计算机科学
COMPUTER SCIENCE

面向最优直方图求解的监督学习模型研究

陈云亮, 刘浩, 朱桂水, 黄晓辉, 陈小岛, 王力哲

引用本文

陈云亮, 刘浩, 朱桂水, 黄晓辉, 陈小岛, 王力哲. 面向最优直方图求解的监督学习模型研究[J]. 计算机科学, 2023, 50(9): 145-151.

CHEN Yunliang, LIU Hao, ZHU Guishui, HUANG Xiaohui, CHEN Xiaodao, WANG Lizhe. Study on Supervised Learning Model for Optimal Histogram Solution[J]. Computer Science, 2023, 50(9): 145-151.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230300065

面向最优直方图求解的监督学习模型研究

陈云亮　刘　浩　朱桂水　黄晓辉　陈小岛　王力哲
中国地质大学(武汉)计算机学院　武汉 430070
(chenyunliang@cug.edu.cn)

摘　要　最优直方图是一类重要的直方图技术,目前用于实现最优直方图的动态规划分组算法存在时间复杂度过高的问题。因此,提出了一种基于概率稀疏自注意力的监督学习模型来学习动态规划分组算法,该监督学习模型可作为动态规划分组算法的替代方案,主要包括3个部分:1)通过Embedding层与位置编码层将输入数值序列映射为对应的向量序列;2)通过概率稀疏的自注意力层捕获输入序列之间的依赖关系;3)通过前馈神经网络层将依赖关系映射到分组"桶"边界下标信息。实验结果表明,基于概率稀疏自注意力的监督学习模型在6个数据集上的准确率超过了83.47%,且其在预测阶段的时间消耗不超过动态规划分组算法的1/3。

关键词:最优直方图;动态规划分组算法;监督学习模型
中图法分类号　TP391

Study on Supervised Learning Model for Optimal Histogram Solution

CHEN Yunliang,LIU Hao,ZHU Guishui,HUANG Xiaohui,CHEN Xiaodao and WANG Lizhe
School of Computer Science,China University of Geosciences,Wuhan 430070,China

Abstract　The dynamic programming binning algorithm is currently used to realize the optimal histogram. However,its time complexity is too high. A supervised learning model based on ProbSparse self-attention is proposed in this paper to learn the dynamic programming binning algorithm. It can be used as an alternative to the dynamic programming binning algorithm. The proposed model consists of three parts:1)mapping the numerical input sequence into the corresponding vector sequence through the embedding and position coding layer;2)capturing the dependence between input sequences through the ProbSparse self-attention;3)the dependency is mapped to the subscript information of the binning "bucket" boundary through the feedforward neural network. Experimental results on six data sets indicate that the proposed model based on ProbSparse self-attention outperforms the dynamic programming binning algorithm. The accuracy of the proposed method is greater than 83.47%. Meanwhile,its time cost in the prediction stage is no more than 1/3 of the compared method.

Keywords　Optimal histogram,Dynamic programming binning algorithm,Supervised learning model

1　引言

直方图技术[1]是一种应用较为广泛的数据摘要技术,目前有许多商业关系型数据库系统使用直方图[2]来汇总数据集。通俗地说,直方图技术就是对待分组的数值序列,按照某种策略将其分配到多个不相交的"桶"中,然后使用一个估计值代替同一个"桶"内所有元素的值,起到数据压缩的作用。数据分布在数据查询中非常有用,但通常因占据太多存储空间而无法准确存储,而直方图可作为一种近似机制发挥作用。

常见的直方图包括等宽(距离)直方图、等深(频率)直方图、end-biased直方图、压缩直方图以及最优直方图[3]等。其中,最优直方图是一类基于平方误差最小化的直方图技术,在估算简单相等连接、估算选择查询[2]的结果大小时,最优直方图是逼近数据库中属性值频率的最佳选择。除此之外,对于一些选择性估计问题[4],最优直方图被证明可以最小化平均误差。因此,最优直方图在学术界和工业界引发了较多的关注。

最优直方图问题可通过动态规划分组算法[5-6]来解决。动态规划分组算法的基本原理是:在给定待分组序列的所有$N-1$分组的动态规划解的情况下,可以递推得到待分组序

到稿日期:2023-03-08　返修日期:2023-06-18
基金项目:国家自然科学基金面上项目(62076224);国家杰出青年科学基金(41925007);国家自然科学基金联合基金(U21A2013);智能地学信息处理湖北省重点实验室开放研究课题(KLIGIP-2022B16)
This work was supported by the National Natural Science Foundation of China(62076224),National Science Fund for Distinguished Young Scholars of China(41925007),Joint Funds of the National Natural Science Foundation of China(U21A2013) and Open Research Project of The Hubei Key Laboratory of Intelligent Geo-Information Processing(KLIGIP-2022B16).
通信作者:黄晓辉(xhhuang@cug.edu.cn)

