

041 | 机器学习排序算法经典模型：LambdaMART

2018-01-05 洪亮劼

AI技术内参

[进入课程 >](#)



讲述：初明明

时长 07:24 大小 3.40M



在这周的时间里，我们讨论机器学习排序算法中几个经典的模型。周一我们分享了排序支持向量机（RankSVM），这个算法的好处是模型是线性的，容易理解。周三我们聊了梯度增强决策树（Gradient Boosted Decision Tree），长期以来，这种算法被用在很多商业搜索引擎当中来作为排序算法。

今天，我们来分享这一部分的最后一个经典模型：**LambdaMART**。这是微软在 Bing 中使用了较长时间的模型，也在机器学习排序这个领域享有盛誉。

LambdaMART 的历史

LambdaMART 的提出可以说是一个“三步曲”。

这里有一个核心人物，叫克里斯多夫·博格斯（Christopher J.C. Burges）。博格斯早年从牛津大学物理学毕业之后，又于布兰戴斯大学（Brandeis University）获得物理学博士学位，他曾在麻省理工大学做过短暂的博士后研究，之后来到贝尔实验室，一待 14 年。2000 年，他来到微软研究院，并一直在微软研究院从事机器学习和人工智能的研究工作，直到 2016 年退休。可以说，是博格斯领导的团队发明了微软搜索引擎 Bing 的算法。

LambdaMART 的第一步来自于一个叫 RankNet 的思想[1]。这个模型发表于 ICML 2005，并且在 10 年之后获得 ICML 的时间检验奖。这也算是在深度学习火热之前，利用神经网络进行大规模商业应用的经典案例。

RankNet 之后，博格斯的团队很快意识到了 RankNet 并不能直接优化搜索的评价指标。因此他们根据一个惊人的发现，**提出了 LambdaRank 这一重要方法**[2]。LambdaRank 的进步在于算法开始和搜索的评价指标，也就是 NDCG 挂钩，也就能够大幅度提高算法的精度。

LambdaRank 之后，博格斯的团队也认识到了当时从雅虎开始流行的使用“梯度增强”（Gradient Boosting），特别是“梯度增强决策树”（GBDT）的思路来进行排序算法的训练，于是他们就把 LambdaRank 和 GBDT 的思想结合起来，**开发出了更加具有模型表现力的 LambdaMART**[3]。LambdaMART 在之后的雅虎排序学习比赛中获得了最佳成绩。

RankNet 的思想核心

要理解 LambdaMART，我们首先要从 RankNet 说起。其实，有了排序支持向量机 RankSVM 的理论基础，要理解 RankNet 就非常容易。RankNet 是一个和排序支持向量机非常类似的配对法排序模型。也就是说，RankNet 尝试正确学习每组两两文档的顺序。那么，怎么来定义这个所谓的两两文档的顺序呢？

其实，我们需要做的就是**定义一个损失函数（Loss Function）来描述如何引导模型学习正确的两两关系**。我们可以假设能够有文档两两关系的标签，也就是某一个文档比另外一个文档更加相关的信息。这个信息可以是二元的，比如 +1 代表更加相关，-1 代表更加不相关，注意这里的“更加”表达了次序关系。

那么，在理想状态下，不管我们使用什么模型，都希望模型的输出和这个标签信息是匹配的，也就是说模型对于更加相关的文档应该输出更加高的预测值，反之亦然。很自然，**我们能够使用一个二元分类器的办法来处理这样的关系**。RankNet 在这里使用了“对数几率损

失函数” (Logistic Loss) , 其实就是希望能够利用 “对数几率回归” (Logistic Regression) 这一思想来处理这个二元关系。唯一的区别是, 这里的正例是两个文档的相对关系。

有了损失函数之后, 我们使用什么模型来最小化这个损失函数呢? 在 RankNet 中, 作者们使用了神经网络模型, 这也就是 Net 部分的由来。那么, 整个模型在这里就变得异常清晰, 那就是**使用神经网络模型来对文档与文档之间的相对相关度进行建模, 而损失函数选用了 “对数几率损失函数”**。

LambdaRank 和 LambdaMART

尽管 RankNet 取得了一些成功, 但是, 文档的两两相对关系并不和搜索评价指标直接相关。我们之前讲过, 搜索评价指标, 例如 NDCG 或者 MAP 等, 都是直接建立在对于某一个查询关键字的相关文档的整个序列上, 或者至少是序列的头部 (Top-K) 的整个关系上的。因此, RankNet 并不能保证在 NDCG 这些指标上能够达到很好的效果, 因为毕竟没有直接或者间接优化这样的指标。

要想认识这一点其实很容易, 比如你可以设想对于某一个查询关键字, 有 10 个文档, 其中有两个相关的文档, 一个相关度是 5, 另外一个相关度是 3。那么, 很明显, 在一个理想的排序下, 这两个文档应该排在所有 10 个文档的头部。

现在我们假定相关度 5 的排在第 4 的位置, 而相关度 3 的排在第 7 的位置。RankNet 会更愿意去调整相关度 3 的, 并且试图把其从第 7 往前挪, 因为这样就可以把其他不相关的挤下去, 然而更优化的办法应该是尝试先把相关度 5 的往更前面排。也就是说, 从 NDCG 的角度来说, 相关度高的文档没有排在前面受到的损失要大于相关度比较低的文档排在了下面。

NDCG 和其他一系列搜索评价指标都是更加注重头部的相关度。关于这一点, RankNet 以及我们之前介绍的 GBDT 或者排序支持向量机都忽视了。

既然我们找到了问题, 那么如何进行补救呢?

