Radboud University Nijmegen

Faculty of Science

Active Learning

A pessimistic approach to beat random sampling in expectation

Bachelor of Science

Author:
Yves van Haaren
s1095186

Supervisor:
Marco Loog

Second reader:
Johannes Textor

June 2025

Abstract

Active learning is a machine learning area that is concerned with reducing labelling effort by strategically choosing which unlabelled data to query for labels during training. We study a pessimistic formulation of active learning, where the goal is to minimize the worst-case risk over all possible labellings of the unlabelled data. This leads to an optimisation problem over the probability simplex, where each point defines a sampling distribution over unlabelled instances. The core idea is to in some sense outperform uniform sampling since we still don't know how to outperform random sampling; it is often used as a baseline because sampling uniformly from the unlabelled data would represent a sampling procedure from the underlying distribution from which the unlabelled instances were sampled. For the case of two unlabelled instances, we derive a closed-form solution. Additionally, theoretical insights that enable exact, but computationally expensive optimisation are developed for the general case when dealing with arbitrary amounts of unlabelled data points. To address scalability, we propose greedy approximation methods with significantly lower complexity. Empirical results show that while one heuristic underperforms, another offers a good trade-off between efficiency and accuracy. Finally, we conclude with potential directions for improving robustness and extending this framework to larger datasets.

Preface

The research conducted in this thesis came to life with a lot of inspiration by my supervisor Marco Loog; he came up with the pessimistic approach a couple years back. Now, I took over the baton with the intention of extending the knowledge about active learning within this pessimistic approach. During my Bachelor's studies in Mathematics and Computing Science, I came to develop an interest in data science and especially machine learning.

This thesis is intended to be read by people who have completed a Bachelor's degree in Mathematics, or even Computing Science, assuming that they have enough mathematical maturity in proofs and mathematical knowledge in general. Moreover, a working knowledge of machine learning theory is assumed.

A special thanks to my supervisor Marco Loog who has spent hours with me on whiteboard sessions, providing me with feedback and laying the groundwork with his idea of the pessimistic approach. Furthermore, I want to thank my parents and my girlfriend who provided me with plenty of snacks and drinks during the never-ending days of thinking and writing.

Contents

1 Introduction

In many machine learning applications, large amounts of data are available, but only a small portion of that data is labelled. Since labelling typically requires human effort, it can be expensive, time-consuming, or simply infeasible at scale. *Active learning* addresses this issue by selecting which unlabelled data points to annotate, with the goal of reducing the total labelling cost while still achieving high model performance.

The central idea behind active learning is that some data points are more useful to label than others. Numerous strategies have been proposed to formalize this idea, such as uncertainty sampling [4], where the algorithm queries the most uncertain predictions, and margin-based methods [6], which focus on samples closest to the decision boundary. Despite their intuitive appeal, these techniques often rely on heuristics or assumptions about the data distribution, and in practice, they do not always outperform a much simpler strategy: *random sampling*.

Random sampling, where labels are requested for unlabelled points chosen uniformly at random, is widely used as a baseline in active learning. It is well justified theoretically because it mirrors the assumption that data is drawn independently and is identically distributed (i.i.d.), which underlies most supervised learning methods. Surprisingly, recent benchmarks have shown that many active learning methods fail to consistently improve over random sampling [8], especially in the absence of strong assumptions about the underlying data.

This thesis investigates whether it is possible to design an active learning strategy that is *guaranteed* to perform better than random sampling—regardless of how the unlabelled data is actually labelled. To explore this, we propose a *pessimistic* formulation of active learning, in which the goal is to minimize the worst-case risk over all possible labellings of the unlabelled data. We define an objective function that compares a given sampling strategy to random sampling, and cast the problem as a minimax optimisation over the probability simplex.

We develop theoretical insights into the structure of this optimisation problem, including its convexity and the behaviour of optimal strategies. For the smallest case of only two unlabelled data points, a closed-form solution is found. For small-scale settings, we solve the problem exactly using the projected subgradient method. However, due to the high computational cost of this approach, we also introduce two greedy algorithms that approximate the optimal strategy with significantly lower time complexity.

Finally, we present empirical results showing that one of these greedy algorithms consistently outperforms random sampling in worst-case scenarios. This offers the first evidence that, even under pessimistic assumptions, it is possible to construct active learning strategies that guarantee improvements over uniform sampling. We conclude by discussing potential extensions and a relaxation of the framework that may lead to scalable algorithms suitable for real-world applications.

2 Background Theory

This thesis assumes a working knowledge of several areas of mathematics to follow the theoretical and algorithmic developments. Readers should be familiar with real analysis (e.g., continuity, convexity, compactness), linear algebra (including projections and normed spaces), and multivariate calculus (e.g., gradients and differentiability). A basic understanding of numerical methods is also presumed, especially first-order optimisation techniques such as gradient descent and subgradient methods. Hypothesis testing and other basic techniques from statistics are also assumed.

Further, the formulation and analysis of active learning strategies rely on concepts from game theory—particularly the use of mixed strategies and minimax optimisation—as well as operations research, including linear and convex programming. Readers should also be comfortable with the geometry and interpretation of the probability simplex, which forms the domain of the optimisation problem in this work.

Familiarity with empirical risk minimization, surrogate loss functions (e.g., logistic loss), and risk-based evaluation in machine learning is essential, as is an understanding of how pessimistic reasoning models worst-case labellings in active learning. Moreover, basic knowledge about theoretical machine learning and complexity theory are assumed. Finally, the use of greedy algorithms as efficient approximations to intractable exact methods assumes knowledge of algorithmic design trade-offs.

3 Theory

To start with, we are going to set up an approach to active learning that, by construction, finds a strategy that guarantees performance improvements over random sampling whenever it exists. We do this for the setting in which we wish to pick a single additional labelled sample for our training set. Subsequently, we introduce two relaxations of the objective function that we derive to make our approach computationally feasible.[1] The first is a standard approach from game theory where one moves from considering pure strategies to mixed ones. The second offers a heuristic that simplifies computations by ignoring certain dependencies between probabilities. The final part of this section discusses extensions to picking multiple points and relaxing the deterministic labelling assumption. We start with some preliminaries.

3.1 Preliminaries

Training instances are independently identically distributed draws from an input space X, which is typically a subset of $\mathbb{R}^d$ for some positive integer d. We consider C-class problems, where the labels come from a set Y of cardinality C. For simplicity, we consider $Y = \{1, \ldots, C\}$. Furthermore, we denote the (surrogate) loss of interest by ℓ and let H be the hypothesis space over which we optimise.

Let us assume we have M labelled training samples $(x_i, y_i) \in X \times Y, i \in \{1, \ldots, M\}$, and furthermore N unlabelled samples $u_j \in X, j \in \{1, \ldots, N\}$. In addition. let $v_j \in Y$ with $j \in \{1, \ldots, N\}$ be the (hypothetical) true labels of the unlabelled set. The full N-vector of labels is denoted by v. Likewise, x, y, and u refer to the collections $(x_i)_{i=1}^M$, $(y_i)_{i=1}^M$, and $(u_j)_{j=1}^N$, respectively.

Note that we use different letters for the observed labelled and unlabelled instances for easy identification. It is important to realize, however, that we assume them all to be independent draws from one and the same distribution. As such, these samples represent this distribution of interest and, therefore, empirical risk minimization over the complete sample, labelled and unlabelled, is justified.
All in all, we aim to construct a well-performing hypothesis $h \in H$ in terms of the empirical risk $R(h)$ over all of the $M + N$ samples, both labelled and unlabelled

$$R(h) = \frac{1}{M+N} \left[\sum_{i=1}^{M} \ell(x_i, y_i | h) + \sum_{j=1}^{N} \ell(u_j, v_j | h) \right] \tag{1}$$

3.2 Active Sampling Strategy and Sampling Risk

Important for our approach is that the active learner can act stochastically. That is, it can decide to assign a probability s_k to every u_k, which indicates the probability of picking this point as the next one to label. Clearly, setting $s_k = \frac{1}{N}$ means we rely on standard random sampling. Note, moreover, that this strategy makes the active learner at least as powerful as deterministic active learning schemes. The latter are just limiting cases of stochastic sampling with one $s_k = 1$ and the remaining s_k set to zero. The primary question to answer now becomes how to set the probabilities $s = (s_k)_{k=1}^N \in \Delta_{N-1}$, where Δ_{N-1} denotes the standard $(N-1)$-simplex.

[1] Under the assumption that training the necessary classifiers are not the bottleneck.

For a given labelling $v \in Y^N$ of u, we can define what we refer to as the sampling risk S. It expresses the expected risk given the sampling probabilities s for the N unlabelled data points:

$$S(s|v) = \sum_{k=1}^{N} s_k R(h_k^{v_k}), \tag{2}$$

where $h_k^{v_k}$ is the hypothesis obtained when training on an additional input u_k that has label v_k. As our classifiers are obtained by means of ERM, we have

$$h_k^{v_k} := \arg\min_{h \in H} \frac{1}{M+1} \left[\sum_{i=1}^{M} \ell(x_i, y_i|h) + \ell(u_k, v_k|h) \right]. \tag{3}$$

We are interested in active learners that perform at least as good as random sampling. For the latter the sampling risk equals $S(\frac{1}{N}\mathbf{1}|v)$, where $\mathbf{1}$ is the all-one vector. The quantity of interest is the difference

$$S(s|v) - S(\frac{1}{N}\mathbf{1}|v) = \sum_{k=1}^{N} \left(s_k - \frac{1}{N} \right) R(h_k^{v_k}), \tag{4}$$

For the active sampling strategy to be at least as good as random sampling, this difference should be smaller than or equal to zero.

To get to our core objective in Equation (4), we assumed to have the labels v_j of the unlabelled data. The point, of course, is that we don't. As a consequence, if we want to be certain that the sampling risk is at least as good as and preferably lower than that of standard random sampling, we need to focus on the worst-case labelling. This leads us to consider the following minimax problem

$$\min_{s \in \Delta_{N-1}} \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v). \tag{5}$$

Lemma 3.1.

$$\min_{s \in \Delta_{N-1}} \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v) \leq 0$$

Proof. Note that $\frac{1}{N}\mathbf{1} \in \Delta_{N-1}$, and therefore,

$$\min_{s \in \Delta_{N-1}} \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v) \leq \max_{v \in Y^N} S(\frac{1}{N}\mathbf{1}|v) - S(\frac{1}{N}\mathbf{1}|v) = \max_{v \in Y^N} 0 = 0.$$

□

In essence, the previous lemma states that there exists a mixed strategy that is at least as good as random sampling with respect to the sampling risk, which is trivial since random sampling is a mixed strategy. However, of course, we are interested in the question whether or not

$$\min_{s \in \Delta_{N-1}} \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v) < 0,$$

since this would imply that there is a mixed strategy that is strictly better[2] than random sampling.

[2]Better meaning having a lower sampling risk.

3.3 The impossibilities of active learning

As it turns out by the following remark, we cannot guarantee outperforming random sampling in the general setting. Moreover, in certain settings there are some 'easy' characterisations which allows us to quickly identify if active learning (with this pessimistic approach) would provide any improvements over random sampling. Moreover, we will look at the ways we can beat random sampling in the setting where we already know that we can beat random sampling with a pure strategy. This section is meant as a gentle introduction to what is, and is not possible within the scope of active learning using the pessimistic approach. Moreover, this section should be seen as a rather independent section compared to other sections in this chapter. We start with the first claim.

Remark: In general, we cannot guarantee that

$$\min_{s\in\Delta_{N-1}} \max_{v\in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v) < 0.$$

Proof. To see this, consider the following example. We take $x = (-1, 1)$, $y = (-1, 1)$, $u = (-2, 2)$, $Y = \{0, 1\}$, and for the loss we use the negative log loss commonly used for logistic regression. Lastly, we use the hypothesis class of logistic regressors

With this setup, we see that the value of the minimax problem is equal to 0, with $s = (0 \quad 1)$, which means that we cannot outperform random sampling in this case.