列的 N 分组的动态规划解。动态规划分组算法考虑的是在最小化所有“桶”的平方误差的约束条件下，递推地构造最优直方图问题的解。其一般流程是：首先解决规模最小的问题，得到将待分组序列分配到 1 个“桶”时的动态规划解；然后，解决规模次小的问题，得到将待分组序列分配到 2 个“桶”时的动态规划解；依次类推，最终利用 $N-1$ 个“桶”时的动态规划解得到最终问题，即 N 个“桶”时的动态规划解。之所以使用最小化平方误差之和作为优化目标，原因是动态规划分组算法假设使用所有“箱”内的平方误差之和表示信息的损失程度，在该假设下，其目标即是尽可能在损失最少信息的情况下进行“分桶”，即在“分桶”的同时尽最大可能保留数据的原始信息。

根据上述动态规划分组算法的实现，该分组算法能够达到最小的分组误差。但是该算法的时间复杂度是 $O(NM^2)$，其中 M 表示序列的长度，N 表示动态规划分组的“桶”数量，其与待分组序列的长度的平方成正比。因此，随着待分组序列长度的增大，其时间成本会快速增加。

针对动态规划分组算法时间复杂度过高的问题，现阶段主要通过引入启发式信息来解决[7]。启发式算法的主要思路是利用某种启发式信息构建局部最优解，然后在局部最优解的基础上，进一步通过搜索等方法来实现最优直方图。这类启发式算法虽然在一定程度上能够降低原始动态规划分组算法的时间消耗，但并没有从全局的角度去考虑实现最优直方图。因此，这类方法一般不能实现全局最优。

本文针对动态规划分组算法存在的时间复杂度过高的问题，提出使用基于概率稀疏自注意力的监督学习模型来进行改进。该模型的训练主要包括两个部分：1）通过动态规划分组算法对输入数据集进行分组操作，得到分组“桶”下标、分组误差与平均分组时间，然后分割得到训练数据、验证数据与测试数据；2）将训练数据中的待分组序列“喂入”Embedding 层与位置编码层以映射为对应的向量序列，然后将向量序列“喂入”概率稀疏的自注意力层以捕获输入序列之间的依赖关系，然后通过前馈神经网络得到预测分组“桶”边界下标，最后通过预测分组“桶”边界与真实分组“桶”边界计算模型误差并通过梯度反向传播更新模型的权重。该模型的目标是，在预测阶段，将测试数据中待分组序列输入模型得到预测分组“桶”边界，使得该预测分组“桶”边界在一定程度上接近动态规划分组“桶”边界，且消耗更少的时间。

本文第 2 章描述最优直方图问题的相关国内外研究现状；第 3 章对动态规划分组算法进行详细介绍，并提出基于概率稀疏自注意力的监督学习模型；第 4 章是实验设计与结果分析；最后总结全文并展望未来。

2 国内外研究现状

近年来，不少学者都对直方图问题进行了深入的研究。Kaushik 等[8]提出了一种高效的 one-pass 算法用于解决最优直方图问题，该算法通过一次处理一个查询，并增量地计算直方图，提升了在查询结果大小估计方面的精度，同时还提高了计算效率。Greenwald 等[9]提出了一种在线计算方法来求解等宽直方图问题，该方法不需要参数的先验知识，同时优化了算法的空间复杂度。Fang 等[10]则对 End-biased 直方图中的冰山查询问题提出了一种计算结果的算法，通过 3 个实际应用中的算法对比，验证了该算法相比传统算法的高效性。由于动态规划算法具有全局性，相比其他传统算法可以得到最优直方图问题的最优解，因此本章后续详述基于动态规划算法求解直方图问题的国内外现状。

Jagadish 等[5]首先提出了适用于最优直方图问题的动态规划分组算法，该方案将最优直方图问题视作最优化问题。针对动态规划分组算法的拓展优化，主体上都是集中在对时间复杂度的优化和对适用对象的拓展两个方面，而对于其时间复杂度过高这一问题的主要优化方向是在牺牲一定动态规划分组算法的准确度的基础上引入启发式信息[11]。Jagadish 等提出了一种新的、更快速的近似解决方案，这种近似解决方案结合了启发式算法与原始动态规划分组算法，该算法可以线性地降低原动态规划解法的时间复杂度。通过该近似解决方案，可以在牺牲一定准确度的情况下将整体算法的最坏时间复杂度降低到 $O((M^2N)/l)$。文献[12]提出使用随机化的启发式算法作为优化查询的一种方法，其总体思路可应用于构建最佳直方图。Poosala 等[13]参考了这种思路，提出可以通过创建随机解决方案，以这些随机解决方案为起点进行改进，找到一种新的解决方案。Ioannidis 等[14]提出可以使用迭代改进算法和模拟退火算法来改进初始的随机解决方案，同时也可以在运用两阶段优化算法的过程中结合两者。此外，对于流数据[15]，通过引入启发式约束条件，也可以降低最优直方图构建的时间复杂度到多项式级别[16-18]。

总的来说，引入启发式策略的分组算法主要是通过启发式地寻找局部的最优解来解决最优直方图问题，这类启发式算法相比动态规划分组算法，在一定程度上消耗更少的时间，但是这类算法并没有从全局考虑实现最优直方图的目标。

