DEPARTMENTS OF MATHEMATICS FACULTY OF EXACT SCIENCES

MSc thesis



**Combined Feature Selection for Microbiome Analysis using Machine Learning methods**

בחירת מאפיינים משולבת עבור ניתוח מיקרוביוטה באמצעות שיטות בלמידת מכוונה

Submitted by:

Anna Belogolovski,

326813060

Supervisor: Prof. Yoram Louzoun

August 2019

Contents

[Introduction 3](#_Toc17492048)

[Machine Learning 3](#_Toc17492049)

[Loss Function 3](#_Toc17492050)

[Deep Learning 4](#_Toc17492051)

[Microbiome 5](#_Toc17492052)

[Taxonomy 6](#_Toc17492053)

[OTU 6](#_Toc17492054)

[UniFrac 7](#_Toc17492055)

[Dimensionality reduction 7](#_Toc17492056)

[Feature projection 7](#_Toc17492057)

[Feature selection 8](#_Toc17492058)

[Feature importance 8](#_Toc17492062)

[(https://medium.com/bigdatarepublic/feature-importance-whats-in-a-name-79532e59eea3) 8](#_Toc17492063)

[Current Solutions 9](#_Toc17492064)

[This thesis 9](#_Toc17492065)

[Methods 10](#_Toc17492066)

[Datasets 11](#_Toc17492067)

[Mucositis 11](#_Toc17492068)

[Allergy 11](#_Toc17492069)

[Preprocess 12](#_Toc17492070)

[Outlier removal 13](#_Toc17492071)

[Similarity algorithm 14](#_Toc17492072)

[improvements 15](#_Toc17492073)

[Neural Networks 16](#_Toc17492074)

[Fully Connected 17](#_Toc17492075)

[LSTM 18](#_Toc17492076)

[Combined 19](#_Toc17492077)

[Grid Search 19](#_Toc17492078)

[Similarity 19](#_Toc17492079)

[Fully Connected 20](#_Toc17492080)

[LSTM 21](#_Toc17492081)

[Bibliography 21](#_Toc17492082)

# Introduction

## Machine Learning

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

The process of learning begins with observations or data, usually referred as the "Training data". This data is used to look for patterns and make better decisions in the future. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly.

The goal of every ML is to find a function between the input and output, in order to achieve this goal the algorithm is trying to minimize the empirical error that is defined as follows:

The loss can be defined in many ways and is usually specific for the type of problem that the algorithm is trying to solve (classification, regression, etc.)

### Loss Function

Machine learning algorithms “achieve the learning” by using a loss function. It’s a method of evaluating how well the specific algorithm model deals with the given data. If predictions deviate too much from actual results, the loss function will produce a very large number, if the prediction is close, it will produce a relatively small number. Gradually, with the help of some optimization function, the model will try to lower the loss, and by doing that it will also reduce the error in prediction.

One would think that the natural loss function to use would be zero-one loss, which means it gets a 1 when the prediction is the same as the true value and 0 otherwise. It turns out that the zero-one loss function is not only non-convex but also non-differentiable at 0, where there is a steep ascent from a loss value of 0 to 1. Even if we were to define a derivative at this point to make the function everywhere differentiable, the function remains non-convex and difficult to optimize. This is why we would rarely use this function, instead we will use the surrogate loss which achieves our goal and is much easier to optimize. In figure 3 one can see different loss functions,   
  
one should notice that all the functions can serve as an upper bound for the zero-one loss.

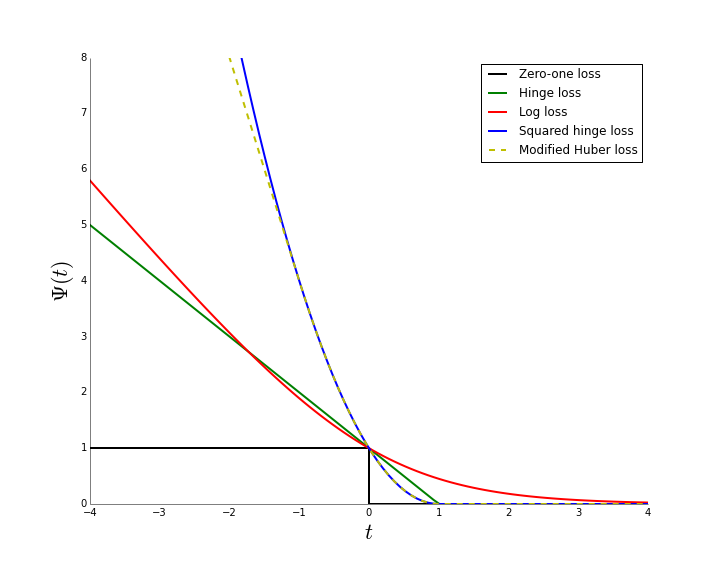


Figure 3 – Different loss functions 24

Broadly, loss functions can be classified into two major categories depending upon the type of learning task we are dealing with — **Regression losses** and **Classification losses**. In this thesis we are dealing with a regression problem and throughout the thesis it will use the Mean Squared Error (MSE) with some changes that will be later discussed.

Assuming the input is and the true output (can be a vector) is and the prediction value is then the MSE is defined as follows:

### Deep Learning

Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large amounts of data. These algorithms can learn by experience and acquire skills without human involvement. These networks are inspired by the human brain, hence the name "Neural Networks".

In this thesis we will focus on feedforward fully connected neural network (FNN) .