Therefore, we cannot guarantee

$$\min_{s\in\Delta_{N-1}} \max_{v\in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v) < 0.$$

without additional assumptions. □

By the previous theorem, we know that we cannot always outperform random sampling. This makes one wonder, when can we use active learning to outperform random sampling? Are there cases where active learning even marginally outperforms random sampling, or should we give up on this pessimistic approach to active learning? The following theorem motivates us to conjecture the latter in the case of having two unlabelled data points.

Theorem 3.2. *There exists a $\tilde{v} \in Y^2$ such that $R(h_1^{\tilde{v}_1}) = R(h_2^{\tilde{v}_2})$ if and only if*

$$\min_{s\in\Delta_1} \max_{v\in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) = 0.$$

Proof. $\Rightarrow$: Suppose that there exists a $\tilde{v} \in Y^2$ such that $R(h_1^{\tilde{v}_1}) = R(h_2^{\tilde{v}_2})$ and assume that

$$\min_{s\in\Delta_1} \max_{v\in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) < 0.$$

Then there exists a $\tilde{s} \in \Delta_1$ such that

$$\max_{v\in Y^2} S(\tilde{s}|v) - S(\frac{1}{2}\mathbf{1}|v) < 0.$$

And by rewriting we see

$$S(\tilde{s}|v) - S(\frac{1}{2}\mathbf{1}|v) = (\tilde{s}_1 - \frac{1}{2})(R(h_1^{v_1}) - R(h_2^{v_2})),$$

where we use the notation $\tilde{s} = (\tilde{s}_1 \quad \tilde{s}_2)^T$ and the fact that $\tilde{s}_1 + \tilde{s}_2 = 1$. Now, by our initial assumption, we get that

$$\begin{aligned} 0 &= (\tilde{s}_1 - \frac{1}{2})(R(h_1^{\tilde{v}_1}) - R(h_2^{\tilde{v}_2})) \\ &< \max_{v \in Y^2} S(\tilde{s}|v) - S(\frac{1}{2}\mathbf{1}|v) \\ &< 0, \end{aligned}$$

which yields a contradiction. Therefore, we conclude that it must be that

$$\min_{s \in \Delta_1} \max_{v \in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) = 0.$$

$\Leftarrow$: Suppose that we have that

$$\min_{s \in \Delta_1} \max_{v \in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) = 0,$$

and assume that there doesn't exist a $\tilde{v} \in Y^2$ such that $R(h_1^{\tilde{v}_1}) = R(h_2^{\tilde{v}_2})$. Then, for all $v \in Y^2$, it holds that $R(h_1^{v_1}) \neq R(h_2^{v_2})$, or equivalently, $R(h_1^{v_1}) - R(h_2^{v_2}) \neq 0$. Let $s \in \Delta_1$ be such that $sign(s_1 - \frac{1}{2}) = -sign(R(h_1^{v_1}) - R(h_2^{v_2}))$ - and also $s_1 \neq \frac{1}{2}$. It then follows that

$$\forall v \in Y^2 \quad (s_1 - \frac{1}{2})(R(h_1^{v_1}) - R(h_2^{v_2})) < 0,$$

from which it follows that

$$\max_{v \in Y^2} (s_1 - \frac{1}{2})(R(h_1^{v_1}) - R(h_2^{v_2})) < 0,$$

and by rewriting we see that

$$S(s|v) - S(\frac{1}{2}\mathbf{1}|v) = (s_1 - \frac{1}{2})(R(h_1^{v_1}) - R(h_2^{v_2})),$$

so then

$$\max_{v \in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) < 0,$$

and thus also

$$\min_{s \in \Delta_1} \max_{v \in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) < 0,$$

but this contradicts our given that $\min_{s \in \Delta_1} \max_{v \in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) = 0$, so we have a contradiction. Therefore, we may conclude that $\exists \tilde{v} \in Y^2$ such that $R(h_1^{\tilde{v}_1}) = R(h_2^{\tilde{v}_2})$.

This concludes the proof. □

Corollary 3.2.1.

$$\min_{s \in \Delta_1} \max_{v \in Y^2} S(s|v) - S(\frac{1}{2}\mathbf{1}|v) < 0.$$

if and only if there does not exist a $\tilde{v} \in Y^2$ such that $R(h_1^{\tilde{v}_1}) = R(h_2^{\tilde{v}_2})$.

Lemma 3.3. *For each $N \in \mathbb{N}_{\geq 1}$, the standard $N-1$ simplex Δ_{N-1} is compact.*

Proof. First, notice that $\Delta_{N-1} \subseteq [0,1]^N \subseteq B(0,1)$, where we use the Euclidean distance for the ball. From this, we conclude that Δ_{N-1} is bounded. Now, let

$$H = \{x \in \mathbb{R}^N \mid \forall i \in \{1, \dots, N\} x_i \geq 0\}$$

and equip it with the subspace topology of the standard topology of $\mathbb{R}^N$, and let $f : H \to \mathbb{R}$ be defined as

$$f((x_i)_{i=1}^N) = \sum_{i=1}^N x_i$$

Take any sequence $(s_i)_{i=1}^\infty \subseteq H$ with $(s_i)_{i=1}^\infty$ such that it converges to some s. Then,

$$\begin{aligned} f(s) &= \sum_{i=1}^N s_i \\ &= \sum_{i=1}^N \lim_{n\to\infty} s_{ni} \\ &= \lim_{n\to\infty} \sum_{i=1}^N s_{ni} \\ &= \lim_{n\to\infty} f(s_n), \end{aligned}$$

so f is sequentially continuous and thus continuous, since this is equivalent in metric spaces. Therefore, f is also topologically continuous, from which we deduce that $f^{-1}(\{1\})$ must be closed, and $f^{-1}(\{1\}) = \Delta_{N-1}$, so Δ_{N-1} is closed.

By the fact that the conjunction of boundedness and closedness are equivalent to compactness in metric spaces, we conclude the proof. □

Lemma 3.4. *The mapping $s \mapsto \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v)$ with domain Δ_{N-1} is a continuous mapping. In particular, the value of*

$$\min_{s \in \Delta_{N-1}} \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v)$$

exists.

Proof. We start by rewriting

$$S(s|v) - S(\frac{1}{N}\mathbf{1}|v) = \sum_{k=1}^N (s_k - \frac{1}{N}) R(h_k^{v_k})$$

to see that

$$s \mapsto S(s|v) - S(\frac{1}{N}\mathbf{1}|v)$$

is continuous, since the projection $\pi_{\mathbb{R}}^{(k)} : \mathbb{R}^N \to \mathbb{R}$ given by $\pi^{(k)}(x) = x_k$ is continuous, $I : \Delta_{N-1} \to \mathbb{R}$ given by $I(s) = (s_k - \frac{1}{N}) R(h_k^{v_k})$ is continuous, composition of continuous functions is continuous, and

$$s \mapsto (s_k - \frac{1}{N}) R(h_k^{v_k}) = (I \circ \pi^{(k)})(s),$$

from which we see that

$$s \mapsto (s_k - \frac{1}{N}) R(h_k^{v_k})$$

is continuous. Moreover, the maximum of an arbitrary finite amount of continuous functions is continuous, so

$$s \mapsto \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v)$$

is continuous. By this, Lemma 3.3, and the Weierstrass theorem, we know that

$$s \mapsto \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v)$$

attains a global minimum on Δ_{N-1}, and thus the value of

$$\min_{s \in \Delta_{N-1}} \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v)$$

exists. □

In practice, what the previous lemma tells us, is that strategies that lie close together will have similar performance in the sense of the minimax value.

Definition 3.5. A strategy $s \in \Delta_{N-1}$ is said to be a strictly mixed strategy if it is not a pure strategy.

Lemma 3.6. *If there exists a pure strategy that beats random sampling, then there also exists a strictly mixed strategy that beats random sampling.*

Proof. In the proof of Lemma 3.4 we have seen that

$$f(s) := \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}|v)$$

is a continuous mapping. Suppose we have a pure strategy $\tilde{s} \in \Delta_{N-1}$ such that $\max_{v \in Y^N} S(\tilde{s}|v) - S(\frac{1}{N}|v) < 0$. It then follows by the definition of continuity that

$$\forall z \in \Delta_{N-1} \forall \varepsilon > 0 \exists \delta > 0 : d(\tilde{s}, z) < \delta \implies |f(\tilde{s}) - f(z)| < \varepsilon,$$

where d denotes the Euclidean distance. Pick $\varepsilon = \frac{1}{2}|f(\tilde{s})|$. It now follows that

$$\forall z \in \Delta_{N-1} \exists \delta > 0 : d(\tilde{s}, z) < \delta \implies |f(\tilde{s}) - f(z)| < \frac{1}{2}|f(\tilde{s})|,$$

which in particular implies that

$$\forall z \in \Delta_{N-1} \exists \delta > 0 : d(\tilde{s}, z) < \delta \implies f(z) < \frac{1}{2} f(\tilde{s}) < 0,$$

so if we pick any $z \in \Delta_{N-1}$ with $d(\tilde{s}, z) < \delta$ and $z \neq \tilde{s}$ - which trivially exists - then we find that $f(z) < 0$. Since this z is strictly mixed, we find a strictly mixed strategy such that

$$\max_{v \in Y^N} S(z|v) - S(\frac{1}{N}\mathbf{1}|v) < 0,$$

which closes the proof. □

Corollary 3.6.1. *If there doesn't exist a strictly mixed strategy that beats random sampling, then there also doesn't exist a pure strategy that beats random sampling. In particular, we cannot beat random sampling.*

Proof. The first claim follows directly from contraposition on Lemma 3.6. The second claim follows from the fact that the space of strategies can be partitioned into a set of strictly mixed strategies and a set of pure strategies, which means that there then would be no strategy that beats random sampling. □

3.4 The behaviour of strategies

We have now seen some results about when it is even possible to beat random sampling. Though, there is still a lot of ground to be covered to understand the problem better. Suppose we focus on strategies that beat random sampling - so not necessarily the optimal strategies. How do these strategies relate? Finding out how strategies relate - in the sense of convex combinations - could provide insights on how to construct a strategy that beats random sampling 'better' than some given strategy that beats random sampling. This is interesting because as we will see in this section, if it is possible to beat random sampling, there exist uncountably infinitely many winning strategies at epsilon-distance from the random sampling strategy. What behaviour do convex combinations of strategies exhibit? In this section, we will provide an exploration to understand the structure of the problem better. That is, we focus on claims that we can make without paying further attention to the hypothesis class, the data, or the loss function.

We will look at the structure of the space of strategies, the structure of our target function

$$s \mapsto \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v),$$

reflections of strategies about the random sampling strategy, the improvement of strategies, the location of the optimal strategies, and we will see a closed form solution to the minimax problem for the case of two unlabelled data points.

Proposition 3.7. *The standard (N-1)-simplex Δ_{N-1} is convex.*

Proof. Pick $\lambda \in [0,1]$ arbitrary. Moreover, pick $s, d \in \Delta_{N-1}$ arbitrary. It is clear that

$$(\lambda s + (1-\lambda)d)_k = \lambda s_k + (1-\lambda)d_k \geq 0,$$

for every $k \in \{1, \ldots, N\}$. This follows from the non-negativity of the entries of s and d and the fact that $\lambda \geq 0$.