另一方面，以神经网络为代表的监督学习模型在策略学习、优化问题等领域取得了一定的突破。Nomer 等[19]在研究中训练神经网络学习相应的贪心策略以解决背包问题，Zaremba 等[20]提出使用自然语言处理领域的 Seq2Seq 模型预测伪代码片段的运行结果，最终训练所得的 Seq2Seq 模型可以根据“喂入”的伪代码序列预测伪代码的执行结果。Graves 等[21]通过训练一种称为神经图灵机的深度神经网络来学习简单的“复制”“有限的排序”与“相关度召回”等策略。Chen 等[22]的研究表明波束搜索算法[23]一定程度上可以通过有监督学习模型学习。Graves 等[24]提出了一种称为可微神经计算机(DNC)的机器学习模型。该模型由一个能读写外部存储器矩阵的神经网络组成，可以在一定程度上模拟自然语言处理中的推理等操作。除此之外，该模型还可以在一定程度上找到图论中点对点之间的最短路径。而且经过强化学习训练后的可微神经计算机模型，甚至能完成一个由符号序列指定变化目标的移动方块拼图任务。Graves 等的研究结果表明，可微神经计算机模型有一定能力解决复杂的结构化任务。

本文针对上述有关最优直方图问题的研究现状与以神经网络为代表的监督学习模型在策略学习方面的研究现状，拟将最优直方图问题与监督学习模型两者结合起来，研究使用监督学习模型学习动态规划分组算法。

3 学习动态规划分组算法的监督学习模型

本章阐述使用监督学习模型学习动态规划分组算法的主要思路：首先应用动态规划分组算法对待分组数据进行分组，得到分组结果；然后，将待分组序列作为特征，动态规划分组结果作为标签训练基于概率稀疏自注意力的监督学习模型，使得该监督学习模型学习动态规划分组算法。从而，在预测阶段，基于概率稀疏自注意力的监督学习模型能够根据输入的待分组序列，预测一个与动态规划分组结果相似的分组“桶”边界。

3.1 动态规划分组算法

动态规划分组算法的伪代码如算法 1 所示，第 1－6 行伪代码的功能是初始化，主要是初始化几个变量以存放中间状态；第 7－32 行是动态规划分组算法的核心代码，按照不同的功能可以分为 3 个部分。

第一部分：第 7－9 行的代码表示顺序遍历 X，得到 X 的前缀和与其平方值的前缀和，便于在接下来的算法中计算平方误差，这部分算法的时间复杂度是 $O(n)$，空间复杂度是 $O(n)$。

第二部分：第 10－24 行的伪代码则表示使用递推关系来一步步构建最终问题的动态规划分组解。其中第 12－15 行的伪代码表示当 $k=1$ 时的动态规划分组解，这也是在整个序列上动态规划分组解的基础；而第 17－24 行的伪代码则表示按照递推关系构建最终问题的动态规划分组解，这部分的时间复杂度为 $O(Bn^2)$，空间复杂度为 $O(Bn)$。

第三部分：第 26－32 行的代码表示根据动态规划分组解的中间过程确定分组“桶”边界的过程，这部分代码的时间复杂度为 $O(B)$，空间复杂度为 $O(1)$。

通过以上对动态规划分组算法的详细分析，可以得到整个动态规划分组算法的时间复杂度为 $O(Bn^2)$，其空间复杂度为 $O(Bn)$。

算法 1 动态规划算法

输入：数值序列 $X=x_1,x_2,\cdots,x_n$，“桶”数量 B，其中 $1<B<n$

输出：动态规划分组的“桶”边界下标数组

```
1. 初始化 p[1,…,n]数组辅助计算平方误差
2. pp[1,…,n]数组辅助计算平方误差
3. err[1,…,n][1,…,B]数组记录分组误差
4. idx[1,…,n][1,…,B]数组临时存放“桶”边界
5. A[1,…,B－1]数组存放分组“桶”边界下标
6. 令 p[1]←x1，pp[1]←x1^2，idx[1]=1
7. for i←2 to n do
8.     then p[i]←p[i－1]＋xi
9.     pp[i]←pp[i－1]＋xi^2
10. for k←1 to B do
11.    for i←1 to n do
12.      if k=1
13.        then s2←pp[i]
14.        s1←p[i]
15.        err[i][k]←s2－s1×s1/i
16.        else
17.        then err[i][k]←inf
18.        for j←1 to i－1 do
19.          then s2←pp[i]－pp[j]
20.          s1←p[i]－p[j]
21.          temp←s2－s1×s1/(i－j)
22.          if err[j][k－1]＋temp＜err[i][k]
23.            then err[i][k]←err[j][k－1]＋temp
24.            idx[i][k]←j＋1
25. 令 i←B,j←n
26. while i≥2
27.    do ep←j
28.    j←idx[j][i]
29.    idx[i－1]←j－1
30.    i←i－1
31.    j←j－1
32. end
33. return(idx,err[n][B])
```

