UNIVERSITY OF AGDER

Determining gastrointestinal tract dysbiosis using machine learning techniques

Sindre Lindahl

Supervisor
Ole-Christoffer Granmo

This master's thesis is carried out as a part of the education at the University of Agder and is therefore approved as a part of this education. However, this does not imply that the University answers for the methods that are used or the conclusions that are drawn.

University of Agder, 2015
Faculty of Engineering and Science
Department of Information and Communication Technology

Abstract

This thesis explores machine learning techniques for the purpose of determining gastrointestinal tract dysbiosis. Dysbiosis is an unbalance of bacteria flora. Stool sample analysis of relevant bacterias can be used in "diagnosis" of this condition. The problem is how to best classify dysbiosis from a healthy balance of bacteria. Pattern recognition methods could be used to create a diagnostic decision support system. The approach includes comparisons between classifiers with the additional use of feature reduction techniques. Experiments show that the accuracy varies significantly depending of which classifier is used. The best classifier for the data set used here was found to be the C4.5 decision tree. Much of the analyzed data is shown to be noisy, confusing and irrelevant to the classifier. Accuracy can be improved by reducing the amount of bacteria species with more than 90%. In addition, results imply that the different microbial stool analysis panels seriously affect accuracy. Which classifier to use and the highly relevant feature subsets found should be helpful for any future work in the field of gut dysbiosis. And the comparisons could be applicable for classification of similar data sets.

Contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background

Data mining has been used for medical diagnosis since the beginning. This is particularly practical for large and complex medical data. The amount of data can be impossible to diggest manually. Instead, machine learning algorithms can discover patterns in data hidden from the human eye. The suggested solution presented here includes applying classifiers to a data set of relevant bacteria species occuring in feces. Measurements of discriminatory effectiveness between healthy and unhealthy patients is . Further steps are also explored to optimize this process. Feature selection methods are used to not only improve accuracy, but also reduce the amount of features, computational time, and noise.

Gastrointestinal tract dysbiosis (referred to later as dysbiosis) is an imbalance of bacterial flora in the digestive system. The gut microbiota is a community of microorganisms that live symbiotic to their host. Some can be mutualistic, existing in a mutually beneficial relationship to the host. Others may be parasitic, degrading the health of its host. Some may be commensal, living in a more neutral state of co-existance without affecting the host. Several hundred different bacteria species live in the gut and fill up to 60% of the dry mass of feces according to [1]. These bacterias can have a range of positive affects on humans, from synthesizing vitamins, blocking space for infectious bacterias and improving the immune system. A healthy status for the digestive system may rely on a certain balance of the gut microbiota. Dysbiosis is the opposite and according to [2]-[11] is associated with various diseases and illnesses like inflammatory bowel disease, chronic fatigue syndrome, obesity, cancer and colitis. Endotoxins like lipopolysaccharide created by some bacterias have negative impact on a wide range of bodily functions.

There are several difficulties when it comes to medical diagnostics using machine learning techniques (I would like to add that the word "diagnosis" is not completely appropriate in this context. Instead, I will use diagnosis in the sense of deciding well from ill).

- real-world data is often noisy
- class labels can be uncertain (for instance, there is no exact definition of dysbiosis)
- small amount of data (expensive and time consuming to collect)

As a result of the lack of exact defintion, some samples may be incorrectly labeled. And laboratory instruments may read more or less incorrect values from samples. Medical data can also hold large amount of features, preventing any manual analysis. In such cases, machine learning and data mining can be the only option.

The data set used here is new and exhibits unknown patterns. There is no single machine learning algorithm that is best in all circumstances according to [12]. And there are a bunch of alternatives to choose from. These range from the simple to highly advanced algorithms. Some are used for decades, some are newer. Some algorithms are improvements of previous ones while others are developed to cover different types of data. For instance, some classifiers may better handle large amount of features and vica versa. There are different "families" of classifiers like artificial neural networks and decision trees. A common approach to find the best classifier for a data set is to make comparisons between a selection. To make a selection, research needs to be done to find viable candidates.

The theme for the topic question lies in the intersection between machine learning and medical diagnostics. As such, the results should be relevant in both fields. As diagnostics, it can be helpful as guidance towards future experiments on the topic of dysbiosis and diagnostics in general. Feature reduction methods reveal important and irrelevant features. These features map directly to bacterias and may give insight into which bacterias affect dysbiosis the most. On the other hand, comparisons of classifiers may be useful for other classification tasks with similar data sets.

The key components of this project are to search for a decent selection of classifiers, compare their results on the data set and optimizing classification by the use of feature reduction methods.

1.2 The dysbiosis data set

A data set has been created by Genetic Analysis (www.genetic-analsys.com). Stool samples were collected in 2013 from 278 individuals across Scandinavia. Both negative and positive samples have been analyzed with regard to 54 preselected bacteria species and/or families. Each bacteria is measured based on how much of it's DNA were found in the feces. Each sample were labeled as positive or negative based on the assumed status of the individual. There are no missing values in the data set, but many are assumed to be noisy.

The analysis was done with laboratory equipment referred to here as "plate". Most samples were split into duplicate parts and analysed on multiple different plates. As is common with noisy real-world data, one sample gets similar but not identical readings if analyzed multiple times on the same plate. But there is a suspicion that plates differ significantly when analyzing the same sample. The samples were collected chronologically based on class with an unfortunate time period between classes. Because of this, each class was analyzed on separate sets of plates. This is a known weakness of the data set because incorrect patterns can occur based on which plates were used instead of which class a sample belongs to.

It is worth mentioning that because the data set is quite fresh and Genetic Analysis has had different priorities, no binary classification experiments have yet been conducted on it. The data set is the property of Genetic Analysis and the selection of the bacterias analysed from the stool samples are part of their trade secret. Because of this the names of bacteria species and families will be anonymized. The stool samples will also be anonymized because the individuals' identities are irrelevant in this work.

Since the samples are collected from Scandinavia, chances are that they are less representative for other countries or regions because of differences in food, bacterias, virues and so forth. All results based on this data set will be overfitted towards Scandinavia population. Results may or may not be representative for other places.

1.3 Research questions

The main title of this work asks the question about how to use machine learning to get best possible classification of dysbiosis. Different classifiers typically have varying results for the same data set. Finding the right tool for the job is therefore essential for the best result. The aim is to find one classifier in a selection of classifiers that out outranks the others. Based on the data set described above, my proposed solution is to use supervised learning with leave-one-out cross-validation. Furthermore the top ranking classifier will then be optimized by feature selection methods. One side quest is added to find out how different laboratory plates used for feature extraction from stool samples affect classification.

The research questions are presented here:

RQ1 To what degree can state-of-art classifiers discriminate between the classes normobiosis (healthy condition) and dysbiosis (unhealthy condition)?

RQ2 To what degree can feature selection methods improve accuracy and point to bacterias relevant to dysbiosis?

RQ3 Classification accuracy depends on individual laboratory analysis equipments used to extract features from samples.

Results should be beneficial for:

- General insight and comparison of machine learning techniques
- Future work on binary classification of dysbiosis in context of diagnostics
- Revealing bacterias that are predictive of the patient's condition
- Distribution of samples across multiple stool analysis panels

1.4 Limitations

Due to a high amount of machine learning algorithms available, the selections proposed here have to be fairly limited in size. The proposed solution presented later is based on a small selection of classifiers which are recommended or have had success in literature. Searching through large numbers of classifiers will probably lead to alot of mediocre results. Instead, my hope is to search among a small but potent collection. Doing multiple sequential experiments can quickly lead to an explosion in dimensionality of results. To avoid discussing large outputs of data, after finding the best classifier for RQ1, the rest will be discarded. Additional experiments for RQ2 and RQ3 will

only use the winning classifier from RQ1.

Because bacterias are anonymous, no discussion or conclusoin will be made regarding specific species or families of bacteria.

1.5 Acknowledgements

I would like to thank my brother Torbjørn Lindahl for access and insight to this data set. And thanks to Genetic Analysis for giving me access to this data set. A special thanks to fellow students John Daniel Evensen, Stian Guttormsen, Jan-Vidar Ølberg and all participants of periodic meetings at UiA for critisism and inspiration. And thanks to supervisor Granmo for valuable feedback.

Chapter 2

State of the art

This chapter explores some of the most relevant work done in the field of classification and diagnosis using machine learning techniques. General comparisons and overlook of alternatives are examined in the first part. Papers which topic includes diagnosis of specific diseases and data sets are discussed in the second part. A summary is presented in the end of this chapter.

2.1 General overview of popular classifiers

Work done by Wu et al. in [13] is highly relevant for choosing a classifier or a selection of classifiers. They have identified 10 data mining algorithms as the most influential. Their selection is not limited to classifiers but also includes association analysis, clustering, link mining and statistical learning. They claim that these are among the most important topics in data mining. The top 10 algorithms selected are C4.5, k-Means, Support Vector Machine (SVM), Apriori, EM, PageRank, AdaBoost, k-Nearest Neighbours (k-nn), Naive Bayes (NB), and CART. Their work is very thurough with nominations and voting from different groups in different steps with requirements of minimum citations for each contestant. Out of the 10 winners, the classifiers relevant here are C4.5, Support Vector Machine, AdaBoost, k-Nearest Neighbours, Naive Bayes and CART. Many of these classifiers are repeatedly used in other work described later in this chapter. The classifiers mentioned here weighs heavily for my selection for the purpose of diagnosis. All of these alternatives have had success in numerous applications.

Another relevant paper is [M] by Andrew P. Bradley. This paper advocates Area Under the Curve for Receiver Operating Characteristics (ROC) as a ranking measure for comparing classifiers. Six datasets have been used to compare results from six different classifiers. The datasets include medical data for cervical cancer analysis, breast cancer diagnostics, post-operative bleeding, diabetes prediction and two independent data sets for heart disease diagnosis. The data sets are summarized in table 1.

Dataset	Classes	Samples	Features (used)
Cervical cell nuclear texture	2	117	54 (6)
Post-operative bleeding	2	134	Over 200 (4)
Breast cancer diagnosis	2	683	9
Diabetes prediction	2	768	8
Heart disease diagnosis, Cleveland	2	297	76 (13)
Heart disease diagnosis, Hungary	2	261	76 (11)

Table 1: Summary of data sets

A cross-section of six popular machine learning techniques were used: Quadratic Discriminant Function with Bayes decision function, k-Nearest Neighbours, C4.5, Multiscale classifier, Perceptron and Multi-layer Perceptron. Evaluation compares accuracy results with ROC. This measure of classification performance is concluded to have desirable properties. The metric is mentioned as one of the best ways to evaluate performance based on a single value. There are few overall differences between learning algorithms, but in general C4.5 and Multiscale classifier performed worse than the rest on the specific datasets. This paper proves the utility of classifiers on medical datasets. It shows that different datasets require different classifiers. Some datasets are used even if there's few samples available and few features.

