华南理工大学

South China University of Technology

本科毕业设计（论文）

基于闪电数据的雷暴路径预测算法设计与实现

学　　院	软件学院
专　　业	软件工程
学生姓名	石望华
学生学号	201630676843
指导教师	汤德佑
提交日期	2020 年 5 月 20 日

摘　要

雷暴是一种灾害性天气，容易引起火灾、山洪和泥石流，破坏房屋建筑、用电器和农作物，危及行人生命等灾害。如果能准确定位、跟踪雷暴发生地点，进而预测雷暴路径，提前做好防范措施，就能避免重大灾难的发生，大大减少经济损失。

本文主要工作如下：基于广州气象局提供的闪电数据，实现了单场和多场雷暴识别算法，定义了样本数据集、答案数据集的构造方法，在参数设置合理的条件下可以比较准确地识别、提取出属于同一场雷暴的闪电数据并分线圈法和中心路径法两种方式进行雷暴可视化；基于多项式回归，以多场雷暴识别算法构造的样本数据集作为输入，实现了路径预测算法，给出了预测效果评估与参数设置方法，定义了预测集与结果集的构造方法；最后设计并完成了雷暴识别实验、路径预测实验，对三个算法进行了可行性验证，路径预测实验的评估结果如下：在参数（共 10 个）设置合理的情况下在单个数据集上以 72%的准确率实现了对每个雷暴样本未来 20.25 分钟的路径预测，漏报率为 0%，误报率为 21%；在整个数据集上以 64%的准确率实现了对每个雷暴样本未来 15.47 分钟的路径预测，漏报率为 0%，误报率为 32%。

关键词：雷暴识别；雷暴可视化；雷暴路径预测；数据集构造

Abstract

The thunderstorm is a kind of disastrous weather, which is easy to cause fire, mountain flood and debris flow, damage buildings, electrical appliances and crops, and endanger the lives of pedestrians. If we can locate and track the location of thunderstorm accurately, then predict the thunderstorm path, and take preventive measures in advance, the occurrence of major disasters can be avoided and economic losses can be greatly reduced.

The primary work of this thesis is as follows: The algorithm of single field and multi field thunderstorm recognition is realized based on the lightning data provided by Guangzhou Meteorological Bureau, and the construction method of sample data set and answer data set is defined; Under the condition of reasonable parameter setting, the lightning data belonging to the same thunderstorm can be accurately identified and extracted, and thunderstorm visualization is implemented in two ways: coil method and center path method; Based on polynomial regression, the path prediction algorithm is implemented by taking the sample data set constructed by the multi field thunderstorm recognition algorithm as input; The prediction effect evaluation method and parameter setting method are designed, and the construction method of prediction data set and result data set is defined; Finally, the thunderstorm identification experiment and path prediction experiment are designed and completed, and the feasibility of the three algorithms is verified. The evaluation result of path prediction experiment under the condition that the parameters (10 in total) are set reasonably is as follows: the path prediction of each thunderstorm sample for 20.25 minutes in the future is achieved with a 72% accuracy rate, a 0% miss forecast rate and a 21% wrong forecast rate on a single data set; the path prediction of each thunderstorm sample for the next 15.47 minutes is achieved with a 64% accuracy rate, a 0% miss forecast rate and a 32% wrong forecast rate on the entire data set.

Keywords: Thunderstorm recognition; Thunderstorm visualization; Thunderstorm path prediction; Data set construction

目　录

第一章 绪论

1.1 研究背景与意义

雷电定位监测技术最早于 20 世纪 70 年代末提出并实现，已经了经过近半个世纪的发展，现在已有至少 40 个国家建立了比较完备的雷电定位系统并将之应用于电力、通信、航天航空、石油石化、防灾减灾等领域。

雷电定位系统（Lightning Location Systems, LLS）是近年来在气象领域应用最广的雷电监测技术之一[1]，是一套覆盖大面积、全自动、高精度的实时监测系统，能实时地检测出雷击发生的经纬度、时间、极性、峰值、运动轨迹和次数等各种雷电参数，同时还能实时共享各个地区的雷电信息。

我国在雷电定位监测技术领域及其系统的自主研发方面积累的数据资料多、起步早、监测区域大。1988 年左右，我国就开始开展监测雷暴的工作。1993 年，第一套国产雷电定位系统在安徽电网投入使用。进入 21 世纪以来，我国电网实施了 LLS 全国性的联网工程，雷电定位系统发展迅速。2006 年，我国建成了覆盖全国电网和绝大部分国土的雷电监测网络。2013 年，南方电网和国家电网的数据共享工程竣工，标志着我国雷电定位系统实现了全国联网[2]，这为雷电定位技术在电力系统的雷电预警方面的应用提供了坚实的硬件基础。

闪电是一种在一定区域内比如云与云（云闪）、云与地（地闪）之间间断性发生强烈放电的物理现象。一道闪电长度可达几百米至数千米，通常只维持数秒，但携带巨大能量的爆发，温度达上万摄氏度，一旦击中人或物都易形成毁灭性打击。据统计，华东和华南的一些近海城市雷暴事故频繁发生，大约占了我国雷害总事故次数的 42%[2]。这些雷暴事故具有覆盖面积大、影响对象多的特点，影响对象中尤其对高压输电的破坏很大。我国高压输电线路由于雷暴引起的跳闸次数占总跳闸次数的比例最高可达 70%[2]，这可能造成大面积停电事故，对人们的日常生活、企业的正常运转带来极大的不方便。此外，雷暴还可能干扰飞机航行，影响重大活动如国庆阅兵和奥运会赛事的举行，摧毁建筑并破坏各种用电设备，危及行人的生命安全、引发火灾、山洪和龙卷风等灾害。如果能够准确地预报雷暴发生的地理位置、强度与走势，就能避免这些重大灾难的发生。这不仅能减少经济损失，还可以减少人员伤亡。

1.2 研究现状

为了准确预测雷暴天气具体发生的地点、强度，避免雷暴灾难的发生，雷暴路径预测算法成为雷电定位系统的核心技术之一。随着计算机性能的提高，该研究领域衍生了很多识别并定位雷暴的算法。

针对目前雷暴预测不能对某个地区在接下来几分钟内雷暴的发生范围、闪电密度和轨迹做出精准预测的问题，文献[3]继承传统的基于密度的聚类算法（Density Based Spatial Clustering of Applications with Noise, DBSCAN）,用核密度估计原理确定聚类参数，基于时间因素加权的欧式距离对实时收集的云地闪数据进行聚类分析。然后根据克里金插值法计算落类密度，考虑了雷云随时间移动导致雷点移动的影响，最终将未来 10 分钟内雷暴中心点的预测值与实际值的距离控制在了 3km 内，闪电次数预测的准确率达 80%之上，提高了雷电临近预报的准确度和数据聚类定位雷电的适用性。但是该文主要是预测落地点与次数，并未对雷暴轨迹进行精准预测与评估。

针对目前雷电预报不能给出指定区域内落雷频数和密度随雷暴的推移而变化的规律，文献[2]基于网格划分法、聚类识别云团法、线性拟合法、DBSCAN 和反距离加权插值法，预测接下来 15 分钟内任意时间点聚类云团的质心移动路径，将聚类云团质心位置预测的相对误差控制在 20%以内。在保持了精确度的同时扩大了可预测时间，为输电线路雷电预测提供了重要依据。但该文的网格间距、集群最少点数、两点间最短距离等参数的人为设定对预测结果起了一定的干扰，而且预测区域只是将当前区域以当前速度推移一个时间间隔即得到，因此这个模型有待改进。类似的运用网格划分法进行雷电识别的文献[4]使用广度优先搜索算法对网格进行聚类分析，用数据重叠法解决相近时间的雷暴识别问题，但其预测方法也较为简单，只是用前 3 个时间片的中心位置去预测下一时刻雷暴的中心位置。

闪电观测数据比雷达、卫星提供的气象数据的实时性和连续性更优、延迟更小，而且可以在数千里外被观测到，因而闪电数据用于雷暴识别是可行的。据此，文献[5]用密度极大值快速搜索算法（Clustering by Fast Search and Find of Density, CFSFD）进行聚类分析，提高了雷暴合并与分裂的识别准确率，相对于 K-means 和 DBSCAN 经典聚类算法降低了算法复杂度，用 Kalman 滤波算法实现雷暴路径的跟踪预测，将雷暴临近预测

时间提高到 60min，但是预测的准确率仍然有较大地下降。

近年来，随着神经网络的兴起，基于门控的循环神经网络开始应用于路径预测这类非线性问题中，如文献[6]-[7]运用门控神经网络 LSTM、GRU 预测未来 6 小时的台风路径，深刻把握路径信息的时间关联性，使得预测模型更加稳定高效。同样，神经网络也可以用来预测雷暴路径，但目前很少有将这一方法应用到雷暴轨迹预测的研究。此外，还有一种基于支持向量机进行改进的雷电预测方法，也取得了不错的预测效果并应用在了重庆市雷电预测系统中[8]。

国外传统的雷暴定位预测方法也有不少。基于雷达数据进行雷暴预测的 TITAN 模型[9]可谓这一领域的经典之作。基于闪电数据、跟踪空间和时间属性的预警决策综合信息支持系统[10]（Warning Decision Support System-Integrated Information, WDSS-II）提供了许多自动算法并已集成在软件中投入了实际运用，得到了可观的预测效果，其核心就是用 k-means 算法做聚类分析。类似的系统有使用时间差技术（Arrival Time Difference, ATD）的 ZEUS 网络[11]、闪电检测网络[12]（Lightning detection NETwork，LINET）。此外，也有运用人工神经网络（Artificial Neural Network，ANN）进行训练的研究，比如文献[13]使用多层感知神经网络（MLPNN）和 Levenberg-Marquardt 训练算法基于马来西亚吉隆坡机场的五种气象数据预测雷电次数，训练出了 R 值（回归相关性）为 0.99999 的模型。

1.3 研究内容

本文主要介绍了基于闪电数据的雷暴识别与路径预测算法的设计与实现，即怎么样从一个混乱的数据集中识别、提取出雷暴，构造出样本集、答案集，进而进行路径预测与评估。具体研究内容如下：

1. 雷暴识别方面，应定义一套识别准则和相关参数，识别出属于同一场雷暴的合理闪电数据并将它们提取出来构造成样本数据集与答案数据集，还需对单场雷暴数据集进行可视化；
2. 提取出多场雷暴数据集后后，通过雷暴路径预测算法，应找出雷暴移动的趋势和规律，在拟合已知数据集的基础上做时间和地点等方面的预测，并与答案数据集相比较，通过所定义的效果评估方法，得出实验结果。在实验中应针对实

际情况调整、改进、优化算法模型，提高准确预测的时间。

1.4 论文结构

本论文分为四章：第一章简述了雷暴识别与路径预测的研究背景、意义以及国内外有关雷暴识别的研究现状等；第二章节详细地讲述了基于卡尔曼滤波的雷暴识别与路径预测的步骤，并简略介绍了基于循环神经网络做路径预测的流程；第三章介绍了两种雷暴可视化方法，重点论述了单场、多场雷暴识别算法和路径预测与评估三种算法的推理过程和步骤，并对每一种算法都提出了一些优化措施，给出了流程图并做了算法复杂度分析；第四章是算法实现与实验，介绍并分析了数据集，贴出了算法实现的关键代码，设计并完成了两个实验。

第二章 路径识别与预测算法相关理论

2.1 基于卡尔曼滤波的路径预测算法

卡尔曼滤波（Kalman Filter）于 1958 年提出，是一种高效率滤波算法，在导航通信、制导控制、天气预报等众多领域都有广泛的应用。由于便于编程实现、能对现实采集的数据实时更新、对于解决很大部分问题是最优方法，卡尔曼滤波成了目前应用最广泛的滤波技术。它通过一组线性系统状态方程从时间序列中分析出噪声数据，利用预测、更新两步程序迭代地减少误差，动态修改预测权值[14],进而对系统的状态变化进行最优估计。这一算法能平滑处理雷暴路径数据，去除噪声，还原出真实数据，在长时间路径预测中也可以得到较高的准确率。