Lastly, we need to check that

$$\sum_{k=1}^{N} (\lambda s + (1-\lambda)d)_k = 1.$$

This is easily shown with the calculation

$$\sum_{k=1}^{N} (\lambda s + (1-\lambda)d)_k = \lambda \sum_{k=1} s_k + (1-\lambda) \sum_{k=1}^{N} d_k = \lambda \cdot 1 + (1-\lambda) \cdot 1 = 1.$$

□

What this result tells us is that every strategy that is on the line segment between two strategies is also a valid strategy. This makes it interesting to look at the behaviour of convex combinations strategies. In particular, the behaviour of convex combinations of winning strategies. For this, we introduce some definitions.

Definition 3.8. The value $\nu : \Delta_{N-1} \to \mathbb{R}$ of some strategy $s \in \Delta_{N-1}$ is given by

$$\nu(s) = \max_{v \in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v)$$

Definition 3.9. The winning strategies are the elements of the set

$$\mathbb{W}_{N-1} := \{s \in \Delta_{N-1} \mid \nu(s) < 0\}$$

Definition 3.10. The losing strategies are the elements of the set

$$\mathbb{L}_{N-1} := \{s \in \Delta_{N-1} \mid \nu(s) > 0\}$$

Definition 3.11. The tying strategies are the elements of the set

$$\mathbb{T}_{N-1} := \{s \in \Delta_{N-1} \mid \nu(s) = 0\}$$

Definition 3.12. We call the elements of the standard basis of $\mathbb{R}^N$ the standard strategies. Similarly, we call $\frac{1}{N}\mathbf{1}$ the base strategy.

Proposition 3.13. *If some winning strategy $s \in \mathbb{W}_{N-1}$ admits a reflection about the base strategy in Δ_{N-1}, then this reflection is a losing strategy.*

Proof. The reflection of s about $\frac{1}{N}\mathbf{1}$ is easily seen to be given by $s_{\text{ref}} = \frac{2}{N}\mathbf{1} - s$. Now we can determine the value of the reflection of the winning strategy s:

$$\begin{aligned}
\nu(s_{\text{ref}}) &= \nu(\frac{2}{N}\mathbf{1} - s) \\
&= \max_{v \in Y^N} \sum_{k=1}^{N} ((\frac{2}{N}\mathbf{1} - s)_k - \frac{1}{N}) R(h_k^{v_k}) \\
&= \max_{v \in Y^N} \sum_{k=1}^{N} (\frac{2}{N} - s_k - \frac{1}{N}) R(h_k^{v_k}) \\
&= \max_{v \in Y^N} \sum_{k=1}^{N} (\frac{1}{N} - s_k) R(h_k^{v_k}) \\
&= \max_{v \in Y^N} -\sum_{k=1}^{N} (s_k - \frac{1}{N}) R(h_k^{v_k}) \\
&= -\min_{v \in Y^N} \sum_{k=1}^{N} (s_k - \frac{1}{N}) R(h_k^{v_k}) \\
&\geq -\max_{v \in Y^N} \sum_{k=1}^{N} (s_k - \frac{1}{N}) R(h_k^{v_k}) \\
&= -\nu(s) \\
&> 0.
\end{aligned}$$

Therefore, we have that $s_{\text{ref}} \in \mathbb{L}_{N-1}$. □

What is noteworthy in the previous proposition is that the reflection of a strategy about the base strategy isn't necessarily a strategy again, as the non-negativity of the reflected strategy might be violated. The reflection of a strategy $s \in \Delta_{N-1}$ about the base strategy is again a strategy if and only if $s_k \leq \frac{2}{N}$ for every $k \in \{1, \ldots, N\}$.

Definition 3.14. We call the following set the solution set:

$$\mathbb{S}_{N-1} = \left\{ s \in \Delta_{N-1} \mid \nu(s) = \min_{s \in \Delta_{N-1}} \nu(s) \right\}$$

The value function we defined in 3.8 has a property which will come in handy for a variety of reasons, which we prove in the following theorem.

Theorem 3.15. *The function $\nu : \Delta_{N-1} \to \mathbb{R}$ is convex.*

Proof. Pick $\lambda \in [0,1]$, and $s, d \in \Delta_{N-1}$ arbitrary. Then,

$$\begin{aligned}
\nu(\lambda s + (1-\lambda)d) &= \max_{v \in Y^N} S(\lambda s + (1-\lambda)d|v) - S(\frac{1}{N}\mathbf{1}|v) \\
&= \max_{v \in Y^N} \sum_{k=1}^{N} (\lambda s_k + (1-\lambda)d_k - \frac{1}{N}) R(h_k^{v_k}) \\
&= \max_{v \in Y^N} \sum_{k=1}^{N} (\lambda s_k + (1-\lambda)d_k - \lambda\frac{1}{N} - (1-\lambda)\frac{1}{N}) R(h_k^{v_k}) \\
&= \max_{v \in Y^N} \sum_{k=1}^{N} \lambda(s_k - \frac{1}{N}) R(h_k^{v_k}) + \sum_{k=1}^{N} (1-\lambda)(d_k - \frac{1}{N}) R(h_k^{v_k}) \\
&\leq \max_{v \in Y^N} \sum_{k=1}^{N} \lambda(s_k - \frac{1}{N}) R(h_k^{v_k}) + \max_{v \in Y^N} \sum_{k=1}^{N} (1-\lambda)(d_k - \frac{1}{N}) R(h_k^{v_k}) \\
&= \lambda \max_{v \in Y^N} \sum_{k=1}^{N} (s_k - \frac{1}{N}) R(h_k^{v_k}) + (1-\lambda) \max_{v \in Y^N} \sum_{k=1}^{N} (d_k - \frac{1}{N}) R(h_k^{v_k}) \\
&= \lambda\nu(s) + (1-\lambda)\nu(d).
\end{aligned}$$

□

This theorem has a lot of consequences.

Firstly, it gives proper insight into our optimisation problem; for convex optimisation, there are a lot of options on how to go about solving the problem. In our case, we are dealing with a convex function which isn't necessarily differentiable. For such a scenario, we resort to solving the problem with a method called the projected subgradient method, which we'll discuss in the next section. Though, as mentioned, there are more options available. For example, mirror descent, bundle methods, proximal methods, cutting plane methods, or ADMM.

Secondly, this theorem implies the convex structure of both the set of winning strategies and the solution set. These results will be shown in the following propositions.

Proposition 3.16. *$\mathbb{W}_{N-1}$ is convex.*

Proof. Take $\lambda \in [0,1]$, and $s, d \in \mathbb{W}_{N-1}$ arbitrary. Then by 3.15,

$$\begin{aligned}
\nu(\lambda s + (1-\lambda)d) &\leq \lambda\nu(s) + (1-\lambda)\nu(d) \\
&\leq \lambda \max(\nu(s), \nu(d)) + (1-\lambda)\max(\nu(s), \nu(d)) \\
&= \max(\nu(s), \nu(d)) \\
&< 0,
\end{aligned}$$

so we conclude that $\lambda s + (1-\lambda)d \in \mathbb{W}_{N-1}$, and thus we have proven the desired result. □

Intuitively, what the previous proposition tells us is that line segments between winning strategies contain only winning strategies. From this, we also see that it is impossible for all standard strategies to be winning strategies, since the base strategy is a convex combination of the standard strategies, and the base strategy obviously is not a winning strategy.

Corollary 3.16.1. *Not all standard strategies can be winning strategies.*

Proof. Suppose that all standard strategies were winning strategies. Then, since the base strategy is a convex combination of the standard strategies and by Proposition 3.16 we get that the base strategy must be a winning strategy. Though, the base strategy s_b satisfies $\nu(s_b) = 0$, yielding a contradiction. Therefore, we conclude that not all standard strategies can be a winning strategy. □

Proposition 3.17. *$\mathbb{S}_{N-1}$ is convex.*

Proof. Take $\lambda \in [0,1]$, and $s, d \in \mathbb{S}_{N-1}$ arbitrary. Then by 3.15,

$$\begin{aligned}\nu(\lambda s + (1-\lambda)d) &\leq \lambda\nu(s) + (1-\lambda)\nu(d) \\ &\leq \lambda \max(\nu(s),\nu(d)) + (1-\lambda)\max(\nu(s),\nu(d)) \\ &= \max(\nu(s),\nu(d)) \\ &= \nu(s).\end{aligned}$$

Moreover, for any $t \in \Delta_{N-1}$ it holds that $\nu(t) \geq \nu(s)$ by the definition of the solution set. So, in particular, $\nu(\lambda s + (1-\lambda)d) \geq \nu(s)$. With the two inequalities, we now get that

$$\nu(\lambda s + (1-\lambda)d) = \nu(s) = \min_{s\in\Delta_{N-1}} \max_{v\in Y^N} S(s|v) - S(\frac{1}{N}\mathbf{1}|v),$$

so then it follows that $\nu(\lambda s + (1-\lambda)d) \in \mathbb{S}_{N-1}$. □

Likewise, line segments between global minima of our optimisation problem contain only global minima. More generally, the convex hull of global minima contains only global minima.

From these results we see that there is a lot of structure in our problem. What we haven't discussed yet is how winning strategies compare to the base strategy. As it turns out from the following proposition, certain convex combinations of winning strategies and the base strategy yield once again a winning strategy.

Proposition 3.18. *For all $s \in \Delta_{N-1}$ and $\lambda \in \mathbb{R}_{\geq 0}$, we have that*

$$\nu(\lambda s + (1-\lambda)\frac{1}{N}\mathbf{1}) = \lambda\nu(s).$$

In particular, if $s \in \mathbb{W}_{N-1}$ and $\lambda \in (0,1]$, then $\lambda s + (1-\lambda)\frac{1}{N}\mathbf{1} \in \mathbb{W}_{N-1}$.

Proof. Pick $\lambda \in [0,1]$ and $s \in \Delta_{N-1}$ arbitrary. Then, we see that

$$\begin{aligned}\nu(\lambda s + (1-\lambda)\frac{1}{N}\mathbf{1}) &= \max_{v\in Y^N} \sum_{k=1}^{N} (\lambda s_k + (1-\lambda)\frac{1}{N} - \frac{1}{N}) R(h_k^{v_k}) \\ &= \max_{v\in Y^N} \sum_{k=1}^{N} (\lambda s_k - \lambda\frac{1}{N}) R(h_k^{v_k}) \\ &= \lambda \max_{v\in Y^N} \sum_{k=1}^{N} (s_k - \frac{1}{N}) R(h_k^{v_k}) \\ &= \lambda\nu(s),\end{aligned}$$

which proves the first part of the proposition. From this, the second part follows immediately, using that $\nu(s) < 0$ and the fact that $\lambda > 0$. □

Note that in the first statement of the proof, if λ is too large, the combination might lie outside of the standard simplex. How large λ can be will be discussed later. What this proposition tells us, is that if we have a winning strategy s, then all the strategies on the line segment from s to the base strategy - but excluding the base strategy itself - are again a winning strategy. Therefore, if we were only focused on finding any strategy that beats random sampling, so not necessarily the most optimal strategy, we could limit our search in the standard simplex to any ε-ball around the base strategy for some arbitrary $\varepsilon > 0$, using the Euclidean metric. Since the standard simplex is uncountably infinite (and so is the intersection of any ε-ball around the base strategy with the simplex), this doesn't help us one bit computationally if we were to actually search the space. Though, it does give motivation to use the base strategy as an initial guess in a numerical method to compute a winning strategy, or perhaps even the optimal strategy.

We now understand what happens between a winning strategy and a base strategy. The following step to understand the problem better would be to research what happens outside of a winning strategy and the base strategy. To be more precise, if we were to extend the line segment from the base strategy to a winning strategy all the way to the boundary of the simplex, what kind of value does the resulting strategy yield? We start with some definitions.

Definition 3.19. We call the continuous function $L_s : \mathbb{R}_{\geq 0} \to \mathbb{R}^N$ given by

$$L_s(t) = \frac{1}{N}\mathbf{1} + t(s - \frac{1}{N}\mathbf{1})$$

the path from the base strategy in the direction of $s \in \Delta_{N-1}$.