Igor Kononenko has written [14] which is very useful in the context of machine learning and diagnostics. The author investigates how machine learning was designed and developed for medical purposes. The paper describes the history, current state and future direction of machine learning in medical diagnosis. The focus is on naive Bayesian classifier, neural networks and decision trees. The naive Bayesian classifier is specifically mentioned as a good alternative that should be tried before any other advanced method. Several important aspects are declared for a machine learning system to be useful for diagnosis: performance, dealing with missing data and noise, transparency of diagnostic knowledge, explaining decisions and the ability to reduce the amount of data and still be reliable.

7 classifiers are compared: Assistant-R, Assistant-I, Lookahead Feature Construction (LFC), naive Bayesian classifier, Semi-naive Bayesian classifier, Backpropagation with weight elimination (multilayered feedforward artificial neural network) and k-nearest neighbour. These classifiers were used for a total of 8 medical data sets not specified in the paper. The result is cited in table 2.

Classifier	Performance	Transparency	Explanation	Reduction	Missing data handling
Assistant-R	Good	Very good	Good	Good	Acceptable
Assistant-I	Good	Very good	Good	Good	Acceptable
LFC	Good	Good	Good	Good	Acceptable
Naive Bayes	Very good	Good	Very good	No	Very good
Semi-naive Bayes	Very good	Good	Very good	No	Very good
Backpropagation	Very good	Poor	Poor	No	Acceptable
k-NN	Very good	Poor	Acceptable	No	Acceptable

Table 2: The appropriateness of various algorithms for medical diagnosis

Based on this result, naive Bayes and semi-naive Bayes was found to be best.

One additional comparison was made using a heart disease data set from Ljubljana, Slovenia. This is shown in table 3.

Dataset	Classes	Samples
Heart disease diagnosis	2	327

Table 3: Additional data set

Of the classifiers mentioned, naive, semi-naive Bayes and Assistant-R achieved the best results on this data set. In general, Naive Bayesian classifier was particularly recommended by the author as a go-to classifier for most new data sets.

2.2 Diagnosis of specific diseases

One example of machine learning methods compared for the use in diagnosis of specific diseases is [15] written by Dreiseitl et al. They compare the 5 classifiers k-nn, logistic regression (LR), conjugate gradient optimization (CGO), decision trees (See5 decision tree software by Rulequest) and support vector machines (SVM-Light implementation). These machine learning methods are suggested as candidates for a decision tool to aid experts.

The data set is for pigmented skin lesions as common nevi, dysplastic nevi or melanoma and is summarized in table 4.

Dataset	Classes	Samples	Features
Skin lesions	3	1619	107

Table 4: Skin lesions data set

K-nn is described as robust and a good starting point to measure the other methods. The performance was evaluated with disregards to cost of model construction and model interpretability. Their only focus is on discriminatory power. According to their tests, logistic

regresseion, artifical neural networks (CGO) and support vector machines performed best. These gave almost identical results. Decision trees performed worst of the selection, but still achieves precision and recall similar to human experts. The rest were much better on this data set.

Chan et al. has in [16] made comparisons for multilayer perceptron, support vector machine, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), Parzen window (PW), mixture of Gaussian (MOG) and mixture of generalized Gaussian (MOGG). The data set is for glaucoma and summarized in table 5. They measured classifiers by the use of ROC.

Dataset	Classes	Samples	Features
Glaucoma	2	345	53

Table 5: Glaucoma data set

Experiments were conducted with and without the use of the feature reduction methods forward selection and backward elimination. The best classifiers according to ROC were found to be support vector machine, quadratic discriminant analysis and mixture of Gaussian.

2.3 Summary

Title	Author(s)	Classifiers	Winners
The use of the Aera Under the ROC curve in the evaluation of machine learning algorithms	Andrew P. Bradley	QDF, k-nn, C4.5, Multiscale, Perceptron and MLP	QDF, k-nn, Perceptron and MLP
Machine learning for medical diagnosis - history, state of the art and perspective	Igor Kononenko	Assistant-R, Assistant-I, LFC, NB, Semi-NB, MLP and k-nn	NB, Semi-NB, Assistant-R
A Comparison of Machine Learning Methods for the Diagnosis of Pigmented Skin Lesions	Dreiseitl et al.	k-nn, LR, ANN, See5, SVM	LR, CGO and SVM
Comparison of machine learning and traditional classifiers in glaucoma diagnosis	Chant et al.	MLP, SVM, LDA, QDA, PW, MOG, MOGG	SVM, QDA and MOG

Table 6: Research summary

My motivation is to make justifiable decisions for the proposed solution based on the relevant literature presented here. The proposed solution will include unique comparisons for a new data set.

Chapter 3

Data set and proposed solution

This chapter describes the data set and proposed solution in detail. Background for selected machine learning techniques and corresponding settings are also discussed here. All of this will be used in the experiments of Chapter 4. I used Weka 3.6 machine learning suite for all machine learning algorithms in this project, with the addition of LibSVM. Both the stand-alone GUI and java library were used. The choice of Weka was made after discussions and recommendations from other students and teachers. With no prior experience, I also chose Weka because of the good documentation and multitude of available guides and walkthroughs.

3.1 Solution Architecture

The answer for the 3 research questions consists of 3 experiments. The experiments are sequential and can affect subsequent experiments. Figure 1 illustrates the architecture of how these experiments will be conducted.

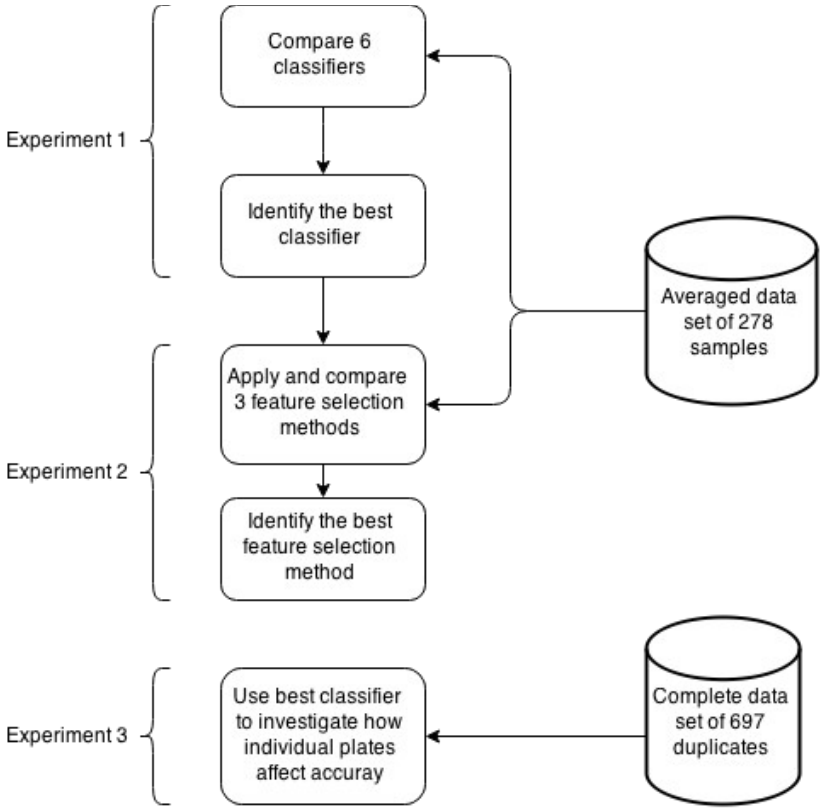

Figure 1: Solution architecture

3.2 Details of the data set

As mentioned in introduction, the data set is built up from 278 unique samples. The binary classes are fairly balanced with 163 negative samples and 115 positive samples. Each sample has 54 features which represent the amount of bacteria DNA found from 54 preselected species/families. Samples were analyzed on a total of 19 microbial stool analysis plates. The 278 samples were split into a total of 697 duplicates and distributed over the 19 plates. The data set also includes the position each sample had on a plate, ordered in rows and columns. The samples' position on plates are simply ignored in this project as it may be irrelevant and if it should be relevant there still wouldn't be enough data to say much of certainty about it.

Negative samples were analyzed on plates 1-13 and positive samples were analyzed on plates 14-19. This can be problematic for evaluating the results. A classifier could easily be misguided by an inaccurate feature extraction from plates. Instead of identifying a sample based on its class-dependent features, it would then be classified based on which plate it was analyzed on. Thus making any results worthless in the context of diagnostics.

Because dysbiosis has no exact definition, classes are labeled depending of the assumed status of the patient. So both class labels and feature values are expected to be noisy to an uncertain degree.

Figures 2-4 illustrates the average feature values and standard deviations between classes across all plates. The x-axis shows feature ID and y-axis shows extracted values representing amounts of bacteria. Although quite similar, for higher values, negative samples seem to have higher values than positives. And for lower values, negative samples have lower values than positives in many

cases. In general, negative class seems to go slightly more into the extremes while positive class stays somewhere in between.

