Bachelor Thesis

Using Approximate Bayesian Computation to Infer the Number of Populations from SNP Genotype Data

Supervisors: Manfred Opper,
Olivier Françios,
Michael Blum

Fabian Bergmann, 372918

Pages: 25

Submission Date: March 18, 2019

Contents

1 Introduction

A central objective in population genetics is to eveluate population structure. Genetical differences of member individuals are analysed to detect systematic genetic similarities and dissamilarites that could indicate the presence of various subpopulations. The definition of a population generally follows the lines of: a population is a "group of organisms of the same species living within a sufficiently restricted geographical area so that any member can potentially mate with any other member of the oppisite sex". Subsequently however, as only a theoretical setting is assumed without any further context, like geography, social hierarchy etc., a population will describe a group of individuals that are associated with one another because their genetic information is sufficiently homogenous to be discriminated as a group from other individuals of the species. An adpted notion of a subpopulation will not be necessary since the described understanding of a population overlaps with that of a subpopulation (each detectable subpopulation would be a new population), also no further hierarchical structure is present in the used theoretical framework that would make further granulation necessary.
An allele is a variant of a gene found at a specific location on a chromosome (locus), and consequently alleles are responsible for the appearance of genetic variation in a species. An allele frequency describes the probability $p(a|k)$ that an individual from population k has the allele a. Since individuals in a population possess similar genotypes, so some alleles are encountered more frequently in these individuals than others and what therefore sets them apart, the allele frequencies sufficiently summarise the population.
The ampleness of information usually found in a genom poses a challenge dimensionality wise if all of it were to be captured. For tasks that solely involve the analysis of genetic variation between individuals and groups of individuals it suffices to rely on the evaluation of a limited amount of genes. Mostly genese are chosen that are known to be subject to genetic variation, these particular genes are called markers. **introduce snp markers now?**
The usual approach to investigate population structure follows the construction of a statistical model according to biological hypotheses which is then evaluated with respect to how well it fits collected genotype data. Many models employ hyperparameters that demand the specifiaction of their values. The hyperparameter values have to be chosen carefully, since for example by merely picking the parameter values that maximise the likelihood of a model could lead to unrealistic values (especially if model assumptions are faulty). In addition, the comparison of models becomes at best tricky if they both propose different values for the same hyperparameters. A tool that estimates realistic hyperparameter values model independently would therfore be of great facilitation to the inference process.
One particular hyperparameter that is frequently required in population genetics is the specifiaction of how many populations are present in the to be analysed data. Subsequently a method will be presented that aims to offer a stable and realistic estimate for the number of populations in given genotype data. Fundamental to the method is the insight that Thereby Usually for this purpose the validity of a certain model is evaluated by analysing the underlying genotype data of the organisms. Models are often susceptible to the realistic choice of hyperparameters, such that for example the likelihood of a model with a certain hyper parameter greater is than for a much more realistc hyper parameter. Since for many models the declaration of the number of populations in the given data is crucial to infer the remaining parameters, a successful and stable technique for determining this parameter would lead to a significant facilitation for the inference process. However, so far no automated solution exists that satisfyingly alleviate this concern.
Throughout the following pages an approximate bayesian computation method will be presented

that intends to deliver a reasonable estimate for the number of populations in SNP genotype data. It is fundamentally based on the insight that the behaviour of the spectrum of a covariance matrix is closely linked to the clustering structure found in the respective data. The eigenvalues are therefore a fruitful choice as summary statistics, in order to decrease the dimensionality in the calculation. The likelihood function for this problem is elusive, therefore a gradient boosting tequnique with decision trees wil be employed to bypass its explicit computation.
Gradient Boosting is a supervised machine learning method, thus it requires a significant amount of data for training and testing. Real world genotype data however is too extensiv to acquire in such a magnitude as is necessary, therefore an artificial data set is simulated and used. The generation of the artificial data follows commonly accepted theories for the simulation of population structure.

1.1 Problem

Given the genotype data of individuals, organised in a matrix X where each row corresponds to an individual, the task is to infer how many populations K are present in X. A population is a "group of organisms of the same species living within a sufficiently restricted geographical area so that any member can potentially mate with any other member of the oppisite sex" (**hartl1997principles**). Especially due to genetic drift, the change of the allele frequencies in a population that occurs because of finite random sampling from the available gene pool, populations are destinguishable in their genetic information, although they might have split from a single population a reasonable amount of generations ago. For further information the reader is referred to (**hartl1997principles**). Therefore, if a sufficient amount of genetic information is used to span the feature space, usually in the form of genetic variations found at genetic markers, individuals should cluster together with other individuals of the same population as their genetic data is more homogenous **add more reasoning???, law of large numbers???**. The number of populations K should accord with K clusters found in the feature space, so the problem simplifies to identifying the number of clusters found in the data matrix X. **give a definition for clusters???, measure of genetic distance???**

1.2 Approaches currently used

A natural approach for problems involving clustering, would be to use a well established method and adapting it to infer the number of clusters, such as by maximising the likelihood of an expactation maximisation in combination with a model quality estimator like the Bayesian Information Criterion to avoid overfitting. In general bayesian approaches, sucha as maximising the likelihood with regard to a theoretical model of the population structure is always a possible approach, however makes the estimation of the number of populations highly dependable on the a-priori assumptions of the model for what determines mathematically a new population **falush2003inference**. Fitting a hierarchical tree model with bayesian methods has also been attempted **corander2004baps**. Different assumptions from different models could lead to different estimations, which would undermine their comparability. Nevertheless, a maximum-likelihood approach was implemented in the software STRUCTURE **pritchard2000inference falush2003inference** and widely applied **rosenberg2002genetic harter2004origin rosenberg2001empirical**. Furthermore, in some cases, especially for data that involves a high number of populations, a very distinctive maximum is not obtained, for the maximum-likelihood function tends to be smoother as higher values are examined **more explanation???**. Some approaches add further heuristics, such as also taking the second order rate of change of the likelihood function into consideration **evanno2005detecting**, which however appears more like mending the performance

of an approach that was solely conceived as preliminary remedy **pritchard2000inference**.
A more "modern" approach involves the insight that a cluster structure is also resembled in a structured form in the spectrum of the respective data matrix. The connection between the spectrum and a matrix was first discovered in graph theory **donath1973lower fiedler1973algebraic** and later introduced into machine learning **shi2000normalized meila2001random ng2002spectral**, for further information see **von2007tutorial**. In general the relevant insight states that: suppose K clusters can be observed in the data matrix X (w.l.o.g. X is a square matrix), then the first $k-1$ eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{k-1}$ are significantly larger than the remaining eigenvalues, also the corresponding eigenvectors **span a subspace that approximates a simplex with the clusters as vertices???**. The number of clusters can thus be inferred by examining the eigenvalues of the data, firstly detected and applied in the context of population genetics by **patterson2006population**.
examples
Approaches have been made with random matrix theory (RMT) to concretise the behaviour of the first $k-1$ eigenevalues **patterson2006population**, including a mathematical threshhold that destinguishes significantly larger eigenvalues from lower ones **bryc2013separation**. Utilising insights from random matrix theory furtherly remains its beginnings, so far no well performing method based on RMT has been developed. **cite ???**

2 Biological Background

ADD MORE !!!

2.0.1 Key words

- **Chromosome:** A DNA molecule that encodes genetic information.
- **Gene:** A DNA (or RNA) sequence that specifies the structure of a particular functional molecule.
- **Locus:** A particular position on the chromosome, like the position of a specific gene.
- **Allele:** A variant form of a given gene. Different alleles can lead to distinct phenotypic traits.

