Model Correlation in Random Forests

Master Thesis

presented by

Benjamin Moser

at the

University of Konstanz

Department of Computer and Information Science

Konstanz, 2023

Abstract

...

Contents

1. Introduction

1.1. Overview

1.2. Contributions

1.3. Statistics

We want to reason in a general manner about algorithms which act on data. We use statistical language for this. Consider a data point X that is provided as input to an algorithm. Our intention is to say that X could be essentially *any* data point from some kind of data source. In other words, we can consider X to be *random*, that is, X is a random variable (a symbol) that can take different values. Which values exactly it can take depends on the data source – the *distribution* of the data. If our data source is a fair 6-sided die, X can take values in $\Omega = \{1, \ldots, 6\}$ and the distribution of values is constant where each outcome has probability $\frac{1}{6}$, i.e. $\mathbb{P}[1] = \cdots = \mathbb{P}[6] = \frac{1}{6}$.

A σ-algebra over Ω can be thought of a set of subsets of Ω, containing Σ, such that it is closed under complement and countable union and intersection.

Definition 1.3.1 *A probability space is a triple $(\Omega, \Sigma, \mathbb{P})$ where*

- *Ω is an arbitrary set modelling the sample space i.e. the set of all possible outcomes.*
- *Σ is a σ-algebra over Ω, modelling the set of events.*
- *$\mathbb{P}$ is a function $\Sigma \to [0, 1]$ such that $\mathbb{P}(\Omega) = 1$ and $\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i)$ for a countable collection of (pairwise disjoint) sets in Σ, and models the probability measure.*

In the following, we assume an underlying probability space implicitly.

Definition 1.3.2 *A random variable is a quantity that depends on a random event, i.e. a function $\Omega \to M$ (commonly, we have $M = \mathbb{R}$).*

Further, we not only want to reason about the behaviour of an algorithm with respect to one point X but many such points. A basic notion is the *expected value* of X. Further, since X is considered random, $f(X)$ is also a random variable and we can talk about the expected value of functions of random variables.

Definition 1.3.3 *The probability density function f_X of a random variable X is a nonnegative function such that*

$$\mathbb{P}(a < X < b) = \int_a^b f_X(x)dx$$

Definition 1.3.4 *The expected value (expectation) of random variable X is defined as*

$$\mathbb{E}[X] := \int x \cdot f_X(x)\, dx$$

where the integral is over the support of X.

A function $g(X)$ of a random variable X is a random variable itself and $\mathbb{E}[g(X)] = \int g(x) f_X(x)\, dx$. To emphasize the random variable the function is dependent on, we sometimes mention it in the subscript and write $\mathbb{E}_X[g(X)]$.

Lemma 1.3.1 *(Linearity of Expectations) A basic property of expected values is that they are linear: For any two random variables X, Y and a constant α it holds that $\mathbb{E}[X+Y] = \mathbb{E}[X] + \mathbb{E}[Y]$ and $\mathbb{E}[\alpha X] = \alpha \mathbb{E}[X]$.*

Lemma 1.3.2 *(Law of total expectation) For a function g:*

$$\mathbb{E}_{(X,Y)}[g(X,Y)] = \mathbb{E}_X[\mathbb{E}_Y[g(X,Y) \mid X]]$$

As a special case of this, it holds that

$$X, Y \text{ independent} \quad \rightarrow \quad \mathbb{E}_X[\mathbb{E}_Y[g(X,Y)]] = \mathbb{E}_Y[\mathbb{E}_X[g(X,Y)]]$$

and we write $\mathbb{E}_{X,Y}[g(X,Y)] =_{def} \mathbb{E}_X[\mathbb{E}_Y[g(X,Y)]]$

A random variable is *discrete* if its set of outcomes is a countable set $\{x_i, \ldots, x_n\}$ with probabilities $\{p_i, \ldots, p_n\}$. The probability density function then is $f_X(x_i) = \mathbb{P}(X = x_i) = p_i$. The expected value of a discrete random variable is given by a sum.

$$\mathbb{E}[X] = \sum p_i x_i$$

The expected value is a statement depending on the entire distribution. Usually, we do not know the distribution itself, but only have a limited set of samples of it. For instance, we may have a certain number of data points or run an algorithm a certain number of times and observe its output. We can use this set of samples to approximate the value of the expectation. Given that the samples are independent and identically distributed, the arithmetic mean of samples approximates the expected value.

$$\frac{1}{n} \sum_{i=1}^{n} x_i \;\rightarrow\; \mathbb{E}[X] \qquad \text{as } n \rightarrow \infty$$

1.4. Supervised Learning

Our goal is to find an algorithm that is able to map objects from $\mathcal{X}$ to outcomes in $\mathcal{Y}$. Objects are described by their *features*. These are commonly numerical, so $\mathcal{X}$ can be thought of as $\mathbb{R}^d$ where d is the number of features. We will call such a representation of an object an *example*.

In *classification* problems, the outcomes are discrete among k possible outcomes and we refer to them as *labels* or *classes*. For sake of simplicity, we identify these with integers, i.e. $\mathcal{Y}$ can be thought of as the set $\{1, \ldots, k\}$. In *binary* classification, there are only two possible outcomes. Depending on what is more convenient in the mathematical context, we assume either $\mathcal{Y} = \{0, 1\}$ or $\mathcal{Y} = \{-1, 1\}$. In *regression* problems, the outcomes are continous. We can think of $\mathcal{Y}$ as $\mathbb{R}$.

The desired mapping $q : \mathcal{X} \rightarrow \mathcal{Y}$ may be nontrivial such that it is not feasible to come up with explicit rules of how to map examples to outcomes. However, we may try to algorithmically infer such a mapping from a given set of examples and their known outcomes. More specifically, we want to find a deterministic *learning algorithm*, also called *learner*, that, given a random input D, produces a mapping q_D [1]. We call q_D

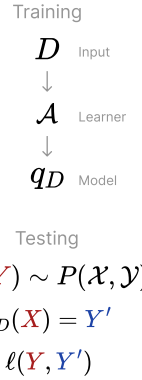

Figure 1.1.: Illustration of the main components of supervised learning. A learning algorithm $\mathcal{A}$ produces a model q_D given some input D. The model is then evaluated on example-outcome pairs of the original data distribution.

1: For the statistical analysis it is essential that the learning algoritm is regarded as deterministic. However, any kind of randomness can be introduced by providing it with random input.

a *model* and note that it is dependent on D . The task of choosing and configuring a learning algorithm is known as *supervised learning*.

To be useful, a model should not only accurately estimate outcomes for the given training examples, but also provide reasonable predictions for examples that were not part of the input to the learning algorithm. To express this formally in a general setting, we use statistical language.

Consider a probability distribution $P(\mathcal{X}, \mathcal{Y})$ from which realisations of example-training pairs are drawn. We write this as $(X, Y) \sim P(\mathcal{X}, \mathcal{Y})$ where (X, Y) are random variables from a joint distribution. This distribution is unknown – else the problem is trivially solved already. In order for our solution to be widely applicable, we strive to make as few assumptions about the distribution P as possible.

The training dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is considered a random vector D drawn from $P(\mathcal{X}, \mathcal{Y})^n$ where n is the number of data points. If there are other sources of randomness in model construction such as, for instance, weight initialisation or neural networks, these can also be considered components of D.

Definition 1.4.1 *A model is a function $q : \mathcal{X} \to \mathcal{Y}$. In supervised learning with a single model, the model depends on the training input D. Its output (prediction) when queried with a random variable X taking values in $\mathcal{X}$ is written as*

$$q_D(X)$$

To shorten notation, we sometimes omit explicitly specifying either random variable, but a dependency on these is always to be understood.

The quality of a single model prediction is measured by means of a *loss function* $\ell : \mathcal{Y} \to \mathcal{Y}$ whose value should be low if the predicted outcome is close to the true outcome. To describe the expected loss across the entire distribution, we consider a pair of random variables $(X, Y) \sim P(\mathcal{X}, \mathcal{Y})$.

Definition 1.4.2 *(Risk and Generalisation Error) The risk of a model is the expected loss over all example-outcome pairs.*

$$Risk(q_D) =_{def} \mathbb{E}_{(X,Y)\sim P}\left[\ell(Y, q_D(X))\right]$$

The quality of a given learning algorithm $\mathcal{A}$ is the expected risk over all possible inputs. We refer to this as the generalisation error.

$$GE(\mathcal{A}) =_{def} \mathbb{E}_D\left[Risk(q_D)\right] = \mathbb{E}_{(X,Y),D}\left[\ell(Y, q_D(X))\right]$$

Note that the choice of the loss function ℓ is fundamental to the evaluation of a learning algorithm.

1.5. Bias, Variance and their Effects

Since we are ultimately interested in the generalisation error of an ensemble, the question arises what forces influence it. To this end, one can strive to mathematically express the generalisation error in terms of specific meaningful quantities. A classical decomposition is the *bias-variance-decomposition*. It is an essential tool to understand learning algorithms in general and ensembles such as Random Forests in particular. For the purpose of this thesis, it is important to understand the decomposition and its

motivation in detail. We will begin by considering the widely known bias-variance-decomposition for regression using the squared-error loss $\ell(y, y') =_{\text{def}} (y - y')^2$. We will then proceed to generalise the decomposition to arbitrary loss functions.

The variance of a random variable with respect to the squared-error loss is defined as the expected distance in terms of loss to the the expected value.

$$\text{Var}_X(X) =_{\text{def}} \mathbb{E}_X\left[(X - \mathbb{E}_X[X])^2\right]$$

As such, $\mathbb{E}_X[X]$ is a centroid to the different realisations of X with respect to the loss function $\ell(Y, Y') = (Y - Y')^2$.

$$\mathbb{E}_X[X] = \arg\min_z \mathbb{E}_X\left[(X - z)^2\right]$$

Recall that a model is a function dependent on the training input D. The variance is a measure of how different the models produced by the learning algorithm will be in terms out outputs if supplied with different realisations of training input D.

$$\text{Var}_D(q_D(X)) = \mathbb{E}_D\left[(q_D(X) - \mathbb{E}_D[q_D(X)])^2\right]$$

$q^\star(X) =_{\text{def}} \mathbb{E}_D[q_D(X)]$ is the *central model* and does not depend on D.

Further, recall that we are considering a joint distribution of example-output pairs (X, Y). We can not assume that the outcomes are not ambiguous. Let $y(X)$ be the outcome associated with X. We measure the variance of actual outcomes $y(X)$ for a given example X around the expected outcome.

$$\text{Var}_Y(y(X)) = \mathbb{E}_Y\left[(Y - \mathbb{E}_Y[y(X)])^2\right]$$

$y^\star(X) =_{\text{def}} \mathbb{E}_Y[y(X)]$ is the *central label* and does not depend on Y.

Using this notation, the bias-variance decomposition for the squared-error loss is given as follows. Note that each variance term is the expected distance in terms of loss to a certain centroid.

$$\mathbb{E}_{(X,Y),D}\left[(Y - q_D(X))^2\right] = \underbrace{\mathbb{E}_{(X,Y)}\left[(Y - y^\star(X))^2\right]}_{\text{Var}(Y)\ \text{("noise")}} \tag{1.1}$$

$$+ \underbrace{\mathbb{E}_X\left[(y^\star(X) - q^\star(X))^2\right]}_{\text{Bias}(Y,q)\ \text{("learner bias")}} \tag{1.2}$$

$$+ \underbrace{\mathbb{E}_{X,D}\left[(q^\star(X) - q_D(X))^2\right]}_{\text{Var}(q)\ \text{("learner variance")}} \tag{1.3}$$

The first term, $\text{Var}(Y)$ is independent of D and q_D. This means we have no means of influencing it with our choice of q_D. $\text{Var}(Y)$ is also referred to as *noise*, *bayes error* or *irreducible error*. The second term, $\text{Var}(q)$ measures the variance of our model around its non-random centroid with respect to different realisations of the random training dataset D. This can be understood as a measure of spread of the learning algorithm with respect to different realisations of D. The third term, $\text{Bias}(q_D, Y)$ is the distance in terms of loss between the expected classifier and the expected label. This can be thought of as a measure of precision of the learning algorithm.

Note that we developed two things: One the one hand, we derived quantities that measure the notions of bias and variance. On the other hand, by virtue of these

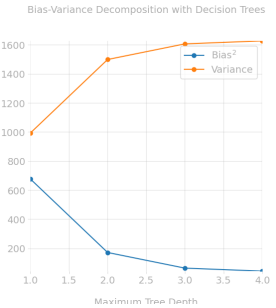

Figure 1.2.: foo!

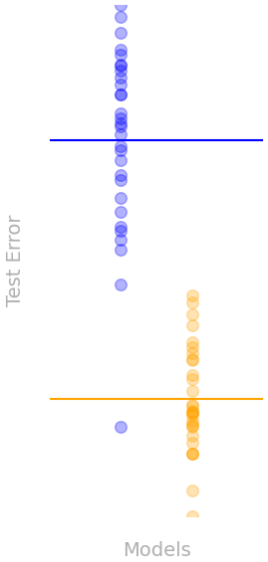

Figure 1.3.: Visualising the variance of Decision Tree and Random Forest models. Each glyph corresponds to the test error of one model trained on a random subset of the full available data. The variation of the test error around the mean test error across many dataset samples is exactly the variance. Not only do Random Forests show lower test errors on average, they seem to also have lower variance. We will explain this observation in **??**

This decomposition is usually derived by expanding the square **[todo]**. The cross-terms then vanish due to that $q^\star = \mathbb{E}_D[q_D]$ and $y^\star = \mathbb{E}_Y[y(X)]$. This is but a special case of a more general structure applying to a certain class of losses. We will provide a more general proof in lemmas 4.4.1 and 4.4.2.

quantities appearing in the error decomposition, we have expressed the *effect* these quantities have on the generalisation error.

While the bias-variance decomposition for the squared-error loss is widely accepted, there are many competing decompositions for a range other other loss functions [**todo**]. For instance, we are particularly interested in the 0/1-loss for classification. A decomposition of it where the model variance is independent of the outcome variable as in term **??** is been proven to not exist [**todo**]. We will now argue that approaching the matter from the perspective of such *loss-effects* allows us to state a general bias-variance decomposition that holds for any loss function. The decomposition for the squared-error loss is a special case of it.

Definition 1.5.1 *(Loss-Effect) For a loss funcion ℓ, and random variables Y, Z, Z', we define the change in loss between Z and Z' in relation to Y as:*

$$LE(Z, Z') =_{def} \ell(Y, Z') - \ell(Y, Z)$$

Note that the arguments of loss-effect appear in inverse order in the difference. This is to give the expression a suggestive shape since in section 4.4, we will show that, for a Bregman divergence B_ϕ, it holds that

$$\mathbb{E}\left[\mathrm{LE}(Z', Z)\right] = B_\phi\left(Z', Z\right)$$

The *variance-effect* is the expected change in loss caused by using q_D instead of the non-random centroid $q^\star(X)$. Likewise, the *bias-effect* is the expected change in loss caused by using the expected model instead of the expected label. In the following, to lighten notation, we will omit explicitly stating the dependence on X. [2] Formally:

$$\begin{aligned}\text{Bias-Effect} &=_{\text{def}} \mathbb{E}_{D,Y}\left[\ell(Y, q^\star) - \ell(Y, y^\star)\right] \\ \text{Variance-Effect} &=_{\text{def}} \mathbb{E}_{D,Y}\left[\ell(Y, q_D) - \ell(Y, q^\star)\right]\end{aligned}$$

2: In the original publication [**james_GeneralizationsBiasVariance_**], bias-effect is called the *systematic effect*, i.e. the effect of the systematic components. However, it is clearer to call this *bias-effect*, particularly when we begin to introduce notions of diversity in **??**.

This allows us to state a decomposition of the generalisation error simply in terms of loss-effects.

$$\mathbb{E}\left[\ell(Y, q)\right] = \mathbb{E}\left[\underbrace{\ell(Y, y^\star)}_{\text{"noise"}} + \underbrace{\ell(Y, q^\star) - \ell(Y, y^\star)}_{\text{"bias-effect"}} + \underbrace{\ell(Y, q) - \ell(Y, q^\star)}_{\text{"variance-effect"}}\right]$$

Note that the individual terms on the right-hand side simply cancel out and reduce to $\ell(Y, q)$. As illustrated in figure **??**, this decomposition divides the the interval from $\ell(Y, Y) = 0$ to $\ell(Y, q)$ into meaningful sections. Note that this decomposition depends solely on the linearity of expectation and is independent of the loss function ℓ or the definitions of $y^\star$ and $q^\star$.

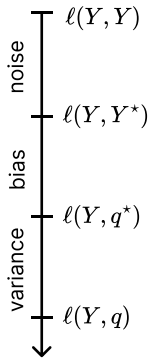

Figure 1.4.: Illustration how the bias-variance-effect decomposition decomposes the loss $\ell(Y, q)$ into meaningful segments.

For the squared-error loss $\ell(Z, Z') = (Z - Z')^2$, bias-effect equals bias and variance-effect equals variance.

$$\begin{aligned}\mathbb{E}\left[\ell(Y, q^\star) - \ell(Y, y^\star)\right] &= \mathbb{E}\left[\ell(y^\star, q^\star)\right] \\ \mathbb{E}\left[\ell(Y, q) - \ell(Y, q^\star)\right] &= \mathbb{E}\left[\ell(q^\star, q)\right]\end{aligned}$$

Thus, the bias-variance decomposition for the squared-error loss is a special case of this.

Theorem 1.5.1 *(Bias-Variance-Effect-Decomposition* [**james_GeneralizationsBiasVariance_**]*)*

Note that while for the squared error the variance **??** compares *model predictions*, the variance-effect is based solely on the *change in loss*. In section 4.4, we will see that the former is in fact only a special property of a specific family of loss functions.

For any loss function L, it holds that

$$\mathbb{E}_{(X,Y),D}\left[\ell(Y, q_D)\right] = \underbrace{\mathbb{E}_{(X,Y)}\left[\ell(Y, y^\star)\right]}_{\text{noise}} + \underbrace{\mathbb{E}_{X)}\left[LE(q^\star, y^\star,)\right]}_{\text{bias-effect}} + \underbrace{\mathbb{E}_{X,D}\left[LE(q_D, q^\star,)\right]}_{\text{variance-effect}}$$

This decomposition holds for *any* loss function since, by linearity of expectation, the individual terms on the right-hand-side simply cancel out. Further, this decomposition is independent of the definitions of $y^\star$ and $q^\star$.

1.6. Classifier Margins

A basic tool in the analysis of classifiers is the notion of margins. The margin can be thought to represent the classifier's confidence in its prediction. $\mathbb{P}[k|x]$ describes the probability estimated by the classifier that X is of class k.

Definition 1.6.1 *(Classifier margins [**tibshirani_ElementsStatisticalLearning_2017**]) The margin for class k of an example X is the difference between the model's confidence that X is of class k and the next-best class:*

$$m(x, y) =_{def} \mathbb{P}[y|x] - \max_{j \neq y} \mathbb{P}[j|x]$$

- *The vector $m(x) = [m_1(x), \ldots, m_K(x)]^\top$, where K is the total number of classes, is called a margin vector iff its components sum to zero.*
- *For a pair (X, y) of example and true outcome, the model's prediction is correct iff $m_y(x) > 0$*

1.7. Bregman Divergences and Centroids

To measure the difference between predicted and ground-truth outcomes, we use a loss function ℓ. The choice of loss function depends on the data domain, the learning task and computational considerations.

Often, learners are analysed with respect to a specific loss function, such as the squared-error loss **[scornet_ConsistencyRandomForests_2015]** for regression or the 0/1-loss **[theisen_WhenAreEnsembles_2023]** or the KL-divergence **[webb_EnsembleNotEnsemble_2019]** for classification. The well-known bias-variance decomposition for the squared-error loss is usually shown directly in teaching materials **[tibshirani_ElementsStatisticalLearning_2017, weinberger_Lecture12Bias_]**. The question then arises which properties are specific to the loss function and which are part of a more general structure.

We will now define a family of loss functions, called *Bregman divergences*, that encompasses many widely used loss functions in supervised learning (see table 1.1).

Definition 1.7.1 *(Bregman Divergence [**pfau**, **adlam22**]) The Bregman divergence $B_\phi(p, q) : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is defined based on a generator function ϕ as follows:*

$$B_\phi(p, q) =_{def} \phi(\mathbf{p}) - \phi(\mathbf{q}) - \langle \nabla\phi(q), (\mathbf{p} - \mathbf{q}) \rangle$$

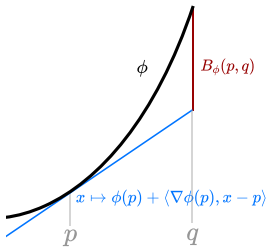

Figure 1.5.: Given a strictly convex generator ϕ, the Bregman divergence for points p, q is the difference between the linear approximation around p and ϕ at the point q.

where $\langle \cdot, \cdot \rangle$ is the inner product, $\nabla\phi(\mathbf{q})$ is the gradient vector of ϕ at $\mathbf{q}$ and $\phi : \mathcal{S} \to \mathbb{R}$ is a strictly convex function on a convex set $\mathcal{S} \subseteq \mathbb{R}^k$ such that it is differentiable on the relative interior of $\mathcal{S}$.

