

机器学习

Traveling Santa 2018 最优路径算法

主 研 人：董乐诗

参 研 人：董乐诗

审 核 人：

方 向：工业算法案例研究

版 本 号：A

声明：本作品权益属中冶赛迪。所含信息、专有技术应予保密。未经本公司书面许可，不得修改、复制、提供或泄露给任何第三方。

CLAIM: This work belongs to CISDI, MCC. All information and know-how shall not be copied, duplicated, altered, submitted and disclosed to any third party without the prior written permission of CISDI.

大数据及人工智能部®

中冶赛迪信息技术有限公司

二〇一八年九月

版本更新

日期	版本	更新描述	作者
2019/3/26	A	初稿	董乐诗

选题表格

时间	竞赛名	竞赛背景描述 (50 字以内)	类型 (分类/回归)
2019/3/18	Traveling Santa 2018 - prime paths (2018 圣诞老人最优路径算法)	帮助鲁道夫寻找圣诞送礼的最佳路径	最优问题 (旅行商问题)

目录

1. 背景描述.....	4
1.1 竞赛赛题描述.....	4
1.2 评估指标描述.....	4
2. 数据来源及描述性统计分析.....	5
2.1 大赛数据来源.....	5
2.2 数据的描述性统计.....	5
2.2.1 数据基本情况描述:	5
2.2.2 数据字段介绍:	5
2.2.3 数据描述性统计.....	6
3. 优秀算法思路.....	6
3.1 方案一.....	6
3.1.1 方案一数据预处理及特征工程部分方案.....	7
3.1.2 方案一模型设计、建立部分方案.....	7
3.1.3 方案一结果、排名等.....	8
3.1.4 方案一算法流程图.....	8
3.2 方案二.....	9
3.2.1 方案二数据预处理及特征工程部分方案.....	9
3.2.2 方案二模型设计、建立部分方案.....	9
3.2.3 方案二结果、排名等.....	10
3.2.4 方案二算法流程图.....	10
4. 算法比较.....	11
表 4-1 算法比较.....	11
5. 总结与展望.....	11
5.1 总结.....	11
5.2 建模思路.....	11

1. 背景描述

Kaggle 每年都会在圣诞节来临的时候出一道专属于圣诞节的赛题，随着 2018 年圣诞节的到来，新一届的优化比赛也缓缓拉开了序幕。

圣诞老人有一只红鼻子的驯鹿鲁道夫，因为有一只红鼻子所以它经常被嘲笑，但是在一个大雾的夜晚，鲁道夫凭借它的红鼻子最快完成了送礼的任务。可见每件事情都是好坏兼具的。鲁道夫一直信奉要聪明的工作而不是只努力的工作，所以它想到通过算法来明显改善圣诞老人在圣诞节前夕赠送玩具的路线。

1.1 竞赛赛题描述

黄金城市（CityId 为素数）的房屋总是为驯鹿留下胡萝卜和普通的饼干和牛奶。这些胡萝卜是驯鹿保持步伐的寄托。实际上，如果驯鹿队不是每 10 步就从一个黄金城市出发，那么他们到达下一个城市的距离会比往常长 10%。

本次比赛的挑战是，构建一种算法以计算出圣诞老人送礼的最佳路线。

背景补充：质数又称素数。一个大于 1 的自然数，除了 1 和它自身外，不能整除其他自然数的数叫做质数；否则称为合数。即当你从第 19 个城市出发到第 20 个城市，如果第 19 个城市的 CityId 是质数， $cost = distance(CityId19, CityId20)$ ，否则 $cost = 1.1 * distance(CityId19, CityId20)$

1.2 评估指标描述

本次比赛的提交的路径将由欧氏距离来评测，但是如果驯鹿队不是每 10 步就从一个黄金城市出发，那么他们到达下一个城市的距离会比往常长 10%。欧氏距离计算公式如下所示：

$$cost = \sum_{i=1}^N distance(CityId_i, CityId_{i+1})$$

提交文件应该包含圣诞老人访问所有城市的有序路径。路径必须在北极开始和结束（CityId = 0），且必须访问每个城市恰巧一次。提交的文件必须具有标题，并且应如下所示：

```
Path
```

```
0
```

```
1
```

```
2
```

```
...
```

```
0
```

2. 数据来源及描述性统计分析

2.1 大赛数据来源

数据来源：本次大赛的数据来源于 kaggle 网站

本次比赛官方提供的数据如下：

[traveling-santa-2018-prime-paths](#)

