

基于改进 Pairwise 损失函数的排序学习方法*

吴佳金, 杨志豪, 林原, 林鸿飞

大连理工大学信息检索研究室, 大连, 116024

E-mail: wujiain@mail.dlut.edu.cn

摘 要: 排序学习是当前机器学习与信息检索交叉领域的研究热点。它利用机器学习方法自动地构造排序模型, 用于排序新的数据。Pairwise 方法是一类重要的排序学习方法, 它以偏序文档对作为训练样例, 通过判断不同文档与查询的相关性大小关系来为文档排序。由于 pairwise 方法不考虑单个文档的相关性大小, 若对文档对内两个文档的相关性均预测错误, 则会导致连锁反应并影响最终排序性能; 为解决该问题, 本文分别基于单层神经网络和双层神经网络的 RankNet 算法, 加入 pointwise 损失函数进行优化, 并分别使用梯度下降算法和反向传播算法训练网络权重值, 得到排序模型。在 OHSUMED 数据集上的实验结果表明, 加入 pointwise 损失函数有助于改善 pairwise 方法的排序性能。

关键词: 信息检索; 排序学习; 损失函数; Pairwise; RankNet

Learning to Rank based on Improved Pairwise Loss Function

Jiajin Wu, Zhihao Yang, Yuan Lin, Hongfei Lin

Information Retrieval Laboratory, Dalian University of Technology, Dalian 116024

E-mail: wujiain@mail.dlut.edu.cn

Abstract: Learning to rank is a hot issue at the intersection of machine learning and information retrieval. It uses methods of machine learning to automatically construct a model for assigning relevance degree to new objects and ranking them according to the given degree. Pairwise approach is an important paradigm in learning to rank, which views document pairs as training instances and ranks documents according to the relevance preference of document pairs. It will lead to poor performance for pairwise approach mistaking the relevance degree of both documents in document pair. In order to solve the aforementioned problem, this paper shows the method of improving pairwise approach based on the one layer and two layers ANN RankNet by incorporating pointwise loss function. Gradient descent and back propagation are used to train the network weights, which are going to be used as the ranking model. Experimental results on OHSUMED dataset show that incorporating pointwise loss in pairwise approach can lead to better ranking performance.

Keywords: Information retrieval; Learning to rank; Loss function; Pairwise; RankNet

1 引言

排序学习是当前机器学习和信息检索交叉领域的热点问题。排序是信息检索领域的关键问题之一, 许多问题实质都是排序, 如: 文本检索、定义发现、问答系统、文档摘要和情感分析等^[1]。由于用作排序的特征不断增多, 传统地利用人工调整参数的方式构造排序

*基金项目: 国家自然科学基金资助项目(编号: 60673039, 60973068)、国家社科基金(编号: 08BTQ025)、国家 863 高科技计划资助项目(编号: 2006AA01Z151)、教育部留学回国人员科研启动基金和高等学校博士学科点专项科研基金资助课题(编号: 20090041110002)。第一作者简介: 吴佳金, 男, 1986 年生, 在读硕士, 研究方向为信息检索、排序学习和机器学习。

模型变得越来越不可行。排序学习借鉴机器学习的思想和方法,通过学习训练集获得排序模型,并用其对文档与查询的相关性(以下简称相关性)进行预测,最后根据相关性大小对文档进行排序。学习的过程通常是根据经验风险最小化原则,通过最小化损失函数值得到的。目前根据训练样例的不同,一般将排序学习方法分为三类: pointwise、pairwise 和 listwise^[2]。

较早提出的排序学习方法属于 pointwise 方法,将单个文档作为训练样例,将对文档集合的排序看作对每个文档相关性大小的预测^[2],即将排序问题转化为分类或回归问题以直接利用已有的机器学习方法来解决。经典的 pointwise 方法有 Subset Regression^[3]和 McRank^[4]等; Pairwise 方法的训练样例是偏序文档对(由具有不同相关性大小的文档两两构成),它将对文档的排序转化为对不同文档相关性大小关系的预测,它的进步意义在于在训练时考虑了不同文档的相关性大小顺序,更接近于排序问题。经典的 pairwise 方法有: Ranking SVM^[5]、RankBoost^[6]和 RankNet^[7]; Listwise 方法则以文档排序列表作为训练样例, ListNet^[8]属于该类方法^[2]。目前, listwise 方法可以分为两类^[2]: 一类将损失函数构造为信息检索结果评价函数的界限,如 SVMMap^[9]; 另一类将损失函数构造为用于衡量预测的文档序列与实际最好的文档序列之间的差异,如 ListNet 和 ListMLE^[10]。