基于卡尔曼滤波的路径预测算法主要包括五个步骤：1.通过聚类分析进行初步识别；2.根据卡尔曼时间更新方程读取并预测雷暴状态；3.基于雷暴状态信息跟踪雷暴路径；4.根据卡尔曼滤波迭代地更新雷暴状态；5.通过迭代外推得到预测路径。

第一步是进行雷暴的初步识别。首先在历史数据集中检测首次或下一次出现闪电的时间，获取未来 t_n 时间段内的闪电数据，然后将闪电坐标网格化，扫描网格，将符合闪电频数阈值和雷暴面积阈值的网格合并，将符合要求的网格标记为闪电区域，合并相邻网格并对区域编号。最后，根据主成分分析，利用椭圆法包络出合理的闪电区域，返回初识别的雷暴数据集。整个初步识别的流程图如下所示：

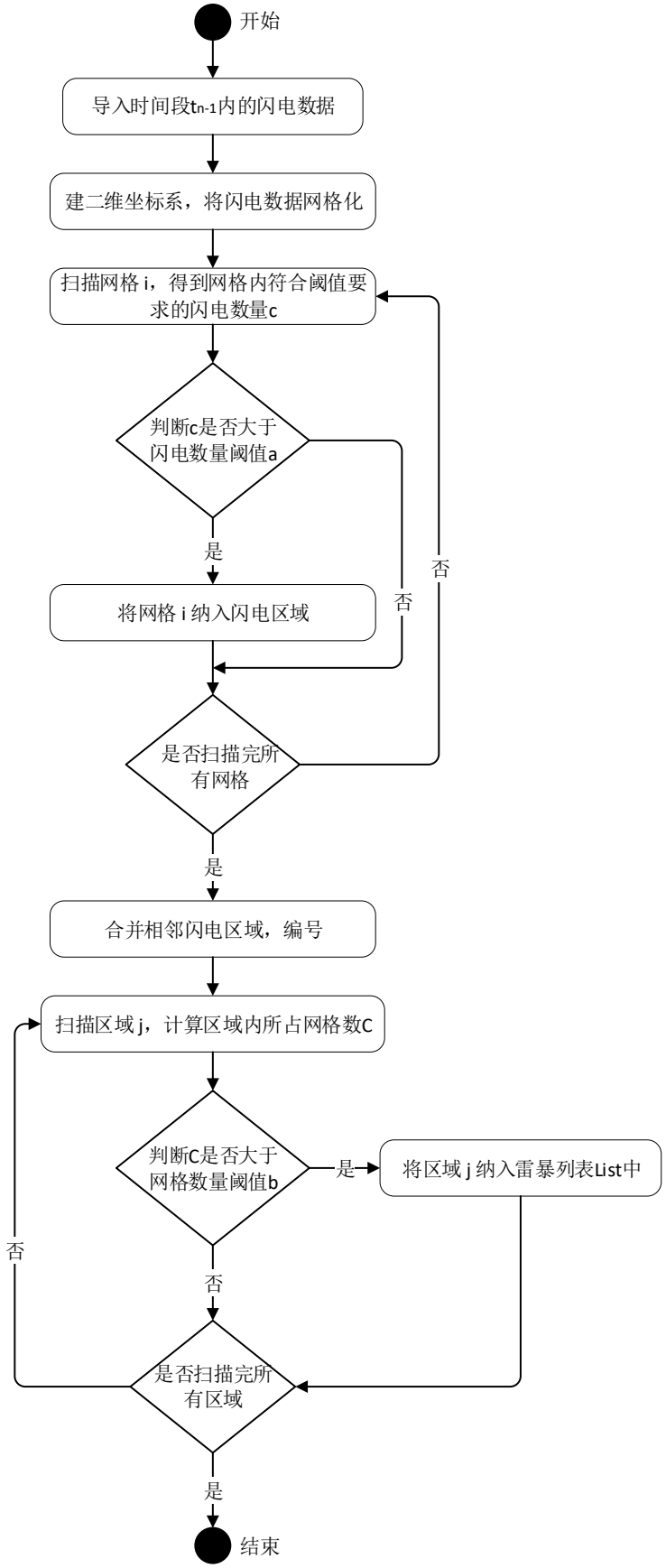


图 2-1 卡尔曼滤波第一步--雷暴初步识别流程图

第二步是根据卡尔曼方程读取并预测雷暴状态。由第一步可得到 t_{n-1} 时间段内的雷暴列表 List，初始化雷暴状态矩阵 X_{n-1} 和协方差矩阵 P_{n-1} 后，遍历该 List 中 t_{n-1} 时间段内所有闪电数据，根据卡尔曼时间更新方程：

$$X_n^- = FX_{n-1}$$
$$P_n^- = FP_{n-1}F^T + Q$$

计算出 t_n 时间预测的雷暴状态矩阵 X_n^- 和协方差矩阵 P_n^-，其中 X_{n-1}=(lx，ly,Vx,Vy)，即 t_{n-1} 时间雷暴的横向坐标、纵向坐标、横向速度、纵向速度，F 是状态转移矩阵，Q 是状态转移噪声矩阵，二者表示如下：

$$F(t)=\begin{pmatrix}1 & t & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & t\\ 0 & 0 & 0 & 1\end{pmatrix}, Q(\Delta t)=\begin{pmatrix}\frac{r^2}{3t} & \frac{r^2}{2t} & 0 & 0\\ \frac{r^2}{2t} & \frac{r^2}{t^2} & 0 & 0\\ 0 & 0 & \frac{r^2}{3t} & \frac{r^2}{2t}\\ 0 & 0 & \frac{r^2}{2t} & \frac{r^2}{t^2}\end{pmatrix}$$

其中 r 是雷暴中心的噪声强度，t 是时间段间隔，单位是分钟，一般取值 5-10 分钟。

第三步是跟踪雷暴路径。首先进行雷暴匹配，将两个相邻时间点的雷暴作为二分图的两个集合 A、B，A 中有识别出的雷暴 N_1 个，取第 i 个雷暴的中心点经纬度坐标为 $\bar{x}_i$ 和 $\bar{y}_I$，面积为 S_i，$0<i\le N_1$，于是 A 中雷暴 i 的状态为($\bar{x}_i$, $\bar{y}_i$, S_i)，同理 B 中雷暴 j 的状态为($\bar{x}_j$, $\bar{y}_j$, S_j)，$0<j\le N_2$。雷暴中心点的位置差 d_p 和雷暴面积平方根的差 d_s 为：

$$d_p=\sqrt{(\bar{x}_i-\bar{x}_j)^2+(\bar{y}_i-\bar{y}_j)^2}, d_s=\left|s_i^{\frac{1}{2}}-s_j^{\frac{1}{2}}\right|$$

令 A 中每个雷暴 i 与 B 中每个雷暴 j 的代价函数 $C_{ij}=d_p+d_s$，将其相反数作为二分图中边的权值，随后利用最小权值匹配找到使代价函数的和最小的雷暴匹配方案。随后，对生于未匹配的雷暴进行分裂与合并处理，返回匹配结果作为观察值。

第四步是更新雷暴状态。遍历第三步得到的观察值列表，如果观察值 Y_n 由合并雷暴产生，则将其速度向量合并，更新 X_n，若 Y_n 不是由合并雷暴产生，则由卡尔曼状态更新方程计算卡尔曼增益 K_n，并更新后验状态矩阵 X_n 和后验协方差矩阵 P_n：

$$K_n = P_n^- H^T (HP_n^- H^T + R)^{-1}$$
$$X_n = X_n^- + K_n(Y_n - HX_n^-)$$
$$P_n = (I - K_n H)P_n^-$$

其中，I 是单位矩阵，H 为观察模型矩阵，R 是观察噪声矩阵，H、R 表示如下：

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \mathrm{R} = \begin{pmatrix} r^2 & 0 \\ 0 & r^2 \end{pmatrix}$$

第五步是通过外推得到下一时间段的雷暴预测路径，若下一时间段存在闪电则继续迭代，若不存在闪电则结束程序。下图举例说明了四个时间段的外推路径：

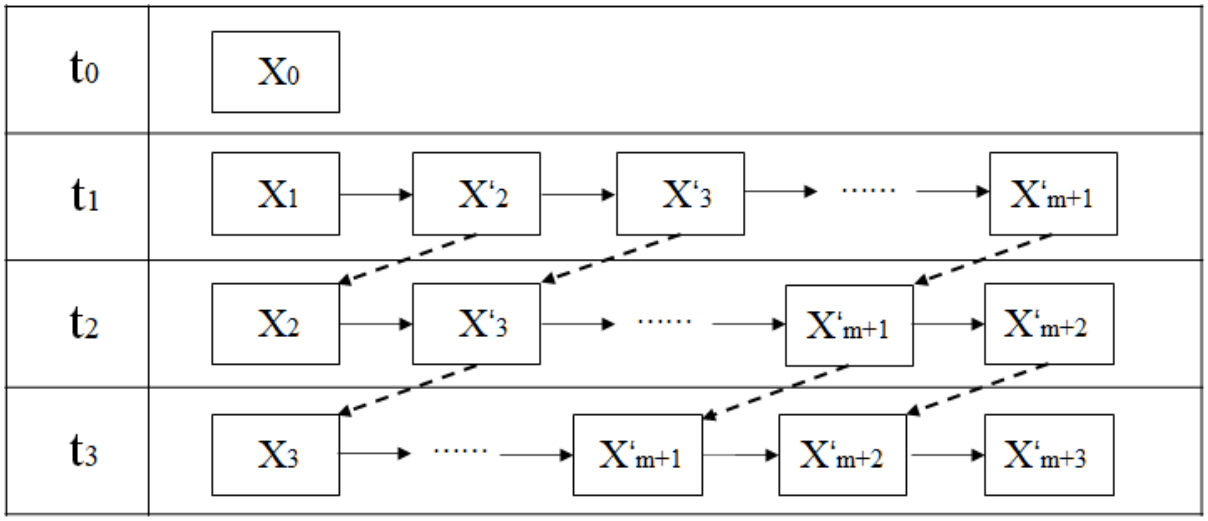


图 2-2 路径预测外推法示意图

在 t_0 时间段第一次检测到雷电的位置，还没有形成雷暴路径，无法预测雷暴运动趋势，因此没有外推路径。在 t_1 时间段根据 X_2 外推得到 t_2、t_3、……、t_{m+1} 的雷暴状态，即 X'_2、X'_3、…、X'_{m+1}，连接这 m 个雷暴的位置即可得到雷暴的预测路径。在 t_2 时间段中，t_1 的预测状态全部更新，即虚线所示，从而可得到更新后的预测路径，在 t_3 时间段继续更新预测状态，随后一直迭代外推直到雷暴消失。

