Statistical Analysis & Machine Learning

Summary VAT paper

VAT - A Tool for Visual Assessment of (Cluster) Tendency

Students:

1. Lam NGUYEN
2. Chansodara DY
3. Liam VARGAS
4. Florian
5. Huong TA

Terms and definitions

Cluster: A set of objects with considerable similarity between each other, while being dissimilar from other objects in other clusters.

Clustering: Clustering is an unsupervised machine learning technique designed to group unlabeled examples based on their similarity to each other.

Single-linkage clustering: In statistics, single-linkage clustering is one of several methods of hierarchical clustering. It is based on grouping clusters in bottom-up fashion (agglomerative clustering), at each step combining two clusters that contain the closest pair of elements not yet belonging to the same cluster as each other.

VAT: Visual Assessment Tool it's a visual approach for assessing cluster tendency that displays a reordered form of dissimilarity data as an intensity image. Clusters are indicated by dark blocks of pixels along the diagonal.

Ordered Dissimilarity Images (ODI): visual depictions of the range of dissimilarity between observations

Minimal/Minimum Spanning Tree (MST): A spanning tree is the minimal set of edges linking all vertices in a graph, essentially a proof that the graph is connected. The minimum weight spanning tree serves as the sparsest possible representation of the structure of the graph, making it useful for visualization.

1. Summary

The paper includes 4 sections:

Section 1 Define the problem:

- In a dataset, there are groups of points (data) that are more similar to each other than others — these points naturally cluster together without being forced
  Clustering is an unsupervised learning technique; the data is not labeled. The goal of clustering is to find hidden structures and natural groups in the data. Even for supervised learning, clustering is a preprocessing step to train the model.
- Many clustering algorithms (like K-means) always cluster, even if the data does not have clear clusters. Tools like VAT help you check in advance whether to apply to cluster or not, avoiding clustering bias: "*The ultimate purpose of imputing data here is simply to get a very rough picture of the cluster tendency in O*".

Section 2 Define the main idea of the VAT approach and then give a description of how it can be implemented.

- Create a dissimilarity matrix between the data points
- Rearrange the order of the points so that similar points are closer together.
- Display the results as a matrix image (grayscale/black and white) – this is the "visual evaluation" step.

=> If there are dark blocks appearing on the diagonal of the image, the clusters exist.
=> If the image does not have clear blocks, the data may not have a clear clustering.

Section 3 Discusses relatives of VAT.
Group visual display methods into three categories:

- Visual displays of clusters
- Visual displays to find clusters
- Visual displays to assess tendency

There have been a number of related articles written about how to classify these clusters; however, the classification is not yet completely accurate.

Section 4 Gives a series of examples using various real and artificial data sets that illustrate various facets of the VAT tool.

2. The application of the VAT

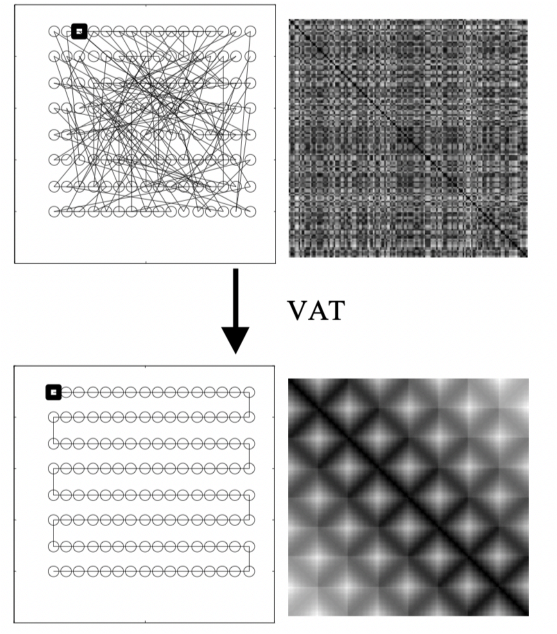


Image 1. Before and After applying VAT

The main goal of VAT is to help users determine whether data has a cluster structure or not before applying clustering algorithms such as K-means, DBSCAN:

- Avoid applying incorrect clustering to data that does not have a cluster structure.
- Allow for the estimation of the number of clusters in the data.
- Useful in the preprocessing stage before choosing a specific clustering algorithm.