Table 1.1.: Examples of commonly used loss functions that are Bregman divergences [**clustering with bregman divergences, wood23**]

Divergence $B_\phi(p, q)$	Generator $\phi(q)$	Domain S	Loss function
$(p-q)^2$	q^2	$\mathbb{R}$	Squared Error
$x \log\left(\frac{x}{y}\right) + (1-x)\log\left(\frac{1-x}{1-y}\right)$	$x \log x + (1-x)\log(1-x)$	$[0,1]$	Logistic loss
$\frac{x}{y} - \log\left(\frac{x}{y}\right) - 1$	$-\log x$	$\mathbb{R}_{>0}$	Ikura-Saito distance
$\|\| x - y \|\|^2$	$\|\| x \|\|^2$	$\mathbb{R}^d$	Squared Euclidean distance
$(x-y)^\top A(x-y)$	$x^\top A y$	$\mathbb{R}^d$	Mahalanobis distance
$\sum_{j=1}^d x_j \log_2\left(\frac{x_j}{y_j}\right)$	$\sum_{j=1}^d x_j \log_2 x_j$	d-simplex	KL-divergence
$\sum_{j=1}^d x_j \log\left(\frac{x_j}{y_j}\right) - \sum_{j=1}^d (x_j - y_j)$	$\sum_{j=1}^d x_j \log x_j$	$\mathbb{R}^d_{\geq 0}$	Generalized I-divergence
$\sum_{j=1}^d x_j \log x_j$	$\sum_{j=1}^d x_j \log x_j$	$\mathbb{R}_{\geq 0}$	Poisson loss

The "statistical" variance $(X - \mathbb{E}[X])^2$ is the expected distance around the centroid as measured by the squared-error loss. Other loss functions imply different notions of variance. In order to talk about variances with respect to Bregman divergences, we need a notion of a centroid with respect to a Bregman divergence. Bregman divergences are in general not symmetric and hence there is a *left* and *right* centroid.

Lemma 1.7.1 *(Left and right Bregman centroids, [**pfau_GeneralizedBiasVarianceDecomposition_**])*
Let B_ϕ be a Bregman divergence of generator $\phi : \mathcal{S} \to \mathbb{R}$. For a random variable Y taking values in $\mathcal{S}$, it holds that

- *the right Bregman centroid is*

$$\arg\min_z \mathbb{E}_X\left[B_\phi(X, z)\right] = \mathbb{E}[X]$$

- *the left Bregman centroid is*

$$\arg\min_z \mathbb{E}_z\left[B_\phi(z, X)\right] = (\nabla\phi)^{-1}\mathbb{E}\left[\nabla\phi(X)\right]$$

  The left Bregman centroid is the expected value in the dual space implied by $\nabla\phi$. Due to this, we define the dual expectation as

$$\mathfrak{E}[X] =_{def} (\nabla\phi)^{-1}\mathbb{E}\left[\nabla\phi(X)\right]$$

A generalised measure of variance is then the expected divergence around a Bregman centroid.

Definition 1.7.2 *The variance around the right Bregman centroid is known as the Bregman information $I_\phi(X)$ [**banerjee_ClusteringBregmanDivergences_2004**].*

$$I_\phi(X) =_{def} \mathbb{E}_X\left[B_\phi(X, \mathbb{E}_X[X])\right]$$

In other words, the choice of loss function implies a measure of variance. Various well-known variance measures can now be seen to actually be implied by a Bregman divergence. For example, let $X = \{X_1, \ldots, X_n\} \subset \mathbb{R}^d$. Then the squared Euclidean distance

corresponds to the *sample variance* [**banerjee_ClusteringBregmanDivergences_2004**].

$$B_\phi(p, q) = || p - q ||^2 \rightarrow I_\phi(X) = \frac{1}{n}\sum_{i=1}^{n}(X_i - \mathbb{E}[X])^2$$

For the KL-divergence, the bregman information is the *mutual information*. Consider a random variable X over probability distributions with probability measure p [**banerjee_ClusteringBregmanDivergences_2004**].

$$B_\phi(u, v) = \sum_{j=1}^{d} u_j \log\left(\frac{u_j}{v_j}\right) \rightarrow I_\phi(X) = \sum_{i=1}^{n}\sum_{j=1}^{m} p(u_i, v_j) \log \frac{p(u_i, v_j)}{p(u_i)\, p(v_j)}$$

2. Ensemble Learning

Ensemble Learning is the method of training M individual models $q_1, \ldots, q_M$ for a given task and aggregating their output via an *ensemble combiner* $\bar{q}$ to form an ensemble prediction [**zhou_EnsembleMethodsFoundations_2012**]. The individual models $q_1, \ldots, q_M$ are referred to as *members*. When all members are constructed using the same learning algorithm, we call it a *homogeneous* ensemble. The learning algorithm is then referred to as the *base learner*. Otherwise the ensemble is *heterogeneous*.

2.1. Methods

There are three main variants of ensemble learning [**mienye_SurveyEnsembleLearning_2022**]:

- *Parallel*: All members are trained independently. The outputs of all members are then aggregated to form the ensemble prediction.
- *Stacking* or *Meta-Learning*: All members are trained independently. The member outputs serve as input data for another learning algorithm, which then provides the ensemble prediction.
- *Sequential*: Members are trained in sequence. The output of the previous ensemble member informs the construction of the next member.

Random Forests [**breiman_RandomForests_2001**] are an example of parallel ensemble construction. M decision trees are constructed independently and the tree's predictions are aggregated by a kind of mean (see section 3.1). A classical example for sequential ensemble construction is *Boosting* [**schapire_BoostingFoundationsAlgorithms_2012**]. In boosting algorithms, the ensemble combiner $\bar{q}$ is not a mean but a (weighted) sum $\bar{q} = \sum_{i=1}^{M} \alpha_i q_i$. The first member q_1 provides a base prediction. Successive members are then trained to predict not an output value but *increments* (*pseudo-targets*) to the base prediction such that the sum $\alpha_1 q_1 + \alpha_2 q_2 + \ldots$ moves towards a more precise prediction. In this thesis, we will focus on the Random Forest learner and variations of it. Although it can be seen as a parallel ensemble construction method, we will see that it can also be understood as a boosting algorithm (see **??**).

2.2. Notation

In the supervised learning setting with a single model, we have defined the model as a function $q_D(X)$. In ensemble learning, M individual models are constructed and their outputs are aggregated via an ensemble combiner $\bar{q}$ to form an ensemble output. The individual models are referred to as *members*.

Analysing ensembles means analysing differences between the members. There are two dimensions in which ensemble members can differ. First, in terms of provided training data. [1] Second, in terms of other randomness used in model construction. [2] To distinguish these two sources, we sometimes denote the training data as a random vector $D = (D_1, \ldots, D_M)$ and other parameters as $\Theta = (\Theta_1, \ldots, \Theta_M)$.

Definition 2.2.1 *(Ensemble member model) The i-th ensemble member model* $q_i : \mathcal{X} \to \mathcal{Y}$

1: For instance, each learner may be trained on a random subset of the available training data (see section **??**)

2: For instance, random numbers used in decision tree construction (see section 3.1).

is a function depending on training input D_i and additional parameters Θ_i.

$$q_{D_i,\Theta_i}(X)$$

To shorten notation, we also write $q_i =_{def} q_{D_i,\Theta_i}(X)$.

Due to that the i-th member then depends only on D_i and Θ_i and is independent of D_j and Θ_j for $j \neq i$, using the law of total expectation (see lemma **??**), we can write

$$\mathbb{E}_D[q_i] = \mathbb{E}_{D_{j\neq i}}[\mathbb{E}_{D_i}[q_i]] = \mathbb{E}_{D_i}[q_i]$$

and likewise for Θ. The ensemble combiner of a set of members constructed given D and Θ is the dual expectation over the member parameters Θ.

Definition 2.2.2 *(Ensemble combiner) The ensemble combiner $\bar{q} : \mathcal{X} \to \mathcal{Y}$ is defined as*

$$\bar{q}_D =_{def} \mathscr{C}_\Theta[q_{\Theta,D}]$$

An ensemble is homogeneous iff $\Theta_1, \ldots, \Theta_M$ are identically and independently distributed. This is the case if the member models are constructed according to the same base learner and do not influence each other.

Lemma 2.2.1 ⋆ *In homogeneous ensembles, the central models of any two members and the combiner are the same. That is, for any $i, j \in \{1, \ldots, M\}$ it holds that*

$$q_i^\star = q_j^\star = \bar{q}^\star$$

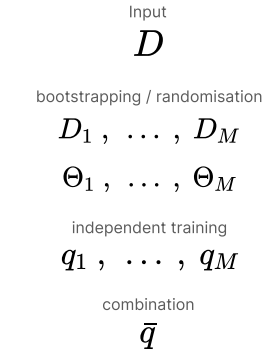

Figure 2.1.: Illustration of parallel ensemble learning.

Proof. If $D_1, \ldots, D_M$ and $\Theta_1, \ldots, \Theta_M$ are identically distributed and independent, it holds that

$$q_i^\star = \mathbb{E}_D[q_{D_i,\Theta_i}] = \mathbb{E}_{D_i}[q_{D_i,\Theta_i}] = \mathbb{E}_{D_j}\left[q_{D_j,\Theta_i}\right] = q_j^\star$$

Both properties also hold for dual expectations for Bregman divergences defined in definition **??**.

For the ensemble combiner, it holds that

$$\begin{aligned}\bar{q}^\star &= \mathscr{C}_\Theta[q_{D,\Theta}]^\star = \mathscr{C}_D[\mathscr{C}_\Theta[q_{D,\Theta}]] \\ &= (\nabla\phi)^{-1}\mathbb{E}_D\left[(\nabla\phi)(\nabla\phi)^{-1}\mathbb{E}_\Theta[q_{D,\Theta}]\right] \\ &= \mathscr{C}_\Theta[\mathscr{C}_D[q_{D,\Theta}]] = \mathscr{C}_\Theta\left[q_\Theta^\star\right]\end{aligned}$$

Due to the result above, $q_\Theta^\star$ is constant over Θ and thus $\mathscr{C}_\Theta\left[q_\Theta^\star\right] = q_\Theta^\star$. □

Similarly to definition 1.6.1, we can define a notion of margin for classification ensembles deciding by majority voting.

Definition 2.2.3 *(Ensemble margins for majority voting [**breiman**]) The margin for class y of an example x is the difference between the number of member votes for class Y and the number of votes for the next-best class.*

$$mr(x, y) =_{def} \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}[q_i = y] - \max_{j\neq y} \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}[q_i = j] \in [-1, 1]$$

If ensemble members are parameterised by Θ, from the statistical point of view, we can also write

$$\mathrm{mr}(x, y) = \mathbb{P}_\Theta\left[q_\Theta = y\right] - \max_{j \neq y} \mathbb{P}_\Theta\left[q = j\right]$$

where the probabilities are conditioned on x. For binary classification under the 0/1-loss, the ensemble margin is linearly related to the ratio of incorrect members $\frac{1}{M}\sum_{i=1}^{M} \ell_{0/1}\left(y, q_i(x)\right) \approx \mathbb{P}_\Theta\left[q \neq y\right]$.

$$\begin{aligned}\mathrm{mr}(x, y) &= \mathbb{P}_\Theta\left[q = y\right] - \mathbb{P}_\Theta\left[q \neq y\right] \\ &= \left(1 - \mathbb{P}_\Theta\left[q \neq y\right]\right) - \mathbb{P}_\Theta\left[q \neq y\right] \\ &= 1 - 2\mathbb{P}_\Theta\left[q \neq y\right]\end{aligned}$$

So, $\mathrm{mr}(x, y) = 1 - 2\frac{1}{M}\sum_{i=1}^{M} \ell_{0/1}\left(y, q_i(x)\right)$.

2.3. Motivation

We will now review some arguments that motivate ensemble learning. We do this to provide context to the results in section **??**, which also clearly show when and how ensemble learning is beneficial.

Ensemble improvement is non-negative The arithmetic mean combiner can be seen as approximating an expectation over member models, i.e. $\bar{q} = \mathbb{E}_\Theta\left[q_\Theta\right]$. This motivated abe to invoke Jensen's inequality . For a loss function ℓ that is convex in its first argument, it holds that

Jensen's inequality, in a probabilistic setting, states that, for a function $\phi : \mathbb{R} \to \mathbb{R}$ and a random variable X

ϕ convex $\to \phi\left(\mathbb{E}\left[X\right]\right) \leq \mathbb{E}\left[\phi(X)\right]$

$$\underbrace{\ell\left(\mathbb{E}_\Theta\left[q_\Theta(X)\right], Y\right)}_{\text{"ensemble loss"}} \quad \leq \quad \mathbb{E}_\Theta\left[\underbrace{\ell\left(q_\Theta(X), Y\right)}_{\text{"member loss"}}\right]$$

and thus

$$\mathbb{E}_\Theta\left[\ell\left(q_\Theta(X), Y\right)\right] - \ell\left(\mathbb{E}_\Theta\left[q_\Theta(X)\right], Y\right) \geq 0$$

Corollary 2.3.1 *For convex loss functions and using the arithmetic mean combiner, ensembling can never hurt performance: The ensemble loss is always smaller-equal than the average member error.*

abe interpret the difference between these quantities, as a measure of ensemble improvement.

Variance reduction for squared-error regression A common motivation for using ensembles over single models is that the combination of models reduces the variance as compared to the expected variance of a single model. Further, the bias is not affected, i.e. the ensemble bias is the same as the bias of any member. In the regression setting, under the squared-error loss and the arithmetic mean combiner, this can be seen as follows [**stackexchange**]. Assume $q_1, \ldots, q_M$ are identically and independently distributed with equal variance σ^2.

$$\mathrm{Var}\left(\bar{q}\right) = \mathrm{Var}\left(\frac{1}{M}\sum_{i=1}^{M} q_i\right) = \frac{1}{M^2}\sum_{i=1}^{M} \mathrm{Var}\left(q_i\right) = \frac{1}{M}\sigma^2$$

As the number of members M increases, the ensemble variance is reduced. Further, one can also see that the interactions between members determine the variance reduction. Assume ensemble members have equal pairwise covariance. Then

$$\rho =_{\text{def}} \frac{\text{Cov}(q_i, q_j)}{\sigma^2} \quad \leftrightarrow \text{Cov}(q_i, q_j) = \rho\sigma^2$$

Further,

$$\text{Var}\,(\bar{q}) = \text{Var}\left(\frac{1}{M}\sum_{i=1}^{M} q_i\right) = \frac{1}{M^2}\left(\underbrace{\sum_{i=1}^{M}\text{Var}\,(q_i)}_{M\sigma^2} + \underbrace{2\sum_{i<j}^{M}\text{Cov}(q_i, q_j)}_{M(M-1)\rho\sigma^2}\right) = \frac{\sigma^2}{M} + \frac{M-1}{M}\rho\sigma^2$$

One can show that $\rho \geq 0$ [**louppe_UnderstandingRandomForests_2015**].

Corollary 2.3.2 *Under the squared-error loss, ensemble variance is minimised if member outputs are uncorrelated.*

Variance reduction for classification margins For classification, a classical analysis is the bound given by **breiman_RandomForests_2001** in [**breiman_RandomForests_2001**], in which Random Forests are first introduced. The basic idea is to consider variances with respect to the ensemble margin (see definition 2.2.3). We open with Chebychev's inequality to bound the generalisation error in terms of the variance of the ensemble margin.

$$\mathbb{E}\left[\ell(Y, \bar{q}(X))\right] = \mathbb{P}\left[\text{mr}(X, Y; D) < 0\right] \quad \leq \quad \frac{\text{Var}\,(\text{mr}(X, Y; D))}{\mathbb{E}_{X,Y}\left[\text{mr}(X, Y; D)\right]^2}$$

We can already see that we have an interaction between the performance of the individual members, as reflected in the ensemble margin $\mathbb{E}_{X,Y}\left[\text{mr}(X, Y; D)\right]$ and the variance of the margin. The generalisation error is in part determined by the ratio of these two quantities.

$s =_{\text{def}} \mathbb{E}_{X,Y}\left[\text{mr}(X, Y; D)\right]$ is also called the *strength* of the ensemble. s is assumed to be non-negative. For binary classification, this is equivalent to the weak-learner assumption (see 4.5.4).

Note that

$$\text{mr}(X, Y; D) = \mathbb{E}_{\Theta}\left[\underbrace{\mathbb{1}\left[q_i = Y\right] - \mathbb{1}\left[q_i = K\right]}_{=_{\text{def}}\ \text{rmg}(X,Y,\Theta)} \mid (X, Y), D\right]$$

where K is the next-best class and we define the *raw margin function* $\text{rmg}(X, Y, \Theta)$ to be the inner part of that expectation. So, $\text{mr}(X, Y; D) = \mathbb{E}_{\Theta}\left[\text{rmg}(X, Y, \Theta) \mid (X, Y), D\right]$. Notably, *mr* is the *ensemble* margin measuring the ratio of incorrect members and *rmg* corresponds to the *classifier* margin , measuring the ratio of incorrectly classfied examples by a member (see definitions 2.2.3 and 1.6.1).

Theorem 2.3.3 *([**breiman_RandomForests_2001**]) The variance of the ensemble margin can be expressed in terms of the covariance between the raw member margins of two members parameterised by i.i.d* Θ, Θ'.

$$Var_{X,Y}\,(mr(X, Y; D)) = \mathbb{E}_{\Theta,\Theta'}\left[Cov_{(X,Y)}(rmg(\Theta), rmg(\Theta'))\right]$$

Proof. For brevity, we write $Z =_{\text{def}} (X, Y)$, $\text{mr}(Z) =_{\text{def}} \text{mr}(X, Y; D)$ and $\text{rmg}(\Theta) =_{\text{def}} \text{rmg}(X, Y; \Theta)$. By the definition of variance, have

$$\begin{aligned}\text{Var}_Z\left(\text{mr}(Z)\right) &= \mathbb{E}_Z\left[\left(\text{mr}(Z) - \mathbb{E}_Z\left[\text{mr}(Z)\right]\right)^2\right] \\ &= \mathbb{E}_Z\left[\text{mr}(Z)^2\right] - \mathbb{E}_Z\left[\text{mr}(Z)\right]^2\end{aligned}$$

For the left term, by the rule of iterated expectation and the fact that Z and Θ are independent (see lemma **??**), it holds that

$$\mathbb{E}_Z\left[\text{mr}(Z)^2\right] = \mathbb{E}_Z\left[\mathbb{E}_\Theta\left[\text{rmg}(\Theta) \mid Z\right]^2\right] = \mathbb{E}_{Z,\Theta}\left[\text{rmg}(\Theta)^2\right] = \mathbb{E}_\Theta\left[\mathbb{E}_Z\left[\text{rmg}(\Theta)^2\right]\right]$$

For the right-hand-side term, we can make use of the fact that, for some function f, it holds that $\mathbb{E}_\Theta\left[f(\Theta)^2\right] = \mathbb{E}_{\Theta,\Theta'}\left[f(\Theta)f(\Theta')\right]$ where Θ and Θ' are independent and identically distributed. We apply the rule of iterated expectation and exploit that Z and Θ are independent.

$$\begin{aligned}\mathbb{E}_Z\left[\text{mr}(Z)\right]^2 &= \mathbb{E}_Z\left[\mathbb{E}_\Theta\left[\text{rmg}(\Theta) \mid Z\right] \cdot \mathbb{E}_{\Theta'}\left[\text{rmg}(\Theta') \mid Z\right]\right] \\ &= \mathbb{E}_Z\left[\mathbb{E}_\Theta\left[\text{rmg}(\Theta)\right]\mathbb{E}_{\Theta'}\left[\text{rmg}(\Theta')\right]\right] \\ &= \mathbb{E}_{\Theta,\Theta'}\left[\mathbb{E}_Z\left[\text{rmg}(\Theta)\right]\mathbb{E}_Z\left[\text{rmg}(\Theta')\right]\right] \\ &= \mathbb{E}_\Theta\left[\mathbb{E}_Z\left[\text{rmg}(\Theta)\right]^2\right]\end{aligned}$$

In summary, we can conclude that the variance of the ensemble margin is equal to the expected variance of the raw classifier margin. This variance can then also be expressed as the covariance between two independent, identically distributed random variables Θ and Θ'.

$$\begin{aligned}\text{Var}_Z\left(\text{mr}(Z)\right) &= \mathbb{E}_\Theta\left[\mathbb{E}_Z\left[\text{rmg}(\Theta)^2\right]\right] - \mathbb{E}_\Theta\left[\mathbb{E}_Z\left[\text{rmg}(\Theta)\right]^2\right] \\ &= \mathbb{E}_\Theta\left[\mathbb{E}_Z\left[\text{rmg}(\Theta)^2\right] - \mathbb{E}_Z\left[\text{rmg}(\Theta)\right]^2\right] \\ &= \mathbb{E}_\Theta\left[\text{Var}_Z\left(\text{rmg}(\Theta)\right)\right] \\ &= \mathbb{E}_{\Theta,\Theta'}\left[\text{Cov}_Z(\text{rmg}(\Theta), \text{rmg}(\Theta'))\right]\end{aligned}$$

□

As we have seen before for the regression case, the generalisation error is lower if individual members are uncorrelated. Unfortunately, due to the initial application of Chebychev's inequality, this is only an upper bound.

