Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies

FIIT-182905-73940

Bc. Jakub Janeček

Interpretability of machine learning models created by clustering algorithms

Master thesis

Study program: Inteligent software systems
Field of Study: 9.2.5 Software engineering, 9.2.8 Artificial Intelligence
Place: Institute of Informatics, Information Systems and Software Engineering
Supervisor: Ing. Jakub Ševcech, PhD.

April, 2019

STU
FIIT

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Zadanie diplomovej práce

Meno študenta: **Bc. Jakub Janeček**

Študijný program: Inteligentné softvérové systémy

Študijný odbor: Softvérové inžinierstvo – hlavný študijný odbor
Umelá inteligencia – vedľajší študijný odbor

Názov práce: **Interpretovateľnosť modelov strojového učenia vytvorených zhlukovacími algoritmami**

Samostatnou výskumnou a vývojovou činnosťou v rámci predmetov Diplomový projekt I, II, III vypracujte diplomovú prácu na tému, vyjadrenú vyššie uvedeným názvom tak, aby ste dosiahli tieto ciele:

Všeobecný cieľ:

Vypracovaním diplomovej práce preukážte, ako ste si osvojili metódy a postupy riešenia relatívne rozsiahlych projektov, schopnosť samostatne a tvorivo riešiť zložité úlohy aj výskumného charakteru v súlade so súčasnými metódami a postupmi študovaného odboru využívanými v príslušnej oblasti a schopnosť samostatne, tvorivo a kriticky pristupovať k analýze možných riešení a k tvorbe modelov.

Špecifický cieľ:

Vytvorte riešenie zodpovedajúce návrhu textu zadania, ktorý je prílohou tohto zadania. Návrh bližšie opisuje tému vyjadrenú názvom. Tento opis je záväzný, má však rámcový charakter, aby vznikol dostatočný priestor pre Vašu tvorivosť.

Riaďte sa pokynmi Vášho vedúceho.

Pokiaľ v priebehu riešenia, opierajúc sa o hlbšie poznanie súčasného stavu v príslušnej oblasti, alebo o priebežné výsledky Vášho riešenia, alebo o iné závažné skutočnosti, dospejete spoločne s Vaším vedúcim k presvedčeniu, že niečo v texte zadania a/alebo v názve by sa malo zmeniť, navrhnite zmenu. Zmena je spravidla možná len pri dosiahnutí kontrolného bodu.

Miesto vypracovania: Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU v Bratislave

Vedúci práce: **Ing. Jakub Ševcech, PhD.**

Termíny odovzdania:

Podľa harmonogramu štúdia platného pre semester, v ktorom máte príslušný predmet (Diplomový projekt I, II, III) absolvovať podľa Vášho študijného plánu

Predmety odovzdania:

V každom predmete dokument podľa pokynov na www.fiit.stuba.sk v časti:
home > Informácie o > štúdiu > harmonogram štúdia > diplomový projekt.

V Bratislave dňa 12. 2. 2018

SLOVENSKÁ TECHNICKÁ UNIVERZITA
V BRATISLAVE
Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4
1

prof. Ing. Pavol Návrat, PhD.
riaditeľ Ústavu informatiky, informačných systémov
a softvérového inžinierstva

STU
FIIT

SLOVENSKÁ TECHNICKÁ
UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY
A INFORMAČNÝCH TECHNOLÓGIÍ

Návrh zadania diplomovej práce

Finálna verzia do diplomovej práce [1]

Študent:

Meno, priezvisko, tituly:	Jakub Janeček, Bc.
Študijný program:	Inteligentné softvérové systémy
Kontakt:	janecekk@gmail.com

Výskumník:

Meno, priezvisko, tituly:	Jakub Ševcech, Ing. PhD.

Projekt:

Názov:	Interpretovateľnosť modelov strojového učenia vytvorených zhlukovacími algoritmami
Názov v angličtine:	Interpretability of machine learning models created by clustering algorithms
Miesto vypracovania:	Ústav informatiky, informačných systémov a softvérového inžinierstva, FIIT STU, Bratislava
Oblasť problematiky:	analýza údajov, strojové učenie

Text návrhu zadania[2]

Interpretovateľnosť je kľúčovou vlastnosťou modelov strojového učenia, ak je našim cieľom presvedčiť expertov z domény, pre ktorú je model navrhnutý, aby ho prijali a používali. Čím lepšie dokážeme vysvetliť správanie nášho modelu, tým väčšia je šanca na jeho prijatie. Preto je potrebné usilovať sa nielen o čo najlepší výsledok modelu, ale aj jeho interpretovateľnosť. Jedno bez druhého má malý význam. V praxi je viac menej isté, že z dvoch modelov s rovnakými výsledkami bude vybratý ten, ktorý vieme lepšie vysvetliť. Dokonca sa môže stať, že model, ktorého výsledky sú do istej miery horšie ako iného, bude vybratý, ak ho vieme lepšie vysvetliť. Tento prístup platí pre učenie s učiteľom aj bez neho. Konkrétne zhlukovanie, ako jedna z aplikácií učenia bez učiteľa je oblasť, ktorá nie je v tomto ohľade dostatočne preskúmaná. To ponúka možnosť priniesť nové zaujímavé riešenia. Príkladom je určovanie dôležitosti atribútov a ich výber pri zhlukovaní, ktorému sa doteraz nevenovala veľká pozornosť.

Analyzujte jednotlivé existujúce spôsoby určovania dôležitosti atribútov pre správanie zhlukovacích modelov. Porovnajte výsledky týchto spôsobov. Navrhnite metódu na určovanie dôležitosti atribútov pre modely zhlukovania. Analyzujte existujúce spôsoby využitia dôležitosti atribútov pre zvýšenie interpretovateľnosti modelov zhlukovania. Navrhnite a implementujte spôsob využitia dôležitosti atribútov pre zvýšenie interpretovateľnosti modelov zhlukovania. Experimentujte s rôznymi datasetmi a algoritmami pre zhlukovanie. Navrhnuté riešenie overte na rôznych datasetoch netriválnej veľkosti a zložitosti.

[1] Vytlačiť obojstranne na jeden list papiera

[2] 150-200 slov (1200-1700 znakov), ktoré opisujú výskumný problém v kontexte súčasného stavu vrátane motivácie a smerov riešenia

Literatúra[3]

- PLANT, Claudia; BÖHM, Christian. Inconco: Interpretable Clustering of Numerical and Categorical Objects. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2011. p. 1127-1135.
- PARISOT, Olivier; GHONIEM, Mohammad; OTJACQUES, Benoît. Decision Trees and Data Preprocessing to Help Clustering Interpretation. In: Proceedings of 3rd International Conference on Data Management Technologies and Applications. SCITEPRESS-Science and Technology Publications, Lda, 2014. p. 48-55.

Vyššie je uvedený návrh diplomového projektu, ktorý vypracoval(a) Bc. Jakub Janeček, konzultoval(a) a osvojil(a) si ho Ing. Jakub Ševcech, PhD. a súhlasí, že bude takýto projekt viesť v prípade, že bude pridelený tomuto študentovi.

V Bratislave dňa 8.1.2018

Podpis študenta

Podpis výskumníka

Vyjadrenie garanta predmetov Diplomový projekt I, II, III

Návrh zadania schválený: áno / ~~nie~~[4]

Dňa: 12. 2. 2018

Podpis garanta predmetov

[3] 2 vedecké zdroje, každý v samostatnej rubrike a s údajmi zodpovedajúcimi bibliografickým odkazom podľa normy STN ISO 690, ktoré sa viažu k téme zadania a preukazujú výskumnú povahu problému a jeho aktuálnosť (uveďte všetky potrebné údaje na identifikáciu zdroja, pričom uprednostnite vedecké príspevky v časopisoch a medzinárodných konferenciách)

[4] Nehodiace sa prečiarknite

ACKNOWLEDGMENTS

I would like to thank my supervisor Ing. Jakub Ševcech, PhD. who helped me alot in course of my thesis, always willing to answer my questions and provided me with guidance. I would also like to thank my family and friends who helped me with support and advices.

DECLARATION

I declare, that this thesis was composed by me and me only, and with help of literature and scientific articles listed in this work.

Bratislava, 30.04.2019

....

Bc. Jakub Janeček

Anotácia

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ
Študijný program: Inteligentné softvérové systémy
Autor: Bc. Jakub Janeček
Diplomová práca: Interpretovateľnosť modelov strojového učenia vytvorených zhlukovacími algoritmami
Vedúci práce: Ing. Jakub Ševcech, PhD.
December, 2019

Strojové učenie sa v našej dobe stalo už viac menej samozrejmosťou pri riešení mnohých výskumných problémov, ale aj reálnych problémov z praxe. Rozdiel v týchto dvoch oblastiach je, že v pri výskumných problémoch nie je absolútnou nutnosťou vedieť vysvetliť vytvorený model a jeho rozhodnutia. Ak však chceme, aby náš model používali v inej oblasti, a verili, že rozhodnutia ktoré robí sú správne, je potrebné aby sme vedeli vysvetliť aj komplexné modely. V našej práci sa zameriavame na zhlukovanie ako oblasť, ktorá bola z tohto hľadiska menej preskúmaná. Konkrétnou špecifikáciou, ktorej sa venujeme je vysvetľovanie rozdielov medzi dvoma segmentami údajov na základe dôležitosti atribútov. Ako nástroje pre túto úlohu sme sa rozhodli využiť topologickú analýzu údajov pre úlohu segmentácie údajov, a regularizáciu lineárnych modelov pre úlohu určenia dôležitosti atribútov. Konkrétne využívame logistickú regresiu s L1 normalizáciou ako zástupný model, ktorý poskytuje riedky vektor atribútov ako výstup, a ten následne využívame na interpretáciu výsledkov modelu zhlukovania (segmentácie).

Annotation

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES
Degree Course: Inteligent software systems
Author: Bc. Jakub Janeček
Master thesis: Interpretability of machine learning models created by clustering algorithms
Supervisor: Ing. Jakub Ševcech, PhD.
April, 2019

In our era, machine learning has become something of a routine for solving many research problems, but also problems from real life. The difference between these two areas is, that for research oriented problems, it is not absolute necessity to be able to explain created model and its decisions. However, if we want for our model to be used in other area and want those using it to trust its decisions, it is necessary to be able to explain even complex models. In our work, we focus on clustering as field, that has been less researched from this angle. The exact specification, that we are focusing on is finding differences between two segments of data based on feature importance. As tools for this task, we decided to use topological data analysis as a segmentation tool, and regularization of linear models as a tool for finding importance of features. More specifically, we use logistic regression with L1 normalization as a surrogate model, which provides a sparse vector of attributes as output, that we in turn use to interpret the clustering (segmentation) model results.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In these days machine learning is used in almost all industries. People from these industries, often do not have to have technical background for complicated algorithms, which are used for machine learning, and in our case specifically segmentation of data. Nevertheless, these people need to understand the models, that the complicated algorithms create as best as possible. Because, if they do not understand the models or the reasons behind the decisions they make, it stirs the disbelief towards the results of the models. That is why it is necessary to design such interpretations of these models, which if not completely describe the models and all their aspects, enable comprehension of them and their decisions. The comprehension of models decision will bring greater trust in these models, and allow their application for real world problems.

Imagine we created the best possible model for determination of inherited diseases for newborns, which complexity is enormous, and we have no means of explanation for our model, by which we could clarify its decision to doctors. If it was so, the doctors would not be able to use this model, because if we can not say why the results are what they are, they can not trust it with health or even life of other people.

That is why various approaches are used to help interpret the results of machine learning models. Be it visualization of processed data, projection of multidimensional observations into two dimensional space and then visualizing them, or revelation of attributes and their importance for decisions of model.

This problem is present in every machine learning technique. We decided to focus on data segmentation, especially on explaining differences between two data segments by attributes. In other words, finding a subset of attributes that can split the two segments.

In chapter two 2, we analyze the existing methods. In chapter three 3 we overview the most common clustering approaches.In chapter four 4, we introduce the topological data analysis and explain the differences between clustering and this approach. In chapter five 5, we analyze the methods of explaining the differences between data segments. In chapter six 6, we describe our goals, hypotheses, motivation and contribution to the problem. In chapter seven 7, we propose the design of our method. In chapter eight 8, we describe the experiments we conducted, present and visualize the results we achieved. In chapter eight 9, we present our conclusions of the work.

Our contributions to the problem area are:

- method based on regularization of linear models for creating of reductions (subsets of attributes from original attribute space),
- reduction evaluation metric (R-Metric) for selecting the best reduction.

Chapter 2

Analysis of existing methods for explaining clustering results and difference between data segments

In this particular field, there is no much previous work done, that is focusing on explaining the difference between data segments by feature importance. In this chapter, we look into those we were able to find and deemed them the most important.

If the algorithm used for segmentation is one of the hierarchical clustering algorithms, it is possible to use dendrogram (figure 2.1). Its structure is similar to the structure of decision tree, however, decisions are not visualized based on the features of dataset. What matters in this diagram, are the lines and their length, which defines, how distant are segments of data. [4]

For every result of clustering, it is possible to interpret it by scatter plot. However, if scatter plot is used, most often only two attributes are used for visualization. But if there are more attributes that contribute to the segmentation, they have to be reduced. This can be done by PCA or t-SNE algorithm (figure 2.2). [11][10]

These reduce dimensionality, but the cost is losing the connections of original features to decisions of segmentation. That is why this approach is not sufficient for our goal.

In [15], decision trees were used to visualize what separates the clusters. Although elegant and easy to understand, this method becomes less feasible with growing dimensionality of the datasets. In high dimensional data, the trees are too complicated for humans to easily understand what separates two data segments. But for data with low dimensionality,

[1] original: https://www.gigawiz.com/hcluster2.html , accessed: 19.10.2017

[2] original: https://www.mathworks.com/products/demos/machinelearning/cluster_genes/cluster_genes.html , accessed: 19.10.2017

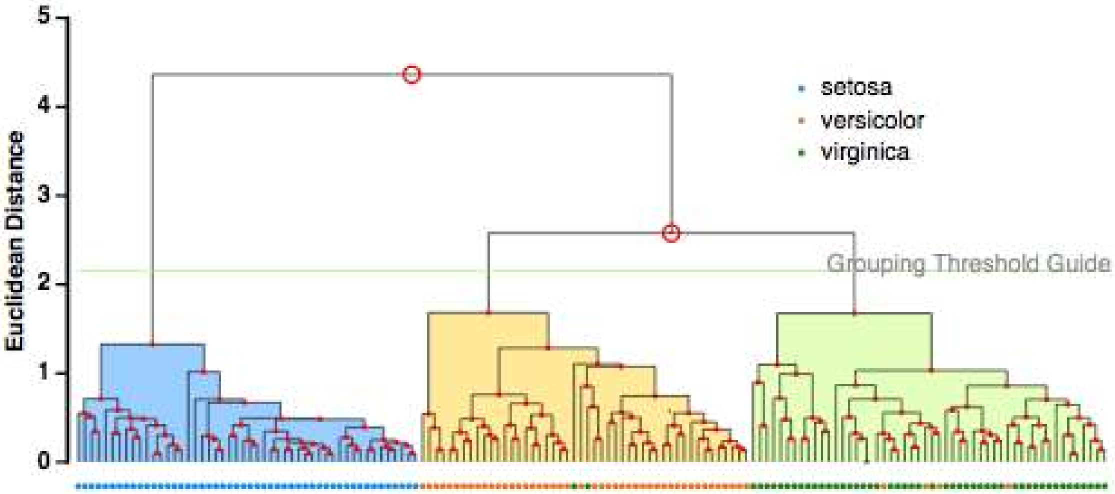


Figure 2.1: *Dendrogram for Iris dataset after hierarchical clustering*[1]

this technique is quite good and can be applied. Also it is important, that in some cases, when the single attribute in dataset can't alone divide the data apart, or its correlation with the class label is too strong, the decision tree will not work very well.

One work, that is particularly close to the problem, for which we are trying to come up with solution, is explanation technique LIME [16]. This technique is meant for explaining black box models for classification, by finding features that best identify the examined observation, with use of surrogate model. It scrambles the input features of observation and looks for the best match of these perturbed inputs' outputs and the output of original input. Figure 2.4 illustrates, how LIME looks for best matching subset of features (in this case parts of picture) to get those most important for output being "tree frog". It does an excellent job explaining why output for an observation is what it is. In their work, authors also proposed a technique for explaining the whole model, by drawing variable number of observations which best showcases and explains unique features, and afterwards finds the most important for the model as a whole. Still, this includes user as the one who needs to be present so that he can be explained the importance of features for selected observations. Interesting part is also using linear models as a tool to locally approximate models.

The solution based on decision trees can be utilized in the way, that only the most important decisions are visualized. For this to work, we would need to extract the importance of individual attribute from the data. There are tree main types of feature selection techniques. There are emebedded techniques, which provide the feature importance or selection of features as a side result of other machine learning task. An example of this technique

[3] original: https://www.oreilly.com/learning/introduction-to-local-interpretable-model-agnostic-explanations-lime, accessed: 20.04.2018

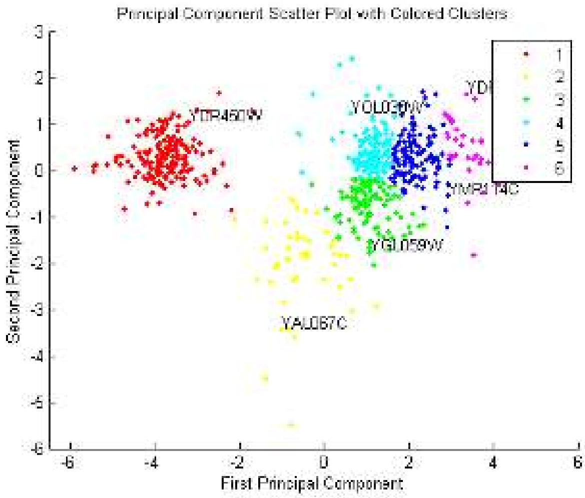


Figure 2.2: *Scatter plot after PCA* [2]

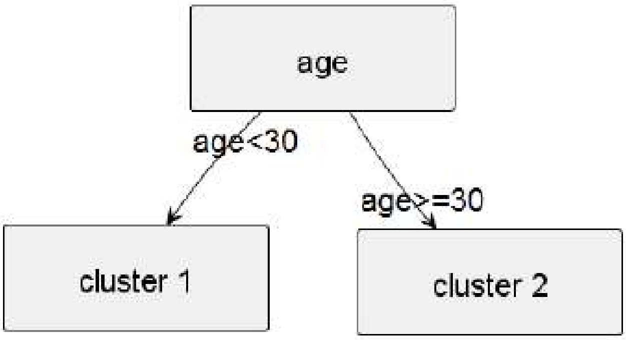


Figure 2.3: *Decision tree representing division of data to segments [15]*

is Decision trees, Random forest algorithm or SVM. The other groups is called filters. They utilize some statistical method to evaluate the importance of individual attributes. Example of this technique can be Kolmogorov-Smirnov score. The last groups consists of wrappers. A technique where a search of combinations of attributes is conducted with a way of evaluating these combinations and choosing the best, for example sequential feature selection.[22]

From the mentioned, the statistical method, Kolmogorov-Smirnov two sample test is non-parametric test, that provides the means to evaluating the difference between one-dimensional probability distributions of two randomly chosen samples of data. It quantifies the distance between empirical distribution functions of these two samples. We can use it to measure the distance between the distributions of each attribute for two chosen groups of data, and find the attributes that differ the most [17].

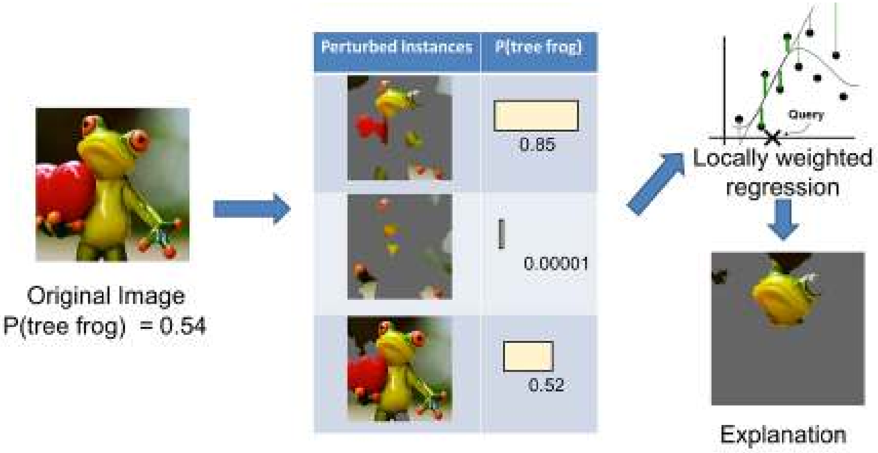


Figure 2.4: *Example of LIME explanation*[3]

Chapter 3

Clustering techniques

Clustering is a field of unsupervised machine learning. Unsupervised in the case of machine learning means, that the observation that serve as an input into clustering algorithms are not assigned into classes. In other words, we don't know the true labels for observations before the machine learning task in most cases, and if we do, they only serve the means of evaluation of the results. It is in fact, the goal of clustering, to provide these labels. To create groups from observations. These groups can vary in many aspects based on used clustering algorithm.

3.1 Centroid based clustering

In centroid based clustering, the segmentation problem is solved by the approach of selecting a vector in feature space of observations that represents a groups of points, their centroid. This technique works in iterations with following steps:

1. select random points (centroids) in feature spaces,
2. assign observations to their nearest centroid,
3. recalculate the position of centroid based on chosen approach,
4. if centroids moved (some observations changed cluster assignation), continue from step 2, otherwise end.

The mentioned approach can be for example mean, where the centroids are calculated as the means of points assigned to them. Or medoid approach can be used, when the centroids are actual observations from the dataset, and with iteration are selected as the points that have shortest distance to all other points in cluster. With the medoid approach, the outliers are better dealt with and do not affect the results as much as with mean approach. The resulting clusters are of a globular shape, and if we want to find clusters of other shape than that, we must use different clustering method.[14]

3.2 Hierarchical clustering

With hierarchical clustering, the observations are grouped in clusters based on the distance between them. Based on the distance criteria, this technique can create groups of different sizes. Moreover, it does not simply create groups from points. It constructs a whole hierarchy of groups of groups forming other groups. To represent this hierarchy, a dendrogram (figure 2.1) can be used. Also there are two main approaches. A divisive, starting with one group and splitting it into more, or agglomerative, starting with observations as groups and joining them into bigger groups. The distance between the clusters can be measured by different distance functions like single linkage, complete linkage or average linkage.[14]

3.3 Density based clustering

Density based clustering defines clusters as areas of attribute space which population is denser then elsewhere in the space. The observations that are located outside these areas are treated as noise, that separates the clusters. As for the clusters, they consist of core points and points that are in their range. Range is a value, that is user defined, and tells the algorithm the size of area where it needs to look for points in neighborhood of a point to determine if it is a core point. The number of points required for point to become a core point is also user defined. Density based clustering creates clusters which shape can be more complex than for example that of centroid based clustering creates.[14]

Chapter 4

Topological data analysis as a data segmentation tool

Topological data analysis (further only TDA) is an approach of applied mathematics, which focuses on analysis of datasets. It is based on techniques from topology, which is another field of mathematics. The area that topology is focused on, is properties of space and its deformations. More specifically, its main interest is in definition of objects in space and their definition according to some set of rules and characteristics. This is what TDA takes from topology, and applies it on analysis of datasets. There are actually two ways of analyzing datasets with TDA. Both have their pros and cons and provide different results. [24]

First is the statistical, or maybe we can say mathematical analysis with TDA. With it, we explore the dataset as we try to find shapes in data. But we are not visualizing any data. We simply want to be able to say that there are some shapes and what type of shapes they are. That sounds good. But there is a slight hiccup. The way we look for shapes, we need to specify various attributes that define the outcome of this search. Probably the most important attribute is the perimeter which defines if two edges (data points) should be connected, because this attribute affects the shapes that are the result. We can see the effect in figure 4.1. [5]

That is why the persistent homology was provided as a solution to the perturbations in shapes related to the changes of radius around points. It can be visualized in form of persistence diagrams or barcodes. In addition, we want to know, in which dimension of the data the shapes are. Both barcodes and persistence diagrams can visualize this information. In figure 4.2 we can see how some shapes are more persistent with the change of perimeter than others. The longer the line is, the more persistent is the shape it represents.

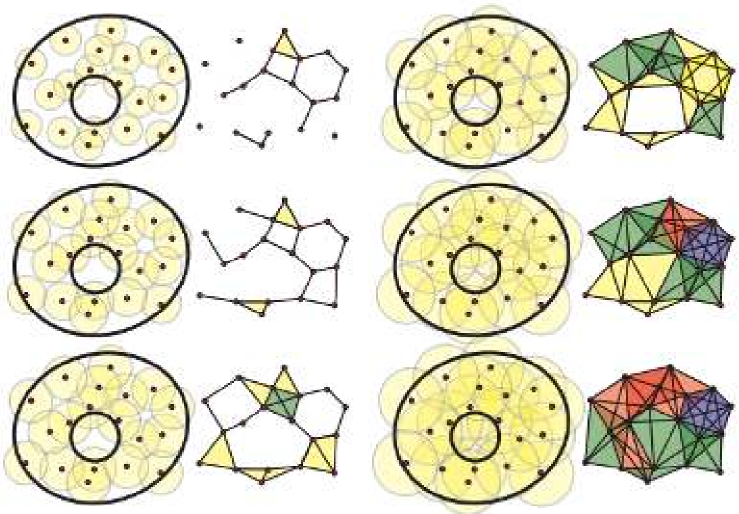


Figure 4.1: *Transformations of shape with changing perimeter [5]. Based on the size of perimeter around the point (yellow circles), the resulting shape can differ very much. The larger the perimeter, the greater the chance of connecting the point with other, and thus resulting in denser connected shape. We can see the results can be misleading, since the torus shape, which is the real shape in data from picture, was only achieved by one or maybe two setups of the visualized solutions.*

The information about dimension is provided by H regions on the left side. The x axis represents the perimeter around point to find overlapping points. In figure 4.3 we can see an example of persistence diagram. Both x and y axis represent the the perimeter around point. The points in the plot represent individual shapes, and their position represents their persistence. The more persistent shape, the further it is located from the diagonal line. Different shapes are used for different dimensions. The Betti number for each dimension holds the number equal to number of shapes that are deemed persistent enough to be a real shape. The cutoff value can variate based on our choice. [6]

But only knowing about shapes, their quantity and position doesn't help us in the tasks of machine learning. It is interesting in a way that we know, there are some persistent shapes in our dataset, and that's something. Because as once was said: "Data has Shape, Shape has Meaning, Meaning drives Value"[1]. But if we want to act based on these information, we need something else.