现存的排序学习方法大都基于单一训练样例,但随着排序学习的不断发展,我们发现单一地考虑训练样例会忽视其它样例对排序的作用: 对于给定的一个查询和它的相关文档集合, pointwise 方法关注每个文档与查询的相关性大小,但是忽略了不同文档与查询相关性的关系,即如果其它文档与查询的相关性很小,那么一个相关性不太大的文档对用户来讲可能也是可行的; Pairwise 方法关注不同文档与查询的相关性大小关系,认为只要尽量保证不同文档的相关性大小关系被正确预测,即可得到令用户满意的文档序列,但这种不考虑单个文档相关性大小的作法,会由于对组成文档对的文档错误预测而引发连锁反应——影响与之有偏序关系的其它文档——从而影响最终排序结果; Listwise 方法将文档序列作为整体考虑,认为用户关心的主要是位于文档序列前列的文档,而非全部文档^[2,10]。

对于 pairwise 方法,假设存在两个偏序文档对 $\{A, B\}$ 和 $\{C, D\}$, 它们的相关性标注分别为 $\{2, 1\}$ 和 $\{1, 0\}$ 。如果排序模型将 $\{A, B\}$ 的相关性误判为 $\{1, 0\}$, 虽然仍满足文档 A 与 B 的相关性大小关系,但文档 B 与 D 的相关性大小关系则会被判断错误,进而会引起连锁反应,导致其它与之具有偏序关系的文档皆受影响而出现偏差,并影响最终的排序结果。

针对该问题,本文在 pairwise 方法中引入 pointwise 级别的损失函数进行优化,不同于文献[11]中利用 pairwise 损失函数作约束条件,优化 listwise 损失函数提出的 IntervalRank 算法,本文在 pairwise 方法中增加了对单文档相关性的考虑。改进后的损失函数同时考虑了不同文档间的相关性大小关系和单个文档的相关性大小,能够纠正上述问题造成的偏差,同时可将相关性较大的文档排在前列。

2 利用 pointwise 损失函数改进 pairwise 方法

2.1 RankNet 算法

RankNet 是一种重要的 pairwise 方法,它是目前为数不多地被用于商业搜索引擎中的排序学习方法之一^[2],具有较好的代表性和性能。本文以它为例进行改进。

RankNet 采用人工神经网络(ANN)^[12]学习预测模型用以预测文档与查询的相关性大小,并据此对文档进行排序。人工神经网络由一系列简单单元相互连接构成,每个单元有一定数量的输入,并产生单一的实数值输出。它被用于从样例中学习值为实数、离散值或向量

的函数，通过一定的学习算法调节网络参数以最佳拟合输入一输出对。这种网络结构能够很好地拟合文档特征向量与其相关性分值的函数，RankNet 算法分别应用了单层感知器单元和双层 sigmoid 单元构建两种网络，并分别选择梯度下降算法，以及反向传播算法迭代地对这两种网络的权重值进行优化，并选取最优的网络权重值，作为排序模型。

RankNet 算法采用交叉熵函数作为损失函数。对于给定的一个查询 q ，记 $D\{x_i | i=1, \dots, m\}$ 为其对应文档集合， $R^d\{(x_i, x_j) | y_i > y_j\}$ 为 D 中所有偏序文档对 $\{x_i, x_j\}$ 的集合，排序模型为 $f: D \rightarrow R$ ，由 f 预测的文档相关性 $x_i > x_j$ 的概率为 $P_{ij} = (i, j=1, \dots, m)$ ， \bar{P}_{ij} 为目标概率， $o_i = f(x_i)$ ， $o_{ij} = f(x_i) - f(x_j)$ ，则其损失函数形式如下：

$$L_{ij} = L(o_{ij}) = -\bar{P}_{ij} \log P_{ij} - (1 - \bar{P}_{ij}) \log(1 - P_{ij}) \quad (1)$$