To tackle the problem mentioned briefly ago, we first examine how far we can travel along the path from the base strategy to some $s \in \Delta_{N-1}$ without leaving the simplex. It is clear that the sum-condition of elements in the standard simplex is satisfied for all $t \in \mathbb{R}_{\geq 0}$, since the path is a path in the hyperplane with all inputs adding up to one. Though, non-negativity might be violated if we travel too far. How far is too far? We want to determine $t_{\max}$ which is the largest $t \in \mathbb{R}_{\geq 0}$ such that $L_s(t) \in \Delta_{N-1}$. Here, note that $t_{\max} \geq 1$, since $L_s(1) = s \in \Delta_{N-1}$. We check under what conditions we have that $(L_s)_k(t) \geq 0$. To this end, we want

$$0 \leq (L_s)_k(t) = \frac{1}{N} + t(s_k - \frac{1}{N}) \quad \forall k \in \{1, \ldots, k\}.$$

Though, we only get an upper bound for t if s_k is such that $s_k - \frac{1}{N} < 0$, since otherwise the inequality flips and we get a lower bound - or if $s_k - \frac{1}{N} = 0$, we have undefined behaviour. This way, we find the following upper bound for t. We can capture this idea with a function $t_{\max} : \Delta_{N-1} \setminus \{\frac{1}{N}\mathbf{1}\} \to \mathbb{R}_{\geq 1}$ defined as

$$t_{\max}(s) := \min\left\{ \frac{\frac{1}{N}}{\frac{1}{N} - s_k} \middle| 1 \leq k \leq N, \frac{1}{N} - s_k > 0 \right\}.$$

Proposition 3.20. *The function t_{max} is well-defined.*

Proof. Suppose that there doesn't exist a $k \in \{1, \ldots, N\}$ such that $s_k - \frac{1}{N} < 0$, then $s_k - \frac{1}{N} \geq 0$ for all k, but since s isn't equal to the base strategy, there must exist a k_0 such that $s_{k_0} \neq \frac{1}{N}$, and so we find that it must hold that $s_{k_0} - \frac{1}{N} > 0$, but then it follows that

$$\sum_{k=1}^{N} s_k \geq \frac{N-1}{N} + s_{k_0} > \frac{N-1}{N} + \frac{1}{N} = 1,$$

which would yield a contradiction with the sum-condition of s. Therefore, there must exist $k_0 \in \{1, \dots, N\}$ such that $\frac{1}{N} - s_k > 0$, and thus

$$\left\{ \frac{\frac{1}{N}}{\frac{1}{N} - s_k} \middle| 1 \leq k \leq N, \frac{1}{N} - s_k > 0 \right\} \neq \emptyset,$$

and so we are allowed to take the minimum since the former set is finite. □

Now that we have established a candidate $t_{\max}$ for the time after which the path from the base strategy to some other non-base strategy would leave the simplex, it is time for us to check whether or not our restrictions were strict enough. That is, for a given $s \in \Delta_{N-1} \setminus \{\frac{1}{N}\mathbf{1}\}$, does it hold that

$$L_s(t_{\max}(s)) \in \Delta_{N-1}?$$

As it turns out, it does. This is illustrated in the following proposition.

Proposition 3.21. *For every $s \in \Delta_{N-1} \setminus \{\frac{1}{N}\mathbf{1}\}$, we have that $L_s(t_{max}(s)) \in \Delta_{N-1}$.*

Proof. Pick any $s \in \Delta_{N-1} \setminus \{\frac{1}{N}\mathbf{1}\}$, and pick any $k \in \{1, \dots, N\}$. Since it is easily checked that $\sum_{k=1}^{N} (L_s)_k(t) = 1$ holds for any $t \in \mathbb{R}$, we will show only that $L_s(t_{\max}(s))$ satisfies non-negativity in this proof. We split into three cases.

First, suppose that k is such that $s_k - \frac{1}{N} = 0$, then it follows immediately from the definition of L_s that $(L_s)_k(t_{\max}(s)) = \frac{1}{N} \geq 0$.

Next, suppose that k is such that $s_k - \frac{1}{N} > 0$. Using the fact that $t_{\max}(s) \geq 1$, and that $s_k - \frac{1}{N} > 0$, it follows that

$$(L_s)_k(t_{\max}(s)) = \frac{1}{N} + t_{\max}(s)(s_k - \frac{1}{N}) \geq \frac{1}{N} + s_k - \frac{1}{N} = s_k > \frac{1}{N} \geq 0.$$

Lastly, suppose that k is such that k is such that $s_k - \frac{1}{N} < 0$. For the remainder of this proof, let k_0 be such that $t_{\max}(s) = \frac{\frac{1}{N}}{\frac{1}{N} - s_{k_0}}$. From the definition of $t_{\max}(s)$ it follows that

$$\frac{\frac{1}{N}}{\frac{1}{N} - s_{k_0}} \leq \frac{\frac{1}{N}}{\frac{1}{N} - s_k},$$

which is equivalent to saying that

$$\frac{1}{N} - s_{k_0} \geq \frac{1}{N} - s_k.$$

From this, we see that

$$\frac{\frac{1}{N} - s_k}{\frac{1}{N} - s_{k_0}} \leq 1,$$

or, equivalently,

$$\frac{1}{N} - \frac{1}{N} \frac{\frac{1}{N} - s_k}{\frac{1}{N} - s_{k_0}} \geq 0.$$

To finish the proof, we turn to the definition of $(L_s)_k(t_{\max}(s))$.

$$\begin{aligned}
(L_s)_k(t_{\max}(s)) &= \frac{1}{N} + t_{\max}(s)(s_k - \frac{1}{N}) \\
&= \frac{1}{N} + \frac{\frac{1}{N}}{\frac{1}{N} - s_{k_0}}(s_k - \frac{1}{N}) \\
&= \frac{1}{N} - \frac{1}{N} \frac{\frac{1}{N} - s_k}{\frac{1}{N} - s_{k_0}} \\
&\geq 0,
\end{aligned}$$

as we saw from the earlier calculation. With this, we have shown that $(L_s)_k(t_{\max}(s)) \geq 0$ for every $k \in \{1, \ldots, N\}$, and thus we have shown the non-negativity property. Therefore, we conclude that $L_s(t_{\max}(s)) \in \Delta_{N-1}$. □

Moreover, by construction of $t_{\max}$ we even have that $t_{\max}(s) = \sup\{t \in \mathbb{R}_{\geq 1} \mid L_s(t) \in \Delta_{N-1}\}$. With this, we have the tools to find strategies on the boundary of the simplex given a direction from the base strategy. The interesting bit about this is that given a winning strategy in the interior of the simplex, we can find an even better winning strategy. This is shown in the following proposition.

Lemma 3.22. *For any $s \in \mathbb{W}_{N-1}$, we have that*

$$\nu(L_s(t_{max}(s))) \leq \nu(s).$$

In particular, if $t_{max}(s) > 1$, it holds that

$$\nu(L_s(t_{max}(s))) < \nu(s).$$

Proof. Using 3.18, we see that

$$\nu(L_s(t_{\max}(s))) = \nu(\frac{1}{N}\mathbf{1} + t_{\max}(s)(s - \frac{1}{N}\mathbf{1})) = t_{\max}(s)\nu(s) \leq \nu(s),$$

since $t_{\max}(s) \geq 1$ and $\nu(s) < 0$. Moreover, the last inequality becomes a strict inequality when $t_{\max} > 1$. □

This lemma tells us that we can improve a found winning strategy if we can 'extend' the strategy along its direction from the base strategy. That is, if we follow the path from the base strategy to a winning strategy, we will find better strategies so long as we keep following the path. In particular, if the path doesn't stop at the supposed winning strategy, we will find a strategy that wins even better. This gives rise to the idea that the optimal solution(s) must lie on the boundary $\partial\Delta_{N-1}$ of the standard simplex.

Moreover, the previous lemma gives inspiration for greedy approaches. Suppose we have some greedy approach that finds winning strategies if they exist - which don't necessarily have to be the optimal strategies. With the previous lemma, we can boost these greedy approaches to make their found winning strategies even better in the sense that they beat random sampling in expectation by a larger margin.

Theorem 3.23. *If $\inf \nu(\Delta_{N-1}) < 0$, then*

$$\mathbb{S}_{N-1} \subseteq \partial\Delta_{N-1}$$

Proof. First note that $\inf \nu(\Delta_{N-1})$ is well-defined, by 3.4

It is clear from $\inf \nu(\Delta_{N-1}) < 0$ that $\frac{1}{N}\mathbf{1} \notin \mathbb{S}_{N-1}$.

Now suppose that we have some (non-base) strategy $s \in \mathbb{S}_{N-1}$ that is not contained in $\partial\Delta_{N-1}$. Then, since s lies in the interior of Δ_{N-1}, we can extend the path from the base strategy to s past s itself, which corresponds to $t_{\max}(s) > 1$. But then

$$\nu(L_s(t_{\max}(s))) < \nu(s)$$

by 3.22, which means that $s \notin \mathbb{S}_{N-1}$, which yields a contradiction. Therefore, it must be that $s \in \partial\Delta_{N-1}$, and thus that $\mathbb{S}_{N-1} \subseteq \partial\Delta_{N-1}$. □

Intuitively, what the former rather abstract statement says, is that if we can beat random sampling, then all optimal solutions are located on the boundary of the standard simplex. What we gain from this theorem, is the insight that we only need to search the boundary of the simplex to find the optimal winning strategies, assuming that they

exist; if they don't exist, we can always fall back to the random sampling strategy. Though, note that here, we treat the concepts interior and boundary from topology rather loosely. Implicitly, we have used an embedding of the standard simplex in a hyperplane so that the interior of the standard simplex is non-empty. For simplicity, this is left out from this thesis, as we aren't that concerned with underlying topologies and the like. Furthermore, using the same embedding as described earlier, we borrow the following result from topology without further proof.

Proposition 3.24. $\partial\Delta_{N-1} = \{s \in \Delta_{N-1} \mid \exists j \in \{1, \dots, N\} : s_j = 0\}$

Using this result, we get an even better understanding of the behaviour of the optimal solutions, using 3.23. Basically, if the base strategy is not contained in the optimal solutions,we have that $\mathbb{S}_{N-1}$ is equal to a subset of an embedding of Δ_{N-2} in Δ_{N-1}. From this, a nice consequence follows for the most basic case where we are dealing with only two unlabelled data points. We show this in the following proposition.

Proposition 3.25. *If* $\inf \nu(\Delta_1) < 0$*, then* $\arg\min_{s\in\Delta_1} \nu(s)$ *is unique. In particular, it holds that* $\mathbb{S}_1 = \{e_1\}$ *or* $\mathbb{S}_1 = \{e_2\}$.

Proof. From 3.23 it follows immediately that $\mathbb{S}_1 \subseteq \partial\Delta_1$. Though, it is impossible for the solution set to contain both the standard strategies by Corollary 3.16.1. Moreover, we know from 3.4 that $\mathbb{S}_1 \neq \emptyset$. Therefore, it must be the case that $\mathbb{S}_1 = \{e_1\}$ or $\mathbb{S}_1 = \{e_2\}$. From this, the uniqueness claim also follows. □

In simpler words, what the previous proposition tells us, is that if we can beat random sampling when dealing with two unlabelled data points, then the optimal active learning strategy is unique and it is a standard strategy. This has the consequence that if the solution set is not equal to the singleton set of either of the standard strategies, then we cannot beat random sampling.

This proposition has a convenient implication. Namely, it provides us a way to compute the value of the minimax problem when we are dealing with two unlabelled data points, as we will see later on in this section.

Corollary 3.25.1. *If* $\mathbb{S}_1 \notin \{\{e_1\}, \{e_2\}\}$*, then* $\inf \nu(\Delta_1) = 0$.