3.2 用于学习动态规划分组算法的监督学习模型

为了学习动态规划分组算法，本节设计了一个基于概率稀疏自注意力[25-29]的监督学习模型。该监督学习模型的结构如图 1 所示。

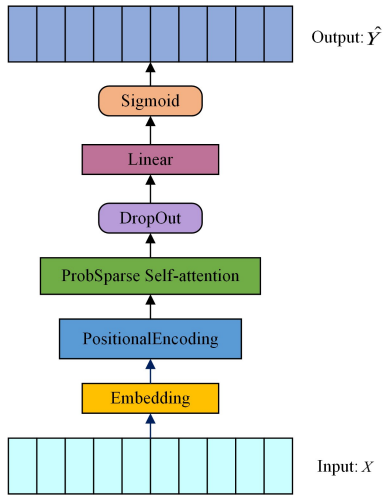

图 1 基于概率稀疏自注意力的监督学习模型

Fig. 1 Supervised learning model based on probSparse self-attention

该模型主要包括 Embedding 层[30]、位置编码层（Positional Encoding）[31]、概率稀疏自注意力层（ProbSparse self-Attention）与前馈神经网络层。其中，Embedding 层将数值转化为一个具有确定维度的向量，通过结合位置编码层的位置信息，将其“喂入”ProbSparse self-Attention 层从而捕捉输入向量之间的相互依赖关系，接着通过前馈神经网络层的线性映射得到一个维度为 $n-1$ 的向量，然后将该向量作为 Sigmoid 激活函数的输入，将其转化为一个维度为 $n-1$ 且每个分量都位于区间[0,1]上的向量作为输出。其输出向量的任意一个分量都表示在该分量对应下标处作为动态规划分组“桶”边界的概率，该分量越大，表示该分量对应下标处作为分组“桶”边界的概率越大。

学习动态规划分组算法的过程可以分为以下 3 个阶段。

数据预处理阶段：该阶段首先按照在 3.1 节的分析得到待分组的数值序列 $X=\{x_1,x_2,\cdots,x_n\}$，然后使用 3.1 节分析的

动态规划分组算法对数值序列执行分组，得到“桶”边界下标序列 $Y=\{y_1,y_2,\cdots,y_{B-1}\}$ 以及动态规划分组的误差 $Error=\sum_{i=1}^{B}Error_i$。为了计算模型损失，需要将 Y 转换为一个 $n-1$ 维的向量 $\boldsymbol{Y}\in\mathbb{R}^{n-1}$，且“桶”边界对应下标处对应的分量设置为 1，其余分量设置为 0。除此之外，对应小批量训练的输入矩阵可表示为 $\boldsymbol{X}\in\mathbb{R}^{m\times n}$，相应的目标矩阵可表示为 $\boldsymbol{Y}\in\mathbb{R}^{m\times(n-1)}$，其中 m 表示小批量训练中一个批次包括的数据样本数量。

监督学习模型的训练阶段：该阶段主要需要考虑的是数据流的前向传播。本节提出的基于概率稀疏自注意力的监督模型的结构如图 1 所示，该模型对应的数据流从第一层传递到最后一层。该前向过程可以分为以下几个部分：

1）Embedding 层。在第一层，矩阵 $\boldsymbol{X}\in\mathbb{R}^{m\times n}$ 作为 Embedding 层的输入。Embedding 层将数字映射到一个确定维度的向量空间，设该向量空间的维度为 d，输入 $\boldsymbol{X}$ 经过 Embedding 层的映射之后，输出为矩阵 $\boldsymbol{X}_e\in\mathbb{R}^{m\times n\times d}$。

2）位置编码层。由于概率稀疏自注意力网络处理序列数据的方式与 LSTM 依次处理的方式不同，自注意力网络先天地丢失了数据序列中元素的先后位置关系，因此需要额外使用一个位置编码层（Positional Embedding）来对序列数据的位置进行编码。该位置编码层使用正弦、余弦函数来对位置进行编码，位置编码层生成一个与输入序列等长的且向量维度相等的位置向量 $\boldsymbol{X}_p\in\mathbb{R}^{m\times n\times d}$，然后将输入向量与位置向量相加作为输出 $\boldsymbol{X}_{ep}\in\mathbb{R}^{m\times n\times d}$。

3）概率稀疏的自注意力层。将 $\boldsymbol{X}_{ep}$ 输入概率稀疏的自注意力层（ProbSparse Self-Attention），在概率稀疏的自注意力层，分两步计算注意力得分与注意力值。首先，计算查询矩阵 $\bar{\boldsymbol{Q}},\boldsymbol{K}$ 与值矩阵 $\boldsymbol{V}$，然后通过第一次矩阵乘法操作 softmax($\bar{\boldsymbol{Q}}\boldsymbol{K}^{\mathrm{T}}$）计算得到注意力得分，最后通过第二次矩阵乘法操作 softmax($\bar{\boldsymbol{Q}}\boldsymbol{K}^{\mathrm{T}}$)$\boldsymbol{V}$ 得到该层的注意力值输出，记为 $\boldsymbol{X}_{\mathrm{attn}}\in\mathbb{R}^{m\times n\times d}$。然后，将各个位置的注意力值向量展开，得到矩阵 $\boldsymbol{X}_{\mathrm{attn}}\in\mathbb{R}^{m\times(nd)}$。