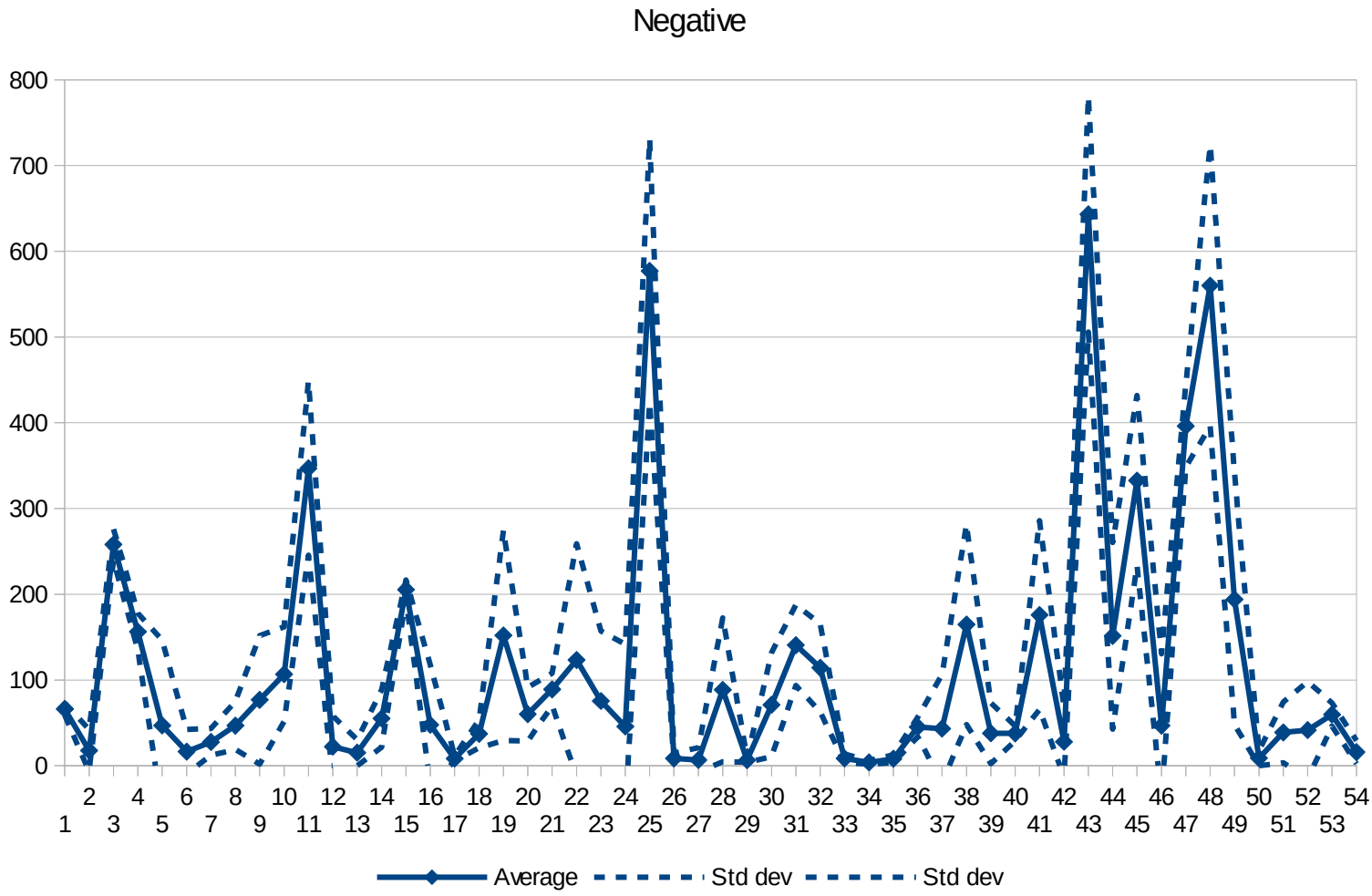

Figure 2: Averages and standard deviations for the negative class

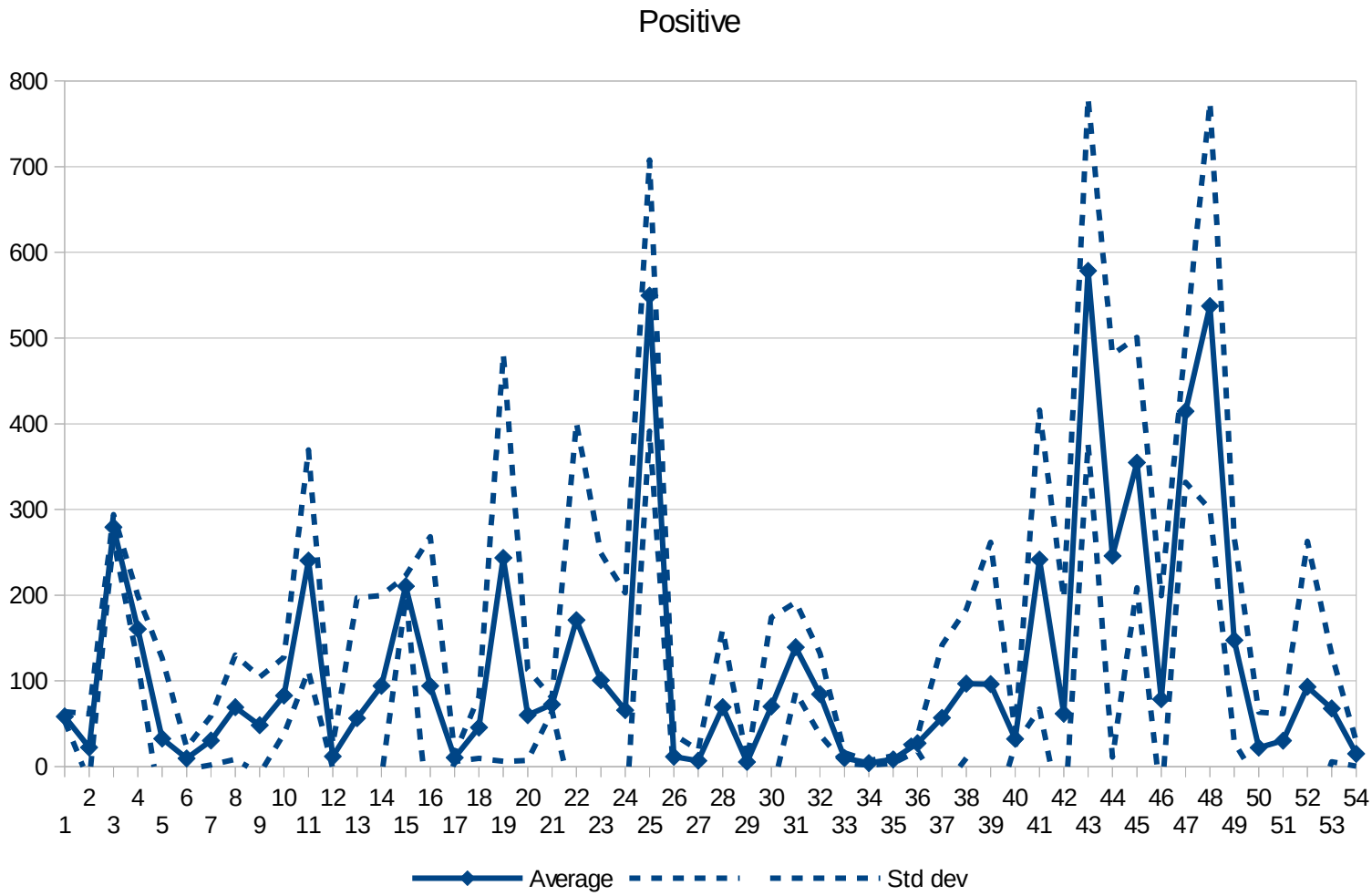

Figure 3: Averages and standard deviations for the positive class

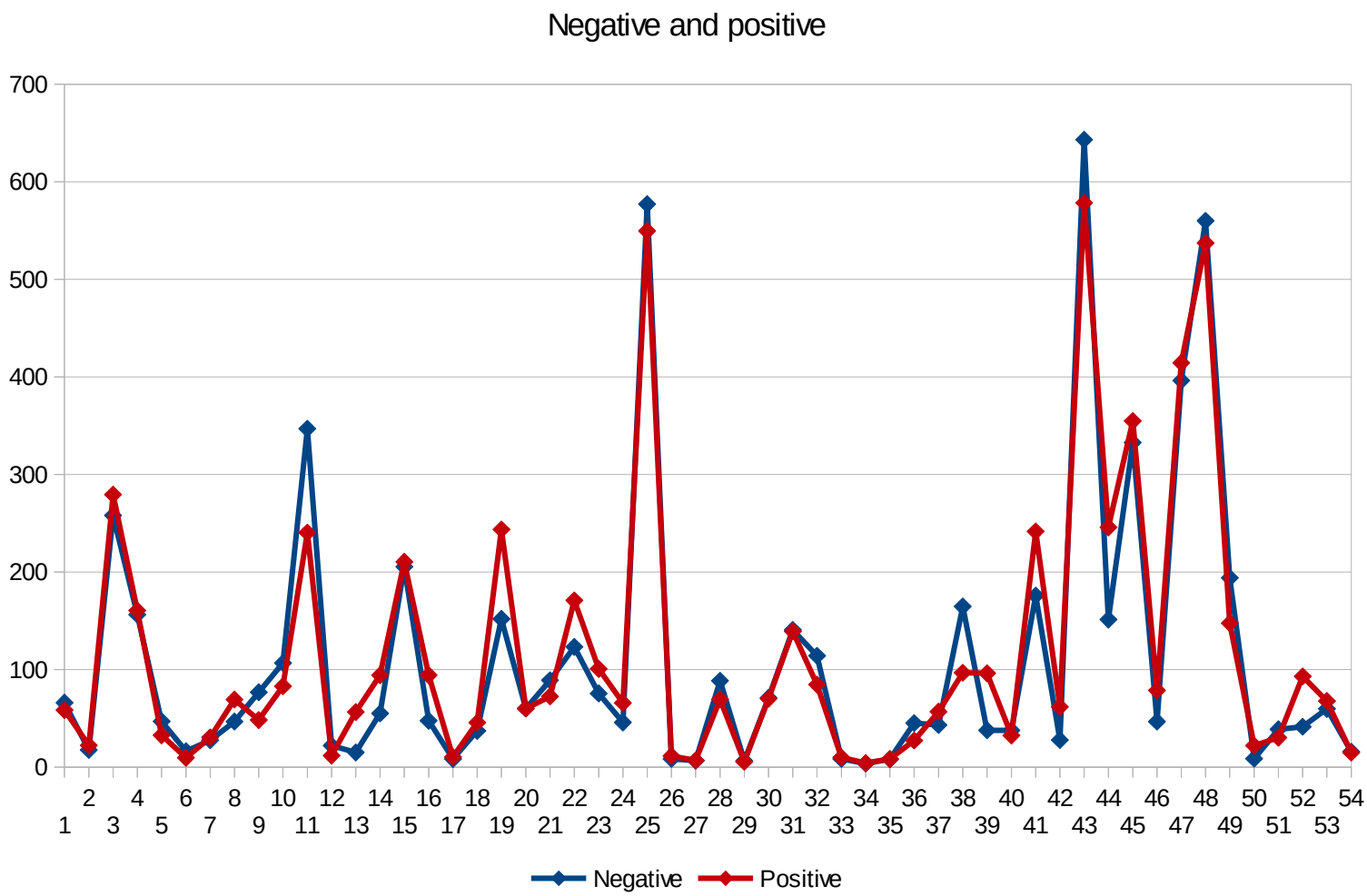

Figure 4: Averages for both classes

3.3 Classifier Selection

The proposed solution is based on a small, but wide search for high performing classifiers. The classifiers are expected to map with varying degree of success a sample's feature vector to a class label (negative or positive). As with many new data sets, it is not intuitively or pre-known which classifier is best fit for the job. The selection is mostly unaffected from any personal experiences as I have only tried naive Bayes and k-nearest neighbours before. I wish to avoid any expert bias by minimizing the search for optimal parameters per classifier. The expert bias means that having good experience and knowledge about one classifier would be a disadvantage to the others in a competition. With less focus on finding the perfect settings, chances are that the classifiers may under-perform and be under-fitted, however this a risk that applies to most of the classifiers. Naive Bayes and k-nearest neighbours which I could potentially have an expert bias towards happens to have no parameters in the case of naive Bayes and just a few in the case of k-nearest neighbours. The parameter search for the classifiers are simply done by adjusting one "lever" at a time. Even though this does not cover all combinations, it does save a lot of time. With some classifiers

having a multitude of settings, the parameter space just explodes because of the classic "curse of dimensionality". Instead, some settings can be guided by other work on similar cases. Based on State of the Art in Chapter 2, I have picked out 6 candidates. My main goal for the selection is not to make the basis for finding the best within a single "family", but instead present a wide selection. Therefore typically no more than one member per branch of machine learning classifiers is included. And after finding a winner, possible future experiments could instead open up for that classifier's branch to be revisited and made basis for a new narrower selection.

