CS @ ILLINOIS COMPUTER SCIENCE

PURE Promoting Undergraduate Research in Engineering

Mutual Information Estimation Utilizing K-Nearest Neighbors

Mentees: Samuel Dovgin and Sehe Tinfang
Mentor: Alan Yang

I. Introduction

Mutual information (MI) is a measure of the mutual dependence between two random variables based on their marginal and joint probability distributions. Intuitively, MI quantifies the information obtained about one variable through the other. It has applications in machine learning as a criterion for feature selection and in optimization of neural networks.

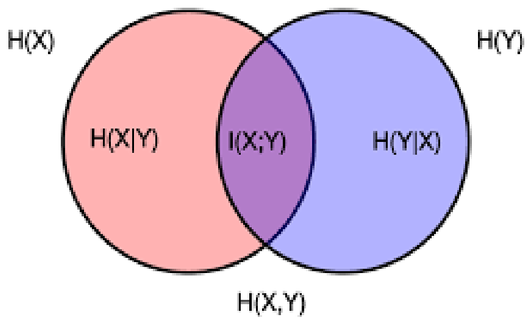

Figure 1: Representation of Mutual Information [3]

II. Problem Statement

Computing **MI** in machine learning can be difficult because usually only datasets are available, not the relevant distributions of random variables. One way to estimate MI from a dataset is by discretizing the data, but finite sampling leads to many errors in estimation.

The **KSG Nearest Neighbor Estimator** attempts to fix this issue by discretizing based on the K Nearest Neighbors to a point in the dataset instead of a fixed box width. Computing the K Nearest Neighbors is a very computationally expensive part of the algorithm, so we investigated a few ways to do this more efficiently (such as **KD-trees** and **Monte Carlo**) than brute force methods and their implementations.

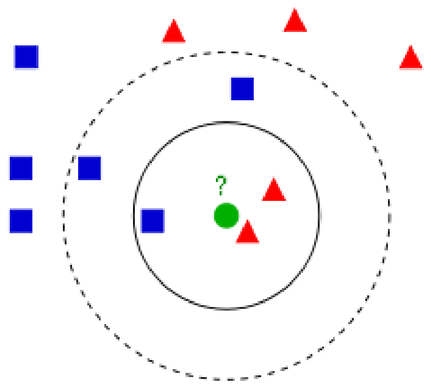

Figure 2: Diagram depiction of KNN which locates the 3 nearest datapoints[3]

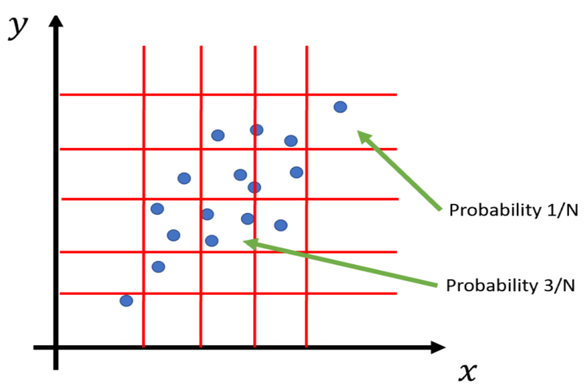

Figure 3: Graph of Bin Discretized Dataset [2]

III. Objective & Methodology

[Main objective] Increase the speed of K Nearest Neighbors for Mutual Information Estimation through several methods including approximation and utilizing new data structures. We expect to retain high accuracy while improving runtime.

We followed these stages to accomplish our objective:
[1st Stage] Learn the basics about the **KSG brute force** implementation and consider different solutions for optimizing its runtime.
[2nd Stage] Study the **KD-tree** data structure implementation and analyze its improvements on the algorithm
[3rd Stage] Finally test **Monte Carlo** random sampling simulators for approximation optimizations. Additionally test mutual information classifiers on data to see how they can be used. Understand the trade-offs of approximations on the accuracy of the function.

IV. Implementation

1st stage

Algorithm 1 Estimator for $I(X;Y)$. $\psi(\cdot)$ denotes the digamma function.

procedure KSG($\{X_i, Y_i\}_{i=1}^N$, k)
  for $i \in \{1, ..., N\}$ **do**
    $\epsilon_i =$ The k-th smallest distance among
      $\left\{ d_{i,j} = \max\left\{\|X_i - X_j\|, \|Y_i - Y_j\|\right\} : j \neq i \right\}$
    $n_{x,i} = \#$ samples with $\max\left\{\|X_i - X_j\|\right\} \leq \epsilon_i$
    $n_{y,i} = \#$ samples with $\max\left\{\|Y_i - Y_j\|\right\} \leq \epsilon_i$
    $\xi_i = \psi(k) - \psi(n_{x,i}) - \psi(n_{y,i}) + \psi(N)$
  end for
  return $\hat{I}(X;Y) = \frac{1}{N}\sum_{i=1}^{N} \xi_i$
end procedure