概率稀疏的自注意力机制的出发点是：传统的自注意力机制是一种全局自注意力模式，即对于任意的一个 query，其在计算注意力得分、注意力值时需要考虑到所有的 key。不失一般性，使用 $\boldsymbol{q}_i$ 表示一个 query 向量，$\boldsymbol{q}_i$ 的注意力值 $\mathcal{A}(\boldsymbol{q}_i,\boldsymbol{K},\boldsymbol{V})$ 可由式（1）计算得到，其中 $\boldsymbol{q}_i,\boldsymbol{k}_i,\boldsymbol{v}_i$，分别代表 $\bar{\boldsymbol{Q}},\boldsymbol{K},\boldsymbol{V}$ 中的第 i 行。

$$\mathcal{A}(\boldsymbol{q}_i,\boldsymbol{K},\boldsymbol{V})=\sum_j\frac{\exp(q_i,k_j)}{\sum_l\exp(q_i,k_l)}v_j=\mathbb{E}_{p(k_j|q_i)}[v_j] \tag{1}$$

如式（1）所示，注意力值的计算也可被视作一个计算期望的过程，而注意力得分相当于一个概率分布 $p(k_j|q_i)$。对于 $p(k_j|q_i)$ 而言，如果其近似于一个均匀分布 $q(k_j|q_i)=1/T$，意味着 $\boldsymbol{q}_i$ 对所有 key 几乎同样重要（或同样不重要），则这个 $p(k_j|q_i)$ 对应的 query 向量 $\boldsymbol{q}_i$ 没有起到“注意力”的作用，因此可认为这类 $\boldsymbol{q}_i$ 相对而言是不重要的，可以不参与注意力值的计算。

4）前馈神经网络层：将展开后的注意力值 $\boldsymbol{X}_{\mathrm{attn}}$ 输入多个前馈神经网络线性映射到更低维度，得到 $\boldsymbol{X}_l\in\mathbb{R}^{m\times(n-1)}$。

5）Sigmoid 激活函数：最后，将 $\boldsymbol{X}_l$ 的每一个元素都使用 Sigmoid 函数进行激活处理，得到最终的输出矩阵 $\hat{\boldsymbol{Y}}\in\mathbb{R}^{m\times(n-1)}$。

在得到预测矩阵 $\hat{\boldsymbol{Y}}$ 之后，使用二分类的交叉熵函数计算 $\hat{\boldsymbol{Y}}$ 与 $\boldsymbol{Y}$ 的之间的损失。然后通过反向传播的方法逐层传递误差并更新基于概率稀疏自注意力的监督学习模型每一层的权重。不断重复以上训练过程，直到模型收敛为止，即可完成基于概率稀疏自注意力的监督学习模型的训练。

第三阶段：对监督学习模型的预测结果进行分析，当监督学习模型训练完成之后，使用该模型的预测结果与真实结果来评估该模型的性能。首先使用该模型进行预测，令矩阵 $\boldsymbol{X}^{\mathrm{test}}\in\mathbb{R}^{t\times n}$ 与矩阵 $\boldsymbol{Y}^{\mathrm{test}}\in\mathbb{R}^{t\times(n-1)}$ 分别表示测试数据集的输入序列矩阵与目标序列矩阵，将 $\boldsymbol{X}^{\mathrm{test}}$ 输入训练后的监督学习模型，便可得到其预测结果 $\hat{\boldsymbol{Y}}^{\mathrm{pre}}\in\mathbb{R}^{t\times(n-1)}$，其中 $\hat{\boldsymbol{Y}}^{\mathrm{pre}}$ 表示对于第 i 条输入序列，其在该序列的下标 j 处作为动态规划分组“桶”边界的概率，选择概率最高的 $B-1$ 个下标作为模型预测动态规划分组的“桶”边界。按照如上策略对 $\hat{\boldsymbol{Y}}^{\mathrm{pre}}$ 进行转换可得到预测的目标序列矩阵 $\boldsymbol{Y}^{\mathrm{pre}}\in\mathbb{R}^{t\times(B-1)}$，然后通过预测的动态规划分组的“桶”边界确定“桶”，并计算模型的分组误差 $Error_{\mathrm{pre}}$。通过 $\boldsymbol{X}^{\mathrm{test}}$ 与 $\boldsymbol{Y}^{\mathrm{test}}$ 计算动态规划分组误差 $Error_{\mathrm{true}}$，最后比较 $Error_{\mathrm{pre}}$ 与 $Error_{\mathrm{true}}$ 两者的差距。除此之外，在处理测试数据集时，还需要分别记录该模型预测分组时的时间消耗与动态规划分组算法的时间消耗，通过对比两者的时间消耗，来评估该模型的时间性能。

4 实验及结果分析

本章主要涉及的是基于概率稀疏自注意力的监督学习模型，该模型被用于学习动态规划分组算法。概率稀疏的自注意力机制通过稀疏化查询向量矩阵 $\boldsymbol{Q}$，相当于实现了一种局部自注意力机制，通过改进的 KL 散度度量能够在 $O(\log T)$ 的时间复杂度与空间复杂度下找到 $\log T$ 个更加具有关联性的 query 组成新的查询向量矩阵，该矩阵标记为 $\bar{\boldsymbol{Q}}\in\mathbb{R}^{\log T\times d}$。然后，按照式（1）计算注意力得分与相应的注意力值。因为新的经过稀疏化后处理后的查询矩阵 $\bar{\boldsymbol{Q}}$ 的维度为 $\log T$，相比 $\boldsymbol{Q}$ 的维度 T 而言降低了一个数量级，在计算注意力得分与相应的注意力值时便可以将时间复杂度与空间复杂度降低一个数量级，达到 $O(T\log T)$。

