom
youtube om	youtube music videos	youtube videos
youtube	youtube com	youtube co
youtub com	you tube music videos	yout tube
youtub	you tube com yourtube	your tube
you tube	you tub	you tube video clips
you tube videos	www you tube com	wwww youtube com
www youtube	www youtube com	www youtube co
yotube	www you tube	www utube com
ww youtube com	www utube	www u tube
utube videos	utube com	utube
u tube com	utub	u tube videos
u tube	my tube	toutube
outube	our tube	toutube

传统的思路

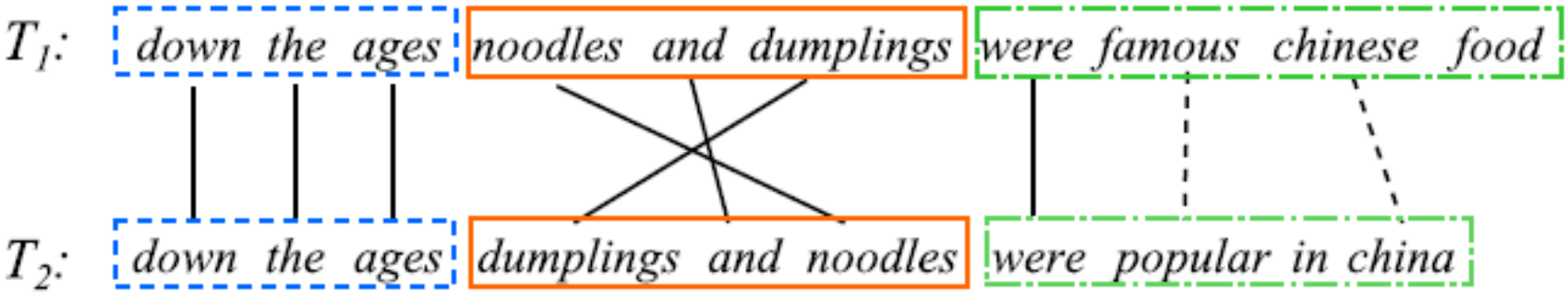
- 查询理解 + 文档理解 (NLP问题)
- 自然语言处理精度成为瓶颈
  - 词法分析(lexical analysis, including word segmentation and part-of-speech tagging): 可实用
  - 句法分析(Syntactic analysis): 勉强可用
  - 语义分析(semantic analysis): 实用困难
- 需要处理随意性很强的互联网数据

	English	Chinese
Pragmatic Analysis	?	?
Semantic Role Labeling	>=87%	>=75%
Syntactic Analysis	>=90%	>=80%
Part of Speech Tagging	>=97%	>=93%
Word Segmentation	NA	>=95%

放弃理解，直接匹配

知其然不知其所以然

语义匹配中的两个核心问题



- 模糊匹配(mismatch)

  famous ⇔ popular; Chinese ⇔ China

- 临近度匹配(proximity)

  noodles and dumplings ⇔ dumplings and noodles

搜索中的语义匹配方法

- 查询改写(query reformulation)
- 词依赖模型(term dependency model)
- 基于话题模型的匹配(matching with topic model)
- 基于机器翻译的匹配(statistical translation model)
- 基于深度学习的匹配(deep learning)
- **基于隐空间模型的匹配(latent space model)**