#### FNN

The feedforward neural network was the first and simplest type of artificial neural network devised. It contains multiple neurons (nodes) arranged in layers. Nodes from adjacent layers have connections or edges between them. All these connections have weights associated with them.

An example of a feedforward neural network is shown in Figure 4.

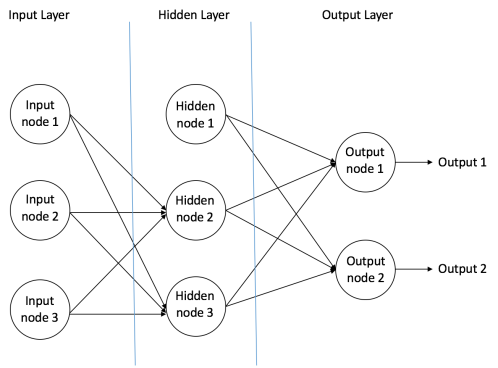


Figure 4 – an example of feedforward neural network 1

A feedforward neural network can consist of three types of nodes:

1. **Input Nodes –** The Input nodes provide information from the outside world to the network and are together referred to as the “Input Layer”. These nodes are connected to the next layers (called hidden layers).
2. **Hidden Nodes –**The Hidden nodes have no direct connection with the outside world (hence the name “hidden”). They perform computations and transfer information from the input nodes to the output nodes. There is no limitation to the number of hidden layers and the number of neurons per layer, nevertheless if one uses a too complex network it can overfit the train data.
3. **Output Nodes –**The Output nodes are collectively referred to as the “Output Layer” and are responsible for computations and transferring information from the network to the outside world.  
   the output nodes can have multiple outputs and can be used for different tasks such as classification and regression.

In a feedforward network, the information moves in only one direction – forward – from the input nodes, through the hidden nodes (if any) and to the output nodes. There are no cycles or loops in the network 1.

## Microbiome

Microorganisms live around us and inside our body, they are affecting both our environment and our health. It is known that microorganisms are involved in a wide range of ecological functions, including coral health promoting and wastewater treatment 2 3.

A link between microorganisms and illness has been known for years but recent studies reveal that the microbial communities within our bodies are associated with many other syndromes including obesity, celiac, diabetes and even allergy 4.

Recent studies also show that the structure of microbial communities, especially in the gut, are highly dynamic and varies in response to many factors such as body state (pregnancy for example), diet, physical activity, body site and growth 5. The fact that our (and other organisms as well) microbiome is strongly related to our health and physiological state results in many studies in the field, which aim to gain insights from this relationship in order to later use it for both medical and ecological applications.

The human microbiome is composed of communities of bacteria (and viruses and fungi) that have a greater complexity than the human genome itself. Large-scale metagenomic projects (community and environmental genomics), such as the European Metagenomics of the Human Intestinal Tract and the Human Microbiome Project, have reported 3.3 million unique protein-encoding genes as compared with the entire human genome, which has around 23 000 genes. These studies have described the beneficial functions of the normal gut microbiota on health down to the genetic level. The human microbiome has extensive functions such as development of immunity, defense against pathogens, host nutrition including production of short-chain fatty acids important in host energy metabolism, synthesis of vitamins and fat storage as well as an influence on human behavior, making it an essential organ of the body without which we would not function correctly. 6

The Microbiome is dynamic and changes with early development. It changes due to environmental factors such as diet and use of antibiotics and especially in response to disease or a treatment to it (e.g. diabetes, IBD, allergy and more). The most dramatic changes in composition occur in infancy and early childhood. A healthy human gut can house at least 1000 different species of bacteria. Elucidating the microbiome composition in healthy individuals is important to understanding what consequences changes in microbiome composition may have to human health and disease susceptibilities. The relationship between changes in microbiome composition and disease pathogenesis is uncertain. The challenge is to identify whether microbial imbalance is related to disease, and to be able to distinguish between cause and effect.

## Taxonomy

Taxonomy is a hierarchical system for classifying organisms to the species level. The broadest classifications are by domain and kingdom; the most specific classification is by genus and species7. The hierarchical groupings in between include phylum, class, family, and order. It has 8 levels in total.

## OTU

An operational taxonomic unit (OTU) is an operational definition used to classify groups of closely related individuals. The term was originally introduced by [Robert R. Sokal](https://en.wikipedia.org/wiki/Robert_R._Sokal) and [Peter H. A. Sneath](https://en.wikipedia.org/wiki/Peter_H._A._Sneath) in the context of [Numerical taxonomy](https://en.wikipedia.org/wiki/Numerical_taxonomy), where an "Operational Taxonomic Unit" is simply the group of organisms currently being studied2. Nowadays, the term OTU refers to clusters of (uncultivated or unknown) organisms, grouped by DNA sequence similarity of a specific taxonomic marker gene 8. In other words, OTUs are pragmatic proxies for microbial "[species](https://en.wikipedia.org/wiki/Species)" at different taxonomic levels.

In this thesis we will use OTU tables that were produced with the following procedure: DNA is extracted from all fecal samples using the PowerSoil kit (Mo-Bio) according to the manufacturer’s instructions. Purified DNA will be used for PCR amplification of the variable V4 region (using 515F-806R barcoded primers) of the 16S rRNA gene. Amplicons will be purified using AMPure magnetic beads (Beckman Coulter) and subsequently quantified using Picogreen dsDNA quantitation kit (Invitrogen) according to the manufacturer’s instructions. Equimolar amounts of DNA from individual samples will be sequenced using the Illumina MiSeq platform. Microbial communities will be analyzed using QIIME (Quantitative insights into microbial 8 ecology) software, version 1.8.0 Paired–end sequences will be grouped into operational taxonomic units (OTUs) using the Green-Genes databases and sequences with similarity of 97% or greater will be grouped into the same OTU. Chimeric sequences will be removed.