All classifiers are selected only with regard to discriminatory power. Any other performance is ignored, be it resource usage, classification time, model construction time or other computational time. Most of them will be almost instant or fairly fast anyway with such a small data set. The data set is not scaled or manipulated unless it is embedded in Weka and required by the classifier.

3.3.1 Naive Bayes

The first contender for the prize of being the best diagnostic tool for dysbiosis is an old and well used classifier. This is the naive Bayes, one my first acquaintances from the world of classifiers. It is a probabilistic classifier based on Bayes Theorem. It can take large amounts of feature vectors and can handle thousands of features with ease. Sometimes called simple or stupid, it assumes independence between features which is why it has the name naive. It learns by calculating features' distribution per class. Therefore the learning size is constant regardless of feature vectors added. New data can also be added to the existing learning data fairly easily. It can make a decision by calculating posterior probability using feature f for each class c and selecting the highest value as seen in figure 5. To avoid multiplying numerous small values which would quickly be rounded to zero, capital pi is replaced with summation of logarithmic probabilities instead.

$$P(c|f) = \frac{P(f|c)P(c)}{P(f)}$$

$$P(c|f) = P(f_1|c)P(f_2|c)P(f_3|c)...P(f_n|c)$$

$$\prod_{j=1}^{n} P(f_j|c_i)$$

$$hMAP = \underset{c \in C}{arg\,max}\ P(c|f)$$

Figure x
Figure 5: Bayes Theorem used in naive Bayes

It is mentioned in [14] as a first choice for classification and a benchmark to measure other classifiers against. Naive Bayes works fast even though this will not affect the ranking in this project. It is also a good start for beginners as it has no parameters to adjust. According to the criteria defined in [14] Naive Bayes has very good performance, good transparency, very good explainability, no reduction and very good missing data handling. The current data set has no

missing values so this is at the moment irrelevant. The data set do have a lot of features however, and this method does not offer any embedded reduction of those. The explainability is useful for users unskilled in machine learning to understand how decisions are made. Naive Bayes is also included in [13] as a top 10 influential data mining algorithm. Naive Bayes does have a lot of modified versions, usually to compensate for the independence assumption. This does increase complexity and reduces the simplicity. So I have chosen to only consider the basic implementation here.

3.3.2 K-nearest neighbours

The k-nn classifier is another classifier included in [13] and should therefore be a safe bet to include in the proposed selection. [15] recommends it as a good benchmark tool for other classifiers as well as describing it as robust. K-nn is categorized as lazy in Weka because it doesn't process the data in any way during learning phase. Instead, it simply just copies entire feature vectors into corresponding class tables. Adding more data later is therefore just as easy as if included from the start. But the size doesn't scale and so can be undesirable or unusable for very large amounts of training data. K-nn uses a few settings, one of which is the k from it's name. K is the number of votes the classifier uses to make a decision. This is illustrated in figure 6.

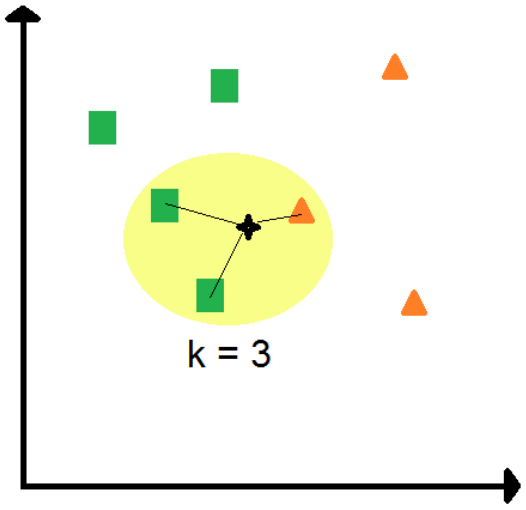

Figure 6: K-nearest neighbours example

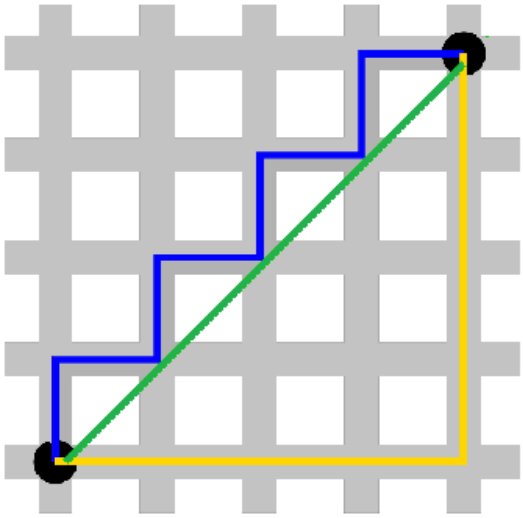

Figure 7: Manhattan and euclidean distance

With a simple data set of 2 dimensions, feature vectors can be plotted as points in a x-y plane. All points from the learning data are iterated through. The distance from each point to the unlabeled sample is calculated by some distance function. Two examples are Manhattan distance and Euclidian distance (Pythagorean theorem). Figure 7 illustrates the difference with blue and yellow marking Manhattan or taxicab distance and green marking the air distance or Euclidean distance. The decision is simply made by majority voting from the k nearest samples of the learning data,

also called neighbours. Therefore k is preferably an odd number to always assure a majority result. With higher dimensionality the math is done exactly the same. Additional distance weighting is possible to give closer neighbours more influence during voting.

The ranking regarding to criterias from [14] give k-nn a very good performance, poor transparency, acceptable explanation, no reduction and acceptable missing data handling. Because k-nn doesn't have any feature reduction, noise could imply a need for higher values of k. During testing however, the optimal value for k was found to be 5. Of the two distance functions mentioned, Manhattan distance performed best. Distance weighting did not improve performance.

3.3.3 Logistic regression

Logistic regression has shown good results on other medical data sets. It is reported as being among the best for some data sets. According to [15], classes are separated using a hyperplane and a logistic function to calculate probabilities of class membership according to distance. It is commonly used for diagnostics purposes allthough it is limited to only calculating linear decision boundaries. Allthough similar to Naive Bayes, a key difference is how correlations between features are taken into account, resulting in more calibrated predictions.

Best performance was gained with maximum number of iterations set to 5.

3.3.4 Multilayer perceptron

According to [14], neural networks lost attention after single-layered perceptrons were proven incapable of solving nonlinear problems. Fortunately developments lead to associative neural networks and later the backpropagation rule for neural networks. Today, this is a common classifier for medical use. For example scoring among the best in the experiments of [15]. The criterias for medical classifiers in [14] ranks it as having very good performance, poor transparency, poor explanation, no reduction and acceptable missing data handling. The reason given for why transparancy and explanation is rated so low is because of typically large amounts of weights which influence the result doesn't easily explain the decisions. Artifical neural networks have often been described as black boxes because of the difficulty to interpret or explain the behaviour. Because the wide-spread use for medical data sets and good results, I think it is a natural choice for inclusion in my selection.

Multilayer perceptron's implementation in Weka has alot of settings. These are configured with inspiration from similar use in [15]. Best results was achieved with 2 hidden layers, learning rate of 0.01, momentum set to 0.2 and 2000 epochs.

3.3.5 Support vector machine

Support vector machine is rated as one of the top 10 data mining algorithms in [13] and described as a "must-try" with high rating for both robustness and accuracy. It is also among the best classifiers found in similar comparisons of Chapter 2. It needs little training data and is insensitive to the amount of features. Support vector machine works by separating the classes with a line, a plane or a hyperplane in the case of more than 3 dimensions. The maxiumum margin hyperplane is only dependant of the support vectors and other data points play no part in the calculation. Thus giving great resilience to overfitting, or in other words better generalization for future data. This could be very useful for the data set used here. A kernel function is used to increase

dimensionality. A non-linear kernel function can allow for classification of non-linear data.

Of all the classifiers selected in this project, Weka has the highest amount of settings for support vector machine. The polynomial kernel function performed best. The greatest performance I was able to achieve was found after normalizing data, setting gamma to 1 and epsilon to 0.001.

3.3.6 C4.5

The C4.5 decision tree is named "J48" in Weka's implementation. This classifier is based on a top-down recursive divide-and-conquer strategy. A feature is selected as root node. Every node is asking a question about its feature. A node for each possible answer is attached as children. This is repeated recursively, leaving all leaves as class labels. The feature that produce the purest node is chosen as root. That is, the feature which discriminate most between classes. The nodes are selected with information gain heuristic. A split in the tree is measured by how much information is gained before the split minus the information gained after the split. The algorithm then prunes the tree to make it smaller. To classify a sample, simply traverse the tree and use the resulting leaf node as class prediction. This classifier has a simple and strong explanatory capability, making it easy for humans to understand its decisions.

Minimum number of objects were set to 0 to achieve better performance.

3.4 Metric

There are alot of alternatives for comparing and ranking classification results. The confuision matrix accounts correct and incorrect classifications per class. This is the normal raw data which a classifier outputs and is the basis for such measures like accuracy, precision and recall. Accuracy is commonly used due to its simplicity. But it must always be compared to a random result. For binary classification, this equals to 50% which represents random chance for a balanced data set. For my project, I have chosen Cohen's kappa to be the judge. This has some benefits over accuracy because it compensates unbalanced size of classes. It is also a scaled measure. Another measurement which is strongly advocated for in literature is ROC. The problem with ROC is that it can't be measured by the confusion matrix alone, it needs to adjust some threshhold inherit in the classifier. Weka calculates ROC automaticly during a standard classification, but some of my experiments will go beyond this and then ROC is not available. Arie Ben-David shows in [17] that ROC and kappa are connected and have alot in common, so I think kappa is still very useful even though ROC may be better in some circumstances. According to [17], kappa is a scalar meter of accuracy which measures the degree of agreements between reality and the classifier.

Table 7 shows the binary class confusion matrix. Figures 8 and 9 shows how different measures are calculated.