Foundations and Trends® in Information Retrieval
Vol. 7, No. 5 (2013) 343–469

DOI: 10.1561/1500000035

now
the essence of knowledge

Semantic Matching in Search

Hang Li
Huawei Technologies, Hong Kong
hangli.hl@huawei.com

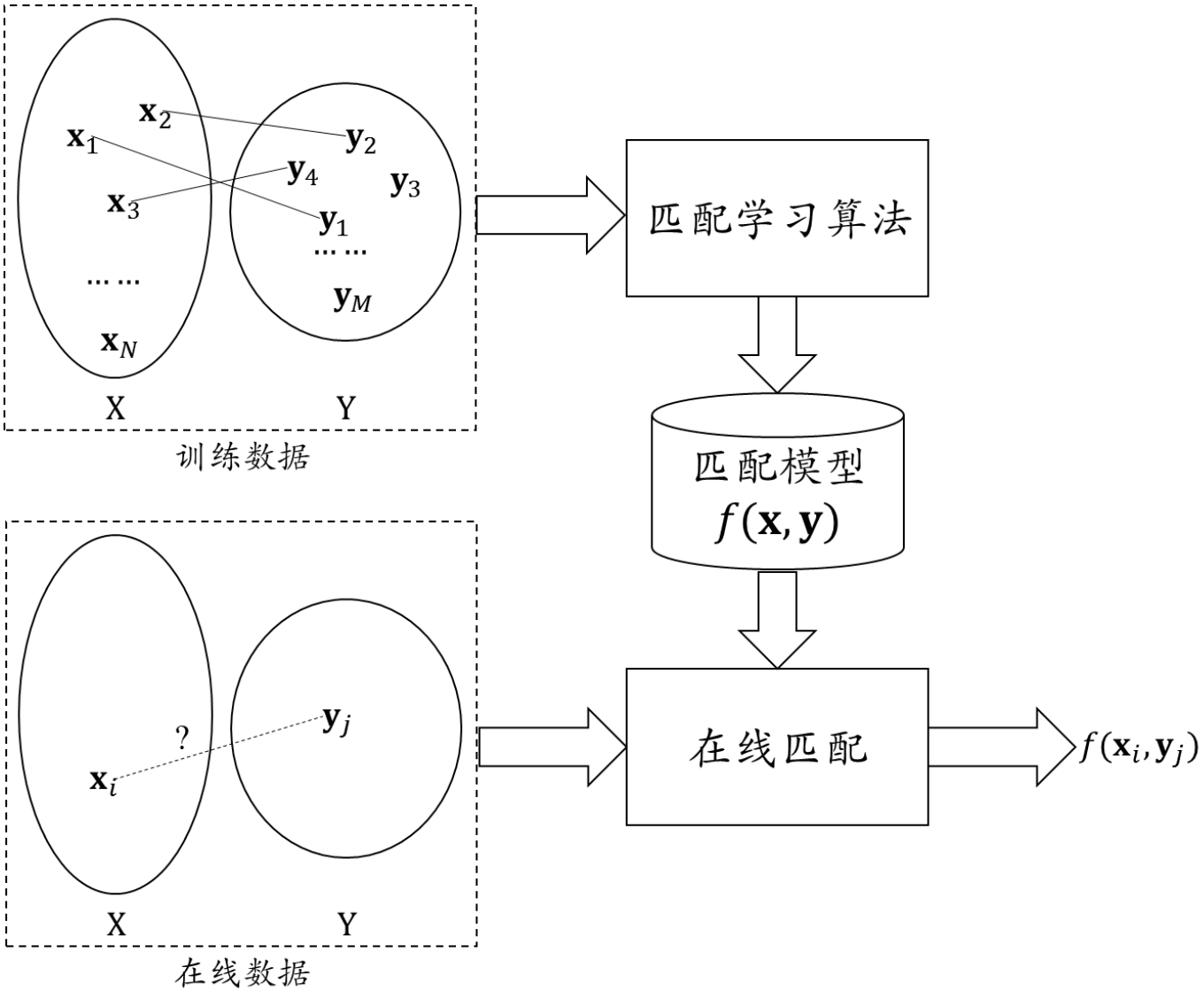
Jun Xu
Huawei Technologies, Hong Kong
nkxujun@gmail.com

http://www.bigdatalab.ac.cn/~junxu/publications/SemanticMatchingInSearch_2014.pdf

提纲

- 排序学习
  - 问题定义
  - 排序学习算法
- 语义匹配
  - 问题定义
  - 匹配学习算法
- 总结

匹配学习


$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_3$
……
$\mathbf{x}_N$
X
$\mathbf{y}_2$
$\mathbf{y}_4$
$\mathbf{y}_3$
$\mathbf{y}_1$
……
$\mathbf{y}_M$
Y
训练数据
匹配学习算法
匹配模型
$f(\mathbf{x},\mathbf{y})$
$\mathbf{x}_i$
?
$\mathbf{y}_j$
X
Y
在线数据
在线匹配
$f(\mathbf{x}_i,\mathbf{y}_j)$

数据集合构造

- Query-title pairs in click-through data (Gao et al., ’10)