3 Modelling

The genetic information to evaluate individuals is gathered from specific locations in the genom (loci) that can exhibit known genetic variants (alleles). These specific locations are called markers allow for an easier handling of the demensionality and complexity of a genom for certain tasks, such as classefication. To detect properly the magnitude of difference between the genotypes of two individuals from the same species, a sufficient amount of loci that carry genetic variation have to be used. For modelling SNP variations will be used. The biological meaning of the loci generated by the following model for the task of inferring the number of populations is irrelevant, solely necessary is the fact that at each loci different alleles should exist. Each population expresses the alleles at different frequencies, such that, by the law of large numbers, the more loci are simulated the more seperable the populations should be from another and individuals easierly assignable to a population. From a modelling view, it is the different allele

frequencies that define a population.

The allele frequencies of a population k at a locus l for an allele a will be denoted as $p_k(l_a)$. For simplification purposes each loci in the model can be interpreted as having only two alleles. The model assumes a point in time where **no different alleles existed??? or alleles where considered the same???** and overtime mutations introduced new alleles that could asser themselves. However the model does not distinguish between the new alleles, only if an individual carries a mutant variant or not (furthermore ploidity is also ignored). The notation therefore also simplifies for the modell to $p_k(l)$ (probability of having a mutant allele at locus l if from population k).

3.1 Further Model assumptions

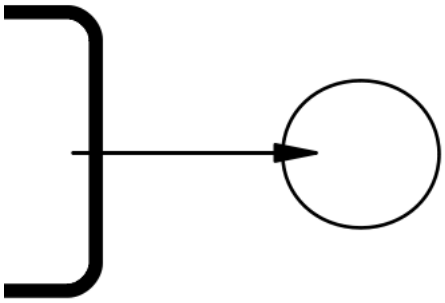
The model assumes several properties:

- Hardy-Weinberg Equilibrium: Relevant to the modelling is that, wiht a Hardy-Weinberg Equilibrium the allele frequencies within a population do not change, the populations are "stationary".
- Linkage equilibrium: Loci are independent from one another.
- **No hierarchical clusters present, like families in a population???**
- **further assumptions ???**

3.2 F-Layer

Distinct populations within a species form mainly due to random genetic drift when there is a fragmentation of the general population into subgroups present. Genetic random drift can be interpreted as a stochastic process where the allele frequencies of a gene change randomly over time. The first modelling atempts applied a markov chain **cite**, where a transition models the change of allele frequencies from one generation to the next. Further refinements have been added over time, such as converting the simplification of discrete populations to the continous case and adding biological idiosyncrasies **cite**. The random change in the stochastic process stems from the fact that a population has a finite amount of members, whose genetic makeup was "drawn" from a pool with probabilites according to the allele frequencies of the population at the time of the conception of the members, therefore it is likely that the allele frequencies in a population are not passed on exactly. Considering the law of large numbers inversely, the smaller a population is the faster genetic drift will make an impact. The allele frequencies change until one allele is able to assert to fixation, meaning no other variants are substantially left and only the one allele is passed on.

All K populations in the model emerged from a single ancestral population A. This is founded on the biological background that for various reasons such as migration, geography, climate, sub popualations form within a single population and as mating between subpopulations decreases or even ceases the populations are effected by distinct random genetic drifts, as a consequence their genetic makeups diverge from another until they can be considered distinct populations. Random mutations and natural selection, if environments are substantially different, in addition support the divergence. **Nonetheless, the fixation of most genes is mostly due to genetic drift???.** As certain alleles becom more dominant in populations the genetic variation declines



and more individuals become homozygous (so less are heterozygous). Using this insight, a divergence statistics that is widely used measures the decrease in heterozygosity between the original population and the newly emerged subpopulation.
Let H_S denote the heterozygosity of the original population and H_T the heterozygosity of the subpopulation. Then

$$F_{ST} = \frac{H_S - H_T}{H_S}$$

gives a percentage by how much heterozygosity decreased.

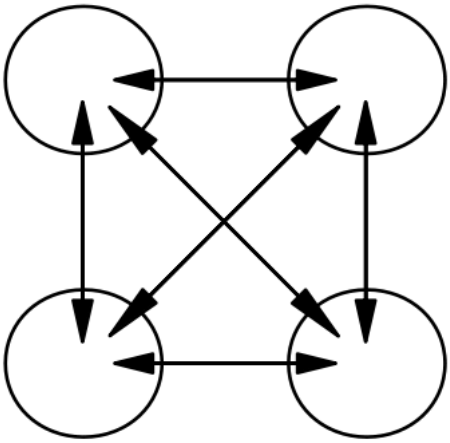
what is the connection to the F-model??? how to get to $\frac{1-F^k}{F^k}$
This divergence measure, or F-statistic, is used analogously to parametarise the genetic drift from the ancestral population, like following citefalush2003inference:

$$Dir(p^A_{l_1}\frac{1-F^k}{F^k}, p^A_{l_2}\frac{1-F^k}{F^k}, \ldots, p^A_{l_{a_l}}\frac{1-F^k}{F^k}) \tag{1}$$

Where $p^A_{l_i}$ is the allele frequency from the ancestral population of the allele i, from a_l alleles, at locus l and F^k is the drift value of population k. At a very low F^k (like 0.05) the fraction $\frac{1-F^k}{F^k}$ is considerably above 1, thus the probability mass is mostly concentrated around ancestral allele frequencies. As the value of F^k increases:

- for a moderate value (like 0.3) the probability mass spreads out further, such that more loci will have different allele frequencies and a divergence to the ancestral population is obervable (for a sufficient amount of loci).
- for a high value (like 0.5) the probability mass concentrates itself at the vertices of the $k-1$ simplex. This corresponds to a high chance of allele fixation. The fixation will be more likely for the allele that was dominant, mathematically because of the multiplication with $p^A_{l_i}$ in the ancestral population, to begin with.

A population is defined by its allele probabilites, whereby the members of the population are approximately homogenous. The allele probalities are a categorical distributions over the possible alleles at each locus, so how often a genetic variant appears in a population. So at principal, to simulate a new population, its allele probabilities have to be determined. The simulated data the modell produces is supposed to resemble SNP data, where no distinction is made between the different variants of a mutation, but solely if a mutation at ll exists compared to an undetermined hypothetical origin population.



The modell has a hierarchical structure starting with allele probabilities that are derived from an unseen ancestral population. Let $p_A(l)$ denote the probability, sampled from a uniform distribution, of an individual from the ancestral population A having a variation at locus l. Subsequently, following the genetic drift modelling from (1) by **falush2003inference**, for each population k an F-value F^k is chosen to introduce the magnitude of genetic drift from the ancestral population. These are used to derive the mutation probabilities of population k at locus l:

$$p_k(l) = beta(p_A(l)\frac{1-F^k}{F^k}, (1-p_A(l))\frac{1-F^k}{F^k}) \tag{2}$$

The dirichlet distribution from (1) degenerates to a beta distribtion as effectively only two alleles are considered for each locus.

Proceeding, the probability values are joined to a vector p_k and then merged with all other K populations to a matrix $\mathbf{F} = [p_1 p_2 \dots p_K]^T$ of size $K \times L$, where L is the number of loci, such that each column of $\mathbf{F}$ gives the probabilities of each population for a specific locus l.

