ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA89

RHIC ELECTRON BEAM COOLING ANALYSIS USING PRINCIPLE COMPONENT AND AUTOENCODER ANALYSIS*

A. D. Tran†, Y. Hao
Facility for Rare Isotope Beam, Michigan State University, East Lansing, MI, USA
X. Gu, Brookhaven National Laboratory, Upton, NY, USA

Abstract

Principal component analysis and autoencoder analysis were used to analyze the experimental data of RHIC operation with low energy RHIC electron cooling (LEReC). This is unsupervised learning which includes electron beam settings and observable during operation. Both analyses were used to gauge the dimensional reducibility of the data and to understand which features are important to beam cooling.

INTRODUCTION

Beam cooling is an important technique that reduces the phase space area that is occupied by the particle's distribution. Strong hadron cooling, for the high energy and high-intensity hadron beam, will greatly benefit machines such as the future electron ion collider (EIC) to achieve a high luminosity result [1].

The optimization of the cooling performance involves tuning the parameters of the hadron accelerator, the cooler accelerator, and the alignment between them. Physics models are usually not precise enough to support physics-based optimization, therefore a data-driven approach may play an important role. With the presence of sufficient data, one can hopefully find hidden correlations and surrogate models for predicting and optimizing machine performance.

In this paper, we analyze the data-driven method using the averaged measured data of low-energy run of Relativistic Heavy Ion Collider (RHIC) with the Low Energy RHIC Electron Cooling (LEReC) [2]. The data were manually collected from each ramp of the RHIC run with cooling. These include the working points and collimating parameters of the RHIC ring, magnetic settings of the cooler magnets, alignments between the ion and electron beam, and the corresponding beam losses and luminosity.

The analysis being done is known as unsupervised learning, which is when we do not have a map between an input and the desired output. This helps us to find relationships between the variables as opposed to supervised learning which tries to predict the output given the input.

The following paper will first apply principle component analysis (PCA) and then an autoencoder analysis, to reduce the dimension and to understand which parameters are important when optimizing the beam cooling.

PCA ANALYSIS

PCA is a linear data analysis technique used to extract the most important information by compressing the size of the data [3]. The data was stored as a $L \times J$ matrix where L are observations and J are the features. Observations are the data from each run while features are the different inputs and outputs. After the dataset was cleaned, the resulted dataset had around 700 observations, each with 38 features. The data was standardized to a mean of 0 and a standard deviation of 1 with respect to the features. This resulted in the dataset X with matrix elements $\{x_{l,j} \in X | l \in L, j \in J\}$.

The inertia of a column is defined as

$$I_j = \sum_{l \in L} x_{l,j}^2, \tag{1}$$

while the inertia of the dataset is the total inertia, which is the sum of all the column inertia. The term inertia is used because values of a column J for all observation L can be thought of like a mass, and if each column has a zeroed mean, I_j would be the inertia. X is made of this inertia so if a column has a large inertia, it is said to explain more of the whole dataset.

PCA was used to project X into principle components and will be referred to simply as components. The first component has the largest inertia and subsequent ones have decreasing magnitudes while also being orthogonal to the previous components. The projection of X was calculated using standard value decomposition (SVD) as follows: Let the SVD of X be

$$X = U\Sigma Q, \tag{2}$$

then $F = U\Sigma$ is the projection of X on the principle components while Q is the projection matrix.

One of the goals of PCA is to project the matrix X into a matrix with fewer components (fewer variables) while maintaining the core information. This was used to create an approximate data matrix X_M with M principle components. The error between these two matrices is the residual sum of squares $RESS_M$,

$$RESS_M = ||X - X_M||^2, \tag{3}$$

where $||\quad||$ is the square root of the sum of all the squared elements of $X - X_M$. This is similar to mean square error (MSE). A lower number corresponds to a better recreation of the original data matrix. The analysis is shown in Fig. 1. With PCA, $RESS$ goes to zero as the number of components reaches the original number of components. The error

* Work supported by the US Department of Energy under contract No. DE-AC02-98CH10886.
† tranant2@msu.edu

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA89

decreased as more components were added and became negligible when 36 out of 38 components were reached. This does not directly suggest two features are insignificant. This only shows that there exist 2 columns, which are linear combinations of features, that are insignificant, and it is possible to reduce the dimension of the data at least by 2. To decide what is an appropriate number of components to keep, the standard technique [3] of only keeping components whose inertia is larger than the average inertia was used, resulting in 28 of the 38 components being kept.