本章从多个角度测试该监督学习模型的性能，其中，对于该模型作为分类器的性能，使用准确率、near-3 准确率（预测的“桶”边界在真实的“桶”边界左边或者右边 3 个单位距离内的准确性）等进行评估，对于该模型的分组性能，从分组误差、分组时间两个方面来测试该替代方案的性能。

根据待分组序列 $X=\{x_1,x_2,\cdots,x_m\}$ 的长度 m 与“桶”数量 n 的不同取值，本章实验涉及的数据集有 6 个，其中，m 的值分别取 100 和 200，n 的值分别取 5，10 与 15。总体而言，本章使用到的数据集的类型及其标记如表 1 所列。

表 1 训练基于概率稀疏自注意力的监督学习模型的数据集

Table 1 Data set for training supervised learning model based on probSparse self-attention

n	m	
	100	200
5	m100n5-Optimal	m200n5-Optimal
10	m100n10-Optimal	m200n10-Optimal
15	m100n15-Optimal	m200n15-Optimal

对于数据集 m100n5-Optimal，m100n10-Optimal 以及 m100n15-Optimal，这三者的数据集规模是一致的，总共包括了 100000 条数据，每条数据都包括一个长度为 100 的序列 X 与长度为 99 的序列 Y，其中序列 X 表示待分组序列，其元素值在区间[0，100]内，序列 Y 表示动态规划分组的“桶”下标序列，其元素值在集合{0，1}内。m200n5-Optimal，m200n10-Optimal 以及 m200n15-Optimal 这 3 个数据集的规模是相同的，总共包括了 100000 条数据，每条数据都包括一个长度为 200 的序列 X 与长度为 199 的序列 Y，其中序列 X 表示待分组序列，其元素值在区间[0，20]内，序列 Y 表示动态规划分组的“桶”下标序列，其元素值在集合{0，1}内。

本实验中，将以上 6 个数据集都随机分割为训练数据集、验证数据集与测试数据集 3 部分，其中训练数据集包含 94000 条数据，而验证数据集与测试数据集分别包含 3000 条数据。

对于以上所有的数据集，本章主要围绕以下两个方面进行实验研究：

1）分类性能：分析该监督学习模型作为分类器的情况下，在测试数据集上的分类性能，主要使用准确率、near-3 准确率等评价参数，以验证模型在测试数据集上的分类性能。

2）分组性能：在测试数据集上，将预测结果用于分组时，记录模型的预测分组误差与预测分组时间，并将预测分组误差与真实的动态规划分组误差相比较，计算两者的相对误差以及相对误差的最大值、最小值、平均值与中位数等；同时，将预测分组时间与动态规划分组的时间进行对比，以综合评估该监督学习模型学习动态规划分组算法的能力。

基于概率稀疏自注意力的监督学习模型可视为序列数据上的二分类标注模型，可使用准确率、near-3 准确率来评估该模型在测试数据集上的分类性能。表 2 列出了该模型在各个测试数据集上的平均准确率与 near-3 准确率。

表 2 平均准确率与平均 near-3 准确率

Table 2 Average accuracy and average near-3 accuracy

测试数据集	平均准确率	平均 near-3 准确率
m100n5-Optimal	0.8658	0.9088
m100n10-Optimal	0.8806	0.9311
m100n15-Optimal	0.9005	0.9610
m200n5-Optimal	0.8580	0.8935
m200n10-Optimal	0.8442	0.8862
m200n15-Optimal	**0.8348**	**0.8710**

分析表 2 可知，该模型在测试数据集上的平均准确率都超过了 83.48%，而平均 near-3 准确率也超过了 87.10%，说明在将动态规划分组算法转换为一个序列标注任务后，该监督学习模型在测试数据集上能够取得良好的分类性能。

若要将基于概率稀疏自注意力的监督学习模型作为动态规划分组算法的一种替代方案，则需要考虑该模型的预测分组误差。表 3 列出了监督学习模型的预测分组误差与动态规划分组误差的对比，包括预测分组误差等于或大于真实分组误差两种情况。在 6 个数据集上，预测分组误差与动态规划分组误差完全相等的测试数据所占的比例在 25.53%～61.60%之间，预测分组误差大于动态规划分组误差的测试数据所占的比例在 38.40%～74.47%之间。而更加详尽的预测分组误差大于动态规划分组误差的测试数据如表 4 所列。

表 3 预测分组误差 vs 真实分组误差

Table 3 Predictive grouping error vs. real grouping error

测试数据集	预测分组误差等于真实分组误差的测试数据量	预测分组误差大于真实分组误差的测试数据量
m100n5-Optimal	**1848**	**1152**
m100n10-Optimal	831	2169
m100n15-Optimal	716	2284
m200n5-Optimal	853	2147
m200n10-Optimal	801	2199
m200n15-Optimal	**766**	**2234**