3.3 Admixture Layer

Apart from populations drifting from another apart, individuals of populations also migrate and mate with other individuals of other populations. Another modelling layer, according to the admixture model presented in **pritchard2000inference**, introduces the prospect of modelling their admixed offspring, whereby the flexibility, by determining the hyperparameters of a dirichlet distribution, allows to specify the probabilites of how populations partake to what degree in the admixture.
An admixed individual is defined by possessing a mixture of genetic data from various populations. The mixture is a weighting modeled by a categorical distribution according to the influence of each population on the individual. For an individual i the mixture weights q_i are sampled from a dirichlet distribution with K influencing hyperparameters $j_1, j_2, \dots, j_K$ each corresponding to a population, thus $q_i \sim dir(j_1, j_2, \dots, j_K)$. A non admixed individual j also receives mixing parameters with the difference that the only non-zero value is a one at position

k, indicating that the individual belongs to population k, so $q_j = [0_1, \ldots, 1_k, \ldots, 0_K]^T$. All M individuals are combined to a mixing matrix $\mathbf{Q} = [q_1, q_2, \ldots q_M]^T$ of dimensions $M \times K$.

3.4 Combining the Layers

The mixing weights of each individual are subsequently applied over the mutation probabilites of all populations at each locus, which equals to the multiplication of both established matrices $\mathbf{P} = \mathbf{QF}$. The resulting matrix $\mathbf{P}$ of dimension $N \times L$ holds the mutation probabilities of each individual for each locus. By using each entry of $\mathbf{P}$ to sample a value from a bernoulli distribution, for either an individual has a mutant allele or not, the simulated SNP genotype data for each individual is obtained. Furthermore, ploidity is neglected, so it is assumed the genetic data is aploidic, since **ploidity adds nothing unique for the means of inference. citation needed???**, each loci of an individual requires only one sample from the bernoulli distribution.

3.5 Summary

In summary the generation of a new population setting for which the number of populations K is known proceeds as following:

1. Sample the ancestral allele frequencies $p_A(l) \sim Uniform(0, 1)$
2. Determine the F-values F^k **???**
3. For each of the K populations:
   (i) Sample $p^k(l) \sim beta(p_l^A \frac{1-F^k}{F^k}, (1 - p_l^A)\frac{1-F^k}{F^k})$
   (ii) Combine allele probabilities into matrix $\mathbf{F}$
4. For each individual i:
   (i) Choose admixture coefficients $q_i \sim Dir(\alpha_1, \ldots, \alpha_k)$
   (ii) Combine admixture coefficients into matrix $\mathbf{Q}$
5. Calculate admixture $\mathbf{P} = \mathbf{QF}$
6. Convert each value p of $\mathbf{P}$ by sampling $bernoulli(p)$

3.6 Looking at Admixture

The admixture of an individual is determined by the Dirichlet distribution. The dirichlet distribution is parameterised by K hyperparameters $\alpha_1, \alpha_2, \ldots, \alpha_k$. K corresponds to the desired dimension of the output. The probability density function

$$f(x_1, \ldots, x_K, \alpha_1, \ldots, \alpha_K) = \frac{1}{\Gamma(\sum} \cdots$$

where $\sum_{i=1}^{K} x_i = 1$ and all $x_1 \geq 0$. So the Dirichlet distribution defines a probability density on the $K - 1$-simplex and is therefore a natural choice for sampling admixture coefficients.
Of particular interest for the described modell are the hyperparameters, also called concentration

parameters, as they control the mode and the variance around it. For values $a_i \geq 1$ the distribution has a single mode, whose coordinates at the maximum x is given by **bishop2006pattern**:

$$x_i = \frac{\alpha_i - 1}{\sum_{k=1}^{K} a_k - K}$$

The mode moves therefore more towards those directions, simplex vertices that have a relatively higher valued corresponding hyperparameter compared to the other hyperparameters. In addition, the variance σ, given by

$$\sigma_i = \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(\alpha_0 + 1)}$$

where $\alpha_0 = \sum_{i=1}^{K} \alpha_i$, reaveals that higher values of hyperparameters leads to a decrease of the variance, meaning a higher concentration around the mode.
These two properties can be exploited to control the probability of sampling certain admixture coefficients. Furthermore, by sampling from the same dirichlet distribution one is able to simulate various population scenarios, such as a detached admixed cluster, which would correspond to a mode with high concentration parameters, or a population that experienced migration, which would coincide with a degenerated dirichlet distribution that only has two nonzero, concentration values for the two involved population, which is in the end a beta distribution.

3.7 Analysis

The population centroids given by the population allele frequencies (they are the probabilities used for the bernoulli sampling and thus the mean) construct the vertices of a simplex in which the individuals approximately lie. Outliers are solely due to the natural variance created by sampling at the end from a bernoulli distribution. The probabilities the genetic information from each individual is sampled from, nonetheless always combine to a vector that lies within the simplex, for the probabilities are through the admixture coefficients a linear combination of the population allele frequencies or, in other words, of the simplex vertices. From another perspective, the matrix $\mathbf{F}$ that holds the centroids of the populations as rows, then the matrix maps every vector s from the support of an L dimensional dirichlet distribution accordingly on to the simplex spanned by the centroids (so $\mathbf{F}s$). The matrix $\mathbf{F}$ linearly transforms the L-simplex to the desired population simplex.
For further assessment a mean of quantifying the disimilarity between two individuals is necessary. As a measure of genetic distance between two individuals i and j a natural choice is to use a normalised manhatten distance because the possible values of the genetic information of an individual lies on a lattice of values zeros and ones. Or more concrete, let N be the number of loci used as genetic markers, then $\{0,1\}^N \subseteq \mathbb{R}^N$ is the set containing all possible values for the genetic information of an individual. The measure of genetic distance is

$$D = \frac{1}{N} \sum_{n=1}^{N} |l_n^i - l_n^j|$$

where l_n^i and l_n^j are the values of individual i and j respectively at locus l_n. The normalisation keeps the measure invariant to the number of loci used, as recovering more genetic information should not increse the genetic distance per se. The measure ranges from 0, as two individuals

are genetically similar, to 1, meaning genetic disimilarity.
Suppose two individuals i and j are generated by the described modell, so sampling from a bernoulli distrebution for each loci l with the respective allele frequencies $p_i(l)$ and $p_j(l)$. The expected genetic difference of both individuals then is:

$$\begin{aligned}\mathbb{E}[D] &= \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}[|l_n^i - l_n^j|] \\ &= \frac{1}{N}\sum_{n=1}^{N} p_i(l)(1-p_j(l)) + p_j(l)(1-p_i(l))\end{aligned}$$

The result is intuitive. It is the the empirical mean of sampling diffirent allele values.
For individuals sampled from the same probabilities the expectation equates to:

$$\frac{2}{N}\sum_{n=1}^{N}(p(l) - p(l)^2)$$

the further apart allele frequencies -> not always higher gentic difference. measure bad ?!
By investigating the variance of two individuals sampled from the same allele frequencies a better impression of the cluster (population) density can be obtained. Also the expected severity of individuals lying outside the simplex can be assessed.
But before calculating the variance, for simplicity reasons the expectation of the genetic difference squared is calculated:

$$\begin{aligned}\mathbb{E}[D^2] &= \frac{1}{N^2}\mathbb{E}[(\sum_{n=1}^{N}|l_n^i - l_n^j|)^2] \\ &= \frac{1}{N^2}\sum_{n=1}^{N}\sum_{m=1}^{N}\mathbb{E}[|l_n^i - l_n^j||l_m^i - l_m^j|] \\ &= \frac{1}{N^2}(\sum_{n=1}^{N}\mathbb{E}[|l_n^i - l_n^j|^2] + \sum_{n=1}^{N}\sum_{\substack{m=1\\m\neq n}}^{N}\mathbb{E}[|l_n^i - l_n^j||l_m^i - l_m^j|]) \\ &= \frac{1}{N^2}(\sum_{n=1}^{N}2(p(l_n) - p(l_n)^2) + \sum_{n=1}^{N}\sum_{\substack{m=1\\m\neq n}}^{N}4(p(l_n) - p(l_n)^2)(p(l_m) - p(l_m)^2)\end{aligned}$$

and since

$$\mathbb{E}[D]^2 = \frac{4}{N^2}(\sum_{n=1}^{N}\sum_{m=1}^{N}(p(l_n) - p(l_n)^2)(p(l_m) - p(l_m)^2)$$