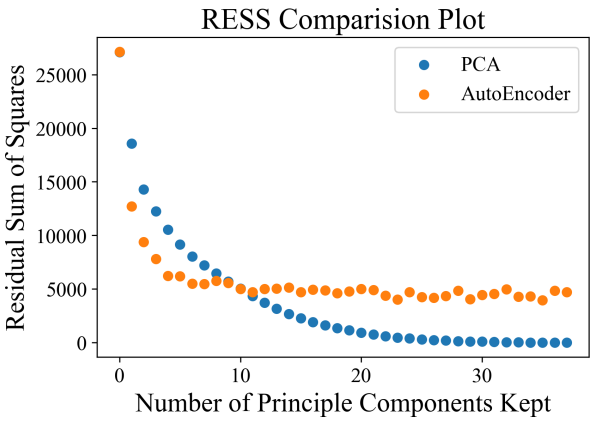

Figure 1: Comparison of RESS from PCA and Autoencoder for 38 different components.

Looking at how each observation is separated will allow for an interpretation of the components. If one component were to separate the observations such that some features are also separated clearly, then that component can be interpreted as corresponding to those features it separates. To do this analysis, all the observations were projected onto the first component. Then, starting with one feature, the observation would be assigned the value of that feature. Some components will have a clear pattern such as high values being grouped for positive projections and negative values grouped for negative projections. Others would be random and no clustering would be observed. This can be quantified using a clustering factor. Let the projected observations $f_{l,j} \in F$ be normalized into the range $[-1, 1]$. The clustering factor c_j is

$$c_j = \sum_{l \in L} x_{l,j} \sin f_{l,j} \ . \tag{4}$$

This works because if there is no clustering, then terms would cancel out, and if there is a clear separation, we would get a large value. This would not work for cases where the high and low pattern alternates, but this is a good low approximation.

The clustering factor was calculated for each feature and the results are plotted in Fig. 2. The large values show that the component is responsible for describing multiple data points. The features 'RF' and 'Ramp' has low clustering factor so it does not contribute to the explanation. Luminosity in this component has a clustering factor around 300, which

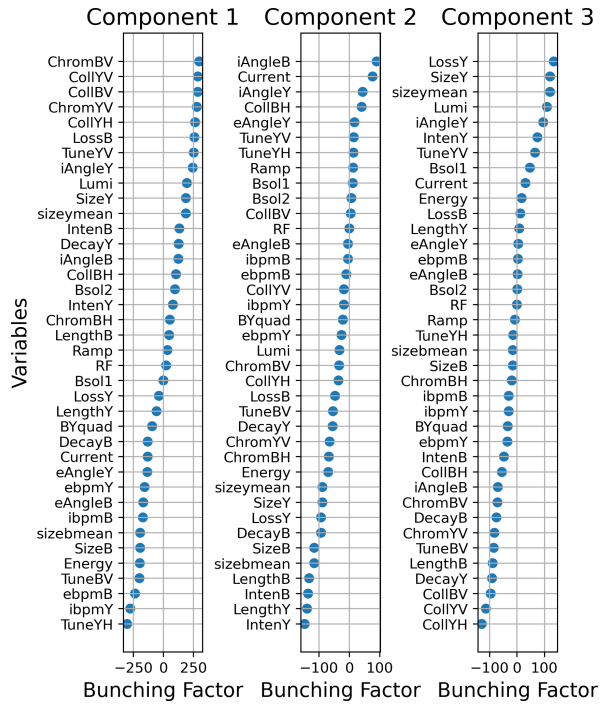

Figure 2: Clustering Factor of the features for the first principle component of PCA.

is the largest while the second largest is in component 3 with a value around 100. In both cases, 'RF' and 'Ramp' are around zero suggesting they are not related to Luminosity.

AUTOENCODER ANALYSIS

Another dimensionality reduction algorithm is autoencoder [4]. This is an unsupervised machine learning algorithm that first takes the inputs, put them through a series of the neural network, and in the end tries to recreate the original inputs. The inner layer of the neural network can be made arbitrarily small allowing the data to be compressed to a smaller latent dimension. This can be verified if the error between the original and recreated input is very small. Non-linear activation functions can also be used in the model making the algorithm able to capture non-linear effects.

Using PyTorch, a model was made where each observation was put through an encoder, a neural network mapping the observation to a variable latent dimension, then the result from this was put through a decoder, another neural network mapping from the latent dimension to the reconstructed observation. It was trained on the whole dataset using a mean square error loss function for latent dimensions from 1 to the total number of features. The RESS was also calculated for each step and the result is plotted in Fig. 1. The model was able to reach its error limit at around 8 latent dimensions. PCA was able to reach a similar level of error with 10 components.

