适应 Web 检索的平滑型排序支持向量机*

何海江

(长沙学院 计算机科学与技术系 长沙 410003)

摘 要 代价敏感的排序支持向量机将样本的排序问题转换为样本对的分类问题,以适应 Web 信息检索. 然而急剧膨胀的训练样本对使得学习时间过长. 为此,文中提出一种支持二次误差的代价敏感的平滑型排序支持向量机(cs-sRSVM),用分段多项式光滑函数近似铰链损失函数,将优化目标转变为无约束问题. 再由 Newton-YUAN 算法求无约束问题的唯一最优解. 在排序学习公开数据集 LETOR 的实验表明,cs-sRSVM 与已有的代价敏感排序算法相比,训练时间更短,而检索性能同样出色.

关键词 代价敏感,排序支持向量机(RSVM),二次误差,信息检索,平滑
中图法分类号 TP 393

Smooth Ranking Support Vector Machine Adapting to Web Retrieval

HE Hai-Jiang
(*Department of Computer Science and Technology*, *Changsha University*, *Changsha* 410003)

ABSTRACT

Cost-sensitive ranking support vector machine converts the order relation of samples into the classification relation of sample pairs, and it is particularly well suited to web information retrieval. However, learning large amounts of sample pairs takes extremely long time. A cost-sensitive smooth ranking support vector machine (cs-sRSVM) using 2-Norm error is presented. Firstly, the optimization object is transformed into unconstrained problem. Secondly, the smooth piecewise polynomial function is approximated to the hinge loss function. Finally, the unique optimal solution is obtained by applying Newton-YUAN method. The experimental results on a public dataset LETOR show that the training time of cs-sRSVM is faster than that of the existing cost-sensitive ranking algorithm, and its retrieval performance is equally impressive.

Key Words Cost-Sensitive, Ranking Support Vector Machine (RSVM), 2-Norm Error, Information Retrieval, Smoothness

* 湖南省自然科学基金项目(No. 06JJ2065)、湖南省教育厅科学研究项目(No. 09C123)资助
收稿日期:2008-09-22;修回日期:2009-03-02
作者简介 何海江,男,1970 年生,副教授,主要研究方向为 Web 挖掘、数据仓库. E-mail: haijianghe@sohu.com.

1 引 言

在 Web 信息检索系统中,用户提交查询后,系统按照相关度从高到低返回一系列 Web 对象(文档).构造排序模型是相关度计算的关键环节,而基于统计的学习算法是已知的排序模型高效构造方法,成为近年来的研究热点.排序学习广泛应用于搜索引擎优化[1]、文档自动摘要[2]、缩略词提取[3]等 Web 检索领域.

从训练样本划分,排序学习大致有三类方式:单样本、样本对和多样本.单样本方式在单个文档上构造排序损失函数,排序学习问题转化成顺序回归[4]或分类[5]问题.样本对方式则将损失函数建立在成对的文档上,或者将查询内不同优先级别的文档实例两两组合成一个样本,排序学习由此变为二元分类,如排序支持向量机(Ranking Support Vector Machine, RSVM)[1,6-8]和神经元的 RankNet[9].或用文档实例对的概率模式反应排序模型的不确定性,如结合高斯过程的 SoftRank GP[10].或干脆基于已有样本对方式排序算法,加装一个非凸优化的元排序学习工具[11].多样本方式将单个查询的所有返回文档视作一个实例,在文档队列上直接建立 Listwise 损失函数[12].除此之外,结合以上样本方式的结构学习也获得成功,如直接优化任意排序性能的 Committee Perceptron[13],每次迭代过程都单独处理不同查询的 IsoRank[14].

互联网是一个巨大的信息库,用户往往只关注检索系统排列在最前面的少部分文档[1].因此不同相关度对象的误判引起的误差大相径庭,排序学习算法必须考虑到这一点[7-8].训练集不同查询返回文档个数极不平衡,学习时同样不能忽略[8].代价敏感的排序学习是解决这两个问题的有效途径.代价敏感学习以最小化总体代价为优化目标,训练方法大致可分为三类:1)在机器学习算法中直接嵌入代价因子[7-8];2)重构训练样本的类别分布[15];3)包装式的元学习算法[16].

文献[7]和文献[8]实现了第 2 节中问题 1 的一次损失(式(1)的 $\theta=1$)代价敏感排序支持向量机,但是巨量的不等式约束使得学习时间超长,不利于排序模型的优化.而我们提出的二次损失(式(1)的 $\theta=2$)代价敏感排序支持向量机(简称为 cs-sRSVM),利用分段多项式光滑函数近似铰链损失函数 $(1-x)_+$,将约束的数学规划转换为无约束优化问题,最后由 Newton-YUAN 算法直接求解无约束问题.cs-sRSVM 的 Web 检索性能同样好,而训练时间更短,这在排序学习公开数据集 LETOR[17] 上得到验证.