其中， P_{ij} 由逻辑函数(logistic function)定义：

$$P_{ij} = \frac{e^{o_{ij}}}{1 + e^{o_{ij}}} \quad (2)$$

排序模型 f 由神经网络的结构(网络层数、网络单元所采用的传递函数等)决定。其中，单层网络(是指除输出层外，包含一层网络单元，双层网络类推)的 f 形式如下：

$$f(x) = g^2(\sum_k w_k^{21} \cdot x_k + b^2) \quad (3)$$

双层网络的 f 形式如下：

$$f(x) = g^3(\sum_j w_j^{32} \cdot g^2(\sum_k w_{jk}^{21} \cdot x_k + b_j^2) + b_i^3) \quad (4)$$

其中 g^i 为网络第 i 层单元的传递函数(或挤压函数)，其作用为将输入映射到固定的范围。 w 与 b 的上标代表其结点所属层数，下标代表该层内的结点标号。为简化起见，本文参照文献[8]的处理方式，采用 $g(y) = y$ 的形式，并省略偏置 b 。

2.2 利用 pointwise 损失函数改进 pairwise 方法

Pairwise 方法的训练样例是偏序文档对，它将对文档的排序转化为对不同文档与查询相关性大小关系的预测；因此，如果因某个文档相关性被预测错误，或文档对的两个文档相关性均被预测错误，则会影响与之关联的其它文档，进而引起连锁反应并影响最终排序结果。而 pointwise 方法的训练样例是单个文档，它解决的问题恰恰是对单个文档的相关性预测。基于此，本文在 pairwise 方法的基础上，增加 pointwise 损失函数，通过融入对单个文档相关性大小的考虑，减小因错误预测单个文档相关性大小而导致连锁反应所造成的损失，来优化 pairwise 方法排序模型，提升其排序性能。

考查现存 pointwise 方法，大多将排序问题转化为分类或回归问题，所用到的损失函数分别为分类或回归问题的损失函数。本文采用回归损失函数，它也是人工神经网络中常用的用于衡量训练样例的预测值与真实值之间误差的函数。记 y_i 为训练集中文档 x_i 的相关性大小标注值， $f(x_i)$ 为排序模型对文档 x_i 的相关性预测值，则其 pointwise 损失函数形式如下：

$$L_i = \frac{1}{2} L(y_i - f(x_i))^2 \quad (5)$$

考虑到 pairwise 方法不会将所有文档的相关性预测错误，且单个文档在不同偏序文档对中存在重复，因而将 pairwise 损失函数与 pointwise 损失函数线性插值，得到改进的损失函数形式如下：

$$L = \sum_{\{x_i, x_j\} \in R^d} (c_1 \cdot L_{ij} + c_2 \cdot \sum_{k \in \{i, j\}} L_k) \quad (6)$$

其中， c_1 与 c_2 是不同损失函数的权重，实验中通过验证集来确定其大小。

对于单层网络，采用梯度下降算法优化网络权重值；对于双层网络，采用前馈网络一般采用的误差反向传播算法优化网络权重值，并从中选取性能最优的模型作为最终的排序模型。反向传播算法是梯度下降算法的一个改进，用于训练多层人工神经网络的权重值，基本思想是将输入看作正向传播，将误差看作由输出层经隐藏层再到输入层逐级反向传播。通过从输出层经隐藏层再到输入层，迭代地更新网络权重值，以减小误差来获得具有最优拟合能力的网络。本文训练双层网络权重值的学习算法如下：

算法 1 反向传播算法(back-propagation)

输入： 训练样例集合 S ，初始学习率 η ，网络输入层单元数 n_{in} ，隐藏层单元数 n_{hidden} ，网络初始权重值 $W = \{\omega\}$ ，迭代次数 T

