Optimization of stepwise clustering algorithm in backward trajectory analysis

Chunsheng Fang[1,2], Jialu Gao[1], Dali Wang[3], Diansheng Wang[1], Ju Wang[1,2]

[1]Environmental Science Department, Jilin University, Changchun 130012, China
[2]Key Laboratory of Groundwater Resources and Environment, Ministry of Education, Jilin University, Changchun 130021, China
[3]Oak Ridge National Laboratory, Oak Ridge, TN, USA

Abstract

In recent years, the backward trajectory model has been widely used in the research of meteorological and atmospheric environmental quality. This paper presents a comprehensive study on a stepwise clustering analysis algorithm in the clustering process of backward trajectory model and an application of the clustering analysis of single-particle backward trajectory in 2016 in Changchun City. This study starts with an analysis of the original stepwise clustering algorithm and its application to a clustering process of 8784 backward trajectories during 48 h in Changchun City as a benchmark test case. Then, two improvements are made in the algorithm: First, in the process of finding the optimal classification, the algorithm complexity is improved from original $O(n^3)$ to $O(\log(n)*n^2)$ through algorithm improvement. The algorithm performance is enhanced by $\log(n)$ times. Second, in the process of re-establishing the classification, the algorithm complexity is improved from the original $O(m*n^2)$ to $O(m*\log(n)*n)$, that is another algorithm performance improvement by a factor of $\log(n)$. Therefore, the accumulative execution efficiency improvement through the algorithm optimization is $2*\log(n)$ times, which has been further verified in the practical application in Changchun City.

1. Introduction

The backward trajectory model is one of the main research methods in the study of the fate of atmospheric pollutants which is called HYbrid Single Particle Lagrangian Integrated Trajectory model (HYSPLIT) and was developed jointly by the Air Resources Laboratory of NOAA and the Meteorological Bureau of Australia, based on the monitoring data of the global multi-year meteorological satellite. It is a professional model for calculating and analyzing the transport and diffusion trajectories of atmospheric pollutants. This model is widely used in the related research of meteorological and atmospheric environmental quality.

Li et al. [1] studied the transport pathways of particulate matter in Shanghai during the period of severe haze in the Middle East of China in 2013. Zhao et al. [2] also used the backward trajectory model to study the distribution and transmission characteristics of $PM_{2.5}$ sources in Shanghai. Shanshan et al. [3] studied the source distribution and transmission characteristics of $PM_{2.5}$ in Beijing by using the backward trajectory model. Xiaoyu et al. [4] used the backward trajectory model and cluster analysis method, combined with the mass concentration data of atmospheric particulate matter in Yinchuan City of China, to analyze the characteristics of year-round and seasonal airflow trajectories and their effects on the concentration of particulate matter in Yinchuan. Yassin et al. [5] used the HYSPLIT model to simulate the trajectory and assessed the sources of dust storms in the State of Kuwait. Koracin [6] used Lagrangian stochastic particle dispersion and HYSPLIT backward trajectory model for regional source identification in the Eastern United States. When analyzing the haze distribution characteristics in different regions of Zhejiang Province, Weng et al. [7] combined the stepwise clustering method to objectively classify 848 haze processes and discussed the atmospheric circulation and air transport characteristics of different spatial distribution haze processes. Wang et al. [8] developed a

statistical downscaling tool (SCADS) to assist obtaining high-resolution climate change scenarios, based on the stepwise cluster analysis method. This paper improves the efficiency of model calculation through algorithm optimization. Under the premise of ensuring accuracy, the efficiency of clustering analysis processing is greatly improved.

2. Materials and method

2.1. Sources of data

All $PM_{2.5}$ concentration data in this article come from the official website of Changchun Environmental Monitoring Center Station. The backward trajectory model uses data from the Global Data Assimilation System (GDAS) provided by the US EPA from 2015 to 2016.