Second way of analysis of datasets using TDA is visualization of dataset. We humans are mostly able to perceive 3 dimensions, maybe 4 in some cases. But more than that is highly confusing for us. That's why there are many dimensionality reduction techniques, which try to narrow the dimensions using different methods. But most of them have one

[1] https://web.stanford.edu/class/archive/ee/ee392n/ee392n.1146/lecture/may13/EE392n_TDA_online.pdf, accessed: 4.12.2018

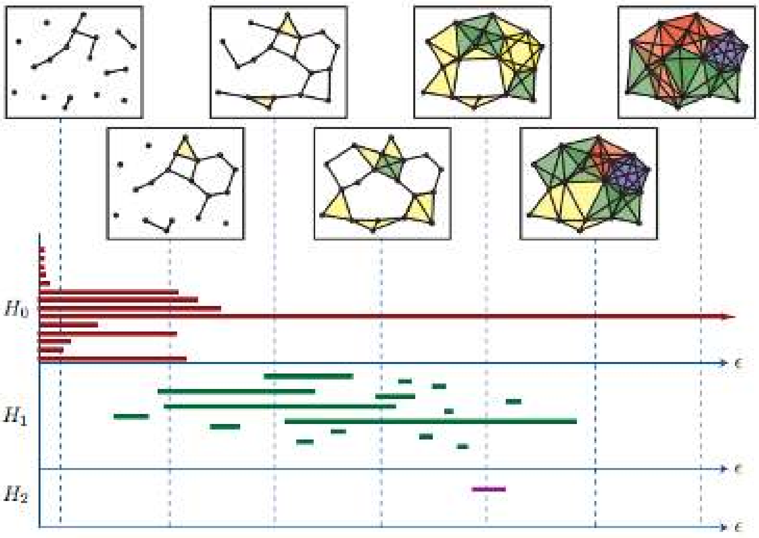


Figure 4.2: *Representation of persistent shapes with barcode. H_x are representing dimensions for Betti numbers. ∈ represents the growing perimeter. [5]. The lines in this diagram represent shapes found in data. We can see, that some lines, are long, and some very short. The length of the line represents the persistence of the shape, with the growing perimeter around points. The short lines can be treated as noise shapes, but if shape persist through big enough change of the perimeter around point, it can be probably treated as a real shape in data.*

thing in common. Their results are not without information loss. That's a benefit of TDA. It doesn't try to narrow the dimensions of dataset. It tries to find possible shapes in the data and visualize them. Example of TDA visualization can be seen in figure 4.4.

[2] original: https://www.ayasdi.com/blog/financial-services/5669-2/ , accessed: 21.04.2018

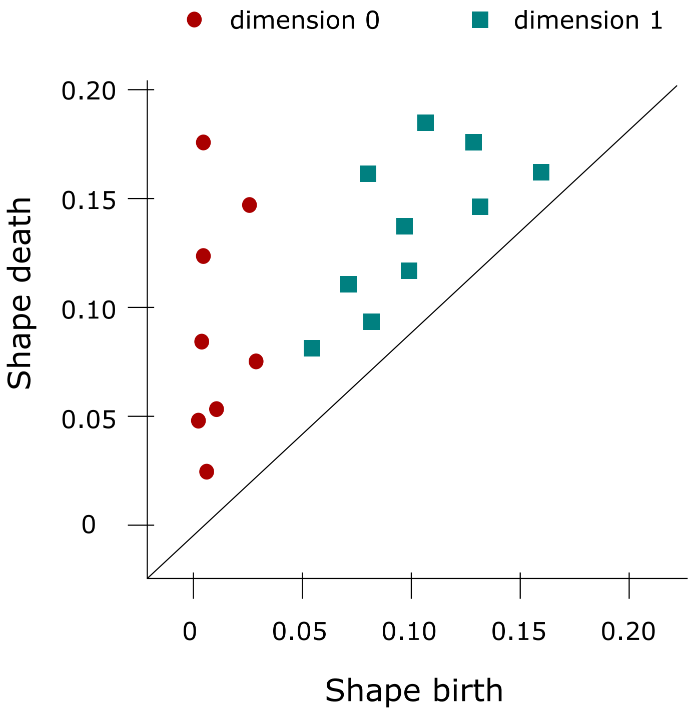


Figure 4.3: *Representation of persistent shapes with persistence diagram.*

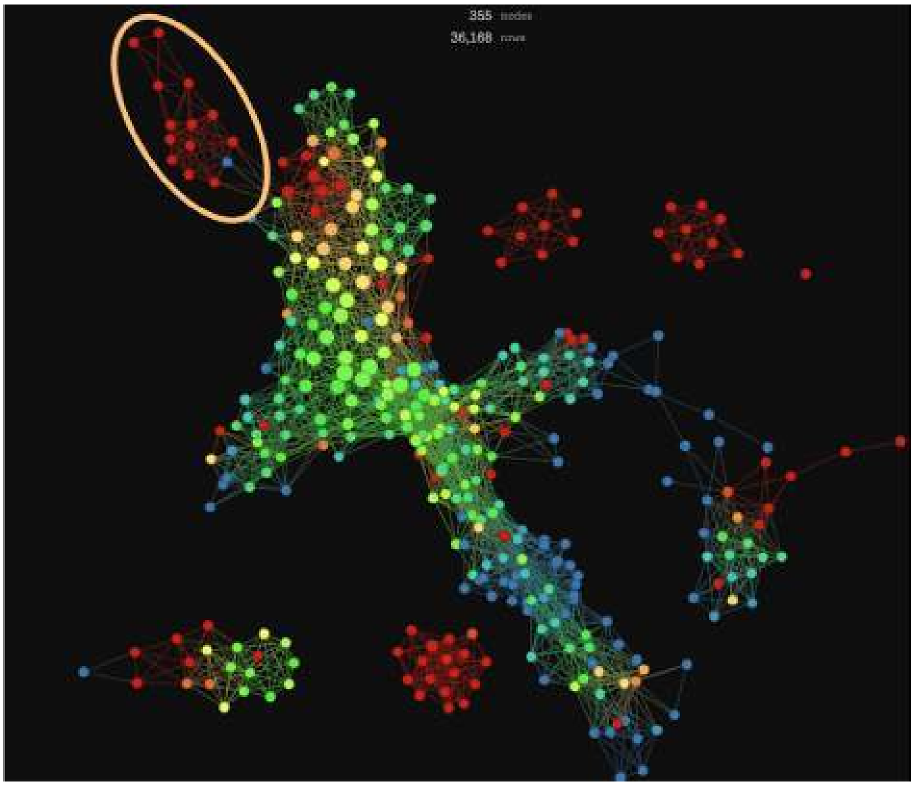


Figure 4.4: *Example of TDA visualization* [2]

Of course to visualize the shapes in multidimensional data, it is necessary that these shapes are represented in a way which can be visualized on the screens of computer. That's where the mapper algorithm comes in. Think of a mapper as a tool which defines a metric, which tells us how distant are the data points in our dataset from each other. If we can describe the relation of two points with a number, we can then apply a force model which spaces out all the data points in a 2D space. Then we can visually analyze the dataset and find some information we are looking for. How exactly mapper manages to transform the dataset into relations of individual data points can be found in 4.1.

In our work, we are focusing on the second approach, analysis through visualization. We are using TDA as a data segmentation tool, which can find shapes in our data. The difference between common clustering algorithms and TDA for high dimensional data is described in 4.2. In short, the approach in TDA and common clustering is different, which in turns produces different result. However, the goal remains the same. Find meaningful division of dataset to segments.

4.1 Role of mapper in TDA

Mapper is a key component of TDA, as it transforms the dataset we are analyzing into a form that can be visualized and shapes can be identified in its space. Let us consider a dataset with many observations and high dimensionality for attributes. Visualizing such a datset is a complicated task. Computer screens have a finite range of dimensions we can use to visualize. For really complex datasets, it becomes really hard to find a representation of the data, from which we can derive some meaningful information. Mapper as a tool, is designed to transform the representation of data to such, that can be visualized, analyzed and is simple enough for human perception. We could split the functionality of mapper into these steps:

- splitting of dataset into overlapping subsets
- clustering of subsets
- finding overlaps of clusters
- creating matrix of inter-point distances

To be able to simplify the representation of dataset, the first step of mapper is to split it into segments based on filter function. Filter function is defined for each point of dataset. To split the data according to the value of defined function, mapper needs to find the range of values for this function. Afterwards, it splits the data into subsets which are overlapping.

For example, if we would have function with values in range from 0 to 1, and we would be using 5 splits, we would create subsets like this:

- 0-0.225
- 0.185-0.425
- 0.385-0.625
- 0.585-0.825
- 0.785-1

The importance of overlapping of these subsets is great. Without it, we would not be able to produce connected vertexes representing the dataset. But more about this in following steps of mapper. For now we have subsets of dataset, which was split based on value of filter function.

Of course, one has to understand, even this approach has limitations. The filter function works only for small amount of metrics. Imagine cutting a thread into pieces of same length. It's simple. Now imagine cutting paper in pieces of same size. Now imagine splitting a cube into same parts. You see, with growing dimension of metrics it get harder to split the range of filter function. And it does not work for more than 3. So even thought this approach is good, its view of data is limited, and result very much depends on selecting the right filter function and its metrics.

The second step is to create clusters inside our subsets. Any clustering algorithm can be applied for this task, so that best possible outcome is achieved by selecting one most suitable for the types of data we have. The result of this step are clusters for whole dataset. Each cluster will be represented as a vertex and is representing a set of data points. This is the key to simplification of dataset representation while maintaining most of its informational value. If we would try to use TDA without this simplification, we would end up with a net of connected vertexes, each representing one data point, and would be overwhelmed with information. This way we have groups, represented with one point. For visualization, the color of these vertexes representing clusters is defined by the majority color of data points inside the cluster. This color for data points is derived from value of filer function.

The next step is finding connections between the created clusters. It is here, that the meaning of overlaps of the segments arises. We now iterate over the clusters, and if we manage to find two clusters, which share a common data point, we create a connection between these clusters. This way the shapes in the data start manifesting. In this step, some variations can be applied. For example, the width of the connection can represent how many data points clusters share.

The last step of mapper is calculating the inter-point distance matrix, which will define how the point will be spaced out in visualization. For this we can use a few metrics, such as density, eccentricity or any other metric, which geometrically describes the space of dataset and relation between data points. At the end we apply a force model which utilizes the values obtained by this metric. [18]

To give it a structure, mapper can be mapped to these steps:

- select filter metric,
- find metric range,
- split data into overlapping intervals based on metric,
- cluster data inside intervals,
- connect clusters with common observations.

We can see the application of these steps on a dataset representing a circle in figure 4.5.

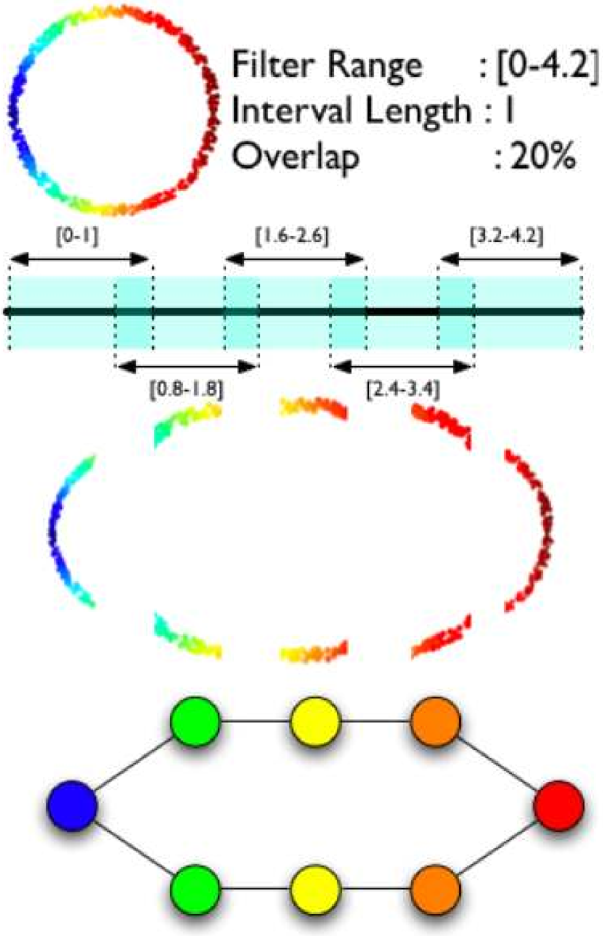


Figure 4.5: *Steps of mapper applied on dataset representing circle [18]*

The second step of mapper can be viewed as the one which affects the resulting shapes we find in data the most. As we mentioned, we use a filter function (view on data / lens), which represents only restricted view on data. It does not use all the attributes. Based on the filter function, we can then interpret the shapes we find. In filter function we not only look for similarity between observations, but we add a sort of interpretation. However not every interpretation has to be good, and if we want to construct a good one, we need to understand how this view transforms data.

4.2 Difference between TDA and common clustering algorithms

The first main difference between TDA and clustering algorithms is that clustering algorithms are segmentation tools, while TDA is a technique of explorative data analysis. This said, there are other differences and maybe some similarities on which we focus in this section.

With clustering algorithms which are used in problems of data segmentation, the goal is to divide dataset into groups. In this case a point either is or is not included in some cluster. These clusters are disconnected and they have no relation with each other. Depending on the algorithm which is used, there is a number of approaches to how dataset is divided. Choices are hierarchical clustering, density based clustering, centroid based clustering or distribution based clustering. [19] Each specifies a different technique for how to obtain segments from data. But result for each of them is the same. A group of disconnected not overlapping segments of data.

On other side, with TDA this is not the goal. With it, we are not trying to divide dataset into groups, merely to find shapes in data. If there are disjointed shapes in the data, that is fine, we take them. But we do not force this split. There is always the possibility, that our dataset will come out as one connected component from TDA. And it will be alright. Because even if a group is connected to the rest of the data, it can still be holding a value as different group then the rest, because of the shape. That is why TDA can work even there, where clustering algorithms can not. Another interesting fact is, that for example, centroid based clustering can only find spherical shaped clusters. On the other hand TDA is not limited to any particular shape, and in fact, it tells us what shape the cluster has. Which can be sometimes very important for us. In figure 4.6, we can see an example what clustering and TDA can achieve. Dataset A consists of three groups that can be clearly separated. Dataset B has three groups that can not be separated clearly. [1][2]

However, TDA is not a clustering algorithm. Yes it can be used for similar purpose - data segmentation, but its main aim is not to provide division of data, but to bring light to what is hidden in the data. What is its shape, what is hidden behind all those attributes. In short, how does our data look in its multidimensional space.

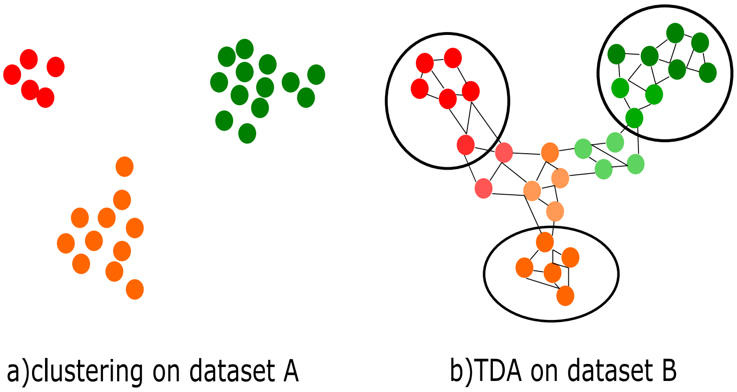


Figure 4.6: *Comparison of clustering and TDA results. On picture a), we see three separated clusters. On picture b), we see one connected component, but we can distinguish three segments of dataset which are in circles.*

Chapter 5

Methods of explaining differences between segments of data

With complex data segmentation models, it is quite hard to say what divides two chosen segments apart. Common techniques that are used to evaluate clustering results, include evaluating compactness and centrality of one cluster (5.1.1). They focus on interpreting a cluster as a sole unit, and don't take relationships between cluster into account. Our goal is to create a algorithm that will be able to tell which attributes from dataset best divide the two chosen cluster/segments. For this purpose numerous metrics could be use. For example in AYASDI (the biggest player on the market for TDA [1]), they used Kolmogorov-Smirnov score to see which attributes differ most for chosen segments. Or we can use some mean metric with normalized values for data. We can also use evaluation of classification models in our case, because even though problem of data segmentation falls into area of unsupervised learning, we are focusing on post processing of segmentation, and so we already know the labels for each data point. We take a better look at all the mentioned approaches in 5.1.

But still, the result would only be contributions of individual attributes to the splitting. But what if we want to know, how relations between more attributes affects the split? What if one individual attribute can not split two cluster good enough, but combination of attributes can? In this case, approaches we mentioned earlier wouldn't be to almost any use to us. We could maybe try the mean metric with normalized values for data, but instead of evaluating individual attributes, we could evaluate all possible combination of attributes. But this would prove computationally expensive, growing exponentially, and because of that not usable in high dimensional data which is our goal.

[1] https://www.ayasdi.com/ , accessed: 13.11.2018

Figure 5.1 shows us, that attribute x can not separate two classes. The same with attribute y. But combination of attribute x and y clearly separates these two classes, and by providing this duo of attributes, we are providing clear explanation what separates them.

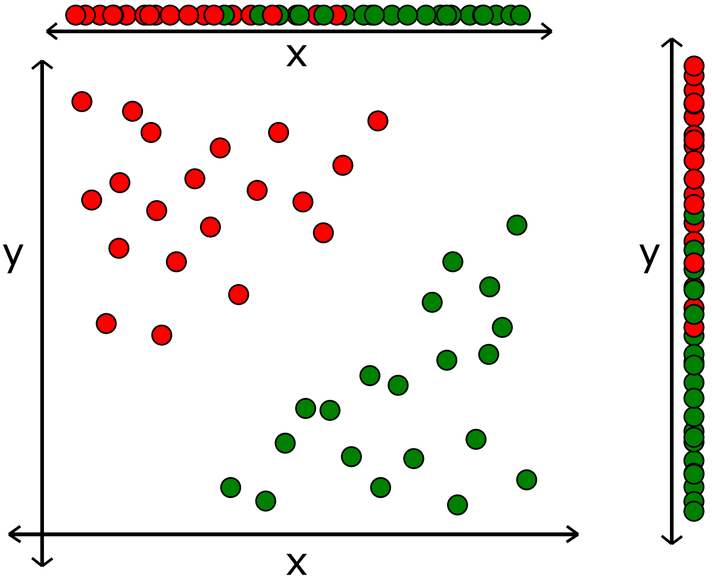


Figure 5.1: *Combination of attributes for explanation*

A solution, that seems plausible is regularization of linear models. Although main application of this method is in regression, it is also viable in classification with linear models. For example logistic regression, or support vector machines (further only SVM) are exactly those. We can use those to produce sparse vectors of attributes, that hold the best combination of features from dataset, that split target classes. On this we elaborate in 5.2.

5.1 Common techniques

5.1.1 Internal evaluation

With internal evaluation of clustering results, one does not use/have the ground truth labels of data, and simply tries to measure, how good the clusters are. We can measure the similarity between points in cluster (intra-cluster similarity) based for example on distance from centroid (in clustering models which work with centroids) and dissimilarity between points from different clusters (inter-cluster similarity). Methods for this evaluation are for example Davies–Bouldin index, or Dunn index. [8]

5.1.2 Interpretation of classifier

As said before, even though problem of data segmentation is a problem of unsupervised learning, we have the advantage, that the data segmentation has already been done, and we do not want to evaluate how good the results are. We simply want to be able to take two segments of data, and tell what attributes split them best. So we can work with the labels from segmentation as ground truth labels, and adjust the problem as binary classification. Each segment as a different class. That way we can use classification algorithms, that provide way of evaluating attributes based on contribution to splitting the classes. They can tell us which attribute provided the most information, on which splits between data were made. However this does not tell us how combinations of more attributes splits the data. For this purpose various classification algorithms can be used. Algorithms like Random Forest, Decision Tree.

5.1.3 Basic comparisons

We can also take a different approach and try to come up with own way of finding attributes that provide best split for chosen segments. As was already mentioned any metric that tells us how attribute for one segment differs from the same attribute in other segment, can provide a way of finding the attribute that differentiate these two groups best. We could visualize this difference for example with the use of box plots for each attribute for each group. Of course, we need to take into account the differences between value ranges of attributes, because difference of 100 in one can be less significant that difference of 0.1 in next. That said, we can use mean values of attributes for clusters, combination of these mean values with standard deviations to find if the attributes keep some stability or are just noise. These are all plausible applications. We could probably come with many more, and they would be to use to some extent. However they would not provide anything unique in contrast with evaluation of classification algorithms.

So to add some value to their results, we could try to measure combination of attributes not just single ones. But to do so, we would have to exhaustively search the whole space of combinations of attributes. And that could prove computationally unacceptable. In low dimensional data, it would maybe not take so long, but imagine what happens if we start adding attributes. The computation would grow exponentially. And since our main goal is to be able to provide a simple explanation of complex models from high dimensional data, this would not work for us.

5.2 Regularization of linear models

Regularization of linear models is a set of techniques, which main goal is to provide means of adjusting the learning process of algorithm to get better results, achieving them by improving generalization of model. The basic problem in machine learning can be described as finding function, that replicates (best approximate) the function which created data on which we are building our model. However our data can incorporate some noise. Does not matter how it got there, we simply must count with the fact that almost every time there is some. So the problem then is, how to tell our model that something is noise and something is real data. In terms of data science how to improve its generalization. That means that it will be able to predict correct result even on new data it was not trained on. And that is what regularization does. In linear model we are trying to find coefficients for each attribute so that the function composed from these attributes best approximates. By different means for each technique, it tries to regulate these coefficients, so that some regions of our function space are penalized and thus being able to generalize better. [12]

There are two basic forms of regularization. L1 regularization also called Lasso and L2 regularization also called Ridge. The main difference is that L1 uses just the sum of the absolute values of weights while L2 uses the sum of the square of the weights to regulate the coefficients. In regression, the goal is to minimize the cost function, which in linear regression is just the residual sum of squares.

$$Cost(W) = RSS(W) = \sum_{i=1}^{N}\{y_i - y_i'\}^2 = \sum_{i=1}^{N}\{y_i - \sum_{j=0}^{M} w_j * x_{ij}\}^2 \tag{5.1}$$

Equation 5.1 represents objective function of linear regression. Variables: y is real value; y' is predicted value; w is coefficient; x is attribute; N is number of observations; M is number of attributes.

With L1 regularization, the sum of absolute values of all coefficients is added. Equation 5.2 represents objective function of linear regression with L1 regularization. Variables: y is real value; y' is predicted value; w is coefficient; x is attribute; N is number of observations; M is number of attributes; **lambda** is regularization coefficient.

$$Cost(W) = RSS(W) = \sum_{i=1}^{N}\{y_i - y_i'\}^2 = \sum_{i=1}^{N}\{y_i - \sum_{j=0}^{M} w_j * x_{ij}\}^2 + \lambda * \sum_{j=0}^{M} |w_j| \tag{5.2}$$

As already said, in L2 the sum of squares of coefficients is added. Equation 5.3 represents objective function of linear regression with L2 regularization. Variables: y is real value; y' is predicted value; w is coefficient; x is attribute; N is number of observations; M is number of attributes; **lambda** is regularization coefficient.

$$Cost(W) = RSS(W) = \sum_{i=1}^{N}\{y_i - y_i'\}^2 = \sum_{i=1}^{N}\{y_i - \sum_{j=0}^{M} w_j * x_{ij}\}^2 + \lambda * \sum_{j=0}^{M} {w_j}^2 \quad (5.3)$$

This has various implications, as can be seen in table 5.1.

Table 5.1: *Differences between L1 and L2 regularization*

L1 and L2 differences	
L1	L2
- Computational efficient due to having analytical solutions - Non-sparse outputs - Feature selection	- Computational inefficient on non-sparse cases - Sparse outputs - Built in feature selection

Although interesting, regularization with L2 is not suitable for our goal. But with L1, we can achieve exactly what we want. Get a sparse vector of coefficient for attributes, which will tell us what attributes we need for learning a model able of distinguishing two segments. How this can be achieved is explained in 5.2.2.

5.2.1 Linear models in classification

In classification there are three main representatives of linear models.