整个雷暴路径预测算法流程图如下所示：

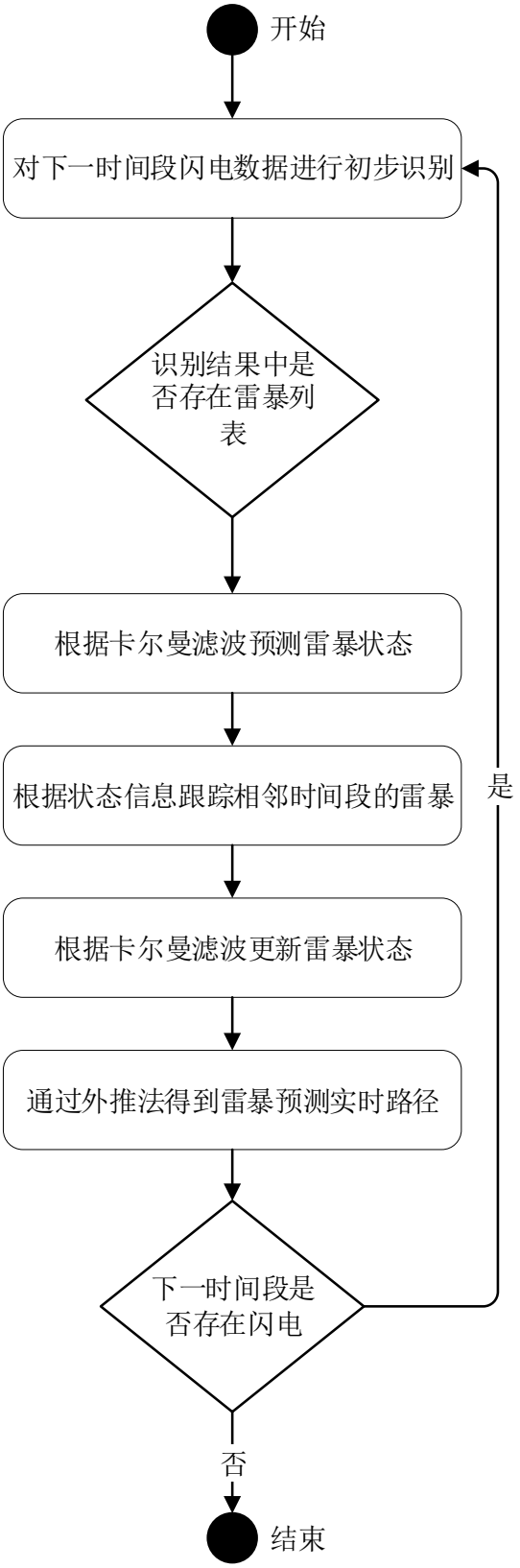


图 2-3 基于卡尔曼滤波的雷暴路径预测算法流程图

2.2 基于循环神经网络的路径预测

循环神经网络（Recurrent Neural Network, RNN）是一种递归神经网络（Recursive Neural Network）[15]，它以序列排列的数据为输入、具有在短时间内的记忆功能。这里的序列数据可以是随时间变化的序列，输入输出不受维度固定不变的限制，而雷暴数据中同一时间可能有多个闪电的发生，预测雷暴路径也需要预测出未来一段时间内多个可能发生闪电的位置点作为输出，可见输入输出维度不确定，同时预测雷暴也需要有一定的记忆功能，未来雷暴的路径可能受过去几十分钟时间内位置变化的影响而不只是过去几分钟的位置就能决定的，因此，循环神经网络也适合于预测雷暴路径，不过前提是已经识别出了合适的雷暴数据集，只需要做路径预测任务，而且要有足够大的雷暴数据集。用循环神经网络预测路径的流程图如下所示：

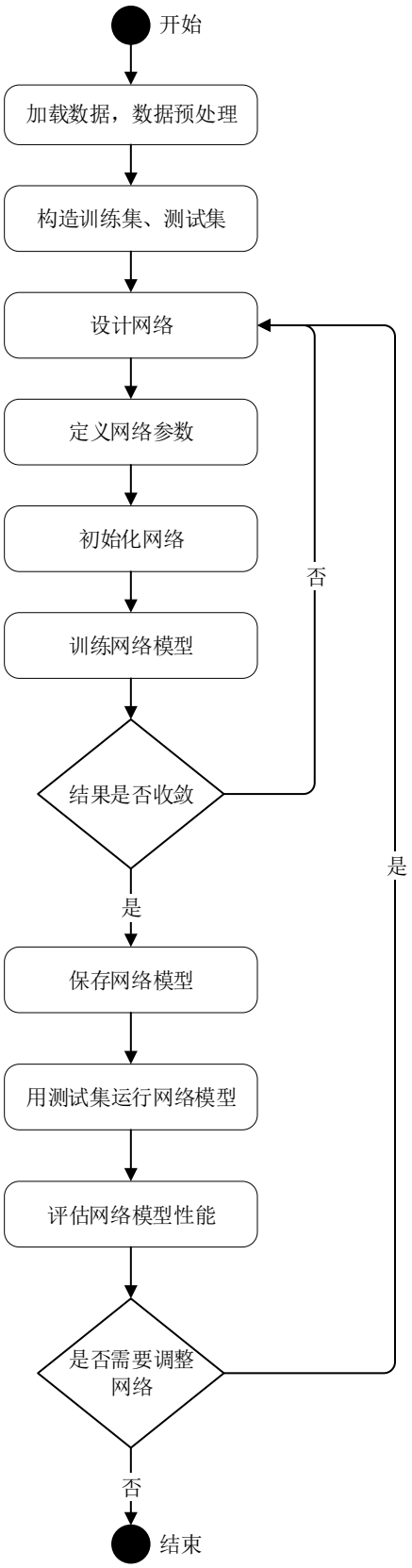


图 2-4 循环神经网络预测路径流程图

第三章 基于闪电数据的雷暴识别及路径预测算法

3.1 算法基本思路

闪电随时间变化明显、能准确地反应雷暴路径的变化、相关数据容易被观测与收集，因此可以以闪电的发生时间、地点表示雷暴发生的时间、地点，以闪电的强弱表示雷暴的强弱，达到跟踪雷暴路径的目的，所以将闪电数据作为算法输入是合理的。

考虑分雷暴识别、路径预测两步实现算法。如果识别出的雷暴数据集有很多异常值不属于同一场雷暴，则将大大增加雷暴路径预测的难度，降低预测准确率。如果能尽可能多地将同一场雷暴中发生的闪电数据归为一类，则可以准确体现出雷暴运动的规律，大大提高雷暴路径预测准确度。因此，科学合理的雷暴识别算法是准确预测路径的保证。

雷暴识别算法的输出分两方面：单场雷暴识别中，应检测查询时间那一时刻是否有雷暴，如有，将提取的单场雷暴数据可视化，绘制成清晰易懂的图；多场雷暴识别中，应在一个大的原始数据集中提取出多场雷暴数据作为样本数据集，构造相应的答案数据集，最后将二者总和起来作为下一步雷暴路径预测算法的输入。

雷暴预测算法的输出应该是针对数据集中每场雷暴预测出接下来闪电可能发生的时间、地点等，形成预测数据集，定义一种针对预测数据集的评估方法，得到结果数据集，定量地做准确度分析与算法评估。

3.2 符号表

表 2-1(a) 算法中定义的符号

符号	释义	单位
NowTime	查询时间：当前时间或用户输入的时间	年 月 日 时 分
BeforeTime/LastTime	雷暴有效最长持续时间	秒
TimeDifference	相邻两个合理数据点之间的最大时间差	秒
ValidDistance	相邻两个合理数据点之间的最大距离	千米
MaxSpeed	相邻两个合理数据点之间的最大速度	千米/秒
MinIntensity	每个合理数据点的最小强度	安培

表 2-1(b) 算法中定义的符号

符号	释义	单位
MinNum	单场雷暴样本数据集的最少合理数据点	个
ECRatio	Error-Correct Ratio 单场雷暴数据中不合理数据点个数除以合理数据点个数 的最大值，即最大筛除比	—
RowInterval	满足 LastTime 时间段内的单场雷暴原始数据点个数最少值	个
MaxAnswerTimeNum	答案集时间点最大个数	个
PredictDegree	多项式拟合的最大次数	次
Score	一次预测任务的评估分数，也即平均准确预测时长	分
MissRate	一次预测任务中的漏报率（有雷暴而预测无雷暴）	次
WrongRate	一次预测任务中的误报率（无雷暴而预测有雷暴）	次
AccuracyRate	一次预测任务的准确率	--

3.3 单场雷暴识别

闪电有很多种类型，根据颜色区分，有黑色、紫色、红色、蓝白色闪电等等，根据形状区分，有线状、球状、片状、带状、联珠状、火箭状闪电等等，此外还有特殊的超级闪电和海底闪电，本文所指闪电均指最常见的蓝白色线状闪电。单次闪电持续的时间极短，通常在 0.1-0.3s 之间，长度通常为几百米至上千米，本文研究在一个大区域内的一场雷暴中多次闪电的移动路径，故把单次闪电的整个过程视为雷暴路径中的一个数据点，即一个基本单元来处理。

在一个混乱的数据集中，可能同一时间有多个闪电发生，地点各不相同，可能有的属于同一个地方，有的距离很远根本没有相关性，可能有的在数据集中相差很远但地点、时间上相近，属于同一场雷暴。雷暴识别的目的就是准确地将属于同一场雷暴中的闪电

数据点尽可能多地识别、收集出来并做可视化，将不属于同一雷暴中的数据点区分开来。为此，需要考虑时间、区域（距离）、速度、强度等因素。

时间上，相距很久的两个数据点（闪电）自然可能已经不属于同一场雷暴了，为此，定义两个常量：单场雷暴的有效最长持续时间 BeforeTime，单位为分钟；查询时间（可以是当前时间，也可以是用户输入的一个时间）NowTime，单位为分钟。将 NowTime 之前雷暴有效的最长持续时间段 BeforeTime 内的数据点视为属于在时间差上合理的同一场雷暴，将 (NowTime-BeforeTime) 时间之前的历史数据视为不影响当前路径预测的历史数据。此外，数据集中相邻两行即两个数据点之间的时间差如果过大，这两个闪电也可能已经属于不同的雷暴了，为此，定义常量相邻两个合理数据点之间的最大时间差 TimeDifference, 单位为分钟，从查询时间 NowTime 往前遍历最长持续时间段 BeforeTime 内的数据，若上一数据点与下一数据点之间的时间差大于 TimeDifference，则把上一数据点视为另一场雷暴中的闪电或者视为一个异常数据，将之剔除。

距离上，单场雷暴可能跨越很远的距离，但与上文的最大时间差 TimeDifference 类似，相邻两个数据点之间的距离差如果过大，也可能已经不属于同一场雷暴了。为此，定义常量相邻两个合理数据点之间的最大距离 ValidDistance，单位为千米，以最靠近当前时间的数据点为中心向当前时间之前的数据点遍历，任意两个相邻的数据点之间的二维球面地理距离（即经纬度距离，不考虑海拔）不能超过 ValidDistance 千米，若超过此值，将上一时间的数据点剔除。此外，常量 ValidDistance 千米也可抽象地视为一个闪电的最大长度。

速度上，雷暴云的移动主要是由于风的作用，所以数据点之间的速度应小于风速，如果大于最大风速则说明产生了不属于同一场雷暴的数据点或异常数据点。为此，定义常量任意两个相邻数据点之间的距离除以时间差所得的最大平均速度 MaxSpeed，单位为千米/秒，在进行上述时间、距离检测遍历的同时在加一条速度限制，将速度超过 MaxSpeed 的两个数据点中时间发生早的数据点提剔除，如果两个数据点是同时发生则可都视为速度上合理的数据点。

根据查询时间 NowTime 将满足上述时间、距离、速度三方面条件的数据点筛选出来后即得到了单场雷暴的数据点集。将查询时间 NowTime 之后最近的在同一时间发生的数据点集合视为下一时间即待预测的答案数据点集，同时这些数据点间的距离也需满

足相邻两数据点距离小于 ValidDistance 千米的限制条件，需特别注意的是答案集的第一个点应该与之前筛选出的样本集的最后一个合理数据点相比较而不是直接与上一行的数据点相比较，故首先应进行针对答案集的一个合理性检测，如果得不到一个合理数据点则答案集构造失败。

单场雷暴数据集筛选出来后应设计可视化方法以方便观测识别效果。将同一时间发生的数据点以同一种颜色在坐标轴上描出并连接成线或圈，该线圈可视为某一时间同一场雷暴活动的形成的一个锋面或者一个区域，不同的时间对应不同的颜色，根据数据点闪电强度的大小确定数据点在图中半径的大小，这样就可以描述出单场雷暴的活动路径图了，把此方法称为“线圈法”。

