# Background Subtraction using Weights

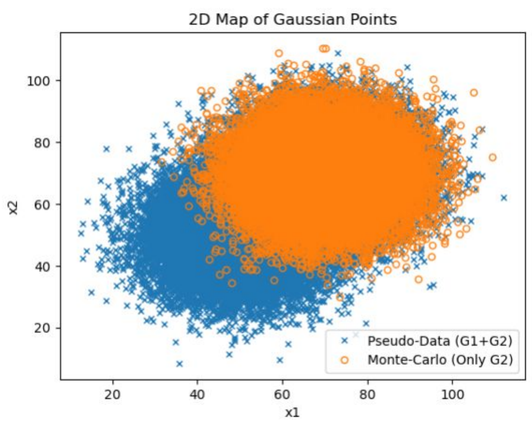
Duy Hoang[1], John Alison[1]

NSF AI Planning Institute for Data-Driven Discovery in Physics

[1]Carnegie Mellon University; Pittsburgh, PA ; Summer 2021

## Abstract

The Large Hadron Collider (LHC) is a particle accelerator that collides 1 billion particles per second [1]. The LHC produces an abundance of data that is a mixture of different samples. However, only some of the background signals can be modeled by simulation and therefore some background is not able to be identified in order to subtract away. By utilizing machine learning techniques such as neural networks, decision trees, etc. one can utilize data generated weights in order to achieve a background subtraction effect to obtain the underlying desired dataset.




## Introduction and Background Information Part I

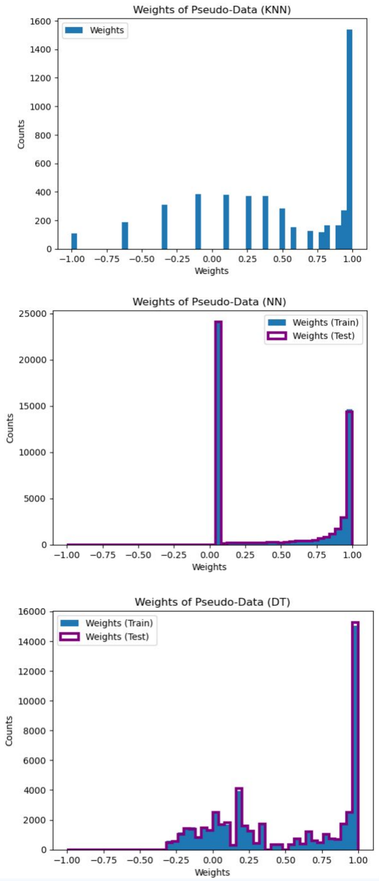
In this toy model, I'm simulating two datasets where one is composed of two 2D gaussians, labelled as pseudo-data, while the other dataset, labelled as monte-carlo, contains only one of the 2D gaussians that is shared. This simulates the situation that we can only model one of these gaussians accurately with G2 being the hypothetical background and G1 being the underlying data that we want to extract. I utilize the K-nearest-neighbors (KNN) algorithm, a neural network, and a decision tree to calculate the probabilities necessary to generate the weights for the data.

## Introduction and Background Information Part II

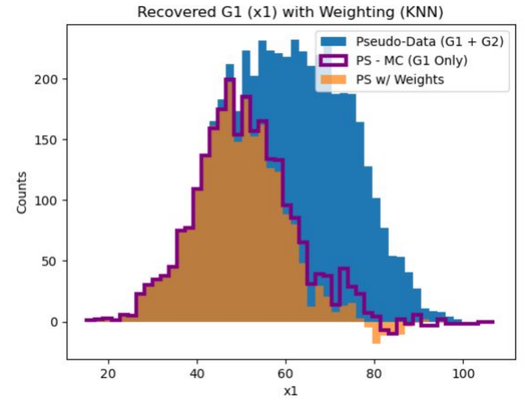
The KNN algorithm is a supervised machine learning algorithm that can be used to solve classification problems [2]. The KNN algorithm assumes that similar things exist in close proximity and by using the euclidean distance formula we can find the K nearest neighbors for each point. The probabilities are then calculated based on the fraction of the neighbors that were either a part of the pseudo data or a part of the monte carlo generated data and from these probabilities we can calculate weights following the equation:

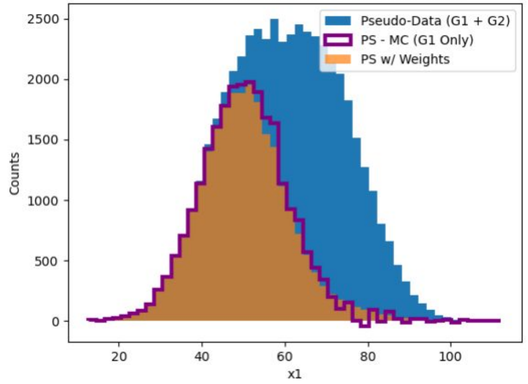
$$\omega_i = \frac{P(pd) - P(mc)}{P(pd)}$$

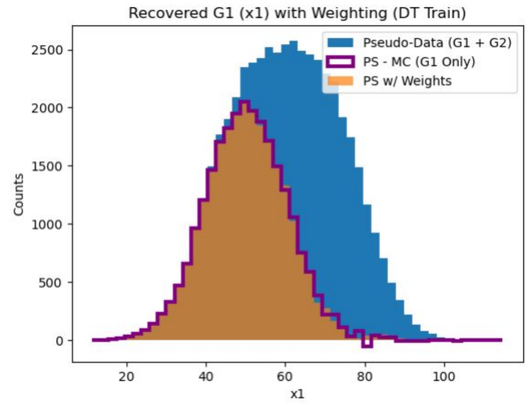
*pd = Pseudo-Data, mc = Monte-Carlo*

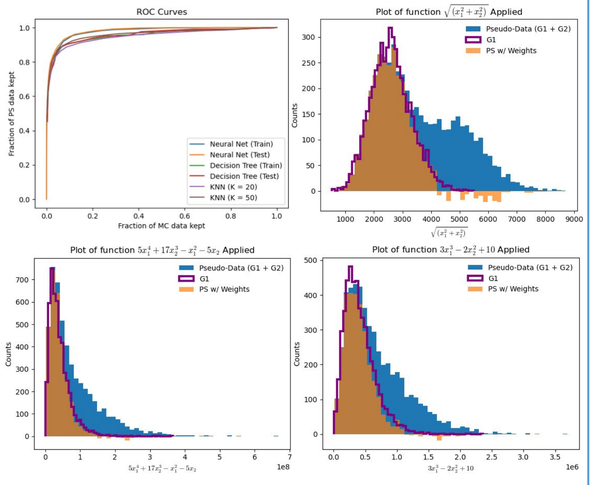



## Results




## Future Endeavors

With these machine learning algorithms all producing fairly accurate results, my future plans are to apply these same techniques onto a more realistic data set. I will be using a convolutional neural network to generate similar weights on a more complicated data set.

## Acknowledgements

I would like to thank John Alison with his continued support and the Carnegie Mellon Summer Undergraduate Research Program at the NSF AI Planning Institute for funding this work. This material is based upon work supported by the National Science Foundation under the NSF AI Planning Institute for Data-Driven Discovery in Physics, grant number PHY2020295. Any opinions, findings, and conclusions or recommendations expressed are those of the participants and do not necessarily reflect the views of the National Science Foundation or the participating institutions.

## References

[1] "CERN Accelerating Science." CERN
[2] Harrison, Onel. "Machine Learning Basics with the K-Nearest Neighbors Algorithm." Medium, Towards Data Science, 14 July 2019,