2 二次损失代价敏感排序支持向量机

依据结构风险最小化原则,以上排序问题的优化目标[1,6-8]为问题 1:

$$\min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}*\boldsymbol{w}+C\sum \lambda_{r(k,i,j)}\mu_{q(k)}\xi_{k,i,j}^{\theta},$$

使得

$$\begin{aligned}&\forall (d_{1,i} >_* d_{1,j}),\\ &\quad \boldsymbol{w}*\phi(q_1,d_{1,i}) \geqslant \boldsymbol{w}*\phi(q_1,d_{1,j})+1-\xi_{1,i,j};\\ &\quad \vdots\\ &\forall (d_{m,i} >_* d_{m,j}),\\ &\quad \boldsymbol{w}*\phi(q_m,d_{m,i}) \geqslant \boldsymbol{w}*\phi(q_m,d_{m,j})+1-\xi_{m,i,j};\\ &\forall k,i,j,\xi_{k,i,j}\geqslant 0,\end{aligned} \tag{1}$$

其中,ϕ 是文档的特征映射函数,$\boldsymbol{w}$ 是排序模型 f 的特征权向量,$*$ 是内积运算,$>_*$ 是优先关系,$\xi_{k,i,j}$ 是使约束条件成立的误差标量,C 是误差和权向量之间的平衡因子,$\lambda_{r(k,i,j)}$ 是错判 z_{ki} 和 z_{kj} 顺序的惩罚系数,$\mu_{q(i)}$ 使得决策函数偏向返回文档较少的查询.

当 $\lambda=\mu=\theta=1$ 时,问题 1 退化为非代价敏感的排序支持向量机 RankSVM[1,6].文献[7]实现 $\lambda=1$、$\theta=1$ 的问题.文献[8]实现 $\theta=1$ 的问题.我们把后两者的算法统称为 RSVM-QL.而 cs-sRSVM 解决 $\theta=2$ 的问题.

2.1 多项式光滑函数

式(1)可重写为

$$\boldsymbol{w}*[\phi(q_k,d_{ki})-\phi(q_k,d_{kj})]\geqslant 1-\xi_{k,i,j},$$

将样本对 d_{ki} 和 d_{kj} 组合成一个样本

$$x_{k,i,j}=\phi(q_k,d_{ki})-\phi(q_k,d_{kj}),$$

并定义

$$y_{k,i,j}=\begin{cases}1, & d_{ki} >_* d_{kj}\\ -1, & d_{kj} >_* d_{ki}\end{cases}$$

t 是两两组合后的样本对数目,$\theta=2$ 的问题 1 转换为问题 2:

$$\min_{\boldsymbol{w}} \frac{1}{2}\boldsymbol{w}*\boldsymbol{w}+C\sum_{i=1}^{t}\lambda_i\mu_i\xi_i^2, \tag{2}$$

使得

$$y_i(\boldsymbol{w}*x_i)\geqslant 1-\xi_i,\ \forall \xi_i\geqslant 0,\ i=1,2,\cdots,t. \tag{3}$$

参考平滑支持向量机[18],合并式(2)、(3),引入 $(1-x)_+=\max(0,1-x)$.问题 2 可变为无约束问题 3:

$$\min_{\boldsymbol{w}} h(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w} * \boldsymbol{w} + C\sum_{i=1}^{t}\lambda_i\mu_i((1 - y_i(\boldsymbol{w} * x_i))_+)^2,$$
$$i = 1,2,\cdots,t. \tag{4}$$

$\boldsymbol{w}$ 不可能为零向量,且上式各个子项都为平方项,显然 $h(\boldsymbol{w})$ 为严格的凸函数,问题 3 有唯一最优解. $(1-x)_+$ 不可微,为了直接用 Newton 法求解问题 3,借鉴分段多项式光滑函数[19],我们定义 $p(x,\eta)$ 近似 $(1-x)_+$:

$$p(x,\eta) = \begin{cases} 1 - x, & x \leqslant 1 - \eta \\ -\dfrac{1}{16\eta^3}(x + 3\eta - 1)(x - 1 - \eta)^3, & 1 - \eta < x < 1 + \eta \\ 0, & x \geqslant 1 + \eta \end{cases}$$

其中 $0 < \eta < 1$. 由上式容易验证 $p(x,\eta)$ 具有二阶光滑性;η 越小,在区间 $(1-\eta,1+\eta)$ 越逼近 $(1-x)_+$. 如图 1 所示,$p(x,0.2)$ 比 $p(x,0.4)$ 更好地近似 $(1-x)_+$.