2.2. Single-particle backward trajectory model

The single-particle backward trajectory model is based on the HYSPLIT 4 version developed by the United States Environmental Protection Agency of the US National Oceanic and Atmospheric Research Center's Air Resources Laboratory (NOAA). This model is an Euler and Lagrangian hybrid diffusion model. In the model, the Lagrangian method is used for the advection and the diffusion part, and the Euler method is used for the concentration calculation. The model is considered to be relatively perfect in the process of transportation, diffusion, and settlement of pollutants, and the minimum time accuracy of simulation is 1 h. In this article, the Changchun Children's Park monitoring point is selected as the starting point of the backward trajectory, the starting height is 200 m, and the calculation time is from January 1, 2016, from 0:00 to December 31, 2016, at 23:00. A backward trajectory was simulated every hour, and the calculation time is 48 h. Thus, a total of 8784 initial back trajectories were obtained.

2.3. Stepwise cluster analysis

This paper adopts stepwise clustering algorithm to analyze the type II model, namely hill climbing method for calculation. The stepwise clustering algorithm is described as follows.

Presume that $X[m]$ is the target array with m elements to be divided into n clusters, then define:

$$E_i = \sum_{i=1}^{n_i} (X_i - \overline{X}_i), \ = \sum_{i=1}^{n} n_i$$

where E_i is the ith group clustering error, n_i is the total element number of ith group, X_i is the ith element of n_i group, and $\overline{X}_i$ is the average value of n_i group.

Thus, the stepwise clustering algorithm is to calculate the minimum value of E:

$$E = \sum_{i=1}^{n} E_i$$

where E is the minimum total clustering error in n groups.

The stepwise clustering algorithm is divided into the following steps:

(1) Select the clustering factor. The clustering factor in this paper is the latitude and longitude coordinates of the backward 48-h orbit of each single particle;

(2) Define the classification objective function E_i as the sum of the distances between all trajectories and the average trajectory of their class;

(3) The initial classification is given that all 8784 trajectories are divided into one category; at this time $E_i = 0$;

(4) Incorporate each category in turn into all other categories and recalculate E_i to find the category with the smallest E value;

(5) The new classification is used as the initial classification, and steps (4)–(5) are repeated until all the trajectories are classified as group number 1. At this time, the classification error E is the maximum value;

(6) According to the E value of each step in the stepwise clustering process, the optimal classification number is determined and the classification result is given.

3. Algorithms and optimization

3.1. The original algorithm

According to the selected climbing-type stepwise clustering analysis model, the clustering object in this paper is a plane backward trajectory map for the whole year, and the outline of forming a stepwise clustering algorithm is as follows.

```
[BEGIN]
TYPE DEFINE PointInfo              ' Definition information of a point in plane space
  PointNumber AS INTEGER           ' The sequence number of the point
  Lang AS SINGLE                   ' The longitude of the point
  Lati AS SINGLE                   ' The latitude of the point
END OF TYPE DEFINE

TYPE DEFINE TrajLineInfo           ' Definition information of backward 48-hour single particle
  trajectory
  LineNumber AS INTEGER            ' The sequence number of each backward trajectory
  Points(49) AS PointInfo          ' Information of 49 nodes in each backward trajectory.
END OF TYPE DEFINE

DIM maxTrajNumber AS INTEGER
DIM Traj[maxTrajNumber] AS TrajLineInfo
INPUT maxTrajNumber
REDIM Traj[maxTrajNumber]
INPUT Traj[maxTrajNumber] FROM DATAFILE

DIM sCAveE(maxTrajNumber) AS TrajLineInfo          ' Stores the average trajectory curve for
each class
DIM iCurCN AS INTEGER              ' The number of clusters that have been completed so far
iCurCN= maxTrajNumber
DIM fDis AS FLOAT
DIM k,l AS INTEGER
DO WHILE iCurCN > 1
    fDis=MAXFLOAT
    k=1
    l=1
    FOR i=1 TO iCurCN-1
        FOR j=i+1 TO iCurCN
            ADD j cluster TO i cluster
            RECALC sCAveE(i)
            fDisOfI=TOTAL CLUSTER ERROR          ' Calculate the total error of the current
classification
```

```
                IF fDisOfI < fDis THEN
                    k=i
                    l=j
                    fDis= fDisOfI
                END OF IF
                DELETE j cluster FROM i cluster
                RECALC sCAveE(i)
            NEXT j
        NEXT i
        ADD l cluster TO k cluster
        REDIM sCAveE(iCurCN-1)
        RECALC sCAveE(k)                ' Recalculate the average trajectory of class k
        SAVE iCurCN, ALL CLUSTERS, fDis TO DATAFILE
        iCurCN= iCurCN-1
    LOOP OF DO WHILE
    [END]
```