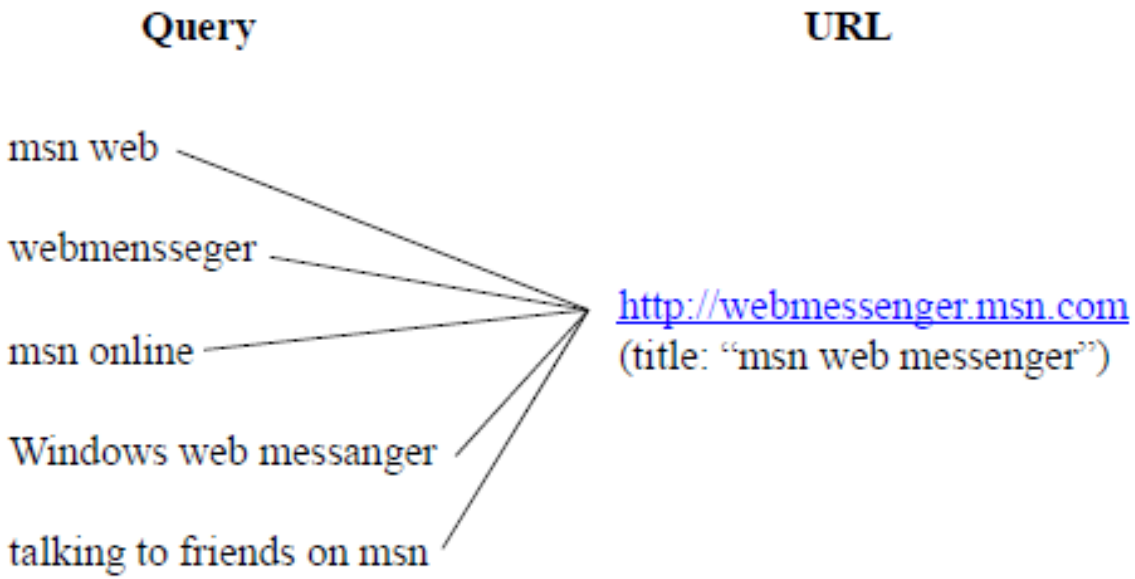


Figure 5.2: An example of URL and its associated queries extracted from click-through data.