## UniFrac

UniFrac is a distance metric used for comparing biological communities. It differs from dissimilarity measures such as Bray-Curtis dissimilarity in that it incorporates information on the relative relatedness of community members by incorporating phylogenetic distances between observed organisms in the computation9.

## Dimensionality reduction

Dimensionality reduction or dimension reduction is the process of reducing the number of random variables under consideration10 by obtaining a set of principal variables. Approaches can be divided into [feature selection](https://en.wikipedia.org/wiki/Feature_selection) and [feature projection](https://en.wikipedia.org/wiki/Feature_extraction)11.

## Feature projection

Feature projection (also called Feature extraction) transforms the data in the high-dimensional space to a space of fewer dimensions. The data transformation may be linear, as in principal component analysis (PCA), but many nonlinear dimensionality reduction techniques also exist12.

The main methods in microbiome context are PCA(principal component analysis) and PCoA (principal coordinates analysis).

* PCA - The main linear technique for dimensionality reduction, principal component analysis, performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized. In practice, the covariance (and sometimes the correlation) matrix of the data is constructed and the eigenvectors on this matrix are computed. The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. The original space (with dimension of the number of points) has been reduced (with data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors.
* PCoA (also known as Multidimensional scaling (MDS)) - Given a distance matrix with the distances between each pair of objects in a set, and a chosen number of dimensions, N, an MDS algorithm places each object into N-dimensional space such that the between-object distances are preserved as well as possible13.

## Feature selection

Feature selection is the process of selecting a subset of relevant features (variables, predictors) for use in model construction. The central premise when using a feature selection technique is that the data contains some features that are either redundant or irrelevant, and can thus be removed without incurring much loss of information. A feature selection algorithm can be seen as the combination of a search technique for proposing new feature subsets, along with an evaluation measure which scores the different feature subsets. There are three general classes of feature selection algorithms: filter methods, wrapper methods and embedded methods14.

## • Filter methods - Filter feature selection methods apply a statistical measure to assign a scoring to each feature. The features are ranked by the score and either selected to be kept or removed from the dataset. The methods are often univariate and consider the feature independently, or with regard to the dependent variable. Common filter methods are mutual information, Pearson correlation, and class distance or the scores of significance tests for each class/feature combinations15 16.

## • Wrapper Methods - Wrapper methods consider the selection of a set of features as a search problem, where different combinations are prepared, evaluated and compared to other combinations. A predictive model is used to evaluate a combination of features and assign a score based on model accuracy. The search process may be methodical such as a best-first search, it may be stochastic such as a random hill-climbing algorithm, or it may use heuristics, like forward and backward passes to add and remove features. An example if a wrapper method is the recursive feature elimination algorithm.

## • Embedded Methods - Embedded methods learn which features best contribute to the accuracy of the model while the model is being created. The most common type of embedded feature selection methods are regularization methods. Regularization methods are also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm (such as a regression algorithm) that bias the model toward lower complexity (fewer coefficients). Examples of regularization algorithms are LASSO or Ridge Regression and any combination of these algorithms with some improvements17.

## Feature importance

## (<https://medium.com/bigdatarepublic/feature-importance-whats-in-a-name-79532e59eea3>)

Feature importance is the process of selecting and ranking the features in our model which contribute to prediction. Several methods exist to get some insight in these black box ML models. There are a few types of feature importance methods. The most popular of them are ensemble tree feature importance, permuted feature importance, LIME and ALE.

* Ensemble tree feature importance is a tree-specific feature importance measure and computes the average reduction in impurity across all trees in the forest due to each feature. That is, features that tend to split nodes closer to the root of a tree will result in a larger importance value.
* Permuted feature importance is a method when after evaluating the model’s performance, you permute the values of a feature of interest and re-evaluate model performance. The observed mean decrease in performance indicates feature importance.
* LIME(Local interpretable model-agnostic explanations) is a technique aiming to explain which features are most important in specific areas of the feature space. The main idea of LIME is to compute a local surrogate model. A surrogate model is an easily interpretable model such as a linear model or a decision tree trained to mimic the behaviour of the more complex model of interest.
* ALE(Accumulated local effects) - describe how features influence the prediction of a machine learning model on average. It shows how the model predictions change in a small ”window" of the feature around some certain grid value v for data instances in that window18

## Current Solutions

Multiple solutions have been proposed for the limitations above in the context of microbiome analysis:

* Preprocessing and dimensionality reduction:
* UniFrac - Principal coordinate analysis (PCoA) based on UniFrac distance matrices.19
* Normalize OTU and PCA –- Principal componenet analysis (PCA) based on normalized(z score) and log transformed OTU.20

• Feature importance - this issue has no appropriate solution at this stage. Most of the current studies suggest one of the traditional feature importance method like permuted feature importance and ALE21 or ensemble tree feature importance. The main problem about those techniques is that one can investigate only the importance of a specific feature but not the group of features. Besides that, you can only understand the contribution of each feature but you are not able to correspond it to a specific class. Some also try to explain the complicated model by replacing it with a simple one (like in case of LIME feature importance). But the replacing is problematic since the answers we get are not related to our initial model which also gives much better performance.