After analysis, the complexity of this prototype algorithm is O(m^2*n^3), where m is the number of nodes per track and n is the total number of backward tracks. In this example, m = 49, n = 8784. The calculated number of loops reaches 1.627×10^{15} times. In addition, the calculation of the distance between two points in each cycle calculation is based on the absolute value of the latitude and longitude difference and the sum of squares. It is a floating-point operation. Therefore, the operation speed of the clustering program is greatly influenced by m and n. The calculation process in this example was performed on a well-configured personal computer (16G memory, CPU for Intel i7-4770@3.4 GHz, and Windows 7 64-bit system). The entire clustering process took 126 h. Figure 1 shows the relationship between the number of clusters and the total clustering error. In the figure, the abscissa is the number of classifications, and the ordinate is the

total classification error. Figure 2 shows the average trajectory clustering results of backward trajectories with the final clustering number of 8.

3.2. Algorithm improvements

In the above prototype algorithm, it is first assumed that the 8784 trajectories are, respectively, classified into one category, so that the average trajectory of each category is the only trajectory in the category; that is, the total classification error is 0. The principle of choosing the optimal clustering method each time is to find two types of k and l in the existing clusters so that after the two types are merged, the total clustering error of this method reaches the minimum among all the other total clustering errors after the two-by-two matching merger, so that the kth and lth class mergers are selected. This cycle continues until all trajectories are merged into the existing category; that is, when the total number of classifications is 1, the clustering is ended.

A careful analysis of the above clustering methods reveals that there are two processes for the most time-consuming steps:

(1) The process of finding the optimal classification. This is a triple-loop nested calculation process. In this process, we need to try to merge the current cluster with other categories to get new classifications, calculate its overall classification error, and then calculate the smallest overall classification error in these classification methods. This classification method is determined as the final classification of this round.

(2) The process of re-establishing classification. After determining the two categories of consolidation, the total number of categories is reduced by 1. Combining classification of k and l, the memory is redistributed to record the parameters of the new classification

track and the average track in each new classification. This process is a double loop performed in the outermost loop.

Based on the above two most time-consuming calculation steps, we propose to create an index array, optimize the original algorithm, and improve computing performance. The implementation is shown in the following algorithm procedure.

```
[BEGIN]
……                    ' Type definitions and variable definitions are unchanged, the following changes
DIM bCIArray(maxTrajNumber, maxTrajNumber) AS BOOLEAN          ' Add an index array

iCurCN= maxTrajNumber
DIM fDis AS FLOAT
DIM k,l AS INTEGER
DO WHILE iCurCN > 1
    fDis=MAXFLOAT
    k=1
    l=1
    FOR i=1 TO iCurCN-1
        FOR j=i+1 TO iCurCN
            IF (bCIArrey(i,1 TO iCurCN) OR bCIArrey(j, 1 TO iCurCN)) IS TRUE
                CALC sCAveE(i AND j)
            ENDIF
            fDisOfI=TOTAL CLUSTER ERROR          ' Calculate the total error of the current
classification
            IF fDisOfI < fDis THEN
                k=i
                l=j
                fDis= fDisOfI
            END OF IF
        NEXT j
    NEXT i
    ADD l cluster TO k cluster
```

```
        COMBINE bClArrey(k,1 TO iCurCN) AND bClArrey(l, 1 TO iCurCN)
        SAVE iCurCN, ALL CLUSTERS, fDis TO DATAFILE
        iCurCN= iCurCN-1
    LOOP OF DO WHILE
    [END]
```

In the optimization algorithm, we added an index array to the two time-consuming operations in the original algorithm to record the trajectory code contained in each category. Since this array is a Boolean variable, it will greatly save the machine time when judging and storing.