输出： RankNet 排序模型

1. 创建具有 n_{in} 个输入单元， n_{hidden} 个隐藏单元，1 个输出单元的网络，按照 W 设置网络初始权重值

2. 对于 $t(t=1, \dots, T)$

2.1 对于训练样例集合 S 中的每个偏序文档对 $\{x_i, x_j\}$

2.1.1 将 $\{x_i, x_j\}$ 输入网络，计算网络输出 f_i 和 f_j 及损失 L_{ij} 、 L_i 与 L_j

2.1.2 对于网络输出单元与隐藏层单元 $h(h=1, \dots, n_{hidden})$ 之间的网络权重值，计算权重值 $\Delta\omega_h^{32}$

$$\Delta\omega_h^{32} = c_1 \cdot \frac{\partial L_{ij}}{\partial o_{ij}} \cdot \frac{\partial o_{ij}}{\partial \omega_h^{32}} + c_2 \cdot \frac{\partial L_i}{\partial o_i} \cdot \frac{\partial o_i}{\partial \omega_h^{32}} + c_2 \cdot \frac{\partial L_j}{\partial o_j} \cdot \frac{\partial o_j}{\partial \omega_h^{32}}$$

2.1.3 对于网络隐藏层单元 h 与其在输入层所有对应单元 $s(s=1, \dots, n_{in})$ 之间的网络权重值，计算 $\Delta\omega_{hs}^{21}$

$$\Delta\omega_{hs}^{21} = c_1 \cdot \frac{\partial L_{ij}}{\partial o_{ij}} \cdot \frac{\partial o_{ij}}{\partial \omega_{hs}^{21}} + c_2 \cdot \frac{\partial L_i}{\partial o_i} \cdot \frac{\partial o_i}{\partial \omega_{hs}^{21}} + c_2 \cdot \frac{\partial L_j}{\partial o_j} \cdot \frac{\partial o_j}{\partial \omega_{hs}^{21}}$$

2.1.4 更新网络所有权重值 ω ： $\omega = \omega - \eta \Delta\omega$

2.2 若此次迭代所有样例误差的均值比上次的大，则调整 η 的大小

对于单个查询 q 及其相关文档集合 S ，上述算法的时间复杂度为 $O(T \cdot n^2)$ ，其中 T 为反向传播算法的迭代次数， n 为相关文档数量。与 RankNet 所采用的反向传播算法相比，时间复杂度相同；不过，由于每次迭代所需处理训练样例的增多，对相同规模的问题，实际

训练模型的时间会有所增加，但由此带来的精度提升是有益的。

3 实验

3.1 实验数据集

本文采用微软亚洲研究院公布的 Letor3.0^[1]的 OHSUMED 数据集，它来源于医学检索任务，包含 3 个相关性标注：2、1 和 0，分别表示文档与给定查询是相关、可能相关和不相关的。集合中文档被表示成特征向量的形式，共包含 45 维文档特征项，其中有如 BM25 和语言模型等基于文档内容的特征，也有如 HITS 和 PageRank 等基于链接分析的特征。数据集被均匀地分为五组，用于五折交叉验证；每个子集由训练集、验证集和测试集构成，三者的大小比例为 3：1：1。规模较小的验证集被用来确定网络的初始权重值、最佳迭代次数等。以第一个子集为例，其训练集包含 63 个查询，每个查询相关文档数目不等，总共 9129 个查询—文档对。

3.2 实验设计

本文采用 Letor3.0 数据集作为实验语料，实验结果与 Regression(pointwise 方法)、Ranking SVM、RankBoost、Frank(三种 pairwise 方法)、ListNet(应用人工神经网络的 listwise 方法)及 IntervalRank 进行比较。

在实验中，首先对原始数据集处理。对同一个查询的相关文档，按照相关性标注大小关系两两作对，得到与查询相关的文档偏序对。以第一个子集为例，共构建了 367, 662 个文档偏序对。文档偏序对的集合与原始数据集共同构成了本文实验的训练数据集。

本文实现了单层神经网络和双层神经网络的 RankNet，分别在其上加入 pointwise 损失函数。对于单层 RankNet，设定网络的初始权重值为 0；对于双层 RankNet，设定其隐藏层权重值为较小的随机值，如 $[-0.5 \sim 0.5]$ ，输入层权重值为 0 或较小的随机值 $[-0.01 \sim 0.01]$ 。初始学习率设为 0.0001，每次迭代后如果所有样例的平均误差大于上次迭代的平均误差，则学习率降为原来的一半。

