Project synthesis report
Application of Tensorflow, K-nearest neighbor and DBSCAN algorithms to particle laden turbulence

Benoit GILES, Anis BNINI, Yahia BRAIHEMAT and Mamoudou KOUME[1]

1 **Aix-Marseille Université, France,** `mamoudou.koume@etu.univ-amu.fr`

Abstract

Small-scale turbulence-cloud interactions focus on the interaction with particles at lower scales in a cloud. Cloud droplets in space can determine the interactions between the cloud and its microphysical properties. In order to determine these interactions, two key physical parameters were introduced in the study. These parameters are the Stokes number and the Reynolds number. To get there, a three-dimensional direct numerical simulation (DNS) of particle-laden isotropic turbulence is performed to obtain turbulent clustering data. We have applied three methods which are Tensorflow, K-nearest neighbor and DBSCAN algorithms in order to account for the turbulent interactions corresponding to different Reynolds numbers and Stokes numbers by highlighting the dependence between these two physical parameters.

Key words : DBSCAN, Tensorflow, KNN, Stokes number, Reynolds number, clustering, cluster, machine learning, particle-laden turbulence

1 Introduction

«Clouds play a crucial role in the heat and water systems of the Earth. To improve our understanding of cloud physics, a large number of observational studies have been conducted to estimate the spatial distribution of microphysical properties of clouds, such as cloud water mixing ratio and the effective droplet radius. Radar is one of the most powerful tools because it can provide two-or–dimensional estimates of the microphysical properties of clouds over a large domain [1].» This statistical study carried out as part of a research project whose general objective is to initiate research activities through which students are placed under the responsibility of academic tutors. The main goal of this project is to use machine learning methods such as Tensorflow, KNN and DBSCAN in order to study the microphysical properties of particle-laden turbulence as well as the influence of two physical parameters (Reynolds number and Stokes number) in their interactions. The first characterizes the turbulence intensity of the flow and the second characterizes the behavior of a particle in a fluid. Both are dimensionless numbers. To apply these methods, a three-dimensional direct numerical simulation of particle-laden isotropic turbulence is used by solving the Navier-Stokes equations on supercomputers. The data have been kindly provided by Dr. Keigo Matsuda [2] through the Mesocenter.

2 Methodology

This study was carried out using three machine learning methods.
First, we have applied the KNN algorithm which is a supervised classification. It takes a bunch of labeled points and uses them to learn how to label other points. The KNN algorithm allow to classify particle laden-turbulence according to their characteristics. We

can therefore choose either the different Reynolds numbers or the different Stokes numbers. The input data will be $1D$ arrays of our initial data, and the result will be either the Stokes number or the Reynolds number associated with Table $1D$.

Then DBSCAN algorithm was used to do the clustering. It integrates a notion of cluster based on density that makes it possible to discover clusters of arbitrary form. Indeed, DBSCAN being an unsupervised learning algorithm, it allows to appreciate the physical properties of droplets such as their interactions as well as the effect that the two physical parameters (Reynolds number and Stokes number) can have on these particles. In addition, the choice of this algorithm was also made from a practical point of view because the parameters which are necessary for it on the input side can be related to physical parameters (eg. distances in space or in the plane depending on whether data is represented in two- or three-dimensional space).

Finally, we applied to the data Tensorflow which is an open source library developed by Google that allows user to perform various machine learning tasks, it is often combined with the framework Keras which is an application programming interface (API). The main goal is to build a model that allows us to classify an image given as an input into its accordingly Stokes number and Reynolds number.

3 Results

Using the KNN algorithm, we started by grouping our samples based on their Reynolds number. After a first test, we obtained an accuracy of about 60%. In order to gain in precision, we deemed it necessary to change the distribution of training/test data. Despite this, the accuracy increased only slightly. Subsequently, we performed a data preprocessing : instead of taking our initial data, made up of more than 70,000 particles, we separated the input graph initially into rectangles and count the number of particles inside each rectangle. By using the modified square images as input, the precision is reduced from 60% to 100% for the model trying to determine the Stokes number associated with a sample flow by data preprocessing. However, when we use do the classification using as input the Stokes numbers, the best accuracy we could get was close to 30%.

