Visualization of Big Data Volumes - project

Marcin Świątkowski
Patryk Krukowski

June 18, 2022

1 Introduction

The purpose of our project is to visualize three datasets (FMNIST, RCV1 Reuters, SMALLNORB) using six methods:

- HUMAP,
- UMAP,
- PaCMAP,
- ISOMAP,
- TriMAP,
- IVHD.

Let's summarize what kind of information is located in considered datasets:

- FMNIST - Fashion MNIST dataset, which contains 70,000 grayscale images in 10 categories. The images show individual articles of clothing at low resolution (28×28 pixels).
- RCV Reuters - is a benchmark dataset on text categorization. It is a collection of newswire articles produced by Reuters in 1996-1997. It contains 804,414 manually labeled newswire documents, and categorized with respect to three controlled vocabularies: industries, topics and regions.

- SMALLNORB - this dataset is intended for experiments in 3D object recognition from shape. It contains images of 50 toys belonging to 5 generic categories: four-legged animals, human figures, airplanes, trucks, and cars. The objects were imaged by two cameras under 6 lighting conditions, 9 elevations (30 to 70 degrees every 5 degrees), and 18 azimuths (0 to 340 every 20 degrees).

Every dataset from above is saved as a csv file.

In the section number 2 we will take a theoretical view onto these methods to understand their key concepts. Then we will use them to visualize mentioned datasets using Python. We will also compare these methods among them, and we will try to find any correlation between type of dataset and visualization method.

In the section number 3 we are going to show visualizations which were done using Python, analyse metrics and measure time of code working.

2 Visualization methods

In this section we want to take a view into theoretical aspects of six visualization methods: HUMAP, UMAP, PaCMAP, ISOMAP, TriMAP and IVHD. The purpose of that section is to present for reader key concepts and main ideas which are beyond of the mentioned methods.

2.1 HUMAP

This technique consists of two main components: Hierarchy construction and Projection. First, in the hierarchy construction component, HUMAP uses a tree-like structure on the high- dimensional dataset using a similarity measure among landmarks. Then, in the projection component, the method embed the hierarchy levels according to the user's demand for more detailed information.

The first step for constructing a hierarchy from bottom to top consists of using a kernel function to find connection strengths (local affinities) of a k-nearest neighbor graph from the data points in the high-dimensional space $\mathbf{R}^m$.

Then, the method uses Finite Markov Chain to find the most visited nodes, which consists of the landmarks for the superior level. Using local and global neighborhood information of each landmark, the similarity between each pair of landmarks is defined as the intersection of these two neighborhoods. Finally, a new neighborhood graph is created using the computed similarity to define a new hierarchy level. Then, for projecting hierarchy levels, the neighborhood graph is first symmetrized, so the strength of each edge helps in the process of finding a suitable position in the low-dimensional representation. With exception to the top hierarchy level, the projection of low levels is influenced by the low-dimensional positions of data points in higher levels for mental map preservation.

More information you can find in [**1**] **HUMAP: Hierarchical Uniform Manifold Approximation and Projection** by Wilson E. Marcílio-Jr, Danilo M. Eler, Fernando V. Paulovich, Rafael M. Martins (2021).

2.2 PaCMAP

The PacMAP technique can be used to dimensionality reduction as well as to visualization (in a contrast to the t-SNE for example). One of the biggest advantages of PaCMAP is that it captures both local and global structure of the data in original space. As we can see in the algorithm, PaCMAP uses three kinds of pairs of points to optimize the low dimensional embedding:

- neighbor pairs,
- mid-near pair,
- further pairs.

Neighbor pairs are just exactly what they mean. Mid-near points are constructed by sampling 6 for each observation, choosing the second closest of the 6, and pair it with i-th observation. The further pairs consist of points which are non-neighbors.

Below we present an algorithm of that method, which we can find in [**2**] **Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for**

Data Visualization by Yingfan Wang, Haiyang Huang, Cynthia Rudin, Yaron Shaposhnik (2021).

Inputs:

- X - high-dimensional data matrix.
- n_{NB} - the number of neighbor pairs (default values: $n_{NB} = 10$).
- MN_{ratio}, FP_{ratio} - the ratio between the number of mid-near pairs and further pairs with to the number of neighbor pairs (default values: $MN_{ratio} = 0.5$, $FP_{ratio} = 2$).
- $n_{iterations}$ - the number of gradient steps (default value: $n_iterations = 450$).
- init - initialization procedure for the lower dim. embedding (default init = PCA, alternatively, init = random, which initializes Y using the multivariate Normal distribution $N(0, 10^4 I)$, with I denoting the two-dimensional identity matrix).
- τ_1, τ_2, τ_3 - beginning of the three optimization phases, satisfying $\tau_1 = 1 \leq \tau_2 \leq \tau_3 \leq n_{iterations}$ (default values: $\tau_1 = 1, \tau_2 = 101, \tau_3 = 201$).
- w_{NB}, w_{MN}, w_{FP} – the weights associated with neighbor, mid-near, and further pairs at iteration t.

Ensure:

- Y - low-dimensional data matrix.
- for $i = 1$ to N do:
    1. construct n_{NB} neighbor edges by computing the $n_N B$ nearest neighbors of x_i using scaled distances $d^{2,select}$. To take advantage of existing implementations of k-NN algorithms, for each sample we first select the $min(nNB + 50, N)$ nearest neighbors according to the Euclidean distance and from this subset we pick the n_{NB} nearest neighbors according to the scaled distance $d^{2,select}$ (recall that N is the total number of observations).

2. construct $n_{MN} = \lfloor n_{NB} \times MN_{ratio} \rfloor$ mid-near pairs. For each pair, construct it by sampling 6 observations, using x i and the 2nd nearest observation to x i as the mid-near pair.

3. construct $n_{FP} = \lfloor n_{NB} \times FP_{ratio}$ further pairs by sampling non-neighbor points.

end for

- apply the initialization procedure init to set the initial values of Y.
- run AdamOptimizer $n_{iterations}$ iterations to optimize the loss function Loss PaCMAP while simultaneously adjusting the weights according to the above scheme.
- return Y.

2.3 ISOMAP

This technique is a modification of the MDS method. The difference is that MDS uses distance matrix with Euclidean distance, while ISOMAP uses geodesic distance. It is more appropriate approach in context of discovering shapes of the datasets which are non-convex.

High level description of the algorithm:

1. Use a KNN approach to find the k nearest neighbors of every data point.
2. Once the neighbors are found, construct the neighborhood graph where points are connected to each other if they are each other's neighbors. Data points that are not neighbors remain unconnected.
3. Compute the shortest path between each pair of data points (nodes). Typically, it is either Floyd-Warshall or Dijkstra's algorithm that is used for this task. Note, this step is also commonly described as finding a geodesic distance between points.
4. Use multidimensional scaling (MDS) to compute lower-dimensional embedding. Given distances between each pair of points are known, MDS

places each object into the N-dimensional space (N is specified as a hyper-parameter) such that the between-point distances are preserved as well as possible.

2.4 TriMAP

TriMap is a dimensionality reduction method that uses triplet constraints to form a low-dimensional embedding of a set of points. The triplet constraints are of the form "point i is closer to point j than point k". The triplets are sampled from the high-dimensional representation of the points, and a weighting scheme is used to reflect the importance of each triplet.

The algorithm tries to find a lower dimensional embedding using batch gradient descent, which preserves the ordering of distances of the triplets. Theoretically, this method manages to include both local and global structure, but in practice it is typically found to be prone to struggle with local structure.

2.5 IVHD

Firstly, it's worth to say that this approach is very quick, because time memory complexity is $O(\alpha \cdot M)$, where α is a constant. It is achieved by the substituting distances matrix with the nearest neighbor nn graph data structure (with small nn) and by assuming binary distance between data vectors. High level description of the algorithm:

1. Construct nn-graph, which approximate n-dimensional non-Cartesian manifold immersed in $\mathbf{R}^N$.

2. Use the fast procedure of graph visualization.

More information we can find in 2-D space presented in work **[3] 2-D Embedding of Large and High-dimensional Data with Minimal Memory and Computational Time Requirements** by Witold Dzwinel, Stan Matwin, Rafał Wcisło.

2.6 Conclusions

On the end of this chapter we want to divide methods mentioned above between two categories - methods that preserve and don't preserve global and local structure of original space.

Table 1: Comparision of visualization methods in the context of global and local structure capturing

Visualization method	Does preserve global structure of original space?	Does preserve local structure of original space?
UMAP	Yes	Yes
HUMAP	Yes	Yes
PaCMAP	Yes	Yes
IsoMAP	Yes	Yes
TriMAP	Yes	**No**
IVHD	Yes	Yes

We will verify this at the end of the analysis of each set using metrics.

3 Case study

Our goal in this section is to check a performance of the method described in 2 on the mentioned earlier datasets.

3.1 FMNIST

We show embedded visualizations of FMNIST dataset. Then we discuss about results with respect to computed metrics.

3.1.1 Embeddings

- UMAP - we use full dataset. We can observe that UMAP has worked very well embedding data in 2D space with a small problem with distinc-

tion between following classes: T_shirt, Trouser, Dress, Coat. It means that the data points from these classes have similar values of conditional probabilities.