3.3 实验结果与分析

对实验结果，本文分别采用了信息检索领域广泛采用的 P@k、Map 和 NDCG@k 三种方式评价。在下面的图表中，用 RankNet(L1)与 RankNet(L2)分别表示单层与双层神经网络 RankNet 方法，用 RankNetPlus(L1)与 RankNetPlus(L2)分别表示二者的改进方法。在表 1 与表 2 中分别列出了 RankNetPlus(L2)同 IntervalRank 的 NDCG@k 与精度值的对比结果，其中 IntervalRank 的实验数据来自论文[11]。在表 3 中列出了其它方法的 Map 值对比结果。

表 1 与 IntervalRank 的 NDCG@k 值比较

Tab.1 Comparison with IntervalRank on NDCG@k

Algorithms	N@1	N@2	N@3	N@4	N@5
RankNetPlus(L2)	0.5418	0.5036	0.4953	0.4840	0.4725
IntervalRank	0.5628	0.5448	0.4900	0.4703	0.4609

表 2 与 IntervalRank 的精度对比

Tab.2 Comparison with IntervalRank on precision

Algorithms	P@1	P@2	P@3	P@4	P@5	Map
RankNetPlus(L2)	0.6429	0.6292	0.6181	0.6022	0.5747	0.4456
IntervalRank	0.6892	0.6522	0.5768	0.5556	0.5488	0.4666

表 3 Map 值对比

Tab.3 Map scores comparison

Algorithms	Fold1	Fold2	Fold3	Fold4	Fold5	Average
Regression	0.2979	0.4302	0.4398	0.4978	0.4442	0.422
Ranking SVM	0.3038	0.4468	0.4648	0.4990	0.4528	0.4334
RankBoost	0.3318	0.4448	0.4562	0.5081	0.4645	0.4411
FRank	0.3327	0.4376	0.4556	0.5128	0.4806	0.4439
ListNet	0.3464	0.4499	0.4606	0.5106	0.4611	0.4457
RankNet(L1)	0.3368	0.4450	0.4243	0.4678	0.4522	0.4252
RankNetPlus(L1)	0.3371	0.4465	0.4258	0.4712	0.4523	0.4266
RankNet(L2)	0.3409	0.4482	0.4478	0.5014	0.4605	0.4398
RankNetPlus(L2)	0.3343	0.4515	0.4657	0.5129	0.4634	0.4456

图 1 与图 2 分别是不同算法在 OHSUMED 数据集五个子集的平均 P@k 与 NDCG@k 值的比较结果, 为使对比结果清晰明了, 除本文算法 RankNetPlus(L2)外, 图中仅选取了如下几种算法: Regression、Ranking SVM、RankNet(L2)、FRank 和 ListNet 等, 上述算法涵盖了三种排序学习方法, 且均为目前性能较优的算法, 具有代表性。其中 FRank^[13]算法是 RankNet 的一个改进算法, 它与 RankNet 的区别在于所使用的损失函数不同, FRank 采用了物理学中的 fidelity 函数作为损失函数。图中纵坐标为评价函数分值, 横坐标为处于文档序列不同位置的评价函数指标, 用 N@k 代表 NDCG@k。