Proof. This follows directly by contraposition on 3.25. □

Proposition 3.26. *If* $\inf \nu(\Delta_1) < 0$*, then there exists* $i \in \{0, 1\}$ *such that*

$$\mathbb{S}_1 = \{e_{2-i}\},$$

$$\mathbb{W}_1 = \left\{\lambda e_{2-i} + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0, 1]\right\},$$

$$\mathbb{L}_1 = \left\{\lambda e_{1+i} + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0, 1]\right\},$$

and

$$\mathbb{T}_1 = \left\{\frac{1}{2}\mathbf{1}\right\}.$$

Proof. Suppose $\inf \nu(\Delta_1) < 0$, then, by 3.25, we get that $\mathbb{S}_1 \in \{\{e_1\}, \{e_2\}\}$. Without loss of generality, assume that $\mathbb{S}_1 = \{e_1\}$, and pick $i = 1$ (if $\mathbb{S}_1 = \{e_2\}$, pick $i = 0$ and proceed the proof analogously). It follows from $\inf \nu(\Delta_1) < 0$ that $\mathbb{S}_1 \subseteq \mathbb{W}_1$ and thus that $e_1 \in \mathbb{W}_1$. Therefore, by 3.18, we see that

$$\left\{\lambda e_1 + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0, 1]\right\} \subseteq \mathbb{W}_1. \tag{6}$$

Moreover, from 3.13 it follows that the reflection of one side of the simplex is given by

$$\left\{\lambda e_1 + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0,1]\right\}_{\text{ref}} = \left\{\lambda e_2 + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0,1]\right\} \subseteq \mathbb{L}_1. \tag{7}$$

Here, we implicitly use that Δ_1 is closed under reflections about the base strategy. From equation 6 and 7, it follows that

$$\begin{aligned}\mathbb{T}_1 &= \Delta_1 \setminus (\mathbb{W}_1 \cup \mathbb{L}_1) \\ &\subseteq \Delta_1 \setminus \left(\left\{\lambda e_1 + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0,1]\right\} \cup \left\{\lambda e_2 + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0,1]\right\}_{\text{ref}}\right) \\ &= \{\frac{1}{2}\mathbf{1}\}.\end{aligned}$$

And since $\frac{1}{2}\mathbf{1} \in \mathbb{T}_1$, we must have that $\mathbb{T}_1 = \{\frac{1}{2}\mathbf{1}\}$. Moreover, from equations 6 and 7, the fact that

$$\left\{\lambda e_1 + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0,1]\right\}_{\text{ref}} = \left\{\lambda e_2 + (1-\lambda)\frac{1}{2}\mathbf{1} \mid \lambda \in (0,1]\right\},$$

and the fact that $\Delta_1 = \sqcup\mathbb{W}_1 \sqcup \left\{\frac{1}{2}\mathbf{1}\right\} \sqcup \mathbb{L}_1$, it follows that we actually must have equality in equations 6 and 7. This completes the proof. □

The former proposition is rather vague due to the cryptic notation that is used to formally describe the situation. In order to make it more clear, we provide a visualization of the proposition. That is, when we are dealing with two unlabelled data points and we can beat random sampling, the standard 1-simplex has a very clear structure that looks as follows:

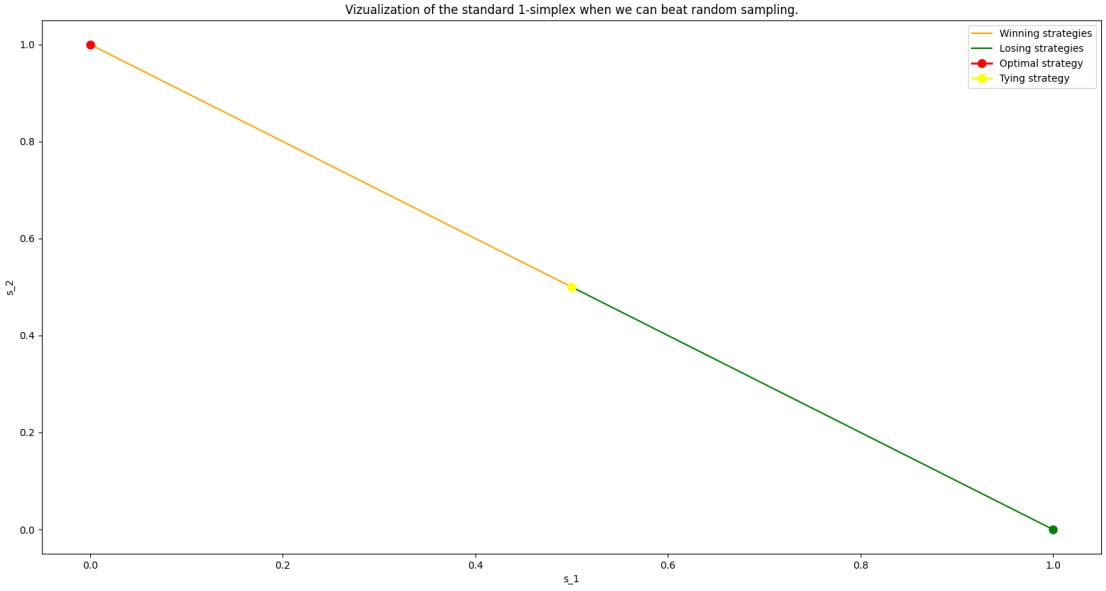

Figure 1: Visualization of Proposition 3.26

Note that the image provided is for the case where $\mathbb{S}_1 = \{e_1\}$, and thus $i = 1$. What we see in this image is that there is a clear development from the value of the first standard strategy to the second strategy.

What makes the previous propositions so notable, is that we now have enough understanding to compute the value $\inf \nu(\Delta_1)$ efficiently. How we do so is shown in the following theorem.

Theorem 3.27.

$$\inf \nu(\Delta_1) = \min\{\nu(e_1), \nu(e_2), 0\}$$

Proof. We keep in mind from Lemma 3.1 that $\inf \nu(\Delta_1) \leq 0$. We prove by case distinction.

First, assume that $\inf \nu(\Delta_1) = 0$. Then, by definition of ν, it follows that $\nu(e_1) \geq 0$ and $\nu(e_2) \geq 0$. Therefore, $\min\{\nu(e_1), \nu(e_2), 0\} = 0$. This proves the first case.

Secondly, assume that $\inf \nu(\Delta_1) < 0$. Then, it follows from Proposition 3.25 that $\mathbb{S}_1 \in \{\{e_1\}, \{e_2\}\}$. Without loss of generality, assume $\mathbb{S}_1 = \{e_1\}$. Then, we have that $\nu(e_1) = \inf \nu(\Delta_1) < 0$. By definition of $\mathbb{S}_1$, we know that $\nu(e_1) < \nu(e_2)$. From this, we conclude that $\min\{\nu(e_1), \nu(e_2), 0\} = \nu(e_1)$. This proves the second case.

With this, we have shown that

$$\inf \nu(\Delta_1) = \min\{\nu(e_1), \nu(e_2), 0\}.$$

□

Now, a question arises naturally: what is the point of knowing how to efficiently compute the optimal active learner when we are dealing with only two unlabelled data points? If we want to apply it in practice, we want to tackle way more than two unlabelled points right? This is a valid concern. Though, what one should realize, is that the previous theorem is the very building block for an algorithm which tackles an arbitrary amount of unlabelled data points. Moreover, there are also active learning settings where we are dealing with two unlabelled data points so that the previous theorem does provide a solution. For example, if we are faced with the situation where we have only two unlabelled data points but the costs to label one of these points is incredibly high, then we would care a lot about which one we should pick to be annotated. The high cost may originate from money, time, both, or required expertise. Examples being medical diagnosis, legal document annotation, or satellite image labelling [5]. This makes the previous theorem impactful.

Something else one might stumble upon is the seemingly inherent exponential time complexity of the problem in the number of unlabelled data points. What is notable about algorithm proposed in the next chapter, is that its time complexity is only quadratic in the amount of unlabelled data points; much faster than exponential time complexity! This makes the algorithm even a viable option for industry.

We will discuss this in a later section.

4 Solving the problem with the projected subgradient method

In the previous section, we have seen that ν is a convex function. Though, ν isn't necessarily differentiable, which makes it harder to optimise. Moreover, we have seen that Δ_{N-1} is convex. With these givens, our problem fits the assumptions for the projected subgradient method. In this section, we will discuss the theory of this numerical method.

4.1 Projecting onto Δ_{N-1}

In the projected subgradient method, we obviously need to know how to do the projection $\Pi_{\Delta_{N-1}}(x)$ for each $x \in \mathbb{R}^N$. That is, we need to find a map $\Pi_{\Delta_{N-1}} : \mathbb{R}^N \to \Delta_{N-1}$ that maps a point in Euclidean space to a point on the standard simplex. For simplicity of this thesis, we will leave out the specifications of the algorithm that computes the projection, and assume that we have it at our disposal from now on. The algorithm has a time complexity of $\mathcal{O}(NlogN)$ [7]. Though, in practice, the time increases only marginally when N is made a lot bigger. For example, it was found to be the case that in similar implementations, when N increases from $N = 5$ to $N = 50$, the time only went up from 1.88s to 2.52s (under certain hardware specifics) [3].

4.2 Subgradients

For this subsection, parts of [1] are used. These are mostly (general) definitions, but also some results - like the result of how to compute the subgradient of the pointwise maximum of convex functions. To this end, we first must prove that we are dealing with this case.

Definition 4.1. For some $v \in Y^N$, we define $\tau_v : \Delta_{N-1} \to \mathbb{R}$ as

$$\tau_v(s) = S(s|v) - S(\frac{1}{N}\mathbf{1}|v).$$

Proposition 4.2. *For each $v \in Y^N$, we have that τ_v is a convex function.*

Proof. Pick $v \in Y^N$, $s, d \in \Delta_{N-1}$, and $\lambda \in [0, 1]$ arbitrary. Then,

$$\begin{aligned}
\tau_v(\lambda s + (1-\lambda)d) &= S(\lambda s + (1-\lambda)d|v) - S(\frac{1}{N}\mathbf{1}|v) \\
&= \sum_{k=1}^{N}(\lambda s_k + (1-\lambda)d_k - \frac{1}{N})R(h_k^{v_k}) \\
&= \sum_{k=1}^{N}(\lambda s_k - \lambda\frac{1}{N} + (1-\lambda)d_k - (1-\lambda)\frac{1}{N})R(h_k^{v_k}) \\
&= \lambda\sum_{k=1}^{N}(s_k - \frac{1}{N})R(h_k^{v_k}) + (1-\lambda)\sum_{k=1}^{N}(d_k - \frac{1}{N})R(h_k^{v_k}) \\
&= \lambda\tau_v(s) + (1-\lambda)\tau_v(d).
\end{aligned}$$

□

Definition 4.3. For some $v \in Y^N$, we define the so-called risk vector to be the vector R_v given by

$$R_v = (R(h_1^{v_1}), R(h_2^{v_2}), \ldots, R(h_N^{v_N}))^T.$$

Proposition 4.4. *For each $v \in Y^N$, we have that τ_v is differentiable. In particular, we have that*

$$\nabla \tau_v(s) = R_v.$$

Proof. Take any $v \in Y^N$ and any $s \in \Delta_{N-1}$. Then,

$$\begin{aligned}\tau_v(s) &= \sum_{k=1}^{N}(s_k - \frac{1}{N})R(h_k^{v_k}) \\ &= \sum_{k=1}^{N} s_k R(h_k^{v_k}) - \frac{1}{N}\sum_{k=1}^{N} R(h_k^{v_k}) \\ &= s^T \cdot R_v - \frac{1}{N} R_v^T \cdot \mathbf{1},\end{aligned}$$

from which it is easy to see that τ_v is differentiable, and that $\nabla \tau_v(s) = R_v$. □

With these results, we have results that will come in handy in a moment after discussing subgradients.