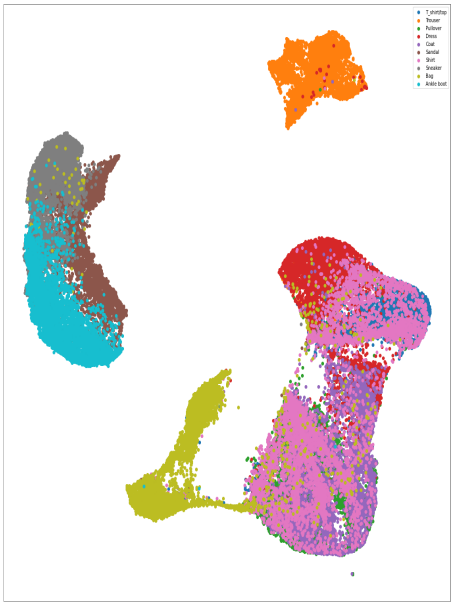
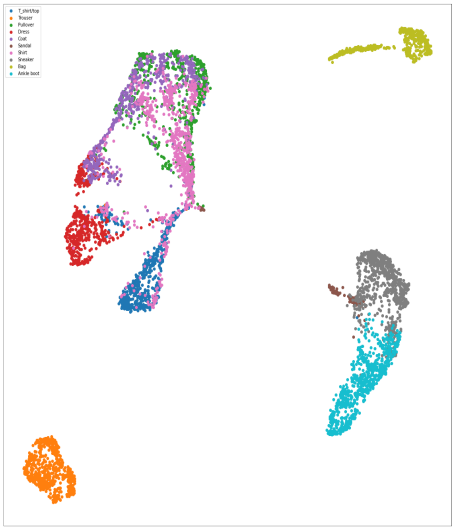
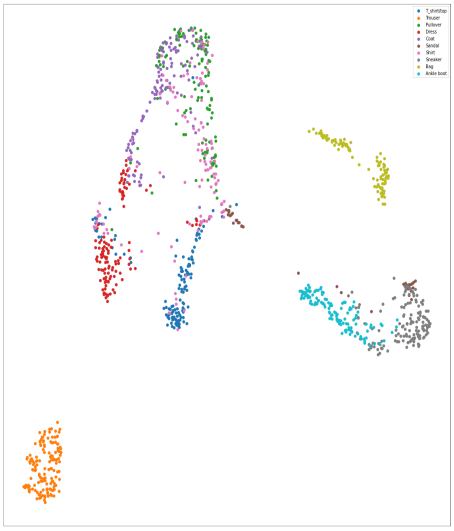


Figure 1: FMNIST dataset embedded in 2D space by UMAP

- HUMAP - we use $30,000$ observations from FMNIST dataset because of computional cost. The first level of hierarchy for the HUMAP method is visually better because more observations of the original dataset were used to construct it. This method has trouble with distinguishing between a shirt and a pullover.

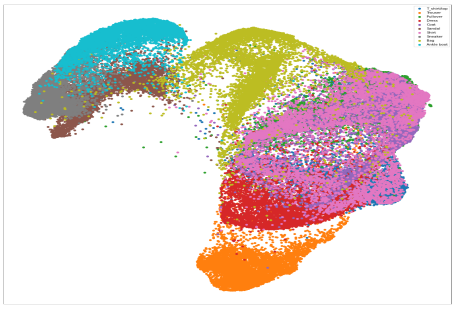


(a) FMNIST dataset embedded in 2D space by HUMAP on the first level of hierarchy

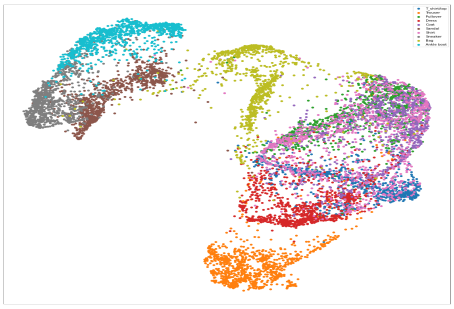


(b) FMNIST dataset embedded in 2D space by HUMAP on the second level of hierarchy

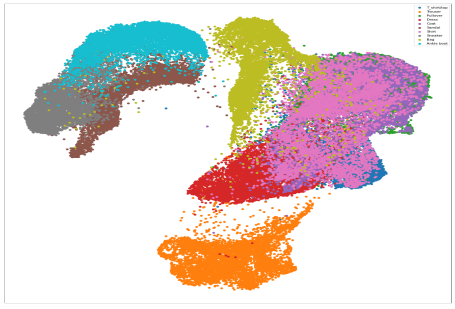
- TriMAP - we use full dataset. In TriMAP, all the used metrics give similar results. We observe a problem with the visualization of pullover, shirt and snekaers classes.



(a) FMNIST dataset embedded in 2D space by TriMAP (cosine distance)



(b) FMNIST dataset embedded in 2D space by TriMAP (euclidean distance)



(c) FMNIST dataset embedded in 2D space by TriMAP (manhattan distance)

- PaCMAP - we use full dataset. It produces similar results to the TriMAP method. However, we can observe a greater separation of clusters from each other.

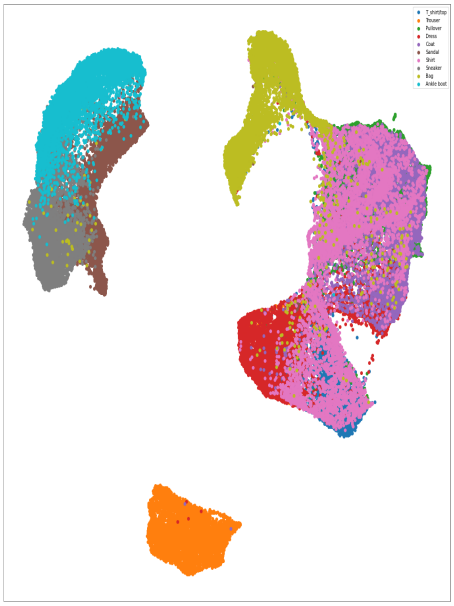
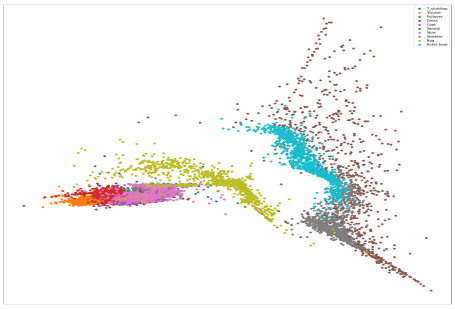
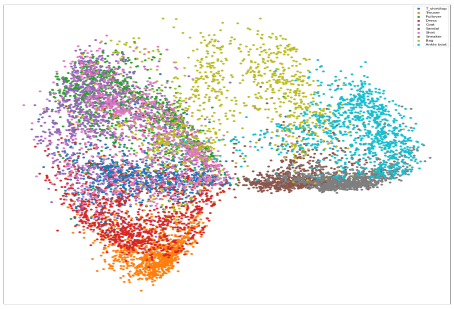


Figure 4: FMNIST dataset embedded in 2D space by PaCMAP

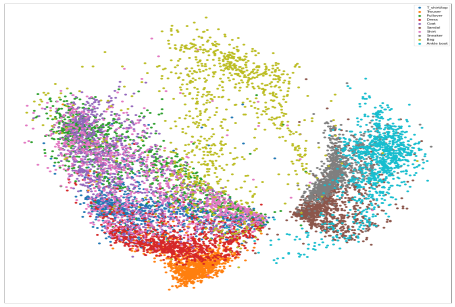
- ISOMAP - we use only $10,000$ observations from FMNIST dataset (computional cost). This method with the metric manhattan and euclidean look quite similar. However, embedding made with their help is of very poor quality. ISOMAP with the cosine metric did a bit better projection of data into 2D space.



(a) FMNIST dataset embedded in 2D space by ISOMAP (cosine distance)

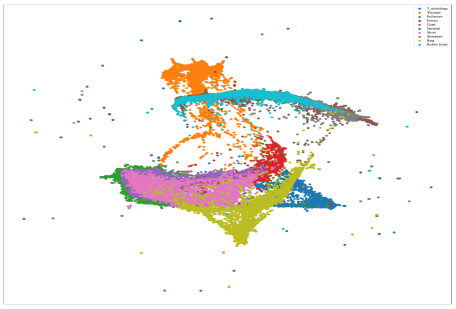


(b) FMNIST dataset embedded in 2D space by ISOMAP (euclidean distance)

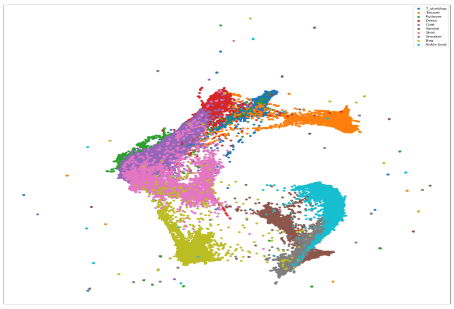


(c) FMNIST dataset embedded in 2D space by ISOMAP (Minkowsky distance)

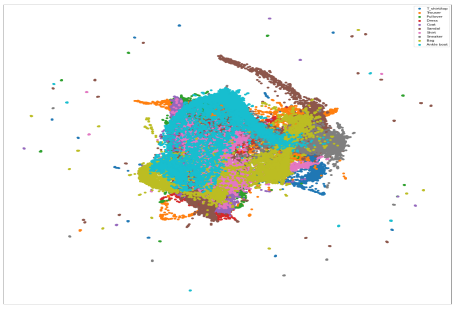
- IVHD for 2 neighbors - we use full dataset to each considered variation of the IVHD method.