官方数据下载地址如下：

<https://www.kaggle.com/c/traveling-santa-2018-prime-paths/data>

2.2 数据的描述性统计

2.2.1 数据基本情况描述：

- cities.csv – 城市及其坐标列表

参赛者需要创建访问所有城市的最短路径。提交的文件只是访问每个城市的有序列表。路径应该具有以下约束：

- 路径必须在北极开始和结束（CityId = 0）
- 必须访问每个城市恰巧一次
- 两条路径之间的距离是 2D 欧几里德距离
- 除非来自一个主要的 CityId，否则每 10 步（ $\text{stepNumber} \% 10 = 0$ ）的长度将

增加 10%。

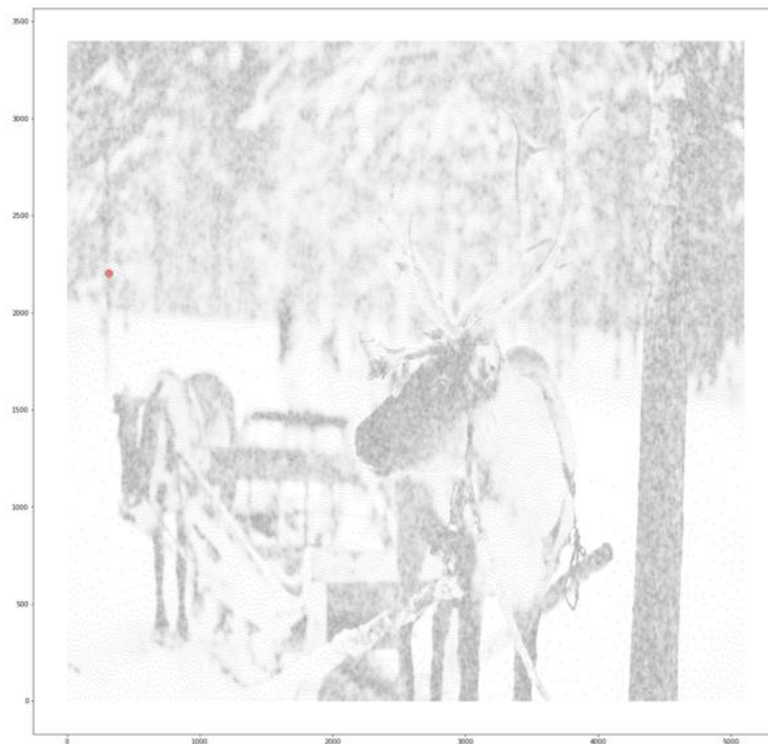
2.2.2 数据字段介绍

数据字段解释如下：

- CityId: 城市的编号
- (X,Y): 城市的横纵坐标

2.2.3 数据描述性统计

如果用 plot 命令把每个城市的位置都画出来，可以观察到一个有趣的现象，如下图所示：



所有城市位置的分布近似于上图的这个麋鹿的图片，途中的红点是北极（CityId = 0）的位置，也就意味着最终的结果应该是始于红点且终止于红点。

由于本题是一个单纯的路径优化问题，所以只给出了所有城市的坐标数据。

3. 优秀算法思路

3.1 方案一

3.1.1 方案一数据预处理及特征工程部分方案

由于本次比赛是一个单纯的 TSP 旅行商问题，所以数据没有复杂的特征，故无需数据预处理和特征工程。对于 TSP 旅行商问题，详情请参照以下链接：

https://blog.csdn.net/houchaoqun_xmu/article/details/54584264

3.1.2 方案一模型设计、建立部分方案

第三名得主在比赛开始的初期使用的是 `concorde`，比赛进行一段时间之后换用了 LKH 算法。建模语言的选用上从初期的 Java 转向了后期的 C 语言来运用 Lin-Kernighan 算法。

简要综述以下整个建模历程的话，分为以下五部分：

- 用 LKH 模型获得简单直接的 TSP 问题结果，得分为 1516321
- 运用惩罚意识下降最快的 Lin-Kernighan 算法，得分为 1514683
- 尝试应用更高的 `opt (k-opts)`，得分为 1514610
- Kick tour,重新优化，得分为 1514245
- Kick tour，用三个技巧重新优化，并与 GPX2 算法重新组合，得分为 1514014

第一阶段，没有过多的处理，直接使用的利用 GPX2 的 LKH 算法。

第二阶段，参赛者运用了 Lin-Kernighan，有效实施了 Lin-Kernighan TSP 启发式的 K-opt 操作。为了快速计算增益，参赛者使用了所有可能的 MOD 10 的正向和反向的累加和。参赛者没有使用单独的参数来限制循环次数和连接它们的交替循环次数。相反，他将全部参数合并到了参数 `maxK` 中，并通过加入最小周期大小（`CycleLen`）的限制来降低复杂性。参赛者首先搜索了整个 `base_opt` 空间，然后选择起始城市，在这些城市找到了改进，并从更高的选项开始搜索。此外，他逐渐将 `CycleLen` 从迭代过程中减少，使参赛者能够进行更高的基础选择。第三阶段，这一