3. Implementation: how you implemented

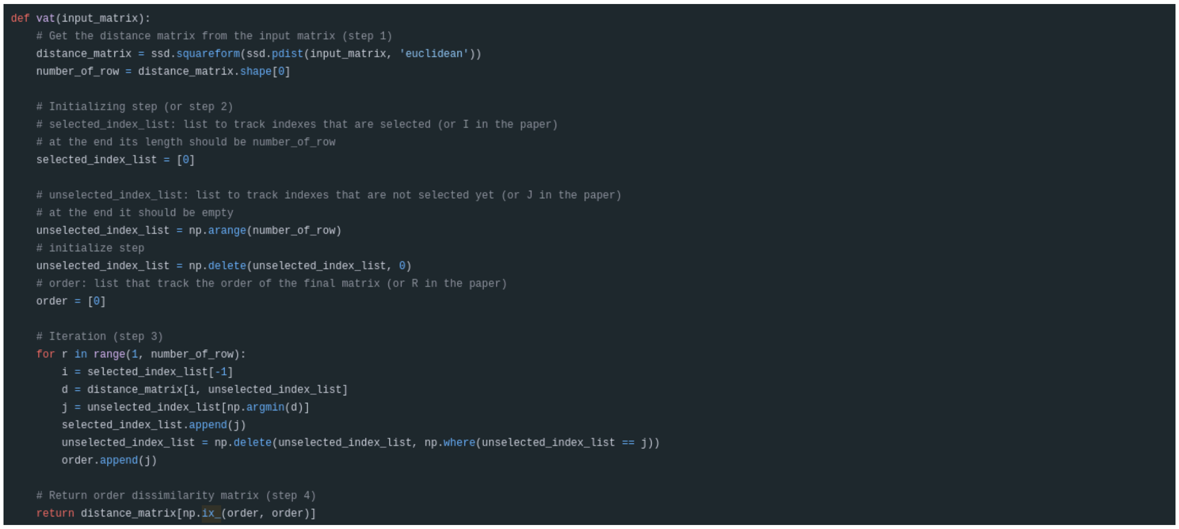
For the implementation and testing of VAT, the sequence of steps described in the article "VAT: A Tool for Visual Assessment of (Cluster) Tendency", specifically the "VAT Ordering and Display Algorithm", was used as a basis. A Python script was developed using version 3.9 along with the numpy, scipy, and matplotlib packages.

A function named *vat (Image 2)* was defined, which takes as input the original data matrix to be processed and returns the order dissimilarity matrix. Additionally, a function called *save_map (Image 3)* was implemented with the input parameters map_data, fig_name, and fig_type, which is used to save the image generated by the vat function's processing.

Within the *vat* function, a square matrix of Euclidean distances is computed from the original data matrix. Then, three arrays are initialized to keep track of the selected, unselected, and final order of the dissimilarity pairs. The process begins by selecting the pair with the maximum dissimilarity value and iteratively continues by comparing the remaining pairs, selecting the one with the closest dissimilarity to the previously selected one.

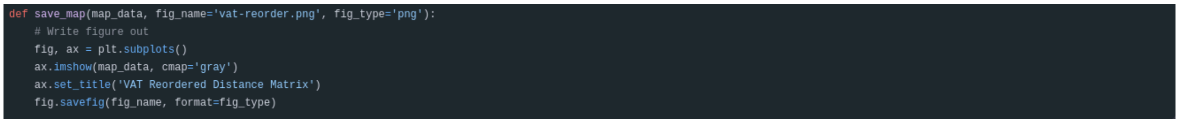
Finally the function returns an ordered dissimilarity matrix with the computed order of dissimilarities.

To visualize the results (*Image 4*), the function *save_map* is called by setting the returned ordered dissimilarity matrix, name and type of the image as input arguments. A gray color map is given to the visualization to complete the VAT.



```python
def vat(input_matrix):
    # Get the distance matrix from the input matrix (step 1)
    distance_matrix = ssd.squareform(ssd.pdist(input_matrix, 'euclidean'))
    number_of_row = distance_matrix.shape[0]

    # Initializing step (or step 2)
    # selected_index_list: list to track indexes that are selected (or I in the paper)
    # at the end its length should be number_of_row
    selected_index_list = [0]

    # unselected_index_list: list to track indexes that are not selected yet (or J in the paper)
    # at the end it should be empty
    unselected_index_list = np.arange(number_of_row)
    # initialize step
    unselected_index_list = np.delete(unselected_index_list, 0)
    # order: list that track the order of the final matrix (or R in the paper)
    order = [0]

    # Iteration (step 3)
    for r in range(1, number_of_row):
        i = selected_index_list[-1]
        d = distance_matrix[i, unselected_index_list]
        j = unselected_index_list[np.argmin(d)]
        selected_index_list.append(j)
        unselected_index_list = np.delete(unselected_index_list, np.where(unselected_index_list == j))
        order.append(j)

    # Return order dissimilarity matrix (step 4)
    return distance_matrix[np.ix_(order, order)]
```