Actual \ Predicted	Negative	Positive
Negative	A (true negative)	B (false positive)
Positive	C (false negative)	D (true positive)

Table 7: Standard binary classification confusion matrix

In addition to using kappa as metric for comparisons, the experiments done in Chapter 4 will also provide corresponding confusion matrices for better interpretation. Other measures will be added where available.

$$\text{Kappa} = \frac{\frac{A+D}{A+B+C+D} - \frac{(A+B)(A+C)}{A+B+C+D} + \frac{(B+D)(C+D)}{A+B+C+D}}{1 - \frac{(A+B)(A+C)}{A+B+C+D} + \frac{(B+D)(C+D)}{A+B+C+D}}{A+B+C+D}$$

Figure 8: Kappa formula

$$Accuracy = \frac{A+D}{A+B+C+D}$$

Figure 9: Accuracy formula

3.5 Feature reduction methods

The proposed solution involves an optimization process. To optimize a classifier , I want to try feature selection methods. More specifically, the plan is to measure the effect of feature reduction on the winning classifier from the selection above. Because the data set has a lot of features with several of them assumed to be noisy, a classifier could benefit greatly from this approach. Some features may be redundant and redundant, others may be noisy and confusing to the classifier. Removing such features can lead to increased accuracy. An added bonus of reducing data is that computational time for learning and classification inevitably decreases. Less data means less time spent, even though such improvements are not the goal here. A good result should be a feature subset with features that are highly predictive of the class and not predictive of other features.

Just as there are alot of classifiers to choose from, there are also alot of feature selection methods avilable. I expect that at least one classifier can achieve a performance to such a degree that very little space is available for improvement. An extensive search for the best feature selection methods is therefore considered too costly. Instead, I present a small selection consisting of one wrapper

and two filters. A wrapper uses a classifier's results to adjust its selection while a filter makes the selection independent of any classifier. These are described below.

Many feature selection methods in weka require a search method. Just as the abundance in alternatives for the previous selections, this option also has multiple possible alternatives. At this point, I did not want further selections and comparisons. To avoid getting too much data out of this, I simply went with the Best First search method for all proposed feature selection methods. Search direction is set to backwards based on suggestion in [18].

3.5.1 Wrapper subset evaluator

The wrapper implementation in Weka is an implementation of John et al's wrapper from [19]. According to [18], John et al were the first to promote wrappers as a general framework for feature selection. With formal definitions for feature relevance, wrappers are claimed to The search space for an optimal feature subset with n amount of features is 2^n. Usually an exhaustive search is needed to find the very best subset. This is in many cases impractical and therefore a heuristic search is used instead. The wrapper uses a learning algorithm's discriminatory performance to evaluate the feature subset, in my case that means the winning classifier from the classifier comparison. The wrapper has one major bias because it will tune the subset for the target classifier.

My original plan was to implement a genetic algorithm-based wrapper, but this is a very time-consuming approach and would require the target classifier to work very fast. It also has alot of options which needed to be adjustet like genetic operators and mutation rates. Instead, I chose Weka's wrapper implementation with default settings.

3.5.2 Correlation-based feature selection

According to Hall in [18], this filter ranks features subsets using a heuristic evaluation. The algorithm aims for subsets that include features of high correlation to class and low correlation to other features. More details about this filter can be found in [18].

3.5.3 Consistency subset evaluator

According to Dash et al. in [20], this filter measures inconsistency in feature subsets. Evaluation conciders samples with same feature values, but belonging to different classes. If such pattern exists, it is concidered as being incosistent using an inconsistency rate. The rate sums up inconsistency count. Consistency measure is described as monotonic, fast, multivariate, capable of reducing some noise and removing redundant and irrelevant features. More details about this filter can be found in [20].

Chapter 4

Experiments

The experiments below have been designed to correspond to the research questions described in Chapter 1. The first goal is to establish that at least one classifier is suitable for use in diagnostics with the current data set. And preferably that some are better suited than others, hence the need for a comparison. This is needed for some of the other experiments from the proposed solution to be viable.

Assuming the first experiment proves classifiers to be usable for the data set, the second experiment will try to optimalize classification with feature reduction methods. The last experiment is used to discover the impact different plates have in the feature extraction process. This could help indicate any weakness in the results of previous experiments.

Experiment 1 (E1)	Run averaged data set on the six classifiers and produce ranking
Experiment 2 (E2)	Optimalize best ranked classifier by the use of feature reduction
Experiment 3 (E3)	Test how each plate affect classification

4.1 Experiment 1 (E1)

The data set includes 278 samples spread out in a total of 697 duplicates across 19 plates. For this experiment, the 697 feature vectors are averaged back to 278 unique samples. The averaged data set therefore includes 278 feature vectors. All 54 features are used in this experiment. To give the results higher generalizability for future unknown data, the leave-one-out cross-validation scheme is used. For each sample k in the data set D, training data T = D – k. This prevents any learning data to be contaminated by the classification sample. Only kappa is used for ranking, but other measures are presented. This experiment results in a ranking of the proposed classifiers.

4.1.1 Naive Bayes

Classification results for naive Bayes on the averaged data set is shown in table 8.

A.\P.	negative	positive
negative	140	23
positive	39	76

Table 8: Naive Bayes results of E1

Additional results are shown in table 9.

Class	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Weighted avg.	0.777	0.257	0.776	0.777	0.774	0.853

Correct	Incorrect	Kappa	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
216	62	0.531	0.222	0.460	45.6232%	93.1015%

Table 9: Additional naive Bayes results of E1

Scores for kappa and ROC are the worst of the entire experiment. Discriminatory power is low for both classes. This data set does not illustrate the strength of naive Bayes classifier. This classifier is unsuited independent of what measurement is preferred.

4.1.2 K-nearest neighbours

Classification results for k-nn on the averaged data set is shown in table 10.

A.\P.	negative	positive
negative	155	8
positive	28	87

Table 10: K-nn results of E1

Additional results are shown in table 11.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Weighted avg.	0.871	0.163	0.875	0.871	0.868	0.933

Correct	Incorrect	Kappa	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
242	36	0.726	0.198	0.329	40.5716%	66.4557%

Table 11: Additional k-nn results of E1

These results are quite impressive when concidering how simple the algorithm is. Allthough kappa score is unsatisfactory, ROC is quite good.

4.1.3 Logistic regression

Classification results for logistic regression on the averaged data set is shown in table 12.

A.\P.	negative	positive
negative	154	9
positive	14	101

Table 12: Logistic regression results of E1

Additional results are shown in table 13.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Weighted avg.	0.917	0.094	0.917	0.917	0.917	0.966

Correct	Incorrect	Kappa	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
255	23	0.828	0.099	0.259	20.3555%	52.3549%

Table 13: Additional logistic regression results of E1

These results are modest. ROC is very good allthough kappa is medium in comparison to the other classifiers.

4.1.4 Multilayer perceptron

Classification results for multilayer perceptron on the averaged data set is shown in table 14.

A.\P.	negative	positive
negatiove	155	8
positive	11	104

Table 14: Multilayer perceptron results of E1

Additional results are shown in table 15.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Weighted avg.	0.932	0.076	0.932	0.932	0.932	0.982

Correct	Incorrect	Kappa	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
259	19	0.859	0.068	0.233	14.0226%	47.2137%

Table 15: Additional multilayer perceptron results of E1

For some reason this classifier has excellent ROC score even if kappa is just slightly above average. I am not sure exactly why this is the case. If ROC should be the desired benchmark measurement for future work, this classifier is highly suggested.

4.1.5 Support vector machine

Classification results for support vector machine on the averaged data set is shown in table 16.

A.\P.	negative	positive
negative	154	9
positive	8	107

Table 16: Support vector machine results of E1

Additional results are shown in table 17.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Weighted avg.	0.939	0.064	0.939	0.939	0.939	0.938

Correct	Incorrect	Kappa	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
261	17	0.874	0.061	0.247	12.5583%	50.0322%

Table 17: Additional support vector machine results of E1

Support vector machine has very strong results all over. Performance is good across both classes. This classifier is a good alternative for future work.

4.1.6 C4.5

Classification results for C4.5 on the averaged data set is shown in table 18.

A.\P.	negative	positive
negative	156	7
positive	5	110

Table 18: C4.5 results of E1

Additional results are shown in table 19.

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Weighted avg.	0.957	0.043	0.957	0.957	0.957	9.530

Correct	Incorrect	Kappa	Mean absolute error	Root mean squared error	Relative absolute error	Root relative squared error
266	12	0.911	0.049	0.200	9.9809%	40.5152%

Table 19: Additional C4.5 results of E1

Of all the classifiers compared in this experiment, C4.5 has the highest kappa score. The embedded feature selection of this decision tree could be the deciding factor why this outperforms the rest. However, ROC score is disproportionate compared to previous classifiers. Performance is proportionate for both classes. This data set really illustrates the strength of C4.5.

Weka can output the tree produced during classification. Because this experiment uses leave-one-out scheme, it means one unique tree is made for each sample classified. Instead of showing all 278 trees, a tree based on the entire data set is shown in figure 10. Correct and incorrect classifications are shown in paranthesis for each leaf (correct/incorrect).