隐空间匹配学习

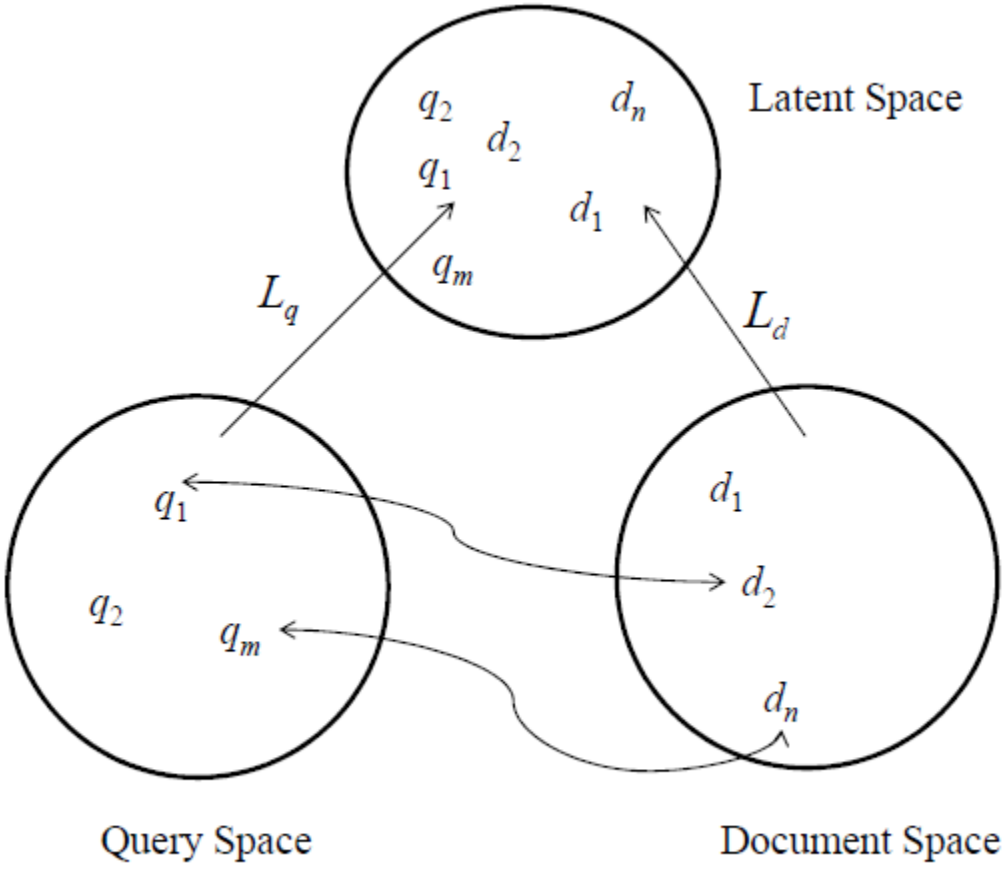


Figure 7.1: Matching in latent space.

Partial Least Square (PLS)

- 匹配模型：$f(q,d) = \langle L_q \cdot q, L_d \cdot d \rangle$
- 给定训练数据集合$D = \{(q_i, d_i, c_i)\}$,其中c_i是查询q_i提交后文档d_i被点击的次数
- 优化问题

$$\arg\max_{L_q, L_d} = \sum_{(q_i, d_i)} c_i f(q_i, d_i),$$

$$L_q L_q^T = I, \quad L_d L_d^T = I$$

- 求解方法：SVD分解求得全局最优解

Regularized Mapping to Latent Space (RMLS)

- PLS的问题：SVD分解复杂度高，无法解决大规模优化问题
- 解决方案：放弃正交约束，采用范数约束

$$\arg\max_{L_q, L_d} = \sum_{(q_i, d_i)} c_i f(q_i, d_i),$$

$$|l_q| \leq \theta_q, \quad |l_d| \leq \theta_d, \quad \|l_q\| \leq \tau_q, \quad \|l_d\| \leq \tau_d$$

- 缺点：L_q与L_d不正交，可能存在重复维度
- 优点：可分布式并行，实际应用效果良好

实际效果对比

Table 7.1: Performances of latent space models in search.

	NDCG@1	NDCG@3	NDCG@5
BM25 (baseline)	0.637	0.690	0.690
SSI	0.538	0.621	0.629
SVDFeature	0.663	0.720	0.727
BLTM	0.657	0.702	0.701
PLS	0.676	0.728	**0.736**
RMLS	**0.686**	**0.732**	0.729

数据集合：94,022 queries and 111,631 documents. Click through data associated with the queries and documents at a search engine is also used. Relevance judgments are made at five levels.

隐空间匹配模型小结

- 监督学习:从大量的用户点击数据中获取训练数据
- 非概率模型:基于空间映射
  - PLS:数学漂亮，大规模求解困难
  - RMLS:放弃正交约束，可分布式并行
  - 其它模型:SSI
- 概率模型
  - 双语话题模型: 生成过程清晰, 只适合于文本
- 基于深度学习的模型
  - DSSM
  - Matching Pyramid

提纲

- 排序学习
  - 问题定义
  - 排序学习算法
- 语义匹配
  - 问题定义
  - 匹配学习算法
- 总结

总结

- 本节课探讨利用机器学习方法构建排序和匹配模型
- 排序学习
  - 将排序问题转换为传统机器学习可以求解的问题(回归和分类)
  - Point-wise、Pair-wise、List-wise
- 语义匹配
  - 匹配关注与查询-文档间的匹配强度，为排序提供强有力的特征
  - 基于隐空间的匹配学习算法，将查询和文档映射到同一隐空间中

存在问题

- 排序学习
  - 在将排序问题转换为分类/回归问题的过程中，违背了机器学习的基本假设
  - 目前只关注了相关性排序，较少关注对搜索结果多样化等更多的信息检索任务
- 匹配学习
  - 长尾：基于机器学习的方法对于罕见的查询和文档缺乏必要的统计信息
    - 罕见查询(文档)中含有高频词
  - 短文本：目前的模型(PLS、RMLS)其基本原理为统计词的共现，天然对短文本不适应(对应NMF、LDA对短文本不适应)

更多的研究工作

- 排序学习
  - 建模pairwise排序学习中的参数交互问题

  Zhang et al., Modeling Parameter Interactions in Ranking SVM. CIKM 2015.
  - 多样化排序学习

  Zhu et al., Learning for Search Result Diversification. SIGIR 2014.

