CYTOAUTOCLUSTER

ENHANCING CYTOMETRY WITH DEEPLEARNING

Internship Project Report

Submitted by: AKSHAYA V

INTRODUCTION:

CytoAutoCluster integrates semi-supervised learning into cytometry workflows. It leverages both labeled and unlabeled data to uncover patterns in cytometric datasets. The project focuses on creating adaptive clustering algorithms for better accuracy. This innovation reduces dependency on large, labor-intensive labeled datasets. By learning from data structure, it improves cell classification efficiency. The result is a faster, more robust solution for cytometry analysis.

PROBLEM OVERVIEW:

Cytometry produces large volumes of complex, high-dimensional data, making it challenging to interpret and classify cell populations. Traditional methods like k-means or hierarchical clustering often struggle with these complexities. Key issues include:

1. High Dimensionality

Cytometric data often includes numerous parameters per sample, making it challenging to analyze and visualize. This complexity can obscure meaningful patterns, requiring advanced techniques for interpretation. Traditional methods struggle to manage the intricate structure inherent in such high-dimensional datasets.

2. Limited Labeled Data

Supervised learning depends on labeled datasets, but creating them is resource-intensive and time-consuming. This scarcity of high-quality labels limits the application of traditional machine learning methods. As a result, many datasets remain underutilized, reducing the potential for accurate analysis.

3. Noise and Variability

Biological differences introduce significant variability in cytometric measurements, often resulting in noisy datasets. This noise makes detecting consistent patterns more difficult, impacting clustering accuracy. Traditional approaches often fail to accommodate this variability effectively.

OBJECTIVES:

1. Develop a Semi-Supervised Learning Framework

Design an algorithm that can effectively combine labeled and unlabeled data. This approach enables learning from data patterns without full reliance on labels. It enhances clustering performance, particularly in complex cytometric analyses.

2. Boost Clustering Accuracy

Incorporate deep learning techniques for more precise classification of cell populations. These methods outperform traditional clustering approaches in handling complex data. Improved accuracy allows for better insights into biological processes.

3. Reduce Dependence on Labeled Data

Utilize unlabeled data to decrease the need for large, labeled datasets. This approach saves time and resources in creating labeled training data. It makes the analysis process more efficient and accessible to researchers.

4. Enhance Result Interpretability

Provide visualization tools to make clustering results easier to understand. Help researchers uncover the biological meaning behind identified cell populations. Better interpretability bridges the gap between data analysis and biological insights.

5. Ensure Scalability and Efficiency

Develop an algorithm that handles large cytometric datasets with ease. Focus on computational efficiency to manage real-world data sizes effectively. Scalability ensures broad applicability across diverse research workflows.

CODE IMPLEMENTATION:

- **DATASET LOADING:**

  The process starts by loading the Levine CytOF dataset, which contains 32 dimensions, into the workspace. This dataset, stored as a CSV file, is imported into Google Colab for analysis**.**

- **DATA EXPLORATION:**

  Once loaded, the dataset is examined to get a sense of its structure by inspecting the first few rows and its overall dimensions. This step helps identify the number of features and records available for clustering.

- **NULL VALUE ANALYSIS:**

  As semi-supervised learning depends on the completeness of data, it's crucial to identify any missing values. This involves thoroughly inspecting the dataset to find columns with gaps and pinpointing the specific rows that are incomplete, which is important for assessing data quality.

- **COMPARING NULL AND NON-NULL VALUES:**

  To gain a clearer understanding of the dataset's structure, visualizations are created to show the distribution of null and non-null values. These graphs help identify which features have missing data and the extent of the missingness, informing the necessary data preprocessing actions.

- **DATA ANALYSIS METHODS:**

  HISTOGRAM - A histogram is a bar chart that shows how data is distributed across different ranges or intervals. It organizes the data into bins, where each bin represents a specific range of values, and the height of each bar indicates the number of data points within that range. This helps visualize the data's distribution, spread, and central tendencies.