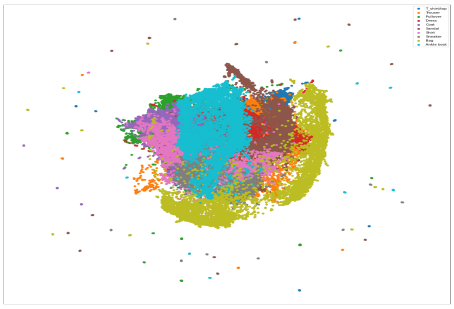
(a) FMNIST dataset embedded in 2D space by IVHD (cosine distance, 2500 iterations)



(b) FMNIST dataset embedded in 2D space by IVHD (euclidean distance, 2500 iterations)

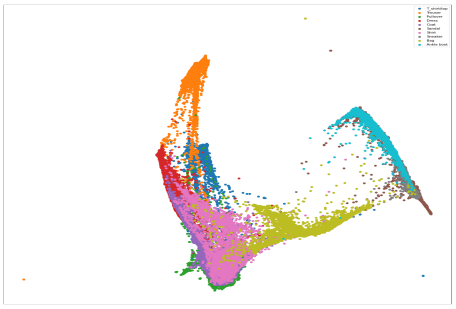


(c) FMNIST dataset embedded in 2D space by IVHD (cosine distance, 200 iterations)

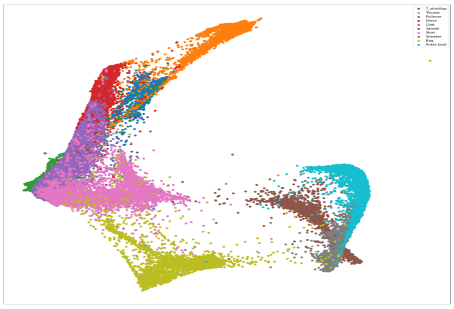


(d) FMNIST dataset embedded in 2D space by IVHD (euclidean distance, 200 iterations)

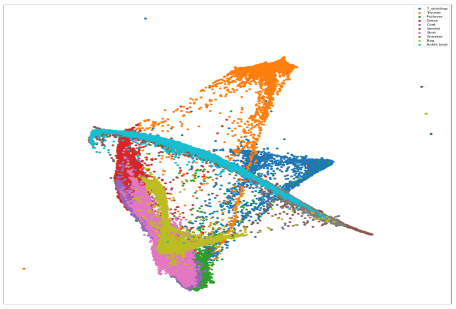
- IVHD for 3 neighbors - we use full dataset.



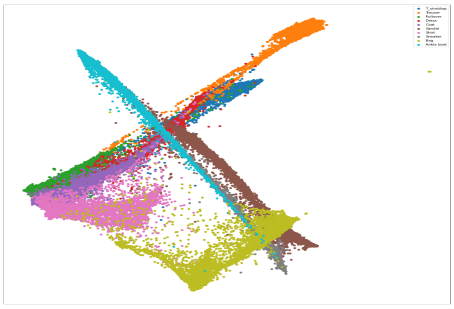
(a) FMNIST dataset embedded in 2D space by IVHD (cosine distance, 2500 iterations)



(b) FMNIST dataset embedded in 2D space by IVHD (euclidean distance, 2500 iterations)



(c) FMNIST dataset embedded in 2D space by IVHD (cosine distance, 500 iterations)



(d) FMNIST dataset embedded in 2D space by IVHD (euclidean distance, 500 iterations)

In our opinion, the best option is the embedding for the following combination of parameters:

- precomputed graph with euclidean distances,
- iterations = 2500,
- nearestNeighborsCount (we should minimize this number) = 2,
- randomNeighborsCount = 1,
- binaryDistances = True,
- reverseNeighborsSteps = 0,
- reverseNeighborsCount = 0,
- l1Steps = 0.

3.1.2 Times

We measured times for comparable methods (with the same shape of data) and the results are as follows:

- UMAP - 1 min 48s,
- TriMAP euclidean - 2 min 10s,
- TriMAP manhattan - 2 min 8s,
- TriMAP cosine - 2 min 2s,
- PacMAP - 1 min 28s,
- IVHD - 6 min 50s.

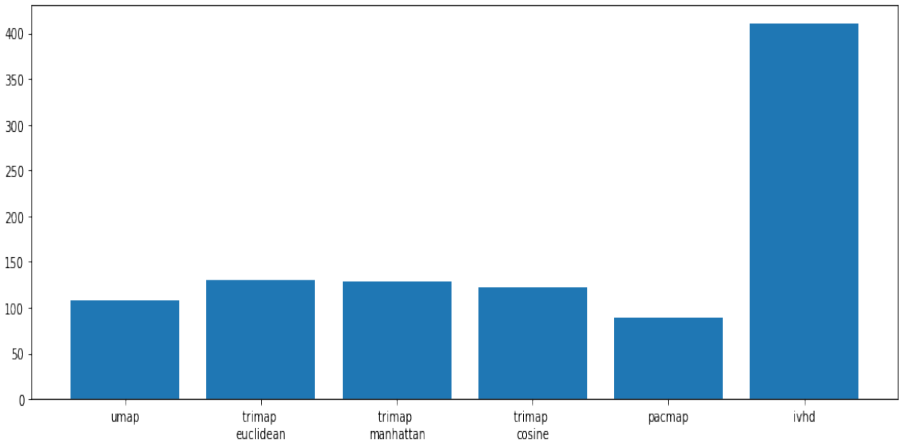


Figure 8: Times for specified methods

The fastest algorithm was PacMAP. One can also spot a significant difference between IVHD and other methods. The main reason is probably a combination of method's parameters - 3500 iterations for 2 neighbors resulted in good embedding but long time of computation.

3.1.3 Metrics

Now we will present the results of the metrics for the methods studied for specific cases.

- DR quality - for full dataset, we computed DR quality for UMAP, PaCMAP, TriMAP and IVHD. In the case of HUMAP and ISOMAP we encountered computational problems. Generally, for any method (up to 1000 neighbors), embeddings improve the quality of data reduction what we can observe on figure 9.

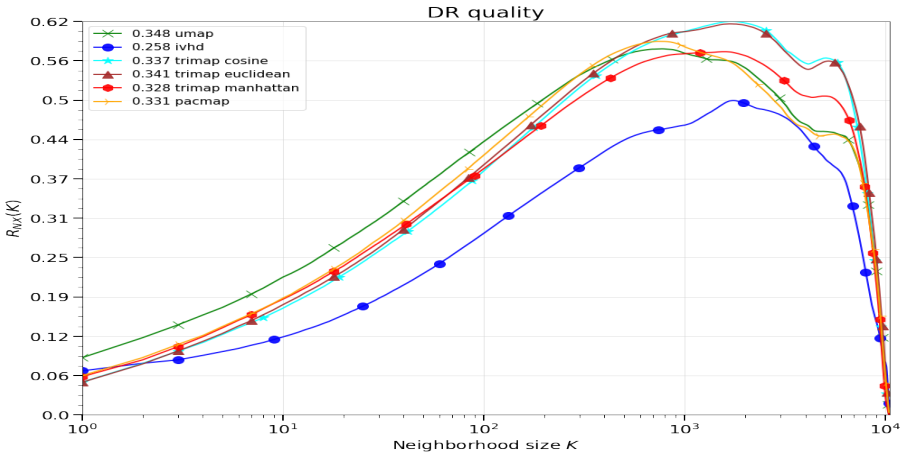


Figure 9: DR quality for specified methods

- KNN gain - for full dataset, we computed KNN gain for UMAP, PaCMAP, TriMAP and IVHD. In the case of HUMAP and ISOMAP we encountered computational problems. The highest KNN value (figure 10) value is achieved for the IVHD (euclidean distance, 2 neighbors and 2500 iterations).