## This thesis

In this thesis a solution to the mentioned needs will be presented. The solution will consist of two parts:

1. An algorithm for preprocessing and dimensional reduction pipeline for microbiome datasets which increases the model performance.

2. An algorithm for selection of a branch of features associated with each condition based on the initial model.

In the discussion session, possible extensions for the presented methodology will be reviewed as well as possible uses for it.

In the appendix section there are 3 research papers I co-authored during my M.Sc. The papers were published (or accepted) in the following magazines:

* *Neonatal antibiotic exposure impairs child growth during the first six years of life by perturbing intestinal microbial colonization.* Authors: Atara Uzan-Yulzari, Olli Turta, Anna Belogolovski, Oren Ziv, Yoram Louzoun, Samuli Rautava, Omry Koren. Nature Medicine (i.f. 32). – accepted.
* *Modulation of cytokine patterns and microbiome during pregnancy in IBD*. Authors: Janine van der Giessen, Dana Binyamin, Anna Belogolovski, Sigal Frishman, Yoram Louzoun, Maikel Petrus Peppelenbosch, Christien Janneke van der Woude, Omry Koren, Gwenny Manel Fuhler. 2019, Gut (i.f. 17). - published
* *Progesterone Increases Bifidobacterium Relative Abundance during Late Pregnancy*. Authors: Meital Nuriel-Ohayon, Hadar Neuman, Oren Ziv, Anna Belogolovski, Yoram Louzon, Omry Koren. 2019, CellReports (i.f. 8). – published

# Methods

Our goal is to come up with methods that take the censored events into account, which means that we need to use the information from these events. We do not plan to develop new methods for the prediction itself.

In order to do this, we will show that each of the algorithms that we purpose indeed shows an improvement compared to "naïve" approach.

Our metrics for comparison will be:

1. Spearman - It assesses how well the relationship between two variables can be described using a [monotonic](https://en.wikipedia.org/wiki/Monotonic) function.
2. Pearson - measures the strength of association between two variables and the direction of the relationship. A value of ± 1 indicates a perfect degree of association between the two variables.
3. Mean Square Error (MSE)

In this section we offer three different types of algorithm, we will show that using each one of them separately is in fact improving the results and when we combine them together the improvement is even greater, thus supporting the claim that using the censored subjects helps to improve the metrics above.

As a side note, when optimizing one metric we usually degrade the other metrics, our interest in this thesis is to optimize the Spearman metric, never the less we will show that the other metrics are still quite good.

In order to get a more accurate result we ran each of the algorithms multiple times, this was don’t so that we will have enough statistics to make sure that the best result we received didn’t happen by chance and it is indeed significant.

## Datasets

In order to test the methods, we will use datasets that have an event of interest. For each subject in these datasets we will find if it is a censored sample or not and use it appropriately.

### Mucositis

This dataset contains 759 samples, each sample has a date and the start date of the Mucositis. For our analysis we only need the 1st time of the event, thus taking only the samples before the event, this results in total of 302 samples, 141 which belong to not censored and 161 to censored.  
After removing outliers we end up with the following data:

uncensored  
mean time to event: 8.821428571428571  
squared mean: 77.81760204081633  
samples number: 140

censored  
samples number: 161

Each sample has a microbiome which consists of 4177 bacteria. We use taxonomy level of 5 (default is 7) which reduces the number of bacteria to 703.

We then use PCA in order to reduce the dimension to 20.

### Allergy

This dataset contains 462 samples, each sample has a date and the response of the patient to the treatment, weather he or she became "non-allergic". This dataset contains allergy to Peanut(134 samples), Milk(112), Sesame(80), Walnut(72), Egg(28), Cashew(18), Hazelnut(9) and Non(9). We will be using the Peanut samples.  
The event of interest is the time at which the patient became "non-allergic" and is calculated (only for not censored) for each sample by taking the last sample of each subject and subtracting from it the samples' date.

By taking only the Peanut subjects and taking only the samples that occur before the event we end up with 48 uncensored and 55 censored samples.

After removing outliers we end up with the following data:

Uncensored  
mean time to event: 114.58333333333333  
squared mean: 13129.340277777777  
samples number: 48

Censored  
samples number: 55  
  
Each sample has a microbiome which consists of 6328 bacteria. We use taxonomy level of 5 (default is 7) which reduces the number of bacteria to 127.

We then use PCA in order to reduce the dimension to 20.

## Preprocess

For every one of the next algorithms we use the same preprocessing stage, the preprocess stage is a pipeline that was developed over time in order to deal with microbiome data, as a part of this thesis we packed it in a framework that is easy to use.

The analysis has been performed on OTU tables generated via Qiime (the detailed process is described in the Generating OTU Data section).

A minimal value will be added to each OTU level (0.1) and then 10-basis log will be taken for each value, this is made in order to be able to notice the relative change.  
The next step will be to preform Statistical Whitening (Z-scoring), this is done by removing the average and dividing by the standard deviation of each OTU.  
The process (removal of mean, and division by standard deviation) will be repeated for each OTU and per each sample.

Later, we use PCA (Principle Component Analysis) in order to reduce the dimensionality, we used 20 components, this number can also be optimized, in this thesis we didn’t optimize it.

Figure 4 shows the importance and the effect of our preprocessing, in figure 5 one can see how the accumulative PCA looks like.