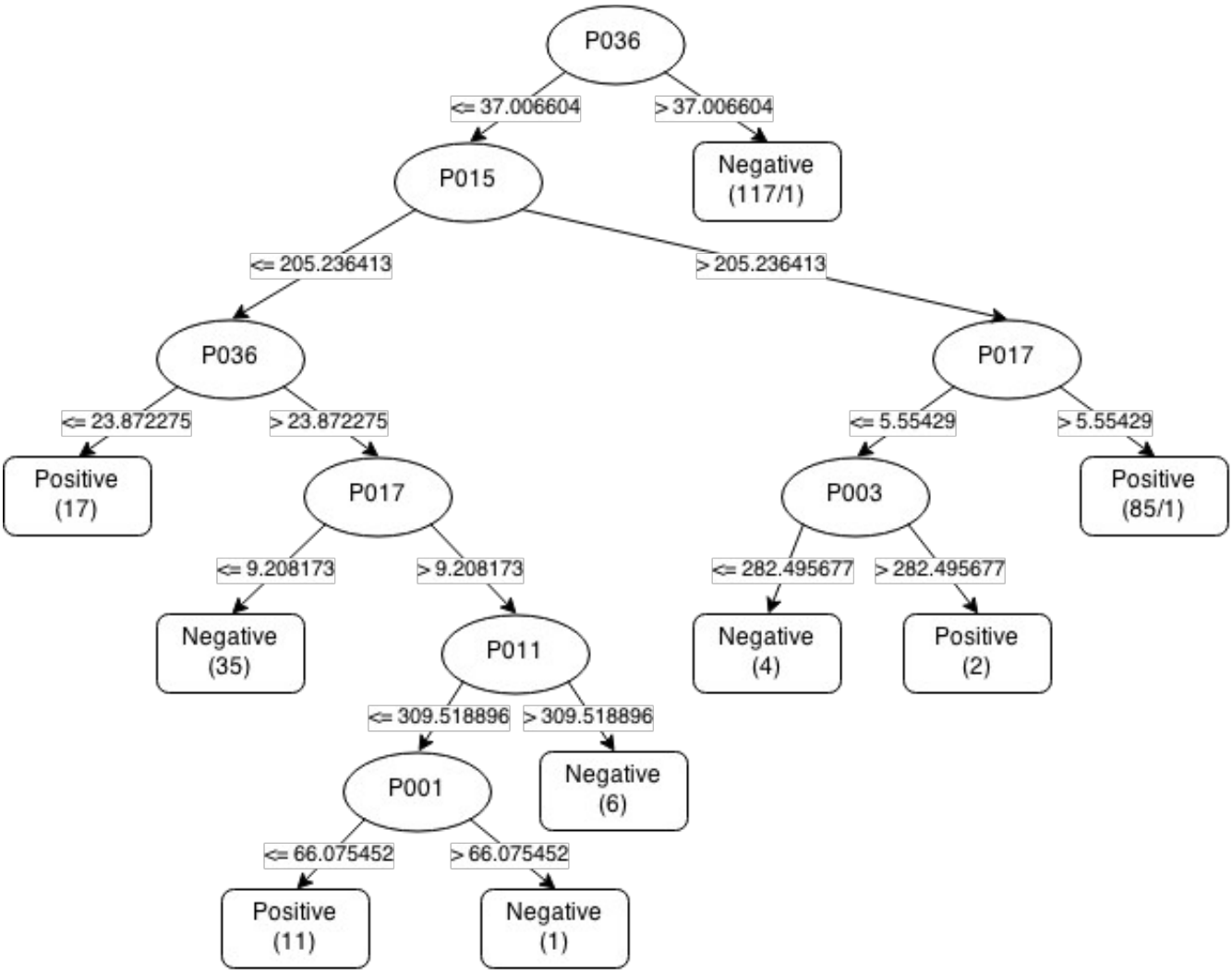

Figure 10: Visualized C4.5 tree from complete data set

The tree has size 17 with a total of 9 leaves. Here we can see perfectly the sort of embedded "feature reduction" typical for decision trees. Out of 54 features, only P001, P003, P011, P015, P017 and P036 are used after pruning.

4.1.7 Discussion

According to summarized results of table 20, the highest ranking classifier is the C4.5 decision tree.

Naive Bayes	0.531	216	62	0.853
K-nearest neighbours	0.726	242	36	0.933
Logistic regression	0.828	255	23	0.966
Multilayer perceptron	0.859	259	19	0.982
Support vector machine	0.874	261	17	0.938
C4.5	0.911	266	12	0.953

Table 20: Summarized results of E1

The difference between the lowest and highest ranked classifiers is substantial. This was expected based on similar work discussed in Chapter 2. The difference therefore justifies the comparison and proves RQ1 correct. But more importantly, it identifies C4.5 as the best alternative. Interestingly, the ROC Area does not follow the same order. If this was the chosen metric, multilayer perceptron would have been the winner and subject of subsequent experiments. Should someone argue for ROC Area or some other metric to be better for comparison, I would advice repeating the subsequent experiments with the best classifier according to that metric.

Because C4.5 scored so high, searching for an even better classifier could easily require a much larger selection. The cost of finding a better classifier is likely relative to C4.5's performance. Since the proposed selection is so diverse, a narrow successive search including only decision trees could be profitable. Further experiments are needed to confirm this.

Due to a relatively small data set, it is possible that another classifier in this list would be better on data set containing more samples. It is important to note that this ranking is only applicable for this specific data set. Any modification in features or size could lead to a different classifier having better performance.

The tree produced by C4.5 based on the complete data set only includes 6 features, indicating that a large number of features may be obsolete at least for this classifier. The strong explanatory power of this classifier may be helpful in pointing out specific bacterias that play a vital role for the dysbiosis condition. Most classifiers in my selection have limited or no embedded "feature selection" functionality. C4.5 includes similar functionality to feature selection which may have helped it gain a foothold in this contest. This is why decision trees get a positive rating for data reduction, as seen in [14]. Classification of complete data set using all features was part of the condition for the comparison and this ordering of experiments prevents an overwhelming amount settings to be adjusted and data to discuss.

4.2 Experiment 2 (E2)

Now that a ranking of classifiers shows that the C4.5 achieved highest kappa score, the second experiment can improve this result by the use of feature selection methods. Even though C4.5 do have some of this functionality embedded, performance can potentially still be improved because of the heuristic information gain search algorithm used. Noisy features can make additional feature selection methods viable, because C4.5 is not guaranteed to find the best subset on its own.

This experiment uses the same data set as in E1 with 278 unique averaged samples. The same leave-one-out cross-validation scheme is used. Each feature selection algorithm produces a threshhold value per feature. Because leave-one-out cross-validation is used, the output is not one single feature subset, but one feature subset per classified sample. The threshhold values express the percentage of how many times the features were used. So a high percentage means the feature was used many times. This can be used to rank features. For each threshhold table presented in the following experiment, the optimal count of features is calculated by classification. This can be done in two equivalent ways:

- increase threshhold step by step to find best value
- start with all features and remove one by one to find best rank number

Each result is presented with feature threshholds, the optimal threshhold value that achieved highest kappa, confusion matrix for classification using this threshhold and additional measures. As mentioned before, the features represent a pre-defined selection of bacteria species and families. Each bacteria is anonymized and labeled P0001-P0054. The feature subsets produced may be helpfull in indicating important relations and identities of such bacterias that play a vital role in dysbiosis.

4.2.1 Wrapper subset evaluator

This method uses a target classifier for evaluation of subsets. The settings used for C4.5 in this experiment is the same as in experiment 1. The feature threshholds produced by the wrapper is shown in table 21.

P0001	P0002	P0003	P0004	P0005	P0006	P0007	P0008	P0009	P0010	P0011	P0012	P0013	P0014	P0015	P0016	P0017	P0018
0.04	0.19	0.74	0.13	0.03	0.05	0.03	0.09	0.10	0.10	0.94	0.05	0.19	0.03	1.00	0.13	1.00	0.09
P0019	P0020	P0021	P0022	P0023	P0024	P0025	P0026	P0027	P0028	P0029	P0030	P0031	P0032	P0033	P0034	P0035	P0036
0.04	0.19	0.12	0.10	0.12	0.01	0.04	0.05	0.07	0.21	0.09	0.09	0.03	0.30	0.08	0.14	0.09	1.00
P0037	P0038	P0039	P0040	P0041	P0042	P0043	P0044	P0045	P0046	P0047	P0048	P0049	P0050	P0051	P0052	P0053	P0054
0.30	0.13	0.02	0.03	0.06	0.06	0.35	0.11	0.12	0.14	0.05	0.14	0.07	0.16	0.11	0.06	0.02	0.03

Table 21: Feature threshholds produced by wrapper

C4.5 achieved best classification score with a threshhold set to 0.74. The corresponding feature subset consists of P0003, P0011, P0015, P0017 and P0036. The classification results for this subset are shown in table 22 and 23.

A.\P.	negative	positive
negative	158	5
positive	4	111

Table 22: Wrapper results of E2

Class	TP Rate	FP Rate	Precision	Recall	F-Measure
Weighted avg.	0.968	0.033	0.968	0.968	0.968
Correct	Incorrect	Kappa			
269	9	0.933			

Table 23: Additional wrapper results of E2

By removing almost 90% of the features available, the Wrapper managed to improve performance compared to E1. The wrapper's use of C4.5 as evaluator produced a subset almost identical to C4.5's own subset shown previously in figure 10. With the good results from E1, there was only a small space for improvement. However, by removing just one feature, P0001, the decision tree went from 12 misclassifications to 9. This makes the already good accuracy even better.

4.2.2 Correlation-based feature selection

The feature threshholds produced by the correlation-based feature selection (CFS) is shown in table 24.

P0001	P0002	P0003	P0004	P0005	P0006	P0007	P0008	P0009	P0010	P0011	P0012	P0013	P0014	P0015	P0016	P0017	P0018
1.00	0.00	1.00	0.00	0.00	0.00	0.00	0.88	0.03	1.00	1.00	0.00	0.83	1.00	0.99	0.98	0.97	0.00
P0019	P0020	P0021	P0022	P0023	P0024	P0025	P0026	P0027	P0028	P0029	P0030	P0031	P0032	P0033	P0034	P0035	P0036
1.00	0.00	1.00	0.98	0.02	0.00	0.00	0.00	0.00	0.96	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
P0037	P0038	P0039	P0040	P0041	P0042	P0043	P0044	P0045	P0046	P0047	P0048	P0049	P0050	P0051	P0052	P0053	P0054
0.00	0.20	0.02	1.00	0.00	0.00	1.00	1.00	0.00	0.00	0.00	0.00	0.02	1.00	0.00	0.00	0.00	0.00

Table 24: Feature threshholds produced by correlation-based feature selection

C4.5 achieved best classification score with a threshhold set to 0.96. The corresponding feature subset consists of P0001, P0003, P0010, P0011, P0014, P0015, P0016, P0017, P0019, P0021, P0022, P0028, P0036, P0040, P0043, P0044 and P0050. The classification results for this subset are shown in table 25 and 26.

A.\P.	negative	positive
negative	158	5
positive	5	110

Table 25: Correlation-based feature selection results of E2

Class	TP Rate	FP Rate	Precision	Recall	F-Measure
Weighted avg.	0.964	0.038	0.964	0.964	0.964
Correct	Incorrect	Kappa			
268	10	0.926			

Table 26: Additional correlation-based feature selection results of E2

With an increase in accuracy and almost 70% reduction of features, these results are also good, allthough not better than the previous.

4.2.3 Consistency subset evaluator

The feature threshholds produced by the consistency feature selection method is shown in table 27.

P0001	P0002	P0003	P0004	P0005	P0006	P0007	P0008	P0009	P0010	P0011	P0012	P0013	P0014	P0015	P0016	P0017	P0018
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.98	0.00	0.00	0.00
P0019	P0020	P0021	P0022	P0023	P0024	P0025	P0026	P0027	P0028	P0029	P0030	P0031	P0032	P0033	P0034	P0035	P0036
0.09	0.12	0.00	0.00	0.01	0.00	0.00	0.00	0.00	0.04	0.94	0.00	0.00	0.97	0.00	0.00	0.00	1.00
P0037	P0038	P0039	P0040	P0041	P0042	P0043	P0044	P0045	P0046	P0047	P0048	P0049	P0050	P0051	P0052	P0053	P0054
0.00	0.92	0.99	0.00	0.90	0.00	0.00	0.91	0.00	0.00	0.00	0.98	0.02	1.00	0.00	0.96	0.26	0.00

Table 27: Feature threshholds produced by consistency

C4.5 achieved best classification score with a threshhold set to 0.97. The corresponding feature subset consists of P0015, P0032, P0036, P0039, P0048 and P0050. The classification results for this subset are shown in table 28 and 29.