下面提出一些优化措施。

1. 如果合理的数据点过多，影响可视化效果，可以定义最小强度常量 MinIntensity，单位是安培，将无足轻重地强度小于 MinIntensity 的一些数据点舍去；
2. 数据点过多时“线圈法”所识别出的雷暴重叠度很大，完全看不出雷暴的运动路径。可以取同一时间发生的多个闪电数据点的经纬度平均值所得的地理位置作为该时间点雷暴活动的中心，对答案集也取中心位置单独描点绘出，这样画出的图可以较为准确的描绘出雷暴运动的轨迹，把此方法称为“中心路径法”；
3. 如果整个雷暴的数据点数过少，可定义常量单场雷暴的最少数据点数 MinNum, 单位为个，若筛选出的雷暴数据点集的数据点个数小于 MinNum, 即可将这场雷暴的数据集视为无效；
4. 如果不合理数据点个数大大超出合理数据点个数，识别出的雷暴可信度必然下降。为提高识别可信度，定义常量最大筛除比 ECRatio，无单位。如果在识别一场雷暴数据过程中不合理的数据点总数除以合理的数据点总数所得的比值大于 ECRatio，则可直接将本场雷暴数据视为不合理数据。

至此，单个雷暴识别完毕，整个过程的流程图如图 3-1 所示。

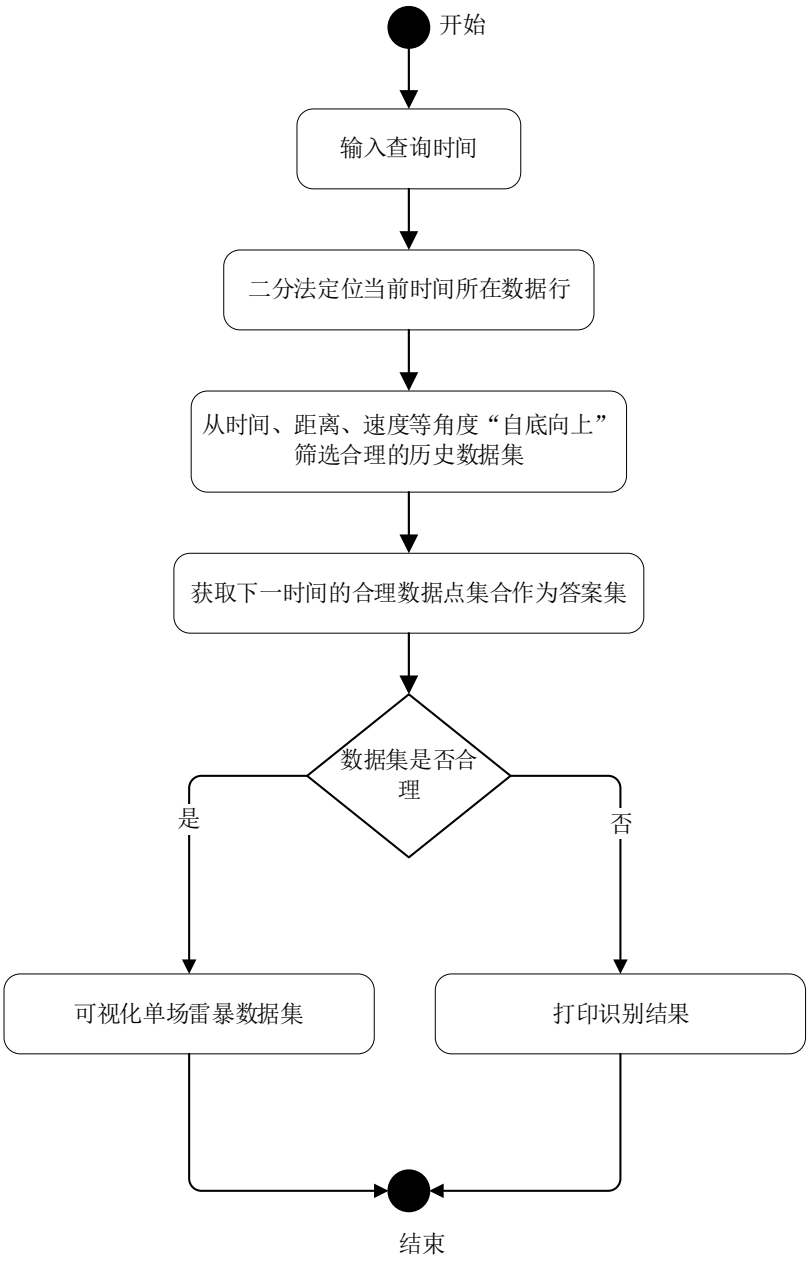


图 3-1　单场雷暴识别流程图

最后，评估一下本算法的复杂度。用于定位当前查询时间所在的数据行号的二分法也叫折半查找法，其复杂度是 O(log N), N 是整个原始数据集的数据点个数。随后对 BeforeTime 时间段内的数据做检测，设数据点个数为 n，每个数据点只检测一次，答案集的时间可忽略不计，故总的时间复杂度是 O(log N + n)。二分法的空间复杂度是 O(1), 做数据检测筛选时需保存合理数据点的行号，最多的情况是 BeforeTime 时间内的数据点都是合理的，都需要保存进样本数据集，故总的空间复杂度是 O(n)。

3.4 多场雷暴识别

多场雷暴识别的目的是利用单场雷暴识别的算法在一个大数据集中迭代地识别出全部合理的雷暴样本数据集并保存起来作为下一步雷暴路径预测的输入。

一种直观的方法是遍历每一个数据点，以该数据点为单个雷暴的开始位置，“自顶向下式”地做雷暴识别（注：单场雷暴识别是先有查询时间，故而是“自底向上式”的识别雷暴。），但这种做法算法复杂度太高了，达到了 O(nN)，n 是单场雷暴样本在 LastTime 时间段内的最大历史数据点个数，N 是整个数据集的数据点个数，也就是做 N 量级次数的 n 量级单场雷暴识别。这里不需要“log N”项了，因为是“自顶向下”依次遍历。

为了降低算法复杂度，定义常量行间距 RowInterval，单位为个，即预估一场雷暴的数据点个数是 RowInterval。对每 RowInterval 个数据点只做一次单场雷暴识别，若在中间检测出无雷暴就立即进入下一次迭代，即下一段的 RowInterval 个数据点；否则就遍历这个雷暴在 LastTime 时间段内的数据点，其个数可能大于、等于、小于 RowInterval，识别完单个雷暴样本数据集后在结束点接着往下开始识别下一个雷暴而不是跳过 RowInterval 个数据点。易知，这种做法的时间复杂度是 O(N)。

上面第一种方法可以将整个数据集里尽可能多地雷暴识别出来，理论上可以把合理的符合单场雷暴识别算法规则的所有样本都提取出来；第二种方法提高了效率，但是每次处理雷暴判断出是不合理雷暴时就跳过 RowInterval 可能会错过一些合理样本数据集，不能将所有的雷暴都提取出来。

为此，可以做一个优化，即在每次迭代的开始时做一次预判：

1. 设本轮迭代起点为第 i 行，读取第 i 行对应闪电的发生时间 StartTime;
2. 计算雷暴理想结束时间 EndTime = StartTime + 持续时间 LastTime；
3. 取 j = i+RowInterval；
4. 如果 j 大于等于数据集最大行数 RowNum，那么程序结束；否则读取第 j 行数据，得到对应闪电的发生时间 jTime；
5. 如果 jTime 小于等于 EndTime，说明这 RowInterval 个数据点都在合理时间范围内，预判这是一场雷暴，继续本轮识别；否则，让 i 加一，返回第一步进行下一轮迭代。

这一预判未改变算法复杂度 O(N)，但可以在 O(1)时间内判断接下来的 RowInterval

行是否可能构成一个雷暴，如果不可能，则只排除一个当前迭代的起点行而不是排除整个 RowInterval 行的数据，因为接下来的 RowInterval-1 行里可能有合适的雷暴起点行，这样就不至于使大量雷暴得不到识别。

此外，上文所述在单场雷暴识别“中间检测出无雷暴就立即进入下一个迭代，即下一段的 RowInterval 个数据点”，这里所述的“中间检测出无雷暴”的情况包括：1.合理的数据点个数小于 MinNum 个；2.样本筛除比大于 ECRatio；3.答案数据集不合理，与历史数据集的最后一个合理数据点相差甚远，不属于同一个雷暴。就此，可以做两个优化 ：

1. 假设本轮识别从起点第 i 行开始，在第 j 行检测到错误，则下次迭代不必从第 i+RowInterval 行开始，而是可以从第 j 行开始。无论检测到哪种错误，既然进行了预判，那么检测到错误时的 j 一定大于等于 i+RowInterval，第 i+RowInterval 行到第 j 行之间的数据已经在本次识别中处理过一次且属于本轮的无效样本数据集了，无需再次从第 i+RowInterval 行开始进行重复检测，这从一定程度上加大了算法复杂度，就实现上而言，下轮迭代从第 j+1 行开始也更加方便。所以无论是检测到雷暴还是未检测到雷暴，下次迭代都可以从本轮最后遍历的结束点处开始；
2. 为提高雷暴识别成功率，可以从数据集位置和合理性检测两个角度优化答案数据集的构造方法。按之前的方法，在多场雷暴识别中，由于是“自顶向下式”地做单场雷暴识别，在识别完历史数据集后，已经筛选出了 LastTime 时间段内的合理数据，答案数据集只能在 LastTime 时间段之后紧接着的数据中搜索，而这时雷暴可能结束，或者有另一起雷暴在干扰，很容易导致检测不合理，使得之前的识别功亏一篑。为避免这一临界点问题，可以在做单场雷暴识别时在 LastTime 时间段的最后留出一个时间点的数据用于构造答案数据集。此外，在答案数据集的合理性检测方面，避免只使用在答案时间点内发生闪电的数据集中的第一个数据点与历史数据集中的最后一个合理数据点相比较进而直接判断数据集的合理性，而是遍历答案时间点对应的数据集中的所有数据，只要有一个数据点与历史数据集最后一个合理数据点检测通过，则认为答案数据集是合理的，本轮雷暴识别成功。

最后，为了方便后面的路径预测算法要预测多个时间点的需求，答案集应该不止只有一个时间点的数据，故可定义常量 MaxAnswerTimeNum，单位为个。假设数据集的最小时间单位为分钟，则其实际意义就是在接下来 MaxAnswerTimeNum 分钟内而不是一分钟内寻找合理的答案集数据点。但是最后提取的答案集中不一定都跨越 MaxAnswerTimeNum 个时间点，因为雷暴可能消失或者某些时间点没有对应的合理数据点。MaxAnswerTimeNum 是一个常量，若设置得过小，则有碍对较长时间预测的效果评估；若设置得过大，则可能使得样本数据点不足，因为上面第二个优化规定了答案集数据是在原来的样本原始数据集 LastTime 时间段内划分出 MaxAnswerTimeNum 个时间点而来的，也就是说单个样本数据集和答案数据集的时间点个数之和等于 LastTime 所跨的时间点个数。此外，由于答案集时间点的扩张，在“中心路径法”可视化中答案集出现了多个雷暴中心位置，这时应注意将答案集第一个中心位置与样本集最后一个中心位置相连接以达到连续的效果。

在单场雷暴识别中可以通过“线圈法”、“中心路径法”两种可视化方法对雷暴识别的效果做定性分析，并可以通过 ECRatio 指标做定量分析。在多场雷暴识别中，也可以在成功识别出一场雷暴后就将其可视化，每轮迭代结束时打印输出本轮结果并对不同的结果做计数，在识别完整个数据集后就可以得到产生各种结果的次数。

