XGBoost - Sparsity-aware Split Finding

滕明卓

2021 年 5 月 14 日

1 背景

在处理数据时，经常会遇到稀疏数据：

1. 数据有缺失值
2. 数据包含大量的 0。
3. 数据是 one-hot 编码

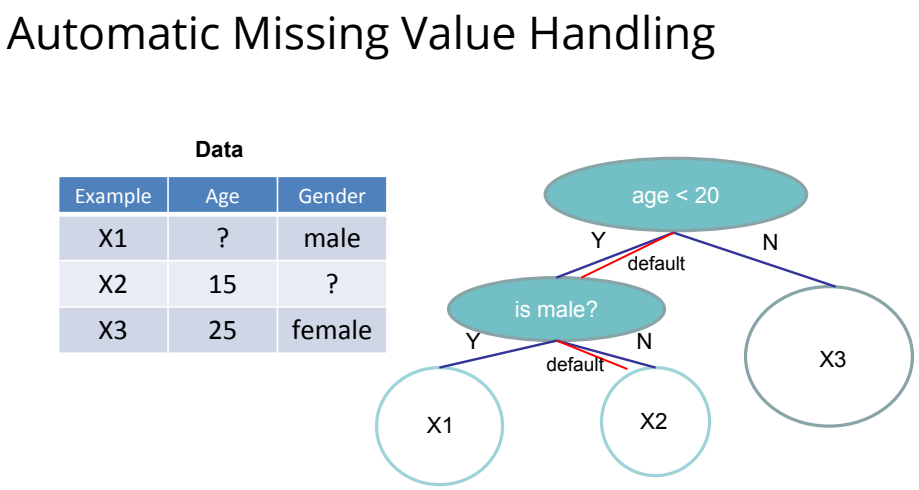
现有的树学习算法只能处理稠密数据，或者需要专门进行数据处理。

2 对于稀疏数据的 Split Finding

在原来的贪心算法中，寻找分割点需要遍历每个样本来计算分割点 (在近似的算法中就是用所有样本来计算候选分割点)。

在计算某个特征的分割点时，如果数据在这个特征上是稀疏的，那么那些具有“缺失值”的样本，都会被分到同一个分支。这样，就不需要挨个遍历样本去寻找分割点了，而只需要遍历有值的那些样本。

这些在这个特征上具有缺失值的样本会划分到“默认”的分支中。



XGBoost learns the best direction for missing values

3 算法

把在这个特征上的缺失数据 (或者是稀疏数据里值为 0 的数据)，统一划分到左子树或者右子树中，然后就只需要处理剩下的数据，来决定划分点。

Algorithm 3: Sparsity-aware Split Finding

Input: I, instance set of current node
Input: $I_k = \{i \in I | x_{ik} \neq \text{missing}\}$
Input: d, feature dimension
Also applies to the approximate setting, only collect statistics of non-missing entries into buckets
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I}, g_i, H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ ***to*** m **do**
  // enumerate missing value goto right
  $G_L \leftarrow 0,\ H_L \leftarrow 0$
  for j *in sorted*$(I_k$*, ascent order by* $\mathbf{x}_{jk})$ **do**
    $G_L \leftarrow G_L + g_j,\ H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L,\ H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
  // enumerate missing value goto left
  $G_R \leftarrow 0,\ H_R \leftarrow 0$
  for j *in sorted*$(I_k$*, descent order by* $\mathbf{x}_{jk})$ **do**
    $G_R \leftarrow G_R + g_j,\ H_R \leftarrow H_R + h_j$
    $G_L \leftarrow G - G_R,\ H_L \leftarrow H - H_R$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split and default directions with max gain