Ensemble bias equals average member bias It can be shown that the bias of a homogeneous ensemble is equal to the average bias of the ensemble members. We will give an illustrative argument for the arithmetic mean combiner here . We will show this in detail in a more general and inuitive way in section **??**.

Lemma 2.3.4 *(Ensemble bias equals average member bias under arithmetic mean combiner [***louppe_UnderstandingRandomForests_2015***])*

$$\mathbb{E}_X\left[\ell(y^\star(X), \bar{q}^\star(X))\right] = \mathbb{E}_X\left[\ell(y^\star(X), q^\star(X))\right]$$

Proof. Consider individual learner inputs $D = (D_1, \ldots, D_M)$. Each member depends on some D_i and the combiner $\bar{q}$ depends wholly on D. Assuming that $D_1, \ldots, D_M$ are independent and identically distributed, we can write

$$\mathbb{E}_{D,\Theta}\left[\bar{q}\right] = \mathbb{E}_D\left[\frac{1}{M}\sum_{i=1}^{M} q_{D_i}\right] = \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_{D_i}\left[q_{D_i}\right] = \mathbb{E}_{D'}\left[q_{D'}\right]$$

where D' is distributed as any D_i. We can conclude $\mathbb{E}_D\left[\bar{q}\right] = \mathbb{E}_D\left[q_D\right]$ (see section 2.2). This implies

$$\bar{q}^\star = \mathbb{E}_D\left[\bar{q}\right] = \mathbb{E}_D\left[q_D\right] = q^\star$$

and thus the bias of the ensemble is the same as the bias of a member model q. □

This argument depends on the linearity of expectations, the arithmetic mean combiner and the fact that the ensemble is homogeneous. We will later show this more directly for any loss function and any combiner (see 4.3).

Corollary 2.3.5 *For homogeneous ensembles under the arithmetic mean combiner, ensemble improvement is solely due to variance reduction.*

2.4. Applications

...

3. Random Forests

In this chapter, we describe the Random Forest learning algorithm. We first motivate and describe decision trees, which are the basic components of a Random Forest. We then proceed to describe a particular property of decision trees: they are likely to exhibit high variance. While this is undesirable for a learning algorithm, we will see that combining several randomized decision trees into a Random Forest ensemble turns exactly that property into a crucial advantage (see section 4.3).

In its essence, a Random Forest is a collection of randomized decision trees. A decision tree is a data-driven recursive partitioning scheme, combined with a means to produce a prediction based on the training points in a partition cell. The Random Forest prediction then is an aggregate of the predictions of all individual trees.

3.1. Decision Trees

As described in section 1.4, we are interested in learning algorithms that, given some training data, produce a model that is able to predict a reasonable outcome when queried with a previously unseen example.

One intuitive approach is to consider the examples in the training data that are "close" to the query point. Then, one might claim that the outcome for the query point must surely be similar to the outcomes of the close points – which we already know. Indeed, finding a proper notion of "closeness" is at the heart of many machine learning algorithms such as k-Nearest-Neigbours, k-Means, etc.

Constructing a decision tree means recursively partitioning the input space $\mathcal{X}$, guided by the training data D. Then, given a query X, we check the partition cell that X belongs to and all the training examples that are in it. These are the examples we consider "close" to X. The tree's prediction will be an aggregation of the outcomes of all training points in that cell.

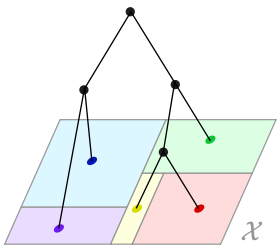

Figure 3.1.: Rendering of a decision tree structure. Each inner node corresponds to a partitioning of the parent edge. In standard decision trees, this is a binary partition. In other words, the examples are *split* at a certain value threshold in a certain feature dimension.

Because we are recursively partitioning the input space, we have at hand a tree structure of decision rules. The cells of the resulting partition are the leaves of the tree. The non-leaf nodes are also referred to as *decision nodes*, but there is no inherent difference between leaf and non-leaf nodes. We will use either of the terms *leaf* and *cell*, depending which aspect we want to emphasize.

In summary, there are the following main components to the implementation of a decision tree:

- The *splitting criterion* to apply recursively to subsets of the training data.
- The *stopping criterion* that determines whether a node should be split further. This will determine the depth of the decision tree.
- The *leaf aggregation function* that produces a prediction for a specific cell. When using the constructed tree for prediction, this will be the leaf node that the query point is assigned in.

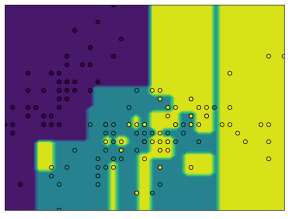

Figure 3.2.: Decision boundaries of a tree in two (arbitrary) features constructed on the *iris* dataset. Visualisation based on [_SklearnInspectionDecisionBoundaryDis

3.1.1. Centroids as leaf combiners

We will consider the leaf aggregation function first. Consider a parent node P that, due to some split, was partitioned into the disjoint union $L \dot\cup R$. Let y_P, y_L and y_R be the output values of the parent and the two new leaf nodes, produced by the leaf aggregation function. Since the leaf output is constant over a single cell, the gain in loss due to a split is the difference between the loss of the parent node and the sum of losses of the two individual child nodes. For brevity, we write $\ell_P(y) =_{\text{def}} \sum_{i \in P} \ell(y, y_i)$.

$$\textit{Loss Gain:} \quad \sum_{i \in P} \ell(y_P, y_i) - \left(\sum_{i \in L} \ell(y_L, y_i) + \sum_{i \in R} \ell(y_R, y_i) \right)$$

$$= \ell_P(y_P) - (\ell_L(y_L) + \ell_R(y_R))$$

In order for a split to yield positive loss gain, the leaf aggregation function needs to be such that the loss does not increase as constraints are removed, i.e. the set of considered examples is reduced. Recall that $\bar z$ is a centroid with respect to a loss function ℓ and a set of outcomes P if and only if $\bar z = \arg\min_z \sum_{i \in P} \ell(z, y_i) = \arg\min_z \ell_P(z)$ (**??**).

Lemma 3.1.1 *For a loss function ℓ, if the leaf aggregator of a node P is $y_P =_{def} \arg\min_z \sum_{i \in P} \ell(z, y_i)$, i.e. the centroid with respect to ℓ, the loss gain is nonnegative.*

Proof. Let $\ell_P(y) =_{\text{def}} \sum_{i \in P} \ell(y, y_i)$. Since $P = L \dot\cup R$, we need to show that $\ell_P(y_P) = \ell_L(y_P) + \ell_R(y_P) \geq \ell_L(y_L) + \ell_R(y_R)$. Assume $\ell_L(y_P) < \ell_L(y_L)$. This contradicts the definition of y_L as the minimizer, and as such $\ell_L(y_P) \geq \ell_L(y_L)$. Likewise, we can conclude that $\ell_R(y_P) \geq \ell_R(y_R)$. Combining the two inequalities yields the statement. □

This rigorously motivates the specific choice of leaf aggregation function. The majority vote is a centroid with respect to the 0/1-loss for classification, while the arithmetic mean is the centroid with respect to the squared error loss for regression.

3.1.2. Splitting criteria greedily minimise loss functions

In the best possible case, all training examples in a given cell correspond to the same (classification) or very similar (regression) outcomes. If a query point then falls within that cell, i.e. it has similar features, one can say with high confidence that the query point should have the same (similar) outcome. In the spirit of greedy optimisation, we aim to split cells such that the resulting child cells are more *pure* with respect to their outcomes.

Consider a split, parameterised by Θ, that partitions a parent node P into the disjoint union $L_\Theta \dot\cup R_\Theta$. Let n, n_L, n_R be the cardinalities of the parent and the two child nodes. Let H be an impurity measure. We will select the split that yields the lowest impurity.

$$\arg\min_\Theta \ \frac{n_L}{n} H(L_\Theta) + \frac{n_R}{n} H(R_\Theta)$$

The gain in purity is then the difference between impurities before and after the split.

$$\textit{Purity Gain:} \quad H(P) - \left(\frac{n_L}{n} H(L_\Theta) + \frac{n_R}{n} H(R_\Theta) \right) \tag{3.1}$$

Note that this is different from the gain in *loss* achieved due to a split.

The notion of local purity is linked to the training error: If a leaf cell is perfectly pure, all training examples in that cell correspond to the same outcome. Hence, the leaf aggregation function, which us usually implemented as some kind of mean, will produce exactly that outcome for any query point that belongs to this cell, in particular any training points. Consequently, for a suitable definition of "error", perfectly pure cells have zero training error.

We now proceed to define two commonly used splitting criteria. These are the Gini impurity for classification and Variance Reduction (also known as Cart) for regression [**tibshirani_ElementsStatisticalLearning_2017**]. For a splitting function criterion to produce useful decision trees, a reasonable impurity measure should also imply a positive loss gain. Often, this is intuitively clear but particularly for the case of Gini impurity, a comprehensive explanation appears to be hard to find in the literature. We clarify this in the following.

Variance Reduction

A commonly used impurity measure for regression is the squared-error variance.

$$H_{\text{var}}(P) =_{\text{def}} \frac{1}{n_P}\sum_{i\in P}(y_i - y_P)^2 \qquad \text{for} \quad y_P =_{\text{def}} \frac{1}{n_p}\sum_{i\in P} y_i$$

To motivate this impurity measure, it remains to be shown that a split guided by this impurity measure actually reduces the value of a specific loss function and which one that is. Luckily, it is easy to see that this holds for the squared error loss. Plugging the definition into the purity gain (eq. (3.1)) yields

$$H(P) = \frac{1}{n_P}\sum_{i\in P}(y_i - y_P)^2 = \frac{1}{n_P}\underbrace{\sum_{i\in L}(y_i - y_P)^2}_{\ell_L(y_P)} + \frac{1}{n_P}\underbrace{\sum_{i\in R}(y_i - y_P)^2}_{\ell_R(y_P)}$$

and

$$\frac{n_L}{n_P}H(L) + \frac{n_R}{n_P}H(R) = \frac{n_L}{n_P}\frac{1}{n_L}\underbrace{\sum_{i\in L}(y_i - y_L)^2}_{\ell_L(y_L)} + \frac{n_R}{n_P}\frac{1}{n_R}\underbrace{\sum_{i\in R}(y_i - y_R)^2}_{\ell_R(y_R)}$$

By lemma 3.1.1, we can directly conclude that the loss gain is positive for any split if the arithmetic mean is used as a leaf combiner.

In binary classification, i.e. if outcomes are in $\{0, 1\}$, the squared error reduces to the 0/1-loss. As such, the mean 0/1-loss, i.e. the error rate, trivially also is a measure of variance.

Another impurity measure is the entropy, defined as

$$H_{\text{entr}}(P) =_{\text{def}} -\sum_{i\in P} p_k(x_i)\log(p_k(x_i))$$

With a similar argument as for the case of variance reduction, one can see that the entropy splitting criterion amounts to minimising the the cross-entropy loss given as

$$-\frac{1}{n}\sum_{i}^{n}\sum_{k}\mathbf{1}[y_i = k]\log(p_k(x_i))$$

Gini Impurity

One may suggest a measure of purity as the probability of drawing two different outcomes from the examples in the current cell. Let $p_k = \mathbb{P}_P[k|X]$ be the probability of drawing an example of class k from node P. The probability of drawing one example of class k and one of a different class is $p_k(1 - p_k)$. The probability of drawing two examples of *any* two different classes then is the *Gini impurity*

$$G =_{\text{def}} \sum_k p_k(1 - p_k) = 1 - \sum_k p_k^2$$

We will now argue that a reduction in the Gini impurity in fact pushes values p_k to the extremes of the probability simplex. We perform a slight shift in perspective and consider the classification *margin* (see 1.6.1) instead of the estimated probabilites.

We will argue that the Gini impurity split criterion, which finds a split such that the Gini impurity is reduced, in fact maximises the classification margins.

Recap: The classifier margin for class k of an example X is the difference between the model's confidence that X is of class k and the next-best class:

$$m_k(X) =_{\text{def}} \mathbb{P}[k|X] - \max_{j\neq k}\mathbb{P}[j|X]$$

For a pair (X, y) of example and true outcome, the model's prediction is correct iff $m_y(X) > 0$ The vector $m(x) = [m_1(x), \ldots, m_K(x)]^\top$, where K is the total number of classes, is called a *margin vector* iff its components sum to zero.

Lemma 3.1.2 ⋆ *Let p be a probability distribution and u an arbitrary vector. Let $G = \sum_k p_k(1 - p_k)$ be the Gini impurity. Then $-G$ is the generator for the Bregman*

divergence

$$B_{-G}(p,u) = \sum_k (p_k - u_k)^2$$

Proof. Let $\phi(q) =_{\text{def}} (-1)\sum_k p_k(1-p_k)$. Then, the first equality follows by definition of a Bregman divergence (see 1.7.1) and the second equality by arithmetic.

$$\begin{aligned} B_{-G}(p,u) &= \underbrace{(-1)\sum_k p_k(1-p_k)}_{\phi(p)} - \underbrace{(-1)\sum_k u_k(1-u_k)}_{\phi(u)} - \underbrace{\sum_k (2u_k-1)(p_k-u_k)}_{\langle \nabla\phi(u), p-u\rangle} \\ &= \sum_k (p_k - u_k)^2 \end{aligned}$$

□

Note further that maximising the value of the generator function ϕ with respect to p while leaving the the other parameter u fixed also maximises the divergence $B_\phi(p,u)$.[1] The sign of the generator value and the sign of the divergence are related in that $B_{-\phi}(p,u) = -B_\phi(p,u)$. This means that minimising the Gini impurity during splitting maximises component-wise sum of squared errors between p and u.

1: TODO explain, at least with intuition. Convexity of $B_\phi(.,.)$ etc

$$\min G \longrightarrow \min B_G(p,u) \longrightarrow \max B_{-G}(p,u) = \sum_k (p_k - u_k)^2$$

If p is a probability distribution and $u =_{\text{def}} [\frac{1}{k}, \ldots, \frac{1}{k}]^\top$, then $p - u$ is a margin vector and the optimisation corresponds to maximising the classification margins as measured by the squared error.

A common approach in training classification models is *margin maximisation* [**schapire_BoostingFoundationsAlgorithms_2012**] in which we aim to maximise the margin of the true label $m_y(X)$. A margin loss function $\ell : \mathbb{R} \to \mathbb{R}$ is a *margin-maximising* loss if $\ell'(m_y(X)) \leq 0$ for all values of m_y [**leistner_SemiSupervisedRandomForests_2009**].

An example for a margin-maximising loss function is the *hinge loss* defined as $\ell(m_y(x)) =_{\text{def}} \max\{0, 1-p\}$. Its subderivative is

$$\frac{\partial \ell}{\partial p} = \begin{cases} -1 & p \leq 1 \\ 0 & \text{else} \end{cases}$$

and hence it is a margin-maximising loss.

A decision tree can also be evaluated based on a margin loss. The empirical error of a decision tree node P with respect to a margin loss ℓ can be written as $L(P) = \frac{1}{|P|}\sum_{i\in P} \ell\left(m_y(x_i)\right)$ where $m_y(x_i)$ is the value of the true margin. Then the following holds [**leistner_SemiSupervisedRandomForests_2009**].

$$\begin{aligned} L(P) &= \frac{1}{|P|}\sum_{i\in P}\sum_k \mathbb{1}[y_i = k] \cdot \ell\left(m_y(x_i)\right) \\ &= \sum_k \frac{1}{|P|}\sum_{i\in P} \mathbb{1}[y_i = k] \cdot \ell(m_k(x_i)) \\ &= \sum_k p_k(x_i)\ell(m_k(x_i)) \end{aligned}$$

Hence we see that the Gini impurity splitting criterion greedily optimises classification margins and thus margin-maximising losses.

Information Gain

Another common impurity measure is the *Shannon entropy*.

$$H =_{\text{def}} -\sum_k p_k \log p_k$$

The purity gain with respect to the entropy is referred to as *information gain*. Seeing that the negative entropy is the Bregman generator of the KL-divergence (see table 1.1), one can apply a similar argument as given for the Gini impurity to see that constructing a tree using the entropy impurity measure corresponds to greedily optimising for a KL-divergence whose second argument is a uniform distribution – which is also known as *relative entropy* or *cross-entropy*.

3.1.3. Splitting criteria as 2-means clustering

As established in section 3.1.2, splitting criteria aim to find a binary partition of the examples in the parent node. The quality of a partition is evaluated according to the purity of the labels. Note that the example features are not considered at all here. If indeed a centroid is chosen as leaf combiner, the notion of impurity is very similar to the objective function of k-means clustering.

Let I_ϕ be the Bregman information as defined in definition 1.7.2. Let X be a random variable representing data points. Let M be a random variable taking values in $\mathcal{M}$ representing the set of cluster centroids. Then the objective for generalised k-means hard clustering [**banerjee**] is to minimise the loss in Bregman information due to the quantisation induced by M:

$$\ell_\phi(M) =_{\text{def}} I_\phi(X) - I_\phi(M)$$

One can show [**banerjee**] that

$$\ell_\phi(M) = \mathbb{E}_\pi \left[I_\phi(X_k) \right] \approx \sum_{h=1}^{K} \sum_{x_i \in \mathfrak{X}_k} v_i B_\phi \left(x_i, \mu_h \right)$$

where K is the number of clusters, $\mathfrak{X}_k$ are the cells of the clustering, v_i is the distribution of the x_i and μ_k is the right Bregman centroid of $\mathfrak{X}_k$.

Classical k-means is a special case of this for the squared-error divergence. The KL-divergence implies the mutual information as a variance and yields *information-theoretic clustering* [**todo**]. The Ikura-Saito divergence yields the *LBG algorithm* [**todo**].

Particularly relevant for decision trees is the following insight: The choice of Bregman divergence implies a measure of variance (see 1.7.2 for examples). Optimising for this measure of variance implies the splitting criterion [2]. This gives a theoretical rationale for choosing splitting function and leaf combiner in decision trees.

2: The leaf combiner, which corresponds to the right Bregman centroid is independent on the chosen divergence, see **??**.

3.1.4. Stopping Criteria & Tree depth

...

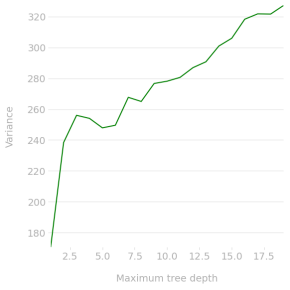

Figure 3.3.: Variances of decision trees of increasing depths. Evaluated for squared-error regression on a synthetic dataset.

3.2. The Random Forest scheme

The deeper a decision tree becomes, the closer the decision regions will fit the training data. This approximation is in fact guided *only* by the training data. In the extreme case, if the tree is fully grown, each partition cell will correspond to a single example and the outcome for that cell will be the outcome of that example. The tree essentially degenerates to a 1-nearest-neighbour scheme with respect to the training dataset. This means that trees constructed with different samples D of training datasets from the

original distribution $P(X, Y)$ potentially predict quite different outcomes for testing datapoints. This is captured in the concept of model variance as defined in 1.5. In **??**, one can indeed observe that trees of greater depth have greater variance. At the same time, their fit to the training data improves, which is captured by lower bias.

Mitigating this strong dependence on the training data is one of the main motivations of Random Forests. The basic idea is as follows: If we produce several uncorrelated decision trees and average their predictions, then the predictions should exhibit lower variance [3]. Exactly how we facilitate uncorrelated trees gives rise to the Random Forest scheme.

3: We use this intuition here to motivate the basic components of Random Forests. We will treat effect of variance reduction through combining multiple models thoroughly in **??**.