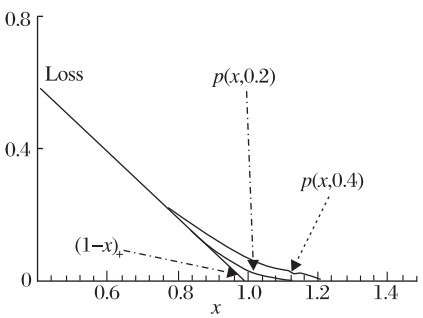

图 1　$p(x,\eta)$ 近似 $(1-x)_+$

Fig. 1　$p(x,\eta)$ approximating to $(1-x)_+$

定理 1　1) $p(x,\eta) \geqslant (1-x)_+$.

2) $p(x,\eta)^2 - (1-x)_+^2 \leqslant 0.0515\eta^2$.

证明　先证明 1).

(1) 当 $x \geqslant 1 + \eta$ 或 $x \leqslant 1 - \eta$ 时,$p(x,\eta) = (1-x)_+$,显然成立.

(2) $x \in (1-\eta,1)$ 时,设

$$pf(x) = p(x,\eta) - (1-x)_+ = p(x,\eta) - (1-x),$$
$$pg(x) = (x - 1 - \eta)^3 + 3(x - 1 - \eta)^2(x + 3\eta - 1).$$

则有

$$\nabla pf(x) = -\frac{1}{16\eta^3}pg(x) + 1,$$
$$\nabla pg(x) = 12((x-1)^2 - \eta^2) < 0,$$

因此,$pg(x)$ 严格单调递减,

$$8\eta^3 < pg(x) < 16\eta^3,\quad 0 < \nabla pf(x) < 0.5.$$

$pf(x)$ 严格单调递增,

$$0 < pf(x) < \frac{3\eta}{16}.$$

(3) $x \in (1,1+\eta)$ 时,设

$$ph(x) = p(x,\eta) - (1-x)_+ = p(x,\eta).$$

由(2) 可知

$$\nabla pg(x) = -\frac{1}{16\eta^3}pg(x),$$

且 $-1 < \nabla pg(x) < -0.5$. 故 $ph(x)$ 严格单调递减,

$$0 < pf(x) < \frac{3\eta}{16}.$$

综上所述,定理 1 的 1) 得证.

再证明定理 1 的 2).

(1) $x \geqslant 1 + \eta$ 或 $x \leqslant 1 - \eta$ 时,显然成立.

(2) 由 1) 可知,$x \in (1,1+\eta)$ 时,$p(x,\eta) > 0$ 且严格单调递减,故

$$p(x,\eta)^2 - ((1-x)_+)^2 = p(x,\eta)^2,$$

严格单调递减,

$$p(x,\eta)^2 < \frac{9\eta^2}{256} < 0.0515\eta^2.$$

(3) $x \in (1-\eta,1)$ 时,令

$$pf(x) = p(x,\eta)^2 - (1-x)^2,$$

其导数为 0 处取得最大值,由二分法求得此时 $x^* = 1 - 0.18895\eta$,则 $pf(x^*) \approx 0.0515\eta^2$.

综上所述,定理 1 的 2) 得证.

2.2　本文算法模型和收敛特性

将 $p(x,\eta)$ 代入式(4),cs-sRSVM 的优化目标为问题 4:

$$\min_{\boldsymbol{w}} \psi_\eta(\boldsymbol{w}) = \frac{1}{2}\boldsymbol{w} * \boldsymbol{w} + C\sum_{i=1}^{t}\lambda_i\mu_i p(y_i(\boldsymbol{w} * x_i),\eta)^2.$$

$\psi_\eta(\boldsymbol{w})$ 是严格的凸函数,问题 4 有唯一最优解. 随着 η 趋向于零,上式的最优解收敛于式(4) 的最优解.

定理 2　若 $\boldsymbol{w}^*$ 是问题 3 的最优解,$\boldsymbol{w}^\#$ 是问题 4 的最优解,则有

$$(\boldsymbol{w}^\# - \boldsymbol{w}^*) * (\boldsymbol{w}^\# - \boldsymbol{w}^*) = (\boldsymbol{w}^\# - \boldsymbol{w}^*)^2 \leqslant C\sum_{i=1}^{t}\lambda_i\mu_i \times 0.0515\eta^2.$$

证明　由一阶最优性条件及 $h(\boldsymbol{w})$、$\psi_\eta(\boldsymbol{w})$ 的强凸性,可知

$$h(\boldsymbol{w}^\#) - h(\boldsymbol{w}^*) \geqslant \Delta h(\boldsymbol{w}^*)(\boldsymbol{w}^\# - \boldsymbol{w}^*) + \frac{1}{2}(\boldsymbol{w}^\# - \boldsymbol{w}^*)^2 = \frac{1}{2}(\boldsymbol{w}^\# - \boldsymbol{w}^*)^2,$$

$$\psi_\eta(\boldsymbol{w}^*) - \psi_\eta(\boldsymbol{w}^\#) \geqslant \nabla\psi_\eta(\boldsymbol{w}^\#)(\boldsymbol{w}^* - \boldsymbol{w}^\#) + \frac{1}{2}(\boldsymbol{w}^* - \boldsymbol{w}^\#)^2 = \frac{1}{2}(\boldsymbol{w}^* - \boldsymbol{w}^\#)^2.$$