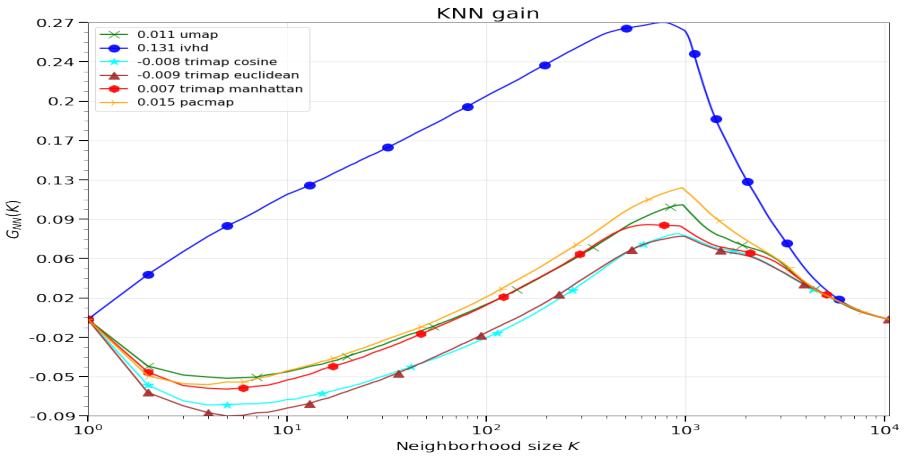
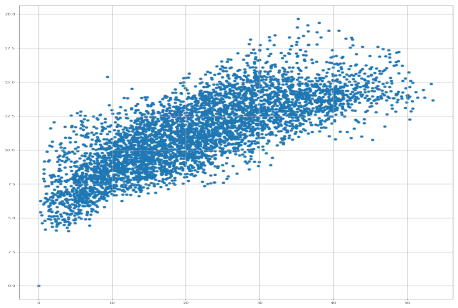
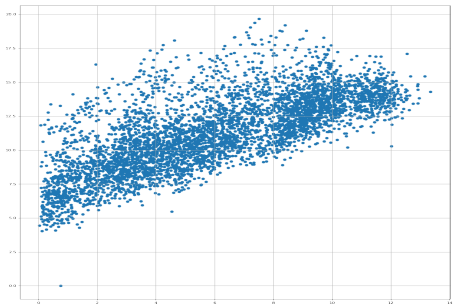


Figure 10: KNN gain for specified methods

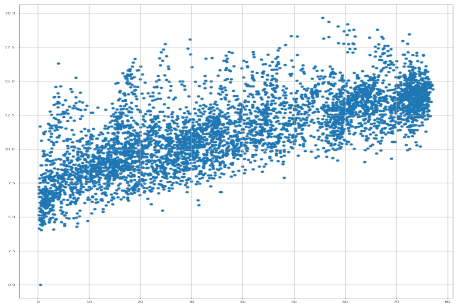
- Shepard's diagrams - computed for 500 observations, in all the methods examined below, we see that the point clouds are arranged along a straight line. This proves that the distance between points in the original data set is well represented. We did not make a Shepard diagram for the IVHD method as the results were very unsatisfactory. This may lead to the conclusion that IVHD may not be a good match for the global nature of the original data (in a context of the FMNIST dataset).



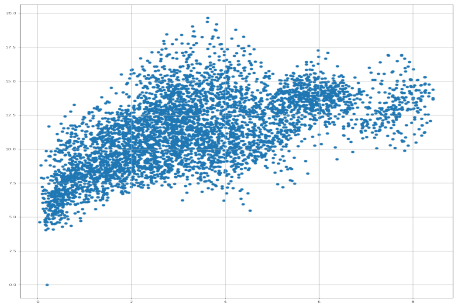
(a) FMNIST dataset embedded in 2D space by ISOMAP (euclidean distance)



(b) FMNIST dataset embedded in 2D space by PacMAP



(c) FMNIST dataset embedded in 2D space by TriMAP (euclidean distance)



(d) FMNIST dataset embedded in 2D space by UMAP

- Trustworthiness - We used trustworthiness implemented in *sklearn.manifold*. Its value is between 0 and 1 and the closer to 1, the better. It captures local structure of original space. The results are presented in the figure 12.

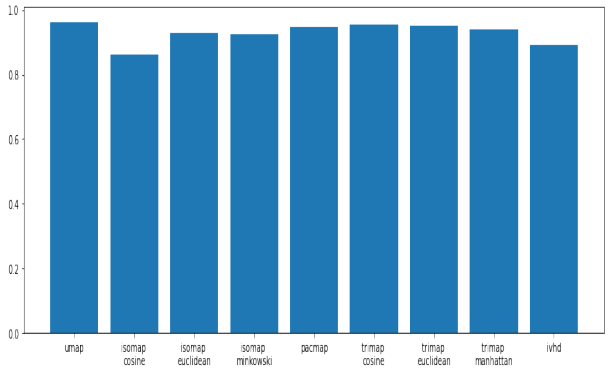


Figure 12: Trustworthiness calculated for FMNIST dataset for following methods: UMAP, IsoMAP cosine, IsoMAP euclidean, IsoMAP minkowski, PacMAP, TriMAP cosine, TriMAP euclidean, TriMAP manhattan and IVHD

3.2 Smallnorb

Now we are going to present embedded visualizations of Smallnorb dataset. Then we are going to discuss about results with respect to computed metrics.

3.2.1 Embeddings

- UMAP - we use full dataset.We observe that UMAP has a lot of trouble embedding the original dataset into 2D space. The original dimension of the data may be so high that UMAP is not able to accuratelly map the original relationships and structure.

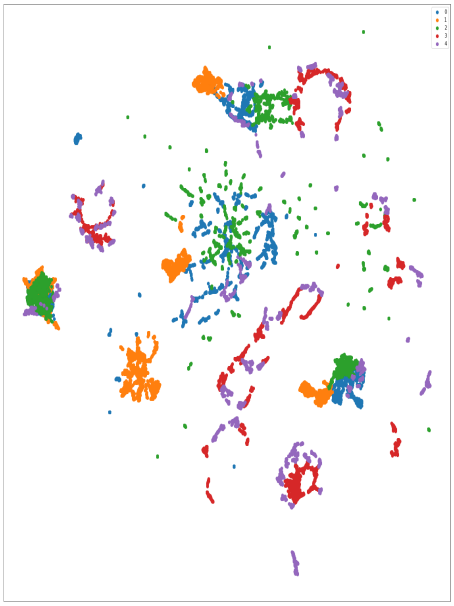


Figure 13: Smallnorb dataset embedded in 2D space by UMAP

- HUMAP - we use only $10,000$ observations from Smallnorb dataset due to the high memory complexity of this method. Figure no. 14 shows 4 clusters distinguished by HUMAP, while in these clusters there are mixed classes of the analyzed data set. Consequently, this method failed to reduce dimensionality. We omitted the visualization of the subsequent layers of the HUMAP method, as they did not add anything to the conclusions we already have.

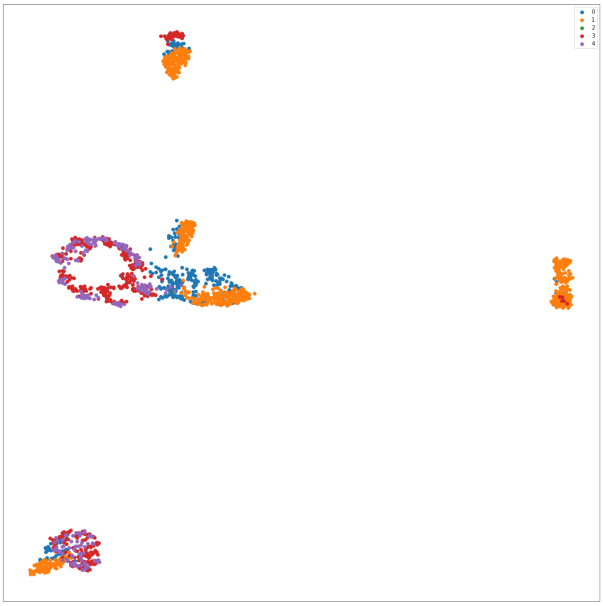
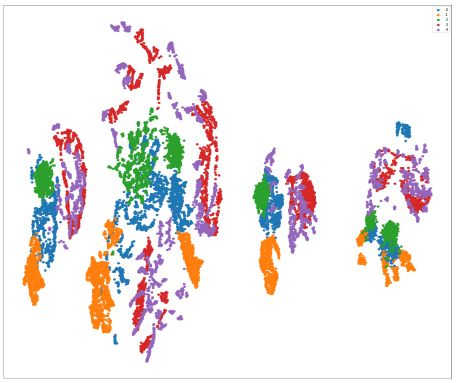
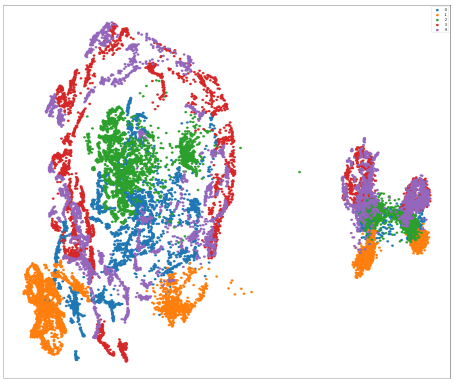


Figure 14: Smallnorb dataset embedded in 2D space by HUMAP on the first level of hierarchy

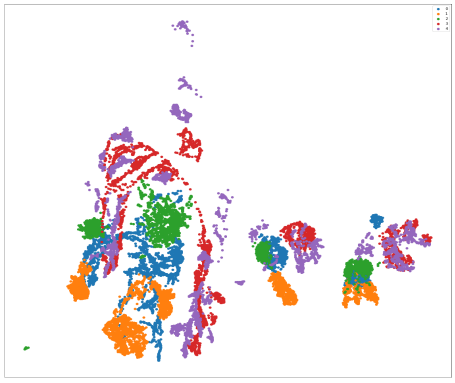
- TriMAP - we use full dataset. The Euclidean and Manhattan metrics give very similar results. In our opinion, the TriMAP for the Euclidean metric works well in terms of the separation of individual classes from each other - the clusters of data points do not intersect as much as was visible using the two previous methods.