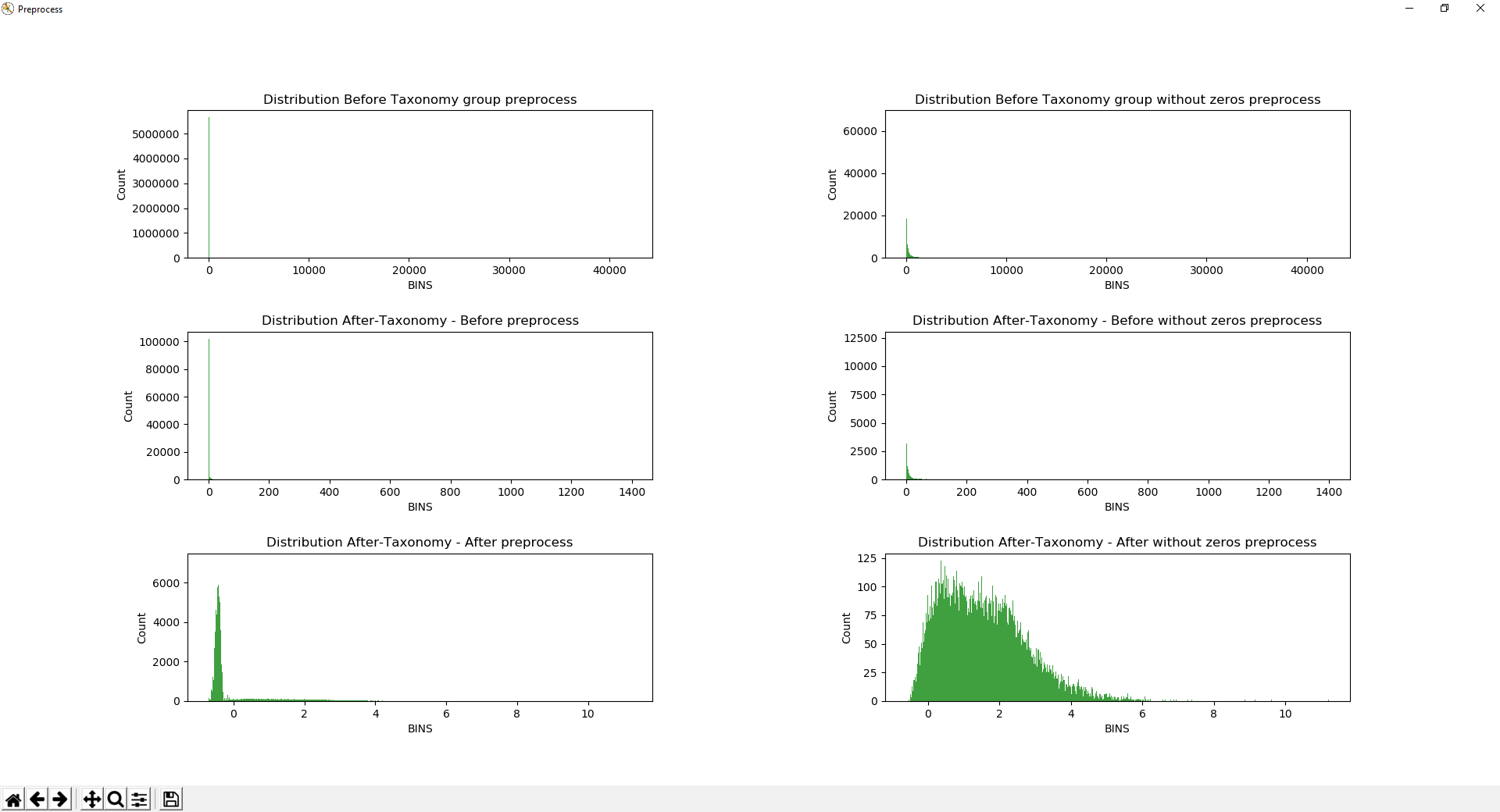


Figure 4 – example of how the preprocess affect the distribution of the data

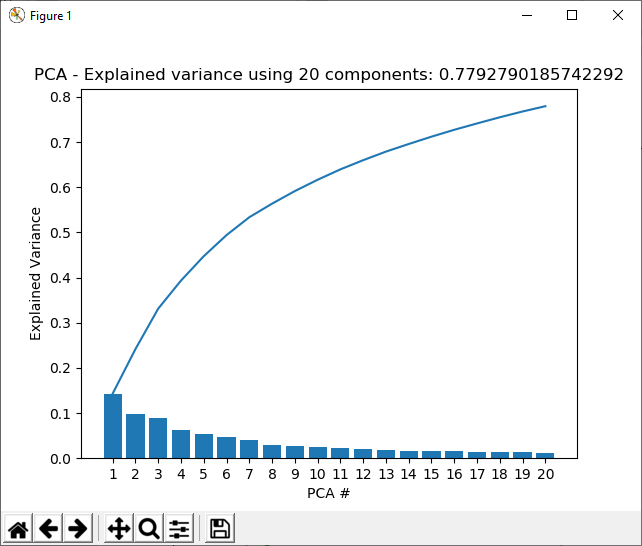


Figure 5 – example of accumulative PCA for 20 components

### Outlier removal

Occasionally our data contains outliers. In this thesis we didn’t develop an advance algorithm to remove these outliers and used a fairly simple one.

We calculate the std of the time for event for the uncensored subjects, and remove samples that are occurring at more than 5\*std.

