Report

Ye Zhentao 2101111526

2022 年 3 月 5 日

1 文件

报告，论文，0.sl

2 实验描述

求解器：basicvsa

实验对象：实验 task 见 0.sl 文件。共设定了 20 个操作，1-18 为 delete，对应每一项展开式的删除；19-20 为 compose 操作，分别为组合了两个 str.++ 和三个 str.++。总计共有 2^{20} 种可产生的新 DSl，用 20 位的 01 序列描述每一个新 DSL，理论上最优解为删除其他不必需的展开式仅保留 str.++ 且不增加任何 compose 语句，即 01111111111111111100，最优解求解时间约为 0.055s。

实验流程：

1. 生成两个初始 X 值（目前为固定初始），生成对应.sl 文件，并求解对应 Y 值（由 vsa 求解器给出的求解时间，timeout=300s）分别作为 X_sample 和 Y_sample

2.fit(X_sample,Y_sample)，生成高斯模型。

3. 最小化采集函数给出下一个采样点 X_next，求解 Y_next，并将（X_next,Y_next）分别加入（X_sample,Y_sample），重复步骤 2。

3 细节解释

采集函数有多种，基本的原理都是通过当前的高斯模型，按照一定规则预测每一个 X 值的预期收益（或预期代价，一个负号的事）。不同的采集函数有不同的理论保证和偏好，目前采用了 Experted Improvement 平衡 explore 和 exploit。

标准的最小化采集函数方法即调用 minimize 算法，一般只适用于连续模型。本次实验中测试了两种最小化方式，其一为 n 步搜索，即设定一个搜索步数 n，随机一个初始点 k 次，每次从初始点出发，搜索 n 步（每步走到相邻点，即 flip 其中一位），将 k*n 里的最小值点作为 EI 推荐的下一个采样点返回。另一种方法即暴力应用 minimize 算法做连续模型求最小值，最后 round 到 01 上，比较奇怪的是，可能是因为定义域空间较大且实验设定了重复轮数为较小的，并未出现 round 到已经采样过的点的情况。（原本可能需要针对这种情况做一些调整，如张老师给的论文里有的调参数操作，但它是针对 UCB 采集函数的调整，还没看出来对于 EI 的应用方法）

4 实验结果

实验轮数为 100 时：上述两种方法最后得到的最优求解时间介于 0.055-0.06，（原来都是 0.07 但是去掉了求解器 LOG 的步骤后回到了接近最优的 0.56，0.06 左右）对应的最优 DSL 都是去掉了大部分 string 的无用展开式且不增加 compose。因为有一些实验数据是在 LOG 去掉之前得到的，完整准确的需要把所有.out 文件更新一下。

作为对比，纯随机得到的最优求解时间也可以达到 0.06。

减少实验轮数为 20 时：实验效果和随机好像还是差不多，都在 0.2-0.3s 的水平

后续又增加了两个 compose 操作为 str.replace 和 str.++ 的组合，理论上不影响最后的求解，实验结果仍然不能和纯随机拉开显著差距。

5 分析

理论上，纯随机想要得到稍好的结果，只需要不去掉 str.++，不引入影响求解的 compose 即可，因此有 1/8 的概率即可得到一个不差于初始值的解，相对来说概率是比较高的。而高斯过程模拟的由于数据点相对于定义域实在是太少的采样，可能会花不少操作去探索不太有用的地方；另一个问题是这种函数不太方便可视化看到采集函数的变化过程。总结：小样本不好超过随机...

6 讨论

1. 其他采集函数，见参考文献。
2. 工程问题：限制深度
3. 多 task，更复杂的操作（目前是提前硬编码的）