The basic idea is to introduce randomness into the decision tree construction. This is achieved by two mechanisms:

- **Bootstrapping**: Each tree is constructed not on the entire training dataset but a random subset of it. Usually, this *bootstrap sample* is produced by drawing the same amount of points with replacement.
- **Random feature selection**: When determining where to split a node, not all features are considered but only a random subset of a certain size.

Thus, a random forest additionally requires the following parameters:

- number of trees...

3.2.1. Bagging

A vital ingredient to Random Forests is the *Bagging* procedure, which stands for *bootstrapping and aggregating*. Bagging is an ensemble learning technique not specific to Random Forests. In Bagging, each member is constructed not on the full training dataset but a *bootstrap sample* of it. The bootstrap sample is usually determined by drawing n out of n examples uniformly with replacement [**breiman**, **others**]. We will refer to this as *uniform bootstrapping*. One can also determine the bootstrap sample by drawing n out of n points with replacement according to a probability distribution $\{p_1, ..., p_n\}$, which we call *weighted bootstrapping*. Bootstrapping means that each member will be trained on a different dataset.

In uniform bootstrapping, if we draw n samples from n available points, the probability of an example being selected in a single draw is $\frac{1}{n}$. Conversely, the probability of an example not being selected in a single draw is $1 - \frac{1}{n}$. We draw n times. Hence, the probability of an observation not being selected in any of the draws is $(1 - \frac{1}{n})^n$. The probability of an example indeed being selected in at least one of the draws then is $1 - (1 - \frac{1}{n})^n$. For large n, one can approximate $\lim_{n\to\infty} 1 - (1 - \frac{1}{n})^n = 1 - e^{-1} \approx 0.632$ [**todo**].

3.2.2. Feature & Split selection

3.2.3. Number of trees

3.2.4. Depth of trees

3.2.5. Random Forests converge

As the number of trees increases, for almost surely all sequences $(\Theta_1, \dots)$, the generalisation error $PE^\star$ converges:

$$\mathbb{E}_{(X,Y),D}\left[\ell(Y, q_D(X))\right] \to \mathbb{P}_{\mathbf{X},Y}\left[\mathrm{mr}(\mathbf{X}, Y) < 0\right]$$

3.2.6. Random Forests do not overfit

3.2.7. Random Forests are consistent

3.2.8. Tree and Forest Partitions

The generalisation error, and consequently individual terms of any decomposition of it have been defined point-wise. That is, they are measured by an expectation over possible realisations of example-outcome pairs (X, Y). They further depend on a random variable D representing the training input to the learner. In order to estimate such an expectation for a given model (a realisation of D), one has to sample realisations of (X, Y). In practise, these example-outcome pairs typically come from a validation dataset that was withheld from training. We have seen in **??** that for some losses, diversity can be expressed independently of the outcome variable Y. To approximate an actual value, we would still need to sample realisations of X.

Decision Trees, particularly if grown deeply, can be considered to *approximate* the training data, i.e. they are a lossy representation of the training data. A grown decision tree model contains two kinds of parameters, both derived from the training data D.

- The tree structure, i.e. the decision boundaries. These are used for determining the leaf node for a query example.
- The output value of a leaf node. This is the predicted value for a query example falling into that leaf. The leaf predictions depend on the decision boundaries but are not solely determined by them.

This leads to the question whether characteristics of a Random Forest model could be expressed solely in terms of its tree parameters, and not in terms of predictions on query points.

Each level of a decision tree induces a partition of the space of examples $\mathcal{X}$. Because each example is associated with an outcome, we can also think of it as a partition of $(\mathcal{X}, \mathcal{Y})$. We call such a partition a *tree partition* and a cell a *tree cell*. Decision trees produce predictions via an aggregate of the queried leaf node's outcomes. Thus, the predictions of a decision tree over a single cell are constant. An ensemble of trees also induces a partition: the partition obtained by intersecting all tree partitions. We call these *forest cells*. Formally, if $T_1, \dots, T_M$ are tree partitions, the forest partition is given as

$$\{c_1 \cap \dots \cap c_M \mid c_1 \in T_1, \dots, c_M \in T_M\}$$

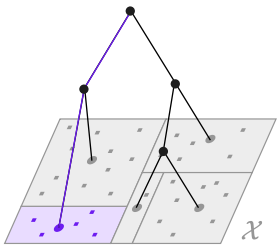

Figure 3.4.: A decision tree partitioning the data space. For a query example, the corresponding leaf node is determined by traversing the tree downwards from the root node and applying the learned decision criteria.

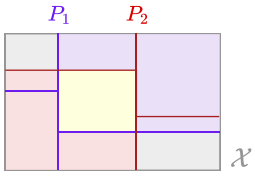

Figure 3.5.: The data space $\mathcal{X}$ (gray) and partitions P_1 (purple) and P_2 (red) of it induced by two decision trees. The intersections of any cell of P_1 and any cell of P_2 form the *forest partition*. One such intersection is highlighted in yellow.

Each forest cell is associated with M tree cells whose intersection constitutes it. For any query point that falls within a certain tree cell, the forest prediction is given by an aggregate over the associated tree cells. Thus, the predictions of a random forest are constant over a single forest cell. This means that also a loss, as well as any decomposition constituents of the loss are constant over forest cells.

However, exploiting that partitions are cell-wise constant, we can apply a special case of the law of total expectation to express it in terms of cells. [4] Consider a forest partition $Z = Z_1 \dot\cup \ldots \dot\cup Z_P$ of $Z = (X, Y)$. Using the linearity of expectations and the law of total expectation, we can write the generalisation error of the forest as follows.

$$\mathbb{E}_{Z,D}\left[\ell(y, \bar{q})\right] = \mathbb{E}_D\left[\sum_{p=1}^{P} \mathbb{P}\left[Z_p\right] \cdot \mathbb{E}_Z\left[\ell(y, \bar{q}) \mid Z_p\right]\right]$$

4: Let $X_1 \dot\cup \ldots \dot\cup X_M$ be a disjoint, countable partition of the sample space of X. Then

$$\mathbb{E}_X[X] = \sum_{i=1}^{M} \mathbb{E}_X[X \mid X_i] \cdot \mathbb{P}[X_i]$$

While we have eliminated the dependence on a query point X, the quantity $\mathbb{E}_Z\left[\ell(y, \bar{q}) \mid Z_p\right]$ still depends on a realisation of an outcome Y. In section **??** we will see that we can decompose this term further into components that are dependent and independent of Y, respectively.

3.2.9. Practical advantages

parallelisable, oob estimation, feature importance, proximity, ...

4. Diversity

It is evident that the interactions between members are a driving force behind member performance (see, for example, section 2.3). Understanding ensembles means understanding how individual member models can be related to each other and how these relationships affect the ensemble performance. In this section, we will give an overview over some of the measures proposed to quantify ensemble diversity. We focus on the classification task.

4.1. Measures of Diversity

For a suitable diversity measure, the following properties are desirable.

- The measure captures the differences between member models.
- There is a relationship between the diversity measure and the ensemble generalisation error.
- The diversity measure is independent of the outcome variable.

Naturally, many diversity measures are based on some notion of spread of the member models. This can, for instance, be a measure of disagreement, variance, impurity, entropy, or covariance.

4.1.1. Disagreement

We will begin with two simple measures based on the contingency table [**zhou**]. To shorten notation, we will refer to its entries as given in table **??**. The *Disagreement Measure* is the proportion of examples on which two members make different predictions.

	$q_i = +1$	$q_i = -1$
$q_j = +1$	$n_{(+,+)}$	$n_{(+,-)}$
$q_j = -1$	$n_{(-,+)}$	$n_{(-,-)}$

Table 4.1.: Notation for entries of the contingency table.

$$\frac{1}{n}(n_{(+,-)} + n_{(-,+)})$$

The Q-*Statistic* is given as

$$Q_{ij} =_{\text{def}} \frac{n_{(+,+)}n_{(-,-)} - n_{(-,+)}n_{(+,-)}}{n_{(+,+)}n_{(-,-)} + n_{(-,+)}n_{(+,-)}}$$

$Q_{ij} = 0$ if q_i and q_j are independent, positive if the members make similar predictions and negative if the members make different predictions.

4.1.2. Ambiguity

[**krogh1995**] propose a decomposition of the ensemble generalisation error into two terms: One describing the average error of ensemble members and a so-called *ambiguity* term.

$$\mathbb{E}\left[\ell(y, \bar{q})\right] = \frac{1}{M}\sum_{i=1}^{M}\ell(y, q_i) - \frac{1}{M}\sum_{i=1}^{M}\ell(\bar{q}, q_i)$$

This perfectly divides the ensemble error into error due to characteristics of the individual member models and error due to interactions between member predictions. Note that the contribution of the ambiguity term here is negative, that is, ambiguity is a beneficial influence. This decomposition was originally given for the squared-error loss

[**krogh_NeuralNetworkEnsembles_1995**] and the KL-divergence [**todo**]. We will treat this quantity in detail in the rest of this chapter. Right now, note that the ambiguity term can be interpreted as a measure of *variance* where individual distances are measured by the loss ℓ. For the squared-error loss and the arithmetic mean combiner, this is exactly the "statistical" variance over the members $\frac{1}{M}\sum_{i=1}^{M}\left(q_i - \frac{1}{M}\sum_{i=1}^{M} q_i\right)^2$. For other loss functions, this yields other well-known quantities (see definition 1.7.2).

4.1.3. Impurity

KohaviWolpert give a bias-variance decomposition of the $0/1$-loss as follows.

$$\begin{aligned}\mathbb{E}\left[\ell_{0/1}\left(Y, q(X)\right)\right] &= \mathbb{E}\left[\frac{1}{2}\left(\sigma_x^2 + \text{bias}(q) + \text{var}(q)\right)\right] \\ \text{for } \text{bias}(q) &= \sum_y \left(\mathbb{P}\left[y^\star = y \mid x\right] - \mathbb{P}\left[q = y \mid x\right]\right)^2 \\ \text{var}(q) &= 1 - \sum_y \mathbb{P}\left[q = y \mid x\right]^2 \\ \sigma^2 &= 1 - \sum_y \mathbb{P}\left[y^\star = y \mid x\right]\end{aligned}$$

Note that the variance and noise terms are of the form of the Gini impurity (see 3.1.2) and can be interpreted as measures of impurity. **kunchevaWhittaker** take inspiration from this variance term and proceed as follows. Instead of considering the impurity over positive or negative labels, they instead consider the impurity over labels for which the ensemble is correct ($\tilde{y} = +1$) or incorrect ($\tilde{y} = -1$), respectively. As such, the probability is not with respect to the distribution of labels but the distribution Θ of members in the ensemble. Let the number of correct classifiers on an example be $m_+(x) =_{\text{def}} \sum_j^M \mathbb{1}\left[q_j(x) = y(x)\right]$ where $y(x)$ is the true label of x. This yields

$$\text{var}'_x =_{\text{def}} 1 - \sum_{\tilde{y}} \mathbb{P}_\Theta\left[q = \tilde{y} \mid x\right]^2$$

Averaging this over the entire training dataset, this can be understood as a measure of diversity. More precisely, it is the impurity as measured by the Gini impurity of the predictions of the ensemble members.

4.1.4. Entropy

CunninghamCarney2000 propose the entropy between member predictions as a diversity measure. For a single point, it is given as

$$\sum_{y\in\{-1,+1\}} -\mathbb{P}_\Theta\left[y \mid x\right] \log \mathbb{P}_\Theta\left[y \mid x\right]$$

where, for an ensemble, $\mathbb{P}_\Theta\left[y \mid x\right] \approx \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}\left[q_i(x) = y\right]$. **ShippKuncheva2002** propose a target-dependent measure that is reminiscent of the entropy. With $m_+(x) = \sum_{j=1}^{M} \mathbb{1}\left[q_i(x) = y(x)\right]$:

$$\frac{1}{M - \lceil \frac{M}{2} \rceil} \min\{m_+(x), M - m_+(x)\}$$

4.1.5. Covariance

Covariance of members In section 2.3, we have already seen evidence that the covariance between members is an essential factor to ensemble performance. Indeed, the notion of covariance and uncorrelatedness has been a guiding thought in the literature [**didaci_DiversityClassifierEnsembles_2013**, **brown_ManagingDiversityRegression_2005**, **buschjager_GeneralizedNegativeCorrelation_2020**]. We will now introduce an exact decomposition of the ensemble error for the squared-error loss that includes the average covariance between member predictions.

Theorem 4.1.1 *(Bias-Variance-Covariance decomposition **??**) It holds that*

$$\mathbb{E}_{(X,Y),D}\left[(y-\bar{q})^2\right] = \overline{bias}^2 + \frac{1}{M}\overline{var} + \left(1-\frac{1}{M}\right)\overline{covar}$$

$$\text{for } \overline{bias} =_{def} \frac{1}{M}\sum_{i=1}^{M}(\mathbb{E}_D\left[q_i\right]-y)$$

$$\overline{var} =_{def} \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_D\left[(q_i-\mathbb{E}_D\left[q_i\right])^2\right]$$

$$\overline{covar} =_{def} \frac{1}{M(M-1)}\sum_{i\neq j}\mathbb{E}_D\left[(q_i-\mathbb{E}_D\left[q_i\right])(q_j-\mathbb{E}_D\left[q_j\right])\right]$$

Proof. We begin by applying the bias-variance decomposition (theorem **??**) to the ensemble model $\bar{q}$.

$$\mathbb{E}\left[\ell(y,\bar{q})\right] = \mathbb{E}\left[\ell(Y,y^\star)\right] + \mathbb{E}\left[\ell(\bar{q}^\star,y^\star)\right] + \mathbb{E}\left[\ell(\bar{q}^\star,\bar{q})\right]$$

For the bias term, it holds that

$$\begin{aligned}\text{bias}(\bar{q}) = \ell(\bar{q}^\star,y^\star) &= (\mathbb{E}_D\left[\bar{q}\right]-y^\star)\\ &= \mathbb{E}_D\left[\left(\left(\frac{1}{M}\sum_{i=1}^{M}q_i\right)-y^\star\right)^2\right]\\ &= \left(\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_D\left[q_i-y^\star\right]\right)^2\\ &= \overline{\text{bias}}^2\end{aligned}$$

The variance of the ensemble can be decomposed into terms describing variances of individual members and covariances between members. [1]

$$\begin{aligned}\text{Var}\left(\bar{q}\right) &= \mathbb{E}_D\left[\ell(\bar{q},\bar{q}^\star)\right]\\ &= \mathbb{E}_D\left[(\bar{q}-\mathbb{E}_D\left[\bar{q}\right])^2\right]\\ &= \mathbb{E}_D\left[\left(\frac{1}{M}\sum_{i=1}^{M}q_i - \mathbb{E}_D\left[\frac{1}{M}\sum_{i=1}^{M}q_i\right]\right)^2\right]\\ &= \frac{1}{M^2}\sum_{i=1}^{M}\mathbb{E}_D\left[(q_i-\mathbb{E}_D\left[q_i\right])^2\right] + \frac{1}{M^2}\sum_{j\neq i}\mathbb{E}_D\left[(q_i-\mathbb{E}_D\left[q_i\right])(q_j-\mathbb{E}_D\left[q_j\right])\right]\end{aligned}$$

1: Using that

$$\left(\sum_{i=1}^{n}a_i\right)^2 = \sum_{i=1}^{n}a_i + \sum_{j\neq i}a_ia_j$$

Rearraging the coefficients yields the form of the theorem. [2] □

2: The coefficients are rearranged using

$$\frac{1}{M^2} = \left(1 - \frac{1}{M}\right) \frac{1}{M(M-1)}$$

The Bias-Variance-Covariance decomposition can be interpreted as a decomposition into characteristics of individual learners (mean bias and variance), plus a quantity describing the interactions between diferent learners. Again, one can see that ensemble performance profits if members are uncorrelated. However, this decomposition is only given for the squared error and the arithmetic mean combiner.

A covariance decomposition for classification **Didaci2013** builds on the decomposition given by **KohaviWolpert** (see equation **??**) and, similar to the derivation of the covariance decomposition for the squared-error loss, derive average bias, average variance and covariance terms. They decompose the bias and variance terms of the ensemble as follows, writing $P\left[y_i\right] =_{\text{def}} P\left[Y = y_i \mid \mathbf{x}\right]$ and $\hat{\mathbf{P}}_q\left[y_i\right] =_{\text{def}} P[q = y_i]$.

$$\begin{aligned}\operatorname{bias}(\bar{q}) &= \frac{1}{2}\sum_{y_i}\left(P\left[y_i\right] - \frac{1}{\sqrt{N}}\sum_j \hat{\mathbf{P}}_{q_j}\left[y_i\right] + \frac{1}{\sqrt{N}}\sum_j \hat{\mathbf{P}}_{q_j}\left[y_i\right] - \hat{\mathbf{P}}_{\bar{q}}\left[y_i\right]\right)^2 \\ &= \overline{\text{bias}} + b,\end{aligned}$$

$$\begin{aligned}\operatorname{var}(\bar{q}) &= \frac{1}{2}\left(1 - \frac{1}{N^2}\sum_{j,y_i}\hat{\mathbf{P}}^2_{q_j}\left[y_i\right] + \frac{1}{N^2}\sum_{j,y_i}\hat{\mathbf{P}}^2_{q_j}\left[y_i\right] - \sum_{y_i}\hat{\mathbf{P}}_{\bar{q}}\left[y_i\right]^2\right) \\ &= \frac{1}{N}\overline{\text{var}} + v,\end{aligned}$$

However, there are remaining terms b and v and, as the authors state, their interpretation is yet unclear.

4.1.6. Relationship between Ambiguity and Covariance

Ambiguity as a generalised notion of variance, and the well-known covariance decomposition. Can see that strictly speaking, these belong to the "variance" section.

We have now seen two approaches to expressing the ensemble generalisation error:

- In terms of *covariance* as in the bias-variance-covariance decomposition of theorem 4.1.1. A very similar notion also appears in when considering ensemble margins as in **??**
- In terms of variation around a centroid with respect to some loss function as in the diversity decomposition of theorems **??** and **??**. For the squared-error loss, this is exactly the variance. The generalisations to Bregman divergences and arbitrary loss functions can be understood as generalised measures of variance. [3]

3: For any loss function:

$$\mathbb{E}_D\left[\text{LE}(q^\star, q)\right]$$

For the squared-error loss:

$$\mathbb{E}_D\left[(q - \mathbb{E}_D\left[q\right])^2\right] = \mathbb{E}_D\left[\ell(q^\star, q)\right]$$

For Bregman divergences, this is analogous to the *Bregman information*, see definition 1.7.2:

$$\mathbb{E}_D\left[B_\phi\left(q^\star, q\right)\right]$$

Inspecting the proofs of theorems 4.1.1 and **??**, one can observe that in both cases, one begins with a measure of variance. Based on this, a covariance expression is then extracted, enabled by the definition of variance involving a square, producing cross-terms. In other words, *the covariance term is an artifact of the squared-error loss*. The notion of variance is more general.

For the case of squared-error loss, **[brownManaging]** relate the ambiguity decomposition (see **??**) and the bias-variance-covariance decomposition (theorem **??**) directly.

$$\mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}(y - q_i)^2 - \frac{1}{M}\sum_{i=1}^{M}(\bar{q} - q_i)^2\right] = \overline{\text{bias}}^2 + \frac{1}{M}\overline{\text{var}} + \left(1 - \frac{1}{M}\right)\overline{\text{covar}}$$

which yields

$$\begin{aligned}\mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}(y-q_i)^2\right] &= \overline{\text{bias}}^2 + \Omega \\ &= \Omega - \left(\frac{1}{M}\overline{var} + \left(1-\frac{1}{M}\right)\overline{\text{covar}}\right) \\ \text{for } \Omega =_{\text{def}} \overline{\text{var}} + \frac{1}{M}\sum_{i=1}^{M}&\left(\mathbb{E}\left[q_i\right] - \mathbb{E}\left[\bar{q}\right]\right)^2\end{aligned}$$

The first quantity describes the average member error, the second the diversity of the ensemble. The appearance of Ω in both terms illustrates that there is a trade-off between individual member error and diversity. In other words, one cannot maximise diversity without also affecting other parts of the ensemble error. Vice versa, optimising for other components of the error will also affect diversity.

4.2. Generalising Ambiguity

In section 1.5, we have derived a generalised bias-variance decomposition by considering the effect on the loss of using a random variable instead of its non-random centroid. For the variance, this would be the expected difference in loss between using a model dependent on the training input D and the expected model, where the expectation is over D. Consider now an ensemble of learners, in which each learner is constructed according to a random parameter Θ. Similar to bias and variance, we may consider the distribution of models over Θ around a central model $\bar{q}$ that is non-random with respect to Θ. The expected *effect* of using the central model instead of some single member is expressed as follows.

$$\mathbb{E}_\Theta\left[\ell(Y, q_\Theta) - \ell(Y, \bar{q}))\right] \approx \frac{1}{M}\sum_{i=1}^{M}\ell(Y, q_i) - \ell(Y, \bar{q})$$

Using the same strategy as for the bias-variance-effect decomposition of theorem **??**, we can use this to formulate a decomposition of the generalisation error with respect to the central model.