  Xia et al., Learning Maximal Marginal Relevance Model via Directly Optimizing Diversity Evaluation Measures. SIGIR 2015.
- 基于深度学习的匹配模型
  - 深度匹配模型DSSM
  - 分布式词(句子)表达模型如Word2vec、LSTM
  - Text matching as image processing (CNN)

  [Pang et al., AAAI 2016]

谢谢！

对$F(q, \mathbf{d}, \pi)$的设计——概率模型

- 认为$F(q, \mathbf{d}, \pi)$是将文档排成π的概率
  - π可以为任意排列
- 给定对多个样本的打分v_i，如何判断生成一个排列的概率？
- Plackett-Luce模型：多步骤生成排列
  - 首先决定第一个样本(概率为$\frac{v_{\pi(1)}}{v_{\pi(1)}+v_{\pi(2)}+\cdots+v_{\pi(N)}}$)
  - 然后决定第二个样本(概率为$\frac{v_{\pi(2)}}{v_{\pi(2)}+\cdots+v_{\pi(N)}}$)
  - …
- Plackett-Luce概率

$$P(\pi|V) = \prod_{i=1}^{N} \frac{v_{\pi(i)}}{v_{\pi(i)} + v_{\pi(i+1)} + \cdots + v_{\pi(N)}}$$

Plackett-Luce模型(续)

- 概率最大的模型

$$\pi^* = \text{argmax}_\pi P(\pi|V)$$

- π^*=按照值进行从大到小排列所得到的排列
  - 如何证明？
- 优点
  - 在Listwise学习过程中任然学习对单个文档的打分
  - 在线排序过程中，直接按照对单个文档打分进行排序即可得到$\text{argmax}_\pi P(\pi|V)$

对$F(q,\mathbf{d},\pi)$的设计

- 基于对单个文档的打分$f(q,d)$设计$F(q,\mathbf{d},\pi)$
- 基于Luce模型
  - 认为网页的分数$f(q,d)$正比于该网页被排序模型排在其他网页前面的概率
  - 在第1,2,…,j-1个网页已经排在了前j—1个位置上的条件下，第j个网页紧随其后的概率

$$p_j = \frac{f\left(q, d_{\pi(j)}\right)}{\sum_{k=j+1}^{N} f\left(q, d_{\pi(k)}\right)}$$

  - $F(q,\mathbf{d},\pi) = p(\pi) = \prod_{j=1}^{N} \frac{f\left(q,d_{\pi(j)}\right)}{\sum_{k=j+1}^{N} f\left(q,d_{\pi(k)}\right)}$

总结

- 语义匹配为搜索中的核心问题之一
  - 模糊匹配(mismatch)
  - 临近度匹配(proximity)
- 从(用户点击)数据中学习匹配模型是行之有效的方法
  - 查询改写(query reformulation)
  - 词依赖模型(term dependency model)
  - 基于话题模型的匹配(matching with topic model)
  - 基于机器翻译的匹配(statistical translation model)
  - **基于隐空间模型的匹配(latent space model)**
  - 基于深度学习的匹配(deep learning)

双语言话题模型
（Bilingual Topic Model, BLTM)

- 隐空间模型的概率化版本
- 训练数据：$D = \{(q_i, d_i)\}$，查询q_i提交后文档d_i被点击一次，即形成一个(q_i, d_i)对
- 同一个话题(ID)在查询空间和文档空间具有不同的表达(词分布)