- Logistic regression
- Perceptron
- Support vector machine classifier

Despite its name, logistic regression is a technique of classification. It models the probability of label being 0 or 1. In linear regression, algorithm chooses coefficients for attributes such, that they minimize the sum of squared errors. But in logistic regression, coefficient are chosen to maximize likelihood of observing sample values. The result of linear regression are continuous numbers with no specific borders. In logistic regression, the use of natural logarithm of the odds of the target variable guarantees results from 0 to 1. [21]

Perceptron is another tool for binary classification, which uses linear function at its core. It is the most basic artificial neural network there is. It consists of input layer (attributes), weights layer (one for each attribute + one for bias) and activation function layer. The attributes from input layer are multiplied with corresponding weights, and added together. Their sum is then passed to activation function, which decides the result. In perceptron, this function is called Heaviside Step function. [3] We can see this process visualized in figure 5.2

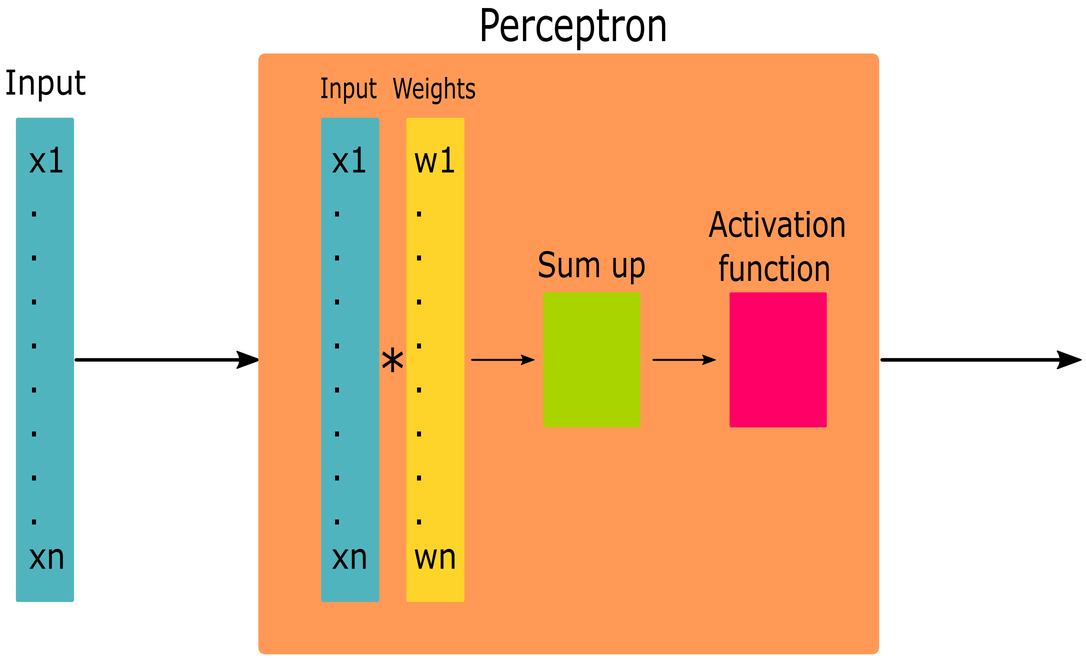


Figure 5.2: *Perceptron simulating one neuron*

Support vector machine (further only SVM) classifier is linear model, which tries to find hyperplane in space of function, that best separates the segments of data. It does so, by optimizing margin. Margin is the distance from hyperplane to the nearest data points on both sides. By doing so, it creates more robust models for data it has not seen yet. In figure 5.3 we can see how 3 different planes are fitted. The point that are closest to hyperplane, produce the support vectors, which are used in calculating the function of hyperplane. When using SVM classifier, one can also define if it should use the hard margin or soft one. If hard margin is used, we do not allow any data points from one class to be on the side of the other one. That in result means, that if the dataset is not linearly separable, we will not find a good hyperplane. If we allow soft margin, we specify, that these cases can occur but with some penalization. [7]

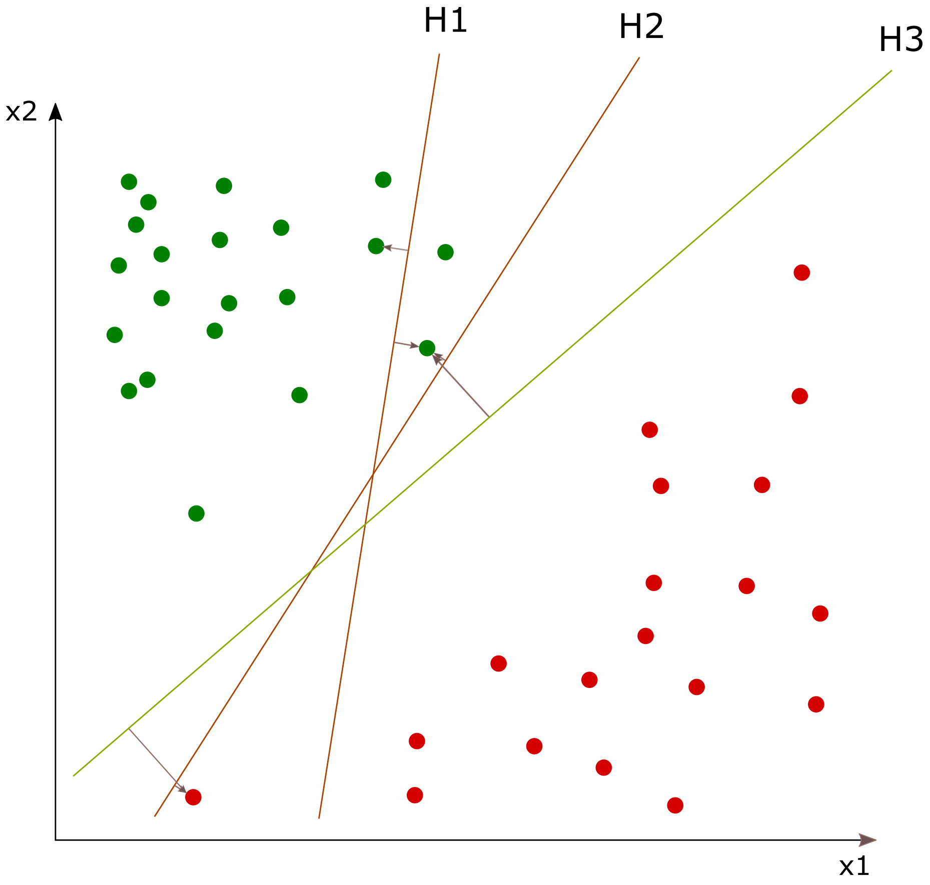


Figure 5.3: *Comparison of three different hyperplanes for SVM. Plane H1, does no split the classes. Plane H2, splits the classes, but does not maximize the margins. Plane H3 splits the classes and maximizes the margins.*

5.2.2 Feature selection using regularization of linear models

To achieve feature selection with regularization, L1 regularization has to be used. Lasso regression is a technique, that utilizes it, and from its first introduction took many forms because of improvements and changes. Honorable mentions include Elastic net, Lars and Fused lasso. The process by which Lasso manages to select features from dataset is based on regularization of function with which we try to reproduce the original one.

In linear models, we are trying to find best coefficients for all attributes, so that our produced function can replicate the labels for observations as best as possible. But if it fits perfectly on the points we have, it is very possible, it is modeling not just original function, but also noise from data. So the reason we regulate the coefficients for attributes is that they don't perfectly fit the training data (overfitting), but so that they generalize. And if we penalize some attribute more below a certain threshold, its significance in contribution to result of function is considered so low, it is set to 0. Result of this is, that it holds no value in prediction, so we can omit it, and work without it. That's what a sparse vector of coefficients means. For some attributes, the coefficients will be set to 0, and thus rendering them excludable from creation of model. In contrast, with L2 regularization, where the values of coefficients are also changing, they can never be 0. They can be very small, meaning some attribute is not really significant, but they are always present in model creation. [13]

5.3 Analysis overview

We analyzed some of existing solutions like dendrograms for hierarchical clustering, some techniques of dimensionality reduction for scatter plots and also usage of decision trees. All these solutions are achieving what they aim for, but are not exactly what we hope to achieve with our solution. The LIME technique is pretty close to our goal, but is more focused on single observations rather than groups of them (in our case clusters). In our method, we want to be able to use feature importance to distinguish two groups of observations apart and say, how these features differ in these groups (at least to be able to say one needs to be present and other not). However our approach is similar to the Lime technique in that sense, that we also use a surrogate model to interpret a black box model.

The TDA (topological data analysis), although being a technique of data exploration, plays the role of data segmentation tool in our work. In the analysis we asses the basics of this technique like Betti numbers, barcodes, persistence diagrams and also the core part of this technique which is mapper algorithm. Since we only use this technique and do not aim to bring any new methods or results in the field of TDA, we do not go deep into underlying mathematical definitions of homology, and how all the concepts of it works. As for the mapper algorithm, we try to break it into simple enough steps to understand, yet not lose any essential aspects of it. We also try to establish the understanding as to what differs TDA from common clustering techniques. To round it up, it is the fact that with TDA we can analyze even parts of data which can not be segmented into fully separated clusters, and also we can find cluster of any shape, or better said, we can find the shape of a cluster or segment of data. The shape can sometimes tell us a lot in addition to the observations it contains.

In the last part, we focus on methods of explaining differences between segments of data. We start with some theories about explaining the differences and adjust our focus on feature importance. We explore the potential of internal evaluation of clusters, like Davies-Bouldin index or Dunn index, some basic comparisons of differences between features in segments and also the possibility of classifier interpretations. The approach with classifier is close to what we strife for, but lacks few characteristics we would like to have in our solution.

As a chosen technique for finding the most important features, we present the regularization of linear models. Explaining the basic concept, what it is commonly used for (improving the generalization of linear models) and how we mean to use to achieve our goal. We explain the difference between the basic L1 and L2 regularization and why we will be using L1 form. In the end we present linear models used in classification.

Chapter 6

Motivation, goals and hypotheses for our work

The motivation for our work comes from the fact, that machine learning models tend to provide complex solutions. Sometimes maybe so complex, that it is difficult to understand them. That is why we decided to focus on post-processing of clustering (data segmentation), to shine some light on the decisions and results of this type of tasks. In figure 6.1 we can see our goal represented by the items of green color.

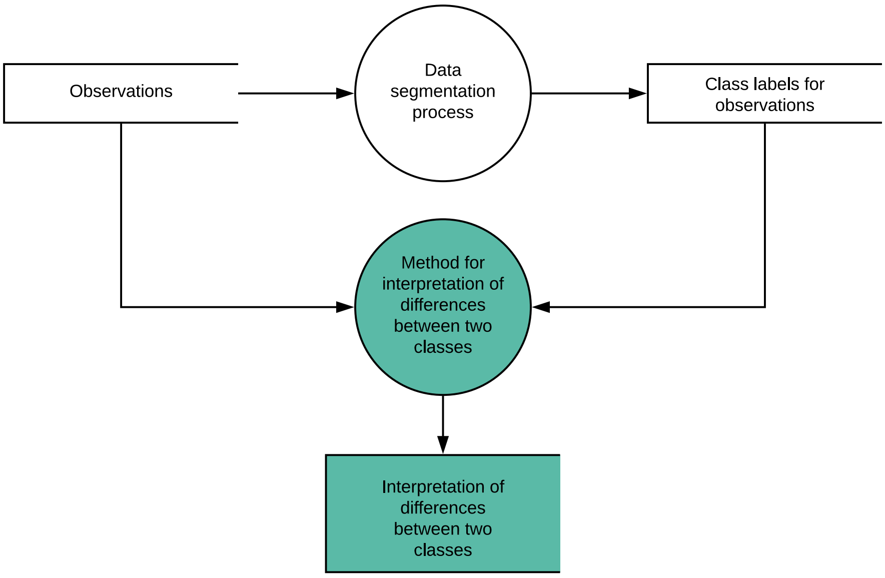

Observations
Data segmentation process
Class labels for observations
Method for interpretation of differences between two classes
Interpretation of differences between two classes

Figure 6.1: *Our addition to the problem area of interpretability of clustering models. Green items in diagram represent our addition to the problem area*

We want to provide a method, that will help with the interpretation of clustering models and their decisions. It is based on regularization of linear models and sparse vectors of attributes as their side result. The design of the structure of this proposed method can be seen in figure 6.2.

As the title of this thesis tells, we focus on interpretability of clustering models. To no surprise then, the goal of our work is: *"To provide a method, which will ease up the interpretation of clustering models and its decisions.*

To achieve this goal, we propose the already described method. To be able to say if our proposed method satisfies this goal, we formulated these hypotheses, with which we will evaluate our method:

1. A surrogate model can be used to find important features for whole selected classes in clustering model.

2. The results of linear classification model with L1 regularization provides enough information about attributes to ease up interpretation of clustering model decision.

In our experiments, both live (user study with people) and statistical, we will try to confirm these hypotheses.

6.1 Statistical evaluation

For the first hypothesis, we will perform a statistical comparison with other methods for feature extraction. We will use Random forest algorithm (RF), Kolmogorov-Smirnov score (KSS), SVM algorithm and sequential feature selection (SFS, the backward pruning option with the possibility to reintroduce already eliminated attribute) to asses the feature importance, and afterwards we will compare the results of these methods with the result of our method. We will evaluate these results in three ways. The simplest one will be to check the overlap of the attributes our method selected with the other algorithms results. As the second approach we will use the Normalized Discounted Cumulative Gain (NDCG), a metric used for recommendation evaluation [9]. It is based on the theory, that not all documents bear the same value to the result and also that the position of document in retrieved set matters. We adjust this metric for our problem this way. We are not retrieving documents, but attributes. The value of the attributes is in one case assigned from the results of the methods we compare ourselves with (RF, KSS, SVM). In second case, we use the results of the compare methods to order the attributes based on their importance and afterwards use the the position of the attribute in these ordered sets as its value. With the NDCG approach we are unable to evaluate the SFS, because the implementation of this algorithm does not provide weight with the selection of attributes. The third approach is Average precision (AP), also a recommendation evaluation technique [23]. With this metric, only the order of the attributes plays the role in the goodness of the retrieved set of attributes.

6.2 User study

For the second hypothesis, we will perform an user study, in which we will try to identify the attributes that are key for human perception of difference between two groups of observations. Afterwards, we will use this information to evaluate, how similar they are with the results our method provides.

We are using MNIST dataset for this user study. The groups used selected for this study were extracted from results of TDA. We selected 3 pairs of groups.

In the user study, the participant is presented with these pairs in three steps. In each one, he is presented with the images of mean values for these groups of images and also the difference between these two mean images. The participant overviews these images to gain information. Afterwards he proceeds to the interactive images. In them he is presented with the already mentioned difference of mean images, but this time overlaid with an explanation of feature importance for each group provided by our method. His task in one of them is to select those point, which in his opinion provide a good explanation of difference of attributes between these two groups of observations. In the other, he selects the point that badly explain the difference of attributes between the mentioned groups. He completes this task for all 3 pairs of the groups we extracted from TDA results.

The evaluation of this user study is based on processing the data to a structure, which tells us which attributes are overall evaluated by most of the participants as good, which as bad, and also identify those attributes, that are problematic for participants, and the decision for those attributes is unclear from the data we collect. Afterwards, we evaluate the overlay of these groups, with subset, that our proposed method identified as key for distinguishing the two groups apart based on attributes.

The complete user study can be seen in appendix B.1.2.

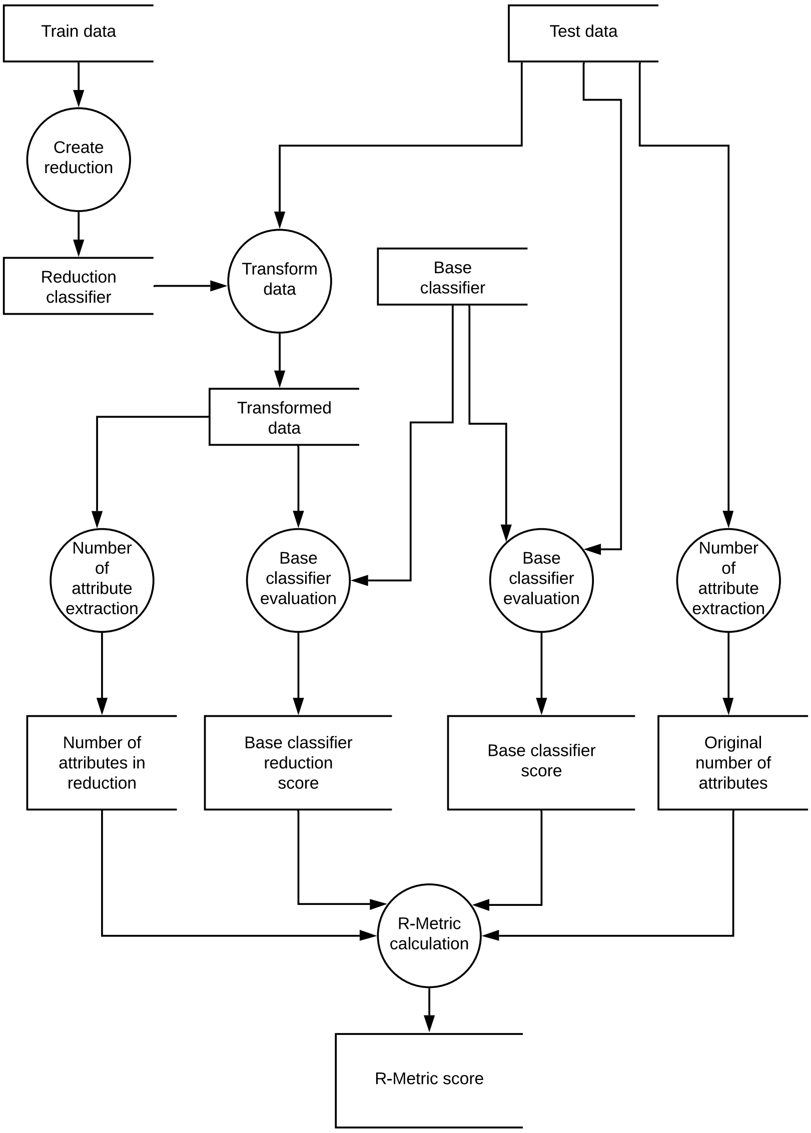

Train data
Test data
Create reduction
Reduction classifier
Transform data
Base classifier
Transformed data
Number of attribute extraction
Base classifier evaluation
Base classifier evaluation
Number of attribute extraction
Number of attributes in reduction
Base classifier reduction score
Base classifier score
Original number of attributes
R-Metric calculation
R-Metric score

Figure 6.2: *Data flow diagram representation of method for interpretation of differences between two classes. The* ***train and test data*** *are combination of observations and clustering labels split to train and test group. Process* ***create reduction*** *utilizes logistic regression with L1 regularization to provide sparse vector of attributes to transform the data with.* ***Base classifier*** *is an instance of logistic regression.* ***Alpha metric*** *is metric, we defined to compare reductions and find the best one (more about this metric in 7.1.3)*

Chapter 7

Design of proposed method

We propose a method, which will be able to select the most important features that split two segments of data apart and in addition provide some means of explaining the importance and difference of these attributes in these two segments. Since we are using linear model as a surrogate model to find important features, the segments we are analyzing need to be separable by this type of model.

7.1 Selection of important attributes

We use regularization of linear models as a tool for selecting the subset of important attributes. As stated in section 5.2, we want to use L1 form of regularization as it provides a sparse vector of weights for attributes as a result. For our linear model, we had two competitors. SVM and logistic regression. Based on some test we ran, we decided to base our method on logistic regression, as it provides similar results as SVM in our case, but is much faster because of lower complexity of its underlying algorithm. So the two main tools for selecting the attribute subset in our method are logistic regression with L1 regularization.

7.1.1 Logistic regression model creation

In our method, the input is data, specifically data consisting of two data segments, each observation labeled by the segment it belongs to. Because we want to be able to validate our decisions, we split the input data into two groups. One will be used to choose subset off attributes, and the other to test the result to find best solution. We will call these groups train and test group.

We use the train group to train the model of logistic regression. The algorithm for it has few parameters, from which the one that interests us is coefficient, which affects how sparse the resulting vector of weights for attributes will be. The problem is we want to a solution as sparse as possible while maintaining enough attributes, that we are able to split the two data segments good enough. Because we don't know many attributes will be omitted with our solution we just test every possible predefined value for C and find the best possible outcome. However, the evaluation of the result consists of two factors. The sparseness of weight vector, and the result of test classifier for subset of selected attributes on test group of data. For this reason we felt the need to come up with a score, which would be able to take these two factors into account and provide us with single number representing the current solution. We could then use this score to find the best solution. But since the input to evaluation score is also a result of evaluation classifier, we need to first define how we intend to test our reduction.

7.1.2 Reduction evaluation

In the beginning we said, that the input of our method is data, and we split it into two groups. The reduction provided by regularization is tested on the second part of the data, the test group. We use 5-fold cross validation which is repeated number of times (this number is input to our method and can be changed, we used 2 repetitions). But before cross validation, the test group must be updated to reflect the reduction we found and are testing. Hence, the attributes which weights were zeroed out in regularization are dropped from data, and classifier is only trained on subset of attributes. Of course, we also need to evaluate the classifier with original, not changed, test group, so we have a baseline to compare the reductions result to. As for classifier which provides this baseline result, we are using the same one which provided the reduction, logistic regression but this time we do not utilize the L1 regularization, but L2, which does not zero out attribute weights. We also use such regularization coefficient, that should yield no regularization at all. We use F1 macro as a metric of goodness and average all the result of cross validation giving us one number as a result. This number is the input to our defined R-Metric.

7.1.3 Proposed reduction evaluation metric

To evaluate the reductions we create, we defined a metric (further only R-Metric). With selected subset of attributes and test score for reduction, we are able to rank the reductions with score. We wanted to define a way, that would allow user to weight all its factors based on his liking. We also wanted this score to fulfill some basics, that would help us interpret its output. These are:

- The output for original data and result would be 1.
- If reduction is worse than original, the output is less than 1.
- If reduction is better than original, the output is more than 1.

With these in mind, we formulated this equation for our R-Metric:

$$R - Metric = (1 + \frac{n_1 - n_2}{n_1} * w_1) * (1 - (r_1 - r_2) * w_2) \tag{7.1}$$

To break it down, the left bracket represent the factor for number of attributes and right bracket represents the factor of evaluation classifier result.

The attribute factor consist of 3 variables. Variable n_1 stands for original number of attributes in data, the n_2 for selected number of attributes in reduction. The fraction in this part gives numbers in range (0;1>. In addition it can be weighted by w_1. The result of this bracket is in range <1*w_1 ; 2*w_1).

In the evaluation result factor, we also have 3 variables. Variable r_1 stands for evaluation score with original data, r_2 stands for evaluation score of reduced data and w_2 is used as a weight. The characteristics of this factor are similar to the attribute factor, yet not the same. Its output is in range <0*w_2 ; 2*w_2>.

For both weights, there is a constraint, that they have to be numbers in range <0 ; 1>.

As to why the R-Metric has this form, and is formulated as it is, these are the drives:

- Both factor have extra 1 in the bracket, so that if the n_1 is equal to n_2 or r_1 is equal to r_2, the result of concerned factor will be one, and if both cases are true, the result of R-Metric is 1.
- The attribute factor can not be less than 1. It is because any reduction we make to the number of attributes we are using, it is always an improvement. So this factor can not affect our score in negative way. That is why there is addition and not subtraction inside this factor.
- The evaluation result factor can affect the result of R-Metric both in negative and also positive way. Because if the result on the reduced data is worse than original, it is negative. But if we get better result on reduced data than original (not likely but possible), it is a positive. That is why we need a subtraction inside this factor to project the negative into output. (results are from range <0 ; 1>)

Chapter 8

Proposed method evaluation

In this chapter we describe the implementation of proposed method, and also experiments we conducted, to test this method. We also illustrate the use of Kepler mapper[20] (library for TDA) and its results. All of the mentioned was achieved using Python 3 as programming language.

8.1 Method implementation

We decided to implement our method, so that it consist of more steps. The core being the creation of reduction. On input, it awaits split data for two groups and coefficient for L1 regularization. Its first step is creation of model for logistic regression and training it on training part of data. This model contains the information about reduction of attributes. The pseudo code for this part is in listing 8.1.

Example 8.1: *Attribute reduction method pseudo code*

```
def create_reduction(train_data, test_data,
                     L1_coef):
    clf = logistic_regression(train_data,L1_coef)

    results = test_reduction(test_data,clf)

    attr_count = clf.get_attr()

    attr_weights = clf.get_weights()

    random_results = test_random_reduction(test_data,
                                           attr_count)
    return attr_counts, attr_weights, results,
           random_results
```

The next step is a method to test the reduction. Its purpose is to test the reduction, so as input it awaits test part of data, attribute that says, if reduction of test data should happen, and model of logistic regression for reduction. It creates another logistic regression model on reduced test data multiple times, and uses cross validation to test the reduction. Since in library we are using (scikit-learn) there is no possibility to use logistic regression without some sort of regularization, we use L2 with coefficient 10000, to signal we do not want any regularization to happen. Alongside the test of reduction, we test the inverted reduction, because it can provide additional information on the reduction. It tells us, how the results of classifier changes, if we remove our selected attributes, we also have a function which measures random reduction of attributes. By comparing these two results, we can say if our attributes were important for the classifier. We also calculate 95% confidence intervals for the results. The pseudo code for this part can be seen here 8.2.

Example 8.2: *Test of reduction pseudo code*

```
def test_reduction(test_data, clf):

    new_data = transform(test_data, clf)

    basic_clf = logistic_regression()

    score = cross_validate(new_data, basic_clf)

    confidence_intervals = get_CI(score)

    return [score, confidence intervals]
```

After the reduction and its testing, we use our defined R-Metric to evaluate each reduction. In appendix B.1, additional information about the proposed method implementation can be found.

8.2 MNIST dataset experiment - statistical evaluation

We decided to use the MNIST dataset, because it was big enough (around 60000 observations), and also had enough attributes (784), that improvement to interpretability would be useful. MNIST dataset represents images of number ranging from 0 to 9. There are 10 classes in this dataset, each for one number. The examples in image representation can be seen in figure 8.1.