The variance simplifies to

$$\begin{aligned} Var(D) &= \mathbb{E}[D^2] - \mathbb{E}[D]^2 \\ &= \frac{2}{N^2}(\sum_{n=1}^{N}(p(l_n) - p(l_n)^2) - 2\sum_{n=1}^{N}(p(l_n) - p(l_n)^2)^2) \\ &= \frac{2}{N^2}(\sum_{n=1}^{N} p(l_n) - 3p(l_n)^2 + 4p(l_n)^3 - 2p(l_n)^4) \end{aligned}$$

This reveals that the variance decreases significantly of the order $\mathcal{O}(N^{-2})$ with the number of loci. Therefore, less severe outliers and better observable clusters are to be expected the more loci are simulated.

3.8 Relationship to LDA

The presented modell is strongly related to a modell more commonly known under the name latent dirichlet allocation (LDA) **blei2003latent**. LDA uses in the setting of natural language processing (NLP) an ensemble of words that are probabilistically associated with certain topics, in order to determine which topics are exhibited by analysed documents, that preferably possess some associated words. Since not all existing words are associated with topics but only a selection, the selected words can be perceived as reasonable indicators of the topic context, as are the used genetic markers to determine population affiliation. A one hot encoding of wether a word is present in a document or not is the respective equivalent of wether an individual possess a gene variant at a genetic maker or not. The encoding of the selected words or respectively of the genetic markers span the feature space in which the topics/populations lie. The topics/populations then span a simplex in which the documents/individuals are mapped according to the admixture.

4 Theory

4.1 Approximate Bayesian Computation

At heart of approximate bayesian computation (ABC) lies the inference of a desired parameter value θ for given data D by relating the conditional probability of that data given the parameter value $P(D|\theta)$ to the symmetric counter part, the conditional probability of the parameter given the data $P(\theta|D)$. This is done by exploiting Bayes' rule:

$$P(\theta|D) = \frac{p(D|\theta)P(\theta)}{P(D)}$$

Where $P(D|\theta)$ is often called the likelihood, $P(\theta|D)$ the posterior, $P(\theta)$ the prior and $P(D)$ the evidence which is used in Bayes' rule solely for nomalisation porpuses.
For many problems the problem space is intractable or dimensionally too large to compute the likelihood. ABC intends to circumvent these problems.

4.1.1 Rejection Algorithm

The rejection algorithm employs a basic approach for finding the posterior distribution of the desired parameter θ for specific data D. Given a known prior distribution of $theta$, the algorithm samples values $\hat{\theta}$ from the prior and then inputs $\hat{\theta}$ into an appropriate model to simulate some data $\hat{D}$. If the simulated data lies with in a margin of error $\epsilon \geq 0$ from data D for a chosen metric ρ , so $\rho(D, \hat{D}) \leq \epsilon$, then the sampled prior value $\hat{\theta}$ is accepted by adding it to the final sample of parameter values for θ. The final sample should approximate the desired posterior. For further infromation and possible refinements such as using linear regression to counter the error at acceptance or using Sequential Monte Carlo - ABC to sample from areas with higher posterior density the reader is referred to **csillery2010approximate**.
If the task is for example to compare different models concerning a specific data set, a variant of the rejection algorithm usually poses a reasonable choice. For tasks that only require a good point estimate of θ that fits well to the data D, like the maximum a posteriori (MAP), the rejection algorithm might be too elaborate. Also, the algorithm does not generalise over all possible data sets, meaning it has to be rerun each time for other data, which could be computationally costly. Furthermore, the explicit construction of a model that simulates data introduces assumptions into the computation, as this is a further approximation it is potentially problematic for the decency of the approximation. **use instead implicit models like GAN???**

4.1.2 In Context of Supervised Learning

In contrast to the rejection algorithm, supervised learning techniques, such as neural networks or boosting decision trees, address all the problems stated above. A supervised learning method attepmts to find a general connection between any given input data D and its desired output θ by training a malleable model. The model is instructed to infer the general connection by adapting itself in such a way that it minimises the emperical risk (for some chosen loss function) when solving a finite training set of size N, which is a data set with "presolved" values $((D_1, \theta_1), \ldots (D_N, \theta_N))$. From another perspective, a supervised learning algorithm attempts to forge a model in such a way that it perfects the approximation of the **problem space???** and thus returns good point estimates for θ given any input data D. **what about the posterior, are the returned values the MAP???**. The prior $P(\theta)$ can be implicitly adjusted by changing the proportions of θ in the training data.
The quality and size of the training set partakes hugely in achieving good results for supervised learning methods. **By generating synthetic data it is possible to easily construct a plenteous data set, however this again introduces prior assumptions into the learning process and undermines the goal of generalisation. A measure to counteract this problem is to rely on summary statistics that focus on the relevant statistics for the task at hand.??? regularisation through insufficient statistics???**

4.1.3 Summary Statistics

Large dimensionality of a data set can undermine the practability of an ABC-method. By summarising the data one attempts to reduce dimensionality, while still sustaining a good approximation of the posterior. So if $S(D)$ is a summary statistics of some data D then the acceptance criterion for the rejection algorithm converts to $\rho(S(D), S(\hat{D})) \leq \epsilon$, whereby $P(\theta|D) \approx P(\theta|S(D))$ holds sufficiently. The use of summary statistics is not confined to the rejection algorithm, rather it is a general tool that allows for a tradeoff between reduction of

dimensionality and the goodness of the approximation, since each summarisation usually forfeits some of the principal information. If no information is lost, so $P(\theta|D) = P(\theta|S(D))$ applies, then the summary statistics is called sufficient. **However, only exponential families have finite sufficient summary statistics, for they are maximum entropy distributions???**. A good informative choice of summary statistics is highly task and data set dependent **nunes2010optimal**. An overview of common heuristics and algorithms for choosing summary statistics can be found in **blum2013comparative**.
To infer the number of populations K expressed in a given dataset X the conditional probability $P(K|X)$ with respect to K is maximised. Since the large dimensionalities of the used datasets pose substantial computational difficulties, the datasets are summarised in an effective manner, such that the approximation $P(K|X) \approx P(K|sum(X))$ is sufficient for the intended inference. Bayes' theorem then yields

$$P(K|sum(X)) = \frac{P(sum(X)|K)P(K)}{P(sum(X))}$$

The calculation of the likelihood $P(sum(X)|K)$ however is intractable because **???** To circumvent this problem the likelihood is implicitly calculated by employing a supervised learning method to estimate the posterior, such as a neural network or boosting decision trees. These methods are trained by trying to link summary statistics of datasets to the corresponding values for the number of populations. The prior $P(K)$ can be implicitly adjusted by changing the proportions of K in the training data. Gradient boosting with decision trees is chosen in this case, for it has demonstrated good results for various classification problems citation needed. **??? Furthermore, some inuitive reasoning exists, as explained later on, for the use of decision trees in this particular case.**