(a) Smallnorb dataset embedded in 2D space by TriMAP (euclidean distance)



(b) Smallnorb dataset embedded in 2D space by TriMAP (cosine distance)



(c) Smallnorb dataset embedded in 2D space by TriMAP (manhattan distance)

- PaCMAP - we use full dataset. It produces similar results to the TriMAP method again. However, we can observe a greater separation of clusters from each other. However, the shape of the clusters of points is quite different.

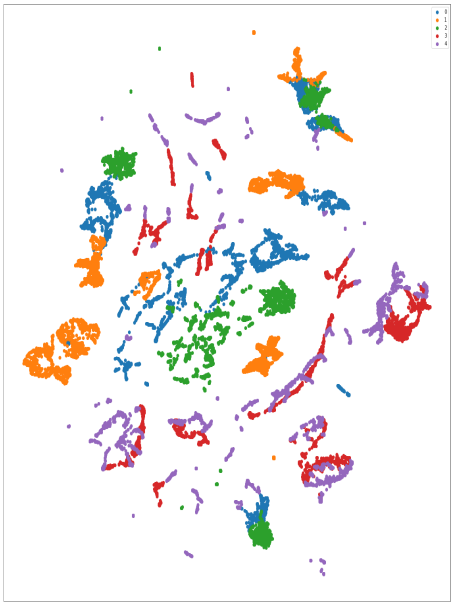
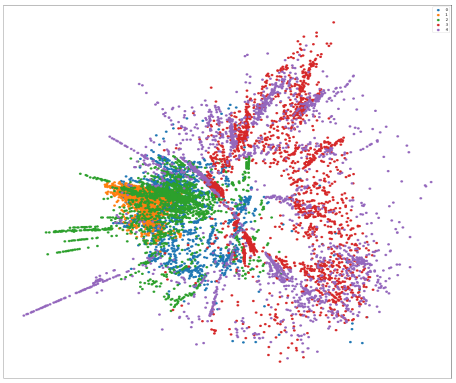
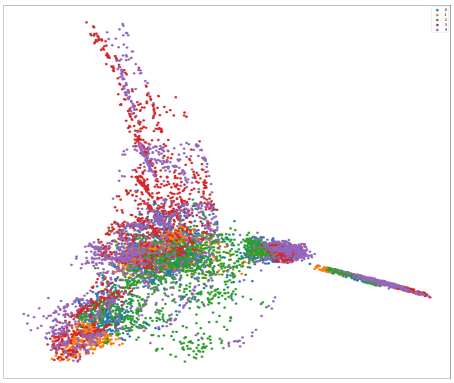


Figure 16: Smallnorb dataset embedded in 2D space by PaCMAP

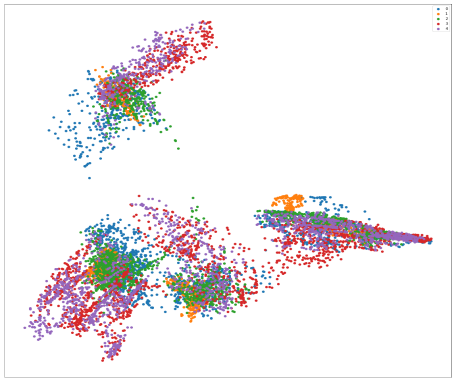
- ISOMAP - we use only $10,000$ observations from Smallnorb dataset (computional cost). This method has a quite good performance in our opinion.



(a) Smallnorb dataset embedded in 2D space by ISOMAP (cosine distance)

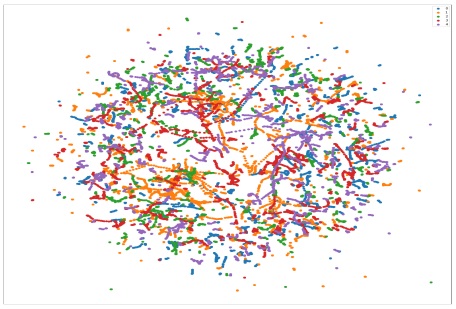


(b) Smallnorb dataset embedded in 2D space by ISOMAP (euclidean distance)

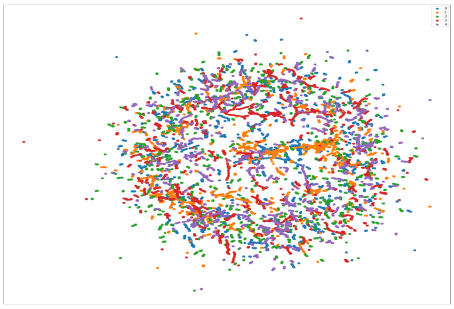


(c) Smallnorb dataset embedded in 2D space by ISOMAP (Minkowsky distance)

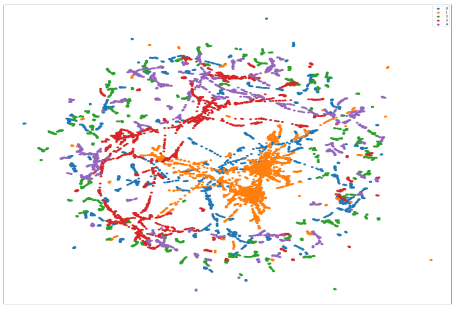
- IVHD for 2 and 3 neighbors - we use full dataset. This method is not able to project Smallnorb dataset into 2D space in the sensible way.



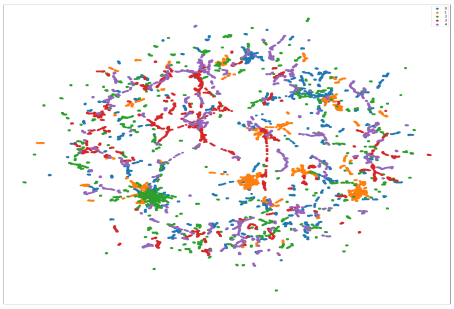
(a) Smallnorb dataset embedded in 2D space by IVHD (cosine distance, 2 neighbors, 3500 iterations)



(b) Smallnorb dataset embedded in 2D space by IVHD (euclidean distance, 2 neighbros, 3500 iterations)



(c) Smallnorb dataset embedded in 2D space by IVHD (cosine distance, 3 neighbros, 3500 iterations)



(d) Smallnorb dataset embedded in 2D space by IVHD (euclidean distance, 3 neighbors, 3500 iterations)

We used the same parameters as for the FMNIST dataset. However, we changed the number of iterations to 3500.

3.2.2 Times

Now we will show the times of operation of individual methods.

- UMAP, full dataset: 2 min 26s,
- HUMAP, small dataset (30 000 samples): 8 min 43s,
- TriMap euclidean distance, full dataset: 1 min 29s,
- PacMap, full dataset: 42 s,

- Isomap manhattan distance, tiny dataset (10 000 samples): 15 min 58s,
- Isomap euclidean distance, tiny dataset (10 000 samples): 3 min 14s,
- Isomap cosine distance, tiny dataset (10 000) samples: 1 min 58s,
- IVHD euclidean distance, full dataset, 2 neighbors, 3500 iterations: 3 min 32s.

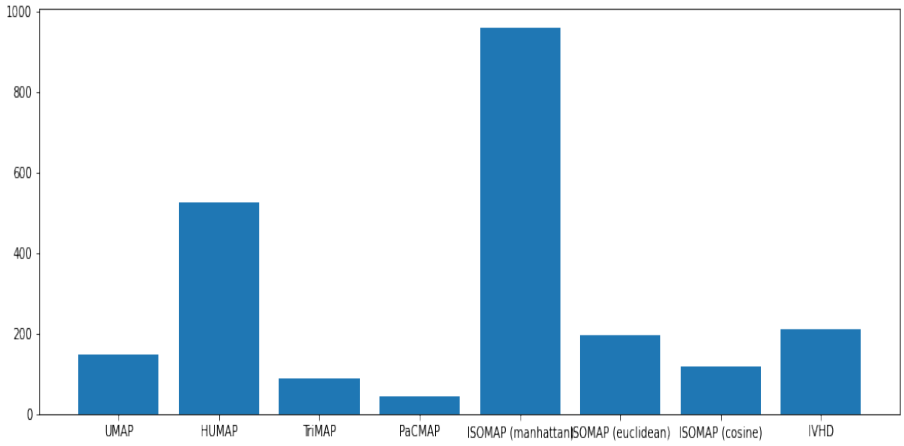


Figure 19: Times for specified methods

3.2.3 Metrics

Now we will present the results of the metrics:

- DR quality - we computed DR quality for UMAP, PaCMAP, TriMAP (manhattan distance) and IVHD (euclidean distance used in precomputed KNN graph, 2 neighbors, 3500 iterations) for full dataset. In the case of HUMAP we encountered problems resulting from the operation of this method. From the figure 20 we see the data reduction quality is not improved for any method (until about 1000 neighbors for UMAP, TriMAP and PaCMAP).