Definition 4.5. We say a vector $g \in \mathbb{R}^N$ is a subgradient of $f : \mathbb{R}^N \to \mathbb{R}$ at $x \in \mathbb{R}^N$ if for all $z \in \mathbb{R}^N$,

$$f(z) \geq f(x) + g^T(z - x).$$

Definition 4.6. A function f is called subdifferentiable at x if there exists at least one subgradient at x.

Definition 4.7. The set of subgradients of f at the point x is called the subdifferential of f at x, and is denoted $\partial f(x)$.

Definition 4.8. A function f is called subdifferentiable if it is subdifferentiable at all $x \in \mathbf{Dom} f$.

A result we borrow from [1] without providing a further proof is the following:

Proposition 4.9. *Suppose $f(x) = \max_{i \in \{1,\ldots,m\}} f_i(x)$, where each f_i is convex and differentiable. Then, we have that*

$$\partial f(x) = \mathbf{Co}\left\{\nabla f_i(x) \mid f_i(x) = f(x)\right\}^3$$

Corollary 4.9.1.

$$\partial \nu(s) = \mathbf{Co}\left\{R_v \mid \tau_v(s) = \nu(s)\right\}$$

Proof. By Proposition 4.4 and 4.2 we know that for each $v \in Y^N$, we have that τ_v is convex and differentiable. Now, by applying Proposition 4.9 and 4.4 we see that

$$\begin{aligned}\partial \nu(s) &= \mathbf{Co}\left\{\nabla \tau_v(s) \mid \nu(s) = \tau_v(s)\right\} \\ &= \mathbf{Co}\left\{R_v \mid \tau_v(s) = \nu(s)\right\},\end{aligned}$$

which closes the proof. □

Proposition 4.10. *There exists a G such that for each $g \in \partial \nu(s)$, we have that*

$$\|g\|_2 \leq G.$$

[3] Note that here, $\mathbf{Co}S$ denotes the convex hull of a set S.

Proof. Note that we can write

$$g = \sum_{v \in Y^N} \lambda_v R_v,$$

where each $\lambda_v \geq 0$ and $\sum_{v \in Y^N} \lambda_v = 1$, since $\partial\nu(s) \subseteq \{R_v \mid v \in Y^N\}$ by Corollary 4.9.1, where we might have that $\lambda_v = 0$ for one or more $v \in Y^N$. Now, let

$$G := \max_{v \in Y^N} \|R_v\|_2.$$

Then,

$$\begin{aligned}
\|g\|_2 &= \left\| \sum_{v \in Y^N} \lambda_v \right\|_2 \\
&\leq \sum_{v \in Y^N} \lambda_v \|R_v\|_2 \\
&\leq \sum_{v \in Y^N} \lambda_v \max_{w \in Y^N} \|R_w\|_2 \\
&= \sum_{v \in Y^N} \lambda_v G = G \sum_{v \in Y^N} \lambda_v = G.
\end{aligned}$$

□

Definition 4.11. Let (X, d) be a metric space, and let $E \subseteq X$. We define the diameter of E to be

$$\operatorname{diam}(E) := \sup\{d(x, y) \mid x, y \in E\}.$$

Proposition 4.12. *diam*$(\Delta_{N-1}) = \sqrt{2}$.[4]

Proof. First, note that for $i \neq j \in \{1, \ldots, N\}$, we have that

$$\|e_i - e_j\|_2 = \sqrt{\sum_{k=1}^{N} (e_i^k - e_j^k)^2} = \sqrt{2},$$

so we clearly have that $\operatorname{diam}(\Delta_{N-1}) \geq \sqrt{2}$.

Moreover, if we take any $x, y \in \Delta_{N-1}$, then we can write

$$\begin{aligned}
\|x - y\|_2^2 &= \|x\|_2^2 + \|y\|_2^2 - 2\langle x, y \rangle \\
&= 1 + 1 - 2\langle x, y \rangle \\
&= 2 - 2\langle x, y \rangle \\
&\leq 2,
\end{aligned}$$

since all entries of x and y are non-negative and thus we have that $\langle x, y \rangle \geq 0$. From this, we may conclude that $\|x - y\|_2 \leq \sqrt{2}$ for every $x, y \in \Delta_{N-1}$. This implies that $\operatorname{diam}(\Delta_{N-1}) = \sup_{x,y \in \Delta_{N-1}} \|x - y\|_2 \leq \sqrt{2}$.

We conclude that $\operatorname{diam}(\Delta_{N-1}) = \sqrt{2}$. □

[4]Here, we are using the Euclidean metric.

4.3 A description of the projected subgradient method

For this description of the method and the convergence proof, parts of [2] are used.

The projected subgradient method is a method which solves the constrained convex optimisation problem

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } x \in \mathcal{C}, \end{aligned}$$

where $\mathcal{C}$ is a convex set. The projected subgradient method is given by

$$x^{(k+1)} = \Pi\left(x^{(k)} - \alpha_k g^{(k)}\right),$$

where Π is the (Euclidean) projection on $\mathcal{C}$, $g^{(k)}$ is any subgradient of f at $x^{(k)}$, and α_k is the step size.

4.4 Convergence of the projected subgradient method

For the convergence of the method, we assume a few things. First of all, we assume that there is a minimizer of f, say $x^\star$.[5] We also make the assumption that the norm of the subgradients is bounded. That is, there is a G such that $\|g^{(k)}\|_2 \leq G$ for all k.[6] Lastly, we'll assume that a number R is known that satisfies $R \geq \|x^{(1)} - x^\star\|_2$.[7]

As opposed to the gradient descent method where the convergence proof is based on the function value decreasing at each step, the (projected) subgradient method is focussed on minimizing the Euclidean distance to the optimal set.

We start with the following observations:

$$\begin{aligned} \|x^{(k+1)} - x^\star\|_2^2 &= \|\Pi_{\Delta_{N-1}}(x^{(k)} - \alpha_k g^{(k)}) - x^\star\|_2^2 \\ &\leq \|x^{(k)} - \alpha_k g^{(k)} - x^\star\|_2^2 \\ &= \|(x^{(k)} - x^\star) - \alpha_k g^{(k)}\|_2^2 \\ &= \|x^{(k)} - x^\star\|_2^2 - 2\langle x^{(k)} - x^\star, \alpha_k g^{(k)}\rangle + \|\alpha_k g^{(k)}\|_2^2 \\ &= \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k g^{(k)T}(x^{(k)} - x^\star) + \alpha_k^2 \|g^{(k)}\|_2^2 \\ &\leq \|x^{(k)} - x^\star\|_2^2 - 2\alpha_k (f(x^{(k)}) - f^\star) + \alpha_k^2 \|g^{(k)}\|_2^2, \end{aligned}$$

where $f^\star = f(x^\star)$. The last line follows from the definition of subgradient, which gives

$$f(x^\star) \geq f(x^{(k)}) + g^{(k)T}(x^\star - x^{(k)}).$$

Applying the inequality above recursively, we have

$$\|x^{(k+1)} - x^\star\|_2^2 \leq \|x^{(1)} - x^\star\|_2^2 - 2\sum_{i=1}^{k} \alpha_i (f(x^{(i)}) - f^\star) + \sum_{i=1}^{k} \alpha_i^2 \|g^{(i)}\|_2^2.$$

Using $\|x^{(k+1)} - x^\star\|_2^2 \geq 0$ and $\|x^{(1)} - x^\star\|_2 \leq R$, we have

$$2\sum_{i=1}^{k} \alpha_i (f(x^{(i)}) - f^\star) \leq R^2 + \sum_{i=1}^{k} \alpha_i^2 \|g^{(i)}\|_2^2. \tag{8}$$

[5] In our situation, this is already the case, given by Theorem 3.4.

[6] This, too, is the case in our situation, as we saw in Proposition 4.10.

[7] This, too, is the case in our situation, since Δ_{N-1} has finite diameter as we saw in Proposition 4.12

Combining this with

$$\sum_{i=1}^{k} \alpha_i (f(x^{(i)}) - f^\star) \geq \left(\sum_{i=1}^{k} \alpha_i \right) \min_{i=1,\ldots,k} (f(x^{(i)}) - f^\star) = \left(\sum_{i=1}^{k} \alpha_i \right) (f_{\text{best}}^{(k)} - f^\star),$$

where we use the notation $f_{\text{best}}^{(k)} = \min\{f(x^{(1)}), \ldots, f(x^{(k)})\}$, we have the inequality

$$f_{\text{best}}^{(k)} - f^\star = \min_{i=1,\ldots,k} f(x^{(i)}) - f^\star \leq \frac{R^2 + \sum_{i=1}^{k} \alpha_i^2 \|g^{(i)}\|_2^2}{2 \sum_{i=1}^{k} \alpha_i}. \tag{9}$$

Finally, using the assumption $\|g^{(k)}\|_2 \leq G$, we obtain the basic inequality

$$f_{\text{best}}^{(k)} - f^\star \leq \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha_i^2}{2 \sum_{i=1}^{k} \alpha_i}. \tag{10}$$

Now if we use step sizes that are square summable, but not summable, we will get convergence. That is, if $\sum_{k=1}^{\infty} \alpha_k^2 < \infty$, and $\sum_{k=1}^{\infty} \alpha_k = \infty$, we get convergence:

$$\lim_{k \to \infty} f_{\text{best}}^{(k)} - f^\star \leq \lim_{k \to \infty} \frac{R^2 + G^2 \sum_{i=1}^{k} \alpha_i^2}{2 \sum_{i=1}^{k} \alpha_i} = 0.$$

An example of a suitable step size is $\alpha_k = \frac{1}{k}$ for $k \in \mathbb{N}_{\geq 1}$. Therefore, it follows that $\lim_{k \to \infty} f_{\text{best}}^{(k)} = f^\star$. Note that there are more convergence results for different types of step sizes, but I choose to limit the convergence results to one class of step sizes for brevity of the thesis.

In our case, we can take $R = \sqrt{2}$ and $G = \max_{v \in Y^N} \|R_v\|_2$, following from propositions 4.12 and 4.10.

With this, we have a suitable method that will solve our original problem. Though, this method doesn't converge very fast; the 'regular' subgradient method already has quite poor performance, and for the projected subgradient method it is even worse, since we have to project onto the standard simplex at each iteration. The known algorithms for this projection have a time complexity of $\mathcal{O}(NlogN)$. Moreover, finding subgradients at each step has a time complexity of at least $\mathcal{O}(2^N)$.

5 Greedy approaches to active learning

In the previous section, we have seen how to solve the problem of finding an optimal strategy that beats random sampling in expectation. Though, this method was at least exponential in terms of time complexity; this time complexity is completely unworkable for large amounts of data[8], so we have to come up with some other method that does not have this problem, so that we can actually use this in practice. With this in mind, we propose two greedy approaches with linear and quadratic time complexity respectively. To this end, we use the results about the $N = 2$ case as building blocks for the algorithms.

The linear time greedy approach will produce a pure strategy: the strategy with 1 at the index which corresponds to the unlabelled data point that would be considered 'the best' in the original solution.[9] That is, if we were to perform an exact search - like the projected subgradient method approach - then we would like the return to be $\tilde{s}$ such that $\tilde{s}_{i_0} = 1$ for

$$i_0 \in \arg \max_{i \in \{1,\dots,N\}} s_i^\star,$$

and $\tilde{s}_j = 0$ for $i_0 \neq j \in \{1, \dots, N\}$, where $s^\star$ corresponds to an optimal solution.

The quadratic time greedy approach pays an additional price as we go from linear time to quadratic time, but the reward we gain for it is that now, the produced strategy contains an ordering similar to that of an optimal solution. That is, if we were to order the indices $i \in \{1, \dots, N\}$ based on $s_i^\star$ for some optimal solution $s^\star$, then we would like the greedy quadratic time algorithm to yield a strategy $\tilde{s}$ such that the ordering of the indices will be the same for $\tilde{s}$ as it was for $s^\star$ - again, when sorted based on s_i.

