Implementation of sampling techniques for initial geometry prediction in heavy ion collision experiment

Abhisek Saha[1],* Debasis Dan[2], and Soma Sanyal[1]
[1]*School of Physics, University of Hyderabad, Gachibowli, Hyderabad, 500046 India and*
[2]*Microsoft India (R & D) Pvt. Ltd. Microsoft Campus, ISB Road, Gachibowli, Hyderabad, 500032 India*

Abstract

We show that three crucial features that determine the initial geometry of heavy ion collision (HIC) experiments can be predicted with excellent accuracy by employing supervised machine learning (ML) techniques. Despite the use of several ML techniques in the past, the prediction accuracy is hugely centrality dependent. Detailed parameter scans and ablation analyses are used to analyse the error spectrum. We use different sampling methods to determine an efficient algorithm that provides a multi-fold improvement in the accuracy of ML model prediction. We discuss how the errors can be minimized, and the accuracy can be improved to a great extent in all the ranges of impact parameter and eccentricity predictions.

Introduction

The final particle spectra are significantly influenced by the collision centrality. It has been observed that the impact parameter (b), a representation of the collision centrality, influences the multiplicity distribution of the several identified particles. Although the centrality cannot be determined from experiments directly, it can be computed with the use of theoretical modeling using the Glauber model or another similar model. Neural networks have also been proposed to determine the impact parameter from the experimental data. We can automate the entire procedure and determine the impact parameter efficiently by using ML models. Utilizing ML has the advantage of making the process more agile by requiring less processing power and time. In this article, we examine different ML algorithms and present a thorough comparison of their efficiency and accuracy using well-defined ML methodologies to highlight a significant difference in the prediction accuracy for central collisions.

*Electronic address: `17phph12@uohydac.in`

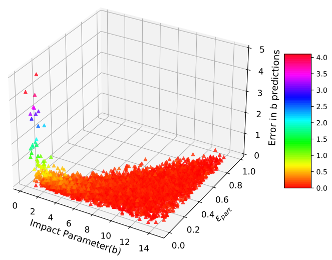

FIG. 1: Error distribution in the prediction of b for 200 GeV Au-Au events using kNN model

Eccentricity (ϵ_2) is another crucial parameter which gives us the initial geometrical shape of the collision region. The participant plane eccentricity is defined by, $\epsilon_{part} = \frac{\sqrt{\sigma_y^2 - \sigma_x^2 + 4\sigma_{xy}^2}}{\sigma_y^2 + \sigma_x^2}$, here σ's are the variances of the positions of the particles. For the ML model training, the transverse momentum (p_T) spectra are taken as input features and the impact parameter, ϵ, ϵ_{part} are taken as the target variable which the model must predict. We have used a multiphase transport (AMPT) model to generate the p_T spectra of Au-Au collision events at 200 GeV collision energy ($\sqrt{s_{NN}}$) [1].

Results and Discussions

All the standard ML models perform reasonably well in case of impact parameter predictions giving more than 90% accuracy. The

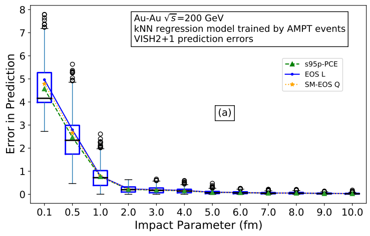

FIG. 2: Error distribution of b predictions of different centrality events of VISH2+1 model

10-fold cross-validation obtained are 97.52% for the k-NearestNeighbors (kNN) model and 95.18% for the ExtraTreesRegressor(ETR) model. But if we see the error distribution, we find that the errors are higher for most central collision ($b \leq$ 2fm) events which is shown in Fig. 1. This is because the event distribution of p_T spectra is left-skewed.

We also found that if the ML model is trained by a particular type of HIC model, it can make accurate predictions for the test data of a different HIC model. In Fig.2, we show the error plots of impact parameter predictions by the kNN model for the VISH2+1 test data [2]. Here also we find the error rapidly increases for the lower impact parameter events ($b \leq$ 2fm).

There are a few sampling techniques in ML for rebalancing datasets e.g., SmoteR, ADASYN [3]. These are python packages that increase(over-sampling) or decrease(under-sampling) the minority and majority data class respectively using the neighbouring data. The error comes down in the lower impact parameter region by using these standard rebalancing methods shown in Fig.3(a) Still we get enough errors that would give a wrong estimate for the low impact parameter events. We then adopt a method of rebalancing using class weights, where different classes are the different impact parameter regimes. The various combinations of distribution region and weights were evaluated through an exhaustive grid search. Based on test set minimum error, we selected events with impact parameter $\leq$ 1.0 fm to be in category 1 and the rest in category 2. The weights assigned to the two classes are in the ratio $4:1$. With our custom method, we were able to minimize the error

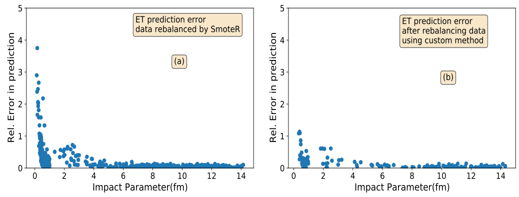

FIG. 3: Error distribution of b predictions of 200 GeV Au-Au collision events after rebalanced using (a) the SmoteR method, and (b) a custom method of giving weights to the input

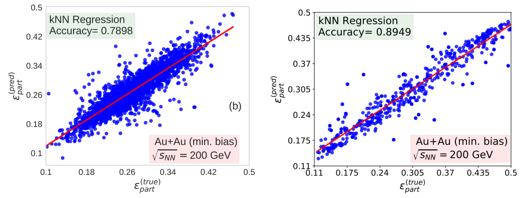

FIG. 4: ϵ_{part} predictions of Au-Au collision events at $\sqrt{s_{NN}}$ = 200 GeV using unbalanced data (left), and rebalanced data using the custom method(right).

to less than 1 shown in Fig. 3(b). This error is acceptable in this impact parameter range as the prediction made in this range will always fall in the most central collision category $(0-5\%)$ for the Au-Au collisions. ,

In the case of ϵ_2 and ϵ_{part} predictions, only three models kNN, ETR and RandomForest(RF) give more than 95% prediction accuracy. It is only true for a smaller eccentricity range $(0.22-0.32)$. For a larger range $(0.1-0.5)$, the accuracy reduced substantially from 98% to 78%, which is shown in Fig.4(left). This is also because of lesser statistics in the wider range. We use a similar rebalancing technique and obtained better accuracy, which is shown in Fig.4(right).

A.S is supported by INSPIRE Fellowship of the Department of Science and Technology, Govt. of India, (Grant no: IF170627)..

References

[1] Z. W. Lin et. al., Phys. Rev. C 72 064901 (2005).

[2] C. Shen et al., Comput. Phys. Commun. 199, 61 (2016).

[3] F. Hu and H. Li, Math. Probl. Eng. 2013, 694809 (2013).