除非另外声明,对 RCV1 和 RCV1-multi 数据集选择 L=7,对 Synthesis 数据集选择 L=8,计算节点数量 W=10.对于其他超参数,候选分裂点数量 q=100,学习速度 η=0.1,特征采样率设为 1.0,树的数量设为 100.

Table 2 Datasets

表 2 数据集

数据集	大小(GB)	样本数量(万)	特征数量(万)	类别数量
RCV1	1.2	69.7	4.7	2
RCV1-multi	0.8	53.4	4.7	53
Synthesis	12	1 000	10	2

4.2 优化方法有效性

本节的实验用来评估本文提出的优化方法,验证它们的有效性,这些优化方法包括数据集转置、稀疏感知建立梯度直方图和比特图压缩.

- 数据集转置.

首先展示分布式数据集转置方法的性能,实验数据如图 6 所示,从 HDFS 上加载 3 个数据集的时间分别是 17s、12s 和 106s.使用简单的直接数据集转置,在 RCV1,RCV1-multi 和 Synthesis 数据集上需要的时间分别是 20s、13s 和 168s;而 FP-GBDT 执行数据集转置的操作,在 3 个数据集上的耗时分别是 9s、6s 和 95s,相比原始的数据集转置方法,速度得到了最高 2 倍的提升.虽然相比于数据并行的策略,FP-GBDT 带来了额外的时间开销,但是在后面的实验中将显示,FP-GBDT 显著提升了整体的性能.

- 稀疏感知建立梯度直方图.

当使用稀疏感知的方法时,对 Synthesis 数据集,建立树的根节点的梯度直方图的时间是 9s.但是当不使用这种方法时,在 1h 之内无法完成建立根节点的梯度直方图,原因是需要读取一个数据样本的所有特征.

- 比特图压缩.

找到最佳分裂点后,处理分裂树节点的任务时,FP-GBDT 广播数据样本的位置时使用比特图的压缩方法.本实验在 Synthesis 数据集上训练梯度提升树算法,建立一棵树的总时间开销中分裂树节点需要 18s;当使用了比特图压缩方法后,此部分时间开销降低为 8s,带来了 2 倍的速度提升.

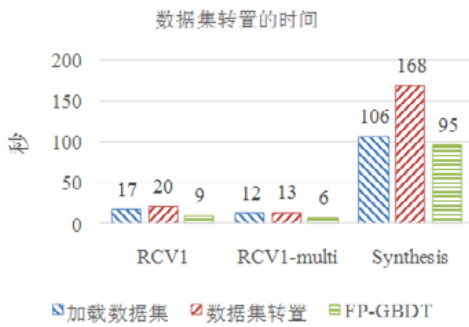

Fig.6 Effects of dataset transposition

图 6 数据集转置的效果

4.3 性能对比

4.3.1 FP-GBDT 与 XGBoost 的性能对比

本节的实验比较 FP-GBDT 与 XGBoost,FP-GBDT 使用特征并行策略,XGBoost 使用数据并行策略,下面分别评估特征数量、类别数量等因素对结果的影响.

- 特征数量的影响.

本文提出的 FP-GBDT 适合高维特征的数据集,为了评估特征数量对于实验结果的影响,本文使用 Synthesis

数据集的一部分特征子集,例如 Synthesis-25K 代表使用 Synthesis 数据集前 2.5 万个特征,Synthesis-100K 代表 Synthesis 数据集的全部特征.分别用 FP-GBDT 与 XGBoost 在 Synthesis 的多个数据子集上训练梯度提升树算法,FP-GBDT 与 XGBoost 对比的实验结果如图 7 所示.当特征数量从 2.5 万增加到 5 万和 10 万时,XGBoost 建立一棵树的时间从 80s 增加到了 144s 和 301s,性能分别下降了 1.8 倍和 3.7 倍.原因是特征数量增大 1 倍时,梯度直方图的大小也增加 1 倍,XGBoost 使用数据并行,通信开销线性增大,从而造成性能几乎线性下降;FP-GBDT 的性能下降较小,分别慢了 1.4 倍和 2.2 倍,FP-GBDT 使用特征并行,通信开销与特征数量无关,树的每一层的开销一样,因而只会受到计算开销增大的影响.总的来说,特征数量增大时 FP-GBDT 比 XGBoost 更加高效.

- 类别数量的影响.