至此，多场雷暴识别结束，整个过程的流程图如图 3-2 所示。

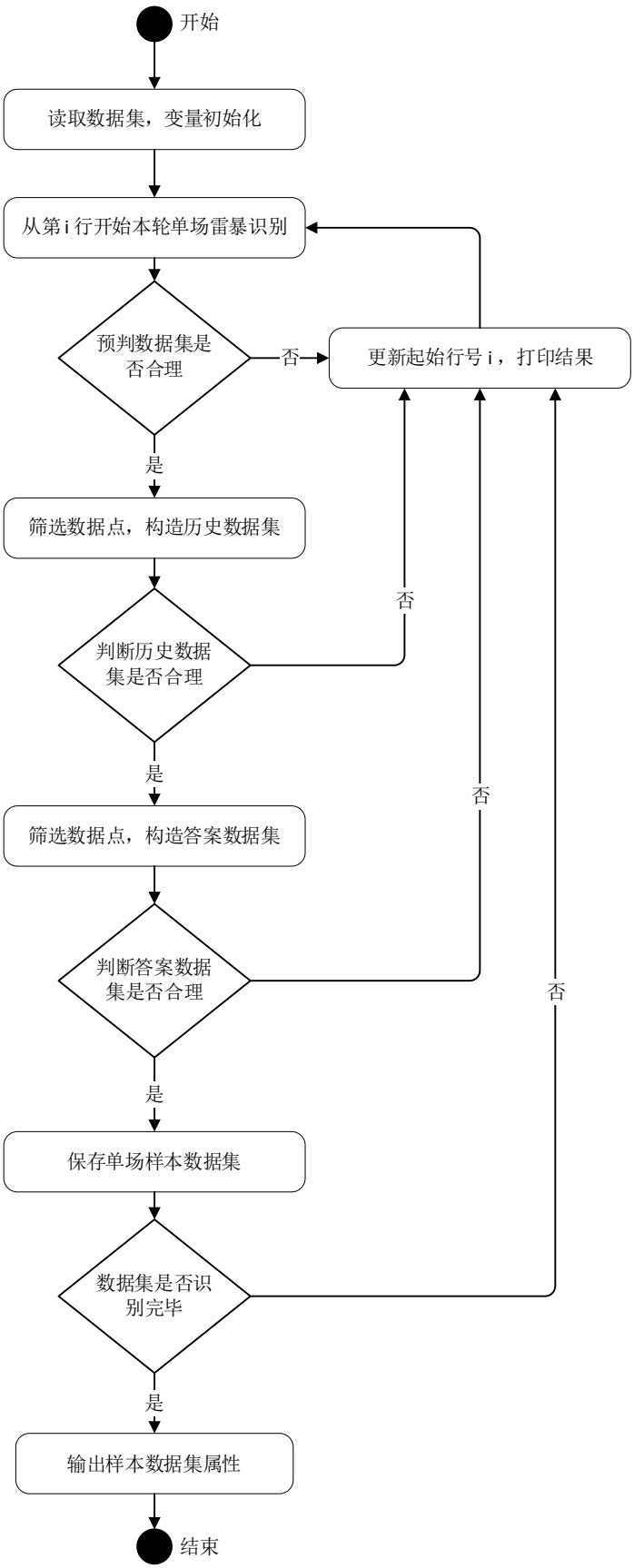


图 3-2 多场雷暴识别流程图

最后说明一下本算法的复杂度。前面已分析出时间复杂度为 O(N)。在空间上，最多的情况是每个数据点都是合理的，需要暂存，将单场雷暴识别提取的样本数据集保存起来，在每次迭代时的空间复杂度相当于单场雷暴识别的空间复杂度，即 O(n)。故总的空间复杂度为 O(N)。

3.5 雷暴路径预测与评估方法

雷暴路径预测与效果评估二者联系紧密，故本文结合起来分析，其基本思路是：经过多场雷暴识别算法提取出多个样本雷暴后，针对每个样本单独进行路径预测，保存在预测结果集中，最后统一通过评估方法将预测结果集与答案数据集相比较，得到一个最终的具有一定实际意义的指标，以此来评定预测效果的优劣。

为了便于量化预测结果，进行定量分析，预测算法应该避免同一时间有多个输出的问题，答案集同理也应避免这一问题，故在开始预测前可以先对所有样本进行一个数据清洗操作，将样本集、答案集的每一个样本中同一个时间点发生的所有数据点通过分别对经纬度取平均值得到雷暴中心的位置（与单场雷暴可视化的第二种方法“中心路径法”相似），这样同一时间只对应一个地点，预测集与答案集在同一时间地理位置的距离就可以作为一个预测效果的评定指标。

数据清洗后，一个样本的数据点个数应该是在 10-100 量级，适合用多项式回归法进行拟合、预测。由于样本个数很少，而且不同样本之间没有必然联系，因此需要对每个样本单独进行一次多项式回归分析，得出对应的预测模型。

和数据清洗一样，继续将经纬度分别处理，分开来进行预测。将时间点作为输入 x，将经度、纬度分别当作 y_1,y_2, 定义常量多项式次数 PredictDegree，根据最小二乘法定义损失函数 Loss，将 Loss 进行 L2 正则化后对多项式的系数矩阵 w 求导并令导数为 0，可直接得到 w 的最优闭式解；也可根据随机梯度下降法迭代地更新系数矩阵 w 直到得到一个损失函数值极小的最优解。

得到最优的系数矩阵 w 后就有了多项式预测函数，将未来时间作为输入带入表达式即可得到经纬度的预测值，随后把每个样本的预测值记录在预测数据集中。这里的未来时间可以事先设定，也可以根据答案集中的实际时间点个数确定应该预测多少个数据，一方面可以避免预测集与答案集大小不一样导致错误，另一方面可以减少运算量，因为

预测过多数据点而答案集中没有比较对象时的预测值是无法利用的。

此外，由于时间是包含年、月、日、时、分的，直接作为输入显然不妥，考虑到经过清洗后的数据点的时间都不一样，而且雷暴是实时监控的，相邻两个数据点之间的差应该是定值，即最小时间单位，也就是说此时的时间序列是一个等差数列，故可以直接将样本集的时间转化为从 1 开始累加 1 的自然数序列，在做多项式回归预测时这与原始时间序列是等效的。

得到预测数据集后，应定义判断一个时间点是否是一次成功预测的方法。首先，根据强度与 MinIntensity 的大小关系确定是否有闪电发生。如果预测值判断无闪电发生而答案集中对应时间点是有雷暴发生，则说明这是一次漏报。为此，定义变量 MissNum 进行漏报计数。如果预测值判断有闪电发生而答案集中对应时间点是无雷暴发生，则说明这是一次误报。为此，定义变量 WrongNum 进行误报计数。如果预测值与答案值都显示无闪电发生，则认为是一次成功的预测。如果预测值与答案值都显示有闪电发生，则计算出预测位置与答案位置的误差距离，若小于 ValidDistance，则可以认为是一次成功的预测。将每个样本的预测数据集与答案集一一比较，将每个时间点的误差距离、是否误报和漏报等数据构成本次预测的结果集。将 MissNum 和 WrongNum 除以总的预测次数，便得到了漏报率 MissRate 与误报率 WrongRate。

为了进一步定量分析整体的预测效果，定义评估分数 Score 与准确率 AccuracyRate。Score 指在一次预测任务中平均准确预测出每个样本未来 Score 分钟内的路径，计算方法为对所有样本成功预测的个数之和除以样本个数。AccuracyRate 指预测集中所有样本在未来 Score 分钟内成功预测的个数之和除以总的预测次数，预测次数等于 Score 乘以样本个数。假设最小时间单位是分钟，k 个样本中有 p 次预测成功，则 Score 等于 p/k，即认为该算法成功预测了未来 p/k 分钟的路径。假设这 k 个样本在未来 p/k 分钟内预测成功的次数为 q 次，则 AccuracyRate 等于 q/(p/k*k)，即 q/p，该算法以 q/p 的准确率成功预测了未来 p/k 分钟内的雷暴路径。

至此，对每次预测任务可以按 Score、AccuracyRate、MissRate、WrongRate 四个指标来评估预测效果，雷暴路径预测与评估算法结束，整个过程的流程图如图 3-3 所示。

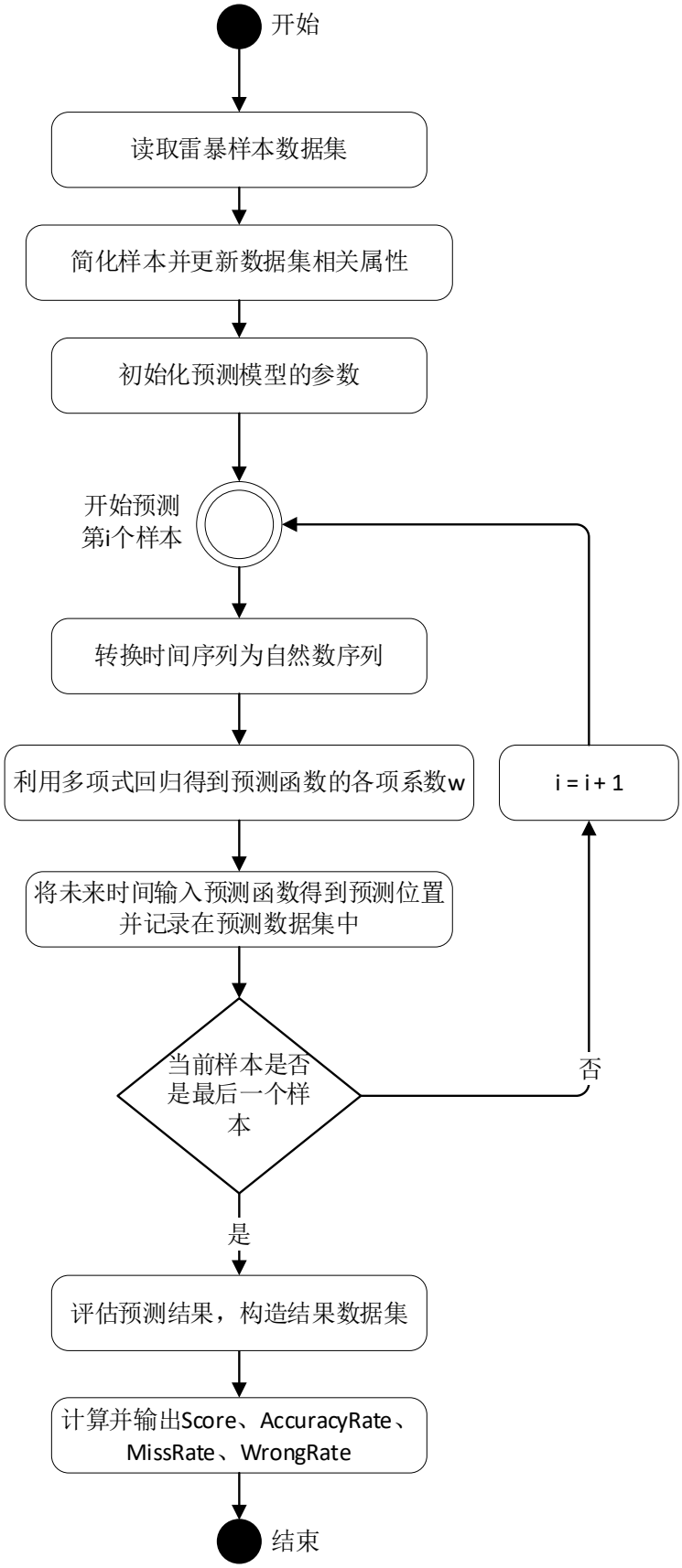


图 3-3 雷暴路径预测与评估流程图

最后分析一下本算法的复杂度。设多项式次数为 d，平均每个答案集大小为 q，原始数据集大小为 N，有样本集 k 个，平均每个样本的大小为 n，则 N >> k * n。简化数据集的时间复杂度为 O(kn)；每做一次多项式拟合的时间复杂度为 $O(d^3+dn')$，其中 n’是化简后的单个样本大小，如果取持续时间为 1 小时，以分钟为单位，则 n’小于等于 60；评估一次预测结果的时间复杂度为 O(q)。d 一般小于等于 10，q 一般小于等于 20（即预测未来 20 分钟的路径），所以 d、n’、q 都可以看成常数，即每做一次预测与评估的时间复杂度为 O(1)，预测与评估任务的时间复杂度是 O(k)。故预测算法总的时间复杂度为 O(kn+k)，即 O(kn)。同理每做一次数据清洗和预测的空间复杂度也可以看成 O(1)，易知本算法的空间复杂度为 O(k)。