CORRELATION MATRIX- A correlation matrix is a table that shows the correlation coefficients between various variables in a dataset, illustrating the strength and direction of linear relationships. Each cell in the matrix holds a correlation value between two variables, ranging from -1 (perfect negative correlation) to +1 (perfect positive correlation), with 0 indicating no correlation. This matrix is useful for detecting relationships, multicollinearity, and dependencies between features, with heatmaps enhancing the visualization of patterns and the intensity of these relationships.

- **FEATURE VALUE RANGE:**

The range of values for each feature is calculated, providing insights into the scale and variability of each attribute.

- **BOX PLOT ANALYSIS:**

Box plots are used to visually examine the distribution of each feature:

- **Median and Quartiles:** They display the data's central tendency, showing the spread and any skewness based on the median and quartiles.
- **Outliers:** Data points that fall outside the interquartile range are marked, helping identify potential outliers that may require attention during feature engineering or data cleaning.

- **SKWENESS AND KURTOSIS ANALYSIS:**

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

1. Right Skewness (Positive Skew)

The distribution has a longer tail on the right, with most data points on the left. The mean is usually greater than the median, often seen in data with high outliers, like income.

2. Left Skewness (Negative Skew)

The distribution has a longer tail on the left, with most data points on the right. The mean is typically less than the median, common in data with low outliers, such as age at retirement.

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

1. **Mesokurtic:** A normal distribution with kurtosis near zero, indicating a typical amount of data in the tails, like the bell curve.
2. **Leptokurtic:** Distributions with high kurtosis (>0) have heavy tails, meaning more data points are in the tails, suggesting a higher likelihood of outliers. These distributions have a sharp peak and flatter tails.
3. **Platykurtic:** Distributions with low kurtosis (<0) have thin tails, fewer outliers, and a flatter peak, indicating a more stable dataset with less extreme values.
   High kurtosis suggests data is more likely to have extreme values, while low kurtosis indicates a more predictable distribution.

- **DIMENSIONALITY REDUCTION TECHNIQUES:**

1. Principal Component Analysis (PCA)

PCA simplifies high-dimensional data by keeping the key features that explain most of the data's variance. It's especially useful for making complex datasets easier to visualize and analyze.

- **Variance Explained:** Identifies the most important components needed to capture at least 95% of the data's variance in fewer dimensions.
- **Dimensionality Reduction:** Reduces the risk of overfitting and computational load, making models more efficient.

2. t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-SNE is a technique designed to visualize data by projecting it into 2D or 3D spaces, revealing hidden patterns and clusters.

- **Local Similarity Preservation:** Unlike PCA, it focuses on capturing non-linear relationships and emphasizing local structures in the data.
- **Cluster Detection:** Helps identify natural groupings, which is valuable in semi-supervised learning for analyzing the structure of unlabeled data.

- **AUTOENCODERS:**

Autoencoders are neural networks designed to extract key features from data in semi-supervised learning. They process both labeled and unlabeled data, making them effective for representation learning. This helps improve performance in tasks like clustering and classification.

- **AUTOENCODER PROCESS:**

 Encoding: Compresses the input data into a smaller set of essential features. This step captures the most important patterns while reducing redundancy.

 Latent Space: Stores the compressed representation of the data in a lower-dimensional form. It serves as the foundation for both reconstruction and analysis.

 Decoding: Rebuilds the original data from the latent space, ensuring minimal loss of critical information. This step validates the quality of the compressed features.

- **APPLICATIONS AND DATA REQUIREMENTS:**

 Input Dimensions: Autoencoders handle high-dimensional cytometry data and transform it into simplified, lower-dimensional formats. This helps in uncovering essential patterns.

 Noise Introduction: Adding noise or dropout to the input data improves the model's ability to extract robust features. This technique enhances the autoencoder's performance by making it focus on meaningful structures.