## Similarity algorithm

The main idea behind this algorithm is to "guess" the event time for the last sample of the censored subject.  
in order to better understand, we will first visualize the main idea and then describe the different steps of the algorithm.

Assume we have the data set which is described in table 1. It has two subjects that have the event and one subject that didn’t have the event, ie censored.

The main idea of the algorithm is to take the last sample of this subject, in the example this would be the sample from the 18.03.18 and see the similarity between this sample to any of the samples of the uncensored subjects. By doing this we will get a similarity matrix that will be denoted as K.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | U1 03.01.18 | U1 05.01.18 | U1 13.01.18 | U1 21.01.18 | U1 22.01.18 | U1 26.01.18 | U2 04.02.18 | U2 05.02.18 | U2 13.03.18 | U2 15.07.18 |
| C1 18.03.18 |  |  |  |  |  |  |  |  |  |  |

For each one of the uncensored samples we know the time for the event, so we can use the coefficients to approximate the event for the last sample of C1.

After the approximation we can than use the "synthetic" event time in our predictor. We used the XGBoost predictor which is based on random forest while using smart boosting technique 22.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Uncensored (name=U1) | 03.01.18 | 05.01.18 | 13.01.18 | 21.01.18 | 22.01.18 | 26.01.18 |
| Uncensored (name=U2) | 04.02.18 | 05.02.18 | 13.03.18 | 15.07.18 |  |  |
| Censored  (name=C1) | 05.03.18 | 10.03.18 | 13.03.18 | 18.03.18 |  |  |

Table 1 – Example for a data set that contains both censored and uncensored data, each cell represents the sample date, the bold cells represents the event.

The detailed algorithm is described next :

* + 1. For each censored subject:

Take the microbiome from the last sample, last sample time will be denoted as N

Compare it to the samples of the uncensored subjects in the following way:

* 1. For each uncensored subject we will take all the samples until the 1st occurrence of the event, this sample of subject will be denoted as
  2. For each sample we have the survival time from this sample, it will be denoted as and lets denote this group as
  3. Next we will define a similarity coefficient between the last sample N of the censored subject m (will be denoted as ) and every sample in the following group (the union of every samples of all subjects that were not censored until the 1st event)
     1. Lets denote this coefficient as where
  4. will be calculated this way:
     1. is the coefficient that controls the weighting, it can be calculated using the std of the distances or be used as a hyper param that needs to be tuned
  5. Next we will calculate the predicted survival time of the censored subject m using the next formula:
     1. The final survival time will be:
        + 1. is the last time that we have knowledge that the subject didn’t have the event (this is a special case when we have the labeling but don’t have the input)
          2. If for any reason the is larger than it means that we will consider it as the censored time
     2. Next we used XGBoost for the prediction (the input are the PCA components and the output is the time for the event)

### improvements

In the process of development, we tried different approaches in order to improve this algorithm

1. Instead of using similarity matrix use the closest neighbor instead of the entire matrix
2. Clustering method
3. Cluster the uncensored data to 2 groups (time based)
4. Convert the clusters of the uncensored to the microbiome space
5. use knn (with k=3) to cluster the censored subjects (using microbiome)
6. run the similarity algo per cluster
7. run xgboost
8. Certainty (it resembles active learning)  
   For each censored mouse we can do the following:
9. Take the microbiome from the last sample, last sample time will be denoted as N
10. Compare it to the samples of the uncensored subjects in the following way:
    1. For each uncensored mouse we will take all the samples until the 1st occurrence of the event (not including the 1st event), this sample of subject will be denoted as
    2. For each sample we have the survival time from this sample, it will be denoted as and , lets denote this group as
    3. Next we will define a similarity coefficient between the last sample N of the censored subject m (will be denoted as ) and every sample in the following group (the union of every samples of all subjects that were not censored until the 1st event)
       1. Lets denote this coefficient as where
    4. will be calculated this way:
    5. Define
       1. for uncensored subject samples = 1
       2. for censored subject Cm and its sample N
    6. Calculate certainty for all uncensored subjects and take the subject which has the highest certainty
    7. Next we will calculate the predicted survival time of the censored subject m using the next formula:
       * 1. We can choose between 2 versions

Will be named ver1



Will be named ver2

This means that the similarity coeff will be factored by certainty

* + - 1. The final survival time will be:

is the last time that we have knowledge that the subject didn’t have the event (this is a special case when we have the labeling but don’t have the input)

If for any reason the is larger than it means that we will consider it as the censored time

* 1. Add the predicted subject sample to the known sample and repeat until no more censored

All the above improvements didn't improve the results, this is due to the fact that they are all based on distance. The Similarity algorithm is already taking all the distance information that is available and weights it appropriately. Due to that from here and on we used only the Similarity algorithm without the improvements.

## Neural Networks

### Fully Connected

We now take another approach, so far, we didn’t use the connection between samples, we only tried to use the last censored sample (different variation as stated above).

Now we want to introduce a new method that is using the fact that we have a time series, we will use this fact when taking account the output of the network.  
We will again use Table 1 (only U1 and C1 for simplicity) to illustrate the main idea behind the algorithm.

the main idea of the new method is to insert the fact that for censored subjects the predicted time of the event at a certain time point can't be prior to the last sample, e.g. if the prediction for the censored subject above at the 05.03.18 is 5 it means that the predictor thinks that this subject will have the event on 10.03.18 which clearly doesn’t make any sense due to the fact it has another sample that occurs later on.

In order to convert the above into formulas we transfer the data to a new representation.