In the first process of searching for the optimal classification, it is no longer necessary to actually merge the two classes through the algorithm improvement. Instead, the new class is formed by judging the Boolean variables in the two class indexed arrays. After this operation, the original algorithm complexity can be increased from $O(n^3)$ to $O(\log(n)*n^2)$, and the efficiency can be improved by $\log(n)$ times.

In the second process of re-establishing the classification, when the two clusters to be merged are determined by searching, only the index arrays need to be valued and merged. There is no need to reallocate the memory, nor does it require repeated assignments to the categorical array. In particular, it saves a lot of floating-point operations in subsequent matrix processing. These operations increase the original algorithm complexity of this part from $O(m*n^2)$ to $O(m*\log(n)*n)$ and increase $\log(n)$ times efficiency.

Since the optimization of these two steps will greatly improve the efficiency of the algorithm execution, it will exceed $2*\log(n)$ times as a whole. However, due to the increase in the related operations of the index array, it will slow down to a certain extent, causing negative effects. It can be stated that during the calculation process, the computer task manager shows that the

memory footprint is about 3.5G when using the original clustering algorithm, and the improved clustering algorithm has a memory footprint of about 4.3G. When the program is not running, the memory usage of the computer is about 2G. But for today's computer configurations, the impact here is almost negligible. The impact, therefore, considers the complexity of the optimization algorithm to increase the computational performance by approximately 2*log(n) times.

The result was verified during the actual operation of the program. The calculation task was performed according to the final optimization algorithm. The execution time of the program from the same computer above is 5 h and 20 min, and the efficiency is increased by 23.7 times, which is basically the same as the estimated 2*log(8784) = 26.2 times. It also proves that the algorithm complexity assessment done in this study is correct. In order to verify and evaluate the reliability of the optimization efficiency improvement, we adopted the 8760 backward trajectories of Songyuan City and Jilin City in 2017, repeated the above steps, and carried out calculations. The results obtained are consistent with the above conclusions.

4. Conclusion

In this paper, we studied the clustering analysis of atmospheric hourly single-particle backward trajectories in 2016 in Changchun by using the hill climbing method model in stepwise cluster analysis. Through the process of clustering of 8784 backward trajectories during 48 h from 0:00 on January 1, 2016, to 23:00 on December 31, 2016, in Changchun City, an initial algorithm model was formed by understanding and analyzing the original algorithm. And a calculation program was written to obtain the clustering results. Then, two improvements are made to the algorithm. First, in the process of finding the optimal classification, the original algorithm complexity is improved from O(n^3) to O(log(n)*n^2) through algorithm improvement. The

execution efficiency is improved by log(n) times. Second, in the process of re-establishing the classification, the original algorithm complexity is improved from the original O(m*n^2) to O(m*log(n)*n) through algorithm improvement. The execution efficiency of the program is increased by log(n) times. Considering comprehensively, the execution efficiency improved by the algorithm optimization is 2*log(n) times. This model analysis result has been verified in practical application that the algorithm optimization improves the execution efficiency but will not affect the final clustering result. The improvement in the implementation efficiency of the algorithm in the practical application process provides technical support and practical tools for the application of backward trajectory clustering methods in environmental science and other fields.

Acknowledgements

This study is supported by Jilin Provincial Science and Technology Department (No. 20130204051SF) and Jilin Provincial Environment Protection Department (No. 2013EP01-03). And we thank the Changchun Central Environmental Monitoring Station for assistance regarding the monitoring data and sampling.

References

1. Li Li, Junlin Cai, Min Zhou (2015) Potential source contribution Analysis of the particulate matters in Shanghai during the heavy haze episode in eastern and middle China transport route and area of period of severe in December 2013. Environ Sci 36(7):2327–2336

2. Zhao QB, Ming HU, Zhang YH (2014) Study of source distribution and transportation characteristics of PM2.5 in Shanghai using backward trajectory model. Adm Tech Environ Monit 26(4):22–26

3. Shanshan Kong, Houfeng Liu, Yizhen Chen (2017) Discussion on the source distribution and transmission characteristics of PM2.5 in Beijing based on backward trajectory model. Chin J Environ Manag 9(1):86–90

4. Yan XY, Gou XH, Wu WL et al (2018) Analysis of atmospheric particulate transport path and potential source area in Yinchuan. Acta Sci Circumst 38(5):1727–1738