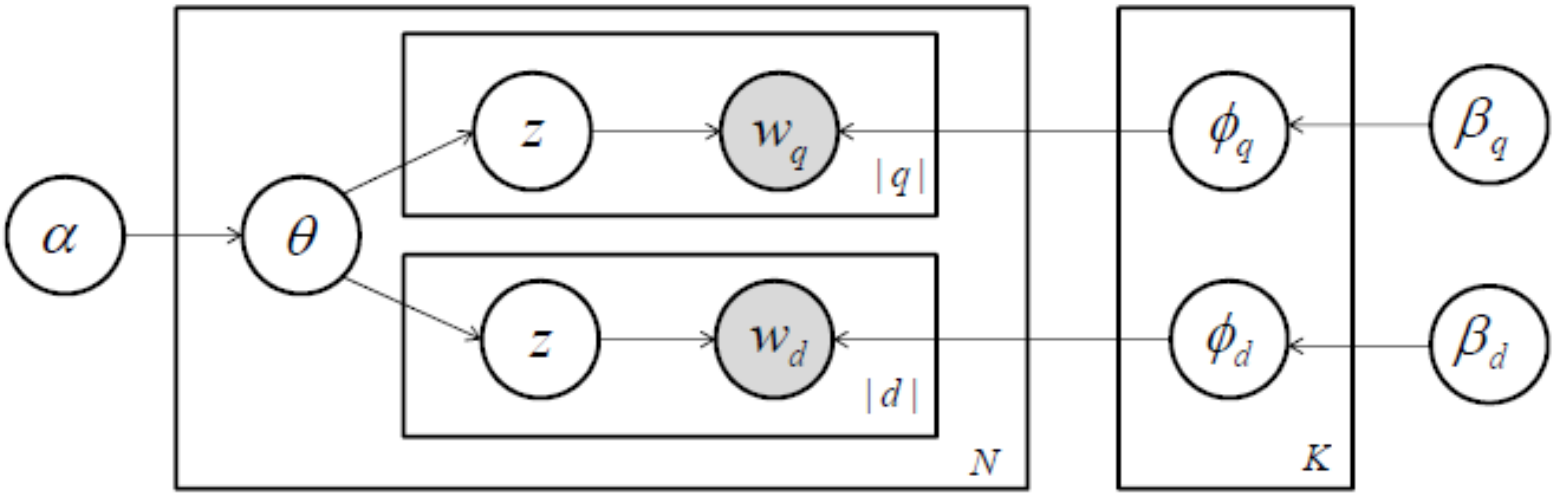


Figure 7.2: Graphical representation of bilingual topic model.

BLTM生成过程

1. Each topic z is represented by a distribution of query words ϕ_q and a distribution of document words ϕ_d. The distributions are selected with Dirichlet priors with parameters β_q and β_d. There are K topics.

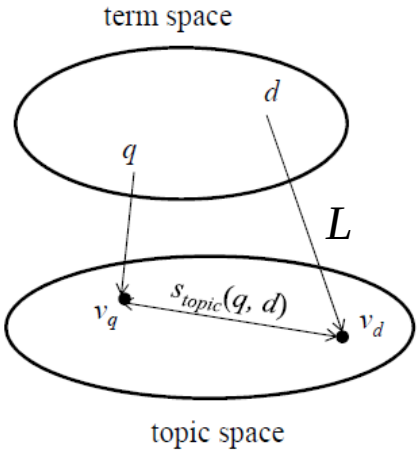
2. For each query-document pair q and d, a distribution of topics θ is selected with Dirichlet prior with parameter α. There are N query-document pairs.

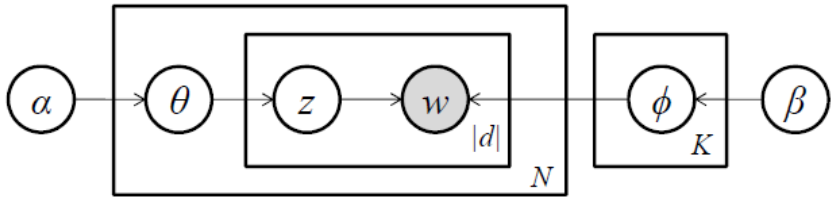
3. In generation of each query, topic z is first selected according to distribution θ, and then query word w_q is selected according to distribution ϕ_q of topic z. There are $|q|$ query words.

4. In generation of each document, topic z is first selected according to distribution θ, and then document word w_d is selected according to distribution ϕ_d of topic z. There are $|d|$ document words.

PLS vs. BLTM as LSI vs. LDA

- LSI vs. LDA: L⇔ ϕ





- PLS vs. BLTM: $(L_q, L_d) \Leftrightarrow (\phi_q, \phi_{\mathrm{d}})$

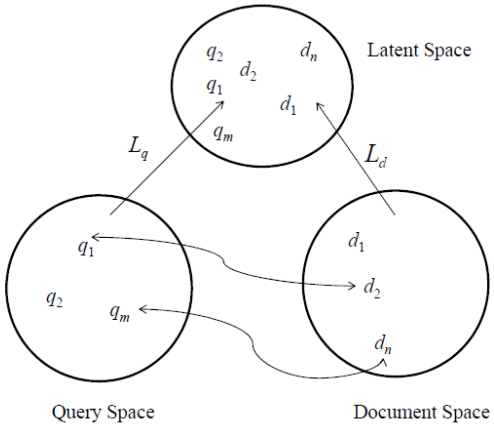


Figure 7.1: Matching in latent space.