如前文所分析,对于多分类问题,类别数量对梯度直方图的大小有直接的影响.为了更好地显示结果,本实验选择了两分类的 RCV1 数据集和 53 分类的 RCV1-multi 数据集,同时又将 RCV1-multi 的 53 类合并为 5 类.用 8 个计算节点对这 3 个数据集训练梯度提升树模型,图 8 显示了实验结果,统计了建立一棵树的平均时间.在 2 分类上的 RCV1 数据集上,XGBoost 平均需要 53s 训练一棵树,而 FP-GBDT 只需要 26s,训练速度快了 2 倍,所以 FP-GBDT 的收敛速度比 XGBoost 快得多.在 5 分类的 RCV1-multi 数据集上,XGBoost 需要 251s 训练一棵树,FP-GBDT 只需要 41s,快了 6.1 倍.类别增加到 5 类时,梯度直方图的大小增大了 5 倍,XGBoost 的性能下降了接近 5 倍,而 FP-GBDT 只慢了 1.5 倍,这说明 FP-GBDT 更适合多分类问题.当使用 53 类的 RCV1-multi 数据集时,FP-GBDT 训练一棵树需要 161s,而 XGBoost 出现了内存溢出(out of memory,简称 OOM)的错误,这证明了 XGBoost 使用的数据并行策略的内存开销较大.总的来说,FP-GBDT 比 XGBoost 更适合多分类任务,特别是类别数量较大的情况.

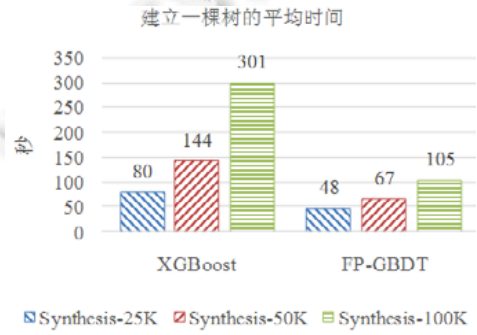

Fig.7 Impact of feature dimensionality

图 7 特征数量的影响

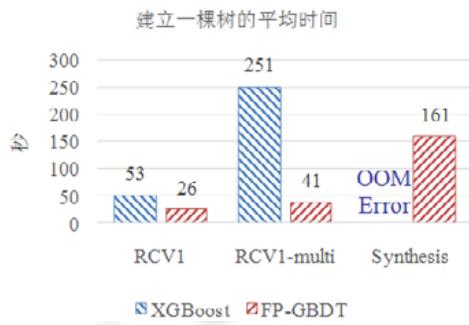

Fig.8 Impact of class number

图 8 类别数量的影响

- 树深度的影响.

接下来的实验将评估树的深度对性能的影响.一般来说,树的深度越大,训练时间越长,模型准确率更高.本实验在 RCV1 数据集上训练梯度提升树算法,逐步地增加树的深度,实验结果显示在图 9 和图 10 中.

深度增大时,XGBoost 和 FP-GBDT 的误差均下降,证明更深的树可以提高模型的准确率,代价是训练时间的增大.当树的深度从 8 增大到 9 时,数据并行策略下梯度直方图的大小增大 1 倍,因此,XGBoost 的运行速度慢了 2 倍;继续将树的深度增大到 10 时,XGBoost 出现了内存溢出的错误.

与此相对应,FP-GBDT 使用特征并行,能够高效地处理更深的树,树的深度增大时,每一层的通信开销保持不变,因此性能下降较为平稳,在可接受范围之内;同时,FP-GBDT 的内存开销较小,树的深度增加时没有出现资源不足的情况.

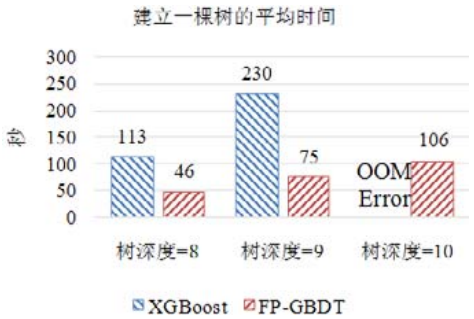

Fig.9 Impact of tree depth (time to build one tree)

图 9 树深度的影响(建立一棵树的平均时间)

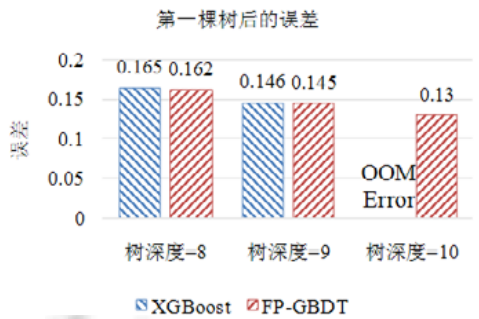

Fig.10 Impact of tree depth (error after the first tree)