两式相加,得

$$(\boldsymbol{w}^{\#}-\boldsymbol{w}^{*})^2 \leqslant \psi_\eta(\boldsymbol{w}^{*})-h(\boldsymbol{w}^{*})-(\psi_\eta(\boldsymbol{w}^{\#})-h(\boldsymbol{w}^{\#})).$$

由定理1的1)知$\psi_\eta(\boldsymbol{w}^{\#}) \geqslant h(\boldsymbol{w}^{\#})$,并将定理1的2)代入上式,故得

$$(\boldsymbol{w}^{\#}-\boldsymbol{w}^{*})^2 \leqslant \psi_\eta(\boldsymbol{w}^{*})-h(\boldsymbol{w}^{*}) \leqslant C\times\sum_{i=1}^{t}\lambda_i\mu_i\times 0.0515\eta^2.$$

证毕.

2.3 Newton-YUAN 算法

由于$\psi_\eta(\boldsymbol{w})$二次可微,结合Newton法和YUAN[20]的一维精确搜索算法求解cs-sRSVM,我们称之为Newton-YUAN算法. 具体步骤如下.

step 1 初始迭代点为$\boldsymbol{w}^{(1)}$,记$k=1$. 多项式平滑参数

$$\eta=\sqrt{\frac{\varepsilon_1}{0.0515tC}},$$

ε_1是$h(\boldsymbol{w})$最优解与$\psi_\eta(\boldsymbol{w})$最优解的最大误差.

step 2 计算$\psi_\eta(\boldsymbol{w})$的梯度$\boldsymbol{g}^{(k)}=\nabla\psi_\eta(\boldsymbol{w}^{(k)})$. 如果$\sqrt{\boldsymbol{g}^{(k)}*\boldsymbol{g}^{(k)}}\leqslant\varepsilon_2$,则迭代终止,$\boldsymbol{w}^{(k)}$为近似最优解;否则继续.

step 3 由Hesse矩阵和梯度计算迭代方向$\boldsymbol{d}^{(k)}$,并用Cholesky分解法解线性方程组:

$$\nabla^2\psi_\eta(\boldsymbol{w}^{(k)})\boldsymbol{d}^{(k)}=-\nabla\psi_\eta(\boldsymbol{w}^{(k)}).$$

step 4 如果方向变动过小,$\sqrt{\boldsymbol{d}^{(k)}*\boldsymbol{d}^{(k)}}\leqslant\varepsilon_3$,则迭代终止,$\boldsymbol{w}^{(k)}$为近似最优解;否则继续.

step 5 YUAN的最速下降法步长选择公式[17]计算Newton法迭代步长,若$\mathrm{mod}(k,3)=0$,则

$$\boldsymbol{w}^{(k+1)}=\boldsymbol{w}^{(k)}+\frac{\boldsymbol{d}^{(k)}}{\sqrt{\dfrac{\sqrt{\boldsymbol{g}^{(k)}*\boldsymbol{g}^{(k)}}}{\sqrt{(\boldsymbol{w}^{(k+1)}-\boldsymbol{w}^{(k)})^2}}}+1},$$

反之,$\mathrm{mod}(k,3)\neq 0$,则$\boldsymbol{w}^{(k+1)}=\boldsymbol{w}^{(k)}+\boldsymbol{d}^{(k)}$. 令$k=k+1$,转到step 2.

3 Web检索评估测度

评估测度NDCG(Normalized Discounted Cumulative Gain)[21]广泛应用于Web信息检索领域,尤其适应级别个数$z>2$的情形. 有查询实际返回文档级别序列G,最理想级别序列为G^*,计算CG和CG^*,

$$CG[n]=\begin{cases}G[1], & n=1\\ CG[n-1]+G[n], & n>1\end{cases} \tag{5}$$

再计算DCG和DCG^*,

$$DCG[n]=\begin{cases}CG[1], & n=1\\ DCG[n-1]+\dfrac{G[n]}{\log_2 n}, & n>1\end{cases} \tag{6}$$

则

$$NDCG@n=\frac{DCG[n]}{DCG^*[n]}.$$

例如,查询q_i返回的文档级别序列为$G=\langle 2,0,1,1\rangle$,显然q_i的最优级别序列为$G^*=\langle 2,1,1,0\rangle$,由式(5)、(6)计算得

$$CG=\langle 2,2,3,4\rangle,\quad CG^*=\langle 2,3,4,4\rangle,$$
$$DCG=\langle 2,2,2.63,3.13\rangle,$$
$$DCG^*=\langle 2,3,3.63,3.63\rangle,$$
$$NDCG=\langle 1,0.667,0.725,0.862\rangle.$$

$@n$指序列的位置,$NDCG@2=0.667$.