阶段的参赛者努力提高了 **opt**，减少了计算过程中的候选点集，即从五个候选点减少到了三个候选点。

第四和第五阶段，对于 **kicks** 部分，参赛者做了一些 **k-opts** 来提高运算成绩。并且使用了 **GPX2** 进行重组，最终的得分是 1514014.62。

该组参赛选手的代码分享到了 GitHub 上，详情请参阅：
<https://github.com/NighTurs/kaggle-traveling-santa-2018-prime-paths>

3.1.3 方案一结果、排名等

结果：得分 1514014.62

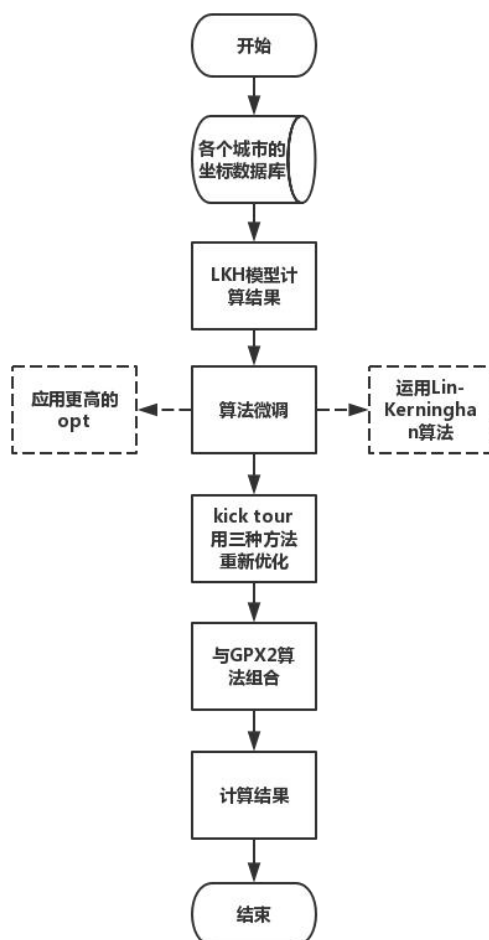
排名：最终排名 top3

详情见下表 3-1：

3	NighTurs	1514014.62...	13	3mo
---	----------	---------------	----	-----

表 3-1 第三名队伍比赛结果

3.1.4 方案一算法流程图



3.2 方案二

由于旅行商问题的解题策略较为单一，所以本次比赛的参赛选手的解题方式并无较大差异，只有在微调部分和实操部分的使用的工具的区别，第八名的选手方案最大的突破点是 python 中 `cumsum` 函数的应用。具体的解题方案如下所示：

3.2.1 方案二数据预处理及特征工程部分方案

本次比赛不涉及特征的处理，故本部分不做过多阐述。

3.2.2 方案二模型设计、建立部分方案

参赛者的解题方案如下所示：

- 1) 参赛者首先使用 LKH 获得了初始结果。
- 2) 模型微调部分，尝试了 2.5-opt 和 3-opt（其中 3-opt 没有翻转段）
- 3) 使用双桥（Double bridge）处理（一种 4-opt 非顺序移动，没有翻转段）。
- 4) 尝试逐渐增加惩罚的技巧（与第二名相同），所得计算结果又进一步提升。
- 5) 使用了用于处理主要评分功能的 CUMSUM 函数（python 中的）。这个函数使得该组参赛者不需要任何 GPU 和任何硬件包就完成了提高计算速度的功能。
- 6) kick 的使用，来避免 local min(局域最小值)。通过各种类型的动作（从 5-opt 到 2.5-opt）kick out 片段，然后通过各种参数快速重新优化所有类型的动作，分数进一步提升。通过 LK 启发式编码 5-opt（包括顺序和非顺序）移动，并使用 `cumsum` 函数运行。
- 7) 当踢 k-opt 移动时，我们尝试使用 EAX 算法（结合 2 个不同的方式）。
- 8) 使用下限估计在最多 200 个城市的段上将城市们重新排列。分数又进一步提升
- 9) 最后第八名的四位参赛选手通过使用共享的 Dropbox 文件夹进行了最后的路径探索，每当谁发现新的路径时，就立即放到该共享文件夹中，其他成员自动获得该路径并继续尝试。这个方法非常有效。