对于测试数据中预测分组误差大于真实分组误差的情况，本实验使用两者的相对误差来衡量两者的差距，表 4 列出了相对误差的相关统计信息，应用了最大值、最小值、平均值与中位数这 4 个统计量。在 6 个测试数据集中，相对误差的最大值在 14.88%～35.06%之间，相对误差的平均值在 1.67%～6.28%之间，且相对误差的中位数表明至少有一半的数据相对误差低于 5.70%。因此，通过分析这 4 个统计量可以看出，即使在预测分组误差大于真实分组误差的测试数据中，其相对误差总体上还是保持在一个低小的水平。

表 4 相对误差的统计信息

Table 4 Statistical information of relative error

测试数据集	最大值	最小值	平均值	中位数
m100n5-Optimal	0.1489	0.00012	**0.0167**	0.0062
m100n10-Optimal	**0.1488**	0.00012	0.0168	0.0062
m100n15-Optimal	**0.3506**	0.00016	0.0442	0.0329
m200n5-Optimal	0.1667	0.00012	0.0584	**0.0570**
m200n10-Optimal	0.2977	0.00052	0.0529	0.0510
m200n15-Optimal	0.3048	0.00069	**0.0628**	0.0554

除了从以上的分类任务的评价指标与分组误差的评价指标来对基于概率稀疏自注意力的监督学习模型进行评估以外，本章还比较了该模型的预测分组时间与动态规划分组时间，两者的对比结果如图 2、图 3 所示。

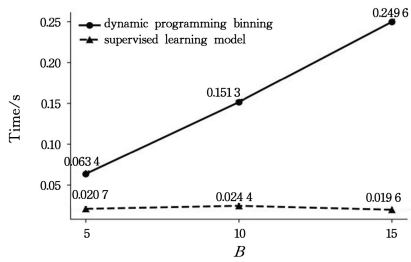

图 2 序列长度为 100 时的时间消耗对比

Fig. 2 Comparison of time consumption when the sequence length is 100

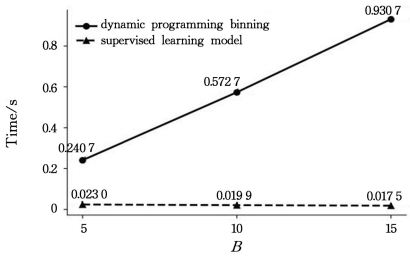

图 3 序列长度为 200 时的时间消耗对比

Fig. 3 Comparison of time consumption when the sequence length is 200

在图 2 与图 3 中，黑色实线表示动态规划分组算法的时间消耗，随着分组“桶”数量的增多，其时间消耗是线性增加的；黑色的虚线表示基于概率稀疏的监督学习模型的预测分组时间消耗，其与分组“桶”数量无关，所以基本上是一条直线。图 2 给出了待分组序列长度为 100 时的时间消耗情况，可以发现基于概率稀疏的监督学习模型的时间消耗大约是动态规划分组算法的 1/12～1/3；图 3 给出了待分组序列长度为 200 时的时间消耗情况，可以发现基于概率稀疏的监督学习模型的时间消耗大约是动态规划分组算法的 1/50～1/10。随着待分组序列的长度、分组“桶”数量增加，监督学习模型相比动态规划分组算法的加速比是逐渐增大的。因此相比动态规划分组算法，本章涉及的概率稀疏自注意力模型在预测阶段的时间消耗要更少。

通过以上对动态规划分组算法相关实验的分析可以看出，基于概率稀疏自注意力的监督学习模型能够在一定程度上学习到动态规划分组算法，且其与动态规划分组算法相比，时间消耗减少了。

结束语 本文主要对最优直方图问题的动态规划分组算法进行了分析，并在此基础上提出了基于概率稀疏自注意力的监督学习模型来学习动态规划分组算法。实验结果表明，在一定程度上，基于概率稀疏自注意力的监督学习模型能够学习到动态规划分组算法。除此之外，相比动态规划分组算法，基于概率稀疏自注意力的监督学习模型在预测阶段的时间消耗较前者更少，验证了动态规划分组算法的可学习性，同时其降低了动态规划分组算法的时间复杂度。但是，本文提出的基于概率稀疏自注意力的监督学习模型仍存在较多的方面可待改进：在模型的通用性方面，从理论与实验角度出发，分别考虑能否确立该模型学习能力的上限；在监督学习模型的可解释性方面，能否通过消融实验等手段来分析模型的各层神经元提取到的特征相对于分组算法本身的“意义”。

参 考 文 献

[1] IOANNIDIS Y. The history of histograms(abridged) [C]// Proceedings 2003 VLDB Conference. Elsevier, 2003: 19-30.

[2] IOANNIDIS Y E, POOSALA V J A S R. Balancing histogram optimality and practicality for query result size estimation [J]. ACM Sigmod Record, 1995, 24(2): 233-244.

[3] NAVAS-PALENCIA G. Optimal binning: mathe-matical programming formulation [J]. arXiv: 2001. 08025, 2020.

[4] IOANNIDIS Y E. Universality of serial histograms [C]// Proceedings of the VLDB. 1993: 256-267.

[5] JAGADISH H V, KOUDAS N, MUTHUKRISHNAN S, et al. Optimal histograms with quality guarantees [C]// Proceedings of the VLDB. 1998: 24-27.

