

项目汇报书

<目的>

本项目的目标为对现有物流在最短时间内配送路径的优化和分析、配送员组合分析和基于配送员经验构建路径预测模型。

<方法>

一. 最短时间内配送路径的优化和分析

1.1 基于直线距离的最短路径计算

使用算法: Branch and Cut, 遗传算法(GA), TSP_solver库

1.2 配送员人数削减

使用算法: Branch and Cut

1.3 基于实际道路距离的最短路径计算

使用算法: 遗传算法

1.4 考虑定时配送的最短路径计算

使用算法: 遗传算法+集成学习

二. 基于老手配送员经验的路径实时预测

使用算法: 卷积神经网络(U-Net)

1.1 基于直线距离的最短路径计算

<概要>

作为课题A (最短配送顺序计算) 的方法, 考虑考虑基于直线距离的巡回配送总长的计算. 基于配送数据的经纬度, 通过Branch and Cut, 遗传算法和TSP_solver库三种方式进行分析, 并进行最短配送顺序的计算.

<分析方法>

将配送场所的经纬度坐标标记点,在不考虑实际道路的情况下,进行点与点的直线距离计算.目标为所有配送点间距的总长最短.

在分析中, 使用了三种不同的算法. 首先是可以计算出一定顶点数(配送场所)的解析解的Branch and Cut算法, 分别进行了1分钟和10分钟的结果分析. 再者是通过遗传算法来求解的近似解,可以发现在遗传算法能在保持一定精度的情况下,快速找到全局最优解,相比Branch and Cut算法在计算时长有着很大的缩减.

<分析结果>

距离的平均减少量,平均计算时间如下所示.

Branch and Cut (计算上限1分钟)	2531 m	4.86秒
Branch and Cut (计算上限10分钟)	2628 m	31.64秒
遗传算法	2628 m	5.56秒
TSP_solver (一种快速计算TSP问题的求解方法)	2269 m	0.02秒

在全部配送路径的距离减少量上,可以发现一些路径有着大幅减少的情况. 优化后路径的平均距离减少量最大的是配送场所为27处,司机ID为222911的路径,大致削减量为20000m. 在其他配送路径上, 配送场所为29处的大致减少了5000m, 并且在少量配送点个数的路径上也有大幅削减的情况发生.

<考察>

在多数情况下可以发现,投递地点越多,有路径优化余地的可能性就越高。另外,在有着大幅削减距离情况的案例中可以发现,其特点为各个配送点之间的间距都很大,因此可能造成配送员的优先配送的判断失误。

1.2 配送员人数的削减

<概要>

除了计算最短路径,同时也考察了配送人数削减的可能性。基于西农运输大垣分店的配送数据,经过分析,可以发现存在司机人数的优化。在此基础上,可以通过减少配送人员的数量来进一步提高人力和物流资源的高效利用。

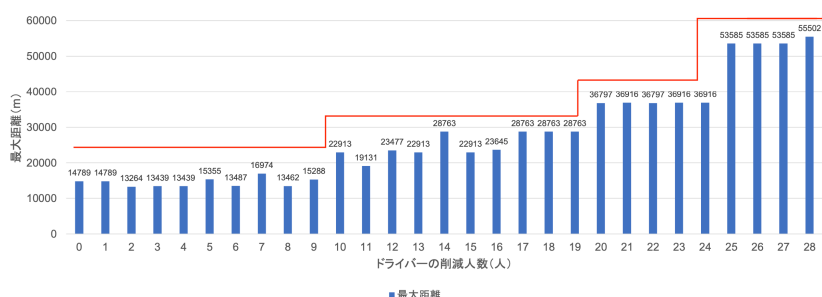
<分析方法>

首先,从各个单日配送数据中,抽取各个司机ID的配送场所,每个ID下的配送场所作为一个簇,单日配送的人员数作为初始设定的聚类总簇数。并利用k-means法依次生成不同总簇下的分布情况,然后并进行最优化路径处理。例如在对2020年6月11日的单日配送分析中,利用k-means一次对总簇数为38到10的分布情况进行了分析,针对每一簇,利用Branch and Cut和tsp_solver求解路径最优解。在处理过程中,优先使用Branch and Cut算法求解,并在针对1分钟内Branch and Cut算法无法进行求解的情况,使用tsp_solver模块进行计算。最终分析考察了削减人数从0到28人对总配送距离带来的影响。