3.6 调参方法

为了得到理想的预测结果，各个参数科学合理的设置成为了一大难题。前面三种算法共定义了 10 个常量参数，每个参数有不同的取值区间，每个参数对 Score 指标的影响程度也不尽相同。根据经验可知，有的参数相关性很小，比如 MinIntensity 与 MinNum、ECRatio 与 PredictDegree。有的参数之间有明显的相关性，比如 TimeDifference 与 MaxSpeed 的积应该约等于 ValidDistance，又比如 TimeDifference、MaxAnswerTimeNum 与 LastTime 之间应该达到一个动态平衡。

考虑到以上因素，可以利用经验法预估出每个参数的大致取值区间，根据每个参数对 Score 的影响力权重进行排序，优先考虑影响权重较大的参数。由于路径预测准确的前提是有识别得好的雷暴样本，因此可以先利用可视化方法进行调参，根据可视化效果的好坏预先确定一套可以得出合理雷暴的参数作为基准。然后利用控制变量法，根据参数权重顺序由高到低依次控制变量、迭代实验、更新基准，在优先确定影响力大的参数取值后再对影响力小的参数进行细化调参，同时考虑对具有相关性的参数进行联合调整，在一个原始数据集上调到最优预测效果后可以将之应用到整个原始数据集上以判断其泛化的能力。具体调参细节与结果将在下一章中系统阐述。

3.7 本章小总结

本章主要讲述了单场雷暴识别、多场雷暴识别、雷暴路径预测三个算法的推理思路、

步骤以及相应的优化方法，定义的变量、常量等符号贯穿三个算法始终，在每个算法小节的末尾给出了具体的流程图，分析了算法的时间与空间复杂度，最后，简略介绍了调参方法。

在单场雷暴识别算法中具体介绍了根据用户输入的查询时间“自底向上”地进行单场雷暴的识别过程，两种对单场雷暴数据的可视化方法，以及样本数据集、答案数据集的构造方法；在多场雷暴识别中具体介绍了在一个现有数据集中识别并提取多场合理雷暴样本的方法，优化了前面两种数据集的构造方法；在雷暴路径预测中简要介绍了对经纬度分别做多项式回归分析的预测方法，同时给出了预测数据集、结果数据集的构造方法，并在最后总结出了 4 个对预测效果的评估指标 Score、AccuracyRate、MissRate 和 WrongRate。

综合以上三个算法的复杂度，从原始数据集到结果数据集的整个过程，总的时间复杂度和空间复杂度都是 O(N)。

第四章 雷暴识别与路径预测实验

4.1 实验平台与数据集

实验平台的各项参数如下表所示。

表 4-1 实验平台参数

参数类型	参数值
操作系统	Windows 10 家庭中文版
处理器	Intel® Core™i5-8250U CPU @1.60GHz 1.80GHz
内存	8GB（7.87GB 可用）
编程环境	PyCharm Community Edition 2019.3.4 x64
Python 编译器版本	3.7

实验数据集由一组“.cvs”格式的文件组成，每一个表格文件包含一个月份检测到的数据，文件名包含年份和月份。根据实验平台的参数和数据集的大小选择具有可行性的数据集，如表 4-2 所示，每个数据集中的数据范围如表 4-3 所示。简便起见，在算法实现中将文件名改成了编号。

表 4-2 原始数据集

文件名	编号	大小（行数）
2015.04.csv	1	598046
2015.10.csv	2	480204
2015.11.csv	3	448060
2016.03.csv	4	981923
2016.04.csv	5	1048575
2016.10.csv	6	232469
2016.11.csv	7	123964
2017.10.csv	8	839183
2018.09.csv	9	276212
总计	9	5000000

表 4-3 数据集各属性的范围

属性	最小值	最大值
时间	2015/04/01/16:25	2018/09/30/23:34
经度（度）	91	123
纬度（度）	9	33
闪电类型	0（地闪）	1（云闪）
强度（安培）	-250000	270000
传感器个数	5	12
回击次数	1	73

其中主要使用的是经纬度和强度数据，从地图上可知整个数据集覆盖的区域囊括了整个东南亚，并不只是局限在广东省，而且数据集是按发生时间排序的，以分钟为基本单位，同一时间可能有成百上千个闪电发生，而这些闪电可能并不属于同一个区域，给雷暴识别加大了难度。

4.2 参数取值与优先级预估

根据经验法和数据集中的数值范围并查找相关资料，得出 10 个参数的取值区间以及其对预测效果的影响力优先级预估结果如表 4-4 所示，优先级中的“1”代表最高优先级，“4”代表最低优先级。

表 4-4(a) 参数取值区间与优先级

参数名	取值区间	优先级
TimeDifference	180-600	1
ValidDistance	5-20	1
PredictDegree	1-10	2
BeforeTime	600-3600	2
MinNum	20-2000	3
RowInterval	300-3000	3
MinIntensity	0-3000	4

表 4-4(b) 参数取值区间与优先级

参数名	取值区间	优先级
MaxSpeed	0.02-0.03	4
ECRatio	1-10	4
MaxAnswerTimeNum	1-20	4

4.3 部分关键代码

4.3.1 单场雷暴识别

“自底向上”搜索合理数据：

```
while(l>0 and EarliestTime<=t):
    l2 = l
    # 后一个合理的数据点的时间
    t2 = datetime.strptime(data[names[0]][l2], "%Y/%m/%d %H:%M")
    # 寻找前一个合理的数据点，这中间可能有多个不合理的数据点，
    # 直到找到一个合理的才跳出循环
    while(l>0):
        l -= 1
        t = datetime.strptime(data[names[0]][l], "%Y/%m/%d %H:%M")
        if(EarliestTime>t): # 超出了最早时间，停止搜索
            break
        dis = Distance(data['LON'][l2], data['LAT'][l2], data['LON'][l], data['LAT'][l])
        timeDif = (t2-t).seconds
        if(dis<=vDis and timeDif < TimeDifference and
        abs(data['PEAKURRENT'][l])>=MinIntensity and
        (timeDif==0 or dis/timeDif<=MaxSpeed )):
            list.insert(0, l)  # 插入头部，按时间顺序排列
            break
```

计算两点距离：

```
def Distance(lng1,lat1,lng2,lat2): #根据经纬度计算两点距离，返回值单位是公里
    # 经纬度转换成弧度
    lng1, lat1, lng2, lat2 = map(radians, [float(lng1), float(lat1), float(lng2), float(lat2)])
    dlon = lng2 - lng1
    dlat = lat2 - lat1
    a = sin(dlat / 2) ** 2 + cos(lat1) * cos(lat2) * sin(dlon / 2) ** 2
    distance = 2 * asin(sqrt(a)) * 6371 * 1000   # 地球平均半径，6371km
    distance = round(distance / 1000, 3)
    return distance
```

随机生成颜色：

```
# 10 进制整数转换为 RGB 格式：“#”+6 位十六进制数的字符串
def Hex_to_RGB(colorNumber):
    r = hex(int(colorNumber/65536) % 256 )[2:]     # [2:] 是为了去掉 0x 两个字符
    g = hex(int(colorNumber/256)%256)[2:]
    b = hex(colorNumber%256)[2:]
    if(len(r)<2):     #   填充，保证 2 位数
        r='0'+r
    if (len(g) < 2):
        g='0'+g
    if (len(b) < 2):
        b='0'+b
    rgb = "#"+r+ g + b
    return rgb
```

折半查找时间和两种可视化方法的代码过长，这里不再贴出。

4.3.2 多场雷暴识别

每轮迭代开始时进行预判：

```
StartTime = datetime.strptime(data[names[0]][i], "%Y/%m/%d %H:%M")
EndTime = StartTime+timedelta(seconds = BeforeTime)
if(EndTime<datetime.strptime(data[names[0]][i+RowInterval], "%Y/%m/%d %H:%M")):
#   原始数据点少于 RowInterval
    i+=1
    continue
```

“自顶向下”搜索合理数据：

```
while(i<RowNum and Stop==0):
    lastTime = datetime.strptime(data[names[0]][i],
    "%Y/%m/%d %H:%M")
    l = i
    while(True):   # 找一个合理的数据点
        l = l + 1
        if (l == RowNum):
            Stop=1
            break
        NextTime = datetime.strptime(data[names[0]][l],
        "%Y/%m/%d %H:%M")
        if(NextTime>EndTime):
            Stop = 1
            break
        dis = Distance(data['LON'][l], data['LAT'][l], data['LON'][i], data['LAT'][i])
        timeDif = (NextTime - lastTime).seconds
        if (dis <= vDis and timeDif <= TimeDifference and
        abs(data['PEAKURRENT'][l]) >= MinIntensity) and
        (timeDif == 0 or dis / timeDif <= MaxSpeed)):
            list.append(l)   # 找到一个合理数据点
            i = l
            break
i=l
```

4.3.3 雷暴路径预测

经度预测核心代码：

```
# x = np.arange(1, dataNum + 1)
#   用于预测的 xp，在 x 结尾增加 predictNum 个位置
xp = np.arange(1, dataNum +1 + predictNum )
lon = self.SampleData[i][names[1]]
fun_lon   = np.polyfit(x, lon, degree)   #   拟合出多项式各次项的系数
poly_lon = np.poly1d(fun_lon)            #   生成预测多项式
lon_vals = poly_lon(xp)   # 拟合经度值值
lon_vals = lon_vals[dataNum:]            # 前面 dataNum 个是拟合出的值，这里没用到
```

样本数据集化简、预测数据集构造以及预测效果评估的代码过长，这里不再贴出。

4.4 雷暴识别实验

实验内容：选择 4 号原始数据集，运行单场雷暴识别算法，根据表 4-4 所估参数的范围进行调参，用两种可视化方法进行定性分析，得出单场雷暴识别图片和一套粗略的基准参数。