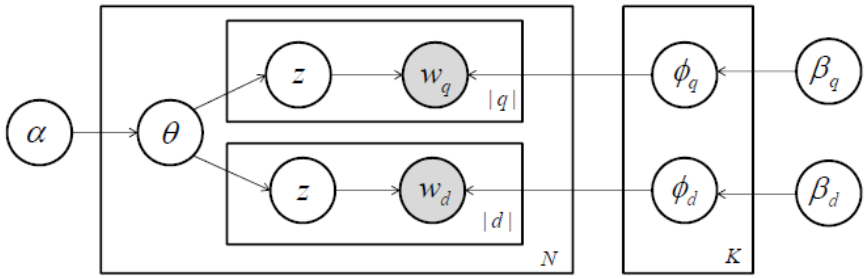


Figure 7.2: Graphical representation of bilingual topic model.

统计机器翻译

- 源语言$C = c_1 c_2 \cdots c_J$➔目标语言$E = e_1 e_2 \cdots e_I$
- 翻译模型

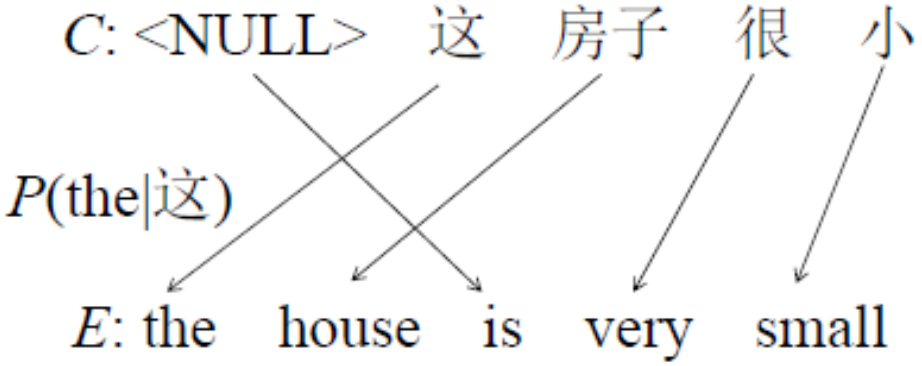
$$E^* = \text{argmax}_E \, P(E|C)$$

- 分解$P(E|C)$

$$\begin{aligned} E^* &= \arg\max_E \frac{P(C|E)P(E)}{P(C)} \\ &= \arg\max_E P(C|E)P(E) \end{aligned}$$

翻译模型　　语言模型

IBM Model One (Brown et al., 1993)



$$P(E|C) = \frac{\epsilon}{(J+1)^I} \prod_{i=1}^{I} \sum_{j=0}^{J} P(e_i|c_j)$$

- 生成过程

1. Choose the length of target sentence I, according to distribution $P(I|C)$
2. For each position $i (i = 1, 2, \cdots, I)$
   (a) Choose position j in the source sentence C according to $P(j|C)$
   (b) Generate target word e_i according to $P(e_i|c_j)$

基于机器翻译的搜索匹配：基本模型 (Berger & Lafferty, '99)

- 源语言：查询$q = q_1 q_2 \cdots q_m$
- 目标语言：文档$d = d_1 d_2 \cdots d_n$
- 匹配度：查询字符串“翻译”成文档字符串的概率

$$P(d|q) \propto P(q|d)P(d)$$

“翻译”概率　　语言模型概率

模型训练数据的搜集

- Query-title pairs in click-through data (Gao et al., ’10)