<分析结果>

通过对单位人数下的平均配送距离的计算,削减人数从0人到28人所得出的结果为,随着削减人数的增加,平均配送距离也跟着从5336m/人增加到33248m/s。这表明了削减人数会增加每个人的配送负担。

接着考察了对单人最大配送距离的影响,通过对比发现,在一定的削减范围内,最大配送距离并不一定是随着削减人数增加而成比例的增加的,而是成阶段性的阶梯状增加。例如下图所示,在对2020年6月11日的分析中,最大削减人数在削减人数为0-9人的情况下并没有太大的变化,而在削减人数为10人后有了比较明显的增加,并且在之后依然呈现阶梯式的增加,表明了如果从最大配送人数上进行考量,当前配送人数有着一定的削减空间。总和比较全年平均可削减情况,人员削减在7-8名为最佳情况。



<考察>

在实际条件下,需要考虑运输载荷量以及卸货时长,而在本次分析中并没有考察到这一点带来的影响,但在实际人员调配上,如果单日配送总量(3-4t)情况下,可以确定的是,配送人员具有一定的削减空间,通过削减人员来提高人员利用率在实际情况下也是一种可行的提案。

1.3 基于道路距离的最短路径计算

<概要>

在实际配送中,真实配送距离往往依托于道路信息,更多是基于曼哈顿距离而不是欧几里得距离。因此在这里基于道路距离考察了实际距离下的最短路径优化情况。

<分析方法>

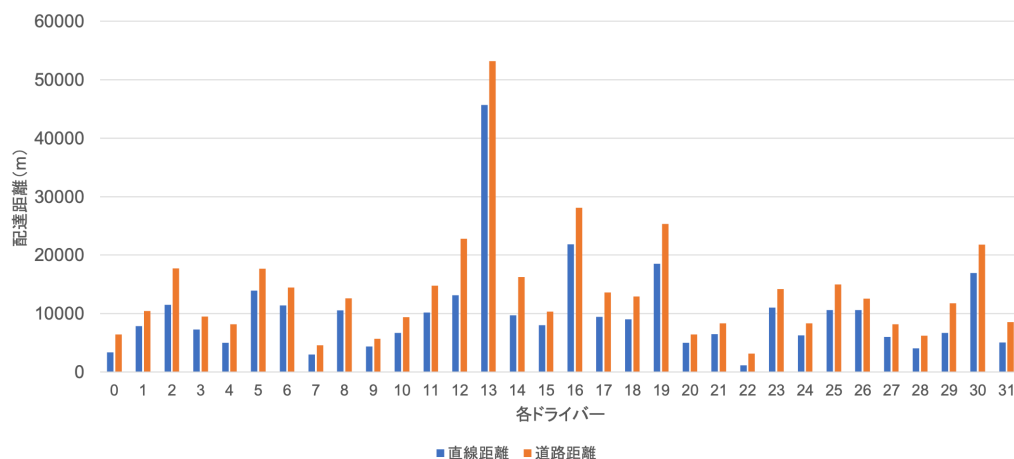
针对1.1中配送数据的坐标信息,通过Google Map API进行实际道路距离的转换后,利用遗传算法进行了考察。

<分析结果>

与1.1中的结果相比,基于实际道路信息计算得到的平均配送距离大约为直线距离的1.37倍。

<考察>

因此,如果想要不依托于Google Map API而对道路距离进行估算,可以通过将直线距离进行1.3~1.5倍扩充后进行近似计算.