A.\P.	negative	positive
negative	151	12
positive	13	102

Table 28: Consistency results of E2

Class	TP Rate	FP Rate	Precision	Recall	F-Measure
Weighted avg.	0.910	0.097	0.910	0.910	0.910
Correct	Incorrect	Kappa			
253	25	0.814			

Table 29: Additional consistency results of E2

Of all the results in E3, these are the worst. This turned out to be a bad choice in this case. The reduction in size is good, from initial 54 to 6. But the specific features have very little overlap with the ones used in C4.5's tree based on full feature set.

4.2.4 Discussion

A summary of features selected in experiment 2 is shown in figure 11.

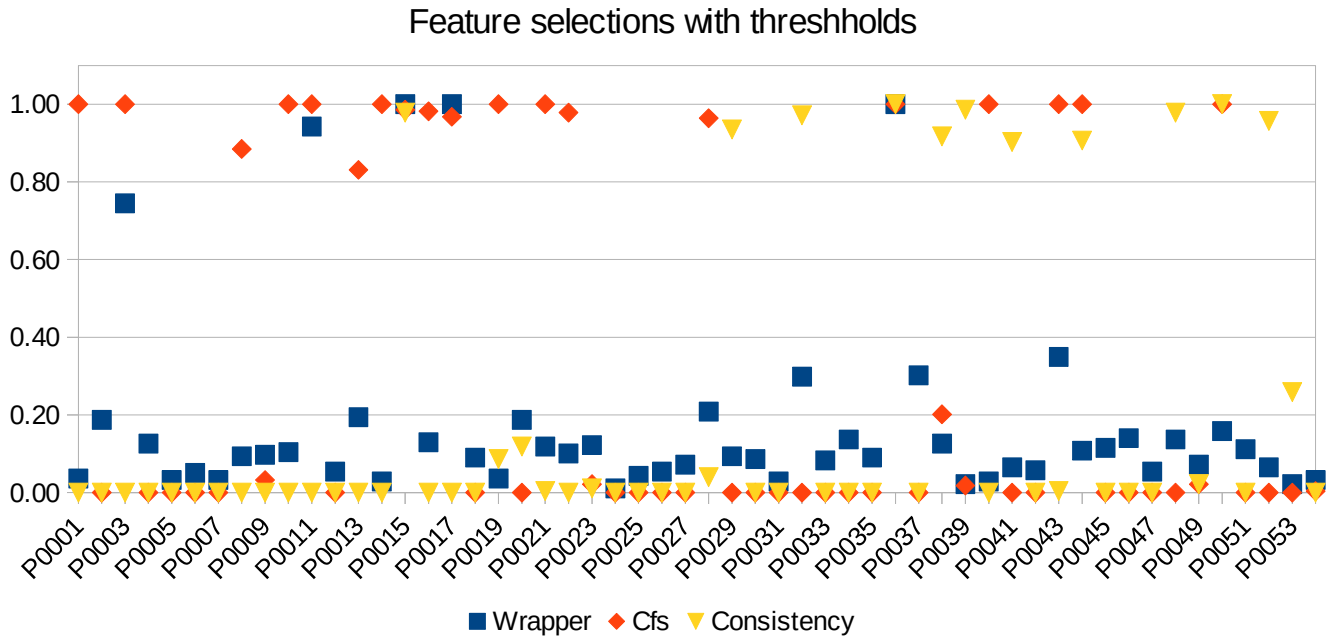

Figure 11: Feature selection comparison

A ranked list of classification results and comparison with full feature set from experiment 1 is shown in table 30.

	Kappa	Correct	Incorrect	Features
Wrapper	0.933	269	9	5
Correlation	0.926	268	10	17
Consistency	0.814	253	25	6
C4.5 Full feature set	0.911	266	12	54 (6)

Table 30: Summary of classification performance with and without feature subsets

Based on this ranking, the wrapper made the best feature subset selection for the C4.5 classifier. The improvement is actually surprisingly good. Originally, C4.5 only had 12 incorrect classifications. This was reduced to 9 misclassifications with the optimized process of wrapper feature selection. The amount of features varies greatly between the methods. Consistency did not give satisfactory results in this case. Perhaps a different search algorithm would be more approriate for the consistency method. Correlation-based feature selection impressed because it had the disadvantage of being a filter. Its evaluation of feature subsets is independent of any classifier, but still managed to give almost as good optimalization as the wrapper which had direct evaluation access of the C4.5 classifier. But the correlation-based feature selection method had one draw-back which is the size of the feature subset. Also interesting to see is that every feature selected by the wrapper is also found in the selections from C4.5 and correlation-based method. The difference between the wrapper's selection and C4.5's selection is the exclution of P0001. After wrapper

removed this feature, C4.5's kappa score increased from 0.911 to 0.933!

The varying success of the feature selection methods shows that also this step benefits from a comparison. And accuracy can both be improved and degraded depending of which method is used. Based on the results shown here, I would advice using wrapper for further work on this data set, allthough correlation-based feature selection is nearly as good. But the wrapper wins also because of a much smaller size of the subset. This experiment also shows that even if the target classifier includes some feature selection functionality, the use of additional methods can still be viable.

Hopefully the features identified here can be valuable for further research on determining if a patient has dysbiosis and which bacterias has most impact for this condition.

4.3 Experiment 3 (E3)

Some samples are analysied multiple times on the same plate. The vast majority of samples are analyzed more than once and on more than one plate. One plate typically has similar but not identical readings when analyzing the same sample twice. Figure 12 shows the distribution of the 278 samples on the 19 plates. Sample 1-163 are in negative class and samples 167-278 are positive. Plate 1-13 were used on negative samples and plate 14-19 were used on positive samples.

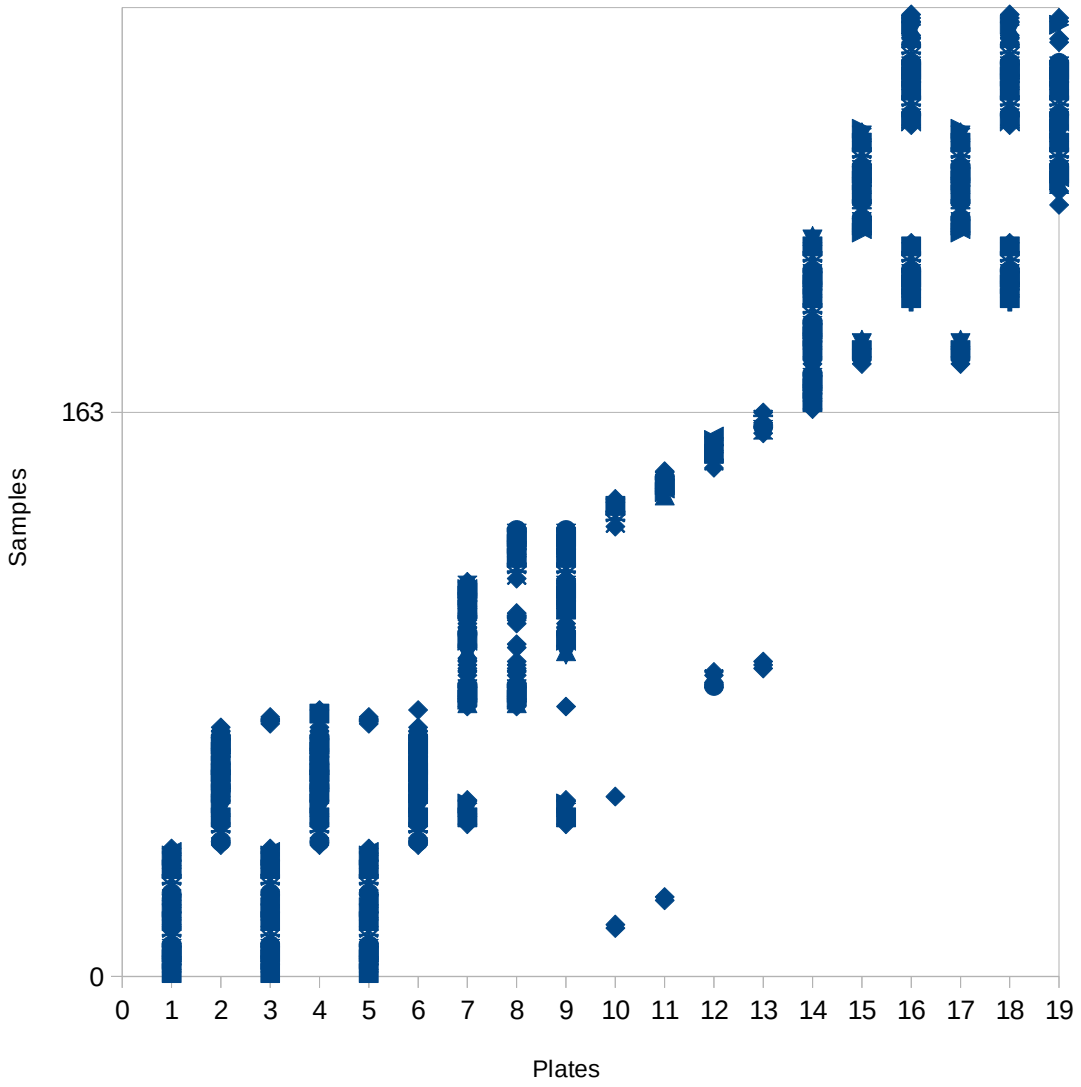

Figure 12: `Sample distribution on plates`

The third experiment explores the possibilty that classification performance is dependant of plates. Ideally each class is precise and each plate extracts identical feature values from the same sample. If this was true, classification performance should be independent of which plate was used to analyze the samples. But with real-world and noisy data, this is rarely the case. Also the classes are only assumede as no formal definition exists for dysbiosis.

This experiment uses winning classifier C4.5 from experiment 1 with the same settings. The data set for this experiment is organized differently however. In the previous experiments, average samples were used and the individual plates were transparent. Now the individual plates are of interest and therefore the complete raw data set consisting of 697 feature vectors are used. Instead of using leave-one-out in this experiment, a scheme of leave-one-set-out is used instead. One sample has multiple duplicates depending on how many times it was analyzed on one or more plates. When one set is left out, the training data is not polluted by duplicates of the classified sample. In this experiment, leave-one-set-out cross-validation means that when classifying a sample, no other duplicate of that sample is used for training the classifier.

4.3.1 Classification results per plate

Figure 13 shows the accuracy of C4.5 on the complete data set per plate.