准确率$P@n$和宏平均准确率(Mean Average Precision, MAP)只适合$z=2$的情形,q_i返回的文档只有相关($r_1=0$)和不相关($r_2=1$)两个级别. 有m个查询,令Dqs是q_i返回的文档集,$|Dqs|$表示文档集个数,$\#Dq(j)$是前j个文档内相关文档个数,$Rel(j)$是第j个文档的相关度级别,则有

$$P@j=\frac{\#Dq(j)}{j},\quad AvgP(q_i)=\sum_{j=1}^{|Dqs|}\frac{P@j\times \mathrm{Re}l(j)}{\#Dq(|Dqs|)},\quad MAP=\frac{1}{m}\sum_{i=1}^{m}AvgP(q_i).$$

4 实验结果及分析

RankSVM和RSVM-QL都通过求解对偶问题得到决策函数,只不过约束不等式的拉格朗日乘子范围不同. 前者的乘子从同一区间取值,而后者乘子范围与问题1的λ和μ有关. RSVM-QL对偶问题求解算法的参数调整细节并不公开,我们用$\mathrm{SVM}^{\mathrm{Light}}$软件包(源代码可从http://www.cs.cornell.edu/People/tj/svm%5Flight下载)的代价敏感功能实现. RankSVM和RSVM-QL的对偶问题求解算法参数都采用$\mathrm{SVM}^{\mathrm{Light}}$的缺省值. 而cs-sRSVM的Newton-YUAN算法的参数设为$\varepsilon_1=10^{-5}$,$\varepsilon_2=\varepsilon_3=10^{-4}$.

4.1 测试数据集

LETOR是比较排序学习算法的公开数据集[6],我们用其中的OHSUMED和TD2004完成3种算法的对比实验. LETOR OHSUMED包括106个查询,所有查询返回的文档都被人工标注为不相关($r_1=0$)、部分相关($r_2=1$)、相关($r_3=2$). 文档的25个特征既有词频、文档长度等基本特征,又有BM25等复合

特征. LETOR TD2004 包括75 个查询,返回的文档与查询要么相关($r_1=0$),要么不相关($r_2=1$),44 个特征反映标题、URL 地址、超链接等内容.

OHSUMED 和 TD2004 被划分为 5 个部分:S1、S2、S3、S4 和 S5. 所有实验结果,除非特别声明,都是五折交叉的平均值. 表 1 是五折的划分情况.

表 1　数据集五折划分表

Table 1　Five folds of datasets

Folds	训练集	验证集	测试集	OHSUMED 训练集样本对数	TD2004 训练集样本对数
Fold1	S1	S2	{S3,S4,S5}	80889	69406
Fold2	S2	S3	{S4,S5,S1}	123612	72512
Fold3	S3	S4	{S5,S1,S2}	163162	126149
Fold4	S4	S5	{S1,S2,S3}	135942	45110
Fold5	S5	S1	{S2,S3,S4}	78983	124395

4.2　Web 检索性能比较

在 TD2004 上 $\lambda=1$,而在 OHSUMED 上,3 种不同的样本对错误排序惩罚系数不同. 当 r_1 级别的文档排在 r_2 级别的文档前,$\lambda_{(0,1)}=1$,另两种错误排序的损失显然大得多,设 $\lambda_{(1,2)}=1.3$,$\lambda_{(0,2)}=2$. 我们将在 4.4 小节讨论 λ 对算法性能的影响. 同一查询所有样本对的 μ 相等,即

$$\mu_{q_i}=\log_e\left(1+\frac{\max\limits_{i=1,\cdots,m}(q_i\text{ 包括的样本对数})}{q_i\text{ 包括的样本对数}}\right).$$

平衡因子 C 与核关联,本文实验全部采用线性核,实际上非线性核并不能提高 LETOR 的检索性能. 有许多方法选择 C,我们采用类似于 UD[22] 的两阶段选择方法:以 AvgNDCG 为优化目标,AvgPrec 和 MAP 亦可充当优化目标,AvgNDCG 和 AvgPrec 的定义如下:

$$AvgNDCG=\sum_{i=1}^{20}\frac{NDCG@i}{20},\ AvgPrec=\sum_{i=1}^{20}\frac{P@i}{20}.$$

第一阶段,分别令 C 为 10^{-5}、10^{-4}、10^{-3}、10^{-2} 和 10^{-1} 学习训练集,计算排序模型在验证集的 AvgNDCG,令 C^* 为当前最佳 C,使得排序模型在验证集获得最大 AvgNDCG 值. 第二阶段,再以 C 为 $0.6C^*$、$0.8C^*$、$1.2C^*$、$1.4C^*$ 学习训练集,求出验证集的 AvgNDCG,与 C^* 比较后得到最佳 C.