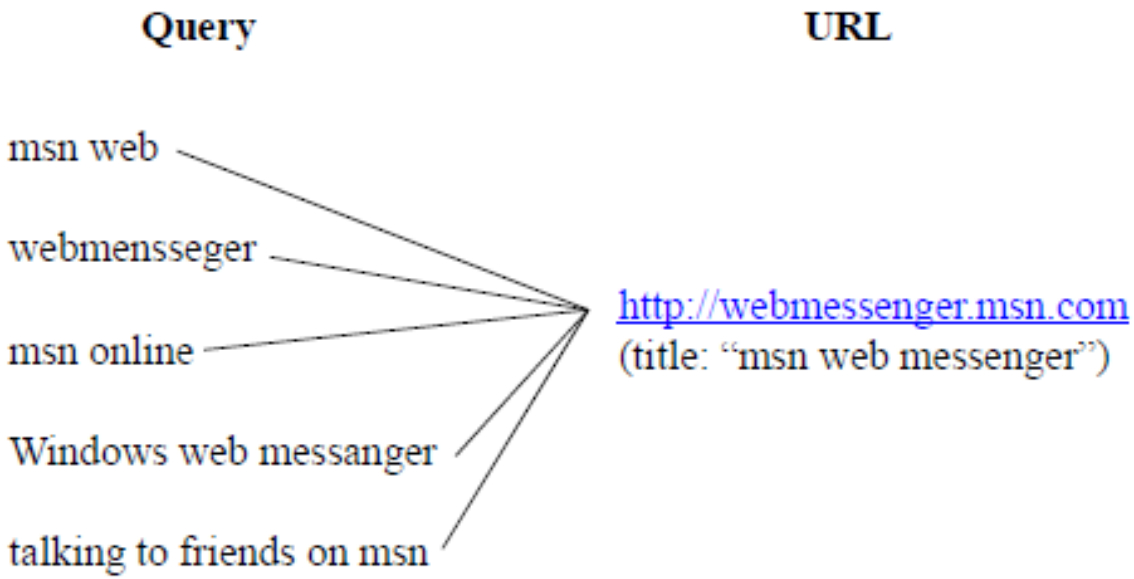


Figure 5.2: An example of URL and its associated queries extracted from click-through data.

为何能够解决语义匹配的问题?

q	$t(q \mid w)$
solzhenitsyn	0.319
citizenship	0.049
exile	0.044
archipelago	0.030
alexander	0.025
soviet	0.023
union	0.018
komsomolskaya	0.017
treason	0.015
vishnevskaya	0.015

$w =$ solzhenitsyn

q	$t(q \mid w)$
carcinogen	0.667
cancer	0.032
scientific	0.024
science	0.014
environment	0.013
chemical	0.012
exposure	0.012
pesticide	0.010
agent	0.009
protect	0.008

$w =$ carcinogen

q	$t(q \mid w)$
zubin_mehta	0.248
zubin	0.139
mehta	0.134
philharmonic	0.103
orchestra	0.046
music	0.036
bernstein	0.029
york	0.026
end	0.018
sir	0.016

$w =$ zubin

q	$t(q \mid w)$
pontiff	0.502
pope	0.169
paul	0.065
john	0.035
vatican	0.033
ii	0.028
visit	0.017
papal	0.010
church	0.005
flight	0.004

$w =$ pontiff

q	$t(q \mid w)$
everest	0.439
climb	0.057
climber	0.045
whittaker	0.039
expedition	0.036
float	0.024
mountain	0.024
summit	0.021
highest	0.018
reach	0.015

$w =$ everest

q	$t(q \mid w)$
wildlife	0.705
fish	0.038
acre	0.012
species	0.010
forest	0.010
environment	0.009
habitat	0.008
endangered	0.007
protected	0.007
bird	0.007

$w =$ wildlife

与传统机器翻译模型的区别

- 查询与文档实际为同一种语言，共享一个词表
- 存在大量的查询词与文档词精确匹配的情况

考虑自翻译(self-translation)

- 基本模型

$$P(q|d) = \epsilon \prod_{j=1}^{m} \left(\frac{n}{n+1} P(q_j|d) + \frac{1}{n+1} P(q_j|null) \right)$$

文档d翻译成查询词q_j:
$P(q_j|d) = \sum_{i=1}^{n} P(q_j|d_i)$

平滑概率：防止0概率发生

- 加入自翻译$P(q_j|d) \Rightarrow P'(q_j|d)$

$$P'(q_j|d) = \beta Q(q_j|d) + (1-\beta) \sum_{w \in d} P(q_j|w) Q(w|d)$$

- 效果：当查询词与文档词精确匹配时，强调精确匹配的效果

模型效果

Table 5.1: Performances of word-based translation models in search.

	NDCG@1	NDCG@3	NDCG@10
BM25 (baseline)	0.3181	0.3413	0.4045
WTM (without self-translation)	0.3210	0.3512	0.4211
WTM (with self-translation)	**0.3310**	**0.3566**	**0.4232**