Theorem 4.2.1 *(Ambiguity-Effect decomposition [**todo**]) For any loss function L, target label Y, ensemble members $q_1, \ldots, q_M$ with combiner $\bar{q}$*

$$\begin{aligned}\ell(Y, \bar{q}) &= \frac{1}{M}\sum_{i=1}^{M}\ell(Y, q_i) - \underbrace{\left(\frac{1}{M}\sum_{i=1}^{M}\ell(Y, q_i) - \ell(Y, \bar{q})\right)}_{\textit{Ambiguity-Effect / Ensemble Improvement}} \\ &= \frac{1}{M}\sum_{i=1}^{M}\ell(Y, q_i) - \frac{1}{M}\sum_{i=1}^{M}LE(\bar{q}, q_i)\end{aligned}$$

If the ensemble combiner $\bar{q}$ is chosen to be the central model, this is a decomposition of the ensemble generalisation error and a generalisation of the ambiguity decomposition described in section **??**. We can then also find an intuitive interpretation of ambiguity-effect: It is the effect of ensembling on the error. If we consider the members to be constructed according to a parameter Θ, a reasonable measure of the member performance is its loss in expectation over the parameter distribution:

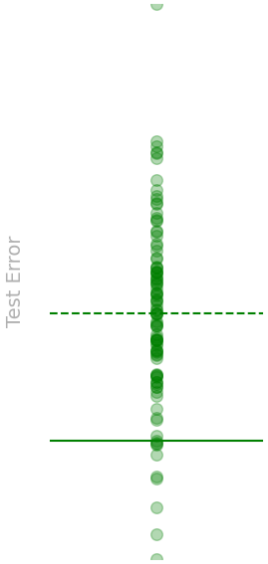

Figure 4.1.: The spread of individual tree predictions in a random forest ensemble. Glyphs correspond to test errors of individual trees. The dashed line is the average test error of individual trees $\frac{1}{M}\sum_1^M L(y, q_i)$. The solid line is the test error of the ensemble $L(y, \bar{q})$. The difference between these values is the *ensemble improvement* or *ambiguity-effect*. (TODO resolve double terms)

$\mathbb{E}_\Theta[L(Y, q(X;\Theta))] \approx \frac{1}{M}\sum_{i=1}^{M} L(Y, q(X;\Theta_i)$. What we gain or lose from using an ensemble $\bar{q}$ over just a single member model is exactly measured by ambiguity-effect. Due to this, this quantity is also known as *ensemble improvement* [**theisen**, **and others?**].

Similar to variance, ambiguity and ambiguity-effect are measures of spread. Variance measures the spread of training error across models trained with different draws of the training dataset D around a model that is a centroid with respect to the distribution of D. Similarly, ambiguity measures the spread of individual member model errors around a model that is centroid with respect to the distribution of Θ, namely the combiner $\bar{q}$. In case of squared loss, this is indeed the statistical variance around the arithmetic mean. For other losses, this is a different quantity.

4.3. The Diversity-Effect decomposition

The ambiguity decomposition divides the ensemble error into the average member error and the variance among members. How does this relate to the well-known bias-variance decomposition? Indeed, nothing prevents us from applying the bias-variance decomposition of theorem 1.5.1 to the error of an individual member, resulting in a decomposition into *average* bias, *average* variance and expected ambiguity, which is also referred to as *diversity* [**wood23**].

$$\begin{aligned}\mathbb{E}[\ell(y,\bar{q})] &= \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\ell(y,q_i)\right] + \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(\bar{q},q_i)\right] \\ &= \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(q_i^\star,y^\star)\right] + \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(q_i^\star,q_i)\right] + \mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(\bar{q},q_i)\right]\end{aligned}$$

In summary, the decomposition is given in the following theorem. Note that it holds for *any* loss function.

Theorem 4.3.1 *(Bias-Variance-Diversity-Effect decomposition)*

$$\begin{aligned}\mathbb{E}_{(X,Y),D}[\ell(Y,\bar{q})] = &\underbrace{\mathbb{E}_Y\left[\ell(y,y^\star)\right]}_{\textit{noise}} \\ &+ \underbrace{\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_Y\left[\ell\left(Y,q_i^*\right)-\ell\left(Y,Y^*\right)\right]}_{\overline{\textit{bias-effect}}} \\ &+ \underbrace{\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_D\left[\mathbb{E}_Y\left[\ell\left(Y,q_i\right)-\ell\left(Y,q_i^*\right)\right]\right]}_{\overline{\textit{variance-effect}}} \\ &- \underbrace{\mathbb{E}_D\left[\mathbb{E}_Y\left[\frac{1}{M}\sum_{i=1}^{M}\left[\ell\left(Y,q_i\right)-\ell(Y,\bar{q})\right]\right]\right]}_{\textit{diversity-effect}}\end{aligned}$$

Figure 4.2.: foo bar baz, avg bias, variance constant, diversity increases

Unchanged bias and reduction in variance In section 2.3 we gave arguments for how in specific cases, the ensemble bias equals the bias of any member. We now have the tools to show this in a more general manner [**wood23**].

For the ensemble bias, application of the ambiguity-effect decomposition (see **??**) to a set of centroid models $q_i^\star$ yields:

$$\underbrace{\mathrm{LE}(y, \bar{q}^\star)}_{\text{ens. bias}} = \underbrace{\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(y, q_i^\star)}_{\text{avg. bias}} - \underbrace{\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(\bar{q}^\star, q_i^\star)}_{\Delta}$$

For the ensemble variance, application of the diversity-effect decomposition (see **??**) while substituting $y \leftarrow \bar{q}^\star$:

$$\underbrace{\mathbb{E}_D\left[\mathrm{LE}(\bar{q}^\star, \bar{q})\right]}_{\text{ens. var.}} = \underbrace{\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(\bar{q}^\star, q_i^\star)}_{\Delta} + \underbrace{\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_D\left[\mathrm{LE}(q_i^\star, q_i)\right]}_{\text{avg. var.}} - \underbrace{\mathbb{E}_D\left[\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(\bar{q}, q_i)\right]}_{\text{diversity}}$$

Due to Lemma (**??**) which states that in homogeneous ensembles $q_i^\star = q_j^\star = \bar{q}^\star$, we can conclude that $\Delta = 0$.

Corollary 4.3.2 *For homogeneous ensembles, we can conclude the following:*

- *The ensemble bias is equal to the average member bias:*

$$\Delta = 0 \quad \rightarrow \quad LE(y, \bar{q}^\star) = \frac{1}{M}\sum_{i=1}^{M} LE(y, q_i^\star)$$

- *Diversity is a component of ensemble variance. The other component is the average member variance. In other words, ensemble variance reduction is measured exactly by diversity.*
- *Diversity is bounded from above by the average member variance.*

4.4. Diversity for Bregman Divergences

Note that bias-effect, variance-effect and ambiguity-effect are all of the form $\mathbb{E}\left[\mathrm{LE}(\circ, \square)\right]$ and depend directly on the target label Y. With variance-effect, we have captured the *effect* of variations between different training datasets) on the prediction error. With ambiguity-effect, we have captured the effect of variations between the different member models on the prediction error. We have already seen that for the squared error, the variance-effect coincides with familiar notion of "statistical" variance between the predictions.

We will now define a class of losses for which the loss-effects reduce to the loss between the two objects. This class covers many widely used loss functions and thus allows us to formulate a unified bias-variance-decomposition.

Bregman divergences have the one key property that their loss-effect terms collapse. That is, with bregman divergences, we have, for the the proper choice of Z, Z':

$$\mathbb{E}\left[\mathrm{LE}(Z', Z)\right] = B_\phi\left(Z', Z\right)$$

This is given by the following two lemmas.

Lemma 4.4.1 *([**pfau**], Theorem 0.1 (b)) Let the generator $\phi : \mathcal{S} \to \mathbb{R}$ be a strictly convex, differentiable function. Let Y be a random variable on $\mathcal{S}$. Then, for any $q \in \mathcal{S}$, it holds that*

$$B_\phi\left(y^\star, q\right) = \mathbb{E}\left[B_\phi\left(Y, q\right) - B_\phi\left(Y, y^\star\right)\right]$$

if $y^\star$ is the right Bregman centroid.

This shows that bias-effect collapses to bias for Bregman divergences:

$$B_\phi\left(y^\star, q^\star\right) = \mathbb{E}\left[B_\phi\left(Y, q^\star\right) - B_\phi\left(Y, y^\star\right)\right]$$

Lemma 4.4.2 *(★, Generalised from [**ref:wood23**]) Let $\mathbf{y}$ be a random vector. Let $\mathbf{q}_Z$ be a random vector dependent on another random variable Z. Then it holds that*

$$B_\phi\left(q^\star, q\right) = \mathbb{E}_Y\left[B_\phi\left(Y, q\right) - B_\phi\left(Y, q^\star\right)\right]$$

if $q^\star$ is the left Bregman centroid.

Proof.

$$\mathbb{E}_{Y,Z}\left[B_\phi\left(y, q\right) - B_\phi\left(y, q^\star\right)\right] = \ldots$$

□

This shows that variance- and ambiguity-effect collapse to variance and ambiguity. The variance (in the bias-variance sense) is the variance of the estimates q_D dependent on D with respect to different realisations of D around its D-centroid.

$$\text{For } q^\star = \mathcal{C}_D\left[q_D\right]\text{:} \quad B_\phi\left(q^\star, q\right) = \mathbb{E}\left[B_\phi\left(Y, q^\star\right) - B_\phi\left(Y, q\right)\right]$$

Ambiguity/Diversity is the variance of the estimates q_Θ dependent on Θ with respect to different realisations of Θ around its centroid.

$$\text{For } \bar{q} = \mathcal{C}_\Theta\left[q_\Theta\right]\text{:} \quad B_\phi\left(\bar{q}, q\right) = \mathbb{E}\left[B_\phi\left(Y, \bar{q}\right) - B_\phi\left(Y, q\right)\right]$$

This yields a generalised bias-variance-diversity decomposition for bregman divergences as a special case of the corresponding effect decomposition **??**.

Theorem 4.4.3 *(Bias-Variance-Diversity decomposition for Bregman divergences)*

$$\begin{aligned}\mathbb{E}_{(X,Y),D}\left[B_\phi\left(Y, q\right)\right] = &\underbrace{\mathbb{E}_{Y|X}\left[B_\phi\left(Y, y^\star\right)\right]}_{noise} \\ &+ \underbrace{\frac{1}{M}\sum_{i=1}^{M} B_\phi\left(\overline{\mathbf{Y}}, \mathbf{q}_i^*\right)}_{\overline{bias}} \\ &+ \underbrace{\frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_D\left[B_\phi\left(\mathbf{q}_i^*, \mathbf{q}_i\right)\right]}_{\overline{variance}} \\ &- \underbrace{\mathbb{E}_D\left[\frac{1}{M}\sum_{i=1}^{M} B_\phi\left(\overline{\mathbf{q}}, \mathbf{q}_i\right)\right]}_{diversity}\end{aligned}$$

The ensemble combiner for Bregman divergences ...

Ensemble improvement for Bregman Divergences In section 2.3, we have used Jensen's inequality to show that the ensemble improvement is non-negative for some cases.

$$\mathbb{E}_\Theta\left[\ell(q_\Theta(X), Y)\right] - \ell(\mathbb{E}_\Theta\left[q_\Theta(X)\right], Y) \geq 0$$

It is evident, that the Jensen gap is but a special case of ambiguity-effect (**??**) for $\bar{q} =_{\text{def}} \frac{1}{M}\sum_{i=1}^{M} q_i$ and convex loss functions. This shows that the ensemble loss is always smaller-equal than the expected member loss, but *only* if the ensemble output is actually produced by an arithmetic mean.

However, it can not be assumed from the outset that the arithmetic mean is the best ensemble combiner. Indeed, for the cross-entropy loss, **abe** proceed to note that the Jensen gap corresponds to a form that is "not immediately recognizable". Although they do find an interpretation of it, it is still necessarily dependent on the outcome Y . As illustrated in 4.4 on Bregman divergences, it seems reasonable to define the ensemble combiner in accordance to the Bregman divergence, i.e. to be the *dual* expectation $\mathscr{C}_\Theta\left[q_\Theta\right]$. Non-negativity is then easily shown since in that case ambiguity-effect reduces to ambiguity (see **??**)

In fact, for the case of cross-entropy, [**wood23**] show that the ambiguity term is still nonnegative, i.e. that the arithmetic mean combiner does not hurt performance.

$$B_\phi\left(\bar{q}, q\right) = \mathbb{E}\left[B_\phi\left(Y, \bar{q}\right) - B_\phi\left(Y, q\right)\right] \quad \text{for } \bar{q} = \mathscr{C}_\Theta\left[q_\Theta\right]$$

and the value of any Bregman divergence is always non-negative. Further, ambiguity is now independent of the outcome.

This shows that for any Bregman divergence, ensembling using the combiner implied by the divergence can not hurt performance. Second, we obtain an intuitive measure of ensemble improvement. Third, this ensemble improvement appears in an exact decomposition of the ensemble generalisation error.

4.5. Diversity for the $0/1$-Loss

Bregman divergences are certainly useful for regression tasks. Further, divergences such as the KL-divergence are useful for estimating class *probabilities* and indeed,

classification tasks can be approached by selecting the class with the highest estimated probability. In other settings, however, we are mainly concerned with wether the correct class is assigned or not.

Definition 4.5.1 *The 0/1-loss between two outcomes is*

$$\ell_{0/1}(Y, Y') =_{def} \mathbb{1}[Y \neq Y']$$

Definition 4.5.2 *(Majority/Plurality vote combiner) For a k-class classification problem, the majority vote combiner is defined as*

$$\bar{q}(X) = \arg\min_{z \in [k]} \mathbb{E}_\Theta \left[\ell_{0/1}(z, q_\Theta)\right]$$

Definition 4.5.3 *For a distribution of members constructed according to input data $D = (D_1, \ldots, D_M)$ and parameters $\Theta = (\Theta_1, \ldots, \Theta_M)$, the ratio of incorrect ("wrong") members is*

$$W(X, Y) =_{def} \mathbb{E}_{D,\Theta}\left[\ell_{0/1}(Y, q_{D,\Theta}(X))\right] \approx \frac{1}{M}\sum_{i=1}^{M} \ell_{0/1}(Y, q_i(X))$$

As with other variables, we sometimes omit explicitly stating the dependence on (X, Y). Further, we write $W_\Theta =_{def} \mathbb{E}_\Theta\left[\ell_{0/1}(Y, q_{D,\Theta})\right]$. For the complement, we write $\overline{W} =_{def} 1 - W$

The ratio of incorrect members in expectation over all examples is equal to the error rate of an average member.

$$\mathbb{E}_{(X,Y)}[W] = \mathbb{E}_{(X,Y)}\left[\mathbb{E}_{D,\Theta}\left[\ell_{0/1}(Y, q_{D,\Theta}(X))\right]\right] = \mathbb{E}_{D_\Theta}\left[\mathbb{E}_{(X,Y)}\left[\ell_{0/1}(Y, q_{D,\Theta}(X))\right]\right] \tag{4.1}$$

Using Markov's inequality, we can readily upper-bound the error of the ensemble in terms of expected errors of the members **[theisen]** [4] .

$$0 \leq \mathbb{E}\left[\ell_{0/1}(Y, \bar{q})\right] = \mathbb{P}[W \geq \kappa] \leq \mathbb{P}\left[W \geq \frac{1}{2}\right] \leq 2\mathbb{E}[W]$$

While there exist examples for which this upper bound is tight **??**, it is reasonable to suspect that the ensemble being worse by a factor of two is only a pathological case and not relevant for practise.

4.5.1. Binary classification

The setting of binary classification using the 0/1-loss allows us to clearly distinguish two cases. An example may be classified correctly, in which case it does not contribute to the overall error. Otherwise, it is classified incorrectly and contributes exactly 1. While diversity-effect (in expectation over (X, Y)) is non-negative for ensembles of weak-learners (see theorem **??**), there may still be example-outcome pairs that contribute negatively to the expectation, as illustrated in figures **??** and **??**

Lemma 4.5.1 *([**kuncheva**]) For a classification problem with $k = 2$ classes, let $y, \bar{q} \in$*

The proper term here would actually be *plurality* vote since, for a class to win the vote, it is required to have more than $\frac{1}{k}$ votes. Strictly speakin, a *majority* vote win requires the majority of all votes, i.e. more than $\frac{1}{2}$. For $k = 2$, majority and plurality voting is equivalent.

4: Markov's inequality states that for a nonnegative random variable X and $a > 0$

$$\mathbb{P}[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$

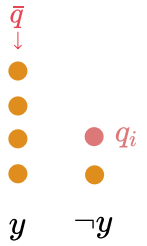

Figure 4.3.: Example of the effect of a member's vote q_i on the diversity on a point for which the ensemble majority vote is correct. Example where q_i has positive contribution to the diversity effect term, i.e. $\ell_{0/1}(y, q_i) - \ell_{0/1}(y, \bar{q}) = 1$. The member q_i is incorrect but due to the discreteness of the majority vote combiner, the ensemble performance does not suffer – unless the majority vote is tipped. Any correct vote while the ensemble already is correct is effectively "wasted" and incorrect votes correspond to diversity.

Figure 4.4.: Example where q_i has negative contribution to the diversity effect term, i.e. $\ell_{0/1}(y, q_i) - \ell_{0/1}(y, \bar{q}) = -1$. Any further incorrect vote while the ensemble is already incorrect would be wasted. The negative effect here eventually results in the 0/1-loss of 1.

$\{-1, 1\}$. *It then holds that*

$$\frac{1}{M}\sum_{i=1}^{M}\left[\ell_{0/1}(y, q_i) - \ell_{0/1}(y, \bar{q})\right] = (y \cdot \bar{q})\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(\bar{q}, q_i) \in \{-1, 0, 1\}$$

Proof. Let $y, \bar{q} \in \{-1, 1\}$.

- Assume the ensemble is correct, i.e. $y = \bar{q}$. Then $\ell_{0/1}(y, \bar{q}) = 0$ and the left-hand-side equals $\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(y, q_i) = \frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(\bar{q}, q_i)$. Further, $y \cdot \bar{q} = 1$.
- Assume the ensemble is incorrect, i.e. $y \neq \bar{q}$. Then $y \cdot \bar{q} = -1$ and, for the left-hand-side, we can write

  $$\frac{1}{M}\sum_{i=1}^{M}\left[\ell_{0/1}(y, q_i)\right] - 1 = -\left(1 - \frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(y, q_i)\right) = -\left(\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(\bar{q}, q_i)\right)$$

  using that, since $y \neq \bar{q}$, $\left(1 - \frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(y, q_i)\right) = \frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(\bar{q}, q_i)$.

□

This shows that, for binary classification under 0/1-loss, diversity-effect can be decomposed exactly between points that contribute positively or negatively to the overall loss. Starting from the ambiguity-effect decomposition **??**:

We can divide the range of X into two disjoint subsets. Let X_+ be the examples on which the ensemble is correct. Ambiguity on these points has a decreasing effect on the overall ensemble error. Let X_- be the examples on which the ensemble is incorrect. Ambiguity on these points has an increasing effect on the overall ensemble error. This yields a decomposition into *good* and *bad* diversity.

An intuition of this is also that of "wasted votes": Under the majority vote combiner, for the ensemble to be correct, we require only at least half of the members to be correct. Any higher ratio of correct ensemble members does not improve the ensemble performance on this point and these can be seen as "wasted". Likewise, the ensemble is incorrect if not more than half of the members are correct. Any positive votes do not influence the ensemble improvement and can be considered "wasted".

Corollary 4.5.2 *([**kuncheva**]) For a classification problem with $k = 2$ classes, let $y, \bar{q} \in \{-1, 1\}$. It then holds that*

$$\begin{aligned}\mathbb{E}_X\left[L(Y, \bar{q})\right] = \mathbb{E}_X&\left[\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(Y, q_i)\right] \\ &\underbrace{- \mathbb{E}_{X_+}\left[\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(\bar{q}, q_i)\right]}_{\text{"good" diversity}} \\ &\underbrace{+ \mathbb{E}_{X_-}\left[\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(\bar{q}, q_i)\right]}_{\text{"bad" diversity}}\end{aligned}$$

As with any ambiguity decomposition, there is a tradeoff between average member error and diversity. Here, however, diversity is not always beneficial. On points where the ensemble is incorrect, disagreements have a negative effect on the overall ensemble error. In other words, for majority vote ensembles, diversity is only beneficial *on points at which the ensemble can actually afford to be diverse*.