1.4 考虑定时配送的最短路径计算

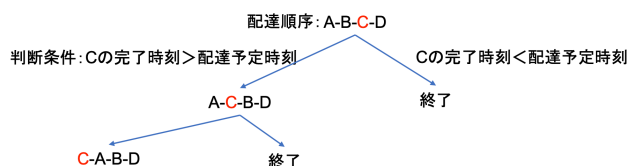
<概要>

在1.1中的计算中,所做的优化处理是基于位置坐标的距离计算,但如果考虑规定配送时间的情况,需要对最短路径进行了基于时间考量的转换.并且在实际配送过程中,两处配送点之间的耗时往往需要拆解成行程时间和卸货时间,但在提供的数据中仅仅只各个配送地点的配送完成时间.考虑到以上的问题,采用中间条件处理较为灵活的遗传算法进行最优解的求解,在配送时间计算上,使用集成学习对过去一年内的配送情况进行学习并实现耗时预测.

<分析方法>

首先基于特征工程,对原始数据进行再加工,对时间信息进行加工中,基于爬取的过去一年的天气信息,增加了天气特征,并将时间特征拆解成时间段特征(早晨,上午,中午,下午,傍晚),休息日特征(是否周末,是否节假日),季节特征,并基于simple-geocoding库将从经纬度信息中提取了配送地点的地址信息(X市X区X町),并结合当前配送载货量和配送行程距离,将过去一年的信息利用Xgboost进行了时间序列的预测.在最短耗时的优化过程中,将训练完的XGboost回归模型融入到遗传算法中的适应度评价过程.

- **算法:** 基于XGboost回归模型的自适应遗传算法
- **训练features:** 时间段,是否休日,季节,地址信息,当前物资载货量,配送行程距离
- **训练label:** 上一次配送完成时间到本次配送完成的耗时时间
- **遗传算法改进:** 自适应交叉算子,自适应变异算子,基于配送准时判断条件的配送顺序调节(如下图),基于XGboost回归的预计配送时间预测模型



(评价方法)

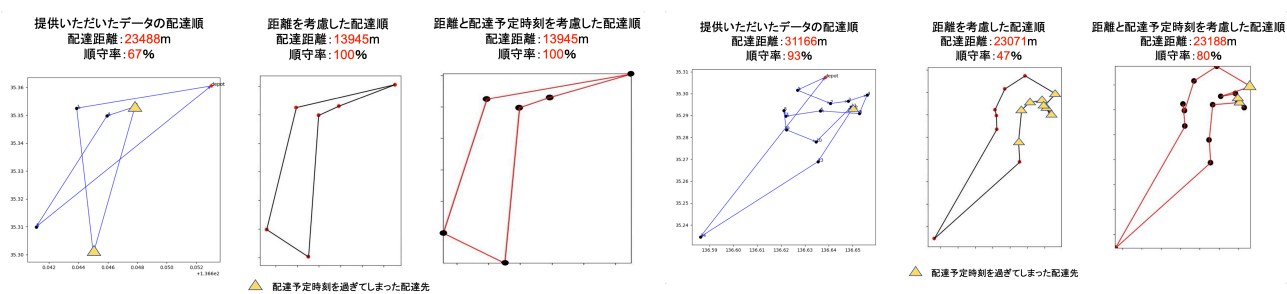
1. 配送点A,B,C,D四处中,当判断C处是否守时.
2. 如果守时,进入D的判断,如果不守时,将C提前配送,并再次更新预计配送时间,再次判断C是否守时,如果守时,返回1,否则重复提前操作
3. 如果所有的都更新完了,仍然存在无法守时配送的个体,则计算守时率,并再次投入遗传算法的迭代中.