[6] BELLMAN R. On the approximation of curves by line segments using dynamic programming [J]. Communications of the ACM, 1961, 4(6): 284.

[7] GUHA S, KOUDAS N, SHIM K. Approximation and streaming algorithms for histogram construction problems [J]. ACM Transactions on Database Systems (TODS), 2006, 31(1): 396-438.

[8] KAUSHIK R, SUCIU D. Consistent histograms in the presence of distinct value counts [J]. Proceedings of the VLDB Endowment, 2009, 2(1): 850-861.

[9] GREENWALD M, KHANNA S. Space-efficient online computation of quantile summaries [J]. ACM SIGMOD Record, 2001, 30(2): 58-66.

[10] FANG M, SHIVAKUMAR N, GARCIA-MOLINA H, et al. Computing Iceberg Queries Efficiently [C]// Proceedings of the 1998 VLDB Conference. New York: Citeseer, 1998.

[11] MIRONCHYK P, TCHISTIAKOV V. Monotone optimal binning algorithm for credit risk modeling [J/OL]. https://www.researchgate.net/profile/Viktor-Tchistiakov/publication/322520135_Monotone_optimal_binning_algorithm_for_credit_risk_modeling/links/5a5dd1a8458515c03edf9a97/Monotone-optimal-binning-algorithm-for-credit-risk-modeling.pdf.

[12] SHEVERTALOV M, STEHLE E, MANCORIDIS S. A genetic algorithm for solving the binning problem in networked applications detection [C]// Proceedings of the 2007 IEEE Congress on Evolutionary Computation. IEEE, 2007: 713-720.

[13] POOSALA V, HAAS P J, IOANNIDIS Y E, et al. Improved histograms for selectivity estimation of range predicates [J]. ACM Sigmod Record, 1996, 25(2): 294-305.

[14] IOANNIDIS Y E, KANG Y. Randomized algorithms for optimizing large join queries [J]. ACM Sigmod Record, 1990, 19(2): 312-321.

[15] GUHA S, KOUDAS N, SHIM K. Data-streamsand histograms [C]// Proceedings of the thirty-third Annual ACM Symposium on Theory of Computing. 2001: 471-475.

[16] GUHA S, INDYK P, MUTHUKRISHNAN S, et al. Histogramming data streams with fast per-item processing [C]// Proceedings of the International Colloquium on Automata, Languages, and Programming. Springer, 2002: 681-692.

[17] GUHA S, KOUDAS N. Approximating a data stream for querying and estimation: Algorithms and performance evaluation [C]// Proceedings 18th International Conference on Data Engineering. IEEE, 2002: 567-576.

[18] GUHA S, SHIM K, WOO J. REHIST: Relative error histogram construction algorithms [C]// Proceedings of the VLDB. 2004: 300-311.

[19] NOMER H A, ALNOWIBET K A, ELSAYED A, et al. Neural

knapsack：A neural network based solver for the knapsack problem[J]. IEEE Access，2020，8：224200-224210.

[20] ZAREMBA W，SUTSKEVER I. Learning to execute [J]. arXiv：14104615，2014.

[21] GRAVES A，WAYNE G，DANIHELKA I. Neural turing machines[J]. arXiv：1410. 5401，2014.

[22] CHEN Y，LI V O，CHO K，et al. A stable and effective learning strategy for trainable greedy decoding [J]. arXiv：1804. 07915，2018.

[23] KOEHN P. Pharaoh：a beam search decoder for phrase-based statistical machine translation models[C]// Proceedings of the Conference of the Association for Machine Translation in the Americas. Springer，2004：115-124.

[24] GRAVES A，WAYNE G，REYNOLDS M，et al. Hybrid computing using a neural network with dynamic external memory [J]. Nature，2016，538(7626)：471-476.

[25] ZHOU H，ZHANG S，PENG J，et al. Informer：Beyond efficient transformer for long sequence time-series forecasting [C]// Proceedings of the AAAI Conference on Artificial Intelligence. 2021：11106-11115.

[26] VASWANI A，SHAZEER N，PARMAR N，et al. Attention is all you need [J]. arXiv. 1706. 03762，2017.

[27] QIU J，MA H，LEVY O，et al. Blockwise self-attention for long document understanding [J]. arXiv：1911. 02972，2019.

[28] LI S，JIN X，XUAN Y，et al. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting [J]. arXiv：1907. 00235，2019.

[29] DAI Z，YANG Z，YANG Y，et al. Transformer-xl：Attentive language models beyond a fixed-length context [J]. arXiv：1901. 02860，2019.

[30] ELEKES Á，ENGLHARDT A，SCHÄLER M，et al. Toward meaningful notions of similarity in NLP embedding models [J]. International Journal on Digital Libraries，2020，21(2)：109-128.

[31] KE G，HE D，LIU T Y. Rethinking positional encoding in language pre-training [J]. arXiv：2006. 15595，2020.

CHEN Yunliang，born in 1979，professor，Ph. D supervisor，is a member of China Computer Federation. His main research interests include cloud computing and data mining.

HUANG Xiaohui，born in 1994，lecturer，is a member of China Computer Federation. His main research interests include geoscience data management，processing and analysis.

（责任编辑：何杨）