1. For each sample calculate the time of the event relative to the earliest event
2. For each sample calculate the delta to the latest event (for uncensored it is the event date and for censored is the latest sample)
3. For each sample we will have an indicator which states if it is censored or uncensored

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Start Date | Delta | Censored/Uncensored |
| U1 | 0 | 23 | Uncensored |
| U1 | 2 | 21 | Uncensored |
| U1 | 10 | 13 | Uncensored |
| U1 | 18 | 5 | Uncensored |
| U1 | 19 | 4 | Uncensored |
| C1 | 0 | 13 | Censored |
| C1 | 5 | 8 | Censored |
| C1 | 8 | 5 | Censored |
| C1 | 18 | 0 | Censored |

next we will use this table as follows:  
for uncensored subject we will use the MSE loss, it wishes to minimize the prediction for the "time to event" (the prediction is Delta for each row).

For censored subject we can't use MSE because the Delta doesn’t represent the actual time to the event (due to the censoring). It represents the time for the last sample of this subject.  
so here we will want something different:

1. Predict delta
2. Calculate the predicted event using start date + predicted delta
3. Calculate the actual max date using start date + delta
4. Define the Loss as max date – predicted delta

\*We can remove steps b, c and d and replace them with delta-predicted delta (they are left for better understanding)

We will name this loss as TimeSense or TS loss:

Formula 1 – Definition of TimeSense Loss

To better understand the algorithm, we will be using the following row as an example:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Start Date | Delta | Censored/Uncensored |
| C1 | 5 | 8 | Censored |

1. Predict delta – **for the example, assume it predicted it as 6**
2. Calculate the predicted event using start date + predicted delta - **5+6=11**
3. Calculate the actual max date using start date + delta - **5+8=13**
4. Calculate the Loss as max date – predicted delta - **13-11=2**

We see that here we got a positive loss, this means that in the next iteration the want to lower it and produce a prediction that makes sense (with respect to time).

This way we can use the censored and the uncensored in the train stage.

Formularizing the above:

, +,

Formula 2 – Combined loss function, MSE and TS

and are coefficients that can be used in order to weight the impact of our two losses.

### LSTM

Until now we managed to use the censored subjects but didn’t use the relationship that the samples have within the same subject. The FNN used only the output of the network, we will use LSTM special architecture in order to consider the connection within the input itself.

In order to use LSTM architecture, we must work in batches, each batch is composed from subjects, each subject has its own sequence (number of samples). The problem we are facing is that it is not guaranteed that every subject will have the same number of samples. To overcome this issue, we will create the batches in the following way:

* 1. Get the batch size
  2. Split the data using the batch size (the last batch might be smaller)
  3. For every batch
     1. Find the longest sequence
     2. Pad the other subjects with zeros to have the same size as the longest sequence

After building the batches as stated we will the same loss function from Formula 1 in order to train the network.

## Combined

Lastly, we offer methods that combine the different approaches from above. This means we end up with the following algorithms:

1. Similarity algorithm and FNN
   1. First apply similarity algorithm
      1. For the last sample of every censored subject we use a special MSE coefficient which is equal to half of the uncensored MSE coefficient.
   2. Train the network on the new data with the new coefficients
2. Similarity algorithm and LSTM
   1. First apply similarity algorithm
      1. For the last sample of every censored subject we use a special MSE coefficient which is equal to half of the uncensored MSE coefficient.
   2. Train the network on the new data with the new coefficients

## Grid Search

Grid search is the process of performing hyper parameter tuning in order to determine the optimal values for a given model. This is significant as the performance of the entire model is based on the hyper parameter values specified 23.

We will use this technique in this thesis for each of the algorithms that we present. This is done in order to compare the best possible results (with respect to the grid we decided to use). In the end we will compare the algorithms using the best hyper params that were found. every algorithm has its own grid.

### Similarity

1. Without censored (it’s the regular xgboost)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| alpha | 0.01, 20, 50, 100 |
| n estimators | 5, 10, 20 |
| min child weight | 0.1, 1, 10, 20 |
| reg lambda | 0, 10, 20 |
| max depth | 20, 50 |

Total number of configurations:

1. With censored (using Similarity+xgboost)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| alpha | 0.01, 20, 50, 100 |
| n estimators | 5, 10, 20 |
| min child weight | 0.1, 1, 10, 20 |
| reg lambda | 0, 10, 20 |
| max depth | 20, 50 |
| beta | 1, 10, 100 |

Total number of configurations:

### Fully Connected

1. Without censored (it’s the regular NN with MSE)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| Epochs | 20 , 80 |
| Dropout | 0, 0.2, 0.6 |
| l2 | 1, 10, 20, 100 |
| Number of layers | 1, 2, 3 |
| Neurons per layer | 20, 50 |

Total number of configurations:

1. With censored (TS+MSE)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| Epochs | 20 , 80 |
| Dropout | 0, 0.2, 0.6 |
| l2 | 1, 10, 20, 100 |
| Number of layers | 1, 2, 3 |
| Neurons per layer | 20, 50 |
| MSE factor list | 0.1, 10, 100, 1000 |

Total number of configurations:

1. With censored (using Similarity +TS+MSE)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| Epochs | 20 , 80 |
| Dropout | 0, 0.2, 0.6 |
| l2 | 1, 10, 20, 100 |
| Number of layers | 1, 2, 3 |
| Neurons per layer | 20, 50 |
| MSE factor list | 0.1, 10, 100, 1000 |
| beta | 1, 10, 100 |