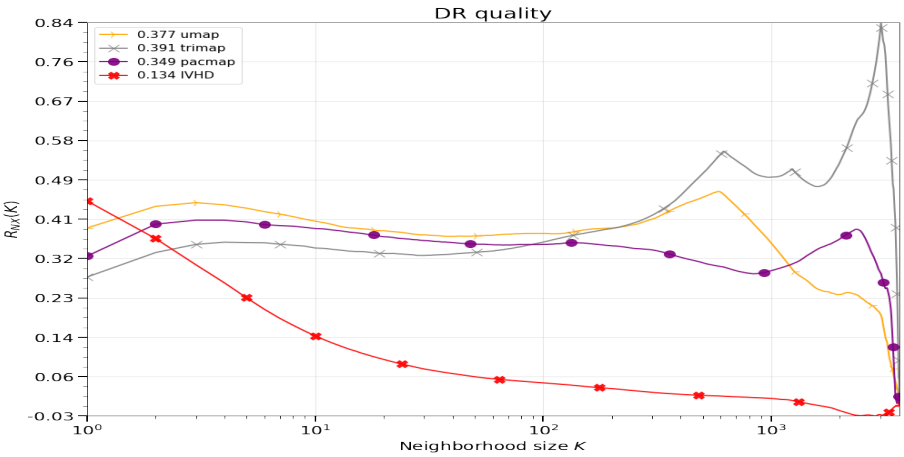


Figure 20: DR quality for specified methods

- KNN gain - we computed KNN gain for UMAP, PaCMAP, TriMAP (manhattan distance) and IVHD (euclidean distance used in precomputed KNN graph, 2 neighbors, 3500 iterations) for full dataset. In the case of HUMAP we encountered problems mentioned above. The greatest benefit of the KNN classifier is achieved for the PaCMAP method.

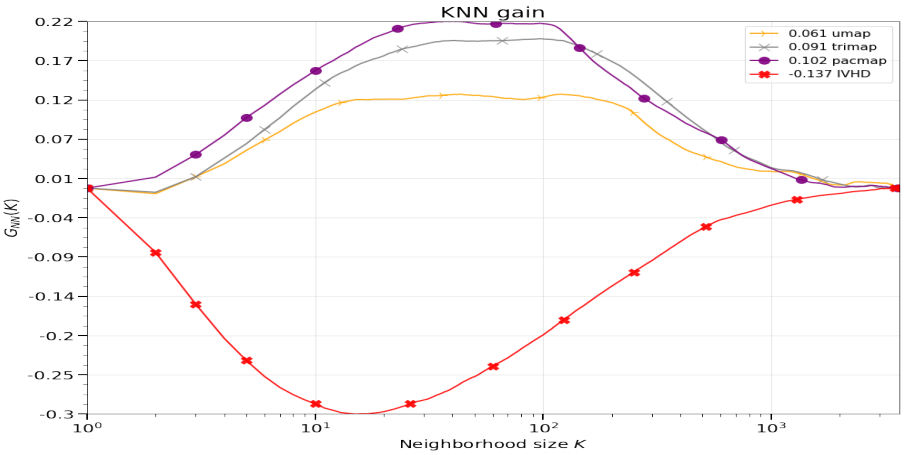
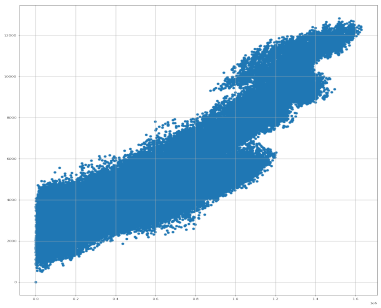


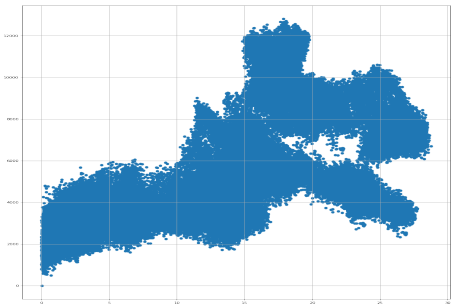
Figure 21: KNN gain for specified methods

- Shepard's diagrams - in all the methods examined below, we see that the point clouds are regular in case of TriMAP and ISOMAP methods and

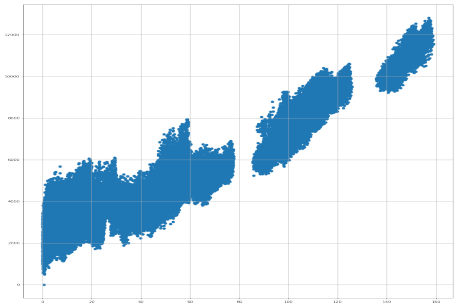
irregular in case of UMAP and PaCMAP. We did not make a Shepard diagram for the IVHD method as the results were very unsatisfactory. This may lead to the conclusion that IVHD may not be a good match for the global nature of the original data.



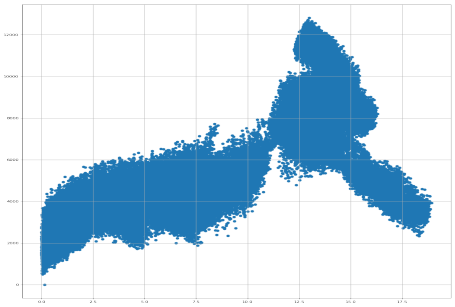
(a) Smallnorb dataset embedded in 2D space by ISOMAP (euclidean distance)



(b) Smallnorb dataset embedded in 2D space by PacMAP



(c) Smallnorb dataset embedded in 2D space by TriMAP (euclidean distance)



(d) Smallnorb dataset embedded in 2D space by UMAP

- Trustworthiness

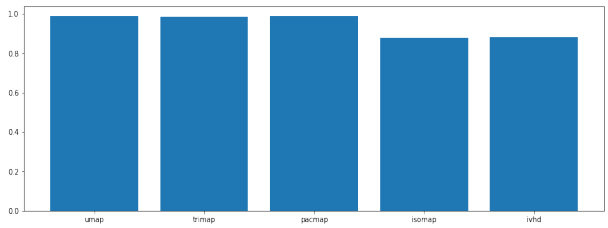


Figure 23: Trustworthiness calculated for Smallnorb dataset for following methods: UMAP, IsoMAP euclidean, PacMAP, TriMAP euclidean, Trimap euclidean and IVHD (euclidean distance, 2 neighbors, 3500 iterations)

3.3 RCV Reuters

Now we are going to present embedded visualizations of Reuters dataset. Then we are going to discuss about results with respect to computed metrics.

3.3.1 Embeddings

- UMAP - we use full dataset. We can not see any regular patterns in embedded data. There are several small clusters but comparing to the size of the whole dataset and number of categories they are not significant.

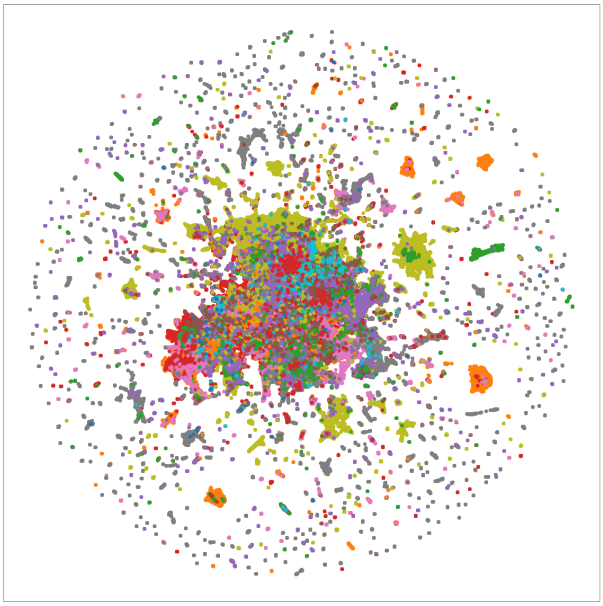


Figure 24: Reuters dataset embedded in 2D space by UMAP

- HUMAP - we use only $10,000$ observations from Reuters dataset due to the high memory complexity of this method. Figure 25 shows the first layer of hierarchy of the HUMAP method. We can see no patterns nor clusters of data points.