4.2 Choosing the Summary Statistics

The choice of adequate summary statistics is essential to obtain significant results. Large dimensional data often times demands it to be summarised, so the intended methods a reasonably applicable. In doing so, the manner summary is of great importance because each summarisation usually forfeits some of the principal information. So one is confronted with the problem of how to effectively manage the trade off between the practicability the method and the loss of information that could endager desired results.
The entropy of a distribution measures the existing uncertainty about which event appears if one samples from the distribution. Mathematically it is defined for a given continous probability mass function $P(X)$ as

$$h(x) = -\int_{supp(P)} P(x)\log(P(x))dx$$

The principal of maximum entropy states that given some prior information about the underlying probability distribution, such as already drawn samples or a constraining property, the maximum entropy distribution that incorporates the prior information is the best distribution to respect the remaining uncertainty **jaynes1957information**. In other words, the maximum entropy distribution is the best distribution to fit the already obtained information if no further assumptions are to be added.
For a given mean μ and covariance Σ the multivariate continuous distribution that maximises the entropy is the multivariate Gaussian, for a proof the reader is referred to **cover2012elements**. The entropy of the multivariate Gaussian is derived as following:

$$
\begin{aligned}
h(x) &= -\int_{-\infty}^{\infty} N(x|\mu,\Sigma)\ln(N(x|\mu,\Sigma))dx \\
&= E[\ln(N(x|\mu,\Sigma)] \\
&= E[\ln(det(2\pi\Sigma)^{-\frac{1}{2}}e^{-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)})] \\
&= \frac{1}{2}\ln(det(2\pi\Sigma)) + \frac{1}{2}E[(x-\mu)^T\Sigma^{-1}(x-\mu)] \\
&= \frac{1}{2}\ln(det(2\pi\Sigma)) + \frac{1}{2}E[trace(\Sigma^{-1}(x-\mu)^T(x-\mu))] \\
&= \frac{1}{2}\ln(det(2\pi\Sigma)) + \frac{1}{2}E[trace(I)] \\
&= \frac{1}{2}\ln(det(2\pi e\Sigma))
\end{aligned}
$$

The only non-constant factor influencing the entropy of a multivariate gaussian is the determinant of the respective covaraince matrix. Since any real symmetric matrix is diagonalisable, $det(\Sigma)$ breaks down to $det(\Sigma) = det(\mathbf{Q}^{-1}) \cdot det(\mathbf{\Lambda}) \cdot det(\mathbf{Q}) = \prod_{i=1} \lambda_i$, thus revealing that actually the eigenvalues of the covariance matrix are responsible for the magnitude of the entropy. In conclusion, by summarising the data by its covariance matrix, one implicitly approximates the data as being guassian and secondly it suffices for the summary to only take the eigenvalues into consideration.
Assume Data is linearly correlated??? connection to PCA???

4.2.1 PCA

add introduction??? Principle component analysis is a statistical method that performs a basis transformation on a given data set, such that no linear correlations are anymore present in the data. Since the direction of a linear correlation corresponds to the direction of the highest variance in a concerning subspace, a new axis musst be aligned according that particular direction. **citation needed ???** This construction of the axes is done by requiering in an iterative manner each new axis to align with the direction that captures the most variance in the data, which however has not been captured by already established axes.
The differences between data points makes them destingiushable, but also determines the magnitude of the observed variance. Decreasing the variance in a data set by projecting it into a subspace, thus endagers destinguishability (in certain features or even in total), which is information. The amount of sustained variance after projecting the data into a subspace can therefore act as an indicator for how much information was retained. So by always maximizing the captured variance of a newly added axis to the transformation, which is a subspace of the principal data set, the highest possible amount of information is retained for a projection into a subspace with a particular rank K (under the assumption that variance corresponds to information). The subspace is spanned by the K largest Eigenvectors **singular values ???** of the empirical covariance matrix, as is subsequently shown.
Let $S = \frac{1}{N}XX^T - \overline{X}\overline{X}^T$ denote the emperical covariance matrix of the data matrix X. Then the expression u^TSu is the emprical variance of u^TX, which is the data X projected on to the vector u.

$$\begin{aligned} u^T S u &= \frac{1}{N} u^T X X^T u - u^T \overline{XX}^T u \\ &= \frac{1}{N} u^T X (u^T X)^T - \overline{u^T X} (\overline{u^T X})^T \\ &= \frac{1}{N} (u^T X)^2 - (\overline{u^T X})^2 \end{aligned}$$

The empirical variance is maximised with the restriction $\| x \| = 1$ because u is supposed to be part of a new standard basis. **mention orthogonality ???**. By using a lagrange multiplier to add this restriction, the equation $\max_u u^T \Sigma u - \lambda(u^T u - 1)$ is obtained.

$$\begin{aligned} \frac{d}{du}(u^T \Sigma u - \lambda(u^T u - 1)) &= 0 \\ \Sigma u - \lambda u &= 0 \\ \Sigma u &= \lambda u \end{aligned}$$

The solution coincides with the definition of the Eigenvectors, where λ is the eigenvalue of u. Since u should be maximised, the overall solution is the eigenvector belonging to the largest eigenvalue.
Add explanation of SVD??

4.2.2 In context to clustering

Cluster analysis intends to group similar data points together. **what are the clustering assumptions??? distribution? group criteria? bounderies?** Data that exhibits reasonable clustering possess a considerably unique structure. This structure also reveals itself to some degree in the orientation of the eigenvectors and the magnitude of the corresponding eigenvalues, such that they can be utilised to infer certain properties of the data, like the number of clusters as is the current intention.
Intuitively, for inferring the number of clusters, it is assumable **cluster assumptions** in a reasonable setting, including for example that each cluster has a at least a minimal amount of members, that the in-between variance between two distinct clusters is significantly greater than the variance whith-in a particular cluster. The in-between variance constitutes itself through the variance of the with-in variance of both concerning clusters and the distance between the clusters (under the assumption that outliers are possible, so cluster membership is not compulsive). While the with-in variance of a cluster is solely confined to the space assigned to that cluster, which concludes to a significantly smaller variance, espacially considering that the distance of each data point to the mean has a squared impact on the variance (definition of variance). For these reasons a new principal component will orient itself in such a way, that it effectively captures the remaining in-between variance of the clusters.
For K many different population clusters $K-1$ significant PCs are obtained, thus allowing an inference of the number of populations. Considering only two clusters, a single significant PC would be observed that is oriented along a line connecting the two centroids of each cluster, as this would maximise the distance of the clusters after a projection on the PC and therefore maximise the variance after projection. In a general setting with K clusters, the PCs would arrange themselves as linear combinations of the in-between variances, as the overall variance is maximised so all in-betwenn variances are taken into account. Since every cluster participates in $K-1$ in-between variances and capturing these in-between variances corresponds to

determining the **exact??? (the centroids???** relative positions of the other cluster, a linear combination of exactly $K-1$ vectors are needed to locate the relative positions of the other clusters **proof necessary??? less: would mean a cluster is admixed, admixture = linear combination of existing clusters $\rightarrow$ contradiction | more: some in-between variance was not captured - contradiction to maximising variance, sufficient???. The centroids lie in the span of the first $K-1$ PCs ???**
The past mentions of population clusters solely referred to clusters that are not admixed. The introduction of admixed population clusters, however does not alter the previously established theory under certain assumptions. Admixed clusters are several individuals with similar admixed genotypes and are sampled from allele probability values that are subject to a mixture weighting of the allele probability values of the non-admixed populations according to their involvement in the admixture as done in the modell (2). This is simply a linear combination, restricted to the coefficients being proportions (summing to one), of the centroids of the other populations, meaning an admixed individual and therefore the centroid of an admixed cluster lie also in the span of the $K-1$ first PCs spanning the non-admixed populations. In general, the centroids of the non-admixed populations constitute the corners of a simplex, that determines if a population cluster is admixed, thus the influence of admixed individuals concerning the maximisation of the variance is neglegible. **further explanation needed???**.