- **BINARY MASKING AND DATA AUGMENTATION:**

To make the clustering model more robust, binary masking and data augmentation are employed:

1. **Binary Masking:** Randomly masks 30% of feature values in the dataset to mimic missing data. Training on such incomplete data helps the model handle real-world scenarios where data may be incomplete.
2. **Shuffling Feature Columns:** Feature columns are shuffled to introduce variability, creating diverse training samples. This enhances the model's ability to generalize and adapt to different data patterns during semi-supervised clustering.

- **DATA SPLITTING AND LABEL HANDLING:**

  After introducing corruption, a mask differentiates between "labeled" (corrupted) and "unlabeled" (original) data, enabling a semi-supervised framework. The data is split into training and testing sets, where the corrupted data trains the model, and uncorrupted data is used for reconstruction. If no label column is available, a new one or the DataFrame index is assigned as the label.

- **POTENTIAL DOWNSTREAM TASKS:**

  The latent features learned by the autoencoder can be applied to multiple tasks:

1. **Denoising:** Cleans the data by eliminating noise and recovering its original features.
2. **Anomaly Detection:** Detects outliers by analyzing reconstruction errors.
3. **Dimensionality Reduction:** Uses the compact latent space as a lower-dimensional representation for clustering or classification tasks.

- **LOGISTIC REGRESSION:**

  Logistic regression is a statistical technique for binary classification that models the relationship between a binary outcome and one or more predictor variables by estimating probabilities through a logistic function. When trained on a dataset, it predicts outcomes based on the input features, allowing the model's performance to be assessed on unseen validation data.

- **XGBOOST:**

  XGBoost (Extreme Gradient Boosting) is a fast and scalable version of gradient boosting that improves prediction accuracy by using parallel processing and regularization methods. When trained on a dataset, it creates a sequence of decision trees iteratively, effectively capturing complex patterns and interactions in the data.

- **LOGLOSS:**

  Log loss is the objective function optimized during training to improve the model’s ability to predict accurate probabilities for binary outcomes. It penalizes incorrect predictions more heavily as the confidence in them increases. Minimizing log loss leads to better probability estimates and model accuracy.

- **ENCODER:**

Encoders are methods used to transform categorical variables into numerical values, allowing them to be processed by machine learning models. Popular techniques include one-hot encoding, which generates binary columns for each category, and label encoding, which assigns a unique integer to each category.

  - ✓ Logistic regression and XGBoost were applied to the encoded dataset.

- **SEMI-SUPERVISED LEARNING:**

Supervised learning is a type of machine learning where algorithms are trained on labeled datasets to make predictions and identify patterns. The model learns from the input-output pairs in the data to map inputs to their correct outcomes.

¬ Applied t-SNE to the data after executing the semi-supervised function. This technique was used to visualize the data in a lower-dimensional space. t-SNE helped reveal any underlying structures or clusters in the semi-supervised dataset.

- **GRADIO INTERFACE:**

Implement a Gradio interface by selecting a subset of unlabeled data (around 100 rows) as a DataFrame, and then plotting the corresponding outputs on a graph.

Gradio provides a web-based GUI for creating interactive demos around semi-supervised learning models, allowing users to easily visualize and interact with the model's predictions.

CONCLUSION:

In conclusion, the integration of semi-supervised learning with tools like Gradio enhances model performance by leveraging both labeled and unlabeled data. The use of techniques like t-SNE for visualization and binary masking for robustness further improves the model's ability to handle complex data patterns. Additionally, applying methods such as logistic regression and XGBoost on encoded data facilitates accurate predictions and insights. Gradio's interface offers an intuitive way to showcase and interact with machine learning models, making them more accessible. By reducing dependency on labeled data, these approaches streamline the training process. Ultimately, this combination of methods helps build more efficient and scalable machine learning systems.

REFERENCES:

- Levine, J.H., et al. : [Data-Driven Phenotypic Dissection of AML](https://www.sciencedirect.com/science/article/pii/S0092867415006376)
- Kim, B., et al. : [VIME: Value Imputation and Mask Estimation](https://arxiv.org/pdf/2006.05278)
- Céline Hudelot, Myriam Tam : [Deep Semi-Supervised Learning](https://arxiv.org/pdf/2006.05278)