之前说到博格斯的团队有一个惊人的发现, 其实就在这里。他们发现, RankNet 的优化过程中使用到的梯度下降 (Gradient Descent) 算法需要求解损失函数针对模型的参数的梯度, 可以写成**两个部分**的乘积。在这里, 模型的参数其实就是神经网络中的各类系数。第一部分, 是损失函数针对模型的输出值的, 第二部分是模型输出值针对模型的参数的。第二个

部分跟具体的模型有关系，但是第一个部分没有。第一个部分跟怎么来定一个损失函数有关系。

在原始的 RankNet 定义中，这当然就是“对数几率函数”定义下的损失函数的梯度。这个数值就是提醒 RankNet 还需要针对这个损失做多少修正。其实，这个损失梯度不一定非得对应一个损失函数。这是博格斯的团队的一个重大发现，只要这个损失的梯度能够表示指引函数的方向就行了。

那既然是这样，能不能让这个损失的梯度和 NDCG 扯上边呢？答案是可以的。也就是说，我们只要定义两个文档之间的差距是这两个文档互换之后 NDCG 的变化量，同时这个变化量等于之前所说的损失的梯度，那么我们就可以指导 RankNet 去优化 NDCG。在这里，博格斯和其他作者把这个损失的梯度定义为 Lambda，因为整个过程是在优化一个排序，所以新的方法叫作 LambdaRank。

有了 LambdaRank 之后，LambdaMART 就变得水到渠成。Lambda 是被定义为两个文档 NDCG 的变化量（在实际运作中，是用这个变化量乘以之前的对数几率所带来的梯度）。那么，只要这个 Lambda 可以计算，模型就可以嫁接别的算法。于是，博格斯的团队使用了在当时比神经网络更加流行的“梯度增强决策树”（GBDT）来作为学习器。不过，梯度增强决策树在计算的时候需要计算一个梯度，在这里就被直接接入 Lambda 的概念，使得 GBDT 并不是直接优化二分分类问题，而是一个改装了的二分分类问题，也就是在优化的时候优先考虑能够进一步改进 NDCG 的方向。

小结

今天我为你讲了 LambdaMART 算法的基本原理。作为配对法和列表排序学习的一个混合经典算法，LambdaMART 在实际运用中有着强劲的表现。一起来回顾下要点：第一，我们简要介绍了 LambdaMART 提出的历史。第二，我们详细介绍了 LambdaMART 的核心思路。

最后，给你留一个思考题，采用 Lambda 这样更改优化过程中的梯度计算，虽然很形象，但是有没有什么坏处？

欢迎你给我留言，和我一起讨论。

参考文献

1. Burges, C.; Shaked, T.; Renshaw, E.; Lazier, A.; Deeds, M.; Hamilton, N. & Hullender, G. Learning to Rank Using Gradient Descent. Proceedings of the 22nd International Conference on Machine Learning, ACM, 89-96, 2005.
2. Burges, C. J.; Ragno, R. & Le, Q. V. Schölkopf, B.; Platt, J. C. & Hoffman, T. (Eds.). Learning to Rank with Nonsmooth Cost Functions. Advances in Neural Information Processing Systems 19, MIT Press, 193-200, 2007.
3. Wu, Q.; Burges, C. J.; Svore, K. M. & Gao, J. Adapting Boosting for Information Retrieval Measures. Information Retrieval, Kluwer Academic Publishers, 13, 254-270, 2010.
4. Chapelle, O. & Chang, Y. Chapelle, O.; Chang, Y. & Liu, T.-Y. (Eds.). Yahoo! Learning to Rank Challenge Overview. Proceedings of the Learning to Rank Challenge, PMLR, 14, 1-24, 2011.

论文链接

[Learning to Rank Using Gradient Descent](#)

[Learning to Rank with Nonsmooth Cost Functions](#)

[Adapting Boosting for Information Retrieval Measures](#)

[Yahoo! Learning to Rank Challenge Overview](#)

AI 技术内参

你的360度人工智能信息助理

洪亮劼

Etsy 数据科学主管
前雅虎研究院资深科学家



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 040 | 机器学习排序算法经典模型：GBDT

下一篇 042 | 基于深度学习的搜索算法：深度结构化语义模型

精选留言 (2)

写留言



yaolixu

2018-11-08



会不会出现梯度消失的情况？如果出现了，怎么破？

展开 ∨



吴文敏

2018-01-31



这样会影响模型的收敛速度吧

展开 ∨