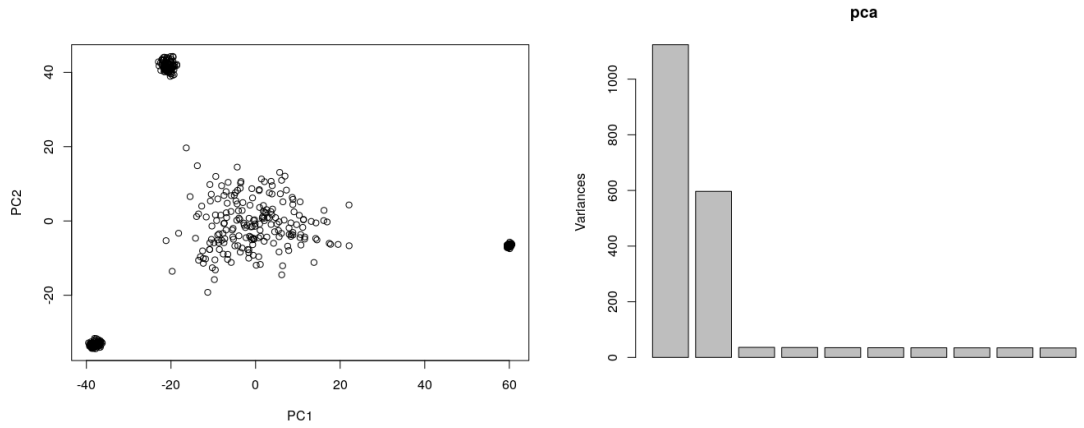
4.3 Examples

A synthetic problem instance generated by the modell, could look like shown in Figure 1, where three populations that span the simplex are observable. The populations have fairly distinct F-values (meaning they drifted away from the ancestral poulation at quite different magnitudes), therefore the clusters are well seperable from one another. Within the simplex is a cluster of several admixed individuals located. They were all sampled from the same dirichlet distribution $Dir(8,8,8)$, with uniform hyperparameters, so they are concentrated around a central mode and all populations participate on average the same amount to the admixture.
Just by looking at the corresponding biggest eigenvalues, it is fairly easy, with the use of the previously insights, to infer the number of populations. The first two eigenvalues are significantly larger than the rest, thus the number of populations should be three.
Figure 2 shows a similar scenario as Figure 1, whereby the only difference consists of a different admixture of the admixed individuals. In this example admixed individuals are sampled from three different dirichlet distributions. In turn one of the hyperparameters is set to zero, thus always on population does not partake in the admixture of an individual. As a consequence the individuals are spread along the edges of the simplex, also because the non-zero hyperparametes model a lower concentration than in Figure 1.
Again the eigenvalues feature two significant large eigenvalues, making the inference of the number of populations using solely the eigenvalues once again a simple task. The natural variance of the descretisation of the allele values through the bernoulli distribution, which would allow individuals to lie outside the simplex, only has a neglectable marginal effect on the eigenvalues.

4.3.1 Difficulties

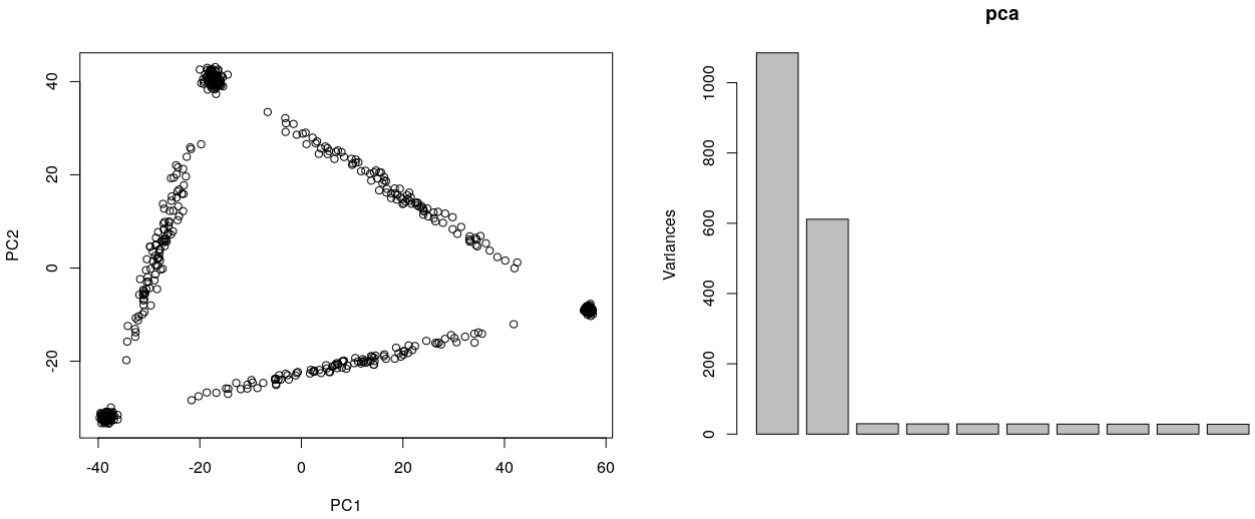
The past examples only demonstrate fairly simple problem instances, for which even a human recognition capabilities suffice. It is also possible to construct more difficult problem instances. Figure 3 is an example of such. The ploy is to simulate a setting where most of the variance is already captured by less than $k-1$, hence making it more difficult to recognise the cut-off for

Figure 1: Projection of three populations and one admixed on to the firs two PCs and corresponding eigenvalues



Three populations with F-values of $0.1, 0.5, 0.9$ and 100 individuals each were sampled. The admixed population derived the proportional weightings for its allele probabilites by sampling for each allele from a $Dir(8, 8, 8)$ distribution. 200 individuals were sampled for the admixed. For this simulation 10000 loci were simulated.

Figure 2: Projection of three populations and one admixed on to the firs two PCs



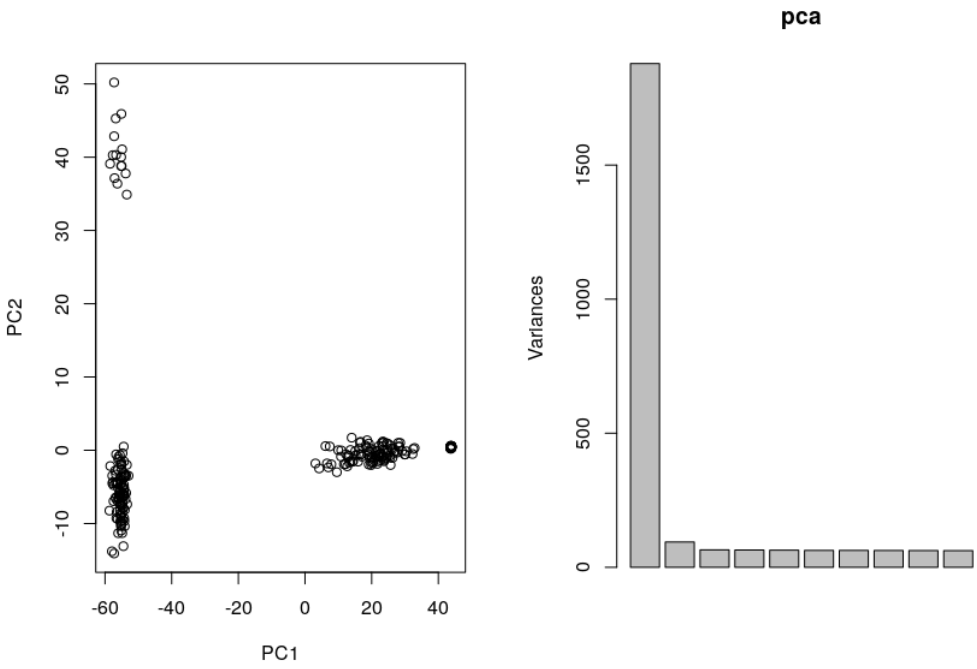
Three populations with F-values of $0.1, 0.5, 0.9$ and with 100 individuals each were sampled. Between each pair of population clusters lies an admixed population sampled from a dirichlet with 5s and a 0 for not involved populations, which corresponds to a $beta(5, 5)$. Each admixed cluster holds 200 individuals. The simulation used again 10000 loci.