<分析结果>

	提供数据	仅考虑距离	考虑距离和配送预定时间
平均配送距离 m/人	43718 m	41089 m	41734 m
配送预定时间的守时率	83%	73%	82%

本次分析的目标为在考虑到现实条件后,对最短路径进行最优化求解. 上表表明,与仅考虑距离的优化路径相

比,在考虑到配送预定路径的条件下,平均配送距离仅仅增加700m.但考虑到守时率的问题,最优解仅仅只有73%,相比原路径,虽然配送距离下降了2629m,但守时率却下降了10%,而优化后的最短路径守时率仅下降了1%,但人均配送距离能下降2000m左右,在综合考虑守时率和平均配送距离的情况下,考虑配送预定时间的算法得出的结果最为理想.



左图显示了通过优化配送路径后,配送路径大幅减少,并且守时率也大大增加的情况,而右图显示了虽然配送平均距离减少了,但守时率却下降了的个例,对比两种优化算法,可以发现,考虑到配送预定时间的配送顺序在路径优化和守时率保持上都或有考量.而在考察全部情况时,发现配送地点超过10处的情况下,守时率相比原提供数据呈现下降的趋势,而在10处以下的情况下,守时率则有所提高.

<考察>

在考虑到现实条件(配送预定时间的守时率)后,距离和守时率达成了一定程度上的平衡,但当配送地点数太多情况下,守时率相比原数据有很大程度的下降,同时观察到原数据下的守时率也只有83%,在某种程度上也说明了,如果要进一步提高服务质量,需要对配送出发时间进行提前调整,再优化路径来减少配送员的工作量.

2. 基于老手配送员经验的路径实时预测

<概要>

在实际中,即使是计算出来的理论最优路径也不代表其路径最合适,还需要考虑到道路交通是否畅通,同时是否存在大坡度的上下坡,以及道路质量是否适合货车行驶等等.但一般经验丰富的配送员在同一区域长期配送后会总结出相对不错的路径,这对于即使是新手配送员或者不熟悉该区域的人来说也可以按照老手配送员的路径去完成配送任务.因此,为了学习到这些经验丰富的配送员的配送路线,可以依托于深度学习的方法来实现对经验路线的学习.在此,我们利用一种名为U-Net的CNN模型来实现对TSP问题的求解.该网络的特点在于可以通过对少量图片的学习实现端对端的训练,现有的论文已经表明了其对TSP问题的求解中有着优秀的表现.在本模型中,将TSP问题求解的场所和路径用点和线在图像上标出,利用图像识别的CNN模型构架进行学习.在训练集中,我们抽取了配送经验超过两年的配送员的配送路径作为监督,并剔除了其中明显表现不是太好的路径.

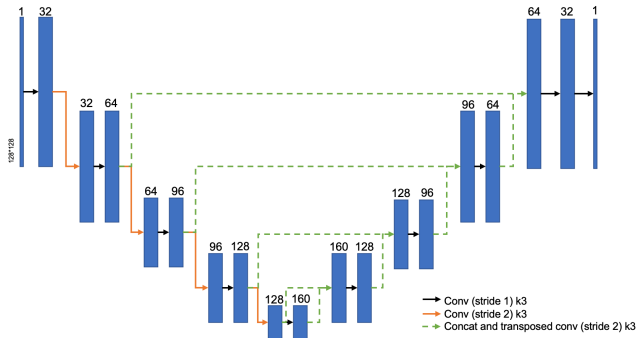
<分析方法>

该CNN模型分为学习和预测两阶段,学习方法和路径计算如下图所示

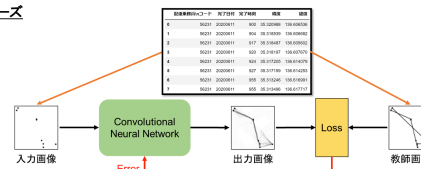
• 学习阶段

其目标为让CNN模型学习到配送路径的模式.在这里,输入图像中用黑色点标记该次配送的所有地点坐标,监督图像为老手配送员的实际配送路径.图像的像素尺寸为128*128. 基于U-net的网络构架如右图所示,在模型构建上,整个网络分为5个尺度,在每个尺度上都采用步长为1的卷积核进行处理,并利用步长为2的卷积核来进行尺度上的过渡,这是与U-net不同的点.在每个尺度上进行一次same padding后将其与转置卷积后的结果进行连接,并输出到上一层,最终实现端到端的训练.损失函数选择上利用总像素点的二乘和作为损失函数,以用来计算像素的相似度