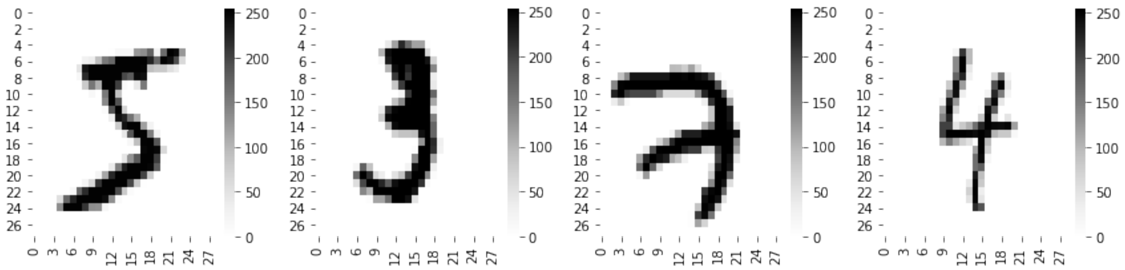


Figure 8.1: *Example of images from MNIST dataset*

8.2.1 TDA mapper application on MNIST dataset

We are using Kepler mapper to create visualization, which serves us as a tool for segmentation of data. Its implementation is pretty straight forward, but we must prepare the data to achieve good results. In section 4.1, we explained how mapper works. For the mapper to be able to split the data into overlapping segments of data, we must transform it into such space, which can be applied as a filter function. This experiment is a reproduced example from Kepler mapper documentation[1]. We applied t-SNE algorithm to reduce the feature space to two dimensions, so we were able to split this space. After this reduced space is split into overlapping segments, the corresponding original observations are assigned to groups based on this reduced space split. Mapper takes care of this split. Then we provide it with clustering algorithm. In our case, we used DBSCAN (form of density based clustering). The visual result can be seen in figure 8.2. We can see that some segments of data can be treated as groups.

The next step would be to allow user interaction to select two groups from visualization, with which we would work further. We used the interactive visualization of Plotly library to interact with the output of Kepler mapper and select groups we wanted to analyze.

8.2.2 Proposed method application on MNIST dataset groups

The mean values for groups we selected can be seen in figure 8.4 .As a first step, we split the data into two parts:

- train data - used to create reduction (40%)

[1] https://github.com/MLWave/kepler-mapper/blob/master/examples/digits/digits.py, accessed: 30.11.2018

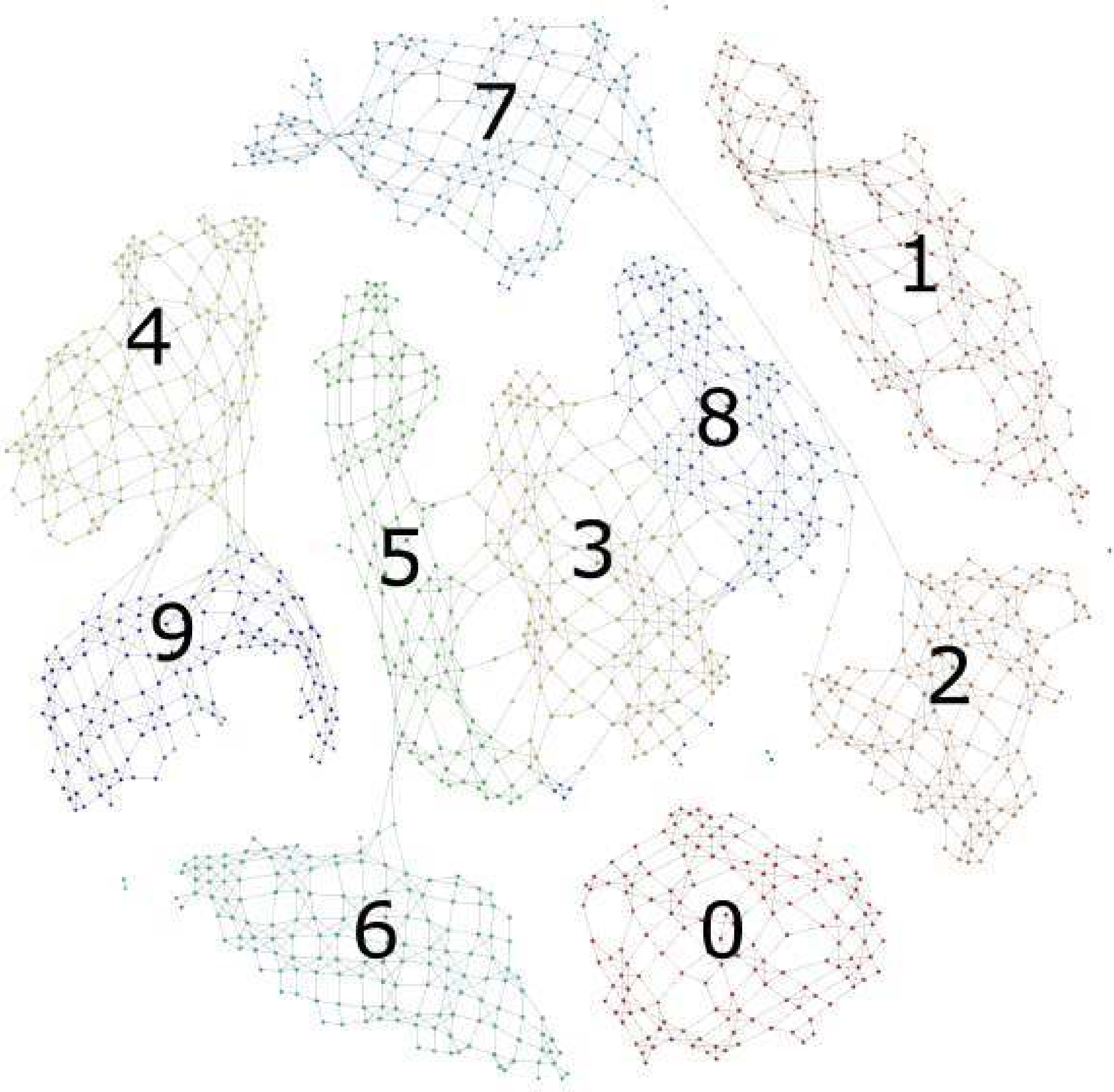


Figure 8.2: *Mapper result for MNIST dataset. Numbers represent the most common class in group*

- test data - used to evaluate the reduction (60%).

Afterwards, we created multiple reductions, which results can be seen in table 8.1. For the evaluation of reductions, we were using logistic regression with L2 regularization with coefficient 10000. The baseline result with f1 metric and no reduction of attributes (784) was 94.43%. The CI (confidence intervals) values do not spike into very large numbers, and that is good, meaning our results are not just a coincidence. For weights in our designed evaluation R-Metric, we assigned weight 0.25 to attribute reduction factor, and weight 1 to score difference factor.

Moreover, we carried out additional tests to gain more information about the reductions. These information are in table 8.2 and can be also seen in 8.3. The *r^{-1} score* stand for result

Table 8.1: *Reductions attributes and results*

alpha	n. attr.	R-Metric	score (f1)	95% CI (score)
5	425	1.1102	94.04%	+-0.003398
3	403	1.1151	93.85%	+-0.002954
1	364	1.1260	93.73%	+-0.004462
0.5	339	1.1303	93.41%	+-0.004446
0.1	330	1.1351	93.58%	+-0.004028
0.05	307	1.1366	93.08%	+-0.004417
0.01	222	1.1873	95.11%	+-0.004549
0.005	185	1.2033	95.46%	+-0.003080
0.001	108	1.2266	95.33%	+-0.003505
0.0007	97	1.2339	95.65%	+-0.002980
0.0003	72	1.2334	94.94%	+-0.002889
0.0001	46	1.2365	94.52%	+-0.002702

with data, from which we dropped the attribute we selected with reduction. This way we can compare these results with *rr score* which represent the result for data with as many attribute as chosen in reduction dropped by random choice. We can now compare the results of r^{-1} *score* with *rr score*, looking for a proof, that the attributes we have chosen are indeed important. The *rr score* should be higher than r^{-1} *score* to confirm this hypothesis.

Table 8.2: *Additional reductions attributes and results*

alpha	n. attr.	r^{-1} score (f1)	95% CI (r^{-1})	rr score (f1)	95% CI (rr)
5	425	65.27%	+-0.008301	94.7%	+-0.002168
3	403	80.50%	+-0.002501	94.59%	+-0.001912
1	364	89.30%	+-0.003801	94.41%	+-0.002102
0.5	339	90.52%	+-0.004901	94.09%	+-0.002372
0.1	330	89.58%	+-0.003461	94.14%	+-0.002527
0.05	307	91.61%	+-0.004819	93.74%	+-0.002426
0.01	222	93.56%	+-0.005566	93.08%	+-0.001980
0.005	185	93.41%	+-0.005278	93.49%	+-0.001788
0.001	108	92.03%	+-0.005916	93.81%	+-0.001955
0.0007	97	92.71%	+-0.003810	93.74%	+-0.001814
0.0003	72	92.86%	+-0.001625	93.95%	+-0.001458
0.0001	46	93.03%	+-0.003024	94.04%	+-0.001326

But logistic regression also provides additional information about the attributes it chooses for reduction. And those are weights for each of them. In case of the MNIST dataset (dataset of pictures) we decided the best way to provide this information would be in images. First, in figure 8.4, we can see mean values for both groups in data, and also

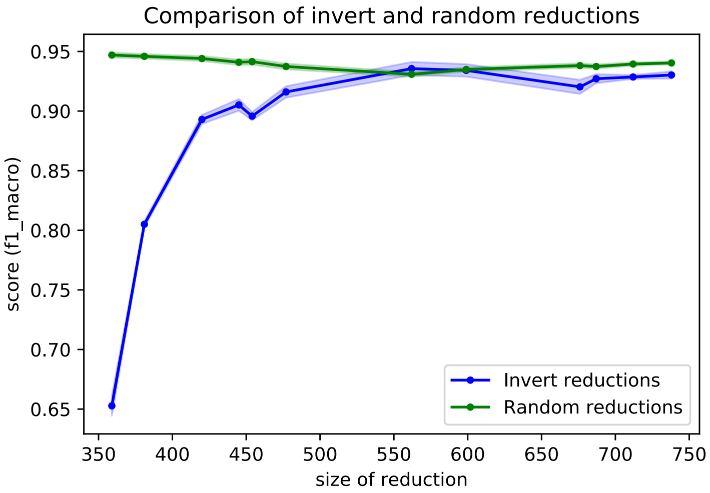


Figure 8.3: *Comparison of invert vs random reductions. In invert reductions, we removed our selected attributes from the former set. In random reductions, we removed the same amount of attributes as our method selected from the former set, but we have chosen them randomly. We can see that the random reductions manage (except for one case) to achieve better results. That provides a solid ground for our claim, that we can select important attributes*

their difference. The information about weights for attributes in reductions can be seen in figure 8.5. These weights have been standardized for better visualization, and values from one reduction are not meant to be compared against values from other reductions. In logistic regression, the algorithm splits data from one class against other. So the weights tell us, which attributes should be present for one class and which for other. With the help of figure 8.4, we can see the red pixels represent the attributes which need to be present for an observation to be taken as belonging to group 1, and blue ones for the group 2.

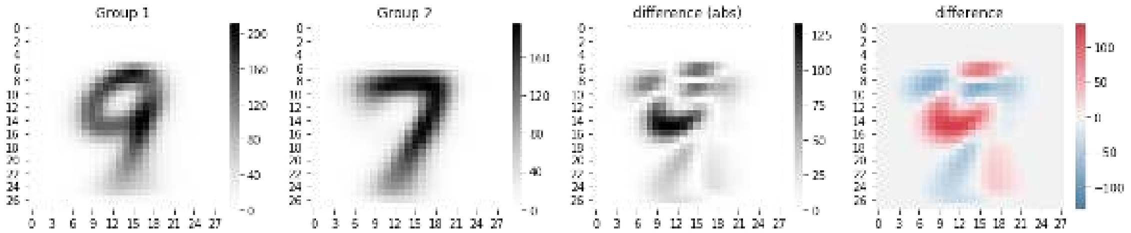


Figure 8.4: *Mean values for classes and their difference*

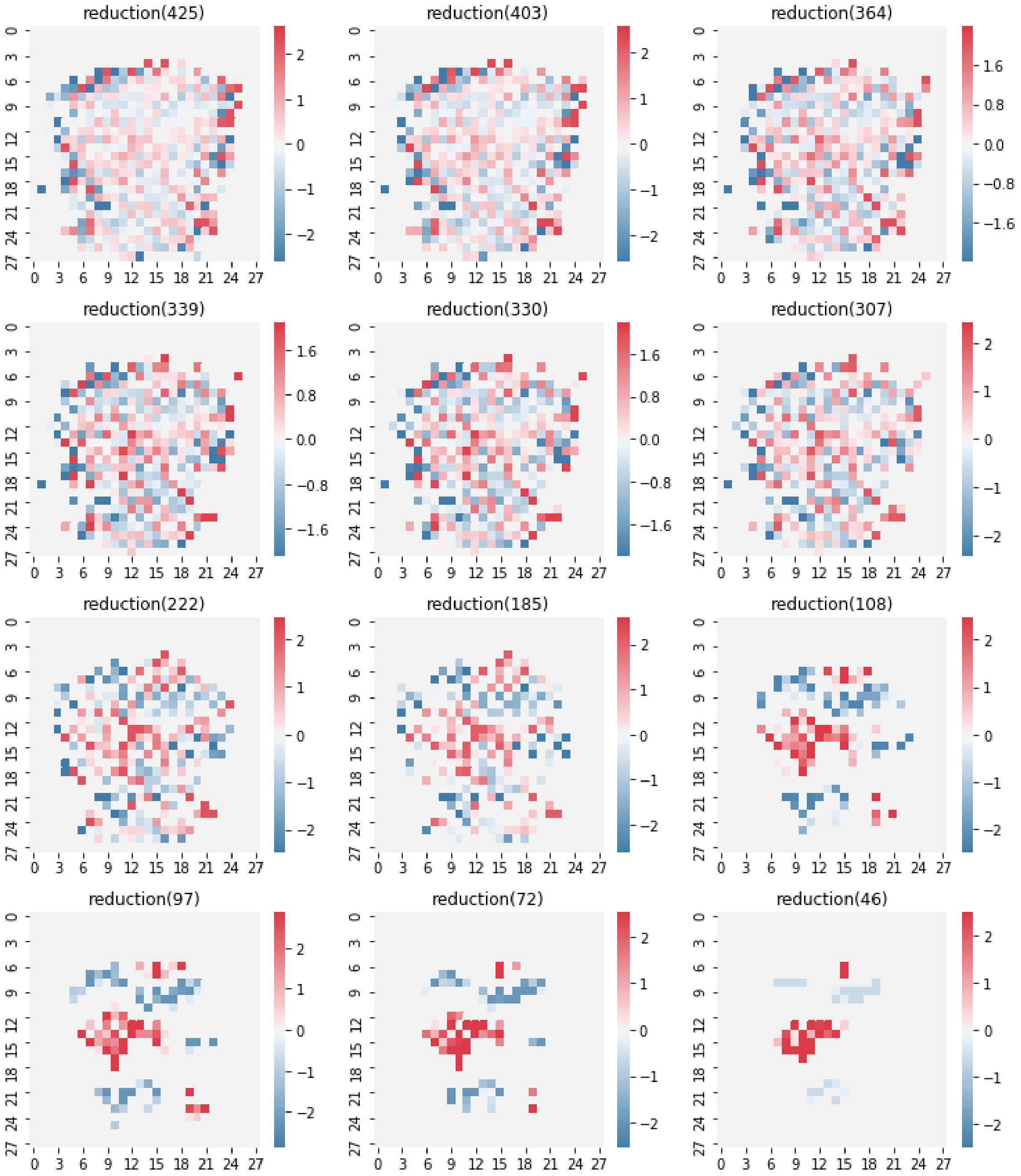


Figure 8.5: *Reductions attribute weights visualization*

In design of our method, we also proposed, that the R-Metric should be implemented in a way, that user can specify the weight for the factors that contribute to the metric. The inputs for different reductions can be seen in figure 8.6, and the corresponding R-Metric results are visualized in figure 8.7. For the weights numbers 0.1 and 1 were used to achieve these combinations. We can clearly see, that in this example, the major attribution to the result consists of changes in subset sizes. The score results for each reduction are all close together in a small range.

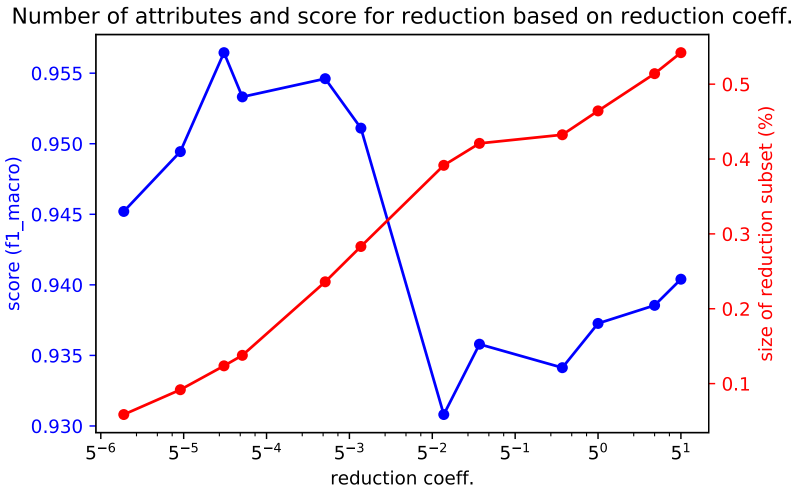


Figure 8.6: *R-Metric inputs for different reductions*

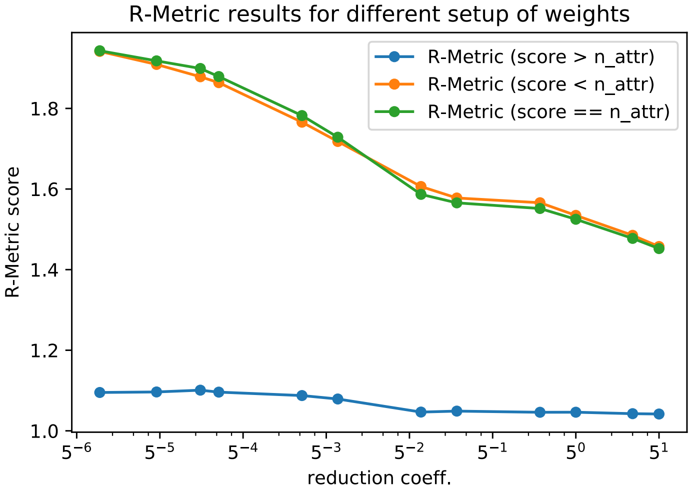


Figure 8.7: *R-Metric results for inputs from figure 8.6 and different weights setups*

8.2.3 Comparison with other methods for feature selection (importance evaluation)

We decided to compare our result with other methods for extracting important features. With this comparison, we want to add another level to of confirmation, if we really extracted the important features with our method.

8.2.3.1 Overlap of attributes subsets

Random forest provides the ranking of feature importance as an attribute after training. However, it does not specify the number of attributes it deems important. In addition, it only has positive weights for attributes, so we don't know, for which class which attribute is important. That is why we decided to simply compare the presence of attributes in both groups. And since the random forest does not select a subset of features, we take the same amount from top of ranking as our method based on logistic regression specifies. The same principle is applied for the results of SVM and KSS. For the SBFS, we simply select the subset of corresponding size the method provides. We can see these results in table 8.3 and also in figure 8.8.

Table 8.3: *Overlap of attribute subsets from proposed method with different feature selection techniques*

n. attr.	RF	KSS	SVM	SBFS
425	399	388	382	341
403	373	363	349	330
364	321	313	303	307
339	290	281	269	282
330	281	270	254	272
307	245	238	232	235
222	137	134	135	151
185	105	101	95	122
108	56	63	30	54
97	51	60	25	36
72	39	40	9	21
46	28	27	2	12

8.2.3.2 NDCG

With NDCG we first use the weights from these reference models and methods as true values of attributes. Since SBFS does not provide such weights it is omitted in this approach

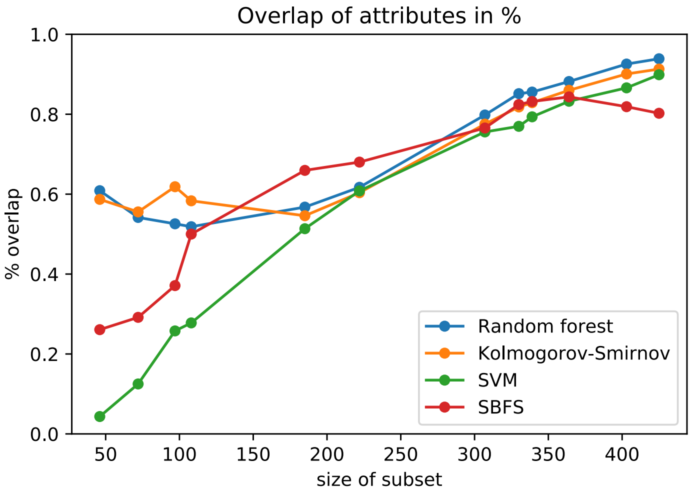


Figure 8.8: *Overlap of attribute subsets from proposed method with different feature selection techniques. We can see that the % overlap with our proposed method is quite similar between the compare methods while we are looking at the bigger reduction subsets, but quite different when it comes to smaller ones. This indicates, that if the algorithms and techniques choose bigger subsets, the difference between the attributes they choose is small. But when it comes to the point where only the best attributes are chosen, there is a difference between them, meaning the order of the attributes is quite different between these methods.*

of evaluation. In figure 8.9 and table 8.4 we can see the results we achieved with this metric.

Table 8.4: *NDCG metric calculated with weights from models as values of attributes in result set*

n. attr.	RF	KSS	SVM
425	0.9501	0.9372	0.9518
403	0.9471	0.9324	0.9488
364	0.9415	0.9232	0.9464
339	0.9370	0.9166	0.9415
330	0.9367	0.9168	0.9411
307	0.9333	0.909	0.9422
222	0.9087	0.8694	0.9204
185	0.8975	0.8558	0.8993
108	0.8753	0.8625	0.8154
97	0.8708	0.8709	0.8020
72	0.8623	0.8816	0.7595
46	0.8523	0.8921	0.7242

Next we use the order of these weights from the reference models and methods to assign the values for attributes. Similar as in the previous use of NDCG, here we also

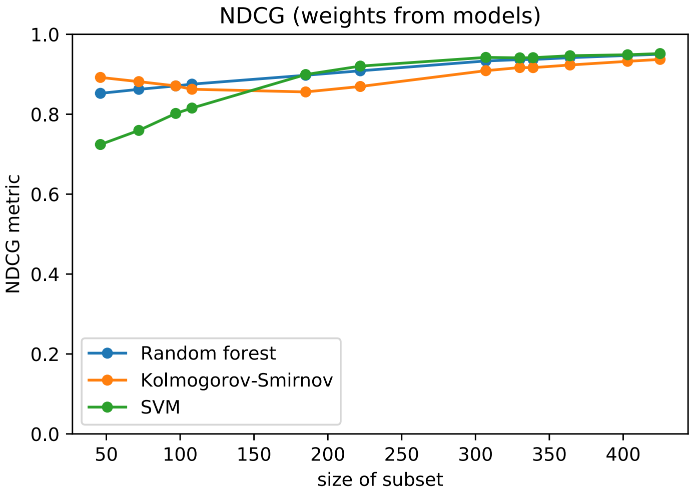


Figure 8.9: *NDCG metric calculated with weights from models as values of attributes in result set. With this metric the difference between the methods is smaller than with the simple % overlap. The question would be, how is that possible? One explanation is, that even though one method can have less attributes matching in the subset, they can be ranked higher than those in other method, resulting in similar results.*

can't evaluate the results from SBFS. If we would use the setup of SBFS, where the once pruned attributes can't be reintroduced to the subset, we would be able to extract the order of attributes based on the steps of the pruning. But since the attribute which was pruned in one step, can be later reintroduced in later subset, this is not possible. Nevertheless, the results we obtained can be seen in figure 8.10 and table 8.5.

Table 8.5: *NDCG metric calculated with order of attributes in model weights list as values of attributes in result set*

n. attr.	RF	KSS	SVM
425	0.8689	0.8649	0.8635
403	0.8620	0.8535	0.8530
364	0.8487	0.8394	0.8510
339	0.8381	0.8274	0.8367
330	0.8447	0.8310	0.8266
307	0.8286	0.8111	0.8350
222	0.7738	0.7619	0.7872
185	0.7626	0.7546	0.7431
108	0.8611	0.8320	0.6297
97	0.8719	0.8616	0.6077
72	0.8979	0.9046	0.5236
46	0.9286	0.9394	0.4985

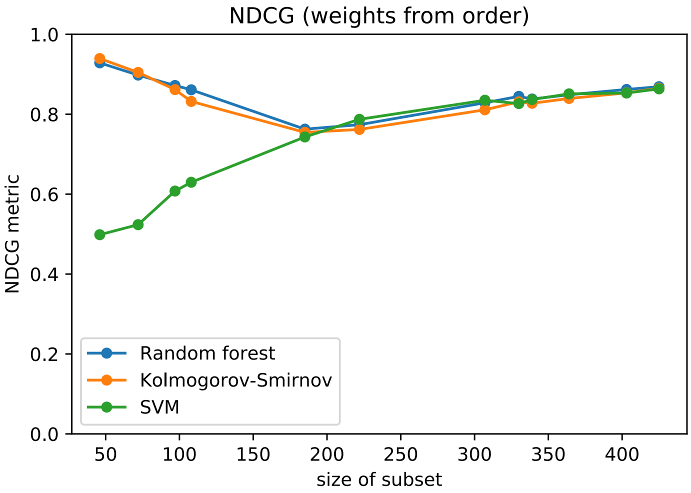


Figure 8.10: *NDCG metric calculated with order of attributes in model weights list as values of attributes in result set. With the use of other weights for attributes, where only their order represented their value, we can see that the difference between SVM and the other two methods got bigger. The reason can be that in SVM the weights got assigned more different weights and that's why its score got worse.*

8.2.3.3 Average precision

With average precision, we only take the order of features from the reference models and methods, and calculate the average precision for each reduction. But the order of attributes is only evaluated in subset corresponding to the size of reduction. That is why, in this step we can again evaluate the results of SBFS, wince it provides the subset that led to the designed reduction (we created all possible subsets). The results for this metric can be seen in figure 8.11 and table 8.6.