The main driver for the proposed algorithms is Theorem 3.27, which stated the closed form solution to the problem when working with only two unlabelled data points. With this theorem, we can find the unlabelled data point which is at least as good as every other unlabelled point (pairwise) - in the sense of the ordering we mentioned in the previous paragraph.

What's more to mention is the fact that in this section, we assume M to be fixed. The point of this is that it simplifies the complexity analysis which allows us to highlight the role of N in the time complexity. We do this because we are mostly concerned with how the size of the set of unlabelled data is affecting the time complexity of the algorithm - assuming that the size of the unlabelled data would create the bottleneck in the active learning process.

5.1 A linear time greedy approach

In a nutshell, this approach boils down to a reduction on the list of unlabelled data points. That is, we iterate over the list of unlabelled data points and for each adjacent pair of unlabelled data points, we throw one away until there is only one left in the list. To this end, we make heavy use of Theorem 3.27. From this theorem - and the proof thereof - it is easy to come up with an algorithm $\mathcal{A}$ which takes two arguments $i, j \in \{1, \dots, N\}$ which represent the indices of two unlabelled data points u_i and u_j and returns the index of the unlabelled data point which is at least as good as the other data point. Note that by fixedness of M, the amount of labelled data points, we can implement $\mathcal{A}$ such that the complexity of $\mathcal{A}$ is $\mathcal{O}(1)$. With this in mind, we propose the following algorithm:

[8] Assuming that finding this strategy is the bottleneck in the process.

[9] Original solution refers to a solution to the minimax problem using an exact method.

Algorithm 1 TwoPointActiveLearner

```
procedure A(i, j)
    u ← [u_i, u_j]
    if ν(e_1) < 0 then
        return i
    else if ν(e_2) < 0 then
        return j
    else
        return SAMPLE_UNIFORMLY([i, j])
    end if
end procedure
```

Algorithm 2 Linear Time Greedy Approach

```
procedure WinningIndex(indices)
    return FOLDR(A, N, indices)          ▷ Using foldr from Haskell
end procedure
procedure GreedyStrategy(indices)
    i_0 ← WinningIndex(indices)
    return e_{i_0}
end procedure
```

In the WinningIndex procedure, we used the foldr method as is known from functional programming. This procedure recurses over the unlabelled data points and at each step of the recursion, the constant time algorithm $\mathcal{A}$ is called, yielding an overall time complexity of $\mathcal{O}(N)$.

5.2 A quadratic time greedy approach

As we will see in the data analysis in the next section, trying to come up with a pure strategy that beats random sampling was not very successful.[10] Though, we can make use of the building blocks used for the linear time approach. Instead of only iterating once over the list of unlabelled data points, we can keep iterating to pick the 'best' point until there are no points left to consider. It is clear that this will yield a time complexity of $\mathcal{O}(N^2)$.

[10] In the sense that on average, the produced value by the algorithm is statistically significantly greater than 0, meaning that we get beat by random sampling in expectation.

Algorithm 3 Quadratic Time Ordering

procedure TwoPointOrder(i, j)
  $u \leftarrow [u_i, u_j]$
  if $\nu(e_1) < 0$ **then**
    return true
  else
    return false
  end if
end procedure
procedure NPointOrder
  indices $\leftarrow [1, \ldots, N]$
  $u \leftarrow [u_1, \ldots, u_N]$
  order $\leftarrow [\,]$
  equalities $\leftarrow [\,]$
  for $i = 1$ **to** N **do**
    best $\leftarrow$ WinningIndex(u)
    indices $\leftarrow$ *indices* $\setminus \{best\}$
    if $i \geq 2$ **then**
      append **not** TwoPointOrder(last element of *order*, best) to *equalities*
    end if
    append *best* to *order*
  end for
  return *order*, *equalities*
end procedure

The procedure `NPointOrder` returns a permutation of the list of indices with the intent of containing importance about the unlabelled data points. That is, suppose we take $i \leftarrow ordered[0]$ and $j \leftarrow ordered[1]$, then we want u_i to be a 'better' point than u_j, in the sense that the optimal solution $s^\star$ would have $s_i^\star \geq s_j^\star$. The procedure `NPointOrder` builds on the procedure `WinningIndex` which was introduced in the previous section. By repeatedly running `WinningIndex`, we try to obtain an ordering on the unlabelled points in terms of 'importance', as we just discussed. Additionally, `NPointOrder` returns a boolean list `equalities`; this list serves the purpose of telling us whether or not two consecutive elements from `ordered` correspond to unlabelled points that are equally good. For example, if `equalities[i] = True`, then this means that the unlabelled data points with indices `ordered[i]` and `ordered[i+1]` are equally good. The point of this, is to then enforce the restriction that the unlabelled data points with indices `ordered[i]` and `ordered[i+1]` get equal probabilities in the eventually proposed strategy. Now suppose after running `NPointOrder`, we find that we have the ordering

$$s_{\sigma(1)} \overset{=}{>} s_{\sigma(2)} \overset{=}{>} \ldots \overset{=}{>} s_{\sigma(N)},$$

where $s_i \overset{=}{>} s_j$ means that either $s_i = s_j$ or $s_i > s_j$ holds, and $\sigma : [1, \ldots, N] \to [1, \ldots, N]$ is a permutation.
Now, suppose we are dealing with six unlabelled points and we find the ordering

$$s_4 > s_1 = s_5 = s_2 > s_3 > s_6,$$

then we first create the (decreasing) sequence

$$(q_i)_{i=1}^6 = (q, q^2, q^2, q^2, q^3, q^4),$$

with $q \in (0,1)$ after which we normalize it to make sure it is a valid probability distribution:

$$s = \left(\frac{q}{Z}, \frac{q^2}{Z}, \frac{q^2}{Z}, \frac{q^2}{Z}, \frac{q^3}{Z}, \frac{q^4}{Z} \right),$$

where we let $Z = q + q^2 + q^2 + q^2 + q^3 + q^4$.

Next, we bind the respective indices to these probabilities so that we can reorder the probabilities in such a way that we have the desired probability distribution over the unlabelled points:

$$\left[(4, \frac{q}{Z}), (1, \frac{q^2}{Z}), (5, \frac{q^2}{Z}), (2, \frac{q^2}{Z}), (3, \frac{q^3}{Z}), (6, \frac{q^4}{Z}) \right].$$

This way, we can order the previous list based on the first entry of each element and then map each element to the second entry to get to the desired probability distribution.

Simply put, we find a probability distribution that respects the found ordering; if two entries of s should be equal according to the found ordering, then this method assigns equal probabilities, and if one entry should be bigger than another, then this method also makes sure this happens. We formalize this in the following algorithm.

Algorithm 4 FindStrategy

```
1:  procedure ExponentiatedDecay(order, equalities, q)
2:      curr ← q
3:      s_ordered ← [curr]
4:      for i = 2 to N do
5:          if not equalities[i − 1] then
6:              curr ← q · curr
7:          end if
8:          append curr to s_ordered
9:      end for
10:     return zip(order, s_ordered)
11: end procedure
```

One thing that is notable about this method, is that as q approaches 1, the resulting proposed strategy will tend more and more towards random sampling. This is easily seen by just substituting 1 for q.

Another interesting point to make, is that if the first $k \in \{1, \ldots, N\}$ entries of $(q_i)_{i=1}^N$ are equal according to `equalities`, and we take the limit of q to 0 in this method, then the resulting strategy s has a very clear structure. Namely, $s_i = \frac{1}{k}$ for $i \in \{1, \ldots, k\}$.

This is seen by the following calculation. Take any $i \in \{1, \ldots, k\}$, then

$$\begin{aligned}
\lim_{q \to 0} s_i &= \lim_{q \to 0} \frac{q}{Z} \\
&= \lim_{q \to 0} \frac{q}{k \cdot q + \sum_{i=k+1}^{N} q_i} \\
&\geq \lim_{q \to 0} \frac{q}{k \cdot q + (N-k-1) \cdot q_N} \\
&\geq \lim_{q \to 0} \frac{q}{k \cdot q + (N-k-1) \cdot q^N} \\
&= \lim_{q \to 0} \frac{1}{k + (N-k-1) \cdot \frac{q^N}{q}} \\
&= \lim_{q \to 0} \frac{1}{k + (N-k-1) \cdot q^{N-1}} \\
&= \frac{1}{k},
\end{aligned}$$

so the first k entries of s are at least $\frac{1}{k}$, and they are all equal, and s sums to 1, yielding that $s_i = \frac{1}{k}$ for each $i \in \{1, \ldots, k\}$.

The point in all of this, is to let the user decide on how much risk they want to take in constructing a strategy; lower q-values will be safer since the resulting strategy will look similar to the random sampling strategy, and ν is continuous, so produced values of ν under these strategies will typically be concentrated around 0. Higher q-values will be more risky as we move further away from random sampling, but in return we get the opportunity to beat random sampling. In particular, the further we move away, the harder we can beat random sampling in potential.

5.3 Data Analysis

Now, how good are these strategies? Are we being too greedy? How do these algorithms' performance relate to that of the exact approach we saw in the previous chapter? To this end, we ran some benchmarks using the following set-up. We sample $N = 3$ unlabelled data points uniformly from the interval $[0, 100]$, $M = 10$ labelled data points from the interval $[10, 100]$ for which the corresponding labels were uniformly picked from the set $\{0, 1\}$. Additionally, for the quadratic time greedy approach, a value of $q = 0.5$ was used. Lastly, the hypothesis class was picked to be $\mathcal{H} = \{x \mapsto ax \mid a \in \mathbb{R}\}$ and the loss function was the MSE. This would form one experiment, and these experiments were conducted for a total of 1000 iterations. For each of the iterations with these generated data, we performed the two proposed greedy approaches, alongside with the exact approach (projected subgradient method) which also ran for maximally 1000 iterations. The generated data can be found in this Github repository in the folder `original` under the name of `resultsq0.5.csv`.

Starting off, we are interested in the iterations where any approach found a winning strategy as the optimal solution, to then compare how the linear and quadratic time greedy approached performed. The reason we are interested in these iterations is simply the fact that in these iterations we know for sure that it is possible to beat random sampling, whereas records where no such winning strategy was found does not confirm that it was in that case impossible to beat random sampling; since the exact approach had a maximum amount of iterations for the method to perform, it is possible that the exact method just didn't get enough iterations to find a winning strategy as the optimal strategy. Therefore, we filter the produced data to retain only those records that contain

a negative value in the column for either of the approaches, yielding the following table where we only display the values that the found strategies for each algorithm returned:

Index	Linear Greedy Val	Quadratic Greedy Val	Projected Subgradient Val
0	-0.151317	-0.069133	-0.151317
1	0.039401	-0.012241	0.000000
2	0.010691	0.003082	-0.018778
3	0.357919	0.031308	-0.033381
4	-0.000335	-0.000084	0.000000
5	0.023132	-0.001723	0.000000
6	-0.000180	-0.004799	0.000000
7	0.098512	0.000000	-0.000000
8	0.006078	-0.013054	-0.039636
9	0.505174	0.018656	-0.008413
10	0.086495	0.021624	-0.004815
11	-0.117747	-0.029437	-0.117747
12	0.979108	0.001348	-0.184126
13	0.006212	-0.000678	0.000000
14	0.000991	0.000000	-0.042967
15	0.031691	-0.001792	0.000000
16	0.392975	0.000000	-0.039464
17	0.227277	-0.026283	-0.068224
18	-0.015252	-0.003813	0.000000
19	0.008151	0.002038	-0.001027
20	0.118009	-0.007115	-0.013110
21	0.298286	-0.001342	-0.003360
22	-0.012004	0.008368	0.000000
23	-0.197014	-0.049253	-0.197014
24	-0.033851	0.005200	-0.028396
25	-0.049798	0.000000	-0.049798
26	-0.367117	-0.091779	-0.367117
27	0.021284	0.000000	-0.081506
28	-0.058131	-0.014533	-0.058131
29	-0.013565	-0.021974	0.000000
30	-0.008478	0.000000	-0.044760
31	0.088592	-0.001716	-0.005341

Table 1: Comparison of values across three search algorithms

What is noteworthy is that it happens on multiple occasions that the 'exact'[11] approach happened to return 0 as the minimax value. This is an example of a case where it might be wise to choose a greedy approach instead of the exact approach, as finding a winning strategy would take a very large amount of iterations to find. Though, we don't know when these cases occur.