5. Yassin Mohamed F, Almutairi Sarah K, Al-Hemoud Ali (2018) Dust storms backward Trajectories' and source identification over Kuwait. Atmos Res 212:158–171

6. Koracin D, Vellore R, Lowenthal DH et al (2011) Regional source identification using Lagrangian stochastic particle dispersion and HYSPLIT backward-trajectory models. J Air Waste Manag Assoc 61(6):660–672

7. Weng Z, Li Liping, Yang W et al (2016) Backward trajectory and circulation features in different haze processes in winter of Zhejiang. Meteorol Mon 42(2):183–191

8. Wang X, Huang G, Lin Q et al (2013) A stepwise cluster analysis approach for downscaled climate projection—a Canadian case study. Environ Model Softw 49(49):141–151

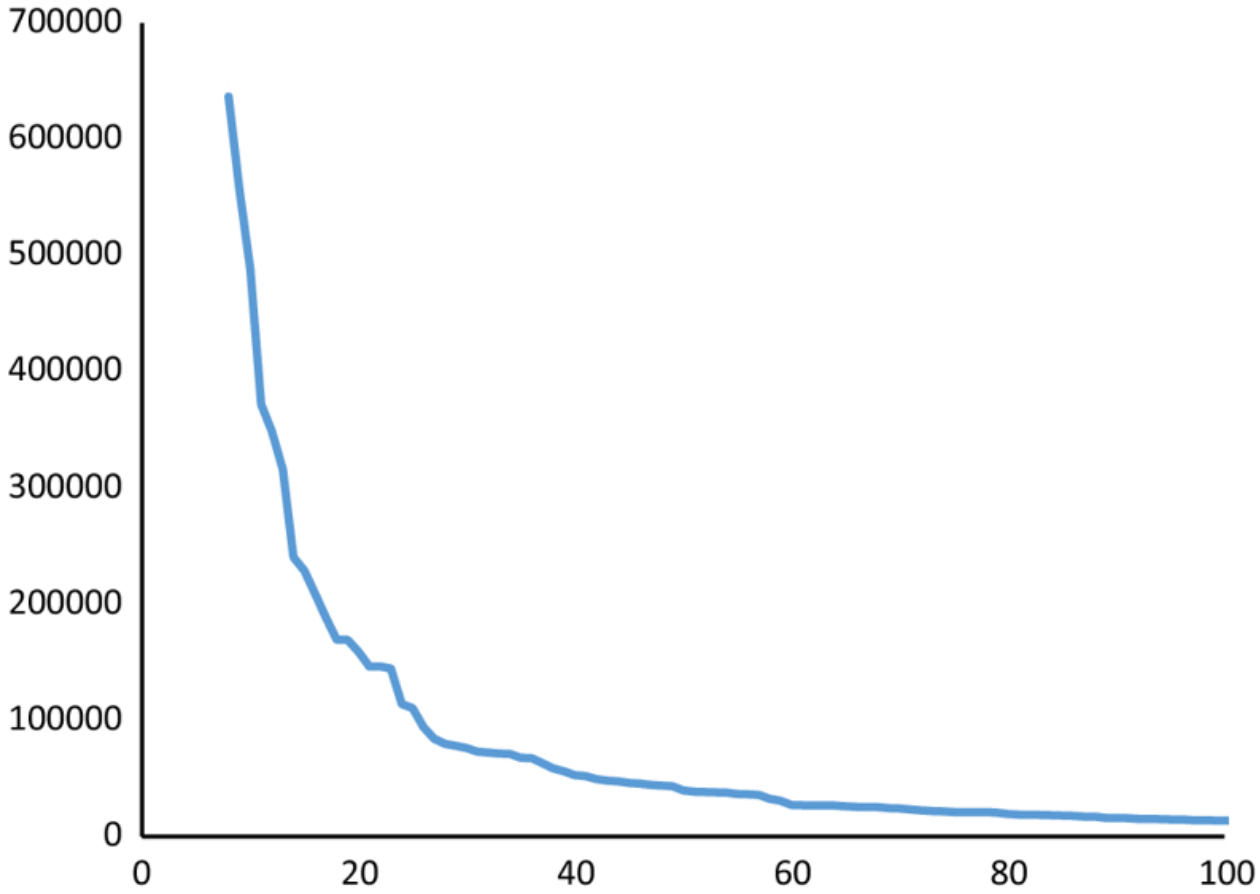

Figure 1. Curve of the total clustering error of the hourly backward 48-h trajectories in 2016 in Changchun City