The question now is what error is acceptable? Two latent dimensions will be taken for visual analysis. Using two latent dimensions, 3 features, ramp, energy, and luminosity

ISBN: 978-3-95450-232-5 ISSN: 2673-7000 doi:10.18429/JACoW-NAPAC2022-MOPA89

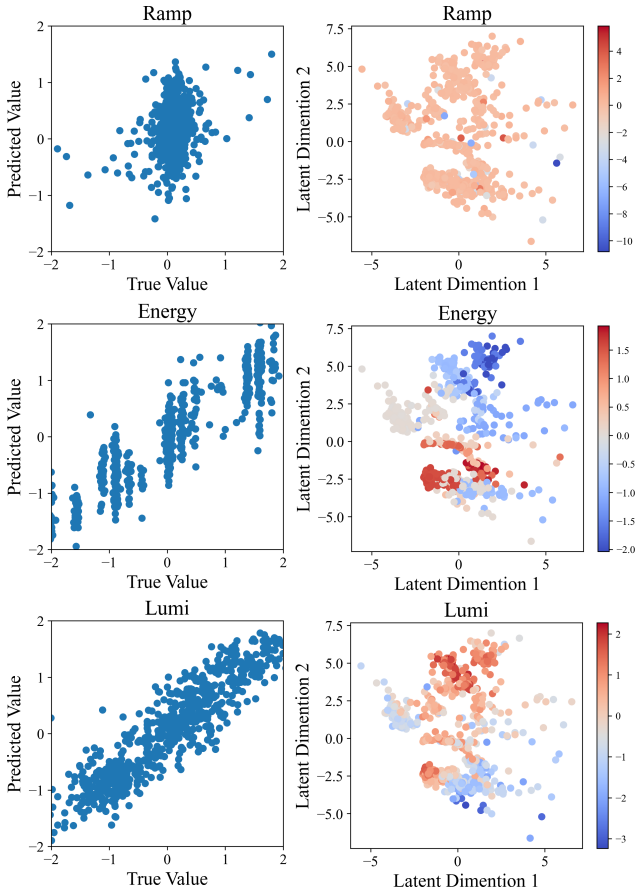

Figure 3: Left: Autoencoder loss. By using an autoencoder, it recreates the original observations better. It need only around 7–8 points until it reaches the limiting error. Right: Colored plots of different features projected onto two latent dimensions using Autoencoders. The color of each point corresponds to its standardized value on the bar.

will be used to showcase the analysis. After training, the true and predicted values of the three are plotted as a correlation plot in the left of Fig. 3. If we have perfect reconstruction, we would get a straight light. This is not the case for 'Ramp', but for 'Energy' and 'Luminosity', we do have a positive correlation; however, it is not perfect as expected. 'Ramp' is just a setting that tells how fast to turn on the voltage. This should not affect the experiment in the first place so seeing that 'Ramp' does not correlate is encouraging.

Another analysis was done by making an autoencoder using 2 latent dimensions. The observations can then be plotted on a 2-dimensional image to see clusters. Then the value of each observation can be colored by a feature to see some patterns. Looking at 'Ramp' in the right of Fig. 3, there is no pattern so it confirms the fact that 'Ramp' is not meaningful. Looking at luminosity, one can see clusters where there are high and low values. Then, by sampling points around regions of high luminosity values in the latent dimension, different beam settings can be reconstructed using the decoder that might be fruitful.

CONCLUSION

Dimensionality reduction is important as it cleans the data by removing unimportant data and allows for an interpretation of the importance of certain features. Using PCA, a linear algorithm, the data requires 28 out of the 38 components to be reduced. It was found that at least two features were unnecessary since the last two components have negotiable contributions and they may be 'RF' and 'Ramp' according to our interpretation analysis. Using an autoencoder, the data can be reconstructed with less error compared to PCA for the first few dimensions, but eventually reach an error limit since the dataset, which is under 1000, is not large and autoencoders need a lot of data to be trained on.

The advantage is that autoencoders can capture non-linear effects allowing it to compress the data in fewer point than PCA if non-linear effects are being dealt with, but is unable to reach usable accuracy without a lot of data to be trained on. PCA will eventually recreate the original dataset with enough components, but because it is a linear algorithm, it has a hard time reducing the dimension of non-linear effects.

A study comparing how well both methods can be used to aid in the prediction of certain outputs, such as luminosity, using a neural network should be conducted to better gauge how well the method can produce meaningful data reduction. At the moment, due to the small amount of data, a reducible set of inputs was not possible if the desired outputs were taken away from the dataset. This will be the topic of further investigation if more data can be collected.

REFERENCES

[1] A. Accardi *et al.*, "Electron-Ion Collider: The next QCD frontier", *Eur. Phys. J. A*, vol. 52, p. 268, 2016. doi:10.1140/epja/i2016-16268-9

[2] A. V. Fedotov *et al.*, "Bunched Beam Electron Cooler for Low-energy RHIC Operation", in *Proc. NAPAC'13*, Pasadena, CA, USA, Sep. 2013, paper TUOAA1, pp. 363–365.

[3] H. Abdi and L. J. Williams, "Principal component analysis", *WIREs Comp. Stat.*, vol. 2, no. 4, pp. 433-59, 2010. https://doi.org/10.1002/wics.101

[4] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders", preprint submitted to arXiv Mar 2020. doi:10.48550/arxiv.2003.05991