实验结果：

表 4-5 “线圈法”可视化的基准参数（有两个这里不需要用到）

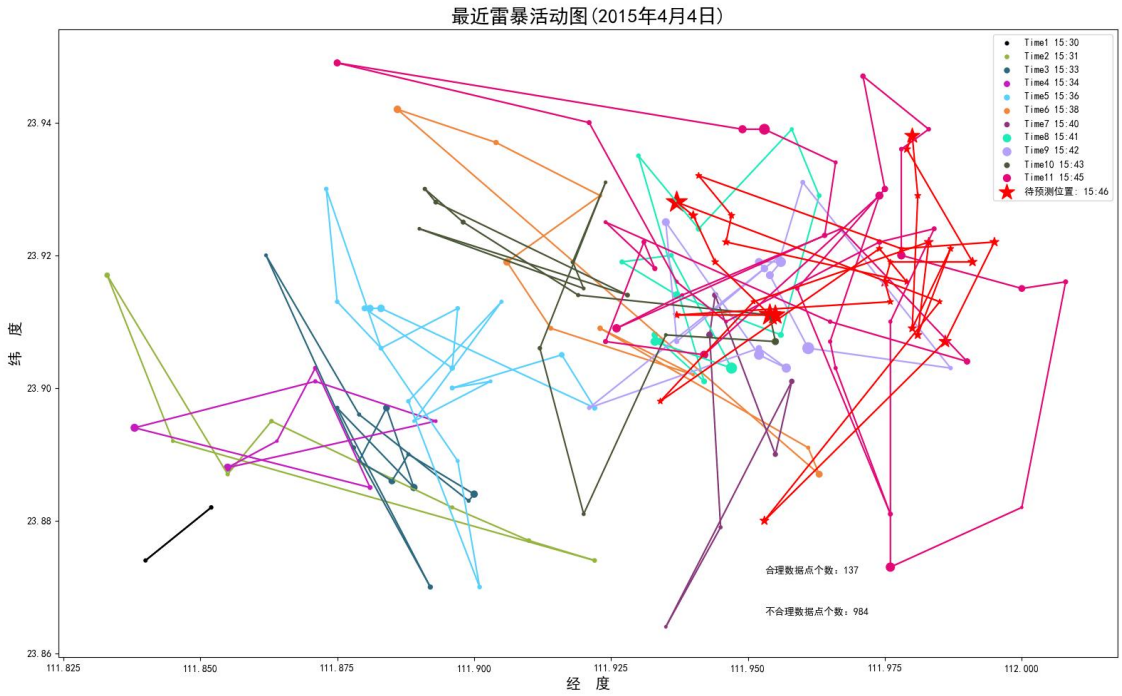
参数名	参数值	参数名	参数值
TimeDifference	180	RowInterval	500
ValidDistance	5	MinIntensity	1000
MinNum	30	MaxSpeed	0.03
BeforeTime	900	ECRatio	10

表 4-6 “中心路径法”可视化的基准参数（有两个这里不需要用到）

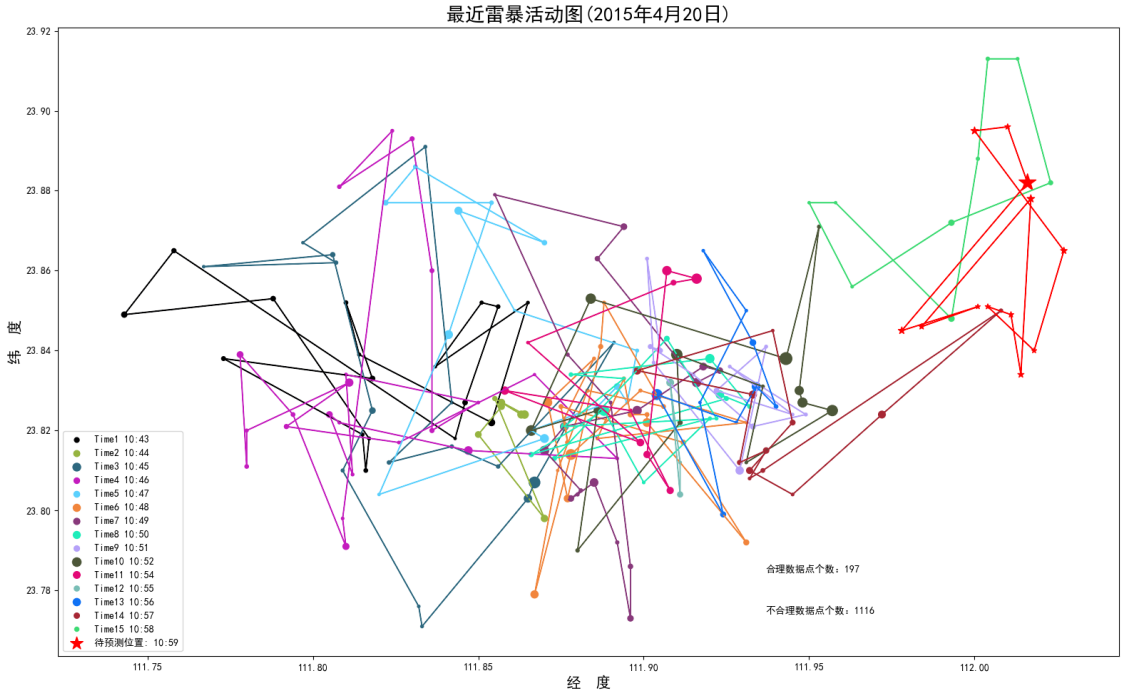
参数名	参数值	参数名	参数值
TimeDifference	300	RowInterval	1000
ValidDistance	9	MinIntensity	1500
MinNum	200	MaxSpeed	0.03
BeforeTime	1800	ECRatio	10

表 4-7 输入时间与对应的识别结果图

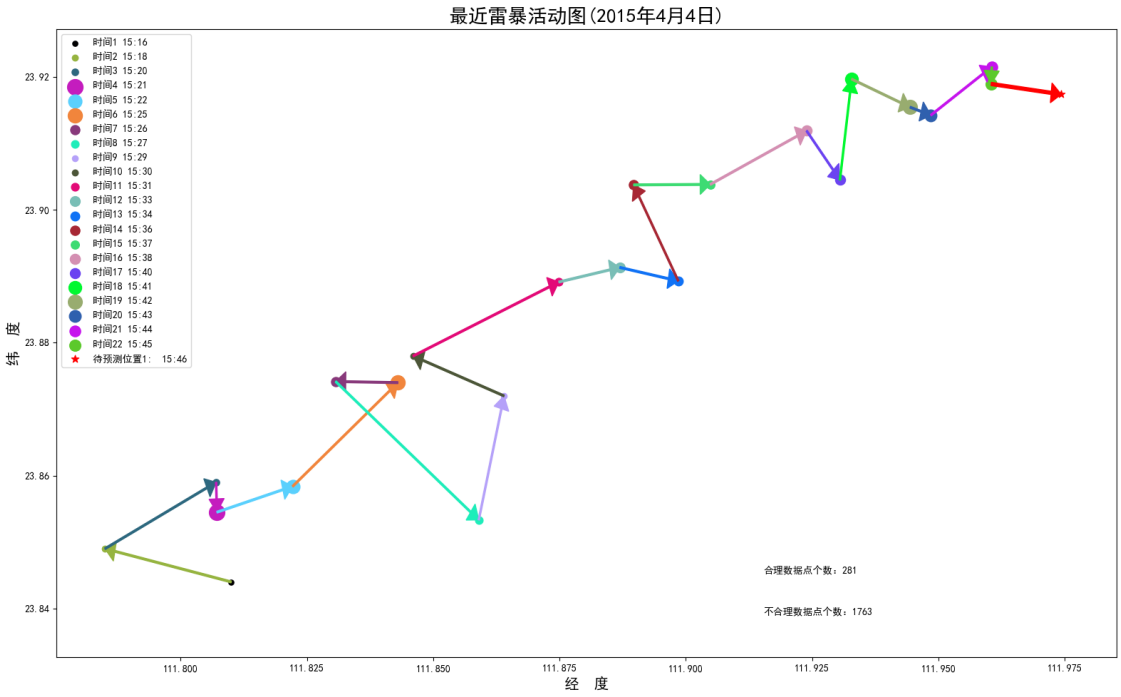
输入时间（年 月 日 分）	图像编号
2015 4 4 15 45	a、c
2015 4 20 10 58	b、d



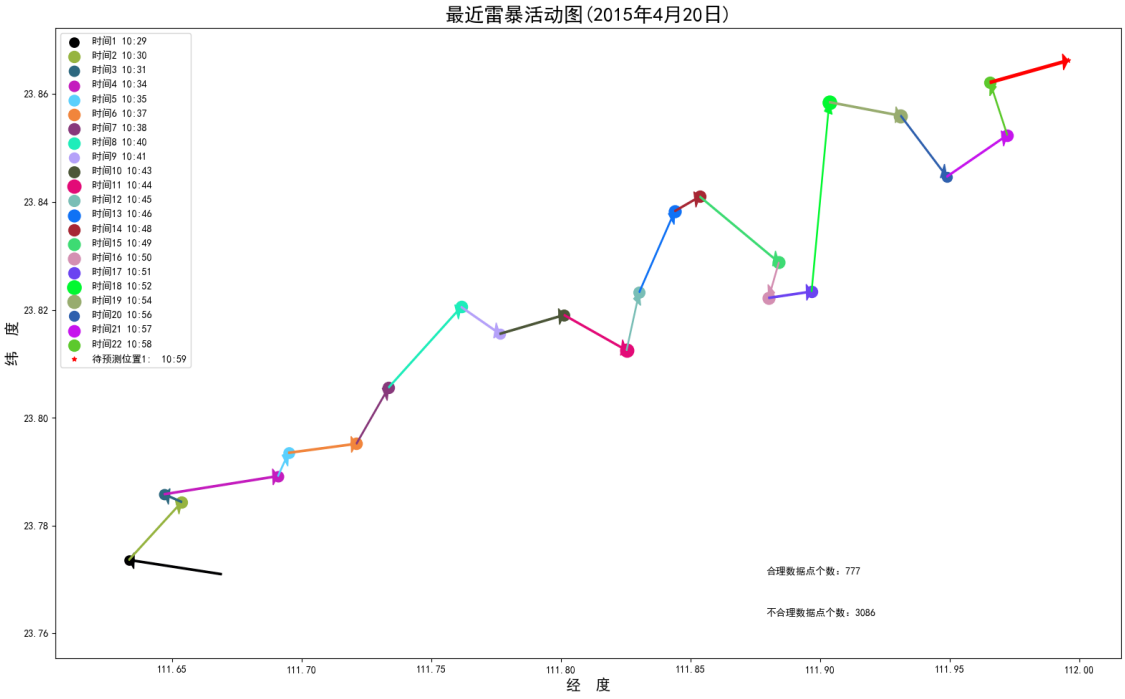
a) 单场雷暴识别之“线圈法”（一）



b) 单场雷暴识别之“线圈法”（二）



c) 单场雷暴识别之“中心路径法”（一）



d) 单场雷暴识别之“中心路径法”（二）

图 4-1 单场雷暴识别效果

图的横坐标是经度，纵坐标是维度，右下角给出了筛选这个雷暴样本时淘汰的不合理数据点个数和样本中现有的合理数据点个数，大红色代表下一时间雷暴的位置，也就是待预测的位置。

“线圈法”画出的图看上去很杂乱，但仔细看还是可以看出雷暴运动的大致趋势的。同一种颜色的线圈代表同一时间雷暴覆盖的区域，图例给出了每一种颜色代表的时间，可以看到图 a 和图 b 中的雷暴都是在向东移动。“中心路径法”取了每一时间发生的各个闪电的中心位置，然后连成带箭头的线段形成路径，可以清晰的看到每一时间雷暴的中心位置和当前的运动趋势，图 c 和图 d 中雷暴的运动趋势都是从西南往东北方向曲折移动，说明整体的识别效果还是不错的。

4.5 路径预测实验

实验内容：选择 1 号原始数据集，运行多场雷暴识别算法与路径预测算法，基于 3.6 节的调参方法、4.2 节估计的参数区间与优先级、4.4 节的基准参数进行调参，得出在单个数据集上的最优解，然后将之作用在 9 个数据集上以观测其泛化的能力。

4.5.1 初步模型

基于 4.4 节“中心路径法”中得出的基准参数，另外设置 MaxAnswerTimeNum = 10，PredictDegree = 3, 对 1 号原始数据集运行多场雷暴识别与路径预测算法，得到一个粗略的模型，相关结果如表 4-8 所示：

表 4-8 初步模型的实验结果

错误 1 个数	错误 2 个数	错误 3 个数	样本个数
1242	12	104	140
Score	AccuracyRate	MissRate	WrongRate
2.26	0.45	0.02	0.42
识别完毕时间	数据清洗完毕时间	预测完毕时间	评估完毕时间
760.0s	782.9s	787.0s	791.5s