significant and insignificant eigenvalues. In figure 3 there are two populations that have similar allele frequencies (F-values 0.05 and 0.1) and one population that has been subject to strong genetic drift and therefore is genetically completely distinct (F-value 0.99). Furthermore, most individuals are distributed over one of the two close populations and the one very far apart. This introduces high a high incentive for the eigenvalues to try to capture the variance of the

individuals of the two populations, since a bigger distance accounts for higher variance. Inversesly, because the population that holds a fewer amount of individuals is not very far away from the positioning of the first eigenvector, such that the second eigenvector, which has to orient itself perpendicular to the first, does not capture very much variance. Also the low amount of individuals in the smaller population makes the orientation of the second eigenvector susceptible to the variance of the other populations or to any outliers. An admixed population also resides between the two greater populations, giving the first eigenvalue even more weight.

The graph with the biggest eigenvalues reveals the described dilemma. The first eigenvector accounts for almost all of the variance between the populations, rendering the other significant eigenvalue almost insubstantial and undiscernible from the other insignificant eigenvalues. In different scenarios the distance between the populations and the distribution of the individuals over the populations could be even more disadvantageous (although the question could more be of the nature to decide what is considerable to be a population, **this is part of the discussion???**). In addition, the situation becomes even more difficult if even more populations are simulated with "extreme" F-values.

Figure 3: Example of a difficult case



Three populations with F-values of $0.05, 0.01, 0.99$. The first population with the smallest F-value has 15 members, while the others have a 100 each. The mixture proportions of the admixed population were sampled from a $Dir(0, 10, 30)$. 10000 loci were simulated.

4.4 RMT

Let $\mathbf{A} = \frac{1}{n}\mathbf{X}\mathbf{X}^T$ be the empirical covariance matrix of $\mathbf{X}$ with $\mathbf{X}$ being an $m \times n$ matrix. $\lambda_1, \lambda_2, \ldots, \lambda_m$ are the corresponding eigenvalues of $\mathbf{A}$. The empirical spectral distribution (**ESD**) for $\mathbf{M}$ is then given by:

$$F^M(x) = \frac{1}{m} \mid \{\lambda_i \leq x \mid i \leq m\} \mid$$

Whereby $\mid \cdot \mid$ denotes the size of a set.

By assuming a theoretical setting in which $m, n \to \infty$ while $y = \frac{m}{n} \to (0, \infty)$ the Marchenko-Pastur Law extends the ESD to the continuous case.
Under the assumption that the entries of $\mathbf{X}$ are random variables iid distributed with mean 0, it states that the probability density of the eigenvalues is given by:

$$p^M(x) = \frac{1}{2\pi x y \sigma^2}\sqrt{(\rho_+ - x)(x - \rho_-)}$$

where $\rho_\pm = \sigma^2(1 \pm \sqrt{y})^2$ and σ^2 is the variance of the random variables.
Insert Plot of distribution

5 Boosting Decision Trees

Subsequently a concise overview of gradient boosting and decision trees is presented. For further details and idiosyncrasies of the methods the reader should consult more elaborate literature, like **trevor2009elements**.

5.1 Gradient Boosting

Gradient boosting is a supervised learning method for classification and regression, that iteratively adds basis learners to a linear combination to reduce an arbitrary differentiable loss function.

Let χ denote the input space. The task then is to approximate the function $f^*(x)$ that maps an arbitrary input $x \in \chi$ to the desired output $y \in \mathbb{R}$. An ensemble of M different basis learners $g_1, g_2, \ldots, g_M : \chi \to \mathbb{R}$ can be used to generalise over a training set $((x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N))$ of N training pairs in a linear fashion as following:

$$f(x) = \sum_{m=1}^{M} \phi_m g_m(x)$$

where $\phi_i \in \mathbb{R}$ are the weights for each basis learner. Linear models for a set of given basis learners can be fitted via konventionell methods such as least squares, lasso, ridge **bishop2006pattern**. $f(x)$ can be used as an approximation for various tasks like regression, classification (by for example using a threshhold),

The model is limited by the explicit choice of the basis learners. Instead an algorithm that finds the best base learners from a hypothesis space $\mathcal{H}$ would increase the adaptive potential of the model. Thus, for a given differentiable loss function $l(x, f(x))$, an algorithm should find the best M base learners $h_1, h_2, \ldots, h_m \in \mathcal{H} : \chi \to \mathbb{R}$ and corresponding weights such that the empirical risk is minimised:

$$\underset{\phi_i, h_i}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} l(y_i, \sum_{m=1}^{M} \phi_m h_m(x_i)) \tag{3}$$

Solving this task is an optimisation problem usually beyond practicability. A common optimisation technique is too update the parameters through the use of gradient descent. Several problems arise in the case of differentiating over the hypothesis space:

- Firstly, the hypothesis space would need to be parameterised with finite dimensions. In the face of a countless manifold of possible base learners a rather daunting task. One is far better off by confining the possibilities of base learners, by choosing a subset $H \subset \mathcal{H}$ like decision trees or neural networks.
- Many modelling frameworks such as neural networks (e.g. number of hidden layers) and decision trees (e.g. tree depth) have to be parametrised at least partly discretely. Of course all discrete model features could be fixed to a constant, but that would greatly degrade the modeling capabilities.
- Lastly, some base learners do not model a differentiable functions. For e.g. decision trees model functions that possess jump discontinuouties.

Gradient boosting algorithms, firstly developed and described by **freund1997decision**; **friedman2001greedy**; **friedman2002stochastic**, circumvent the necessity of differentiability by growing the ensemble of base learners iteratively that minimises the empirical risk with respect to so called pseudo residuals.

The backbone of an gradient boosting algorithm is constituted by forward stagewise additive modeling, which works as following: Let $H \subset \mathcal{H}$ be the chosen set of possible base learners.

1. Initialise with constant like $f_0(x) = 0$
2. For each stage $m \in 1, \dots, M$:
   (i) solve
   $$\underset{\phi_i, h_i}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} l(y_i, f_{m-1}(x_i) + \phi_m h_m(x_i))$$
   (ii) Set $f_m(x) = f_{m-1}(x) + \phi_m h_m(x)$

The optimisation step, although the amount of parameters is reduced compared to (3), only possess closed viable solving techniques for a limited amount of loss functions, like L2-loss or exponential loss. More information is provided in **friedman2000additive**. To expand the optimisation step to any arbitrary differentiable loss function, the numerical optimisation via gradient boosting is used.

The optimisation procedure fixates at stage m the current estimates made by $f_{m-1}(x)$ of the training data in a vector $\mathbf{f}_m = [f_{m-1}(x_1), f_{m-1}(x_2), \dots f_{m-1}(x_N)]^T$.