Further, from corollary **??**, one can already see that the ensemble improvement (i.e. diversity-effect) in binary classification problems is only non-negative if good diversity outweighs bad diversity.

Good and bad diversity can be expressed solely in terms of the ratio of incorrect members.

Lemma 4.5.3 ⋆ *Let y be the true outcome for a given example. Let $\neg y$ be an outcome that is not y. Write $\ell_{0/1}(q_i, \neg y) =_{def} \sum_{k \neq y} \ell_{0/1}(q_i, k)$ for the indication whether q_i is incorrect. Then the following identities hold.*

$$\mathbb{E}_{X_+}\left[\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(q_i, \bar{q})\right] = \mathbb{E}_{X_+}\left[W_1^M\right]$$

$$\mathbb{E}_{X_-}\left[\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(q_i, \bar{q})\right] = \mathbb{E}_{X_-}\left[\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(q_i, \neg y)\right] = \mathbb{E}_{X_-}\left[1 - W_1^M\right]$$

Analogous equalities hold in expectation over member parameter Θ.

4.5.2. Weak Learners

A common idea is that, in order for ensembling to be effective, the performance of individual members must not be too bad. In the remainder of this section, we will review some assumptions that imply ensemble effectivity and show that they are in fact tightly related.

Definition 4.5.4 *(Weak learner) need to check / reformulate this*

Theorem 4.5.4 *([**wood23**]) In an ensemble of weak learners, diversity-effect is non-negative:*

$$\mathbb{E}_D\left[\mathbb{E}_Y\left[\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}(Y, q_i) - \ell_{0/1}(Y, \bar{q})\right]\right] \geq 0$$

Proof. ... essentially using jury theorem □

However, this is not a sufficient condition, as example **??** shows.

4.5.3. Competence and Diversity-Effect

[**theisen**] consider the question of ensemble improvement under the 0/1-loss. They define an assumption on the ratio of inccorect members.

Definition 4.5.5 *(2-competence, [**theisen**]) An ensemble is 2-competent iff*

$$\forall t \in \left[0, \frac{1}{2}\right] : \mathbb{P}_{(X,Y)}\left[W(X,Y) \in \left[t, \frac{1}{2}\right]\right] \geq \mathbb{P}_{(X,Y)}\left[W(X,Y) \in \left[\frac{1}{2}, 1-t\right]\right]$$

The condition is illustrated in figure **??**. It is used to show two results:

- In 2-competent ensembles, diversity-effect is non-negative. [5]
- bounds on the generalisation error (see **??**)

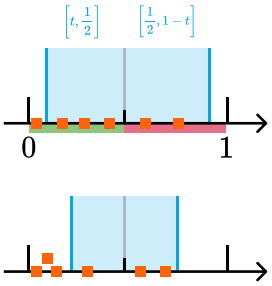

Figure 4.5.: Illustration for the competence condition **??** for binary classification. Red squares correspond to pairs (X, Y) from the joint distribution of examples and outcomes. For each of these pairs, the average/expected member error $W_\Theta(X, Y) \approx \frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}((, y), q_i)$ is the ratio of incorrect members. The center $\frac{1}{2}$ is the majority vote threshold. Informally, an ensemble is competent, if, for any two intervals defined by t left and right of the threshold, more examples are in the left part. For the upper example, this holds. For the lower example, even though many examples are classified correctly by many members, the ensemble is *not* competent.

5: Although not acknowledging the role of diversity-effect as a component of the ensemble generalisation error and thus referring to it only as "ensemble improvement", one of the main results of [**theisen**] is that, in competent ensembles, ensemble improvement (i.e. diversity-effect) is non-negative.

One can see that competence is essentially determined by the distribution of examples (X, Y) over the range of $W(X, Y)$ which is divided by the majority vote threshold $\frac{1}{2}$. We have already seen that, similarly, diversity-effect in its apparent form of good and bad diversity is determined by just the same characteristics. How are these two related? In this section, we will argue that non-negative diversity-effect is in fact equivalent to a notion of competence generalised to $k > 2$ classes. Unless otherwise noted, all expectations and probabilites are over the distribution of (X, Y).

While proving that 2-competence implies non-negative diversity-effect, **[theisen]** establish the following fact.

$$\text{ens. 2-competent} \leftrightarrow \mathbb{E}\left[W \mathbb{1}\left[W < \frac{1}{2}\right]\right] \geq \mathbb{E}\left[\overline{W} \mathbb{1}\left[\overline{W} \leq \frac{1}{2}\right]\right] \tag{4.2}$$

We can rearrange this into a more suggestive form. Recall that $W = \mathbb{E}_{D,\Theta}\left[\ell_{0/1}\left(Y, q_{D,\Theta}(X)\right)\right]$. We now split off the expectation over D and instead consider $W_\Theta = \mathbb{E}_\Theta\left[\ell_{0/1}\left(Y, q_{D,\Theta}(X)\right)\right]$. Rearranging the above and exploiting the linearity of expectation, we obtain

$$d =_{\text{def}} \mathbb{E}_{(X,Y),D}\left[W_\Theta \mathbb{1}\left[W_\Theta < \frac{1}{2}\right]\right] - \mathbb{E}_{(X,Y),D}\left[\overline{W_\Theta} \mathbb{1}\left[\overline{W_\Theta} \leq \frac{1}{2}\right]\right] \geq 0$$

The indicator functions are mutually exclusive and can thus be understood as a case distinction. With slight abuse of notation, where the expectations are only over the subset of the distribution for which the case condition holds, we can write

$$d = \begin{cases} \mathbb{E}\left[W_\Theta\right] & \leftrightarrow W_\Theta < \frac{1}{2} \\ \mathbb{E}\left[\overline{W_\Theta}\right] = \mathbb{E}\left[1 - W_\Theta\right] & \leftrightarrow \overline{W_\Theta} \leq \frac{1}{2} \end{cases}$$

For $k = 2$, the majority vote threshold is $\frac{1}{2}$ and thus the conditions correspond exactly to the ensemble being either correct or incorrect.

$$k = 2 \rightarrow \begin{cases} W_\Theta < \frac{1}{2} \leftrightarrow \bar{q}(X) = Y \\ \overline{W_\Theta} \leq \frac{1}{2} \leftrightarrow q(\bar{X}) \neq Y \end{cases} \tag{4.3}$$

Recalling the characterisation of good and bad diversity of lemma **??**, we can see that d is nothing else but the diversity-effect. This means that non-negative effect is exactly equivalent to 2-competence for a classification problem with $k = 2$ classes.

However, for $k > 2$, the equivalence in equation 4.3 is not given. While it is sufficient (a class with more than $\frac{1}{2}M$ votes will win any plurality vote), it is not necessary: a plurality vote can be won with less than $\frac{1}{2}M$ votes. Thus, there are ensembles which have non-negative diversity-effect (ensemble improvement) that are not 2-competent.

The key difference is that for $k > 2$, the voting threshold for a pair (X, Y) is no longer the same value for all examples. Since a class wins if and only if it has more votes than any other class, the voting threshold depends on the distribution of class votes, which is potentially different for any pair (X, Y). Nevertheless, there is still *a* classification threshold, namely the ratio of votes for the next-best class. Because we will be considering the ratio of incorrect votes as a basic quantity, we will now define

it from the reciprocal perspective:

$$\kappa(X,Y) = 1 - \max_{Z \neq Y} \mathbb{E}_\Theta \left[\mathbb{1}\left[q_\Theta = Z\right]\right]$$

and it holds that

$$\begin{aligned} W_\Theta < \kappa \quad &\leftrightarrow \quad \bar{q}(X) = Y \\ \overline{W_\Theta} \leq 1-\kappa \quad &\leftrightarrow \quad \bar{q}(X) \neq Y \end{aligned}$$

Definition 4.5.6 ⋆ *(k-competence) An ensemble is k-competent iff*

$$\forall t \in [0,1] : \mathbb{P}_{(X,Y)}\left[W \in [t,\kappa]\right] \geq \mathbb{P}_{(X,Y)}\left[W \in [1-\kappa, 1-t]\right]$$

for $\kappa =_{def} 1 - \max_{Z \neq Y} \mathbb{E}_\Theta \left[\mathbb{1}\left[q_\Theta = Z\right]\right]$.

For classification problems with $k = 2$, k-competence is exactly 2-competence as of definition **??** since the voting threshold is always $\frac{1}{2}$.

[**theisen**] showed that 2-competence implies non-negative diversity-effect. We now show that a very similar line of argument instead using k-competence actually holds in both directions.

Theorem 4.5.5 ⋆ *Consider an ensemble for a k-class classification problem. Then*

$$\textit{k-competence} \quad \leftrightarrow \quad \textit{diversity-effect} \geq 0$$

The main work lies in the following lemma, which is a generalised form of equation .

Lemma 4.5.6 ⋆ *(Generalised from [**theisen**]) For an increasing function f with $f(0) = 0$, it holds that*

$$\textit{k-competence} \;\leftrightarrow\; \mathbb{E}\left[f(W)\,\mathbb{1}\left[W < \kappa\right]\right] \geq \mathbb{E}\left[f(\overline{W})\,\mathbb{1}\left[\overline{W} \leq \kappa\right]\right]$$

where $\kappa =_{def} 1 - \max_{Z \neq Y} \mathbb{E}_\Theta \left[\mathbb{1}\left[q_\Theta = Z\right]\right]$.

Proof. We begin by observing that, for all $x \in [0,1]$

$$\begin{aligned} \mathbb{P}\left[W \in [x,\kappa]\right] \cdot\ \mathbb{1}\left[x \leq \kappa\right] &= \mathbb{P}\left[W\ \mathbb{1}\left[W < \kappa\right]\ \geq\ x\right] \\ \mathbb{P}\left[W \in [1-\kappa, 1-x]\right] \cdot\ \mathbb{1}\left[x \leq \kappa\right] &= \mathbb{P}\left[\overline{W}\ \mathbb{1}\left[\overline{W} \leq \kappa\right]\ \geq\ x\right] \end{aligned}$$

where the first factors on the left-hand-side appear in the definition of k-competence. Since W is nonnegative, using that $\mathbb{E}\left[X\right] = \int \mathbb{P}\left[X \geq x\right]\, dx$, we can conclude that, for any $x \in [0,1]$

$$\begin{aligned} (k\text{-comp.}) \;&\leftrightarrow\; \mathbb{P}\left[W\ \mathbb{1}\left[W < \kappa\right]\ \geq x\right]\ \geq\ \mathbb{P}\left[\overline{W}\ \mathbb{1}\left[\overline{W} \leq \kappa\right]\ \geq x\right] \\ &\leftrightarrow\; \mathbb{E}\left[W\ \mathbb{1}\left[W < \kappa\right]\right]\ \geq\ \mathbb{E}\left[\overline{W}\ \mathbb{1}\left[\overline{W} \leq \kappa\right]\right] \end{aligned}$$

$$\begin{aligned} &\mathbb{P}\left[W \in [1-\kappa, 1-x]\right]\ \mathbb{1}\left[x \leq \kappa\right] \\ =&\mathbb{P}\left[W \in [1-\kappa, 1-x]\right]\ \mathbb{1}\left[1-x > 1-\kappa\right] \\ =&\mathbb{P}\left[W\ \mathbb{1}\left[W > 1-\kappa\right]\ <\ 1-x\right] \\ =&\mathbb{P}\left[W\ \mathbb{1}\left[\overline{W} \leq \kappa\right]\ <\ 1-x\right] \\ =&\mathbb{P}\left[\overline{W}\ \mathbb{1}\left[\overline{W} \leq \kappa\right]\ \geq\ x\right] \end{aligned}$$

□

Using this, we can now directly prove theorem **??**.

Proof. (Theorem **??**, generalised from [**theisen**])

$$\begin{aligned}
0 &= \mathbb{E}\left[(W-1)\,\mathbb{1}\left[W \geq \kappa\right]\right] - \mathbb{E}\left[(W-1)\,\mathbb{1}\left[W \geq \kappa\right]\right] \\
&= \mathbb{E}\left[(W-1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[(1-W)\,\mathbb{1}\left[W \geq \kappa\right]\right] \\
&= \mathbb{E}\left[(W-1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[\overline{W}\,\mathbb{1}\left[\overline{W} < \kappa\right]\right]
\end{aligned}$$

Applying lemma 4.5.6 for $f = \mathrm{id}$ to the second term yields

$$\begin{aligned}
&\mathbb{E}\left[(W-1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[\overline{W}\,\mathbb{1}\left[\overline{W} < \kappa\right]\right] \\
\leq\ &\mathbb{E}\left[(W-1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[W\,\mathbb{1}\left[W < \kappa\right]\right]
\end{aligned}$$

The above already is nothing but the diversity-effect:

$$\begin{aligned}
0 \leq\ &\mathbb{E}\left[(W-1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[W\,\mathbb{1}\left[W < \kappa\right]\right] \\
=\ &\mathbb{E}\left[W\right] - \mathbb{E}\left[\,\mathbb{1}\left[W \geq \kappa\right]\right] \\
=\ &\mathbb{E}\left[W\right] - \mathbb{P}\left[W \geq \kappa\right]
\end{aligned}$$

The first term is the expected member error (see eq. 4.1) and the second term is the ensemble error (see eq. **??**). □

4.5.4. Bounds for Competent Ensembles

2-competence was used to show upper and lower bounds for the diversity-effect [**theisen**]. We show that, with minor adjustments, the same bounds can be derived from k-competence. Besides giving performance guarantees, these bounds are interesting due to that they are expressed in terms of disagreements between members, which until now we have only seen for Bregman divergences.

Theorem 4.5.7 *(Upper bound) In k-competent ensembles,*

$$\mathbb{E}\left[W\right] - \mathbb{P}\left[W \geq \kappa\right] \leq \mathbb{E}_{\rho,\rho'}\left[D(q_\rho, q_{\rho'})\right]$$

Proof. The upper bound does not make use of competence and therefore still holds for k-competent ensembles. The proof can be found in [**theisen**]. □

Theorem 4.5.8 *(Lower bound) In k-competent ensembles,*

$$\frac{2(k-1)}{k}\mathbb{E}\left[D(q_\rho, q_{\rho'})\right] - \frac{3k-4}{k}\mathbb{E}\left[W\right] \quad \leq \quad \mathbb{E}\left[W\right] - \mathbb{P}\left[W \geq \kappa\right]$$

Proof. Lemmas 3 and 4 hold without adjustments for k-competence and are shown in

[**cited-by-theisen**] and [**theisen**], respectively.

$$
\begin{aligned}
&\mathbb{P}\left[W \geq \kappa\right] \\
&\leq 2\mathbb{E}\left[W^2\right] && \text{(Lemma 4.5.9)} \\
&= 2\mathbb{E}_{\rho,\rho'}\left[L(q_\rho, q_\rho)\right] && \text{(Lemma 3 in [\textbf{theisen}])} \\
&= \frac{4(k-1)}{k}\left(\mathbb{E}\left[W\right] - \frac{1}{2}\mathbb{E}_{\rho,\rho'}\left[D(q_\rho, q_{\rho'})\right]\right) && \text{(Lemma 4 in [\textbf{theisen}])}
\end{aligned}
$$

Rearranging the terms yields the statement. □

Lemma 4.5.9 ⋆ *(Generalised from [**theisen**]) In k-competent ensembles it holds that*

$$\mathbb{P}\left[W \geq \kappa\right] \leq 2\mathbb{E}\left[W^2\right]$$

Proof. Note that

$$\mathbb{P}\left[W \geq \kappa\right] \leq 2\mathbb{E}\left[W^2\right] \;\leftrightarrow\; \mathbb{P}\left[W \geq \kappa\right] - 2\mathbb{E}\left[W^2\right] \geq 0$$

$$2\mathbb{E}\left[W^2\right] - \mathbb{P}\left[W \geq \kappa\right] = \mathbb{E}\left[(2W^2 - 1)\,\mathbb{1}\left[W \geq \kappa\right]\right]$$

We will aim to show that this above expression is nonnegative. The final inequality is due to applying lemma 4.5.6 to the second term.

$$
\begin{aligned}
&\mathbb{E}\left[2W^2\right] - \mathbb{P}\left[W \geq \kappa\right] \\
&= \mathbb{E}\left[2W^2\right] - \mathbb{E}\left[\,\mathbb{1}\left[W \geq \kappa\right]\right] \\
&= \mathbb{E}\left[(2W^2 - 1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[2W^2\,\mathbb{1}\left[W < \kappa\right]\right] \\
&\geq \mathbb{E}\left[(2W^2 - 1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[2\overline{W}^2\,\mathbb{1}\left[\overline{W} < \kappa\right]\right]
\end{aligned}
$$

$$
\begin{aligned}
&W \geq \kappa \\
&\leftrightarrow\; 1 - W < 1 - \kappa \\
&\leftrightarrow\; \overline{W} \leq 1 - \kappa
\end{aligned}
$$

Note that for $k \geq 2, \kappa > 1-\kappa$ and thus $\mathbb{E}\left[\,\mathbb{1}\left[\overline{W} \leq \kappa\right]\right] \geq \mathbb{E}\left[\,\mathbb{1}\left[\overline{W} \leq 1 - \kappa\right]\right]$, allowing us to continue

$$
\begin{aligned}
\ldots &\geq \mathbb{E}\left[(2W^2 - 1)\,\mathbb{1}\left[W \geq \kappa\right]\right] + \mathbb{E}\left[2\overline{W}^2\,\mathbb{1}\left[\overline{W} < 1 - \kappa\right]\right] \\
&= \mathbb{E}\left[1 - 4\overline{W} + 2\overline{W}^2\,\mathbb{1}\left[\overline{W} < 1 - \kappa\right]\right] \\
&\geq 0
\end{aligned}
$$

□

4.6. Dependency of diversity on outcomes

In the general case, diversity cannot be expressed independently of the outcome variable Y. If ℓ is a metric, then diversity-effect can be bounded from above by a target-independent term that is reminiscent of the diversity term for Bregman divergences introduced in theorem 4.4.3.

Lemma 4.6.1 ⋆ *Under a metric loss, diversity-effect is bounded from above by diversity.*

$$\ell \text{ metric} \quad \rightarrow \quad \mathbb{E}_{(X,Y),D}\left[\frac{1}{M}\sum_{i=1}^{M}\ell(Y,q_i)-\ell(Y,\bar{q})\right] \leq \mathbb{E}_{(X,Y),D}\left[\frac{1}{M}\sum_{i=1}^{M}\ell(\bar{q},q_i)\right]$$

Proof. For a metric $d : X \times X \rightarrow \mathbb{R}$ and $a, b, c \in X$, due to the triangle inequality and symmetry of d, it holds that

$$\begin{aligned} & d(a,b) - d(b,c) \\ & \leq d(a,c) + d(c,b) - d(b,c) \\ & = d(a,c) \end{aligned}$$

□

4.7. Diversity is a measure of model fit

"Sweet spot" of diversity. Review of previous experiments. Also that it'll be a tradeoff (can this be seen from ambiguity decomp?)

(some refs where people try to find these correlations, also for member selection (SFS, SBS)) We claim that the search for a diversity measure that is directly correlated with ensemble performance is misguided. As can be clearly seen from the diversity decomposition and also argued in other places, diversity is a dimension of model fit, just like bias and variance. As such, there will always be a tradeoff and just as with bias and variance, better diversity can not per se be expected to directly lead to better prediction performance. Further, diversity is *subtractive* to the ensemble error. We have seen that, under this tradeoff, encouraging diversity often comes with an equal increase in average member error (basin shape).

4.8. Diversity in Random Forests

4.9. Diversity in Neural Networks

...

4.9.1. Adversarial Robustness

... maybe move to outlook, probably too half-cooked if we put that in here?

5. Growth Strategies

Some general intuition on the idea behind encouraging diversity (always tradeoff with avg member error)

5.1. $0/1$-classification and Dynamic Random Forests

5.1.1. Guiding ensemble construction with example weights

(introduction ensemble construction, tree-by-tree with adaptive weights)

Now that we know exactly how correct and incorrect classifications affect the ensemble error, we will work towards leveraging this insight to find an ensemble construction scheme ...

On points for which the ensemble is correct, diversity is beneficial and corresponds directly to the average member error. This means that we might expect to see an increase in ensemble performance if disagreement on correct points is encouraged – as long as it does not cross the majority vote threshold. Indeed, in the perfect case, all examples would be correctly classified with a member error just below the majority vote threshold. This would result in an ensemble with large average member error but also with high diversity which mitigates the member error.