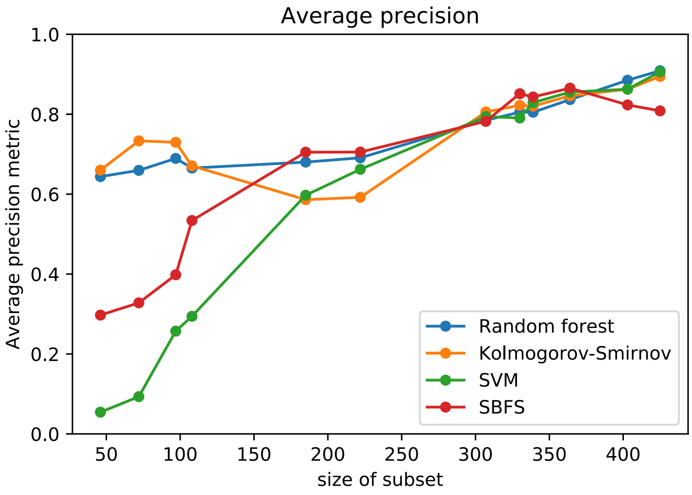


Figure 8.11: *Average precision metric calculated based on order of attributes in models results*

Table 8.6: *Average precision metric calculated based on order of attributes in models results*

n. attr.	RF	KSS	SVM	SBFS
425	0.9089	0.8946	0.9060	0.8084
403	0.8851	0.8626	0.8626	0.8233
364	0.8366	0.8461	0.8554	0.8653
339	0.8050	0.8202	0.8290	0.8432
330	0.8052	0.8222	0.7906	0.8514
307	0.7849	0.8057	0.7941	0.7822
222	0.6909	0.5920	0.6621	0.7053
185	0.6802	0.5861	0.5973	0.7050
108	0.6652	0.6707	0.2943	0.5340
97	0.6894	0.7297	0.2572	0.3982
72	0.6595	0.7334	0.0932	0.3280
46	0.6439	0.6600	0.0544	0.2973

8.3 MNIST dataset experiment - user study

From our user study, we managed to gather data from 21 participants for 3 groups comparisons. For each of this example, we create 3 sections of points. Each section represent some sentiment group of points based on participants opinions. First section represent the points, that overall participants marked as good for explanations of difference of attributes between groups. Second one consist of points deemed bad for explanations of difference between attributes of groups. And the third is section of points, which sentiment is unclear. Unclear means that these points were marked as good by some participants and as bad by others (size of both groups of participants needed to be greater than 15% of all participants for the border points). We then evaluate the true positive rate (TPR), false positive rate (FPR), true negative rate (TNR), false negative rate (FNR) and border overlap of points our method choose and those it did not for interpreting the difference based on attributes between two groups of data with the sentiment groups. We hold the sentiment groups as the objective truth against which we evaluate the results of our method. We want to maximize TPR and TNR values while minimizing FPR and FNR. If we overlap with unclear (border) section, we take it as a neutral indication that these points are interesting, because they are good for some participants, but bad for others.

Each example of the user study is evaluated separately. The groups from the examples can be seen in figures 8.12, 8.15 and 8.18. Based on these images, the participants in our user study has provided us with data, which we did in turn analyze, and identified the sentiment groups for each example presented in figures 8.13, 8.16 and 8.19. The results we calculated based on these sentiment groups for our reduction method are visualized in figures 8.14, 8.17 and 8.20. For results, we decided to create two graphs. In one (top image) we showcase the results for the points that our method selected. Here the labels are:

- TPR - points selected by our method and marked by people as good for explanation,
- FPR - points selected by our method and marked by people as bad for explanation,
- Border - points selected by our method and marked by some people as good and by some as bad for explanation.

In the second graph (bottom image) the visualization represents the results for point that our method did not select. The labels here are:

- TNR - points not selected by our method and marked by people as bad for explanation,
- FNR - points not selected by our method and marked by people as good for explanation,
- Border - points not selected by our method and marked by some people as good and by some as bad for explanation.

The percentage for TPR and FNR is calculated from points users selected as good and for FPR and TNR from points user selected as bad. The border percentage is calculated from all border points.

In the first example, we can see from the user data, that for the participants, there was a key group of points, that they really thought was good as an explanation of difference between the two groups, but the points for the bad explanations are more spread out, and not so consistent between users. From the results we can also see, that as the reduction got smaller (the interpretation got simpler), the difference between TPR and FPR got bigger. The border percentage acted similar to both TPR and FPR. We can also observe, that with the smallest reduction, we were able to almost zero out FPR which is really good, while keeping the TPR at around 20%.

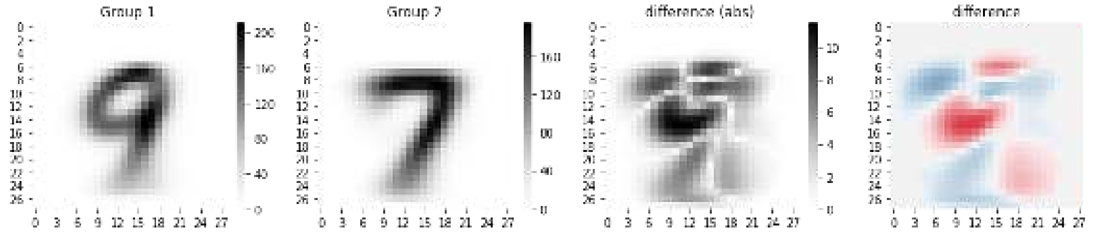


Figure 8.12: *Group means and difference of these groups from example 1*

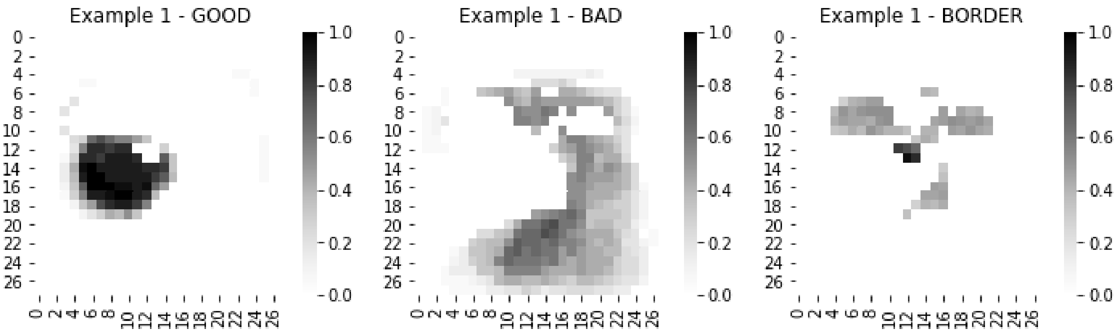


Figure 8.13: *Sentiment groups identified from participants data from example 1 (groups representing numbers 9 and 7)*

In the second example, we can again see from the user data, that there was a key group of points, that the participants really think was good as an explanation of difference between the two groups. The spread of those points selected as bad explanations of difference got even bigger. The results again show that the smaller reduction subsets got better results as those bigger ones. Once again, with the smallest reduction, we were almost able to zero out the FPR but in this case, we were also discarding more good points. This can be interpreted as a result of the reduction being too strict.

In the third example, which showcased possibly the most similar groups from the three examples, we can see that there is not any area as dominant for the good explanations as in the first two. But we can clearly see, that the unknown group is particularly bigger that in the first two examples. This forshadows the results, as we can see that for all the reductions made, we were not really able to achieve the same ratio between TPR and FPR as we were with some reductions in previous examples. However the overlap with the unknown sentiment group retained its value even in this example.

If we only look at one reduction from each example, that was marked as the best by the R-Metric score, and calculate the average metrics presented in the graphs across these examples, we get the numbers:

- TPR - 18.79%,
- FPR - 5.96%,
- good border - 25.44%.
- TNR - 94.04%,
- FNR - 81.21%,
- bad border - 74.56%.

We can see that the TPR is about at the value of FPR, which is good. It means, we are indeed selecting more point that are good than those that are bad for explanation to people. The high values in FNR is in part a result of the simpler explanations and the fact our method does not focus on single difference.

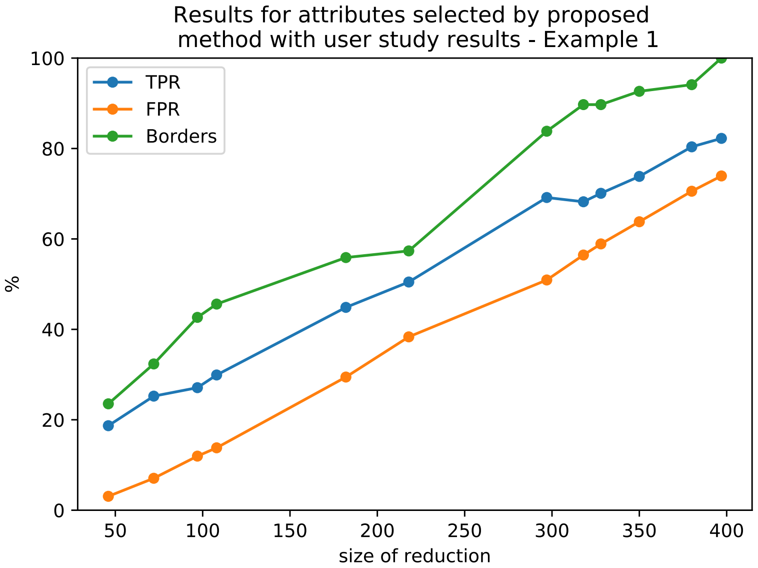
The main takeaway from this user study can be summarized in these points:

- With logistic regression, we are able to select a portion of points people want to see.
- People will focus on major difference, and in majority don't care about the smaller ones.
- People prefer simpler explanations.

8.4 Discussion

Based on the results we achieved, we can say our method is able to extract important features from data to split two groups apart. The techniques we compared our results with (Random forest, Kolmogorov-Smirnov score, SVM and Sequential feature selection) are all based on different approach. Some of them evaluating just the difference between the attributes in the groups, others take the interaction between attributes into account.

The results we achieved for the chosen evaluation metrics (overlap, NDCG, average precision) in addition to the evaluation of inverse reduction and random reduction, we can conclude, that the method we used, can indeed select the important attributes. We also managed to design the R-Metric, evaluation technique for selecting the best of the constructed reductions based on its test results and attribute reduction. It is possible to parameterize the R-Metric to achieve different results. This is a really important fact, since different task may require more focus on one aspect rather than the other. Also the results from the conducted user study tend to point in direction, that with our method, it is possible to provide an interpretation of difference between two groups based on attributes, which is to some degree good for people. Interesting fact we observed from the the data gathered by the user study participants, is that for humans, it is common to focus on one major difference between two segments of data. In our case it was the one important group of points, even though, there were multiple groups of points different between the segments. The participants showed a tendency for accepting the dominant difference as good explanation, but could not really agree on the minor ones. It is also important to say, that from the statistical point of view a group of 21 people, can't really hold a major statistical importance, but can be viewed as a proof of concept, that there is some tendency in the human behaviour and perception. In the case of the study we conducted, it was proven that the smaller (more simpler) interpretations of difference proved better for humans, as the results for chosen metrics with sentiment groups got better with decreasing size of attribute subset in most cases. For the best reductions across the examples, we managed to achieve triple values of FPR for TPR. Yes, still the the border percentage is still big for points chosen by our method and also those not, but that is not necessarily bad.



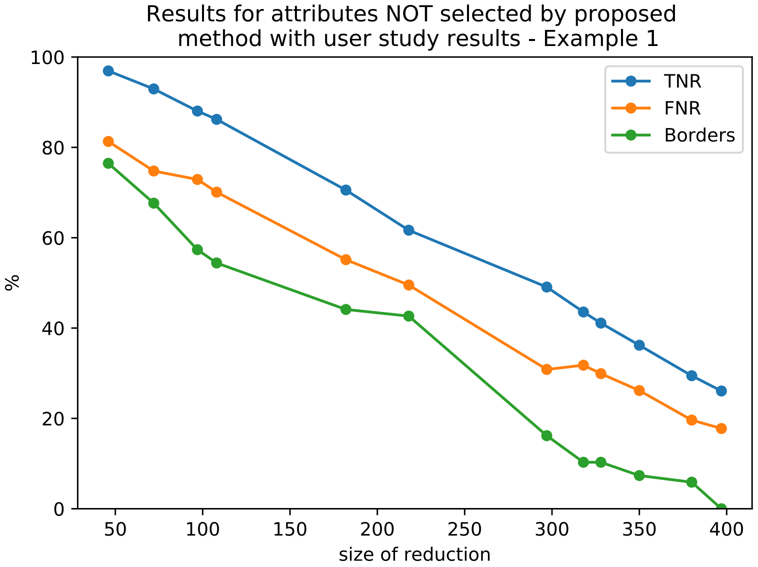


Figure 8.14: *Evaluation of different reductions on data from first example with sentiment groups identified by user study participants. Top image represents evaluation of the points our method selected as good ones. Bottom one represents those not selected, implicitly marked as bad ones. From the results we can see, that the smaller was the reduction subset our method provided, the better was the ratio between TPR and FPR and also TNR and FNR. We can say that from these results, and the characteristics of solutions our method provides, it is possible to deduce, that people want to see a simple explanations of the difference between the groups.*

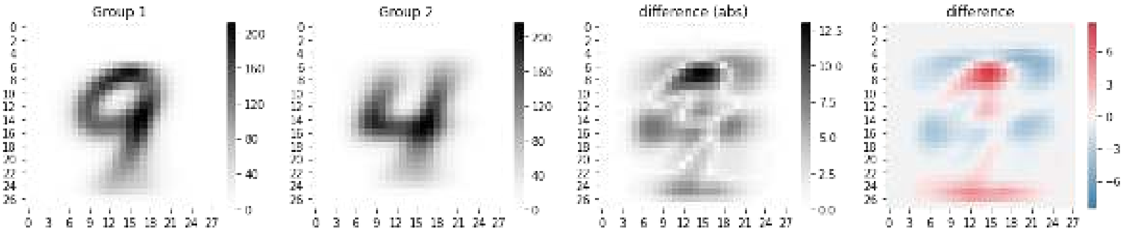


Figure 8.15: *Group means and difference of these groups from example 2*

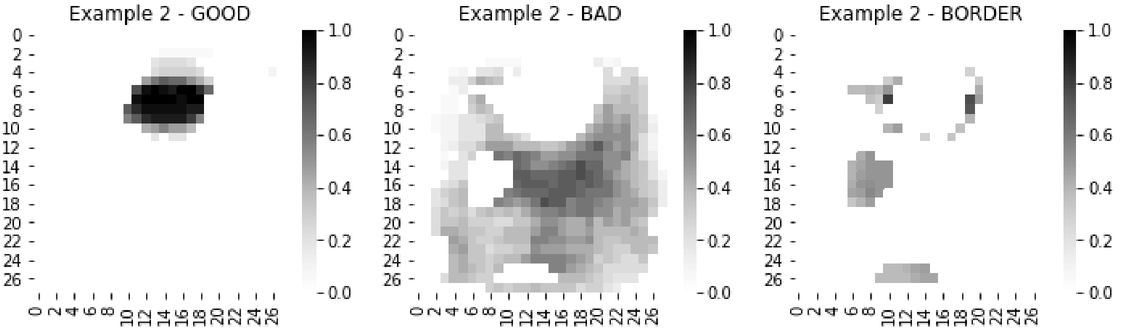
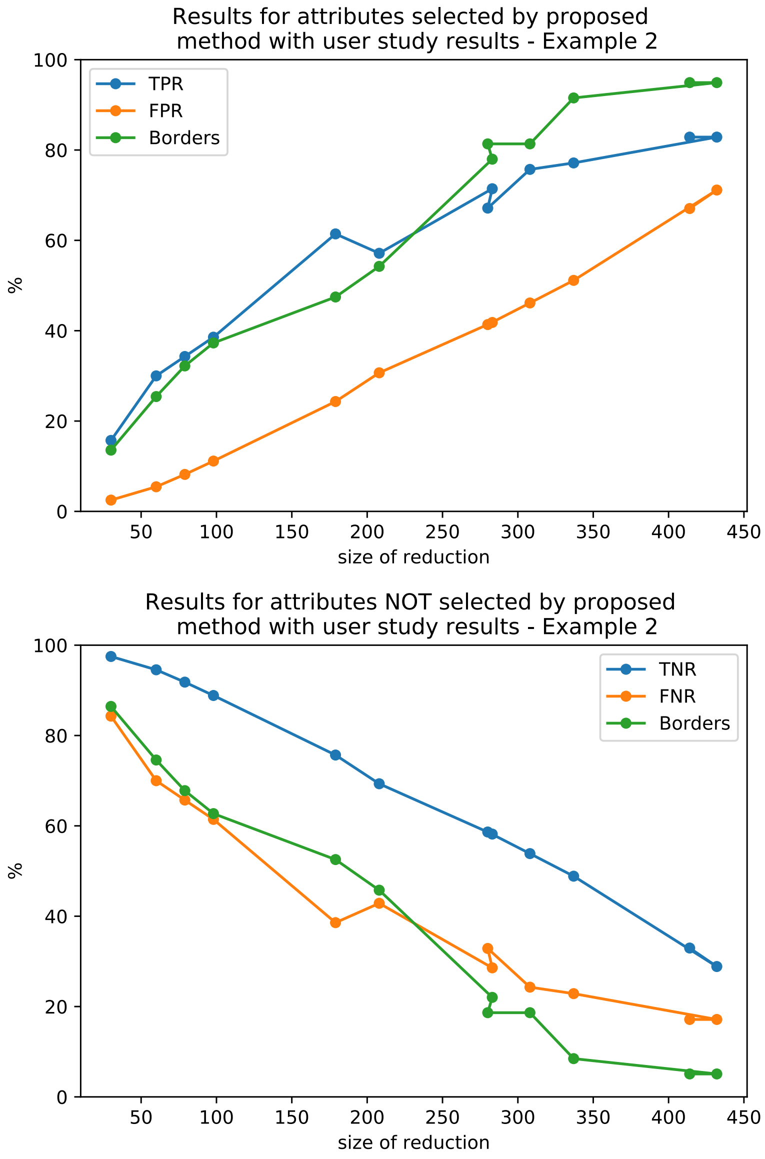


Figure 8.16: *Sentiment groups identified from participants data from example 2 (groups representing numbers 9 and 4).*


Results for attributes selected by proposed method with user study results - Example 2

Results for attributes NOT selected by proposed method with user study results - Example 2

Figure 8.17: *Evaluation of different reductions on data from second example with sentiment groups identified by user study participants. Top image represents evaluation of the points our method selected as good ones. Bottom one represents those not selected, implicitly marked as bad ones. In the second example, we can observe similar results as in the first one, except that with the smallest reduction, we can see more significant drop in the TPR. This can be a result of too strict reduction.*

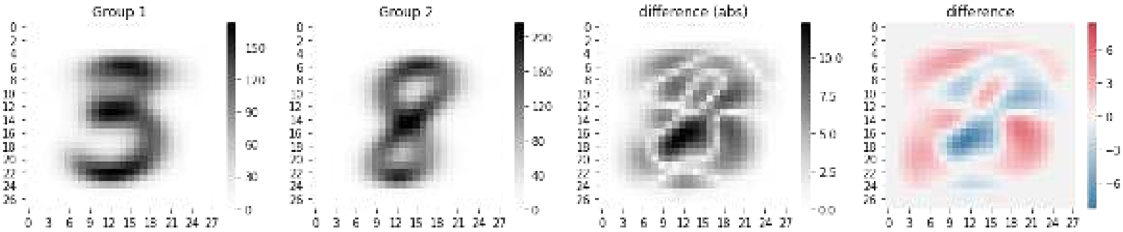


Figure 8.18: *Group means and difference of these groups from example 3*

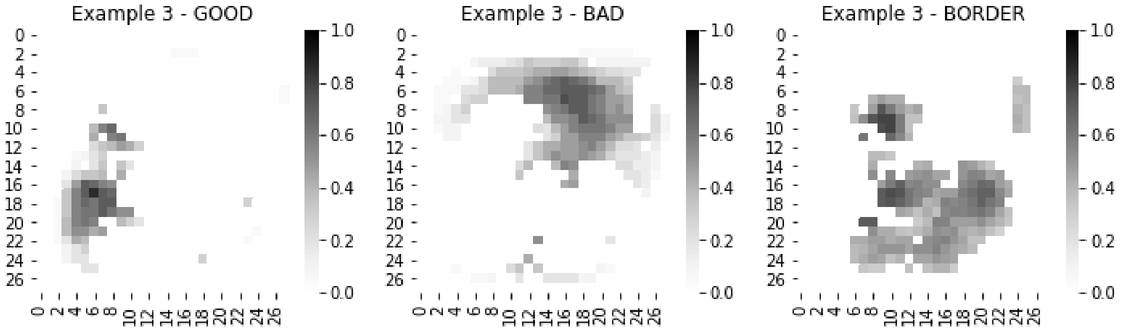


Figure 8.19: *Sentiment groups identified from participants data from example 3 (groups representing numbers 3 and 8).*

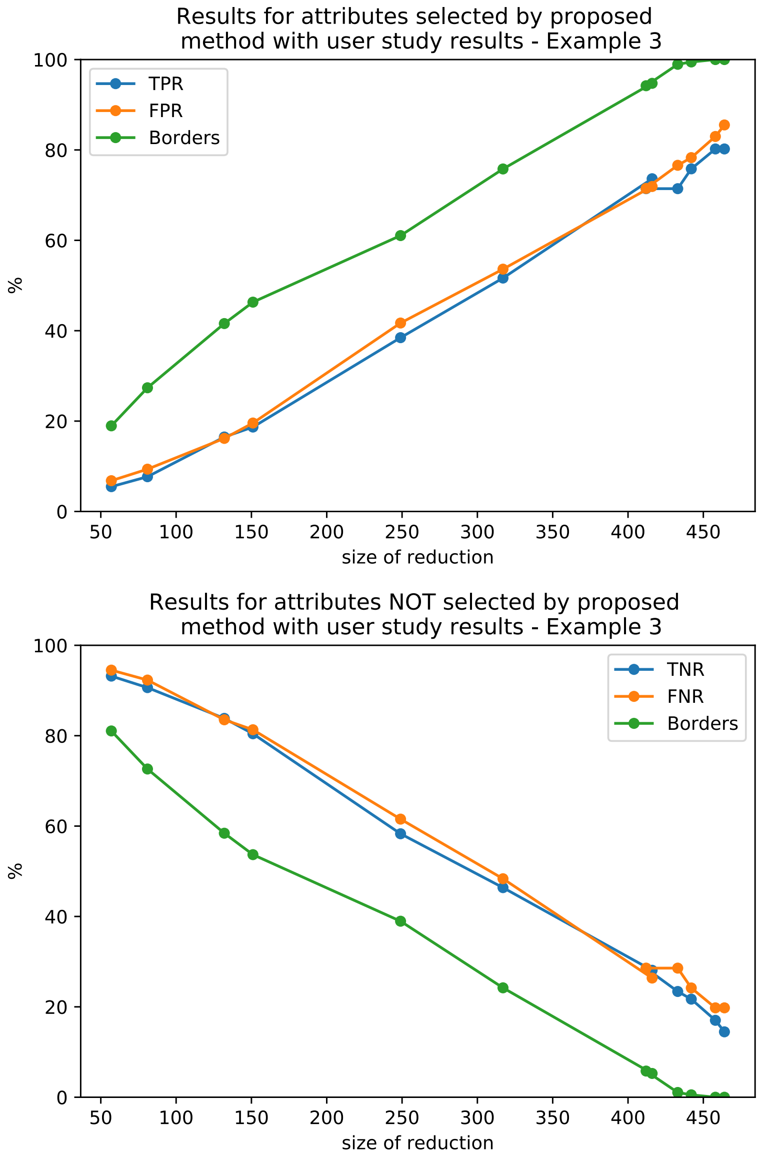


Figure 8.20: *Evaluation of different reductions on data from third example with sentiment groups identified by user study participants. Top image represents evaluation of the points our method selected as good ones. Bottom one represents those not selected, implicitly marked as bad ones. We can see that for this example, we were not really able to achieve the results from the previous examples. But we can see the border percentage is acting similar in spite of the change of TPR and FPR ratio. This can be caused by the fact, that the unknown sentiment group is bigger than in other examples. That can be the result of high similarity between the groups in example.*

Chapter 9

Conclusions

We, in our opinion, managed to achieve what we set out to do. Provide a method to interpret the result of clustering, to be more specific, to provide a method to interpret the difference between two segments of data, be it two classes or clustering groups or data from any other source. This method is model agnostic, meaning that it does not enforce the input data to be a result of a specific algorithm. The implication of this is that the explained difference between segments, is somewhat correlated to the method of selection that provided the input data. To put it simply, the method tries to explain with the use of surrogate model, why the former model decided to split these two segments apart. Also the fact that in our work we focused on MNIST dataset (image data), does not mean this method can't be used on different data. The only requirement for this method is that the input data need to be in numerical form.

