CYTOAUTOCLUSTER

Enhancing Cytometry with Deep Learning

INTRODUCTION :

CytoAutoCluster aims to integrate semi-supervised learning approaches within cytometry workflows. By utilizing both labeled and unlabeled data, this project seeks to develop a robust clustering algorithm that can adaptively learn from the inherent structure of cytometric data. This innovative approach not only aims to enhance the accuracy of cell classification but also to reduce the reliance on extensive labeled datasets, which can be labor-intensive and time-consuming to create.

PROBLEM OVERVIEW :

Cytometry generates vast amounts of high-dimensional data, often leading to challenges in the interpretation and classification of cell populations. Traditional clustering methods, such as k-means or hierarchical clustering, may struggle with the complexity and variability of cytometric data. Key challenges include:

1. **High Dimensionality**: Cytometric data often consists of multiple parameters, making it difficult to visualize and analyze using conventional methods.

2. **Label Scarcity**: High-quality labeled datasets are crucial for supervised learning but are often scarce in biological research, limiting the effectiveness of traditional machine learning approaches.

3. **Noise and Variability**: Cytometric data can be noisy and exhibit significant variability due to biological differences, which can adversely affect clustering performance.

OBJECTIVES :

The primary objectives of CytoAutoCluster are as follows:

1. **Develop a Semi-Supervised Learning Framework**: Create an algorithm that can effectively utilize both labeled and unlabeled data to enhance clustering performance in cytometric analyses.

2. **Improve Clustering Accuracy**: Implement deep learning techniques to achieve higher accuracy in classifying complex cell populations compared to traditional methods.

3. **Reduce Labeling Requirements**: Minimize the need for extensive labeled datasets by leveraging unlabeled data, thereby reducing the time and resources required for data preparation.

4. **Enhance Interpretability**: Provide tools and visualizations that help researchers understand the clustering results and the underlying biological significance of identified cell populations.

5. **Scalability and Efficiency**: Ensure that the developed algorithm is scalable and efficient, capable of handling large datasets commonly encountered in cytometry.

DAY-1 | 07-10-2024 :

- ✓ Referred and Downloaded a valid semi-supervised cytometry mass data set from resources like kaggle,google scholar,paperswithcode.

DAY-2 | 08-10-2024 :

- ✓ My data set got rejected because it contains complete labelled data as we need some I showcased my downloaded data set to the mentor and took my inputs.
- ✓ percentage of unlabelled data to perform semi supervised learning.
- ✓ As per the feedback from mentor, I must find mass cytometry data with more than 40 columns and unlabelled data must be more than 60% and it is to be completely in tabular format.

DAY-3 | 09-10-2024 :

- ✓ Gained a brief knowledge on the components of the cytometry in the data set with an sample data set (Levine_32dim_notransform.csv).
- ✓ Learned about some basic GIT commands to use in the repository of the github.
- ✓ Bagged the idea of how the code in collabed repository is used by the multiple users (push ,merge and commit).

DAY-4 | 10-10-2024 :

- ✓ Finalized the data set i.e., Levine_32dim_notransform.csv
- ✓ Created Python Environment and uploaded the data set , created a dataframe.

DAY-5 | 11-10-2024 :

- ✓ Removed Unnecessary columns (Cell_length,file_number, event_number) from the data set.
- ✓ Performed EDA Techniques (Info,Histogram,Label and Unlabel percentage,non-null values in each column) and took inputs from the mentor.

HISTOGRAM

A histogram is a type of bar chart that represents the frequency distribution of a dataset. It displays data by grouping it into bins or intervals, with each bin representing a range of values. The height of each bar shows the frequency or count of data points within that range, helping to visualize the distribution, spread, and central tendencies of the data.

DAY-6 | 14-10-2024 :

- ✓ Performed EDA Techniques (Range of each feature , Correlation matrix, Class Label Distribution ,Box Plot) and took inputs from mentor.

BOXPLOT

Outliers in a boxplot are data points that fall outside the whiskers, typically defined as 1.5 times the interquartile range (IQR) above the third quartile or below the first quartile.

CORRELATION MATRIX

A correlation matrix is a table displaying the correlation coefficients between multiple variables in a dataset, showing the degree and direction of linear relationships. Each cell in the matrix contains the correlation value between a pair of variables, ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation), with 0 indicating no correlation. This matrix is a valuable tool for identifying relationships, multicollinearity, and feature dependencies using heatmaps to make patterns and strengths of relationships clearer.

DAY-7 | 15-10-2024 :

- ✓ Performed EDA Techniques (Kurtosis, Skewness , pair plot) and took inputs from my mentor.