图 10 树深度的影响(第 1 棵树后的误差)

4.3.2 FP-GBDT 与 LightGBM 的性能对比

如第 1.4 节所述,LightGBM 也实现了一个特征并行的梯度提升树算法,但是 LightGBM 的实现需要每个计算节点上已经存有完整的数据集,训练时,LightGBM 将整个数据集加载进内存,这样,LightGBM 不需要进行数据集转置的处理,也不需要在节点之间广播分裂后数据样本的位置.严格意义上,这不是一种真正的特征并行方式,在真实的应用中,单机的内存常常无法存储完整的数据集;在真实的环境中,数据集常常以存储在分布式文件系统上,LightGBM 在这种情况下无法工作.因此,LightGBM 的特征并行实现无法用在实际中.

但是为了给出更加全面的结果,本节使用 Synthesis 数据集比较了 FP-GBDT 和 LightGBM 的性能,实验设置与之前的实验保持一致.由于 LightGBM 无法从分布式文件系统读取数据,首先使用 hadoop fs 命令从 HDFS 上下载到每一个计算节点上,在所有计算节点下载完成之后,在每个计算节点上启动 LightGBM,从 HDFS 下载数据集加上本地读取数据的总时间是 LightGBM 的数据预处理的时间.

图 11 给出了实验结果,FP-GBDT 的数据预处理(包括加载数据集和数据集转置)时间是 201s,而 LightGBM 需要 820s 来对数据进行预处理,比 FP-GBDT 慢了 4 倍多;在训练阶段,FP-GBDT 建立一棵树的平均时间是 105s,LightGBM 需要 90s;在内存开销方面,LightGBM 由于需要在每个节点上存储完整的数据集,因此每台机器上消耗了 30GB 的内存,是 FP-GBDT 的内存开销的 6 倍.总体来看,LightGBM 单棵树的速度略快.这主要是因为每个计算节点加载完整的数据集,从而避免了通过网络传输数据样本的分裂位置;很多工业界的数据集大小常常达到 TB 级别,单个计算节点的内存显然无法存储这样大的数据集;对于存储在分布式文件系统上的数据集,LightGBM 首先需要将数据集下载到每个计算节点上,这带来了很大的开销.因此,对于较小的数据集和实验室级别的环境,并且资源较充裕时,LightGBM 是一个不错的选择;但是对于较大的数据集和真实的生产环境,LightGBM 常常是不可用的,而 FP-GBDT 的适用性很广,更加适合大数据时代的需求和发展趋势.

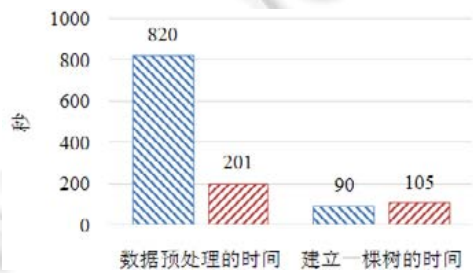

Fig.11 Performance comparison of FP-GBDT and LightGBM

图 11 FP-GBDT 与 LightGBM 的性能对比

5 总 结

本文研究面向高维特征和多分类问题的分布式梯度提升树算法.根据一个严格的代价模型,比较了数据并行策略和特征并行策略,证明了特征并行策略更适合高维和多分类场景;基于理论分析的结果,本文提出一种使用特征并行的分布式梯度提升树算法 FP-GBDT.FP-GBDT 首先利用对数据集进行分布式转置,转换为特征并行需要的数据表征;在建立梯度直方图时,FP-GBDT 设计了稀疏感知的方法;在分裂树节点时,FP-GBDT 使用一种比特图压缩的方法传输数据样本的位置.实验结果显示,FP-GBDT 与使用数据并行的 XGBoost 相比,性能的提升最高达到 6 倍以上,证明了特征并行的 FP-GBDT 在高维和多分类问题上的高效性.

References:

[1] Friedman J, Hastie T, Tibshirani R, *et al*. Additive logistic regression: A statistical view of boosting. The Annals of Statistics, 2000, 28(2):337−407.

[2] Friedman JH. Greedy function approximation: A gradient boosting machine. The Annals of Statistics, 2001, 1189−1232.

[3] Li P. Robust logitboost and adaptive base class (abc) logitboost. arXiv:1203.3491. arXiv Preprint, 2012.

[4] Burges CJ. From ranknet to lambdarank to lambdamart: An overview. Learning, 2010,11(23-581):81.