9.1 Future work

There are still some possible upgrades to this method, we did not have the time to incorporate into this work, and remain a possible interest for the future.

The first would be an automated way of choosing the best reduction. In current state, the user need to specify the coefficient for all reductions to be made, and from these the best reduction is chosen. It would be much simpler and more sophisticated (also better result could possibly be achieved) if a sort of binary search would be present. The user would specify a starting coefficient and the algorithm would began search for best reduction. However, there re some problems that complicate the task.

- The first obvious one are the local maximums, that could occur for the metric and the search would stop prematurely.

- The second one is the problem of L1 regularization in logistic regression. We tried the greedy approach of searching for the best reduction, and it showed. The theory is that the smaller the coefficient used in the regularization, the stronger is the regularization, and in case of L1, the sparser the attribute vector should be. At least so we thought. But when we decreased the regularization coefficient only a little, we sometimes managed to get a bigger attribute vector than in the previous reduction.

We managed to implement a basic binary search (maybe closer to simulated annealing) of the coefficient, with s simple check for local minimum of R-Metric score, that was able to achieve some results, but we were not able to test it properly and it remains as more of a proof of concept. There is also the area of delivering the information of difference between two segment of data to the target audience (humans). In this work, we focused on MNIST dataset and for it, we were able to convey the results in the form of images, because the form of data in this dataset was suitable for it. However, since our method can be used on not just image data, it could be interesting to look at different means of visualizing the information we are able to provide.

The second would be adjusting the proposed method based on the user study results. Since the results showed that participants preferred a single group of points describing the major difference between the groups, it could interesting to try to incorporate this information into attribute selection process of the proposed method.

Chapter 10

Resumé

10.1 Úvod

V dnešnej dobe sa strojové učenie čoraz častejšie používa takmer v každom odvetví. Ľudia z týchto odlišných odvetví avšak nemusia mať technické zázemie pre tieto zložité algoritmy. Bez ohľadu na to však potrebujú rozumieť prečo sa výsledné modely správajú tak ako sa správajú. Ak totiž nerozumujú aj pozadiu rozhodnutí tohto modelu, musia sa spoliehať na slepú dôveru. A to nie je možné v každom odvetví, ako príklad môžeme uviesť medicínu. Z tohto dôvodu sa používajú rôzne prístupy na pomoc pri interpretácií modelov strojového učenia. My sme sa rozhodli zamerať na problém zhlukovania, konkrétne vysvetľovanie rozdielu na základe atribútov medzi dvoma skupinami z výsledku zhlukovania.

Našimi prínosmi pre túto problémovú oblasť sú:

- metóda založená na regularizácií lineárnych modelov pre vytvorenie redukcie (podmnožina atribútov z pôvodného priestoru atribútov)
- metrika na vyhodnotenie redukcie (R-Metric) pre výber najlepšej redukcie.

10.2 Analýza existujúcich metód pre vysvetľovanie výsledkov zhlukovania a rozdielu medzi segmentami dát

Ak bolo na vytvorenie zhlukov použité hierarchické zhlukovanie, je možné vysvetliť rozdiely medzi zhlukmi pomocou dendrogramu. Ten sa štruktúrou podobá stromu. Rozdiely v tomto strome sa však vizualizujú nie na základe atribútv dát. [4]

Každý výsledok zhlukovania je možné interpretovať pomocou bodového diagramu. Keďže však ľudia dokážu pre takýchto vizualizáciach efektívne vyhodnocovať maximálne 3 dimenzie (aj zobrazovacie zariadenia predstavujú rovnaké obmedzenie) je nutné pristupovať k redukcií dimenzionality, pri ktorej dochádza k strate informácií. Používajú sa napríklad algoritmy PCA alebo t-SNE . [11][10]

V práci [15], využili rozhodovacie stromy na vizualizáciu rozdielov medzi zhlukmi. Rozhodovacie stromy sú ľahko interpretovateľné pokiaľ sa jedná o malo rozmerné dáta. Pri veľkej dimenzionalite sa stávajú neprehľadnými. Taktiež je dôležité, že rozhodovacie stromy môžu mať problém ak neexistujú atribúty, ktoré vedia samostatne oddeliť skupiny alebo ich korelácia s triedou pozorovaní je príliš veľká.

Práca, ktorá sa najviac približuje k problému, ktorý riešime je technika vysvetľovania LIME [16]. Táto technika je zamýšľaná na vysvetlenie modelov klasifikácie ako čiernych skriniek, pri ktorých sa pomocou rozrušenia pôvodných atribútov snaží odhaliť, ktoré atribúty prispievajú, pre dané pozorovanie ku konkrétnej triede. Toto riešenie lokálne približuje modely pomocou lineárnych modelov.

Pri využití riešenia by bolo možné využiť dôležitosť atribútov na osekanie stromov a ich zjednodušenie pri viac dimenzionálnych dátach. Okrem rozhodovacích stromov sa na získanie dôležitosti atribútov dajú použiť napríklad SVM algoritmus alebo logistická regresia alebo rôzne štatistické metódy.

Jednou z nich je aj Kolmorogov-Smirnov dvojvzorkový test, ktorý na základe rozdelenia vzoriek dokáže určiť ich odlišnosť. [17]

10.3 Zhlukovacie techniky

Zhlukovanie je prístup učenia bez učiteľa. Dáta sú pri tomto prístupe na základe zvolenej techniky rozdelene do skupín. Poznáme viacero techník, pričom výsledky jednotlivých techník na rovnakých dátach sa medzi sebou môžu líšiť.

10.3.1 Zhlukovanie založené na ťažiskách

Pri tomto prístupe skupiny vznikajú iteratívne priraďovaním bodov k ťažiskám skupín. Existuje viacero prístupov ako vyberať ťažisko, pričom každý prístup vie ináč ovplyvniť výsledok. Výsledkom tohto typu zhlukovanie je dopredu určený počet skupín, ktoré majú tvar hypergule.[14]

10.3.2 Hierarchické zhlukovanie

Pri hierarchickom zhlukovaní sa pozorovania spájajú do skupín na základe ich vzdialenosti v priestore atribútov. Už názov hierarchicky napovedá tomu, že výsledkom je hierarchia skupín a ich vzťahov. Poznáme dva základne princípy tohto zhlukovania. Rozvratné (jedna pôvodná skupina sa rozdeľuje na menšie) a aglomeratívne (pozorovania sa spájajú do väčších skupín).[14]

10.3.3 Zhlukovanie založené na hustote

Toto zhlukovanie vytvára zhluky na základe hustoty pozorovaní v priestore atribútov. Husté oblasti (zhluky) sú oddelené oblasťami s riedkou hustotou pozorovaní (šumom). Výsledkom tohto zhlukovania je možné identifikovať zhluky rôznych tvarov.[14]

10.4 Topologická analýza údajov ako nástroj segmentácie údajov

Topologická analýza údajov (ďalej iba TDA), je princíp aplikovanej matematiky, ktorý sa využíva na analýzu datasetov. Je založený na technikách z topológie (ďalšia oblasť matematiky). Topológia sa zaoberá definíciou telies v priestore na základe zbierky pravidiel a deformáciou daných objektov.[24] Existujú dva spôsoby analýzy datasetu pomocou TDA. Prvý je štatistická (matematická) analýza. Pri tomto spôsobe nevizualizujeme žiadne tvary. Zisťujeme len to, či sa tam nejaké tvary nachádzajú, a aké majú charakteristiky. Pri hľadaní týchto tvarov je nutné zašpecifikovať rôzne atribúty pre hľadanie tvarov. Jedným z najdôležitejších je polomer oblasti okolo bodu, ktorý nám hovorí, či majú byť dva body spojené. Príklad môžeme vidieť na obrázku 10.1[5].

Preto sa zaviedol pojem odolná homológia ako riešenie zmien v tvaroch v závislosti od zmeny polomeru okolo bodov. Dá sa vizualizovať napríklad pomocou diagramov odolnosti alebo čiarových kódov. Cieľom je identifikovať tvary, ktoré pretrvávajú aj naprieč zmenami v polomere, a naopak byť schopný odfiltrovať tvary ktoré nepretrvávajú ako šum. Toto je teda prvý z dvoch vyššie spomínaných postupov.

Druhým spôsobom je vizualizácia datasetu. Keďže ľudia väčšinou nie sú schopný vnímať viac ako 3 dimenzie (niekedy 4), používa sa často pri vizualizáciach dát redukcia dimenzionality. Pri nej však dochádza k strate informácii. A to je výhodou TDA, ktoré sa nesnaží vizualizovať dáta ako také, ale len tvary nájdené v týchto dátach. Samozrejme, aj

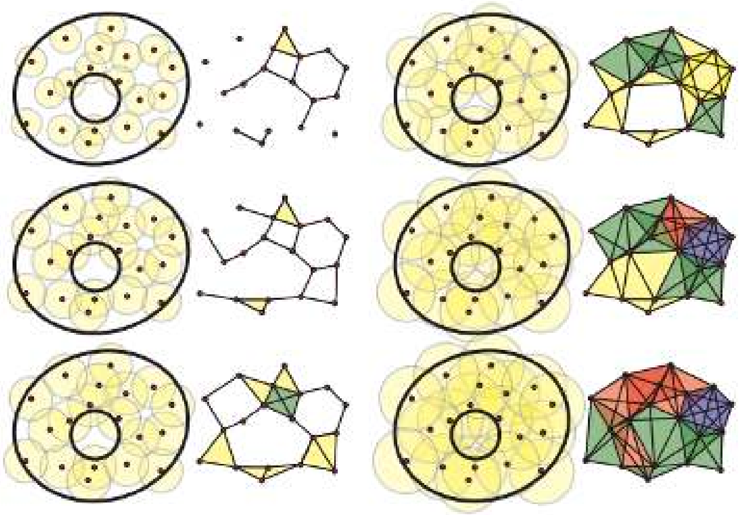


Figure 10.1: *Zmena nájdených tvarov so zmenou polomeru [5]. Na základe veľkosti polomeru okolo bodu (žlté kruhy), sa môžu výsledné tvary veľmi líšiť. Čím väčší polomer, tým väčšia šanca na spojenie dvoch bodov, a teda hustejšie prepojené tvary. Môžeme vidieť, že výsledky môžu byť zavádzajúce, pretože torus tvar, ktorý je naozaj v dátach z obrázku, sa podarilo dosiahnuť len jedným alebo dvomi nastaveniami vizualizovaných riešení.*

tieto tvary je nutné reprezentovať v podobe zobraziteľnej na zobrazovacích zariadeniach, a práve preto sa v TDA využíva mapovač. V tejto práci sa sústredíme na tento postup.

10.4.1 Úloha mapperu v TDA

Mapovač by sa dal označiť za nástroj, ktorý z pôvodnej podoby prevedie údaje do takej, ktorá následne umožní analyzovať údaje pre výskyt tvarov a ich vizualizáciu. Kroky mapovača by sa dali rozdeliť na nasledovné:

- rozdelenie datasetu na prekrývajúce sa podmnožiny,
- zhlukovanie podmnožín,
- nájdenie prekryvu zhlukov,
- vytvorenie matice vzdialeností medzi zhlukmi.

Týmito krokmi vznikne sieť prepojených bodov, reprezentujúcich tvary. Vzdialenosti medzi zhlukmi zaručia rozloženie v priestore. Ukážku krokov mapovača môžeme pozorovať na obrázku 10.2.

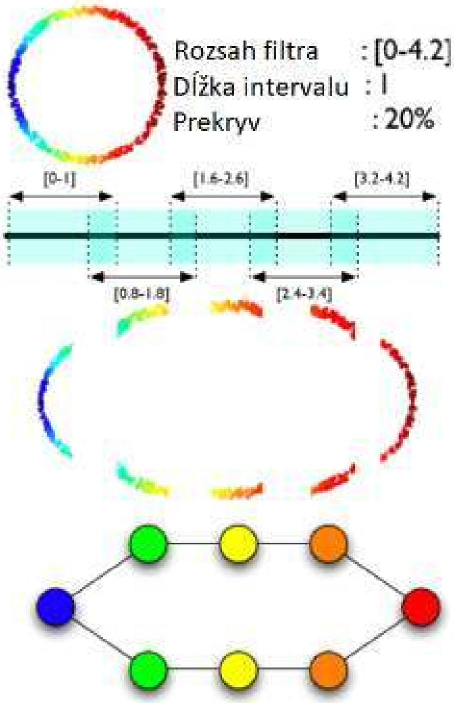


Figure 10.2: *Kroky mapovača aplikované na dataset reprezentujúci kruh [18]*

10.4.2 Rozdiel medzi TDA a bežnými zhlukovacími algoritmami

Základným rozdielom medzi TDA a zhlukovacími algoritmami je, že zatiaľ čo zhlukovacie algoritmy sú segmentačné nástroje, TDA je technika exploratívnej analýzy. Druhým je, že výsledok zhlukovania sú oddelené skupiny pozorovaní. Pri TDA môžeme analyzovať aj výsledok, keď sa žiadne skupina úplne neoddelí, ale môže sa nachádzať niekde na okraji. S TDA teda vieme analyzovať aj údaje, ktoré by pomocou zhlukovacích algoritmov nedávali rozumné výsledky. TDA taktiež na rozdiel od niektorých zlukovacích algoritmov, napríklad tých založených na ťažiskách, dokáže identifikovať aj skupiny iných tvarov ako hyperguľa. Treba však povedať, že TDA nie je prioritne určené na segmentáciu údajov, ale skôr na analýzu údajov, aby sme videli s čim pracujeme.

10.5 Metódy vysvetľovanie rozdielov medzi segmentami údajov

Zhlukovanie sa najčastejšie vyhodnocuje metrikami, ktoré vyhodnocujú kvalitu zhlukov ako takých. V našom prípade ale ani tak nejde o kvalitu zhluku, skôr o rozdiel medzi zhlukmi v atribútoch. Pre tento účel by bolo možné využiť rôzne metódy, či už štatistické ako Kolmogorov-Smirnov skóre, alebo nejakú priemernú hodnotu atribútu pre zhluk. Taktiež by sa dalo využiť interpretovanie klasifikátorov. V obidvoch prípadoch by ale vo veľkej miere išlo o vyhodnotenie dôležitosti samostatného atribútu. My by sme však chceli nájsť takú kombináciu atribútov, ktorá dokopy, aj vzťahmi medzi atribútmi dosahuje najlepšie rozdelenie segmentov. Ako možné riešenie sa ponúkajú lineárne modely a ich regularizácia.

10.5.1 Regularizácia lineárnych modelov

Regularizácia lineárnych modelov je množina techník, ktorých hlavným cieľom je upraviť proces učenia lineárnych modelov pre dosiahnutie lepších výsledkov zlepšením ich schopnosti generalizovať (zabrániť preučeniu). Pri strojovom učení je pretrénovanie častý problém, kedy sa model až príliš prispôsobí na konkrétne dáta, na ktorých bol vytvorený, a potom na nových dátach dosahuje horšie výsledky. Práve tomuto sa snaží regularizácia zabrániť pri lineárnych modeloch. Poznáme dve základné formy regularizácie. L1 taktiež nazývanú Lasso a L2 nazývanú Ridge. Základným rozdielom medzi nimi je, že L1 používa súčet absolútnych hodnôt váh na regulovanie koeficientov, zatiaľ čo L2 využíva súčet druhých mocnín týchto váh. Výsledkom týchto rôznych aplikácií, je okrem iného aj fakt, že L1 regularizácie produkuje pre lineárny model riedky vektor atribútov. Teda niektoré atribúty nadobudnú váhu 0 a pri trénovaní nehrajú žiadnu úlohu. To ako veľmi riedky je tento vektor atribútov záleží od zvoleného regularizačného koeficientu. Na rozdiel od toho, s použitím L2 regularizácie si váhy atribútov vždy zachovajú hodnotu väčšiu ako 0 a teda nedochádza k vzniku riedkeho vektoru.[12] Pre klasifikáciu existujú tri základné lineárne modely. Jedná sa o logistickú regresiu, SVM klasifikátor a perceptron.

10.6 Motivácia, ciele a hypotézy našej práce

Motivácia našej práce vychádza z faktu, že modely strojového učenia zvyknú poskytovať komplexné riešenia, ktoré sú niekedy až tak komplexné, že je ťažké im rozumieť. Preto sa v našej práci sústredíme na spracovanie výsledkov zhlukovania pre objasnenie rozhodnutí zhlukovacieho modelu. Cieľom našej práce je: *Poskytnúť metódu, ktorá uľahčí interpretáciu zhlukovacích modelov a ich rozhodnutí.* Na dosiahnutie tohto cieľa sme navrhli metódu založenú na L1 regularizácií lineárnych modelov, pomocou ktorej chceme identifikovať podmnožinu atribútov, ktorá dokáže oddeliť 2 segmenty pozorovaní, je čo najmenšia, ale zároveň si zachováva schopnosť dostatočne oddeliť dané segmenty pozorovaní.

Na vyhodnotenie našej práce, sme sa rozhodli sledovať tieto hypotézy:

- Zástupný model môže byť použitý na nájdenie dôležitých atribútov pre celé zvolené triedy zo zhlukovacieho modelu.
- Výsledok lineárneho modelu klasifikácie s L1 regularizáciou poskytuje dostatok informácií o atribútoch na zjednodušenie interpretácie rozhodnutí modelu zhlukovania.

10.6.1 Štatistické vyhodnotenie

V štatistickom vyhodnotení sa sústredíme na vyhodnotenie prvej hypotéz. A teda či pomocou zástupného modelu vieme identifikovať dôležité atribúty pre celé triedy. Toto vyhodnotenie ma dve vrstvy. V prvej vyhodnotíme, či po odstránení nami vybranej podmnožiny z pôvodnej množiny atribútov dosiahneme horšie výsledky, ako keď odstránime náhodnú rovnako veľkú podmnožinu. V druhej časti vyhodnotíme našu metódu voči inými metódam na selekciu atribútov. Algoritmy, s ktorými sa budeme porovnávať sú Náhodný les (Random forest), SVM, Kolmogorov-Smirnov skóre a sekvenčná selekcia atribútov. S týmito metódami sa porovnáme pomocou prekryvu vybratých atribútov, normalizovaný diskontovaný kumulatívny zisku (NDCG - normalized discounted cumulative gain)[9] a priemernej presnosti[23].

10.6.2 Používateľská štúdia

V rámci používateľskej štúdie sme sa zamerali na vyhodnotenie, toho či atribúty, ktoré vyberie nami navrhnutá metóda sú tie, ktoré chcú ľudia vidieť ako rozdiel medzi dvoma skupinami. Štúdia je postavená na MNIST datasete. Používatelia v nej vyberajú skupiny atribútov, ktoré dobre vysvetľujú rozdiel medzi skupinami a tie, ktoré ho vysvetľujú zle. Následne vyhodnotíme úspešnosť našej metódy voči zozbieraným údajom od používateľov.

10.7 Navrhovaná metóda

Navrhujeme metódu, ktorá bude vedieť vybrať dôležité atribúty a poskytnúť spôsob vysvetlenia rozdielu v týchto atribútoch pre dva dané segmenty.

10.7.1 Výber dôležitych atribútov

Na výber dôležitých atribútov využívame logistickú regresiu s L1 regularizáciou.

10.7.2 Vytvorenie modelu logistickej regresie

Dáta pre dve skupiny zo zhlukovania rozdelíme na trénovaciu a testovaciu množinu, aby sme mohli vyhodnocovať vytvorené redukcie. Keď že veľkosť redukcie je ovplyvňovaná parametrom regularizácie, vyskúšame niekoľko týchto parametrov a na základe výsledkov prislúchajúcich redukcií vyberieme tú najlepšiu. Kvalita nášho riešenie však nezáleží len od výsledkov na testovacej množine, ale aj od veľkosti redukcie. Chceme nájsť kompromis medzi zhoršením výsledkov a redukciou počtu atribútov.

10.7.3 Vyhodnotenie redukcie

Redukcie vyhodnocujeme na testovacej množine pomocou krížovej validácie. Pre porovnanie taktiež vyhodnotíme výsledok pre pôvodnú množinu atribútov.

10.7.4 Metrika hodnotenia redukcie: R-Metric

Vstupom pre R-Metric je výsledok pôvodnej množiny atribútov na testovacej množine, redukcie na testovacej množine a veľkosť pôvodnej množiny atribútov a redukovanej množiny atribútov. Z týchto hodnôt sa vypočíta finálne hodnotenie redukcií, podľa ktorého vyberieme najlepšiu.

10.8 Vyhodnotenie navrhovanej metódy

Pre navrhovanú metódu vyhodnocujeme samostatne štatistické vyhodnotenie a používateľskú štúdiu.

10.8.1 MNIST dataset expriment - štatistické vyhodnotenie

Aplikovali sme TDA (knižnica Kepler mapper) na MNIST dataset. Z vizualizácie sme následne izolovali dva segmenty pozorovaní, ktoré majoritne reprezentovali čísla 9 a 7.

Z redukcií, ktoré sme vytvorili z pôvodných 784 atribútov sme ako najlepšiu na základe výsledkov R-Metric vybrali tú s veľkosťou 46 atribútov. Ktorá mala v porovnaní s pôvodným výsledkom na testovacej množine (94,43%) výsledok 94,52%. Taktiež sa nám potvrdilo, že výsledok invertnej redukcie bol o 1% horší ako ten náhodnej redukcie rovnakej veľkosti. Pokiaľ sa jedná o výsledky v porovnaní s inými metódami na selekciu atribútov výsledky sú uvedené v tabuľke 10.1.

Table 10.1: *Výsledky rôznych metrík pre porovnanie s metódami pre selekciu atribútov pre najlepšiu redukciu o veľkosti 46 atribútov. RF - random forest, KSS - Kolmogorov-Smirnov skóre, SVM, SBFS - sekvenčná spätná selekcia atribútov*

metrika	RF	KSS	SVM	SBFS
overlap	28	27	0.2	12
NDCG (váhy)	0.8523	0.8921	0.7242	——
NDCG (poradie)	0.9286	0.9394	0.4985	——
priemerná presnosť	0.6439	0.6600	0.0544	0.2973

10.8.2 MNIST dataset expriment - používateľská štúdia

Našej používateľskej štúdie sa zúčastnilo 21 ľudí. Ich úlohou bolo v troch príkladoch ohodnotiť dobré a zlé atribúty pre oddelenie dvoch skupín. V prvom príklade sa jednalo o skupiny obrázkov čísel 9 a 7, v druhom 9 a 4 a v treťom 3 a 8. Zo zozbieraných údajov sme nakoniec pre každý príklad vytvorili 3 skupiny sentimentu - dobré body na vysvetlenie, zlé body a body na rozmedzí, keďže pre niektorých účastníkov boli zlé a pre iných dobré. Pre tieto skupiny sme následne v každom príklade vyhodnotili porovnanie s atribútmi, ktoré vybrala, resp. nevybrala naša metóda. Pre najlepšie redukcie z každého príkladu môžeme vidieť výsledky v tabuľke 10.2.

Table 10.2: *Výsledky user study pre najlepšie redukcie z každého príkladu.*

príklad	veľkosť redukcie	TPR	FPR	dobrá hranica	TNR	FNR	zlá hranica
č. 1	46	18,69%	3,07%	23,53%	96,93%	81,31%	76,47%
č. 2	60	30,00%	5,45%	25,42%	94,55%	7,00%	74,58%
č. 3	82	7,69%	9,36%	27,36%	90,64%	92,31%	72,63%

10.9 Zhodnotenie

Na základe dosiahnutých výsledok môžeme konštatovat', že sa nám podarilo splnit' ciel' práce. Pokial' sa jedná o výsledky štatistického vyhodnotenia podarilo sa nám ukázat', že vieme vybrat' pomocou našej metódy dôležité atribúty. Pri porovnaní s inými metódami selekcie atribútov boli niektoré výsledky slabšie, čo sa však do istej miery dá pripísat' rôznemu princípu algoritmov, ktoré boli zodpovedné za výber. Z výsledkov používatel'skej štúdie sa ukázalo, že nie je také jednoduché trafit' sa do očakávania l'udí. Zaujímavým poznatkom je, že l'udia sa sústredia pri vyhodnocovaní rozdielu na najväčší rozdiel, a ostatné menšie rozdiely pre nich nie sú vel'mi (skoro vôbec) dôležité.

Bibliography

[1] Gunnar Carlsson. Why tda and clustering are not the same thing. https://www.ayasdi.com/blog/machine-intelligence/why-tda-and-clustering-are-different/, March 2016. Accessed: 2018-04-29.

[2] Gunnar Carlsson. Understanding the distinction between clustering and tda. https://www.ayasdi.com/blog/bigdata/understanding-distinction-clustering-tda/, July 2017. Accessed: 2018-04-29.

[3] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, Dec 1999.

[4] Tal Galili. dendextend: an r package for visualizing, adjusting and comparing trees of hierarchical clustering. *Bioinformatics*, 31(22):3718–3720, 2015.

[5] Robert Ghrist. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society*, 45(1):61–75, 2008.

[6] Michael Gromov. Curvature, diameter and betti numbers. *Commentarii Mathematici Helvetici*, 56(1):179–195, Dec 1981.

[7] Steve R Gunn et al. Support vector machines for classification and regression. *ISIS technical report*, 14(1):5–16, 1998.

[8] Marwan Hassani and Thomas Seidl. Using internal evaluation measures to validate the quality of diverse stream clustering algorithms. *Vietnam Journal of Computer Science*, 4(3):171–183, Aug 2017.

[9] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Trans. Inf. Syst.*, 20(4):422–446, October 2002.

[10] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[11] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303 – 342, 1993.