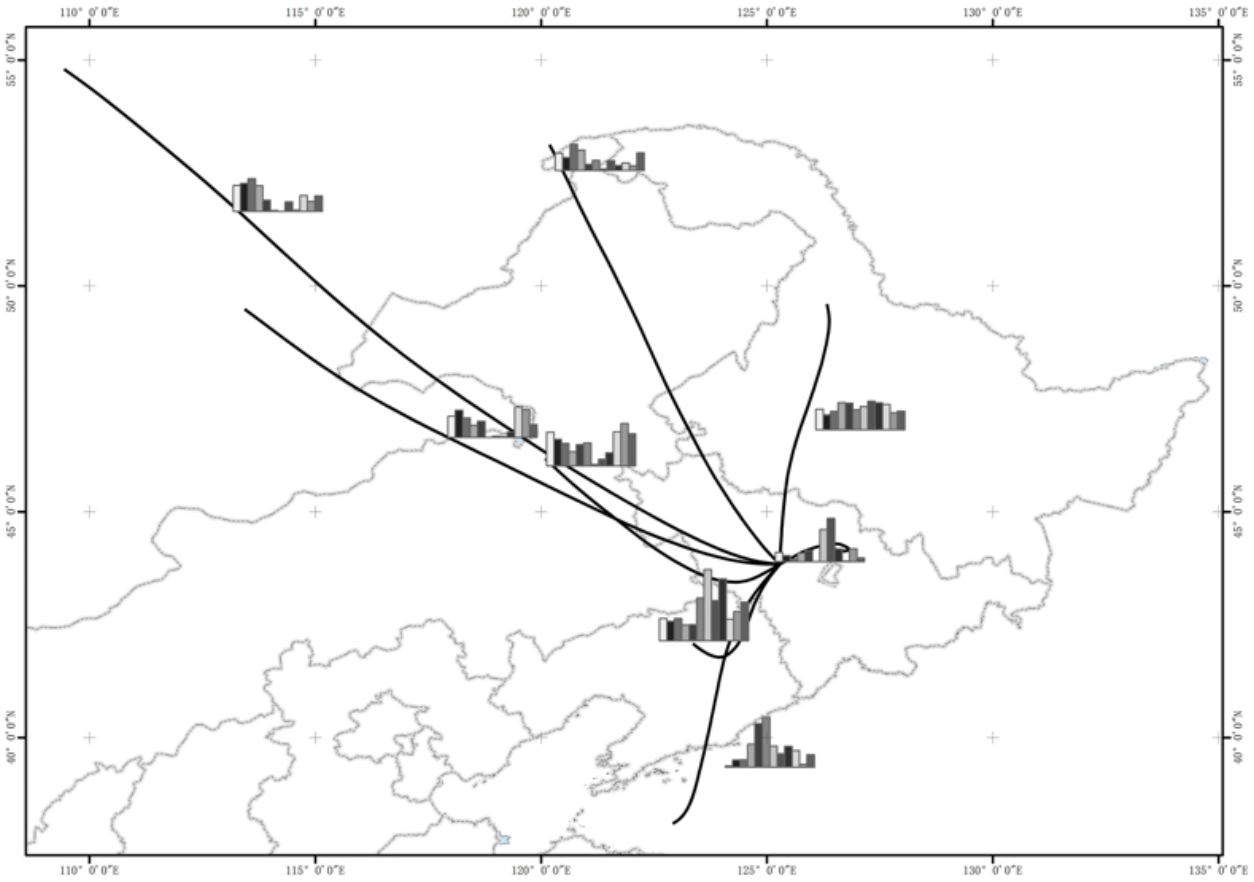

Figure 2. Clustering results of the hourly backward 48 h trajectory in 2016 in Changchun City