在整个优化过程中，cumsum 函数扮演了重要角色。

该组参赛选手将代码放到了 GitHub 上，详情请参阅读：

<https://github.com/voanhkha/Traveling-Santa-2018-Kaggle>

3.2.3 方案二结果、排名等

结果：得分 1514438

排名：最终排名 top8

详情见下表 3-2：


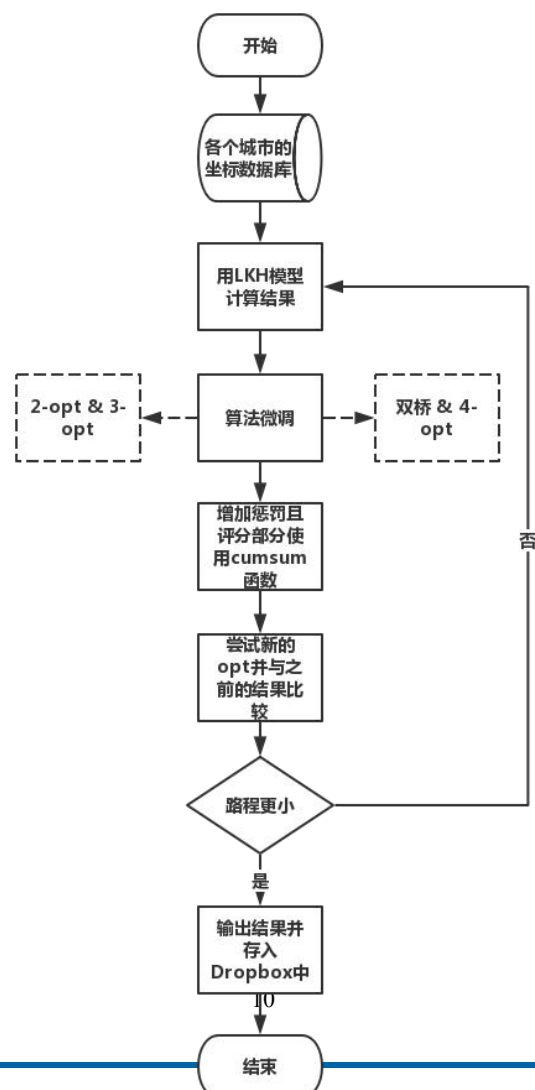
8	Zidmie Kha Marc Simon		1514438.00...	142	3mo
---	-----------------------------	--	---------------	-----	-----

表 3-2 第八名队伍比赛结果

3.2.4 方案二算法流程图



4. 算法比较

由于旅行商问题解决方案的单一性，两组参赛选手选择了相同的算法 LKH，方案的区别在于模型微调的部分。第二个共同点是两组选手都发现了增加惩罚可以得到更好的结果。两种方法都是很优秀的算法方案。

表 4-1 算法比较

	评估指标数值	特征工程	基础算法	基本库	语言
算法一	1514014.62	无	LKH 模型	stdlib.h、stdio.h、math.h、stdafx.h、time.h 等	C 语言
算法二	1514438	无	LKH 模型	numpy 等	Python