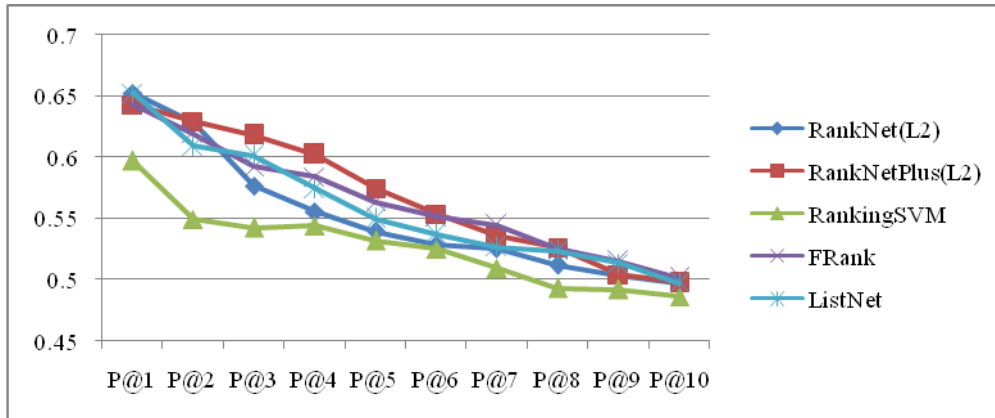


图 1 平均 P@k 值曲线

Fig.1 Average P@k scores curve

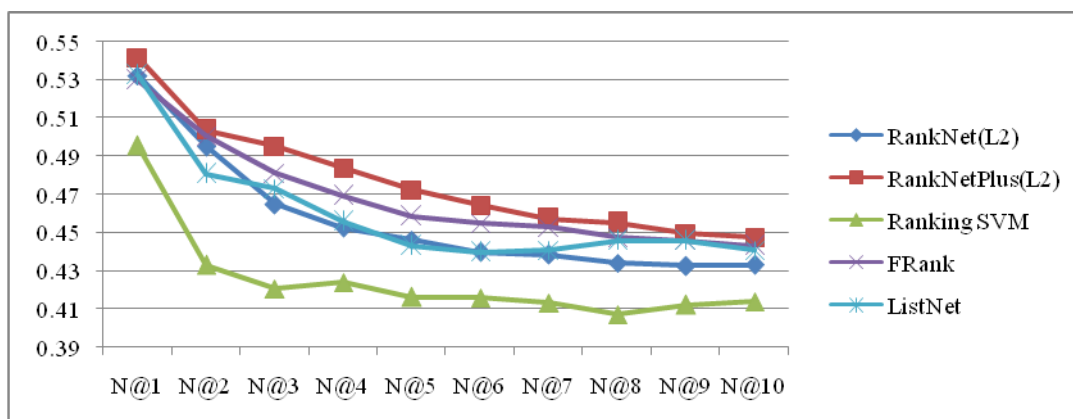


图 2 平均 NDCG@k 值曲线

Fig.2 Average NDCG@k scores curve

从实验结果中不难看出，双层神经网络 RankNet 的性能优于单层神经网络 RankNet，这是因为双层神经网络的表征能力强于单层神经网络。另外，无论单层或双层神经网络 RankNet，加入 pointwise 损失函数后排序性能都得到了提升，并且在双层神经网络 RankNet 基础上，加入 pointwise 损失函数后性能得到了明显提升，整体上优于其它 pairwise 方法。

RankNetPlus(L2)与 IntervalRank 相比，NDCG@k 与 P@k 值，当 $k=3, \dots, 5$ 时占有优势，且与前两项相比，降幅并不明显，由此可知 RankNetPlus(L2)的排序性能更加稳定；但前两项的值与 Map 值处于劣势，忽略实验细节问题，我们认为这一方面是由于 IntervalRank 考虑了 listwise 训练样例的原因，另一方面与其采用 SVM 训练模型有关，SVM 可以保证模型得到的解是全局最优解。RankNetPlus(L2)方法与其它方法相比，在 OHSUMED 三个子集上的平均 Map 值达到了最高，平均 Map 值仅比 ListNet 方法低 0.0001；P@k 与 NDCG@k 值在绝大多数指标上也领先于其它方法。

加入 pointwise 损失函数能够改善 pairwise 方法，原因在于 pairwise 方法存在以下缺陷，而加入 pointwise 损失函数解决了这个问题：pairwise 方法的训练样例是偏序文档对，因而只能尽可能地保证文档间相关性大小被预测正确，而如果在此基础上，排序模型对偏序文档对内单个文档的相关性预测错误，则导致与其它文档的相关性大小关系出现偏差，进而引起连锁反应，并影响最终排序结果。我们的方法引入 pointwise 损失函数，可以对排序模型错误预测单个文档相关性的情况进行惩罚，而加之 pairwise 方法本身对不同文档相关性大小关系的考虑，通过 pointwise 损失函数提高了对单个文档的相关性预测精度后，也会改善其它与该文档有偏序关系的文档，从而使整体效果改善。同时 pointwise 损失函数也使得相关性较高的文档易被排在前列，而这正是检索用户所希望的。实验中，有些文档集合的结果 P@1 值高达 0.8 也证明了方法的有效性。