On points for whch the ensemble is incorrect, diversity hurts the ensemble performance. One could argue that, to minimise bad diversity, the average member error should be large, i.e. all members (instead of only some) should be driven to *mis*-classify the example. However, we conjecture that this would cause the ensemble construction procedure to "give up" on misclassified examples. We will consider a different strategy first. Instead of giving up, the ensemble construction scheme should put more emphasis on these points, in the hope of eventually pushing it over the majority vote threshold towards a correct classification. This means that we are effectively *increasing* bad diversity, in the hope that it eventually turns into good diversity.

In standard Random Forests, each next tree is constructed independently of the ensemble constructed so far. Instead, we may try to construct the next tree in a coordinated manner such that it more optimally complements the ensemble constructed so far. Both good and bad diversity can be expressed in terms of the average member error. Each member that is added to the ensemble contributes to it. Consequently, we might be able to steer the development of ensemble performance by encouraging the next member to either correctly or inclassify a point, given how the ensemble constructed so far performs on this point.

We now proceed to define some weighting functions informed by diversity (see **??**).

Definition 5.1.1 *(DRF weighting scheme [**bernard-drf**]) Let $\bar{q}$ be the ensemble constructed so far. For a pair $(X, Y) \in D$, define the Dynamic Random Forest weighting scheme as*

$$w_{DRF}(X) =_{def} \frac{1}{M} \sum_{i=1}^{M} \ell_{0/1}(Y, q_i(X))$$

This will have the effect that correctly classified examples are assigned lower weight and incorrectly classified examples are assigned higher weight.

This weighting scheme was first proposed in [**bernard-drf**]. However, they only give a heuristic, intuitive justification in that if a high number of trees misclassifies an example, the next tree should put more emphasis on it, similar to boosting strategies (**??**). We derive and motivate this weighting scheme from a perspective of diversity and give insight into how exactly it works – namely, that it does *not* flat-out increase the performance (as in boosting schemes) but instead encourages diversity.

XuChen re-iterate on the DRF weighting scheme and propose an alternative scheme.

$$w_{\text{XuChen}}(X) =_{\text{def}} \begin{cases} \varepsilon^2 & \varepsilon \leq \frac{1}{2} \\ \\ \sqrt{\varepsilon} & \varepsilon > \frac{1}{2} \end{cases} \qquad \text{for } \varepsilon =_{\text{def}} \frac{1}{M} \sum_{i=1}^{M} \ell_{0/1}\left(Y, q_i(X)\right)$$

Again, the authors provide only a heuristic motivation, which is that, compared to w_{DRF}, their method has a more drastic effect of up- and downweighting. We can now give a more informed interpretation. Inspecting w_{DRF}, which is continuous around the majority vote threshold, one can see that very similar weights are assigned to examples which are classified just barely correctly (resulting in a 0/1-loss of 0) and examples which are classified just barely incorrectly (resulting in a 0/1-loss of 1). This may mean suboptimal guidance in ensemble construction since both cases have very similar weights, but their effect on the ensemble loss is actually dramatically different. One disadvantage is that we take a heuristic step away from theory.

Now we just have to figure out how to actually influence the training of the next member such that it's performance on some points is (likely) increased or decreased. One way that is suited well for Random Forests is to assign *weights* $w : X \to [0, 1]$ to examples and consider them during member training. In section **??**, we will argue in detail how example weights influence the ensemble.

For random forests, these weights can possibly come into effect via two mechanisms:

- *Weighted bootstrapping*: Instead of drawing the bootstrap samples uniformly, draw a sample with probability according to its weight. If the bootstrap sample is large, examples with higher weight are more likely to be oversampled and thus appear multiple times in the bootstrap sample.
- *Weighted tree construction*: We have seen in **??** that tree construction according to some impurity measure greedily optimises a loss function. Likewise, weighting examples during computation of the impurity measure optimises a weighted loss.

5.1.2. Experimental Evaluation of the DRF weighting schemes

We compare different variants of the weighting scheme introduced in section 5.1.1. We look at the overall ensemble generalisation error, as well as the components of the error as given by the bias-variance decomposition **??** and the diversity decomposition **??**. Further, we analyse the ensemble margins (definition 2.2.3).

Experiment Setup

Compared learners In summary, we compare the following learning algorithms. For a learner, any configuration is similar to the one mentioned before unless otherwise

specified. Since for the weighted variants the construction of the next tree depends on the performance of the ensemble so far, trees are constructed in sequence.

standard-rf-classifier: Standard Random Forest implementation based on *sklearn*. The tree hyperparameters are such that trees are grown until each leaf contains one data point. The number of randomly sampled candidate dimensions to search for the best split is set to $\sqrt{d}$ where d is the total number of dimensions. The impurity measure is the Gini impurity as defined in section 3.1.2. Each tree is grown on a bootstrap sample determined by sampling n out of n data points uniformly, with replacement.

drf-weighted-bootstrap-classifier: Each tree is grown on a bootstrap sample determined by sampling n out of n points according to the DRF weighting scheme (see 5.1.1). To yield a valid probability distribution, the weights are normalised via $w'(x_i) \leftarrow \frac{w(x_i)}{\sum_{j=1}^{n} w(x_j)}$.

drf-weighted-fit-classifier: Each tree is grown on a uniform bootstrap sample. For tree construction, namely measuring impurity, each example is weighted according to w_{DRF} (again, normalised).

drf-weighted-fit-oob-classifier: The example weights for a point x_i are determined based only on *out-of-bag* trees for x_i. These are those trees whose bootstrap sample has not included x_i.

Compared datasets We give a brief motivation for the classification datasets we have selected for evaluation. A detailed summary of each dataset can be found in **??**. *cover* is a dataset with a relatively high number of examples and low feature dimensionality. *mnist-subset* is a dataset with a moderate number of examples and high dimensionality. *diabetes* is a dataset with relatively high error rates. *bioresponse* is a small dataset with a very high number of features ($d \approx \frac{1}{2}n$). *qsar-biodeg* is a small dataset used for quick testing. Further, [**bernard**, **xuChen**] evaluated on *mnist* (although not just a subset of it), *spambase-openml* , *digits* and *diabetes* .

Approximating statistical quantities For a dataset with n examples, $n_{\text{train}} =_{\text{def}} \frac{3}{4}n$ examples were assigned to be part of the *training split*, the other n_{test} for the *testing split*. Examples in the testing split are used for evaluation only and were never used in training a model. If X is a random variable taking values in the space of examples $\mathcal{X}$, expectations over X are approximated as the arithmetic mean over given examples in the testing split, i.e. for a function g:

$$\mathbb{E}_X [g(X)] \approx \sum_{i=1}^{n_{\text{test}}} g(x_i)$$

If D is a random variable corresponding to the input to a learner, for instance the training dataset or randomness in the learning algorithm, an expectation over D is approximated by an arithmetic mean over results of a fixed number of trials. In our case, we performed 3 trials.

Results

We describe and analyse the results here and provide several plots. The full results plotted for comparison across learners and datasets are given in section B.2.

Generalisation error and diversity

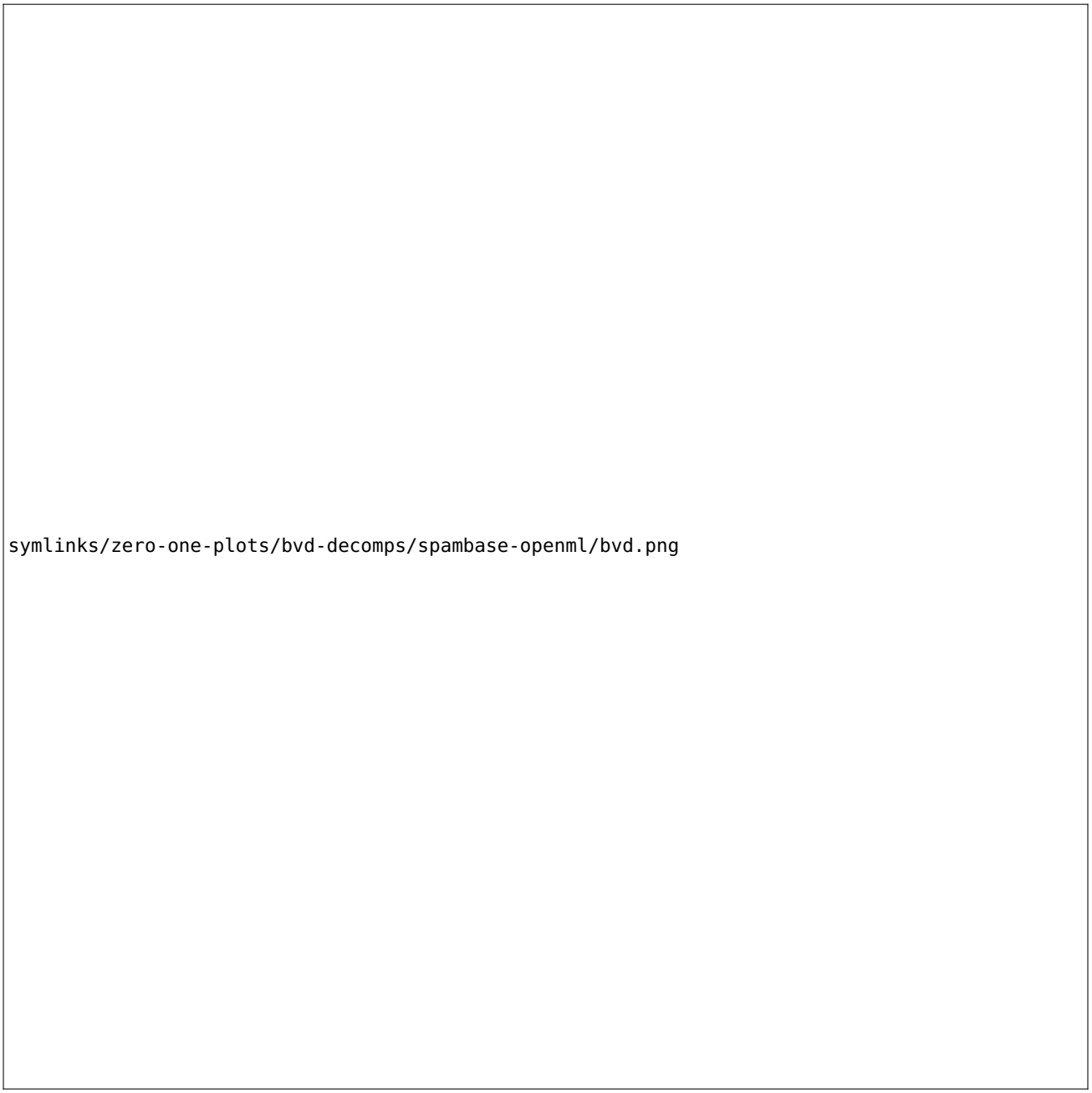

Figure 5.1.: Comparison of components of the ensemble generalisation error on the *spambase-openml* dataset. Visualised are ensemble generalisation error, average member bias, average member variance and diversity.

We can see that weighted bootstrapping and weighted tree construction (see **??**) behave quite differently. The case for the *spambase-openml* dataset is given in figure **??**.

It is striking that, for every dataset, weighted bootstrapping initially brings a sharp increase in average bias and, consequently, generalisation error. The average variance stays mostly constant after an initial slight increase. As the number of trees grows, the generalisation error and the average bias diminish. A sharp initial increase in diversity mitigates the increase in average bias. The average bias then continously decreases to a similar or slightly higher level as the other learners. The average member variance seems to be very similar to that of other learners.

Weighted bootstrapping seems to be able to achieve similar or, often, better generalisation error than any other learner. It also consistently produces higher diversity ensembles than the other learners. It is interesting to note that on *spambase-openml* , where the initial increase in average bias and diversity appear most pronounced, diversity actually *decreases* as more trees are added to the ensemble.

For weighted tree construction, average bias and average variance are mostly constant and the decrease in generalisation error with a growing number of trees is solely due to increasing diversity. This is the same behaviour we also observe and motivate theoretically for standard Random Forests. Weighted tree construction performans as good as or slightly worse than standard Random Forests in terms of generalisation error. This is somewhat surprising since the intuitive motivation was that weighted tree construction will influence the splitting criteria.

Weighted tree contruction with out-of-bag weights does not appear to bring any advantage.

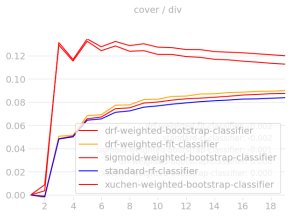

Figure 5.2.: Diversity-effect by number of trees for different learners on the *cover* dataset. Weighted bootstrapping amounts to a sharp increase in diversity for the first couple of trees. Weighted tree construction only causes a slight increase in diversity as compared to a standard Random Forest.

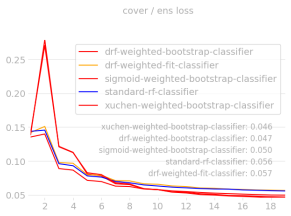

Figure 5.3.: Expected generalisation error for different learners on the *cover* dataset. Weighted tree construction and standard Random Forests behave almost identically. For weighted bootstrapping, an initial increase in error is followed by a consistently lower error rate.

Ensemble bias and variance The initial increase in average bias is also reflected in an initial increase in ensemble bias. At the same time, ensemble variance is decreased. Note how the average variance stays almost constant, while the ensemble variance varies greatly. This is the "outside view" on how diversity is a component of ensemble variance.

As the number of trees grows, for some datasets, ensemble variance is higher. For others, it is similar to standard Random Forests.

Ensemble margins

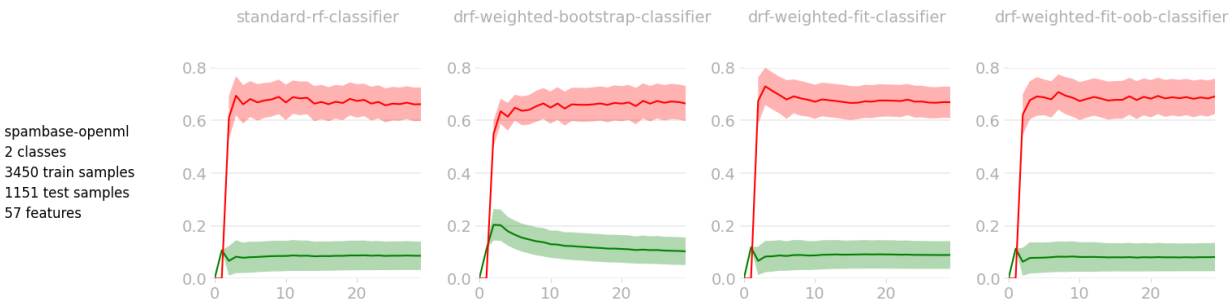

Figure 5.5.: Ratio of incorrect trees per number of trees in the ensemble, plotted separately for correctly (green) and incorrectly (red) classified examples.

The weighting schemes are essentially thought of to influence the ratio of incorrect members and thus bias, variance and diversity. Under the weighting schemes defined in **??**, on points for which the ensemble prediction is correct, the ratio of incorrect trees should *increase* (more diversity) and on points for which the ensemble prediction is incorrect, the ratio of incorrect trees should *decrease*. We can directly observe the ratio of incorrect members. We plot the average ratio of incorrect trees $\frac{1}{M}\sum_{i=1}^{M}\ell_{0/1}\left((,y),q_i\right)$ separately for correctly and incorrectly classified examples. For weighted bootstrapping, across all datasets, we can indeed observe the expected effect. This is another view on the sharply increasing diversity as seen in e.g. figure 5.1. Again, the effect diminishes as the number of trees grows.

Figure 5.4.: Ensemble generalisation error, ensemble bias and ensemble variance of weighted bootstrapping on the *digits* dataset.

Further, we can look at the distribution of ensemble margin per example. We plot a histogram of examples with respect to the number of trees incorrectly classifying that example. In binary classification, more than $\frac{1}{2}M$ trees being incorrect leads to an incorrect ensemble prediction. For instance for the *cover* dataset, we can observe that the distribution of ensemble margins indeed seems to be skewed slightly to the right (i.e. in direction of more disagreement) for positive examples (see figure 5.6)

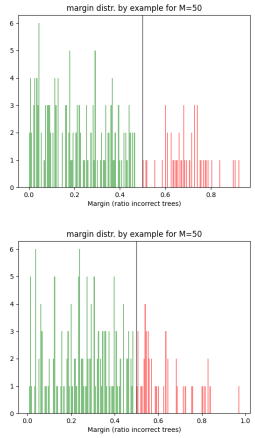

Figure 5.6.: Histograms of ensemble margins per example for the standard Random Forest (top) and weighted bootstrapping (bottom) learners grown each of $M = 50$ trees on the *diabetes* dataset. For weighted bootstrapping, the distribution of the margins appears slightly skewed to the center, reflecting that the weighting schemes encourages disagreement on points where many ensemble members are correct and agreement on points where many ensemble members are incorrect.

Comparing weighted bootstrapping and weighted tree construction It is worth discussing why weighted bootstrapping and weighted tree construction produce such different behaviour. In practise, any bootstrap sample is finite. The bootstrap sample is drawn with replacement, thus a bootstrap sample does not necessarily include all examples from the training dataset. Under uniform bootstrapping, each example has equal chance to be included in the bootstrap sample (see 3.2.1). It is then considered during tree construction according to its weight. On the other hand, under weighted bootstrapping, examples with high weight are more likely and examples with low weight are less likely to appear in the bootstrap sample. In particular, this means that examples with low weight are more likely to not be included at all in the bootstrap sample and consequently not be considered at all during tree construction. This might be an intuitive explanation why weighted bootstrapping shows a stronger effect than weighted tree construction in our experiments. A thorough discussion has to be left for future work.

Comparison to [bernard-drf] **bernard-drf** evaluated only the following approach: They would perform both weighted bootstrapping and weighted tree construction. Further, weights would be determined only on out-of-bag-trees (see section 5.1.2). Additionally, unrelated to the question at hand, they also employed a different way to determine candidate split features. Unlike in standard random forests, where a fixed number of candidate split features is sampled from all available features, the number of sampled candidate features was left fully random here. It was left unanswered which of these components actually affect the ensemble to what extent. Further, they did not provide any explanation or empirical analysis in terms of diversity. Looking at our results, the following points seem likely:

- An improvement in generalisation error is still obtained with standard candidate split feature sampling.
- The improvement can be explained using the notions of diversity.
- Weighted tree construction alone appears to have only very little effect as compared to a standard Random Forest.
- Weighted bootstrapping seems to provide the main effect.
- Determination of weights using out-of-bag-trees only does not improve performance for weighted tree construction.

Ambiguity-effect decomp plots ...

Scatter Plots ...

5.1.3. "Boosting" rationale for example weighting in classifier ensembles

re. spike: not super surprising, if there are only 1, 2, 3 trees, weights as ratio of incorrect trees are super extreme – motivation to dampen

would be good to put here to explain "spike". Can such a spike be observed in boosting methods?

initial increase in bias also observed for classical boosting models? (wood23 p26)

5.1.4. Outlook: Relationship to Competence

...

5.1.5. Outlook: Generalising to other losses

We will consider transferring this approach to Bregman divergences.

The ambiguity-effect decomposition holds for both the 0/1-loss and Bregman divergences. The key difference is that Bregmand divergences are non-negative and thus there are no points with "bad" diversity which contribute negatively to the ambiguity-effect term. In other words, any kind of ambiguity is beneficial under Bregman divergences.

$$B_\phi(y, \bar{q}) = \frac{1}{M}\sum_{i=1}^{M} B_\phi(y, q_i) - B_\phi(\bar{q}, q_i)$$

However, as can be seen from the ambiguity-decomposition, there is a tradeoff between average member error and ambiguity. Since $\frac{1}{M}\sum_{i=1}^{M} B_\phi(y, q_i) - B_\phi(\bar{q}, q_i) = B_\phi(y, \bar{q}) \geq 0$, ambiguity is upper-bounded by the average member error. This means that any improvement due to diversity can only happen by reducing the amount of error introduced by the individual member error. If there is little average member error to begin with, encouraging diversity will not necessarily improve ensemble performance. However, this does not mean that ensemble performance has to decrease: Empirical experiments [**buschj-etc**] have confimed that, as diversity is increased, there exist regions where the overall ensemble error stays constant – that is, increasing diversity leads to increased member error but the two perfectly outweigh each other. In other words, one can find a wide range of ensemble models with equal performance but varying diversity.