表 2 是 OHSUMED 第 2 折的比较数据. 表 3 是 TD2004 第 1 折的比较数据. 显然,3 种算法有不同的最佳 C, AvgNDCG、MAP、AvgPrec 是该 C 值时学习到的排序模型在测试集取得的结果. 同样地,在各自最佳 C 值时获得文章其余实验数据.

表 2　OHSUMED 第 2 折时 3 种算法比较

Table 2　Comparison among 3 algorithms on OHSUMED Fold2

算法	最佳 C	AvgPrec	MAP	AvgNDCG
RankSVM	1.0e−3	0.333	0.307	0.482
RSVM-QL	1.2e−3	0.339	0.310	0.501
cs-sRSVM	8.0e−3	0.341	0.312	0.502

表 3　TD2004 第 1 折时 3 种算法比较

Table 3　Comparison among 3 algorithms on TD2004 Fold1

算法	最佳 C	AvgPrec	MAP	AvgNDCG
RankSVM	1.2e−5	0.182	0.271	0.336
RSVM-QL	6.0e−5	0.194	0.281	0.358
cs-sRSVM	1.0e−4	0.208	0.292	0.388

图 2 是 OHSUMED 上 3 种算法 NDCG 的比较. 由于 cs-sRSVM 和 RSVM-QL 都是代价敏感的,显然优于 RankSVM,说明引入 λ 和 μ 能够改善 Web 检索性能. 而 cs-sRSVM 只比 RSVM-QL 略微占优,但没有明显差别. 检索性能应该与数据集有关,毕竟这 2 种算法只是损失次数 θ 不同,差别可被 C 消除. TD2004 上得到的结果与 OHSUMED 相同.

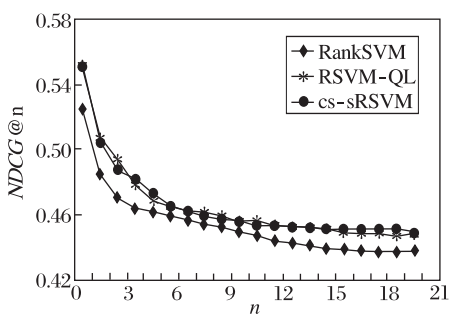

图 2　3 种算法在 OHSUMED 上的 NDCG 比较

Fig. 2　Comparison of NDCG among 3 algorithms on OHSUMED

即使以 AvgNDCG 为优化目标,由表 3 可知,RankSVM 的 MAP 和 P@n 也不如另两个算法,只在 $P@5$、$P@7$、$P@9$ 处 RankSVM 高过 RSVM-QL. 相比 cs-sRSVM,RankSVM 则始终较低. OHSUMED 上得到类似结果. 因为 MAP 和 P@n 是两级别评价指标,计算时,OHSUMED 的部分相关文档归于不相关类.

4.3　训练时间比较

cs-sRSVM 求解无约束问题,显然比 RSVM-QL 和 RankSVM 快得多. 相比 RankSVM,RSVM-QL 要慢一些,因为其对偶问题的约束更严格. cs-sRSVM 的训练时间与两个因素相关:1) Hesse 矩阵和梯度的计算,样本对 t 越大,每次迭代的时间越长;2) 线性方程组系数矩阵的特征值,特征值的极大值与极小

值之比可能决定迭代次数. 对 cs-sRSVM 来说,除 TD2004 第 3 折和第 5 折的 Newton-YUAN 算法迭代 12 次外,其余的迭代次数在 4 ~ 6 之间.

表 4 TD2004 上 MAP 和 $P@n$ 的比较

Table 4 Comparison of MAP and $P@n$ among 3 algorithms on TD2004

算法	RankSVM	RSVM-QL	cs-sRSVM
MAP	0.332	0.337	0.341
$P@1$	0.372	0.391	0.381
$P@2$	0.349	0.360	0.379
$P@3$	0.322	0.338	0.334
$P@4$	0.301	0.304	0.315
$P@5$	0.281	0.276	0.289
$P@6$	0.264	0.264	0.276
$P@7$	0.253	0.251	0.268
$P@8$	0.241	0.241	0.247
$P@9$	0.233	0.232	0.237
$P@10$	0.221	0.225	0.226

表 5 是 3 种算法的部分训练时间(除去从文件读取训练集数据的时间)对比,取选择最佳 C 时 9 次训练时间的平均值. RankSVM 训练时间一般不到 RSVM-QL 的 1/2. TD2004 第 3 折和第 5 折时,cs-sRSVM 比 RankSVM 快 200 多倍,比 RSVM-QL 快 300 多倍. 其余的实验中,cs-sRSVM 训练时间不到 RSVM-QL 的千分之一,甚至 1/8000. Web 信息库巨大而复杂,要学习到人们的行为特征,训练集会越来越大,cs-sRSVM 更能凸显其训练快的优势.