In order to apply the DBSCAN algorithm, we first searched for the hyperparameters. To calculate the epsilon parameter we have to determine the knee on the k-distance graph using the nearest neighbor method. The second parameter was set to 5 for all files. In addition, the number of clusters obtained decreases with the number of stokes for a fixed Reynolds number but also it is higher for the largest largest Reynolds number. Also, the percentage of noise for all the parameters considered does not exceed 10% and decreases as a function of Stokes. With regard to the dimension of the clusters, there is a certain growth compared to the Stokes number but conversely the larger the Reynolds number corresponds to smaller dimensions. Finally, the algorithm presents less performance with regard to the silhouette coefficient which is found to be very low for most clusters.

With the tensorflow method, we have tried to build several neural networks with the integrated high level API of Keras. We set the lot size to 16 using 70% of the images for training and 30% for validation. After running the algorithm we get a correct prediction but with an estimated lower confidence of 57.20%. The model trained for the classification of images based on the Stokes number achieved an accuracy rate of 79% and passed the prediction test despite its a little too stubborn.

4 Conclusion

Through this research project on particle-laden turbulence, we applied three machine learning methods (Tensorflow, KNN and DBSCAN) in order to form clusters but also to be able to analyze the dependence that exists between the Reynolds number and the Stokes number in the interactions between the polydisperse cloud droplets. These particles in turbulence are from a three-dimensional direct numerical simulation of particle-laden isotropic turbulence [2].

We obtained some results when clustering the data using both KMeans and KNN algorithms. Indeed KMeans was useful when trying to partition particle in one sample flow but but we could not get very precise information about the clusters created via the different methods tested. However, using KNN it allowed us to determine which Reynolds number was associated to a fluid. After the data training, we reached 100% accuracy when we introduced the test data. The DBSCAN algorithm also allowed us to have convincing results. After using KNN for computing optimal epsilon values and having formed the clusters, we focused on the aspects of these clusters in particular their size, dimension, percentage of noise as well as their homogeneity and separation. The main goal is to see how the Stokes number and the Reynolds number vary according to these properties. What we can draw from this analysis is that the formation of clusters is closely related to these two physical parameters. Nevertheless, DBSCAN algorithm presents some limits at the level of the silhouette coefficient which presents low values thus translating a low homogeneity of the clusters independently of the physical parameters. Finally in the last chapter which deals with the tensorflow method, we have implemented an image classification algorithm which predicts the Stokes number and Reynolds number of particle-laden turbulence as a function of the image of their distribution, we also tried to generate synthetic spatial data of charged particles in turbulence with Tensorflow. In addition, after training the GAN model, we generated artificial data which retained most of the properties of the original data, but due to a precision of only 70%.

References

[1] Keigo Matsuda and Ryo Onishi. “Turbulent enhancement of radar reflectivity factor for polydisperse cloud droplets”. In: (2019). URL: https://acp.copernicus.org/articles/19/1785/2019/acp-19-1785-2019.html.

[2] MATSUDA and al. *Influence of microscale turbulent droplet clustering on radar cloud observations.* Atmos. Sci.71(10), 3569–3582., 2014.

[3] Benjamin DEVEZE and Matthieu FOUQUIN. *DATAMINING C4.5 - DBSCAN.* 2004.

[4] Statistical tools for high-throughput data analysis. “DBSCAN: density-based clustering for discovering clusters in large datasets with noise - Unsupervised Machine Learning”. In: (). URL: http://www.sthda.com/english/wiki/wiki.php?id_contents=7940.

[5] Martin Ester and al. *A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.* Institute for Computer Science, University of Munich, 1996.

[6] David Richter and al. “Rayleigh-Benard turbulence modified by two-way coupled inertial, nonisothermal particles, Physical Review Fluids”. In: (2018).

[7] https://penseeartificielle.fr/clustering-avec-lalgorithme dbscan/. “Clustering avec l’algorithme DBSCAN”. In: (2019).