4 结论与展望

本文通过对 pairwise 方法引入 pointwise 损失函数，优化了 pairwise 损失函数，改进后的损失函数同时考虑了不同文档间的相关性大小关系，以及单个文档的相关性大小，解决了因错误预测单个文档相关性而导致连锁反应，进而影响最终排序结果的问题，同时能够

将相关性较大的文档排在前列。实验结果证明了, 引入 `pointwise` 损失函数有助于改善 `pairwise` 方法排序性能。同时, 本文也提供了一种综合利用不同训练样例构造排序学习方法的思路, 这将会是排序学习今后发展的趋势之一。

未来的工作可以从以下几个方面继续进行:(1) 尝试对其它的 `pairwise` 方法, 如 `Ranking SVM` 进行改进。(2) 寻找更为恰当的损失函数来综合利用不同训练样例, 以最大程度地发挥不同样例的优势。

参 考 文 献

- [1] T. Qin, T.Y. Liu, J. Xu, et al. LETOR: Benchmark Collection for Research on Learning to Rank for Information Retrieval[C]. SIGIR 2007 Workshop on Learning to Rank for Information Retrieval (LR4IR 2007), Amsterdam, The Netherlands, 2007, 3-10.
- [2] T.Y. Liu. Learning to Rank for Information Retrieval[J]. Foundations and Trends in Information Retrieval, 2009, 3(3): 225-331.
- [3] D. Cossock, T. Zhang. Subset ranking using regression[C]. The 19th Annual Conference on Learning Theory(COLT'2006), Carnegie Mellon University, Pittsburgh, Pennsylvania, 2006, 605-619.
- [4] J. Lafferty, C. Zhai. Document language models, query models and risk minimization for information retrieval[C]. International Conference on Research and Development in Information Retrieval(SIGIR'2001), New Orleans, Louisiana, USA, 2001, 111-119.
- [5] T. Joachims. Optimizing Search Engines using Clickthrough Data[C]. The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'2002), Edmonton, Alberta, Canada, 2002, 133-142.
- [6] Y. Freund, R. Iyer, R. E. Schapire, et al. An Efficient Boosting Algorithm for Combining Preferences[J]. Journal of Machine Learning Research, 2003, 4: 933-969.
- [7] C. Burges, T. Shaked, E. Renshaw, et al. Learning to Rank using Gradient Descent[C]. Proceedings of the 22th International Conference on Machine Learning (ICML'2005), Bonn, Germany, 2005, 89-96.
- [8] Z. Cao, T. Qin, T.Y. Liu, et al. Learning to Rank: From Pairwise Approach to Listwise Approach[C]. Proceedings of the 24th International Conference on Machine Learning (ICML'2007), Corvallis, Oregon, 2007, 129-136.
- [9] Y. Yue, T. Finley, F. Radlinski, et al. A Support Vector Method for Optimizing Average Precision[C]. International Conference on Research and Development in Information Retrieval (SIGIR'2007), Amsterdam, The Netherlands, 2007, 271-278.
- [10] F. Xia, T.Y. Liu, J. Wang, et al. Listwise approach to learning to rank—Theorem and algorithm[C]. International Conference on Machine Learning(ICML'2008), Helsinki, Finland, 2008, 1192-1199.
- [11] T. Moon, A. Smola, Y. Chang, et al. IntervalRank—Isotonic Regression with Listwise and Pairwise Constraints[C]. Third ACM International Conference on Web Search and Data Mining (WSDM'2010), New York City, New York, USA, 2010, 151-159.
- [12] 曾华军, 张银奎. 机器学习[M]. 北京: 机械工业出版社, 2003.
- [13] F. M., T.Y. Liu, T. Qin, et al. Frank: A ranking method with fidelity loss[C]. International Conference on Research and Development in Information Retrieval (SIGIR'2007), Amsterdam, The Netherlands, 2007, 383-390.
- [14] 张学工. 统计学习理论的本质[M]. 北京: 清华大学出版社, 2000.

基于改进Pairwise损失函数的排序学习方法

作者:

[吴佳金](#), [杨志豪](#), [林原](#), [林鸿飞](#)

作者单位:

[大连理工大学信息检索研究室, 大连, 116024](#)

本文链接: http://d.g.wanfangdata.com.cn/Conference_7505976.aspx