However, we may still consider the above equation and how choosing the next member q_{M+1} affects it. Suppose the average member loss $\frac{1}{M}\sum_{i=1}^{M} B_\phi(y, q_i)$ is large. Then this is either mitigated by the diversity $\frac{1}{M}\sum_{i=1}^{M} B_\phi(\bar{q}, q_i)$, in which case the ensemble error is still low; or it is not, in which case the ensemble error too is large. However, if the average member loss $\frac{1}{M}\sum_{i=1}^{M} B_\phi(y, q_i)$ itself is small, there cannot be much diversity since, it is bounded by the average member error. Because of this, we need to consider the difference relative to the average member loss (which, interestingly, is the *ensemble improvement rate* of [**theisen**]).

$$\frac{\frac{1}{M}\sum_{i=1}^{M} B_\phi(y, q_i) - B_\phi(\bar{q}, q_i)}{\frac{1}{M}\sum_{i=1}^{M} B_\phi(y, q_i)} \in [0, 1]$$

If this quantity is low, then there is high member error but also high diversity.

If it is high, then the diversity relative to the average member error is low

5.1.6. Outlook: Alternatives to example weights

In section 5.1, we use example weights to influence the construction of the next tree of the ensemble. However, example weights are just one way to do so. Here, we explore some alternatives.

Leaf refinement The leaf outcomes of a tree can be re-adjusted after it has been grown [**others**]. This can be done for example using gradient descent [**buschj-negative-correlation-forests**]. However, this requires an extra stage of optimisation. Further, only the leaf outputs are adjusted, but the tree structure is left untouched. Our approach instead works exactly through influencing the tree structure through weighted examples.

Quantifying diversity and average member error in terms of cells As we have seen in section **??**, the generalisation error can be expressed as a weighted sum over forest cells. We can also express the ambiguity decomposition (see **??**) in terms of forest cells. For sake of clarity we omit the expectation over D and write $Z =_{\text{def}} (X, Y)$. Let $Z = Z_1 \dot\cup \ldots \dot\cup Z_P$ be a forest partition of $Z = (X, Y)$.

$$\begin{aligned}
\mathbb{E}_Z\left[\ell(y, \bar{q})\right] &= \sum_{p=1}^{P} \mathbb{P}\left[Z_p\right] \underbrace{\frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_Z\left[\ell(y, q_i)\right]}_{\text{err}(Z_p)} - \underbrace{\frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_Z\left[\text{LE}(\bar{q}, q_i)\right]}_{\text{div}(Z_p)} \\
&= \sum_{p=1}^{P} \mathbb{P}\left[Z_p\right] \frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_Z\left[\ell(y, q_i) - \text{LE}(\bar{q}, q_i) \mid Z_p\right]
\end{aligned}$$

$\mathbb{P}\left[Z_p\right]$ corresponds to the area of the forest cell p and can be determined based solely on the decision boundaries of the corresponding tree cells. $\text{div}(Z_p)$ depends solely on the outputs of the tree cells that constitute Z_p. It is simply the generalised variance (see **??**) of these leaf outputs. If leaf outputs are saved with the tree model after construction, these can be directly read off the model. However, these values still depend directly on the training data. Instead, we would like to infer the value based on the decision boundaries alone. In Random Forests, trees are grown *deeply*, that is, until each leaf contains only a single point. Consequently, if the number of data points n is large, leaf cells will become small. Provided that the regression function $m(x) = \mathbb{E}[Y \mid X = x]$ is uniformly continous, small leaf cells imply that the variation of m throughout a leaf is bounded. In fact [**scornet**] show that, as n grows, the variation in a leaf becomes arbitrarily small. This motivates that, instead of the data-dependent leaf output $q_i(x)$, one could instead use the center of the cell $\tilde{q}_i(x)$, depending only on the decision boundaries. If the variation inside a cell vanishes, then also the error of using $\tilde{q}_i(x)$ over $q_i(x)$ vanishes.

Similarly, we can express the average member error in terms of tree cells. Due to the appearance of y, this is still directly dependent on realisations of Y.

$$\mathbb{E}_Z\left[\frac{1}{M}\sum_{i=1}^{M}\mathrm{LE}(y,q_i)\right] = \sum_{p=1}^{P}\mathbb{P}\left[Z_p\right]\underbrace{\frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_Z\left[\ell(y,q_i)\mid Z_p\right]}_{=_{\mathrm{def}}\mathrm{err}(Z_p)}$$

A diversity-aware splitting criterion Imagine growing M trees simultaneously as follows: In each iteration, one split is determined for each tree. Which node is being split is left to the choice of the splitting criterion. One could then attempt to derive a splitting criterion that optimises not only for a split that is pure (improves the prediction performance of the tree), but also one that implies predictions different to those of other trees (improves diversity). Each split of a tree node yields a new forest partition $Z_1, \ldots, Z_P$.

We have already seen that splitting criteria in standard Random Forests optimise a loss function. Hence, the objective function for a general splitting criterion for the i-th member could be written as $\ell(y, q_i)$, which is simply the member error. However, informed by the above discussion, one could define the splitting criterion in coordination to other ensemble members, i.e.

$$\ell(y, q_i) - \lambda\ell(\bar{q}, q_i)$$

Note that this is directly analog to the NCL objective. In the NCL objective, however, this criterion is evaluated per point. In our case, it is evaluated per split, and (under some assumptions), only depends on previous splits in the forest and no extra sampling to determine the values.

Continuing from here, probably many other simplificiations could be made on how to actually faster / more simply determine these values...

5.2. Generalized Negative Correlation Learning

Consider the bias-variance-covariance decomposition for the squared error loss given in theorem 4.1.1. The covariance term contributes positively to the ensemble error if the outputs of the members are positively correlated. This suggests the idea that it might be beneficial to train ensemble members in coordination such that their outputs are uncorrelated.

[**LiuYao**] provide an implementation of this idea for neural network ensembles called *Negative Correlation Learning* (NCL). Neural network training involves a *forward* and a *backward* pass. In the forward pass, the neural network in its current state is queried with training data. Its prediction is evaluated according to a loss function. Then, the model parameters are updated according to the gradient of the loss function with respect to the parameters. A common practise is to add a *regularisation term* to this loss function in order to bias the model towards e.g. sparse predictions [**todo**]. In Negative Correlation Learning, forward and backward pass are performed synchronously for all members. This enables using loss functions that depend not only on an individual member but the entire ensemble.

The i-th member network is trained using a loss function ℓ_i that contains a regularisation term p_i, which is intended to influence the training of the i-th member such that its predictions are uncorrelated with the other members. The hyperparameter $\lambda \in \mathbb{R}$

determines how much weight is put on either of the two components. The training objective for the i-th network can be written as

$$e_i =_{\text{def}} (y - q_i)^2 \quad + \quad \lambda \, p_i$$

where the first term is the individual error of the i-th member and p_i is the regularisation term. [**LiuYao**] defined p_i as

$$p_i =_{\text{def}} \left((\bar{q} - q_i) \sum_{i \neq j}^{M} (\bar{q} - q_i)) \right)$$

This is reminiscent in shape of the contribution of the i-th member to the covariance term in the bias-variance-covariance decomposition — however, the expectation over D is replaced with the ensemble combiner $\bar{q}$! Seeing that here the ensemble combiner is the arithmetic mean of member outputs, this means that instead of a variance under variations in D, this considers the variance between member model outputs.

Note that the penalty term corresponds to the contribution of the i-th member to the ambiguity (**??**). [1]

$$p_i(y, x) = -(\bar{q} - q_i)^2$$

Indeed, as [**brown2005**] noted, the NCL approach can be motivated theoretically via the ambiguity decomposition. The error of the full ensemble can be written as [2]

$$e_{\text{ens}} =_{\text{def}} \frac{1}{2}(\bar{q} - y)^2 = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2}(y - q_i)^2 \quad - \quad \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2}(\bar{q} - q_i)^2$$

In summary, a diversity-encouraging loss function for an individual neural network ensemble member can be defined as follows.

Definition 5.2.1 *(NCL neural network objective for squared error [**brown2005**]) The loss function of the i-th neural network ensemble member is defined as*

$$e_i(y, x) =_{def} \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2}(y - q_i)^2 \quad - \lambda \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2}(\bar{q} - q_i)^2$$

The penalty coeffient λ smoothly interpolates between training q_i to either maximise its individual performance or the ensemble performance, since

$$\frac{\partial e_i}{\partial q_i} = \frac{1}{M} ((q_i - y) - \lambda(q_i - \bar{q}))$$

and

$$\lambda = 0 \quad \rightarrow \quad \frac{\partial e_i}{\partial q_i} = \frac{1}{M}(q_i - y) = \frac{1}{M} \frac{\partial e_i}{\partial q_i}$$

$$\lambda = 1 \quad \rightarrow \quad \frac{\partial e_i}{\partial q_i} = \frac{1}{M}(\bar{q} - y) = \frac{\partial e_{\text{ens}}}{\partial q_i}$$

Several authors have attempted to generalise Negative Correlation Learning or transfer it to other loss functions. For instance, [**webb21**] directly prove an ambiguity decomposition for the KL-divergence K and propose the objective

$$e_i(x, y) =_{\text{def}} \frac{1}{M} \sum_{i=1}^{M} K(y \mid\mid q_i) \quad - \quad \lambda \, \frac{1}{M} \sum_{i=1}^{M} K(\bar{q} \mid\mid q_i)$$

1: $\bar{q}$ is the arithmetic mean:

$$\bar{q} = \frac{1}{M} \sum_{i=1}^{M} q_i$$

The sum of deviations around the mean is zero:

$$\sum_{i=1}^{M} (\bar{q} - q_i) = 0$$

Omitting one member from the sum implies

$$\sum_{i \neq j} (\bar{q} - q_i) = -(\bar{q} - q_j)$$

2: We additionally use a factor of $\frac{1}{2}$. This makes it easier to express gradients of this function and is equivalent for optimisation.

where $\bar{q}$ is the geometric mean combiner (see **??**). The proof is not trivial and does not give insight into whether such a decomposition might also exist for other loss functions.

[**buschjaeger**] approached the problem by attempting to derive a generalised decomposition of the ensemble error that incorporates diversity. The basic idea is to consider a Taylor approximation around the ensemble combiner, which is assumed to be the arithmetic mean combiner. $\bar{q} =_{\text{def}} \mathbb{E}_\Theta [q_\Theta] \approx \frac{1}{M} \sum_{i=1}^{M} q_i$.

$$\begin{aligned}\mathbb{E}[\ell(y,q)] = \mathbb{E}[\ell(\bar{q})] &+ \mathbb{E}\left[(q-\bar{q})^\top \nabla_{q^\star}(\ell(\bar{q}))\right] \\ &+ \mathbb{E}_\Theta\left[\frac{1}{2}(q-\bar{q})^\top \nabla^2_{\bar{q}}(\ell(q^\star))(q-\bar{q})\right] \\ &+ \mathbb{E}[R_3]\end{aligned}$$

R_3 is the remainder of the Taylor approximation, which vanishes if the third derivative of ℓ is zero. By definition of $\bar{q}$, $\mathbb{E}_\Theta\left[\nabla_{\bar{q}}(\ell(\bar{q}))\right] = \nabla_{\bar{q}}(\ell(\bar{q}))$ and $\mathbb{E}_\Theta[q-\bar{q}] = 0$ and thus the second term vanishes. The expectations can be approximated as follows:

$$\bar{q} = \mathbb{E}_\Theta[q_\Theta] \approx \frac{1}{M}\sum_{i=1}^{M} q_i$$

$$\mathbb{E}_\Theta\left[\frac{1}{2}(q-\bar{q})^\top \nabla^2_{\bar{q}}(\ell(\bar{q}))(q-\bar{q})\right] \approx \frac{1}{2}\frac{1}{M}\sum_{i=1}^{M} d_i^\top D d_i$$
$$\text{for } D =_{\text{def}} \nabla^2_{\bar{q}}(\ell(\bar{q}), y) \text{ and } d_i =_{\text{def}} (\bar{q} - q_i)$$

$$\mathbb{E}_\Theta[R_3(x)] \approx \tilde{R}$$

Note that here they *assume* the expected model $\bar{q}$ to also be the ensemble combiner, i.e. $\bar{q} \approx \frac{1}{M}\sum_{i=1}^{M} q_i$. Based on this, they propose the following generalised training objective.

Definition 5.2.2 *(Generalised NCL objective as proposed by [**buschj**])*

$$e_i =_{def} \frac{1}{M}\sum_{i=1}^{M} \ell(y,q_i) - \frac{1}{2}\frac{1}{M}\sum_{i=1}^{M} d_i^\top D d_i$$

where $D =_{def} \nabla^2_{\bar{q}}(\ell(y,\bar{q})$ *and* $d_i =_{def} (q_i - \bar{q})$

This is based on the assumption that the remainder to the Taylor approximation R_3 is negligibly small. Further, it does not apply to all loss functions. Let us consider some examples of commonly used loss functions.

- For the squared error, the third derivative vanishes and thus the decomposition is exact.
- For the negative log-likelihood, the third derivate is not bounded and thus this decomposition can not be used.
- For the cross-entropy loss, the third derivative is bounded and while the decomposition is not exact, it can be used for approximation.
- For any other loss function, this would have to be checked.

Squared error loss:

$$\ell(y,x) =_{\text{def}} (y-x)^2$$

Negative log-likelihood:

$$\ell(z,y) =_{\text{def}} -\sum_{i}^{k} y_i \log(z_i)$$

Cross-entropy loss:

$$\ell(z,y) =_{\text{def}} -\sum_{i}^{k} y_i \log \frac{e^{z_i}}{\sum_j e^{z_j}}$$

It is evident however, that we already have a fully general ambiguity decomposition at hand, namely the ambiguity-effect decomposition (**??**). This decomposition is exact

and holds for *any* loss function (including the 0/1-loss). For Bregman divergences, this reduces to the ambiguity decomposition **??**. We claim that the adequate generalisation of the NCL objective follows this structure.

Definition 5.2.3 *We propose the following generalisation of the NCL neural network objective. For general loss functions ℓ:*

$$e_i =_{def} \frac{1}{M}\sum_{i=1}^{M} L(y, q_i) \; - \lambda \left(\frac{1}{M}\sum_{i=1}^{M} L(y, q_i) - L(y, \bar{q}) \right)$$

And for Bregman divergences:

$$e_i =_{def} \frac{1}{M}\sum_{i=1}^{M} B_\phi\,(y, q_i) \; - \lambda \left(\frac{1}{M}\sum_{i=1}^{M} B_\phi\,(\bar{q}, q_i) \right)$$

NCL for the squared error loss as introduced by [**LiuYao**] and [**brown2005**], as well as for the KL-divergence as given by [**webb**] are special cases of this.

This provides a general framework for Negative Correlation Learning with arbitrary loss functions. Because it is founded on the exact and intuitive bias-variance-diversity decomposition, this also yields a natural and intuitive means for understanding and analysing Negative Correlation Learning and its effects.

Note that the Dynamic Random Forest approach **??** implicitly optimises a very similar objective. The construction procedure of individual decision trees greedily optimises its own performance (**??**), which corresponds to the first term in the NCL neural network objective. The introduction of example weights based on the ensemble diversity influences the decision tree construction to improve the ambiguity, which corresponds to the regularisation term. Note also that Negative Correlation learning already is "dynamic" in the sense that the ambiguity term is evaluated per point.

Although the term *Negative Correlation Learning* in the literature refers specifically to neural networks, we can now see that it is rather a style of training ensemble members with respect to the ambiguity decomposition. To the best of our knowledge, only one other algorithm has been published that realises this: [**negative-correlation-forests**] proposes to refine the leaf predictions in a given Random Forest using using gradient descent according to the objective defined in definition 5.2.2.

5.2.1. Experiments

As a proof of concept, we perform Negative Correlation Learning with small neural networks based on the cross-entropy loss (refthm:cross-entropy-decomp).

The case of KL-divergence was already investigated empirically in detail in [**Webb**].

Experiment setup, NN architecture? (also see margintable command/environment)

5.3. Outlook: Theoretical analysis

Individual learners no longer independent! But: can consider this from the perspective of boosting.

6. Conclusion

...

Appendix

A. Additional notes

The cross-entropy loss is a special case of the KL-divergence.

Lemma A.0.1 *([**wood23**], theorem 5) Theorem* **5** *Let* $\mathbf{y}$ *be a one-hot class vector of length* k*, and* $\mathbf{q} \in \mathbb{R}^k$ *be a model's prediction of the class distribution. Define a set of such models* $\{\mathbf{q}_i\}_{i=1}^M$*, and their combination* $\overline{\mathbf{q}}$ *as their normalised geometric mean. The following decomposition holds.*

$$\underbrace{-\mathbb{E}_D[\mathbf{y} \cdot \ln \overline{\mathbf{q}}]}_{\substack{\textit{expected} \\ \textit{cross-entropy}}} = \underbrace{-\frac{1}{M}\sum_{i=1}^{M} \mathbf{y} \cdot \ln \mathbf{q}_i^*}_{\textit{average bias}} + \underbrace{\frac{1}{M}\sum_{i=1}^{M} \mathbb{E}_D\left[K\left(\mathbf{q}_i^* \| \mathbf{q}_i\right)\right]}_{\textit{average variance}} - \underbrace{\mathbb{E}_D\left[\frac{1}{M}\sum_{i=1}^{M} K\left(\overline{\mathbf{q}} \| \mathbf{q}_i\right)\right]}_{\textit{diversity}},$$

B. Experiments

B.1. Implementation

what technologies we used, what other projects we built upon, where one can find the code, ...

B.2. Binary classification and Dynamic Random Forests

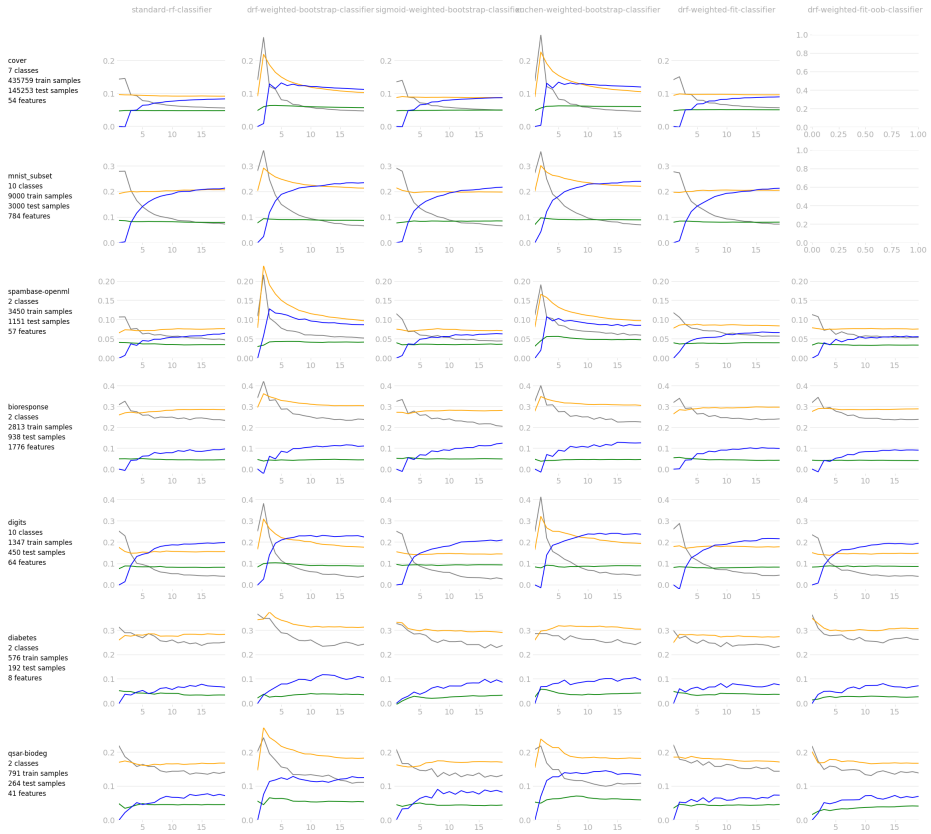

Figure B.1.: Full empirical results for 0/1-classification. Rows correspond to different datasets. Columns correspond to different learner variants. The plots show the components of the Bias-Variance-Diversity decomposition of the ensemble error **??**.

Figure B.2.: Full empirical results for 0/1-classification. Rows correspond to different datasets. Columns correspond to different learner variants. The plots show the components of the decomposition of the ensemble error in ensemble bias and ensemble variance **??**.

B.3. Boosted Random Forests

just drop derivation and results in here...

advantages of boosting but without the risk of overfitting?

C. Outlook

Fuctional Bregman divergences ...

... relationship pairwise comparisons and comparison to centroid – covariance approach vs mean dists approach

properties of metric losses

generalised k-means as splitting function – i'm sure someone tried this with standard k-means already. would then also need to relate to spaeh.

...

variance-effect as impurity criterion? what would this reduce to? in general, that as template, ref bregman info

theisen: For example, Figure 2 suggests that the relationship between EIR and DER can be even more finely characterized. Is it possible to refine our analysis further to incorporate information about the data and/or model architecture? Second, can we formalize the connection between ensemble effectiveness and the interpolation point, and relate it to similar ideas in the literature?