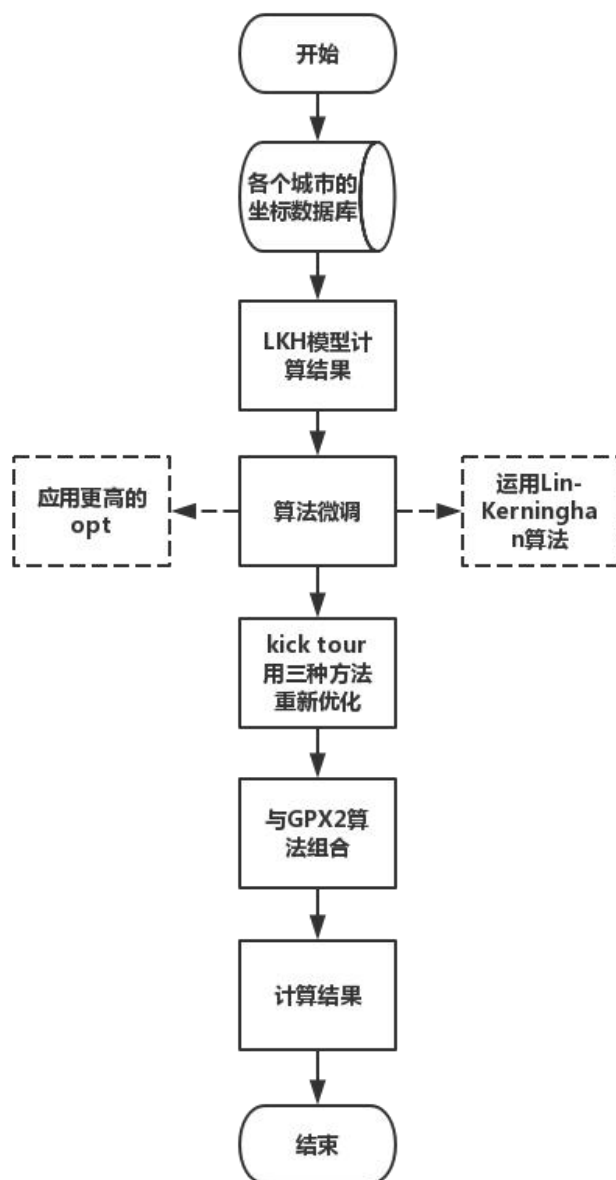
5. 总结与展望

5.1 总结

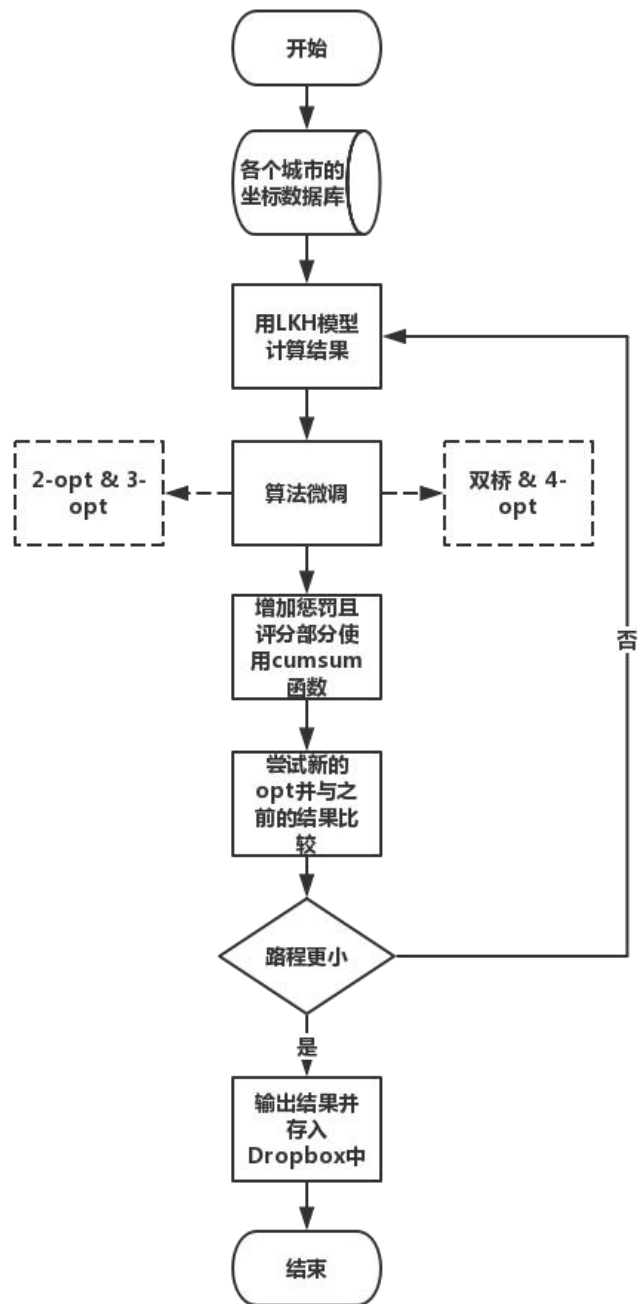
本次比赛是 kaggle 一年一度的圣诞节赛题，所以不论是从赛题的性质还是从解决方案上都和 kaggle 大多数的比赛不太一样，就本题而言，与其说这个比赛是一个机器学习的问题，不如说这个比赛是一个单纯的旅行商最短路径优化问题。旅行商问题是 NP 难问题，这类问题的解决方案也较为单一，就本次打参赛者们而言，最多使用的就是 LKH 模型和 `concorde` 算法。能拉开差距的主要策略是不同参赛选手在模型微调部分所作的操作和计算效率的问题。第八名（本文档方案二）的团队在提高计算效率上运用了很好的方法（如 `comsum` 函数和 `Dropbox` 的运用），值得参考。

5.2 建模思路

方案一参赛选手解题思路的流程图如下所示：



方案二参赛选手解题思路的流程图如下所示：



我的想法：和大多数选手的思路一样，我会以 LKH 算法为基础，尝试多种 opt，并且学习第八组参赛选手在评分部分运用 comsum 函数来提高计算效率。