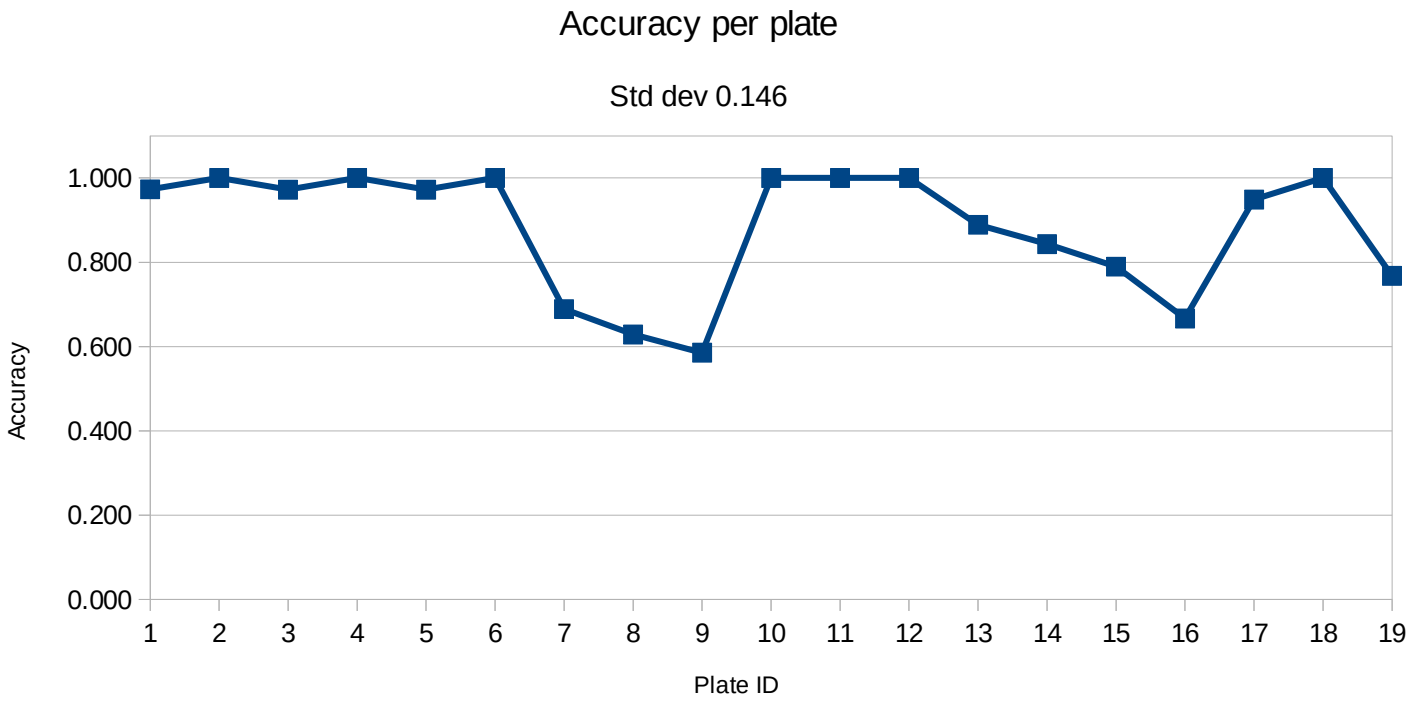

Figure 13: Accuracy per plate

4.3.2 Discussion

Results from experiment 3 shows that for negative class, plate 7, 8 and 9 are particularly bad. Many samples analyzed on these plates are not analyzed on other plates. Performance is also lower on plate 13, 14, 15, 16 and 19 in positive class. Several samples are only analyzed on some of these plates. All in all, the difference in classification accuracy between plates is very apparant. There can be many reasons why this is the case. Perhaps some samples are labeled incorrectly due to lack of precise definition of dysbiosis. Or some samples analyzed on these plates are particularly noisy. It is also tempting to conclude that precision in measure of bacteria is

inconsistent between plates. The result from this experiment indicates the possibility that some plates are simply less accurate for analyzing quantity of bacteria dna present in human feces.

Chapter 5

Conclusion and future work

I investigated the use of machine learning techniques on the binary classification problem of gut dysbiosis. The proposed solution included a unique selection of 6 classifiers which was recommended by or proved succesful in literature for medical diagnosis. In addition, feature selection methods were applied to improve accuracy and to reveal the identities of bacterias that were most relevant for classification. The last experiment classified samples with regard to the individual stool analysis plates they analyzed on.

Experiments were conducted using supervised learning classifiers and a leave-one-out cross-validation scheme. Comparisons showed that the discriminatory performance varied greatly between classifiers. The C4.5 decision tree performed best, with close competition from support vector machine, multilayer perceptron and logistic regression. Naive Bayes and k-nn did not perform suitably for this data set.

The accuracy of C4.5 was optimalized by the addition of feature selection methods. 3 alternatives were compared. Wrapper subset evaluator and correlation-based feature selection both produced strong feature subsets. Wrapper achieved the greatest improvement of accuracy by removing more than 90% of the features. Consistency subset evaluator did not perform satisfactory for the C4.5 classifier on this data set.

The samples were analyzed on 19 plates. Classification varied significantly depending of which plate was used to analyze the sample. This may indicate a certain level of noise and that some of the laboratory equipment used have inaccurate measurements.

The results will be useful guidelines for future work with the gut dysbiosis data set. Specific feature identities mapping to bacterias have been shown to be particularly predictive of classes. Any future binary classification will benefit greatly from selecting the apropriate algorithms and avoiding the others.

5.1 Future work

If for some reason there is reason to believe a different classifier is better suited or preferred, it could be compared to the current results. There is still a possibility of applying a classifier with better performance or a better feature subset. If more samples are collected to increase or alter the data set, it could be purposefull to re-evaluate the ranking of classifiers and feature selection methods. It would be interesting to see any re-ordering of the rankings if the data set is modified.

With a larger data set, a stronger result could be achieved by dividing it in two parts. One part for tuning and selecting features and the other part for evaluating efficiency.

Sources

[1] Alison M. Stephen and J. H. Cummings. The Microbial Contribution to Human Faecal Mass. In *Journal of Medical Microbiology*, volume 13, pages 45-56. Society for General Microbiology, 1980.

[2] C. P. Tamboli, C. Neut, P. Desreumaux and J. F. Colombel. Dysbiosis in inflammatory bowel disease. In *Gut*, volume 53. BMJ Group, 2004.

[3] P. Seksik. Gut microbiota and IBD. In *Gastroentérologie Clinique et Biologique*, volume 34, issue 4, supplement 1, pages 48-55. Elsevier, 2010.

[4] Philippe Marteau. Bacterial Flora in Inflammatory Bowel Disease. In *Digestive Diseases*, volume 27, supplement 1, pages 99-103. Karger 2009.

[5] P. Lepage, M. C. Leclerc, M. Joossens, S. Mondot, H. M. Blottiere, J. Raes, D. Ehrlich and J. Dore. A metagenomic insight into our gut's microbiome. In *Gut*, volume 62, pages 146-58. BMJ Group, 2012.

[6] Shaheen E. Lakhan and Annette Kirchgessner. Gut inflammation in chronic fatigue syndrome. In *Nutrition & Metabolism*, volume 7. BioMed Central, 2010.

[7] Peter J. Turnbaugh, Ruth E. Ley, Michael A. Mahowald, Vincent Magrini, Elaine R. Mardis and Jeffrey I. Gordon. An obesity-associated gut microbiome with increased capacity for energy harvest. In *Nature*, volume 444, issue 7122, pages 1027-1031. Nature Publishing Group, 2006.

[8] Peter J. Turnbaugh, Micah Hamady, Tanya Yatsunenko, Brandi L. Cantarel, Alexis Duncan, Ruth E. Ley, Mitchell L. Sogin, William J. Jones, Bruce A. Roe, Jason P. Affourtit, Michael Egholm, Bernard Henrissat, Andrew C. Heath, Rob Knight and Jeffrey I. Gordon. A core gut microbiome in obese and lean twins. In *Nature*, volume 457, issue 7228, pages 480-484. Nature Publishing Group, 2009.

[9] Mauro Castellarin, René L. Warren, J. Douglas Freeman, Lisa Dreolini, Martin Krzywinski, Jaclyn Strauss, Rebecca Barnes, Peter Watson, Emma Allen-Vercoe, Richard A. Moore and Robert A. Holt. Fusobacterium nucleatum infection is prevalent in human colorectal carcinoma. In *Genome Research*, volume 22, pages 299-306. Cold Spring Harbor Laboratory Press, 2011.

[10] Aleksandar D. Kostic, Dirk Gevers, Chandra Sekhar Pedamallu, Monia Michaud, Fujiko Duke, Ashlee M. Earl, Akinyemi I. Ojesina, Joonil Jung, Adam J. Bass, Josep Tabernero,

José Baselga, Chen Liu, Ramesh A. Shivdasani, Shuji Ogino, Bruce W. Birren, Curtis Huttenhower, Wendy S. Garrett and Matthew Meyerson. Genomic analysis identifies association of Fusobacterium with colorectal carcinoma. In *Genome Research*, volume 22, pages 292-298. Cold Srping Harbor Laboratory Press, 2012.

[11] Sarkis K. Mazmanian. Capsular polysaccharides of symbiotic bacteria modulate immune responses during experimental colitis. In *Journal of pediatric gastroenterology and nutrition*, volume 46, supplement 1. Wolters Kluwer, 2004.

[12] Steven L. Salzberg. On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach. In *Data Mining and Knowledge Discovery*, volume 1, pages 317-328. Kluwer Academic Publishers, 1997.

[13] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand and Dan Steinberg. Top 10 algorithms in data mining. Springer, 2007.

[M] Andrew P. Bradley. The use of the Area Under the ROC Curve in the evaluation of machine learning algorithms. Pergamon, 1996.

[14] Igor Kononenko. Machine learning for medical diagnosis: history, state of the art and perspective. Elsevier, 2001.

[15] Stephan Dreiseitl, Lucila Ohno-Machado, Harald Kittler, Staal Vinterbo, Holger Billhardt and Michael Binder. A comparison of machine learning methods for the diagnosis of pigmented skin lesions. In *Journal of Biomedical Informatics*, volume 34, pages 28-36. Ideal Library, 2001.

[16] Kwokleung Chan, Te-Won Lee, Pamela A. Sample, Michael H. Goldbaum, Robert N. Weinreb and Terrence J. Sejnowski. Comparison of machine learning and traditional classifiers in glaucoma diagnosis. In *IEEE Transactions on Biomedical Engineering*, volume 49, number 9. IEEE 2002.

[17] Arie Ben-David. About the relationship between ROC curves and Cohen's kappa. In *Engineering Applications of Artificial Intelligence*, volume 21, issue 6, pages 874-882. Pergamon Press, 2008.

[18] Mark A. Hall. Correlation-based feature selection for machine learning. The University of Waikato, 1999.

[19] Ron Johavi and Geroge H. John. Wrappers for feature subset selection. In *Artificial Intelligence*, volume 97, pages 273-324. Elsevier, 1997.

[20] Manoranjan Dash and Huan Liu. Consistency-based search in feature selection. In *Artificial Intelligence*, volume 151, pages 155-176. Elsevier, 2003.