表 5 3 种算法的训练时间对比

Table 5 Training time comparison among 3 algorithm

s

数据集及 Fold	RankSVM	RSVM-QL	cs-sRSVM
OHSUMED, Fold3	7604	17385	2.3
OHSUMED, Fold5	2710	7052	1.2
TD2004, Fold4	1756	3497	2.9
TD2004, Fold5	1803	2562	8.2

就单个查询而言,新训练集样本对数目呈二次增长,若增加所有查询的返回文档,训练时间二次增长. 但查询间的文档并不比较,增加查询而导致文档数增多时,样本对数目呈线性增长,训练时间与文档数为近似的线性关系. 表 6 是 OHSUMED 上 5 次实验的训练时间对比,Trial1 选 S1 为训练集,Trial2 选 S1 和 S2,Trial3 选 S1、S2 和 S3,依次类推.

表 6 OHSUMED 上 5 次实验的训练时间对比

Table 6 Training time comparison of 5 tests on OHSUMED

实验	Train1	Train2	Train3	Train4	Train5
样本数	2570	5646	9219	12757	16140
训练时间/s	1.14	4.04	6.22	8.39	9.75

4.4 λ 的影响

λ 是代价敏感学习的一个重要因素,我们在 OHSUMED 第 2 折考察 λ 对 cs-sRSVM 检索性能的影响. 图 3 固定 $\lambda_{(0,2)}=2$,$\lambda_{(1,2)}$ 从 1 递增到 2. 图 4 固定 $\lambda_{(1,2)}=2$,$\lambda_{(0,2)}$ 从 2 递增到 7. C 始终等于最佳 C 值 0.008. 算法学习训练集后,直接在测试集获得实验数据. 一般来说,$\lambda_{(0,2)}$ 和 $\lambda_{(1,2)}$ 越大,级别为 2 的文档越可能排在前面,越能改善较小 n 位置的 $NDCG@n$ 和 $P@n$. 但太大的 $\lambda_{(0,2)}$、$\lambda_{(1,2)}$ 又导致过学习. 训练集相关度级别为 0、1、2 的文档个数分布情况将决定这 2 个参数. 将 $\lambda_{(0,2)}$、$\lambda_{(1,2)}$ 固定为较大的值,并使得 $\lambda_{(0,2)} > \lambda_{(1,2)}$ 是一个不错的选择.

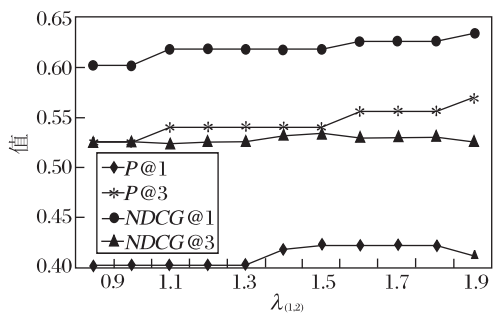

图 3 $\lambda_{(0,2)}=2$ 时,$\lambda_{(1,2)}$ 对 cs-sRSVM 检索性能的影响

Fig. 3 Effect of $\lambda_{(1,2)}$ on retrieval performance of cs-sRSVM when $\lambda_{(0,2)}=2$

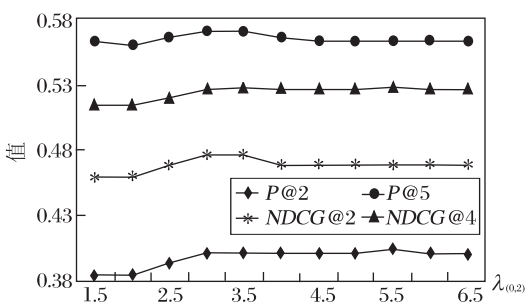

图 4 $\lambda_{(1,2)}=2$ 时,$\lambda_{(0,2)}$ 对 cs-sRSVM 检索性能的影响

Fig. 4 Effect of $\lambda_{(0,2)}$ on retrieval performance of cs-sRSVM when $\lambda_{(1,2)}=2$

5 结束语

已有的代价敏感排序支持向量机都由对偶问题

求解决策函数,而学习巨量的 Web 信息时受到不等式的过多约束,收敛速度减慢,训练时间变长. 我们提出的 cs-sRSVM,支持二次误差函数,同样属于样本对方式,同样允许代价敏感(直接嵌入代价因子),将约束问题转化为无约束问题,利用二次可微的多项式光滑函数近似铰链损失函数后,得以由 Newton-YUAN 算法直接解线性方程组. 与现有算法相比,cs-sRSVM 训练速度提高几个数量级,而 NDCG、MAP 等检索性能丝毫不逊色. 文中还讨论误排序惩罚系数对 cs-sRSVM 算法的影响.

cs-sRSVM 的优化目标是错误排序的总体代价,与 Web 检索评估测度并无直接对应关系. 寻找能够近似 NDCG 等测度的二次可微光滑函数将是以后工作的重点.