[12] Arnold Neumaier. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM review*, 40(3):636–666, 1998.

[13] Andrew Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the Twenty-first International Conference on Machine Learning*, ICML '04, pages 78–, New York, NY, USA, 2004. ACM.

[14] Mahamed GH Omran, Andries P Engelbrecht, and Ayed Salman. An overview of clustering methods. *Intelligent Data Analysis*, 11(6):583–605, 2007.

[15] Olivier Parisot, Mohammad Ghoniem, and Benoît Otjacques. Decision trees and data preprocessing to help clustering interpretation. In *Proceedings of 3rd International Conference on Data Management Technologies and Applications*, DATA 2014, pages 48–55, Portugal, 2014. SCITEPRESS - Science and Technology Publications, Lda.

[16] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144. ACM, 2016.

[17] Matteo Rucco, Lorenzo Falsetti, Damir Herman, Tanya Petrossian, Emanuela Merelli, Cinzia Nitti, and Aldo Salvi. Using topological data analysis for diagnosis pulmonary embolism. *arXiv preprint arXiv:1409.5020*, 2014.

[18] Gurjeet Singh, Facundo Mémoli, and Gunnar E. Carlsson. Topological methods for the analysis of high dimensional data sets and 3d object recognition. In *SPBG*, 2007.

[19] T Soni Madhulatha. An overview on clustering methods. *IOSR Journal of Engineering*, 2, 05 2012.

[20] Hendrik Jacob van Veen and Nathaniel Saul. Keplermapper. http://doi.org/10.5281/zenodo.1054444, Jan 2019.

[21] STROTHER H. WALKER and DAVID B. DUNCAN. Estimation of the probability of an event as a function of several independent variables. *Biometrika*, 54(1-2):167–179, 1967.

[22] Cen Wan. *Feature Selection Paradigms*, pages 17–23. Springer International Publishing, Cham, 2019.

[23] Mu Zhu. Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, 2:30, 2004.

[24] Afra Zomorodian. Topological data analysis. *Advances in applied and computational topology*, 70:1–39, 2012.

Appendix A

Plans for work

A.1 Evaluation of the last winter semester

In the last winter semester we were able to fulfill most of goals we set for it. The choice of mapper was between two of the implementations. Based on the tests we conducted and possibilities they provided we have chosen one of them. Because the alterations to the mapper are not a key component and goal of our work, we decided to push this task to next semester and focus on the goal of our work, the method for interpretation of clustering models. We were able to fulfill this task and constructed such a method. We also conducted test with MNIST dataset. In this semester, we didn't manage to test the whole pipeline from TDA through our method to results.

Table A.1: *Plan for each month for last winter semester*

Month	**Plan**
September	Studying of various free implementations of TDA mapper and choosing the best option
October	Adjustments to chosen implementation of mapper to fulfill our requirements for choosing segments
November	Implementation of method for interpreting features that best distinguish two data segments
December	First tests of created method on datasets, possible adjustments to the method based on tests
January	Test for whole pipeline from TDA through our method to results.

A.2 Evaluation of the last semester

In this semester, we focused on evaluation of our method in comparison with other feature extraction algorithms (random forest, kolmogorov-smirnov score, SVM, sequential feature). We adjusted the mapper results so that we are able to select segments of data, and test the whole pipeline. We thought about adding additional functionality to our method by adding a deep look into segment parts analysis, more exactly how parts of one segment differ one from each other based on attributes (similar to hierarchical clustering), but because of the lack of time we didn't pursue this task. We carried out an experiment with people to assess the goodness of our solution. In the end we polished the evaluation of our work, the text of the work and presentation of the results. During this semester, we also presented out work on IIT.SRC conference, which took some amount of our time, but was good experience and we got some feedback for our work.

Table A.2: *Plan for each month for last semester*

Month	**Plan**
February	Evaluation of feature extraction method and segment selection from mapper results
March	Implementation of deep segment analysis
April	Polishing of work and results presentation

Appendix B

Technical documentation

In this appendix, we focus on specification of our method, core used libraries and other technical requirements.

B.1 Specification

We used scikit-learn library for machine learning tasks.

B.1.1 Proposed method

Our proposed method in fact consist of multiple python methods, which we extracted into a standalone python file. In this python file we defined class LinearReductionMethod. The methods in this class are:

- alpha_metric()
- get_conf_int()
- test_reduction()
- normalized_atribute_reduction()
- test_random_reduction()
- get_best_reduction()

The use of this class is quite simple. User creates new instance of this class and to find the best reduction he just calls the method get_best_reduction(). In figure B.1 we can see the use of these methods for reduction creation in form of sequence diagram.

Method get_best_reduction

The inputs of this method are:

- observations data,
- observations classes,
- list of regularization coefficients to try,
- attribute weight for R-Metric,
- score weight for R-Metric,
- number of parallel tasks to start.

As first step the observations data are split to train and test. Then baseline results are calculated. Afterwards the results for each regularization coefficient are calculated. From these results the best reduction is chosen.

Method normalized_atribute_reduction

In this method, the reduction using logistic regression and L1 regularization is created based on train data. Afterwards the results on test data are calculated for reduction and inverse reduction. Also the random reduction are calculated for the corresponding reduction.

Method test_reduction

This method calculates the results for created reduction and inverse reduction if transform flag is set as parameter. Otherwise calculates the baseline results for test data and original attribute set.

Method test_random_reduction

The random reduction results corresponding to a created reduction are calculated in this method. The method averages the result from number of tries specified by parameter to get more consistent results.

Method get_conf_int

Calculates 95% confidence intervals from cross validation scores presented as input.

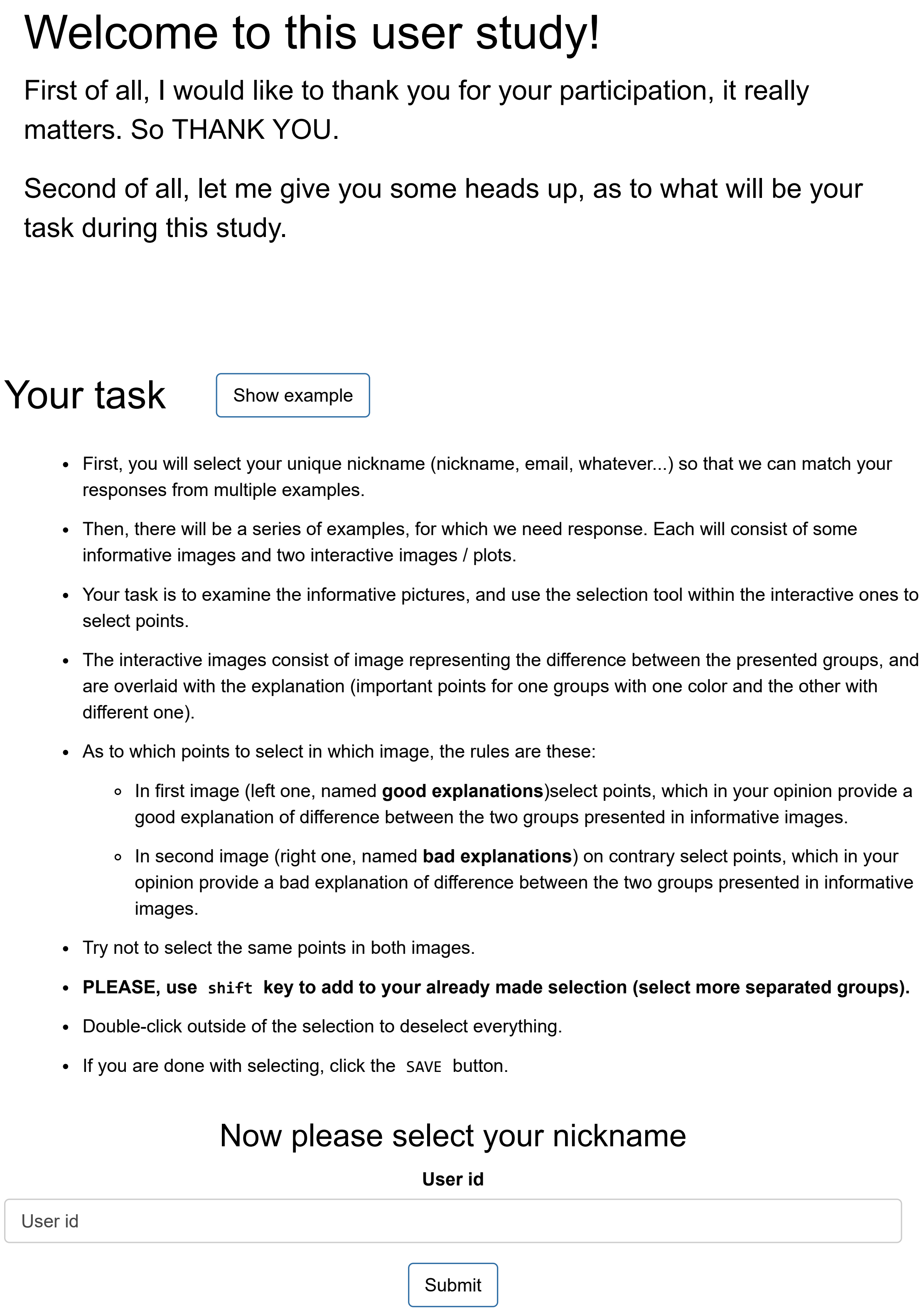
Method alpha_metric

Calculates the metric specifying the goodness of reduction based on the baseline scores, original attribute set size, reduction score and reduction attribute set size. This metric uses weights to add importance to either score on attribute set sizes.

B.1.2 User study

Our user study was conducted through a web application we crated for this. This app is present on the electronic medium, and here, we will showcase the web pages present in the study. On the first page, participant read his task and could open an example of the task. He also selected his nickname for the user study. On the following pages he completed the same task for three examples. Afterwards, he could leave us a note about the user study. The example the user could view can be seen in figure B.2.

Following pages contain the converted pages from user study web application.



Welcome to this user study!

First of all, I would like to thank you for your participation, it really matters. So THANK YOU.

Second of all, let me give you some heads up, as to what will be your task during this study.

Your task

- First, you will select your unique nickname (nickname, email, whatever...) so that we can match your responses from multiple examples.
- Then, there will be a series of examples, for which we need response. Each will consist of some informative images and two interactive images / plots.
- Your task is to examine the informative pictures, and use the selection tool within the interactive ones to select points.
- The interactive images consist of image representing the difference between the presented groups, and are overlaid with the explanation (important points for one groups with one color and the other with different one).
- As to which points to select in which image, the rules are these:
  - In first image (left one, named **good explanations**)select points, which in your opinion provide a good explanation of difference between the two groups presented in informative images.
  - In second image (right one, named **bad explanations**) on contrary select points, which in your opinion provide a bad explanation of difference between the two groups presented in informative images.
- Try not to select the same points in both images.
- **PLEASE, use `shift` key to add to your already made selection (select more separated groups).**
- Double-click outside of the selection to deselect everything.
- If you are done with selecting, click the `SAVE` button.

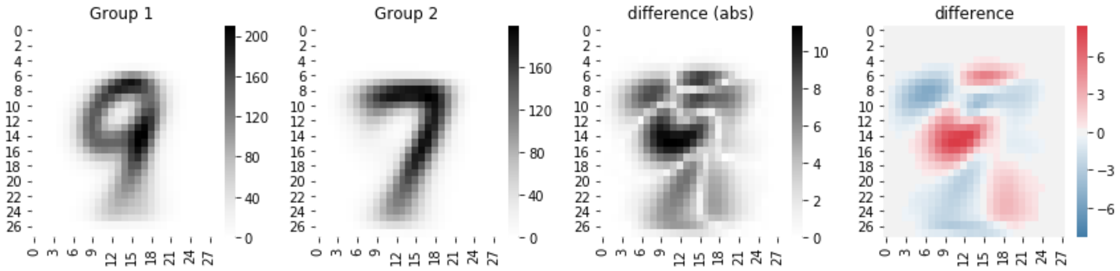
Now please select your nickname

User id

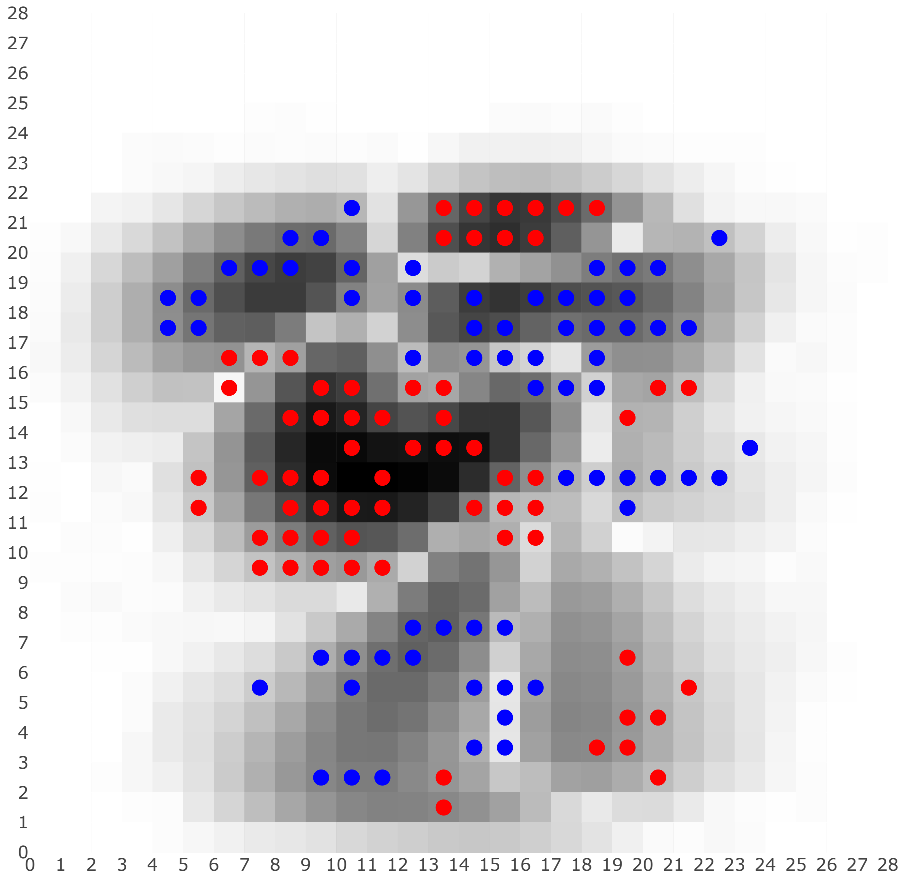
1/3

Your task Show / Hide

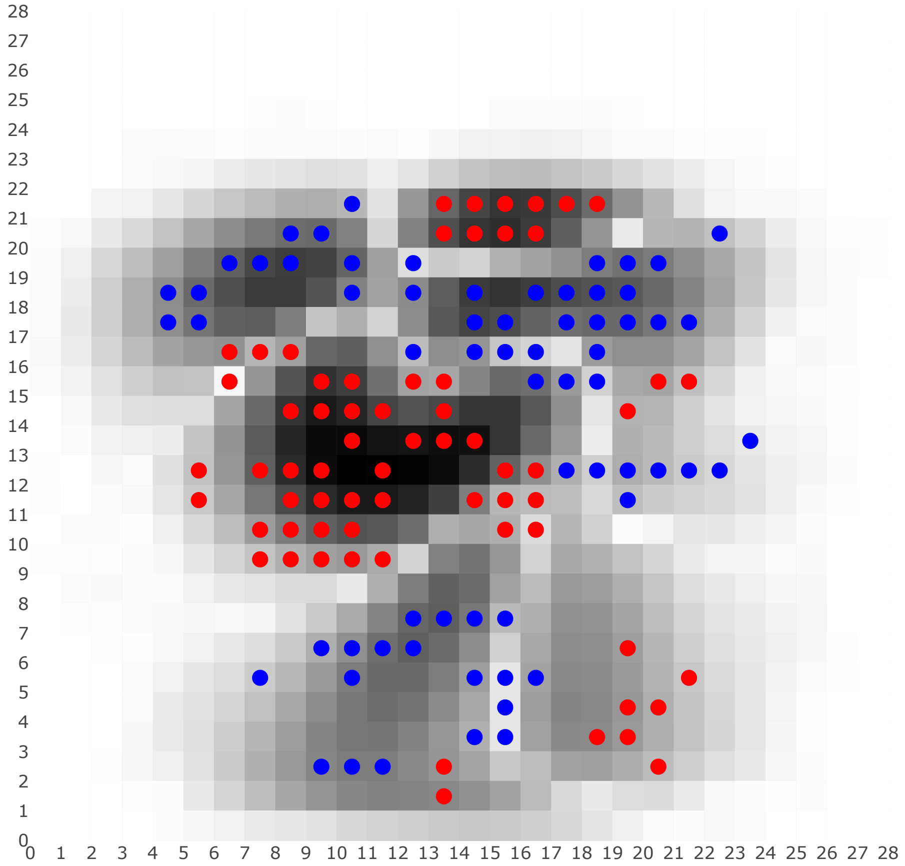
Show example



Good explanations



Bad explanations



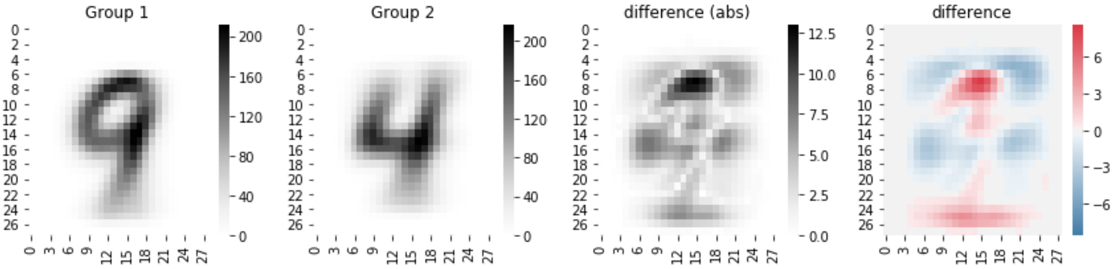
Save

- Data from previous example saved successfully!

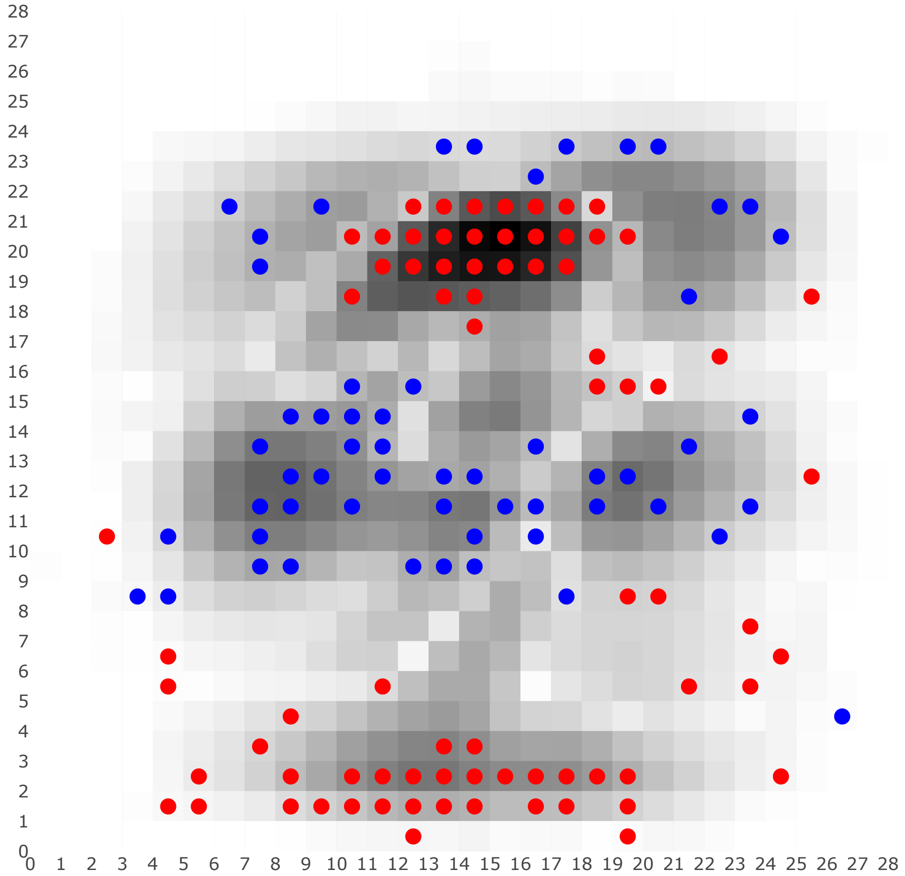
2/3

Your task Show / Hide

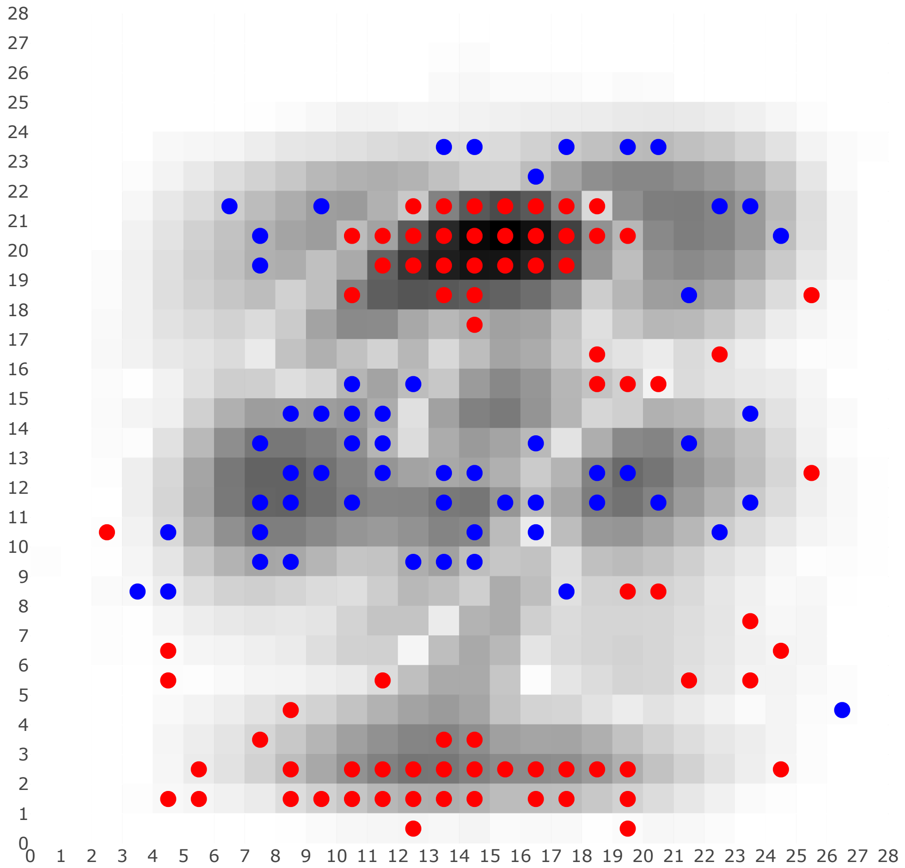
Show example



Good explanations



Bad explanations



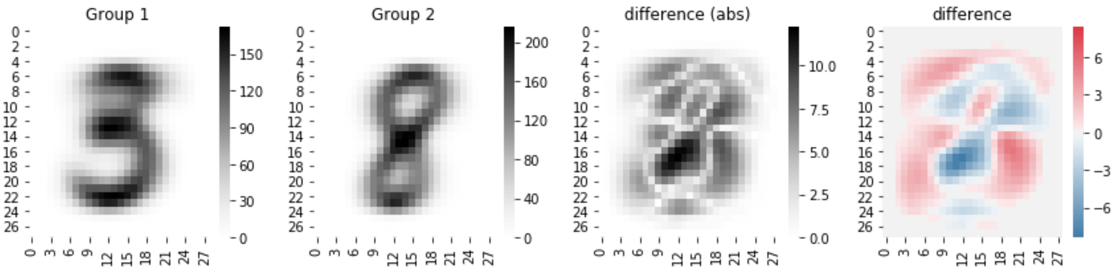
Save

- Data from previous example saved successfully!