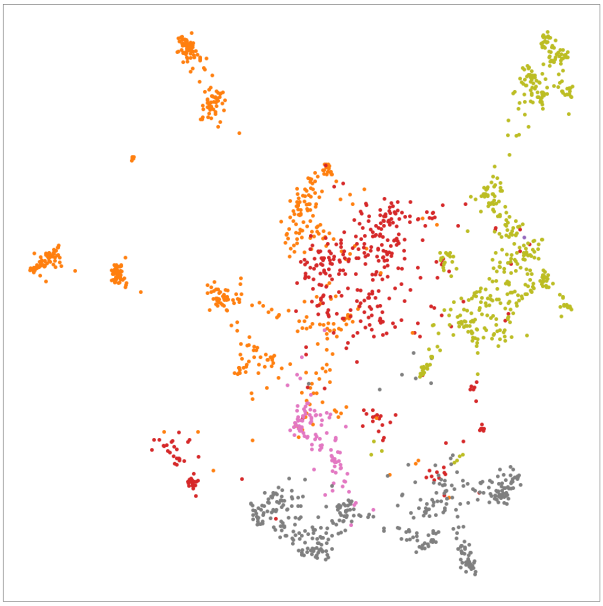
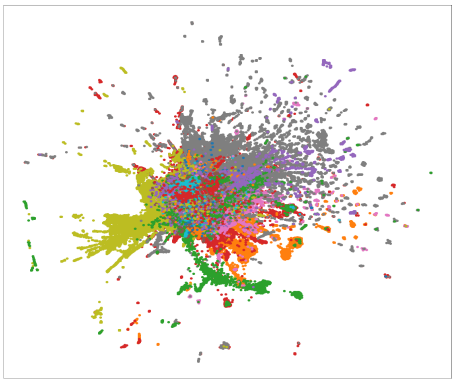
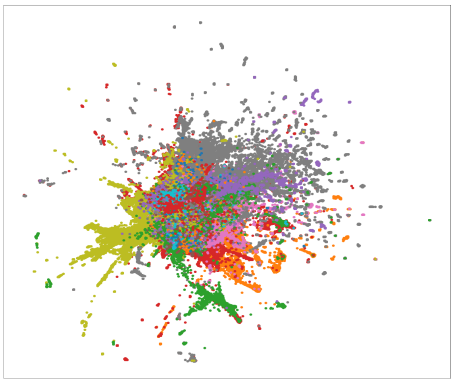


Figure 25: Reuters dataset embedded in 2D space by HUMAP on the first level of hierarchy

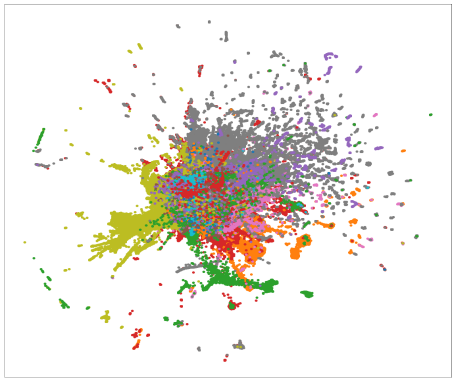
- TriMAP - we use full dataset. Points are still cumulated near center of plot, but we can spot some relatively big distinguishable clusters. The Euclidean and Manhattan metrics give very similar results. In our opinion, the TriMAP for the Euclidean metric works well in terms of the separation of individual classes from each other. Only IVHD seems to work better.



(a) Reuters dataset embedded in 2D space by TriMAP (euclidean distance)



(b) Reuters dataset embedded in 2D space by TriMAP (cosine distance)



(c) Reuters dataset embedded in 2D space by TriMAP (manhattan distance)

- PaCMAP - we use full dataset. It produces similar results to the TriMAP method but this time points are more scattered. Otherwise we can spot its attempts to distinguish clusters from big aggregation in center of plot. It is, however, less successful than with TriMAP.

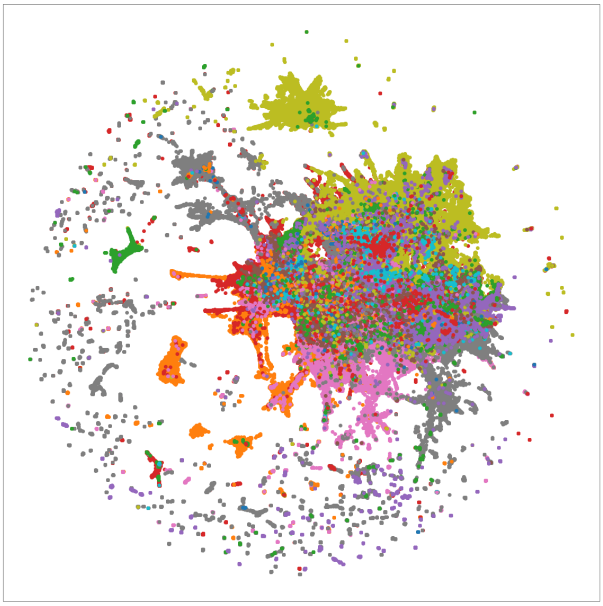
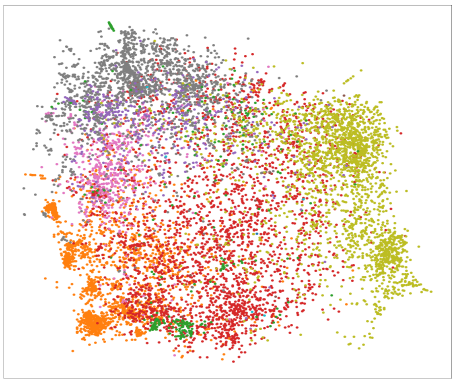
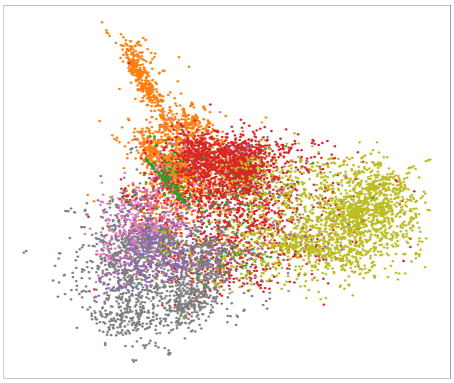


Figure 27: Reuters dataset embedded in 2D space by PaCMAP

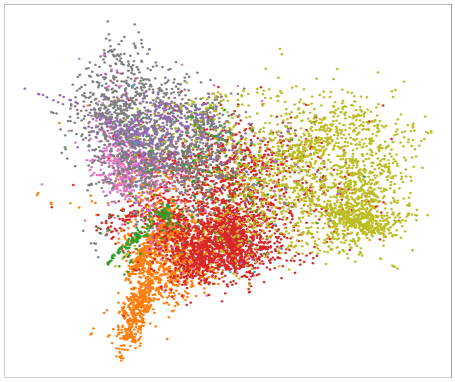
- ISOMAP - we use only $10,000$ observations from Reuters dataset (computational cost). For the reduced dataset ISOMAP presents quite good separation of clusters to the points they are distinguishable, but still remarkably mixed.



(a) Reuters dataset embedded in 2D space by ISOMAP (cosine distance)

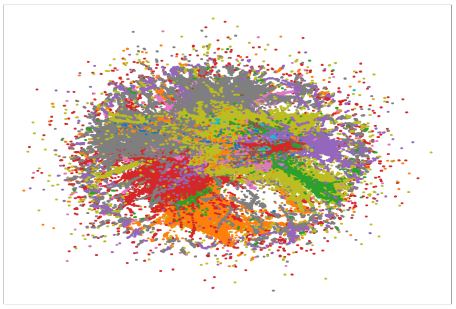


(b) Reuters dataset embedded in 2D space by ISOMAP (euclidean distance)



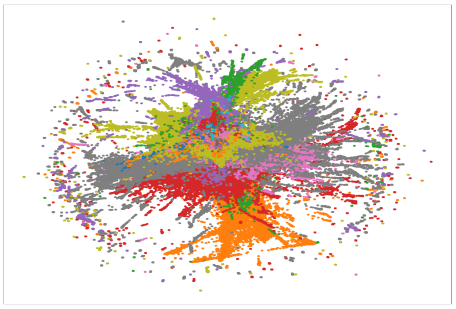
(c) Reuters dataset embedded in 2D space by ISOMAP (Minkowsky distance)

- IVHD for 2 and 3 neighbors - we use full dataset. The best embedding of original dataset was made by IVHD launched for 3 neighbors and euclidean distance in the precomputed KNN graph. This embedding makes disjoint clusters of data points and we can observe that there are not as much non-clustered data points as in the other clusters. Additionally, IVHD has well handled the Reuters dataset in a comparison to the other methods.

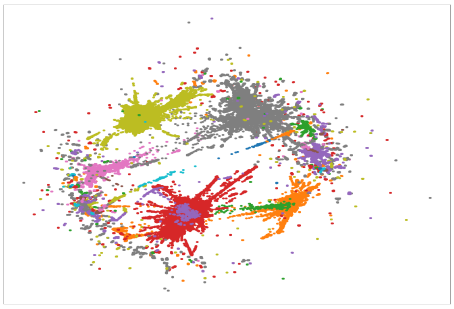


(a) Reuters dataset embedded in 2D space by IVHD (cosine distance, 2 neighbors, 3500 iterations)

(b) Reuters dataset embedded in 2D space by IVHD (euclidean distance, 2 neighbors, 3500 iterations)



(c) Reuters dataset embedded in 2D space by IVHD (cosine distance, 3 neighbors, 3500 iterations)



(d) Reuters dataset embedded in 2D space by IVHD (euclidean distance, 3 neighbors, 3500 iterations)

We used the same parameters as for the FMNIST and Smallnorb dataset including number of iterations.

