
第 1 章 数据关联的具体实现

1.1 SORT 的实现

SORT 论文中说到采用的是匈牙利算法。实际上，这是将数据关联问题转换成了一个分配问题，进而可以使用匈牙利算法进行求解。分配问题也可以用一个优化问题来进行描述：

对于 $n \times n$ 的代价矩阵 C ，需要求得最小的总代价。

$$\min \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}$$

并且满足以下约束条件：

- 每个人只能完成一个任务且必须做一个任务：

$$\sum_{i=1}^n x_{ij} = 1 \quad \forall j = 1, 2, \dots, n$$

- 每个任务只能由一个人完成且必须完成：

$$\sum_{j=1}^n x_{ij} = 1 \quad \forall i = 1, 2, \dots, n$$

- x_{ij} 是二进制变量：

$$x_{ij} \in \{0, 1\} \quad \forall i, j = 1, 2, \dots, n$$

其中 c_{ij} 表示第 i 个任务由第 j 个人完成的代价， x_{ij} 表示第 i 个任务是否由第 j 个人完成。

利用匈牙利算法可以准确得到最优解，其算法流程如 Algorithm 1 所示。

但是在实际追踪任务中，不一定出现正好 $n \times n$ 的情况，因此需要进行微调。例如出现 $n \times m$ 的情况，解决方法是扩充成 $n \times n$ ，代价矩阵直接补 0。在具体实现中，SORT 还考虑了其它内容，例如去掉未达到阈值的目标，因此其代码还是进行了许多微调，具体如 Algorithm 2 所示。

其中线性分配算法的实现则采用的是 `lap` 库的 `lap.lapjv` 函数。

Algorithm 1 匈牙利算法伪代码

输入: 检测框列表 D , 轨迹列表 T , IoU 阈值 θ

输出: 匹配列表 M , 未匹配检测框列表 U_D , 未匹配轨迹列表 U_T

1. 构建代价矩阵:

$cost_matrix \leftarrow$ 计算检测框与轨迹之间的代价矩阵

2. 矩阵预处理:

for 每一行 i **do**

$cost_matrix[i, :] \leftarrow cost_matrix[i, :] - \min(cost_matrix[i, :])$

end for

for 每一列 j **do**

$cost_matrix[:, j] \leftarrow cost_matrix[:, j] - \min(cost_matrix[:, j])$

end for

3. 寻找零元素并标记:

$marked_zeros \leftarrow$ 使用最少数量的线覆盖所有零元素

4. 调整矩阵:

while 覆盖线的数量不等于矩阵的行数或列数 **do**

调整矩阵, 重新寻找零元素并标记

找到步骤 3 中未被一行覆盖的最小元素 (称为 k)。从所有未覆盖的元素中减去 k , 然后将 k 添加到覆盖两次的元素中。

end while

5. 提取匹配:

$M \leftarrow$ 从标记的零元素中提取匹配对

$U_D \leftarrow$ 未在匹配列表中的检测框索引

$U_T \leftarrow$ 未在匹配列表中的轨迹索引

返回结果: **return** M, U_D, U_T

Algorithm 2 SORT 的数据关联伪代码

初始化:

if 追踪器列表为空 **then**

 返回空匹配列表，所有检测框作为未匹配检测框

end if

IoU 矩阵计算:

计算检测框和追踪器之间的 IoU 矩阵

匹配查找:

根据 IoU 阈值将 IoU 矩阵转换为二值矩阵

if 存在一一对应关系 **then**

 直接使用索引作为匹配

else

 使用线性分配算法找到最佳匹配

end if

未匹配识别:

通过检查哪些索引不在匹配列表中，识别未匹配的检测框和追踪器

过滤匹配:

过滤掉 IoU 低的匹配，并更新未匹配检测框和追踪器列表

返回结果:

返回匹配列表、未匹配检测框列表和未匹配追踪器列表

第 2 章 测量指标

1. 匹配单个结果

正负例 (positive/negative) 的是依据预测值，真假 (True/False) 是依据实际值。

表 2-1 基本概念

	预测正例	预测负例
实际正例	TP	FN
实际负例	FP	TN

2. 召回率

召回率（**Recall**）是指在所有实际正例中，被模型正确预测为正例的比例。其数学表达式如下：

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

第 3 章 再述数据关联

3.1 数据关联的最大问题

数据关联的最大瓶颈是在计算量爆炸的问题。

参考文献

- [1] Xin S, Zhang Z, Wang M, et al. Multi-modal 3D Human Tracking for Robots in Complex Environment with Siamese Point-Video Transformer[C]//2024 IEEE International Conference on Robotics and Automation (ICRA). 2024: 337-344.
- [2] Wu H, Li Y, Xu W, et al. Moving event detection from LiDAR point streams[J]. nature communications, 2024, 15(1): 345.
- [3] Nagy M, Khonji M, Dias J, et al. DFR-FastMOT: Detection Failure Resistant Tracker for Fast Multi-Object Tracking Based on Sensor Fusion[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). 2023: 827-833.
- [4] Zhang Y, Sun P, Jiang Y, et al. Bytetrack: Multi-object tracking by associating every detection box [C]//European conference on computer vision. 2022: 1-21.