Total number of configurations:

### LSTM

1. Without censored (it’s the regular NN with MSE)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| Epochs | 20 , 100, 1000 |
| Dropout | 0, 0.2, 0.6 |
| l2 | 1, 10, 20, 100 |
| Number of layers | 1, 2, 3 |
| Neurons per layer | 20, 50 |

Total number of configurations:

1. With censored (TS+MSE)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| Epochs | 20 , 100, 1000 |
| Dropout | 0, 0.2, 0.6 |
| l2 | 1, 10, 20, 100 |
| Number of layers | 1, 2, 3 |
| Neurons per layer | 20, 50 |
| MSE factor list | 0.1, 10, 100, 1000 |

Total number of configurations:

1. With censored (using Similarity +TS+MSE)

|  |  |
| --- | --- |
| Parameter Name | List of Values |
| Epochs | 20 , 100, 1000 |
| Dropout | 0, 0.2, 0.6 |
| l2 | 1, 10, 20, 100 |
| Number of layers | 1, 2, 3 |
| Neurons per layer | 20, 50 |
| MSE factor list | 0.1, 10, 100, 1000 |
| beta | 1, 10, 100 |

Total number of configurations:

# Bibliography

1. ujjwalkarn. A Quick Introduction to Neural Networks – the data science blog.  *August 9, 2016* (2016).

2. Long, C. A., Sokal, R. R. & Sneath, P. H. A. *Principles of Numerical Taxonomy*. *Journal of Mammalogy* **46**, (Narnia, 1965).

3. Jiang, L. *et al.* The use of microbial-earthworm ecofilters for wastewater treatment with special attention to influencing factors in performance: A review. *Bioresour. Technol.* **200**, 999–1007 (2016).

4. de Vos, W. M. & de Vos, E. A. Role of the intestinal microbiome in health and disease: from correlation to causation. *Nutr. Rev.* **70**, S45–S56 (2012).

5. Palmer, C., Bik, E. M., DiGiulio, D. B., Relman, D. A. & Brown, P. O. Development of the human infant intestinal microbiota. *PLoS Biol.* **5**, e177 (2007).

6. Amon, P. & Sanderson, I. What is the microbiome? *Arch. Dis. Child. Educ. Pract. Ed.* **102**, 257–260 (2017).

7. John S. Wilkins. What is systematics and what is taxonomy? Available at: https://pandasthumb.org/archives/2011/02/what-is-systema.html. (Accessed: 5th June 2019)

8. Blaxter, M. *et al.* Defining operational taxonomic units using DNA barcode data. *Philos. Trans. R. Soc. B Biol. Sci.* **360**, 1935–1943 (2005).

9. Lozupone, C. A., Hamady, M., Kelley, S. T. & Knight, R. Quantitative and qualitative β diversity measures lead to different insights into factors that structure microbial communities. *Applied and Environmental Microbiology* (2007). doi:10.1128/AEM.01996-06

10. Roweis, S. T. & Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science (80-. ).* (2000). doi:10.1126/science.290.5500.2323

11. Pundil, P. & Novovicova, J. Novel methods for subset selection with respect to problem knowledge. *IEEE Expert* (1998).

12. Ding, C., He, X., Zha, H. & Simon, H. D. Adaptive dimension reduction for clustering high dimensional data. in *Proceedings - IEEE International Conference on Data Mining, ICDM* (2002).

13. O’Connell, A. A., Borg, I. & Groenen, P. Modern Multidimensional Scaling: Theory and Applications. *J. Am. Stat. Assoc.* (1999). doi:10.2307/2669710

14. Guyon, I. & De, A. M. *An Introduction to Variable and Feature Selection André Elisseeff*. *Journal of Machine Learning Research* **3**, (2003).

15. Yang, Y. & Pedersen, J. O. A Comparative Study on Feature Selection in Text Categorization. *undefined* (1997).

16. Forman, G. *An Extensive Empirical Study of Feature Selection Metrics for Text Classification*. *Journal of Machine Learning Research* **3**, (2003).

17. Zare, H., Haffari, G., Gupta, A. & Brinkman, R. R. Scoring relevancy of features based on combinatorial analysis of Lasso with application to lymphoma diagnosis. *BMC Genomics* **14 Suppl 1**, S14 (2013).

18. Apley, D. W. Visualizing the Effects of Predictor Variables in Black Box Supervised Learning Models. (2016).

19. Bhute, S. *et al.* Molecular characterization and meta-analysis of gut microbial communities illustrate enrichment of prevotella and megasphaera in Indian subjects. *Front. Microbiol.* (2016). doi:10.3389/fmicb.2016.00660

20. Nunberg, M. *et al.* Interleukin 1α-Deficient Mice Have an Altered Gut Microbiota Leading to Protection from Dextran Sodium Sulfate-Induced Colitis. *mSystems* (2018). doi:10.1128/msystems.00213-17

21. Galkin, F. *et al.* Human microbiome aging clocks based on deep learning and tandem of permutation feature importance and accumulated local effects. *bioRxiv* (2018). doi:10.1101/507780

22. Chen, T. & Guestrin, C. XGBoost: A Scalable Tree Boosting System. (2016). doi:10.1145/2939672.2939785

23. Krishni Hewa. An introduction to Grid search – Data Driven Investor – Medium.

24. Algorithmia. Introduction to Loss Functions | Algorithmia Blog. *APRIL 30, 2018* (2018).