参 考 文 献

[1] Joachims T. Optimizing Search Engines Using Click through Data// Proc of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Edmonton, Canada, 2002: 133-142

[2] Svore K, vander Wende L, Burges C J C. Using Signals of Human Interest to Enhance Single-Document Summarization // Proc of the 23rd AAAI Conference on Artificial Intelligence. Chicago, USA, 2008: 1577-1580

[3] Gao Yongmei, Huang Yalou, Ni Weijian, *et al*. A Ranking SVM Based Algorithm for Automatic Extraction of Acronym. Pattern Recognition and Artificial Intelligence, 2008, 21(2): 186-192 (in Chinese)
(高永梅,黄亚楼,倪维健,等. 基于排序支持向量机的缩略词自动提取方法. 模式识别与人工智能, 2008, 21(2): 186-192)

[4] Chu W, Keerthi S S. Support Vector Ordinal Regression. Neural Computation, 2007, 19(3): 792-815

[5] Li Ping, Burges C J C, Wu Qiang. Mcrank: Learning to Rank Using Multiple Classification and Gradient Boosting // Proc of the 21st Annual Conference on Neural Information Processing Systems. Vancouver, Canada, 2007: 845-852

[6] Herbrich R, Graepel T, Obermayer K. Large Margin Rank Boundaries for Ordinal Regression // Proc of the 9th International Conference on Artificial Neural Networks. Edinburgh, UK, 1999: 97-102

[7] Xu Jun, Cao Yunbo, Li Hang, *et al*. Cost-Sensitive Learning of SVM for Ranking // Proc of the 17th European Conference on Machine Learning. Berlin, Germany, 2006: 833-840

[8] Cao Yunbo, Xu Jun, Liu Tieyan, *et al*. Adapting Ranking SVM to Document Retrieval // Proc of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, USA, 2006: 186-193

[9] Burges C, Shaked T, Renshaw E, *et al*. Learning to Rank Using Gradient Descent // Proc of the 22nd International Conference on Machine Learning. Bonn, Germany, 2005: 89-96

[10] Guiver J, Snelson E. Learning to Rank with SoftRank and Gaussian Processes // Proc of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Singapore, Singapore, 2008: 259-266

[11] Carvalho V R, Elsas J, Cohen W W, *et al*. A Meta-Learning Approach for Robust Rank Learning // Proc of the SIGIR Workshop on Learning to Rank for Information Retrieval. Singapore, Singapore, 2008: 15-23

[12] Cao Zhe, Qin Tao, Liu Tieyan, *et al*. Learning to Rank: From Pairwise Approach to Listwise Approach // Proc of the 24th Annual International Conference on Machine Learning. Corvallis, USA, 2007: 129-136

[13] Elsas J L, Carvalho V R, Carbonell J G. Fast Learning of Document Ranking Functions with the Committee Perceptron // Proc of the 1st ACM International Conference on Web Search and Data Mining. Palo Alto, USA, 2008: 55-64

[14] Zheng Zhaohui, Zha Honguan, Sun G. Query-Level Learning to Rank Using Isotonic Regression // Proc of the 46th Annual Allerton Conference on Communication, Control and Computing. Urbana-Champaign, USA, 2008: 1108-1115

[15] Zadrozny B. Learning and Evaluating Classifiers under Sample Selection Bias // Proc of the 21st International Conference on Machine Learning. Banff, Canada, 2004: 114-121

[16] Sheng V S, Ling C X. Thresholding for Making Classifiers Cost-Sensitive // Proc of the 21st National Conference on Artificial Intelligence. Boston, USA, 2006: 476-481

[17] Liu Tieyan, Xu Jun, Qin Tao, *et al*. LETOR: Benchmarking Learning to Rank for Information Retrieval // Proc of SIGIR Workshop on Learning to Rank for Information Retrieval. Amsterdam, Netherlands, 2007: 3-8

[18] Lee Y J, Mangasarian O L. SSVM: A Smooth Support Vector Machine. Computational Optimization and Applications, 2001, 20(1): 5-22

[19] Yuan Yubo, Yan Jie, Xu Chengxian. Polynomial Smooth Support Vector Machine (PSSVM). Chinese Journal of Computers, 2005, 28(1): 9-17(in Chinese)
(袁玉波,严 杰,徐成贤. 多项式光滑的支撑向量机. 计算机学报, 2005, 28(1): 9-17)

[20] Yuan Yaxiang. Step-Sizes for the Gradient Method // Proc of the 3rd International Congress of Chinese Mathematicians. Hongkong, China, 2004: 785-796

[21] Järvelin K, Kekäläinen J. Cumulated Gain-Based Evaluation of IR Techniques. ACM Trans on Information Systems, 2002, 20(4): 422-446

[22] Huang C M, Lee Y J, Lin D, *et al*. Model Selection for Support Vector Machines via Uniform Design. Computational Statistics and Data Analysis, 2007, 52(1): 335-346