Image 2. vat Python function



```python
def save_map(map_data, fig_name='vat-reorder.png', fig_type='png'):
    # Write figure out
    fig, ax = plt.subplots()
    ax.imshow(map_data, cmap='gray')
    ax.set_title('VAT Reordered Distance Matrix')
    fig.savefig(fig_name, format=fig_type)
```

Image 3. save_map Python function

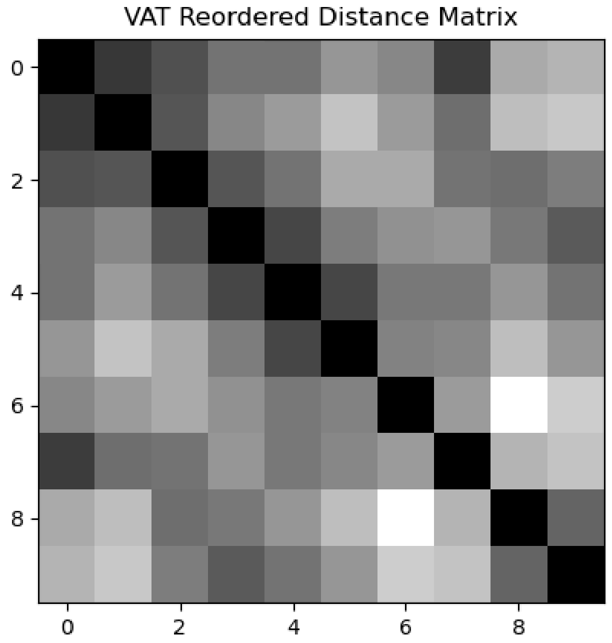


Image 4. Visualization of VAT reordered distance matrix

To further delve into the VAT’s usefulness in terms of visualization for data exploration, here are the results for the well known dataset “Iris” as input:

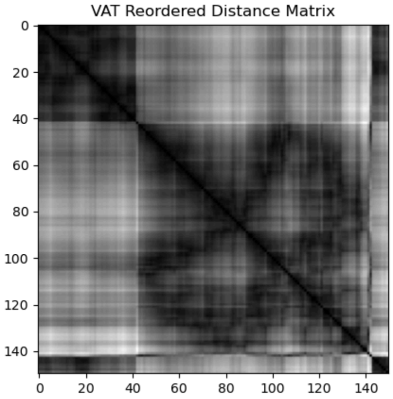


Image 4. Iris VAT scaled

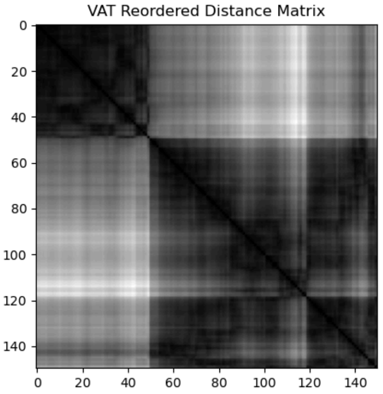


Image 5. Iris VAT unscaled

Notice how Iris setosa is in its own cluster and for Iris verginica and Iris versicolor they are overlapping.

4. Conclusion: What you learned from the paper

Although there are more popular techniques for visualizing data clusters, VAT provides an easy-to-digest alternative in terms of both implementation and practical use. Perhaps just because it is not widely adopted by the "industry," it is a good time to start using it for simple data exploration tasks and evaluate its effectiveness, before applying it to more complex tasks (as the user sees fit).

5. References

Bezdek, J. C., & Hathaway, R. J. (n.d.). VAT: A tool for visual assessment of (cluster) tendency. *Department of Computer Science, University of West Florida; Mathematics and Computer Science Department, Georgia Southern University*.

Skiena, S. S. (2017). *The Data Science Design Manual*. Springer.

Single-linkage clustering. (n.d.). *Wikipedia*. Retrieved April 2, 2025, from https://en.wikipedia.org/wiki/Single-linkage_clustering

Cambridge University Press. (n.d.). *Excerpt from The Data Science Design Manual*. Retrieved from https://assets.cambridge.org/97811087/93384/excerpt/9781108793384_excerpt.pdf