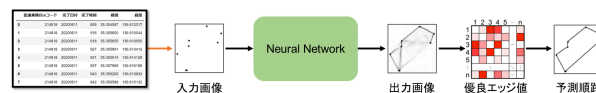
• 预测阶段



1. 学習フェーズ



2. 予測フェーズ



在完成训练后模型上,求解配送路线首先将配送地点的坐标等比例转换到图像上,并经过端到端的学习后输出预测图像.并通过以下的公式进行优良边的判断.在图像中,任意一个点 $(x, y) \in \{1, 2, \dots, 128\}$. $p(x, y)$ 为输出图像, $l_{ij}(x, y)$ 为只绘制Edge(i, j)的图像.通过下式计算通过Edge(i, j)的似然 v_{ij} .在直观上, v_{ij} 描述了点 i 到点 j 之间的边的浓度,边越浓代表Edge(i, j)被选中的概率越高.最后通过计算巡回路径下所有边的总似然值 $\sum v_{ij}$ 最大,则输出该路径.

$$v_{i,j} = \frac{\sum_{x=1}^{128} \sum_{y=1}^{128} p(x, y) l_{ij}(x, y)}{\sum_{x=1}^{128} \sum_{y=1}^{128} l_{ij}(x, y)}$$

<分析结果和考察>

下图和表显示了原配送路线和预测路线.

从表中可以看到所有路径的平均评价值为 0.432 (Levenshtein距离),该评价指数越接近0表示与原路径越接近,其评价的计算方式如右图所示,预测路径经过的置换/插入等操作数与路径字符串长度比值即为评价指数.通过该评价指数,可以直观的对路径的相似度进行评价.

从表中可以发现,随着配送地点的增加,预测路径与原始路径的差异性也在逐渐增大,表明了模型在配送地点很多的情况下存在过拟合的现象.从数据本身来看,其原因主要为原始配送路线中,即使是经验丰富的司机也存在对同一配送地点采用不同路线.从而导致模型学习无法准备进行判断.其二为模型本身较为复杂,并且存在许多地点的配送次数过少,而当配送场所很多的情况下,由于本身路线组合的复杂度大幅上升,使得模型在没有大量数据的学习的情况下难以很好的学习.

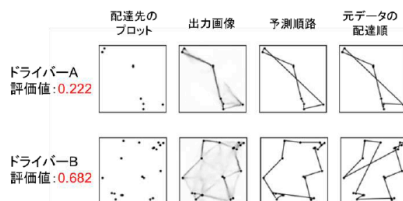
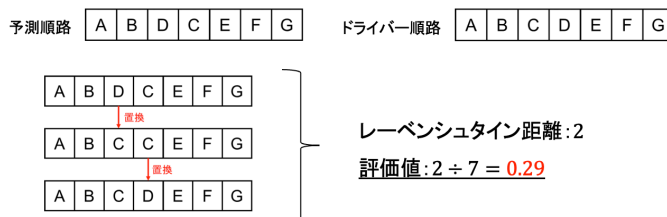


図 2. 元データの配達順路と予測順路の例

配達先の数	評価値の平均
All	0.432
~10	0.322
11~20	0.523
21~	0.631

表 1. 配達数による評価値の平均

3. 总结

在本次项目中,发现了现有路径的不足,并提出了基于遗传算法的路径优化方法,同时基于老手司机的经验配送路线,构建了一种基于CNN的模型来学习其经验,该模型在经过训练后可以快速获取较好的配送路径.