3.3.2 Metrics

Now we will present the results of the metrics:

- DR quality - we computed DR quality for UMAP, PaCMAP, TriMAP (manhattan distance) and IVHD (euclidean distance used in precomputed KNN graph, 3 neighbors, 3500 iterations) for truncated dataset (70000 observations). In the case of HUMAP we encountered problems resulting from the operation of this method. From the figure 30 we see that the data reduction quality is improving until the specified value of neighbors

for each method.

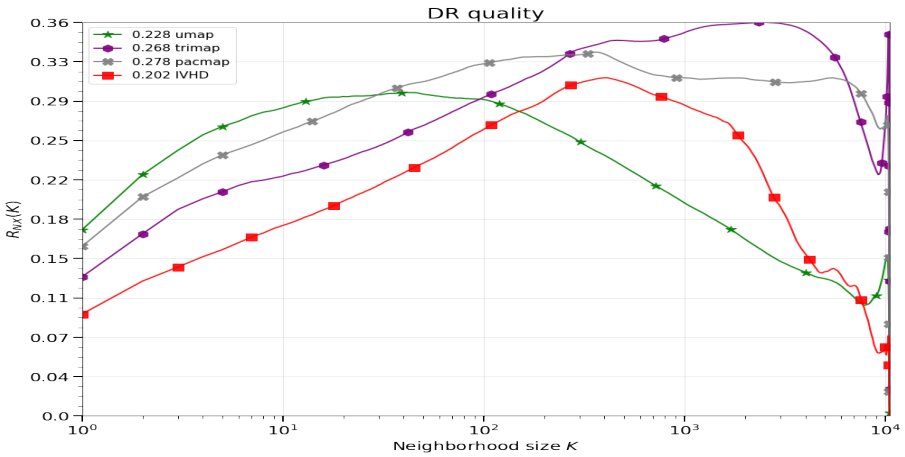


Figure 30: DR quality for specified methods

- KNN gain - we computed KNN gain for UMAP, PaCMAP, TriMAP (manhattan distance) and IVHD (euclidean distance used in precomputed KNN graph, 3 neighbors, $3,500$ iterations) for truncated dataset ($70,000$ observations). In the case of HUMAP and ISOMAP we encountered problems mentioned above. The greatest benefit of the KNN classifier is achieved for the IVHD method (for the embedding which was chosen by us as the best).

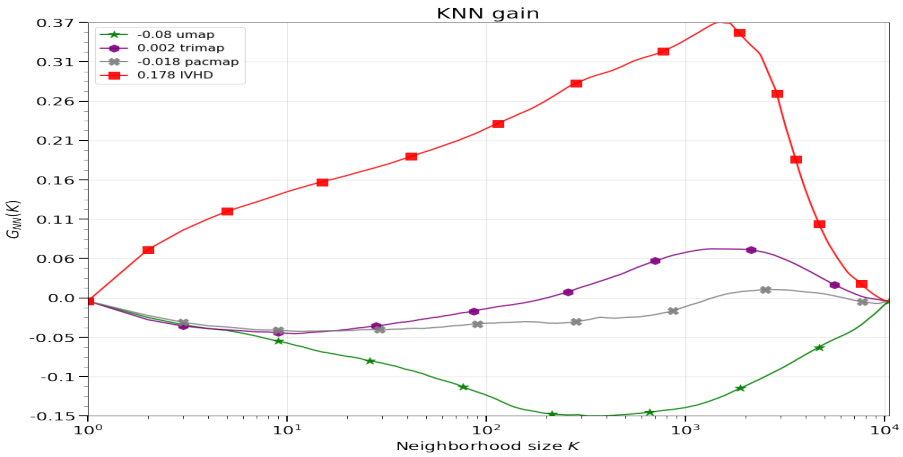
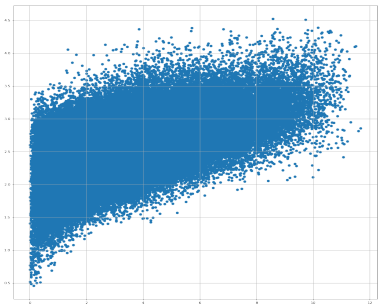
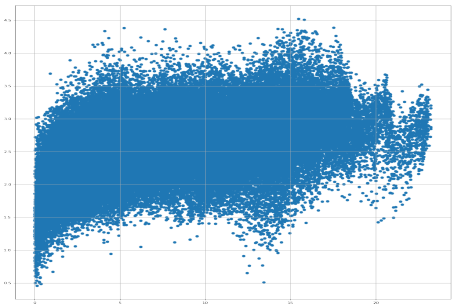


Figure 31: KNN gain for specified methods

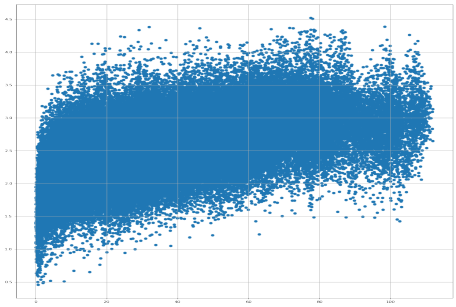
- Shepard's diagrams - in all the methods examined below, we see that the point clouds are lying on the line $y = x$ in case of ISOMAP, TriMAP and UMAP.



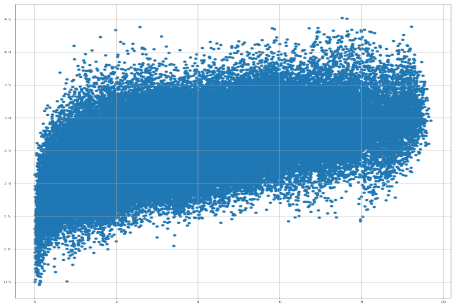
(a) Reuters dataset embedded in 2D space by ISOMAP (euclidean distance)



(b) Reuters dataset embedded in 2D space by PacMAP



(c) Reuters dataset embedded in 2D space by TriMAP (euclidean distance)



(d) Reuters dataset embedded in 2D space by UMAP

- Trustworthiness - generally, every method has a good trustworthiness measure.

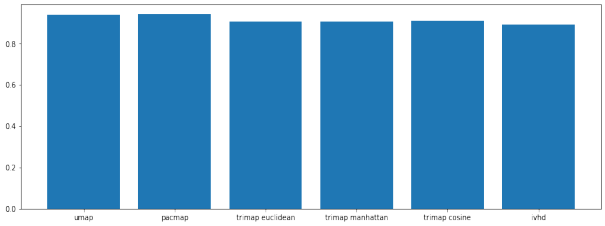


Figure 33: Trustworthiness calculated for Reuters dataset for following methods: UMAP, IsoMAP euclidean, PacMAP, TriMAP euclidean, Trimap euclidean and IVHD (euclidean distance, 3 neighbors, $3,500$ iterations)

3.3.3 Times of embeddings execution

Now we will show the times of operation of individual methods.

- UMAP, full dataset: 27 min 43s,
- TriMap euclidean distance, full dataset: 33 min 05s
- TriMap cosine distance, full dataset: 33 min 55s,
- TriMap manhattan distance, full dataset: 34 min 20s,
- PacMap, full dataset: 21 min 51s,
- IVHD euclidean distance, full dataset, 3 neighbors, 3500 iterations: 2 hours 3 min 16s.

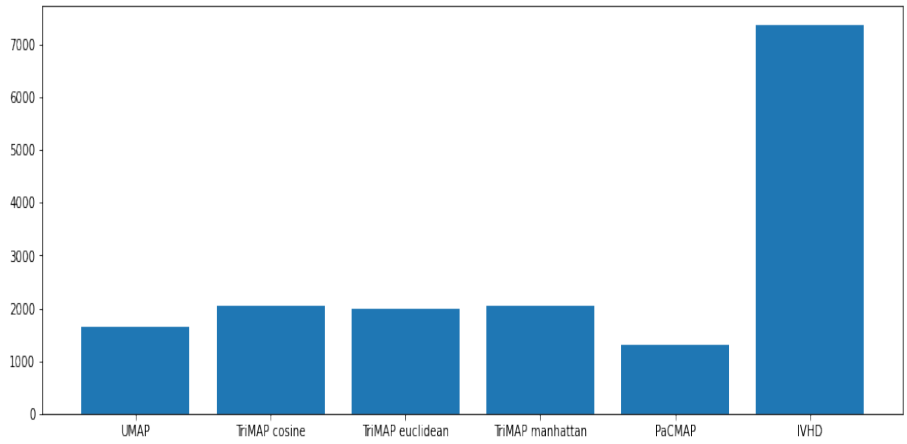


Figure 34: Times for specified methods

4 Bibliography

[1] HUMAP: Hierarchical Uniform Manifold Approximation and Projection by Wilson E. Marc ılio-Jr, Danilo M. Eler, Fernando V. Paulovich, Rafael M. Martins (2021)

[2] Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization by Yingfan Wang, Haiyang Huang, Cynthia Rudin, Yaron Shaposhnik (2021)

[3] 2-D Embedding of Large and High-dimensional Data with Minimal Memory and Computational Time Requirements by Witold Dzwinel, Stan Matwin, Rafał Wcisło