Figure 4 (left): KSG Mutual Information Estimator algorithm. The kNN method is utilized within the algorithm. [2]

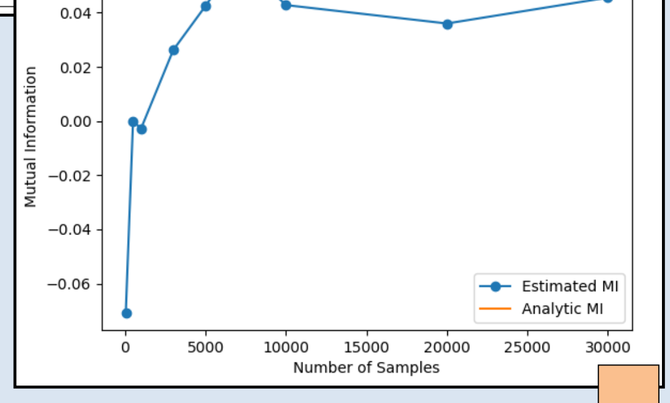

Figure 5 (right): Graph of actual Mutual Information of a dataset (horizontal orange line) with the KSG Brute-Force Estimator (blue plot). The algorithm is more accurate with larger samples and converges on the actual Mutual Information. [2]

2nd stage

Figure 6: Approximately linear graph of the KD-Tree time gain in reference to the Brute-Force K Nearest Neighbor Algorithm.

Using a KD-tree (k-dimensional tree) we were able to quickly parse the samples and find the KNN of each point. Comparing this to the Brute-Force version of the we saw significant magnitudes in conserved time. KD-trees do not greatly increase the **space complexity** of the dataset and as such we found this method to be efficient on multiple fronts.

3rd stage

Using a mutual info classifier we tested the ability of MI to predict the most important features in an example dataset from the **UCI Machine Learning Repository**. In this case, we tested a mushroom dataset with two classes, poisonous or edible. Through mutual info alone we were able to accomplish an essential machine learning task. This selection picks only the useful data, saving time.

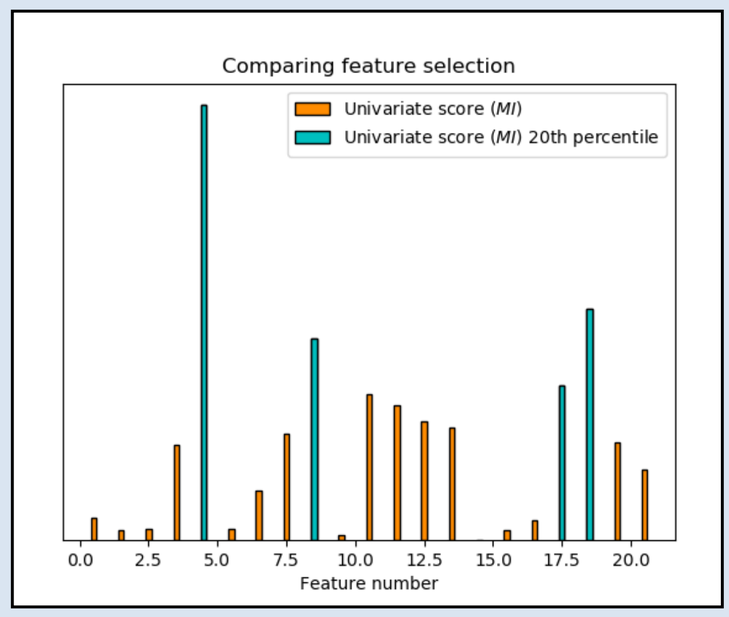

Figure 7: This diagram charts the most important features in a specified dataset. The top 20th percentile of data were selected most significant and are highlighted blue

V. Conclusions

We were able to develop several strategies and compare different methods of more efficiently computing K Nearest Neighbors. This can be applied to our choice of the **KSG Mutual Information Estimator** however it can improve other algorithms as well. Future work would be to continue improving KNN efficiency or that of other algorithms.

Throughout this project we have learned how to utilize new data structures such as KD-trees and implementing them to decrease runtime. Additionally we have learned how to parse large machine learning datasets and apply specific classifiers in **PYTHON.** Experience with python libraries Matplotlib and Numpy will be extremely useful.

Acknowledgments

We appreciate the support of our mentor for teaching us and guiding us to the necessary knowledge and skills required for this research. We are also thankful to the **PURE committee** for proving this opportunity to get introduced to research with grad students.

References

[1] Estimating mutual information: Alexander Kraskov, Harald Stögbauer, and Peter Grassberger
[2] Mutual Information Write-Up: Alan Yang
[3] Mutual Information Website: www.scholarpedia.org