[5] Stephen T, Weinberger KQ, Agrawal K, Paykin J. Parallel boosted regression trees for Web search ranking. In: Proc. of the 20th Int'l Conf. on World Wide Web. 2011. 387−396.

[6] He X, Pan J, Jin O, Xu T, Liu B, Xu Tao, Shi Y, Atallah A, Herbrich R, Bowers S, *et al*. Practical lessons from predicting clicks on ads at facebook. In: Proc. of the 8th Int'l Workshop on Data Mining for Online Advertising. 2014. 1−9.

[7] Meng X, Bradley J, Yavuz B, Sparks E, Venkataraman S, Liu D, Freeman J, Tsai DB, Amde M, Owen S, *et al*. Mllib: Machine learning in apache spark. The Journal of Machine Learning Research, 2016,17(1):1235−1241.

[8] Chen TQ, Guestrin C. Xgboost: A scalable tree boosting system. In: Proc. of the 22nd ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining. 2016. 785−794.

[9] Meng Q, Ke GL, Wang TF, When W, Ye QW, Ma ZM, Liu TY. A communication-efficient parallel algorithm for decision tree. In: Proc. of the Advances in Neural Information Processing Systems. 2016. 1279−1287.

[10] Ke GL, Meng Q, Finley T, Wang TF, Chen W, Ma WD, Ye QW, Liu TY. Lightgbm: A highly efficient gradient boosting decision tree. In: Proc. of the Advances in Neural Information Processing Systems. 2017. 3149−3157.

[11] Karnin Z, Lang K, Liberty E. Optimal quantile approximation in streams. In: Proc. of the 57th IEEE Annual Symp. on Foundations of Computer Science (FOCS). 2016. 71−78.

[12] Greenwald M, Khanna S. Space-efficient online computation of quantile summaries. ACM SIGMOD Record, 2001,30:58−66.

[13] Yahoo. Data sketches. https://datasketches.github.io/

[14] Jiang J, Jiang JW, Cui B, Zhang C. Tencentboost: A gradient boosting tree system with parameter server. In: Proc. of the 2017 IEEE 33rd Int'l Conf. on Data Engineering. 2017. 281−284.

[15] Jiang JW, Cui B, Zheng C, Fu FC. DimBoost: Boosting gradient boosting decision tree to higher dimensions. In: Proc. of the 2018 Int'l Conf. on Management of Data. 2018. 1363−1376.

[16] Abadi M, Barham P, Chen JM, Chen ZF, Davis A, Dean J, Devin M, Ghemawat S, Irving G, Isard M, *et al*. Tensorflow: A system for large-scale machine learning. OSDI, 2016,16:265−283.

[17] Jiang JW, Cui B, Zhang C, Yu LL. Heterogeneity-aware distributed parameter servers. In: Proc. of the 2017 ACM Int'l Conf. on Management of Data. 2017. 463−478.

[18] Li M, Andersen DG, Park JW, Smola AJ, Ahmed A, Josifovski V, Long J, Shekita EJ, Su BY. Scaling distributed machine learning with the parameter server. OSDI, 2014,14:583−598.

[19] Zaharia M, Chowdhury M, Das T, Dave A, Ma J, McCauley M, Franklin MJ, Shenker S, Stoica I. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: Proc. of the 9th USENIX Conf. on Networked Systems Design and Implementation. 2012. 2.

[20] Vavilapalli VK, Murthy AC, Douglas C, Agarwal S, Konar M, Evans R, Graves T, Lowe J, Shah H, Seth S, *et al*. Apache hadoop yarn: Yet another resource negotiator. In: Proc. of the 4th Annual Symp. on Cloud Computing. 2013. 5.

[21] RCV1 dataset. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html#rcv1.binary

[22] RCV1-Multi dataset. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multiclass.html#rcv1.multiclass

[23] Lewis DD, Yang YM, Rose TG, Li Fan. Rcv1: A new benchmark collection for text categorization research. Journal of Machine Learning Research, 2004,5:361−397.

江佳伟(1990－),男,湖北洪湖人,博士,CCF 学生会员,主要研究领域为机器学习.

符芳诚(1996－),男,学士,主要研究领域为机器学习.

邵蓥侠(1988－),男,博士,副研究员,博士生导师,CCF 专业会员,主要研究领域为数据库,知识图谱数据管理,并行图计算,知识工程.

崔斌(1975－),男,博士,教授,博士生导师,CCF 杰出会员,主要研究领域为数据库,大数据管理分析.