KURTOSIS

Kurtosis measures the "tailedness" of a probability distribution, indicating how much data is in the tails compared to a normal distribution. There are three types of kurtosis:

1. **Mesokurtic**: This is a normal distribution with kurtosis close to zero, indicating average tail presence, like the bell curve.
2. **Leptokurtic**: Distributions with high kurtosis (>0) are leptokurtic. They have heavy tails, meaning more data falls in the tails, suggesting more outliers. This results in a sharp peak and flatter tails.
3. **Platykurtic**: Distributions with low kurtosis (<0) are platykurtic, with thin tails and fewer outliers. They have a flatter peak and less extreme values in the tails.

High kurtosis implies data is prone to extreme values, while low kurtosis shows a more consistent, predictable dataset.

SKEWNESS

Skewness measures the asymmetry of a probability distribution. A perfectly symmetrical distribution has zero skewness, but real-world data often leans to one side. There are two main types:

1. **Right Skewness (Positive Skew)**: Here, the tail on the right side of the distribution is longer, meaning the majority of data points lie on the left. It indicates that the mean is typically greater than the median, and it's common in distributions with high outliers, like income data.
2. **Left Skewness (Negative Skew)**: In left-skewed distributions, the tail on the left side is longer, with most data points on the right. Here, the mean is often less than the median, and it occurs in data with low outliers, such as age at retirement.

PAIRPLOT

A pairplot is used to show pairwise relationships in a dataset by creating a matrix of scatterplots for each pair of variables. It helps identify trends, distributions, and correlations among features.

DAY-8 | 16-10-2024 :

- ✓ Performed the individual plotting of each feature of kurtosis and skewness in the data set.
- ✓ Read the Research paper about the project (https://www.sciencedirect.com/science/article/pii/S0092867415006376)
- ✓ Learnt a Brief about the research paper from the mentor.

DAY-9 | 17-10-2024 :

- ✓ Could not get expected results by performing t-SNE and pca on the data set.
- ✓ Found that we must Standardize the relevant features range from 0 to 1 and remove columns with null values .
- ✓ So performed the t-SNE on a MNIST data set to understand the working of the model.

STANDARDIZING VALUES

Standardizing values in a DataFrame scales features so they have a mean of 0 and a standard deviation of 1.

MNIST

The MNIST dataset is a large collection of 70,000 grayscale images of handwritten digits (0–9), commonly used for training and testing image processing systems.

DAY-10 | 18-10-2024 :

- ✓ Gained brief knowledge about what are t-SNE and PCA and How they work.

DAY-11 | 21-10-2024 :

- ✓ Performed t-SNE and PCA Techniques.

PCA

Principal Component Analysis (PCA) is a dimensionality reduction technique used in transforming the data into a set of new, uncorrelated variables called principal components, PCA captures the most important information (or variance) with fewer dimensions.This reduction helps in visualizing high-dimensional data, speeding up algorithms, and minimizing noise.

t-SNE

t-Distributed Stochastic Neighbor Embedding (t-SNE) is used for visualizing high-dimensional data in a low-dimensional space, typically 2D or 3D.t-SNE preserves the local structure of data, so similar points in the high-dimensional space stay close in the visualization. It does this by minimizing the differences in probability distributions of point distances between high and low dimensions, resulting in clusters that reflect the original relationships.

DAY-12 | 22-10-2024 :

- ✓ Performed some operations of PCA like Standard Deviation , Proportion of Variance, Cumulative Proportion.
- ✓ Performed 3-D plot of PCA.
- ✓ Learnt a brief on Auto Encoders like on what domain encoders are used and some model names that are implemented using auto encoders.

DAY-13 | 23-10-2024 :

- ✓ Studied the Research paper (https://arxiv.org/pdf/2006.05278) about Deep Semi-Supervised Learning.

DAY-14 | 25-10-2024 :

- ✓ Gained a brief knowledge about semi-super vised learning from the research paper .

DAY-15 | 28-10-2024 :

- ✓ Learnt about Consistency Reguralization , Entropy Minimization in semi super vised learning.

CONSISTENCY REGURALIZATION :

Consistency regularization enforces that a model's predictions remain stable under various perturbations or augmentations of the input data.

ENTROPY MINIMIZATION :

Entropy minimization is used to reduce the uncertainty of the model's predictions on unlabeled data by encouraging it to produce confident outputs.

- ✓ Learnt about how to perform binary masking on set of rows in a frame.

DAY-16 | 29-10-2024 :

- ✓ Performed Binary Masking on a set of rows.

BINARY MASKING :

Binary mask is used to selectively focus on certain parts of the input data, allowing the model to learn from both labeled and unlabeled data effectively. This technique helps enhance the model's ability to capture relevant features while ignoring noise, thereby improving performance on tasks with limited labeled data.

- ✓ Read the abstract of VIME (Value Imputation and Mask Estimation) :Extending the Success of Self- and Semi-supervised Learning to Tabular Domain.