上表中错误 1 表示样本集数据点不够，错误 2 表示答案集不合理，错误 3 表示数据筛选比太大。可以看到，识别失败的主要原因是数据点不够多。其中错误 2 只出现了 12 次，这是合理的，因为答案集的构造方法经 3.4 节的优化后基本可以避免找不到合理的答案数据的情况了。

上表中的 Score 和 AccuracyRate 值表明该模型只能以 42%的准确率预测未来 2.26 分钟的路径，可见预测效果几乎是微乎其微的，可能有的参数设置极其不合理或者数据集中存在违背预测算法中的理想假设的情况。

上表中最后的四个时间点表明算法耗时主要集中在雷暴识别部分，其次是数据清洗部分，二者的时间复杂度分别是 O(N)、O(kn)，预测和评估部分耗时很少，时间复杂度是 O(k), 这验证了第三章对算法复杂度的分析的正确性。

4.5.2 调参优化实验

调参实验：基于 1 号原始数据集不断调参测试，得到最优效果时的参数如表 4-9 所示，将该套参数应用到对 9 个数据集的样本提取与路径预测中进行泛化测试，最终的预测效果如表 4-10 所示，可见泛化能力还是不错的。

表 4-9 最优预测结果对应的参数

参数名	参数值	参数名	参数值
TimeDifference	360	RowInterval	2000
ValidDistance	12	MinIntensity	500
MinNum	1000	MaxSpeed	0.03
LastTime	3600	ECRatio	10
MaxAnswerTimeNum	30	PredictDegree	1

表 4-10 最优预测效果

数据集	样本个数	Score	AccuracyRate	MissRate	WrongRate
1 号	8	20.25	0.72	0	0.21
1-9 号	43	15.47	0.64	0	0.32

上表显示在 1 号数据集上以 72%的准确率预测了雷暴未来 20.25 分钟的路径，在 9 个数据集上以%的准确率预测了雷暴未来分钟的路径。1 号数据集中检测的样本只有 8 个，是因为参数的限制条件更加严格了；漏报率为 0 是因为预测总是有雷暴发生，于是没有漏报情况；误报率单指没有雷暴而预测有雷暴的情况，实际有雷暴、预测也有雷暴但位置预测错误的情况不计在列，所以准确率、漏报率与误报率的和不必等于 100%；误

报率这么大可能是因为有些时间点有雷暴发生但没有成功识别出任何一个合理的数据点，导致答案集中显示这一个点没有雷暴，从而以为产生了误报。基于这一点，提出了下面的优化。

优化实验：详细观察实验数据后发现样本集的数据并不是连续的，相邻的两个数据点可能相差不只 1 分钟，这与之前算法的假设相悖，导致时间序列转换成自然数序列后带来了很大的误差。基于此，让转换后的时间序列保持原有差距，即本来在时间上相差 t 分钟则转换后也相差 t 分钟。采用这一优化后得到了表 4-10 所示的预测效果，在保持了样本个数、准确率、漏报率、误报率基本不变的前提下，两种情况的预测时间都有了较小的延长。估计限制预测效果的瓶颈还可能是因为不同数据集的属性不一样，算法中的参数可能需要随着数据集的大小而修改。

表 4-11 优化时间序列转换后的预测效果

数据集	样本个数	Score	AccuracyRate	MissRate	WrongRate
1 号	8	20.88	0.74	0	0.21
1-9 号	43	15.88	0.65	0	0.32

最后，给出一个将样本数据集、答案数据集和预测数据集同时用“中心路径法”可视化的例子，如图 4-2 所示。红色表示未来雷暴的待预测路径，黑色表示预测的雷暴路径，右下角的合理数据点个数是化简后的数据点个数，所以都小于最大时间点数 60。

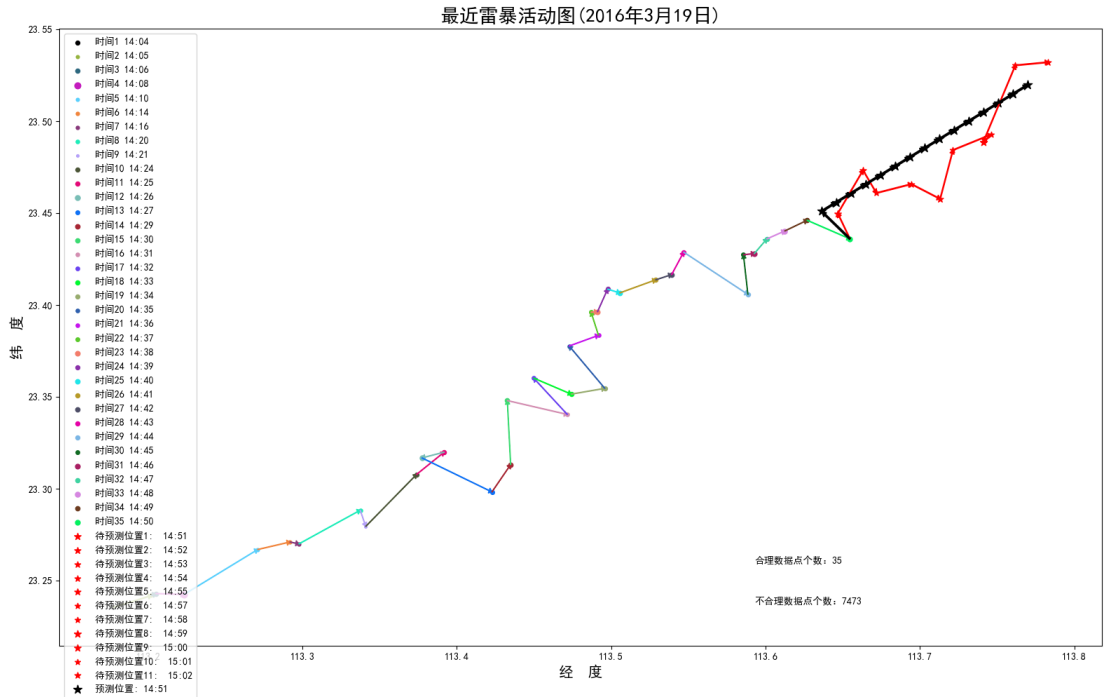


图 4-2 某个样本的预测结果图

结论

1. 论文工作总结

本文主要解决了从一个混乱数据集中识别、提取雷暴并进行雷暴路径预测与评估的问题。为此，提出了单场雷暴识别、多场雷暴识别、雷暴路径预测与评估三个算法，设计了 10 个与最终预测结果直接相关的参数，并从各种角度提出了一些优化方法。在最优情况下，平均以 64%的准确率预测了雷暴未来 15.47 分钟的路径。

2. 工作展望

接下来可以通过以下几点来进一步优化算法，有的本文未来得及采纳，有的实现了但未能通过实验验证。

1. 换用其他更高级的预测函数，比如 scipy.plotfit，scipy.stats.linregress，scipy.optimize.curve_fit，numpy.linalg.lstsq，statsmodels.regression.linear_model.OLS，sklearn.linear_model.LinearRegression 等回归函数，但有的函数需要有大量样本数据集；
2. 在路径预测算法的第一步进行数据清洗时可以根据强度的大小决定当前位置的经纬度对雷暴中心位置贡献的权重大小，即把该点的强度除以同一时间所有点的强度之和作为该点经纬度的权重；
3. 有些参数的设置可能与所在的数据集大小有关，比如 ECRatio、RowInterval、MinNum；也可能与数据集主要覆盖的区域有关，比如内陆与海洋上的 MaxSpeed 是不一样的。对每个数据集统一用相同大小的参数有失公允，可以尝试将之与数据集大小相绑定；
4. 在雷暴识别时是通过两个 while 循环来搜索样本数据集，内层的 while 是为了寻找到下一个合理的数据点，但是在不少情况下可能连续成千上百个数据都不合理，直到当前数据点的时间超过雷暴持续时间，这才发现本轮雷暴识别失败。可以通过设置不合理数据点上限（当内层循环迭代次数超过这个上限时认为后面没有合理数据点了）、判断与上一个合理数据点的时间差是否大于 TimeDifference（如果大于则后面的数据点都大于，都必然不合理）两种方法进

行剪枝，快速发现异常。这样一方面可以略微加快检索速度，一方面可以大大提高雷暴样本的识别成功率，从而为路径预测提供更多样本；

5. 雷暴识别时相邻两个合理数据点之间的距离不能超过 ValidDistance，这样的做法有点笼统。因为相邻两个合理数据点可能是在同一时间发生，它们之间的距离可能较小；可能是不在同一时间点发生，这时的距离较大，因为有一个雷暴云移动的过程。故可以令设一个变量 SvalidDistance 作为相同时间发生的两个数据点之间的距离的最大值，原来的 ValidDistance 作为不同时间发生的两个数据点之间的距离的最大值，后者应该略小于前者。

参考文献

[1] 陈家宏, 张勤, 冯万兴, 等. 中国电网雷电定位系统与雷电监测网[J]. 高电压技术, 2008,34(3):425-431.

[2] 高文胜, 张博文, 周瑞旭, 等. 基于雷电定位系统监测数据的雷暴云趋势预测[J]. 电网技术, 2015,39(2):523-529.

[3] 黄礼忠, 苏盛, 杨鑫, 等. 基于 LLS 的雷暴运动趋势临近预测[J]. 电磁避雷器, 2019,1:76-83.

[4] 刘学谦, 刘娟. 基于网格的雷暴识别与追踪技术[J]. 计算机工程与设计, 2015,36(1):254-257.

[5] 周康辉, 郑永光, 蓝渝. 基于闪电数据的雷暴识别、追踪与外推方法[J]. 应用气象学报, 2016,27(2):173-181.

[6] 徐高扬, 郑海涛, 黄国庆, 等. 基于门控单元循环神经网络的台风路径预测[J]. 计算机应用与软件, 2019,36(5):119-125.

[7] 徐高扬, 刘姚. LSTM 网络在台风路径预测中的应用[J]. 计算机与现代化, 2019,5:64-73.

[8] Xianlun Tang, Ziming Li, Minghui Xiang, Zexin Wu and Zhong Wang, "Lightning prediction method based on class-weighted dual v-support vector machine," 2010 8th World Congress on Intelligent Control and Automation, Jinan, 2010, pp. 4649-4653.

[9] Dixon M, Wiener G.TITAN: Thunderstorm Identification, Tracking, Analysis, and Nowcasting—A Radar-based Methodology[J].Journal of Atmospheric and Oceanic Technology,1993,10(6):785-797.

[10] Kohn M, Galanti E, Price C, et al. Nowcasting thunderstorms in the Mediterranean region using lightning data[J]. Atmospheric Research,2010,100(4).

[11] Chronis T G , Anagnostou E N . Evaluation of a long-range lightning detection network with receivers in Europe and Africa[J]. IEEE Transactions on Geoscience and Remote Sensing, 2006, 44(6):p.1504-1510.

[12] Lagouvardos K , Kotroni V , H.-D B , et al. A comparison of lightning data provided by

ZEUS and LINET networks over Western Europe[J]. Natural hazards and earth system sciences, 2009, 9(5):1713-1717.

[13] Ramzi, Mastura Mohd , et al. "Lightning Prediction Modelling Using MLPNN Structure. Case Study: Kuala Lumpur International Airport (KLIA)." 2018 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS) IEEE, 2018.

[14] 潘迪夫, 刘辉, 李燕飞. 基于时间序列分析和卡尔曼滤波算法的风电场风速预测优化模型[J]. 电网技术, 2008, 32(7):82-86

[15] 邱锡鹏.神经网络与深度学习[M].北京:机械工业出版社,2018:130-154

致谢

非常感谢汤德佑老师与陈靖宇师兄在毕业设计与毕业论文中对我的支持、指导、与帮助！

非常感谢我的家人一直以来对我的鼓励与肯定！