DS250 : Project Phase3 - Report

PC-Expo: A Metrics-Based Interactive Axes Reordering Method for Parallel Coordinate Displays

Vetsa Tarun(12041720)

---

1 Datasets Used:

Cars Dataset: (Cars.csv)

The cars dataset contains information about various car models such as their miles per gallon (mpg), number of cylinders, horsepower, weight, acceleration, year of production, and country of origin.

Systems Dataset

The systems dataset used in the paper contains information about the total requests, cost, requests per second, number of reads, latency, and number of writes for various computer systems. But the dataset is not found anywhere.

So, another datasets we can use are Dataset1.csv, Dataset2.csv

2 Algorithms Implemented:

I have implemented the algorithm which was proposed in the project paper.

Algorithm:

The algorithm is to reorder the axes and plot the PCP(Parallel Coordinate Plots) using the weighted sum of all 12 metric properties mentioned in the project.

- No, the code for the implementation of the algorithm was not available previously.
- I have wrote nearly 300 - 350 lines of code in the project and the code will be submitted along with the report.

3 Work on extensions

1. The code supports for the negative weight to the properties, so that it can actively ignore certain properties.
2. Also, when the data dimensionality goes beyond 15 dimensions then the calculation of metrics become slower. So, to overcome this issue data filtering and data reduction technique(PCA) has been used which reduces to 10 dimensions.

```python
from sklearn.decomposition import PCA

if df_o.shape[1] > 15:
    # Apply PCA for dimensionality reduction
    pca = PCA(n_components=10)
    df = pca.fit_transform(df_o)
else:
    df=df_o

df.to_csv('my_data_reduced.csv', index=False)
```

Final Result:

Finally, the main output is the PCP plot after reordering the axes which is the order obtained from calculation of metrics, heat map and donut chart showing that how the final plot was created.

Elaboration

Code Explanation Describing Algorithm:

1. The project aims to develop a Python-based tool for analyzing high-dimensional data using Parallel Coordinate Plotting (PCP) techniques by using metrics.
2. Initially we plot the given data on Parallel coordinate plot .
3. If the dataset contains more than 15 dimensions, the calculation of metrics becomes very slow, so the components has been reduced to 10 if initially the no.of components are greater than 15 by using pca.
4. The tool allows users to input their data in CSV format and select the set of metrics they want to use for analysis.
5. Now, we clean the data by removing the rows if having any empty entries in a row.
6. Calculating all the 12 metrics for each axis pair and then storing values of each metric in a list and then, Normalize the values in each list.
7. The 12 properties are:
   (a) Clear Grouping
   (b) Density Change
   (c) Split Up
   (d) Neighborhood
   (e) Positive Correlation
   (f) Negative Correlation
   (g) Positive Variance
   (h) Negative Variance
   (i) Positive Skewness
   (j) Negative Skewness
   (k) Fan
   (l) Outliers
8. Take the percentage weights for each metric from the user. So that it calculates weighted value of all the metrics between each axis pair.
9. Here user can give negative weights particularly to a metric so that while calculating it actively ignores that metric. Giving negative weight for a metric means it is giving more weightage to other metrics.
10. Based on the percentage of weights given by users the final ordering will be calculated which will be useful for data storytelling.
11. After taking inputs it create the donut chart of summarizing the weights of each metric considered in 100% of the final value.
12. Then it creates dictionary for each axis contains the final value for each other axis.
13. Then we create a heat map representing the values for each axis pair values which summarize the final order of the axes pair based on how high the respective values will be.
14. The ordering is taken such that high value in heat map is considered as 1st axes and the corresponding column as 2nd axes, then consider the 3rd axes at which the 2nd axes have high value and so on.
15. Then we reorder the columns of data based on the final values of calculation of metric and then we obtain the final PCP plot after reordering axes.
16. The tool is especially useful for data analysis in fields such as finance, healthcare, and marketing, where large amounts of data are common.
17. In summary, the code provides a powerful and intuitive way to analyze high-dimensional data and discover hidden patterns and relationships that would be hard to spot otherwise.