DAY-17 | 30-10-2024 :

- ✓ Performed binary masking by randomly shuffling the column values into the data frame.

CORRUPTION OF DATA :

Corrupted data refers to values that have been altered or distorted, either intentionally (for testing purposes) or unintentionally (due to errors in data collection, processing, or storage).

- ✓ Performed the corruption of data by creating a new data frame called corrupted data frame and framed and equation x*(1-m) + x_shuffled*m = x_corrupted
  I.e; x is original data frame
  m is binary mass matrix
  x_shuffled is shuffled data frame .

DAY-18 | 01-11-2024 :

- ✓ Performed Binary Masking on our data set
- ✓ Created an corrupted data frame such that the binary mass matrix when represents 1 indicates that the value in that index is corrupted and vice versa to the 0.
- ✓ Created 4 variables named x_labeled, y_labeled ,x_unlabeled and y_unlabeled to split the data in the data set into catogeries.

  x_labeled represents the rows of columns that have values except the target column(label).

  X_unlabeled represents the rows having null values.

  Y_labeled represents the rows of label that contains values.

  Y_unlabeled represents the rows of label that have null values.

DAY-19 | 04-11-2024 :

- ✓ Performed split method on x_label,y_label and y_unlabel such that size of train is 70% and size of test is 30%.

TRAIN SPLIT

Train-test splitting is a technique used to divide a dataset into two subsets: a training set, which is used to train the model, and a test set, which is used to evaluate the model's performance on unseen data. This process helps to prevent overfitting and ensures that the model generalizes well to new data by providing an unbiased assessment of its predictive capabilities.

DAY-20 | 05-11-2024 :

- ✓ Performed Logistic Regression and xgboost on train split data of our data set and also calculated the Log loss values .

LOGISTIC REGRESSION

Logistic regression is a statistical method used for binary classification that models the relationship between a dependent binary variable and one or more independent variables by estimating probabilities using a logistic function. When applied to a train split of a dataset, it helps in predicting outcomes based on the features present in the training data, enabling evaluation of the model's performance on unseen validation data.

XGBOOST

XGBoost (Extreme Gradient Boosting) is an efficient and scalable implementation of gradient boosting that enhances predictive performance through parallel processing and regularization techniques. When applied to a train split of a dataset, it builds a series of decision trees in an iterative manner, allowing for robust handling of complex patterns and interactions in the data.

LOGLOSS

Log loss is used as the objective function to optimize during training, guiding the model to make more accurate probability estimates for binary outcomes.

DAY-21 | 06-11-2024 :

- ✓ Created an model under the guidance of mentor I.e; created an function called self_supervised with some relevant parameters and a part of code on it.

DAY-22 | 07-11-2024 :

- ✓ Performed further coding on the self_supervised function like created encoder variable and to return encoder result.
- ✓ Run the self_supervised funtion.

DAY-23 | 08-11-2024 :

- ✓ Performed encoder method.

ENCODER

Encoders in a dataset are techniques used to convert categorical variables into numerical formats, making them suitable for machine learning algorithms. Common methods include one-hot encoding, which creates binary columns for each category, and label encoding, which assigns a unique integer to each category.

DAY-24 | 11-11-2024 :

- ✓ Performed logistic regression and XGboost on encoded data.

DAY-25 | 12-11-2024 :

- ✓ Discussed and Corrected the mistakes in encoder model and took my inputs from the mentor.

DAY-26 | 13-11-2024 :

- ✓ Started working on semi supervised learning model, created 3 functions model,train and semi_supervised.

Where,

```
model(input_dimension,hidden_dimension,label_dimension,activation=tf.nn.relu)
```

Here input_dimension represents the input layer,
hidden_dimension represents the number of neurons,
label_dimension represents the output of x,
activation is a relu function.

```
train(feature_batch,label_batch,unlabeled_feature_batch,model,beta,supv_loss_fn,optimizer)
```

Here, feature_batch is used in getting outputs for labeled data,
Label_batch is used in calculating supervised log function for labeled data,
Unlabeled_feature_batch is used in getting outputs for unlabeled data,
Beta is a hyperparameter i.e, we have to enter our own value for this,
Supv_loss_fn is used in calculating supervised log function for labeled data,
Optimizer is used in making the changes to the weights.

DAY-27 | 14-11-2024 :

✓ Continued developing code on semi-supervised learning model.

```
semi_supervised(x_train,y_train , x_unlabaled, x_test, parameters, mask_probability, K , beta)
```

Here, parameters is dictionary.

DAY-28 | 15-11-2024 :

✓ Completed developing code on semi_supervised function.

DAY-29 | 19-11-2024 :

✓ Discussed and corrected the errors regarding the semi_supervised function and took inputs from the mentor.