The loss function then can be reformulated as:

$$L(\mathbf{f}) = \sum_{i=1}^{N} l(y_i, \mathbf{f}_i)$$

Then the gradient of the loss function is calculated w.r.t $\mathbf{f}$.

$$\begin{aligned} \hat{h}_m &= \nabla_{\mathbf{f}_m} L(\mathbf{f}_m) \\ &= [\partial_{\mathbf{f}_1} l(y_1, \mathbf{f}_1), \dots, \partial_{\mathbf{f}_N} l(y_N, \mathbf{f}_N)]^T \end{aligned}$$

Like in conventional gradient descent algorithms the model is adjusted in a manner that the empirical risk is minimsed along the direction of steepest descent. The direction of steepest descent corresponds to the negative of the gradient, which is $-\hat{h}_m$. Base learners that output differentiable function approximations can express the gradient via chainrule through the adjustable parameters of the model itself (like the weights in a neural network) and hence these are reciprocally changed to minimise the loss. Other base learners on the other hand have to resort to a different strategy.
The embedding in the forward additive model allows for the negative gradient $-\hat{h}_m$ to be approximated directly by a base learner. The objectiv of the new base learner $h_m(x)$ consequently concludes therein to approximate

$$h_m(x) \approx -\hat{h}_m$$

as well as possible. A possible approach would be to train a base learner on the training data, but where the labels are exchanged by $-\hat{h}_m$.
Intuitively, a base learner $h_m(x)$ should be considered as an approximate step in the direction of steepest descent of the emirical risk. Following this setup, the weight $\phi_m > 0$ can be considered as the corresponding step size to be adjusted to ones taste. As a conclusion the iterative construction of the final linear model with gradient boosting

$$f(x) = \sum_{m=1}^{M} f_{m-1}(x) + \phi_m h_m(x)$$

is a sequence of gradient descent steps towards a minimum of the empirical risk.

5.2 Decision Trees

Typical decision trees are a supervised learning method that solve a regression or classification problem by segmenting the feature space in to distinct regions, whereby all data points lying in the same region are assigned the same value by the tree.

Let $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ be the training data. The input dimension of a point x_i is D.

Suppose a tree T partitions the featurespace $X_1, X_2, \ldots, X_D$ into M regions $R_1, R_2, \ldots, R_M$. The response function outputed by T is given by:

$$f_T(x) = \sum_{m=1}^{M} c_m I(x \in R_m)$$

where I is an indicator function signaling if the input x is part of a particular region. c_m is the response for a region R_m. The optimal value of c_m depends on a chosen loss function that should be minimsed w.r.t c_m over all the training points assigned to region R_m. As an example, for square loss in a regression setting this would accord to the empirical average of the label values from the training points lying in the region R_m.

A far more challenging optimisation task is finding the optimal tree that partitions the feature space into M regions, for a given loss function l.

$$\underset{R_m, c_m}{\operatorname{argmin}} \sum_{i=1}^{N} l(y_i, \sum_{m=1}^{M} c_m I(x_i \in R_m))$$

The possibilities of partitioning feature space grows exponentially with the number of features, rendering the encounter of an optimal tree in most cases computationally infeasable.

As an alternative, a greedy approximation approach can be used (CART add citation). The greedy algorithm chooses the best dimension X_d and splitting point s that minimises the loss for the two new regions that arise (w.l.o.g. only binary trees will be considered). The two new regions are defined as:

$$R_1(d, s) = \{X | X_d \leq s\} \qquad R_2(d, s) = \{X | X_d > s\}$$

The feature space is divided by a plane that is orthogonal to the axis corresponding to feature X_j. The plane cuts the value of s on that axis.
The optimisation objective thus reduces to the choice of positioning the best partitioning plane orthogonal to an axis:

$$\underset{s,d}{\operatorname{argmin}}[\underset{c_1}{\operatorname{argmin}} \sum_{x_i \in R_1(d,s)} l(y_i, x_i) + \underset{c_2}{\operatorname{argmin}} \sum_{x_j \in R_2(d,s)} l(y_j, x_j)]$$

The optimisation step yields, for a naive implementation that checks for every dimension the splitting value of a plane between every two neighbouring data points, a worstcase runningtime of $\mathcal{O}(D \cdot Nlog(N))$ ($Nlog(N)$ for sorting feature values), which is computationally much more feasable. However between two neighbouring data points x_1 and x_2 there are infinitely many positions to place the splitting plane that evaluate to the same empirical risk. As a convention the splitting value s that lies in the middle of x_1 and x_2, when they are projected on the axis X_d, is chosen by convention. The reasoning is that no further assumptions are to be added through the positioning of the plane and since the splitting of the feature space can be viewed as a bernoulli experiment (data point either lies left or right of plane) the bernoulli distribution with expected value of 0.5 is the maximum entropy bernoulli.

After finding the best split, the procedure is repeated on the newly constructed regions until the desired depth of the tree is reached. Regression and classification trees differ from one another only through the selection of a different loss function. For regression standard loss functions like square loss would be reasonable choices. Classification trees have revealed better results for loss function that reward node purity, so to which degree does a node hold data points of a single class. Such loss functions are for example the gini index or cross entropy loss for classification. From an information theory perspective, node purity leads to a greater reduction in entropy.

Growing a too big tree T_0 is susceptible to overfitting. A regularisation technique that aids in the construction of a well generalising tree is cost-complexity pruning. The goal of pruning is to find a sub-tree $T \subseteq T_0$, by conflating all hierarchically lower nodes that lead off an internal node into that internal nodes, that minimises:

$$C_\alpha(T) = \sum_{i=1}^{N} l(y_i, f_T(x_i)) + \alpha|T|$$

where $f_T(x)$ is the estimate of Tree T for input x and $|T|$ denotes the number of terminal nodes of T. The tuning parameter $\alpha \geq 0$ punishes larger more complex trees according to its value. It resembles the regularisation term of ridge regression.
The optimal sub-tree corresponding to a particular tuning parameter T_α can be found using weakest link pruning. Weakest link pruning conflates those terminal nodes into an internal node to which the terminal nodes are all adjacent, such that the empirical risk increases minimally. Continuing with this procedure until only the root stub remains gives a sequence of subtrees $T_1, T_2, \ldots, T_n$ in which with a probability of 1 the optimal subtree T_α can be found **breiman1984classification**. Cross validation can be used to find the optimal value for α.

Decision trees exhibit high variance, meaning that completely different splits occur and thus the output prediction rules change cosiderably when there are minor changes added to the training data. The reason resides inherently with how the splits are chosen in a greedy fashion. For

example, two promising features could reduce the loss almost equally much, but just the best of both is considered for the next split. Adjusting the training data slightly by for example adding new data points could possibly change the value of the loss function enough to choose the other feature for a split the hierarchical nature consequently propagates the difference further down the tree. This behaviour reveals that the confinement to the greedy perspective when constructing a tree to some degree neglects the goal of generalisation in return for tractability.

Remedies that address model stability involve the introduction of bias. The bias-variance tradeoff possess an eminent role in machine learning as its a principle that is prominent for many models. In summary, it describes the forfeit of expected accuracy in return for decreasing the variance of an estimated parameter when the training sample is being varied. Creating an ensemble of decision trees is a widely used and fruitful approach. Ensemble approaches for decision trees include bagging, random forests and as well gradient boosting. Several further refinements improve the quality of an ensemble tree models, including randomly masking different features and training data entries for each tree, as this generates differing trees that place their splits differently and therefore deliver more uncorrelated predictions.

6 Results

7 Discussion