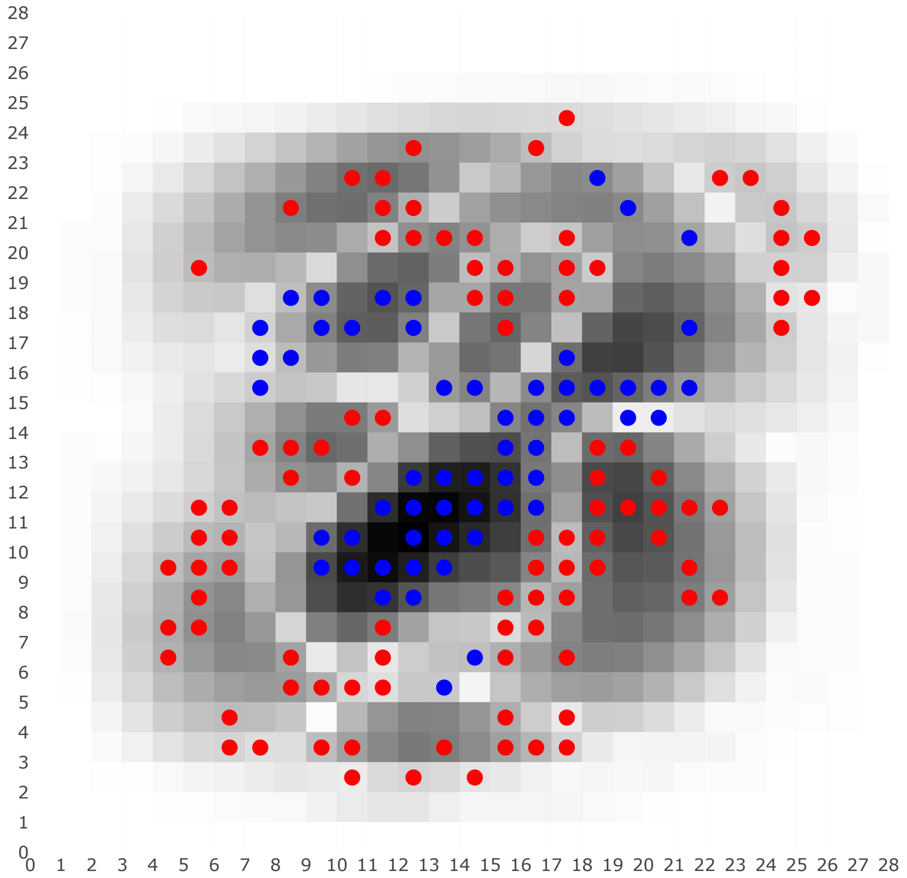
3/3

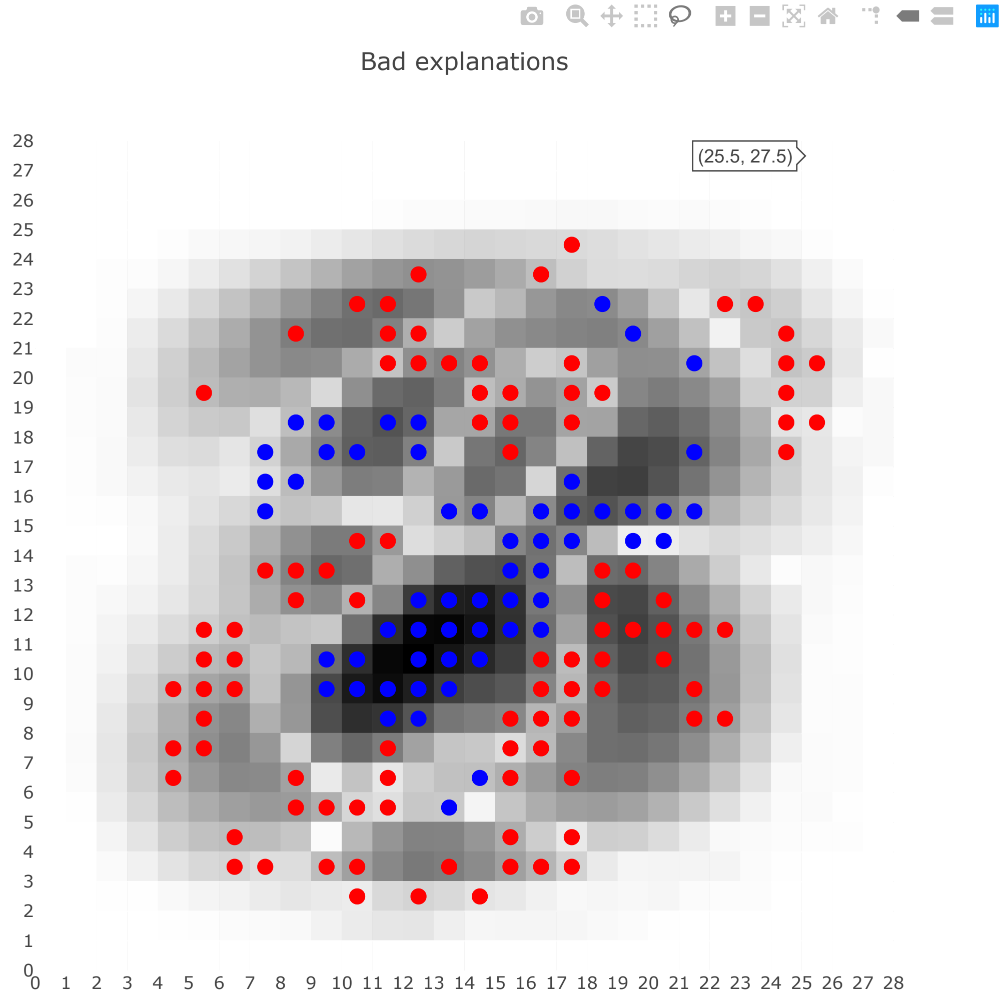
Your task Show / Hide

Show example



Good explanations




Bad explanations

Save

And that's all!

Once again, thank you for your participation in this user study, we really appreciate your effort and time you have given us. So THANK YOU.

If you want to leave us a note, you can do so here. Otherwise just close the browser tab / window.

Text

Text

Submit

B.2 Instalation

To install all the required python libraries, one has to have Python 3.6 or higher installed first (we were using version 3.7). Afterwards, simply run :

- *pip install -r requirements.txt*

in directory with provided requirements.txt file. Afterwards there is one library that needs to be installed from the file provided in installation folder. Choose one based on your python version and architecture of your system and run command:

- *pip install* library_version.*whl*

These steps will install all the libraries required for running the programs. The tasks we completed in this work are implemented in Jupyter notebooks. Because of that, one needs to have also Jupyter notebook installed on his machine (Anaconda distribution or other). The main addition method for linear reduction is contained in standalone python script file.

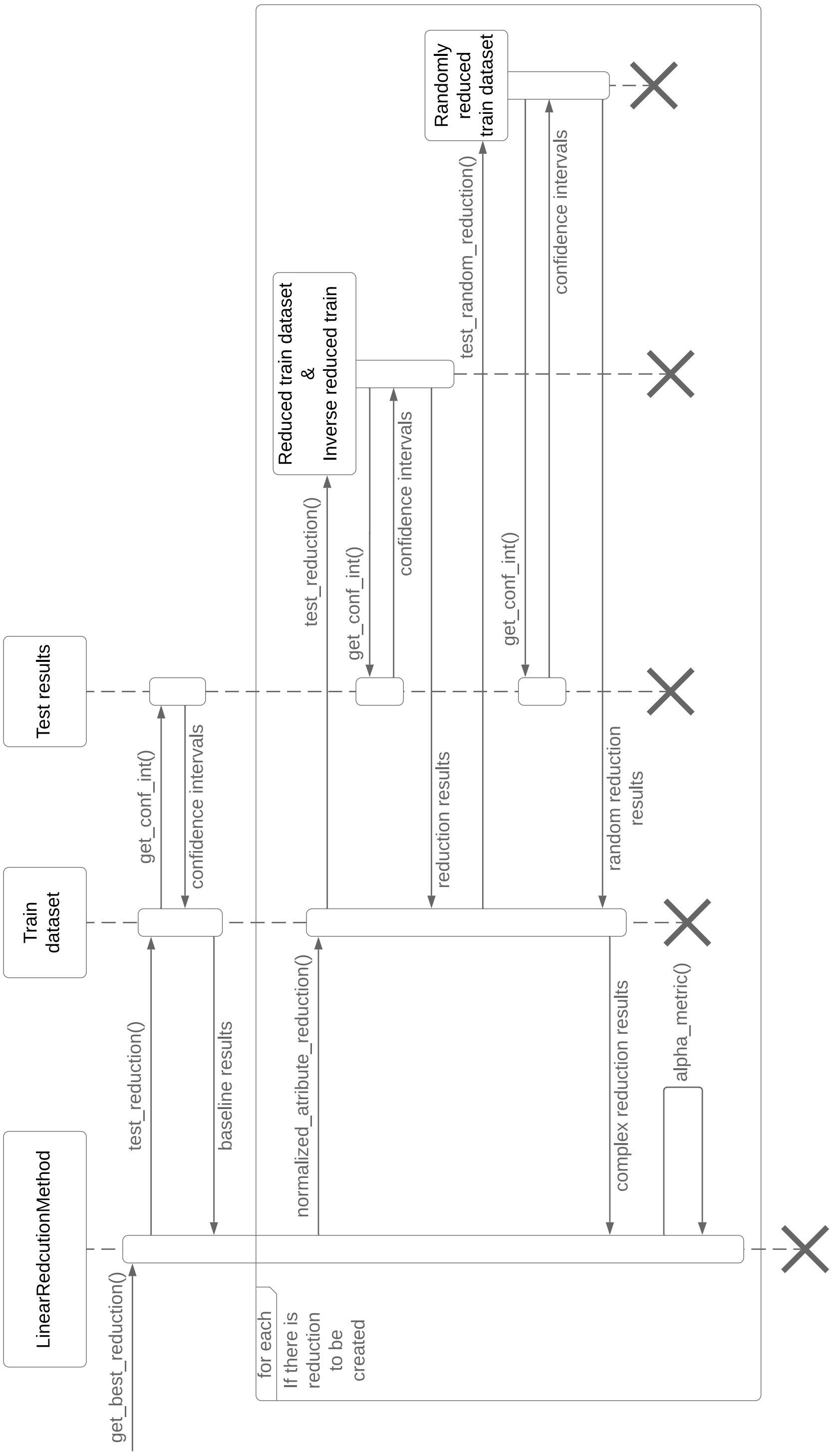


Figure B.1: *Process of reduction creation in form of sequence diagram.*

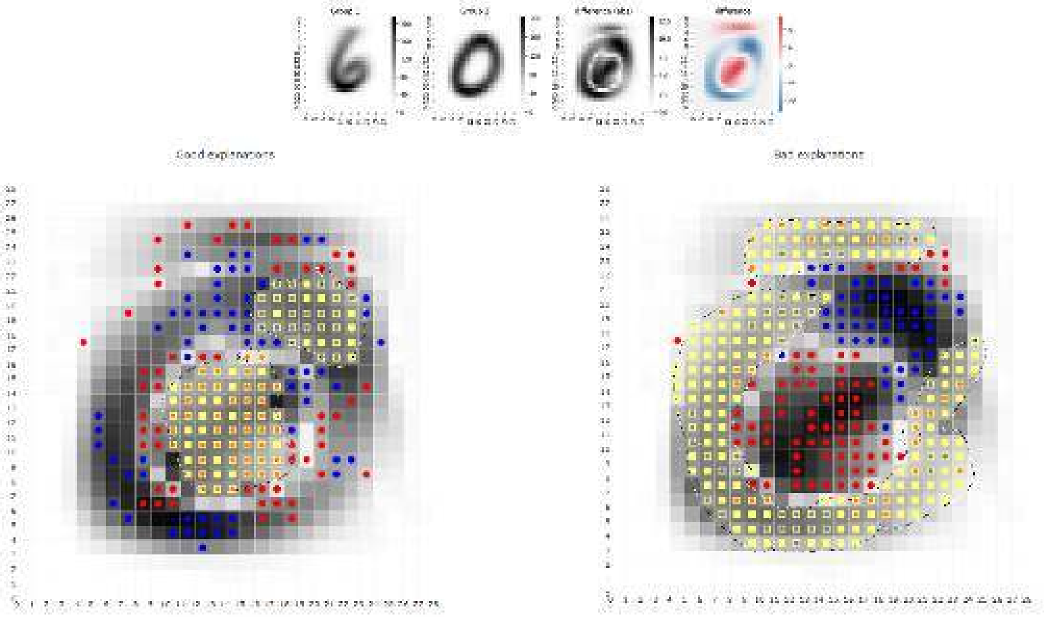


Figure B.2: *Example of completed user study task*

Appendix C

IIT-SRC submission

We processed our work into an article we submitted for the students conference that took place at STU FIIT on April 17. 2019. This article was published in the proceedings of the conference.

Interpretability of machine learning models created by clustering algorithms

Jakub JANEČEK*

Slovak University of Technology in Bratislava
Faculty of Informatics and Information Technologies
Ilkovičova 2, 842 16 Bratislava, Slovakia
xjanecekj@stuba.sk

Abstract. Nowadays, machine learning has become a common tool for solving many research problems, but also problems from real life. With growing algorithm complexity and data dimensionality, the need for model interpretations that are easier to understand is on rise. In this paper, we propose the use of logistic regression as surrogate model for interpreting clustering models. In our work we use topological data analysis (explorative analysis tool) to find interesting segments of data. With the use of logistic regression, we examine the importance of attributes for these segments and draw the best subset of these attributes that can split the two examined segments.

1 Introduction and motivation

In these days machine learning is used in almost all industries. People from these industries, often don't have to have the technical background for complicated algorithms, which are used for machine learning, and in our case specifically segmentation of data. Nevertheless, these people need to understand the models, that the complicated algorithms create as best as possible. Because, if they don't understand them or the reasons behind their decisions, it stirs disbelief towards the results of these models. That's why, it's necessary to design methods for interpretations of these models, which if not completely describe the models and all their aspects, at least enable comprehension of them and their decisions. The comprehension of models decision will bring greater trust in these models, and allow their application for real world problems.

Imagine we created the best possible model for determination of inherited diseases for newborns, which complexity is enormous, and we have no means of explanation for our model, by which we could clarify its decision to doctors. If it was so, the doctors wouldn't be able to use this model. If we can't say why the results are what they're, doctors can't trust it with health or even life of other people.

This problem is present in a lot of machine learning techniques. We decided to focus on data segmentation, especially on explaining differences between two data segments by attributes. In other words, finding a subset of attributes that can split the two segments sufficiently and describing how each of these selected attributes contribute to the division.

2 Analysis of existing solutions

From the existing approaches and works we encountered, the two most similar to ours, in terms of their goals and approach to the problem are LIME technique [5] and the other is explanation using decision trees [4].

2.1 LIME technique

LIME technique [5] is particularly close to the problem, for which we're trying to come up with a solution. This technique is meant for explaining black box models for classification, by finding features that best identify the examined observation, with use of surrogate model. It perturbs the input features of observation and looks for the best match of these perturbed inputs' results and results of the original input. It does an excellent job explaining why output for an observation is what it is. In their work, authors also proposed a technique for explaining the whole model, by drawing variable number of observations which best showcase and explain unique features, and afterwards finds the most important for the model as a whole. But this approach is based on single observations and is mainly intended for that. Explaining why the observation

[*] Master study programme in field: Intelligent software systems
Supervisor: Dr. Jakub Ševcech, Institute of Informatics, Information Systems and Software Engineering, Faculty of Informatics and Information Technologies STU in Bratislava

can be labeled as belonging to some class. Interesting part is also using linear models as a tool to locally approximate models. Linear models can't globally approximate complex black box models, but can provide an approximation of local part of that complex model, which is not a faithful interpretation of the original black box.

2.2 Explanation using decision trees

In [4], decision trees were used to visualize what separates the clusters. Although elegant and easy to understand, this method becomes less feasible with growing dimensionality of the datasets. In high dimensional data, the trees are too complicated for humans to easily understand what separates two data segments. But for data with low dimensionality, this technique is quite good and can be applied. Also a slight problem might occur, if we try to prune the tree to make them smaller, because of the way we would choose the subset of attributes. If we would base this selection only on attribute importance, we could miss some key interactions between attributes that together form a difference between segments.

3 Analysis of algorithms and techniques that could be used for our method

In order for us to be able to design a new way of interpreting clustering models, we needed to analyze the algorithms that could be used for this task. based on this analysis we could choose the best approach.

3.1 Dimensionality reduction techniques

It's possible to interpret the results of clustering by a scatter plot. However, if scatter plot is used, visualization beyond 2 attributes is very problematic. But if there are more attributes that contribute to the segmentation, they have to be reduced. Using methods such as PCA or t-SNE algorithm. [2] [1]. These are often used to help the visualization of the data. But this approach has its obvious downside. The cost of these dimensionality reduction techniques is losing the connections of original features to decisions of segmentation. That's why this approach is not sufficient for our goal.

3.2 Feature selection techniques based on feature importance

Another approach could be based on techniques that evaluate the importance of individual features. Afterwards we would simply choose top N features to provide the interpretation of the model. We would need to find a way to selecting enough attributes to keep the information of the model close to the original model. That would be possible. But more importantly, there's one issue we wouldn't be able to solve for some algorithms. That's that some algorithms that provide the importance of features in model, taking each feature as a single unit, don't take into account the interactions between features. But sometimes, it's these interactions that are key to separation of the segments in data. Similar to these algorithmical techniques is the statistical method of Kolmogorov-Smirnov two sample test. It's a nonparametric test, that provides the means to evaluating the difference between one-dimensional probability distributions of two randomly chosen samples of data. It quantifies the distance between empirical distribution functions of these two samples. We can use it to measure the distance between the distributions of each attribute for two chosen groups of data, and find the attributes that differ the most [6].

3.3 Linear models with L1 and L2 regularization

Inspired by the LIME technique [5], we took a look at the linear models and application of regularization to them.

Regularization of linear models is a set of techniques, which main goal is to provide means of adjusting the learning process of algorithm to get better results, achieving them by improving generalization of model. In linear model we're trying to find coefficients for each attribute so that the function composed from these attributes best approximates. By different means for each technique, it tries to regulate these coefficients, so that some regions of our function space are penalized and thus being able to generalize better. [3] There are two basic forms of regularization. L1 regularization also called Lasso and L2 regularization also called Ridge. The main difference is that L1 uses just the sum of the weights while L2 uses the sum of the square of the coefficients to regulate the coefficients.

The main difference between L1 and L2 regularization is that with L2 the weight for every coefficient is always bigger than 0. Which means that the solution is never a sparse vector. On contrary the with L1 regularization can be sparse, meaning that some features will be marked as non essential for solution and their weights will be zero. That's what we want. To get a sparse vector of coefficient for attributes, which will tell us what attributes we need for learning a model able to distinguish two segments.

4 Proposal of method for improving interpretation of clustering models

Based on the analysis, we decided to formulate our method around regularization of linear models, more specifically, the L1 form of it. The input to this method is the result of clustering, or any other marked data with two groups (segments). First a baseline surrogate linear classification model is created with no regularization applied. Afterwards the same type of model - this time regularized - is created, to provide a interpretation of former clustering model based on feature importance for each group. The L1 regularization will provide us with a sparse vector, from which we will derive the interpretation. But for the linear model to be able to provide us with this sparse solution, we need to specify the extent of the regularization - sparseness of the vector. This measure is key to the degree of reduction that's done. It's a single input into the regularization of linear model. Moreover, the reduction that we create, needs to hold enough information, that it can sufficiently distinguish the two segments of the data. These two aspects, pushed us to define a equation for reduction score, which would be able to evaluate the created reduction and find the best one possible.

4.1 Reduction score equation

In most cases of linear classification, when the model with L1 regularization is used, it's a common approach to find the best coefficient for regularization (let's call it *alpha* coefficient) with the hyperparameter tuning and pick the model that gives the best results. However, this way we don't take into account the amount of attributes, that we marked as not essential, thus making the model smaller (more interpretable). That's why our equation consist of two parts. One makes account for the goodness of results of linear model, and the other for the amount of not essential attributes found. We designed this equation to fulfill some basics, that would help us interpret its output. These are:

– The output for baseline linear model is 1.
– If reduction is worse than baseline, the output is less than 1.
– If reduction is better than baseline, the output is more than 1.

With these in mind, we formulated the equation:

$$Score = (1 + \frac{n_1 - n_2}{n_1} * w_1) * (1 - \frac{r_1 - r_2}{r_1} * w_2) \quad (1)$$

To break it down, the left bracket represent the factor for number of attributes and right bracket represents the factor of evaluation classifier result on test data.

The attribute factor consist of 3 variables. Variable n_1 stands for original number of attributes in data, the n_2 for selected number of attributes in reduction. The fraction in this part gives numbers from 0 to 1. In addition it can be weighted by w_1. The result of this bracket is number from 1 to 2.

In the evaluation result factor, we also have 3 variables. Variable r_1 stands for evaluation score of linear model with original test data, r_2 stands for evaluation score of linear model with reduced test data (reduced attributes) and w_2 is used as a weight. The characteristics of this factor are similar to the attribute factor, yet not the same. Its output fits to range ¡0, 2).

For both weights, there's a constraint, that they have to be numbers in range 0 to 1.

With this equation, we can create multiple reductions and find the best based on comparison of a single number - the score from this equation.

5 Proposed method evaluation

We decided to evaluate our method in two ways.

The first one being the statistical evaluation, in which we carry out an experiment with MNIST dataset (image dataset of written numbers from 0 to 9). In this experiment we will compare the selected essential attributes with our method for interpreting the difference between two segments with Random forest algorithm and Kolmogorov-Smirnov score, which both can provide feature importance. With this experiment, our goal is to find out, if truly, the linear model with L1 regularization can find the important attributes for dividing these two segments of data.

The second is a study with people, by which we want to prove the hypothesis, that our method can boost the interpretability of clustering models. More concretely, if our method can indeed identify the features in data, that are key for people to distinguish two segments of data apart.

5.1 MNIST dataset - statistical evaluation

In first part of evaluation of our method, we decided to select two segments from MNIST dataset. These segments were identified with the use of Topological data analysis [7]. We split the data into two parts:

– train data - used to create reduction (40%)
– test data - used to evaluate the reduction (60%).

Afterwards, we created multiple reductions, which results can be seen in table 1. For the evaluation of reductions, we're using a baseline model (logistic regression with L2 regualarization with coefficient 10000 - no regularization). The baselines result with f1 metric and no reduction of attributes (784) was 94.43%. These numbers serve as input to our evaluation equation. The results on test data were obtained with cross-validation, which in turn gave us the option to calculate the confidence intervals (CI). The CI values don't spike into very large numbers, and that's good, meaning the results of these reductions aren't just a coincidence. For weights in our designed evaluation equation, we assigned weight 0.25 to attribute reduction factor, and weight 1 to score difference factor.

Table 1. Reductions attributes and results

alpha	n. attr.	red. eq. score	score (f1)
5	425	1.1102	94.04%
3	403	1.1151	93.85%
1	364	1.1260	93.73%
0.5	339	1.1303	93.41%
0.1	330	1.1351	93.58%
0.05	307	1.1366	93.08%
0.01	222	1.1873	95.11%
0.005	185	1.2033	95.46%
0.001	108	1.2266	95.33%

Random forest and Kolmogorov-Smirnov score provide the ranking of feature importance, however, they don't specify the number of attributes they deem important. In addition, they only have positive weights for attributes, so we don't know, for which class which attribute is important. That's why we decided to simply compare the presence of attributes in both groups (attributes from reduction vs. attributes from control algorithm). We take the same amount from top of the rankings of control algorithms as our method based on logistic regression selects. Afterwards, we compare these two groups and evaluate how many attributes are present in both groups. Based on the fact that the control algorithm and our method are based on different principles, we take it as a success if these two groups share at least half the attributes. In table 2 we can see the results. The second and third column represent the number of attributes the reduction of size given in first column has the same as the subset from control algorithm.

Table 2. Comparison of reduction feature subsets with control algorithms.

size of subset	Random Forest	Kol.-Smirnov
425	399 (93.88%)	388 (91.29%)
403	373 (92.56%)	363 (90.07%)
364	321 (88.19%)	313 (85.99%)
339	290 (85.55%)	281 (82.89%)
330	281 (85.15%)	270 (81.81%)
307	245 (79.80%)	238 (77.52%)
222	137 (61.71%)	134 (60.36%)
185	105 (56.76%)	101 (54.59%)
108	56 (51.85%)	63 (58.33%)

5.2 User study

We designed the study this way. A person will be presented with pictures representing the mean values of two segments (groups) of observations from MNIST dataset, and also their difference. He will examine these images. Afterwards, he will be presented with the visualization of the difference between these two segments, overlaid with some explanation of the difference between these two groups provided by our method based on feature importance. In this visualization, he will need to identify those segments (pixels) that represent the good explanations of the difference, and those that represent the bad ones. Both pictures will represent the mean values of pixels for observations from these groups.

The evaluation of the study consist from comparing the regions (pixels) in pictures identified by people with the ones our proposed method is able to identify.

In the time of writing of this article, the user study is just starting.

6 Conclusions

In our work, we were able to construct a method that utilizes linear model with L1 regularization to identify subsets of attributes that can divide two segments of data apart. This method is based on surrogate linear classification model, and can be used on any result of clustering or other data to find the differences between groups. Based on the results that we have so far, we can conclude that we succeeded in our task to improve the intepretability of clustering model, by being able to interpret the difference between two segments of data.

References

[1] Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE. *Journal of machine learning research*, 2008, vol. 9, no. Nov, pp. 2579–2605.

[2] Maćkiewicz, A., Ratajczak, W.: Principal components analysis (PCA). *Computers & Geosciences*, 1993, vol. 19, no. 3, pp. 303 – 342.

[3] Neumaier, A.: Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM review*, 1998, vol. 40, no. 3, pp. 636–666.

[4] Parisot, O., Ghoniem, M., Otjacques, B.: Decision Trees and Data Preprocessing to Help Clustering Interpretation. In: *Proceedings of 3rd International Conference on Data Management Technologies and Applications*. DATA 2014, Portugal, SCITEPRESS - Science and Technology Publications, Lda, 2014, pp. 48–55.

[5] Ribeiro, M.T., Singh, S., Guestrin, C.: Why should i trust you?: Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2016, pp. 1135–1144.

[6] Rucco, M., Falsetti, L., Herman, D., Petrossian, T., Merelli, E., Nitti, C., Salvi, A.: Using topological data analysis for diagnosis pulmonary embolism. *arXiv preprint arXiv:1409.5020*, 2014.

[7] Zomorodian, A.: Topological data analysis. *Advances in applied and computational topology*, 2012, vol. 70, pp. 1–39.

Appendix D

Electronic medium

Evidence number of this work in information system: FIIT-182905-73940
The content of electronic medium (zip archive) provided with this work consists of following directories and files:

/Final_code

/datasets

- datasets used in tests of proposed method

/methods

- method implementation
- statistical evaluation of method
- user study evaluation
- mapper usage

/mapper_results

- results of mapper usage

/results

- results of the work

/installation

- requirements.txt - list of libraries needed for this work
- library versions for one that needs to be installed from file

readme.txt - description of media content

/Final_text

- pdf file of the text of this work

- latex source files for documentation