To summarize the distributional characteristics in the table above, we provide three boxplots:

[11] Here, we put 'exact' in quotes since in practice there is a maximum amount of iterations the method performs, and therefore the implementation is not exact as we might not be iterating till convergence.

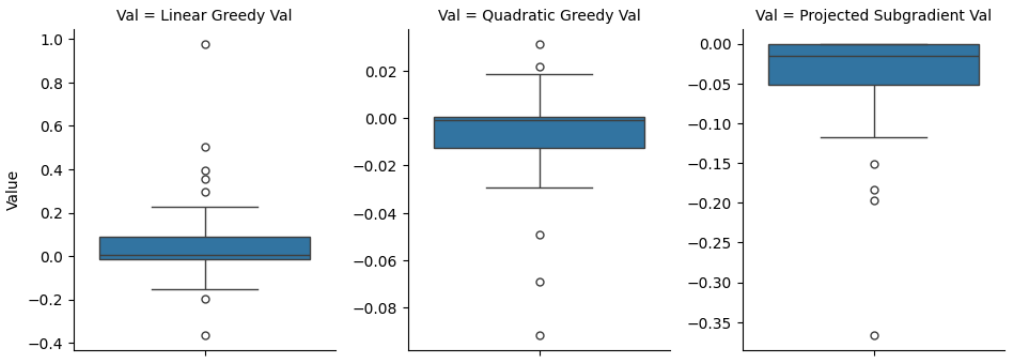

Figure 2: Boxplots of the performance of three algorithms ($n = 32$)

What we see in these boxplots is that for data for which we know it is possible to beat random sampling, the linear time greedy approach performs worst, the quadratic time greedy approach performs second best, and the projected subgradient method performs best. A clear trade-off between time complexity and performance seems to exist. Though, since we also care about the performance of the algorithms on data on which it is not possible to beat random sampling, we will look at all samples to draw conclusions about the general performance of the algorithms. To make further claims about the performance of the greedy approaches and which is better, we resort to hypothesis testing. As a metric to evaluate algorithm performance, we will look at the mean of the values it gave. We will denote the mean of the values of the linear time greedy approach with μ_l, and we will denote the mean of the values of the quadratic time greedy approach with μ_q. We want to perform a paired t-test in this case, with the following hypotheses:

$$\begin{cases} H_0 = \mu_l \leq \mu_q \\ H_a = \mu_l > \mu_q, \end{cases}$$

where we let the significance level be $\alpha = 0.05$ for the remainder of this section. We can also reformulate the hypotheses to come at the paired t-test by letting $\mu_d = \mu_l - \mu_q$. Then,

$$\begin{cases} H_0 : \mu_d \leq 0 \\ H_a : \mu_d > 0 \end{cases}$$

With these hypotheses and data ($n = 1000$), we obtain a t-statistic of $t = 1.9913$, and the corresponding one-sided p-value $p = 0.0277 < \alpha$. With this, we reject H_0 and conclude that there is statistically significant evidence to support the claim that the mean of the group of values of the linear time greedy approach group is greater than the mean of the group of values of the quadratic time greedy approach. The 95% confidence interval for μ_d is $[-0.001916, 0.160311]$.

If we view the expected value of the algorithms as a performance measure, then we may now conclude that the quadratic time greedy approach outperforms the linear time greedy approach.[12] This then gives rise to a trade off: time for performance, as one would expect.

Apart from the fact that the quadratic time greedy approach performs better than the linear time greedy approach, we also aim to quantify the expected performance of each approach individually. To this end, we perform two more hypothesis tests.

[12]Under the assumptions that were made at the beginning of the data analysis section.

First, we focus on the performance of the linear time greedy approach - or rather the lack thereof. We use the following set of hypotheses where we reuse earlier notation:

$$\begin{cases} H_0 : \mu_l \leq 0 \\ H_a : \mu_l > 0 \end{cases}$$

With these hypotheses and data ($n = 1000$), we obtain a t-statistic of $t = 1.6996$, and the corresponding one-sided p-value $p = 0.0496 < \alpha$. With this, we reject H_0 and conclude that there is statistically significant evidence to support the claim that the mean of the group of values of the linear time greedy approach group is greater than 0. The 95% confidence interval for μ_l is $[-0.014219, 0.156419]$.

Therefore, we conclude that it would make little sense to use the linear time greedy approach if one a priori has no knowledge about the data or anything like that, since the data analysis tells us we can expect to lose if we were to use the linear time greedy approach.
Lastly, we investigate the expected value of the quadratic time greedy approach. We use the following set of hypotheses:

$$\begin{cases} H_0 : \mu_q \geq 0 \\ H_a : \mu_q < 0 \end{cases}$$

With these hypotheses and data ($n = 1000$), we obtain a t-statistic of $t = -1.8952$, and the corresponding one-sided p-value $p = 0.0337 < \alpha$. With this, we reject H_0 and conclude that there is statistically significant evidence to support the claim that the mean of the group of values of the quadratic time greedy approach group is less than 0. The 95% confidence interval for μ_q is $[-0.016812, 0.000617]$.

Therefore, we have reason to believe that the quadratic time greedy approach has an expected value of less than 0, making it an effective algorithm in the sense that we expect to win when we use this algorithm.

One should note that these results originate from the assumptions and assertions $N = 3$, $M = 10$, the intervals from which points were sampled, $q = 0.5$, $\mathcal{H} = \{x \mapsto ax \mid a \in \mathbb{R}\}$, $\ell = MSE$, $max\_iter = 1000$, and $n = 1000$. It is possible that these results hold for these specific configurations, though this can only be established in further research. At the very least, we have gained the insight that there are configurations in which the linear time greedy approach fails to perform, and where the quadratic time greedy approach does perform well. Additionally, we have found examples where the greedy approaches did find a winning strategy, whilst the projected subgradient method failed to do so - probably as a consequence of the maximum amount of iterations it could do to find the solution.

6 Synthesis and Outlook

6.1 Discussion and reflection

In this thesis, we have come across numerous roadblocks, and certain results are limited. For example, during the experimentational phase of the greedy approaches, we also incorporated the value of the projected subgradient method on the generated data, causing the experimentational time to explode - it took over five days to run the 1000 iterations. Had we left out this value from the experiment, we would have gotten the results much quicker. Having said that, it would have been possible to program the experiment such that iterations were executed in parallel. Prior to putting my focus on the sequential implementation of the experiment, we tried to program a parallel version, but the speed-up was only marginal and the results seemed a bit strange as they deviated quite a bit from what was generated sequentially. Therefore, we decided to stick with my sequential implementation, though we would advise others to go with a parallel approach, since this could speed-up the implementation by a lot. This is the case since the experiment is mainly one large loop of individual instances of experiments; this makes the program highly parallelizable.

Moreover, as discussed in the Future Research Section, the empirical results are rather limited as we only looked at a certain configuration of the problem, and it is possible that other configurations exhibit different behaviour. What's more, is that this thesis lacks a formal exploration of how much the quadratic time greedy approach deviates from what the projected subgradient method would have found. We omitted this due to a lack of time and it seemed very complex to do so.

Furthermore, this thesis approaches the active learning setting with a pessimistic approach; it might be the case that this approach is 'too pessimistic', in the sense that we don't have to look at every possible labelling of the data, as in practice certain labellings would be very unlikely.

6.2 Future Research

In this thesis, we discussed a pessimistic approach to active learning. We have looked at the structure and inherent properties of the problem, as well as an exact method to solve the problem and several greedy approaches. The exact method is not feasible in practice as it suffers from exponential time complexity. The greedy approaches, on the other hand, don't suffer from this computational bottleneck, but they perform less well in terms of finding a winning strategy.

An important open question right now is what behaviour the $R(h_k^{v_k})$ exhibit for different hypothesis classes, loss functions and data. Even when working with machine learning algorithms that have closed form solutions, it is hard to analyse the behaviour of the $R(h_k^{v_k})$ [13] and to then make use of it in the minimax problem as the target function becomes rather complex. The reason that this open question is important, or impactful if answered, is that we then have more insight into the objective of the minimax. With sufficient understanding of $R(h_k^{v_k})$, it might be possible to prove properties of the problem - for specific loss functions or data configurations.

Moreover, right now we are tackling the active learning problem with a pessimistic approach; we are considering all possible labellings of the data so that we can guarantee to be better than random sampling. Though, some of these labellings are not

[13] Working out the $R(h_k^{v_k})$ for MSE and a simple linear regressor classifier already yields atrociously large and ugly expressions.

very realistic in practice nor is it feasible to consider all of them from a computational point of view. Therefore, it might be interesting to look at a probabilistic algorithm that converges to the right solution in probability under certain circumstances. An example of how one could approach this is the following adaptation of the projected subgradient method. Instead of considering all possible labellings, we uniformly sample $S \subseteq \{1, 2, \ldots, |Y|^N\}$ indices of the enumeration of labellings $(E_i)_{i=1}^{|Y|^N}$. Though, how large should this sampled set of indices be for the method to converge? We introduce some notation. Let

$$\hat{\nu}(s) = \max_{i \in S} \tau_{E_i}(s),$$

we let

$$\varepsilon_S = \nu(s) - \hat{\nu}(s)$$

and we let

$$A_s = \left\{ i \in \{1, 2, \ldots, |Y|^N\} \mid \nu(s) = \nu_{E_i}(s) \right\}.$$

Then,

$$\begin{aligned}
\mathbb{P}(\hat{\nu}(s) < \nu(s)) &= \mathbb{P}(A_s \cap S = \emptyset) \\
&= \frac{\binom{|Y|^N - |A_s|}{|S|}}{\binom{|Y|^N}{|S|}} \\
&\leq \left(1 - \frac{|A_s|}{|Y|^N}\right)^{|S|} \\
&\leq e^{-\frac{|A_s| \cdot |S|}{|Y|^N}}.
\end{aligned}$$

If we now pick S such that $|S| \in \omega\left(\frac{|Y|^N}{|A_s|}\right)$, then it follows that

$$\begin{aligned}
\lim_{N \to \infty} \mathbb{P}\left(\hat{\nu}(s) = \nu(s)\right) &= \lim_{N \to \infty} 1 - \mathbb{P}(\hat{\nu}(s) < \nu(s)) \\
&\geq \lim_{N \to \infty} 1 - e^{-\frac{|A_s| \cdot |S|}{|Y|^N}} \\
&= 1 - \lim_{N \to \infty} e^{-\frac{|A_s| \cdot |S|}{|Y|^N}} \\
&= 1.
\end{aligned}$$

Recall that finding subgradients in the projected subgradient method took exponential time in our case, so what the previous calculations show is that if we pick the sampled set of indices S large enough, we find the exact subgradient asymptotically almost surely. However, the question remains, how do we pick S? To answer this, we must investigate how the set A_s behaves, or at least the cardinality thereof. By doing so, it might be possible to adapt the projected subgradient method in such a way that it becomes feasible to execute for large N.

More generally, one could look further into how the minimax problem for N relates to the minimax problem for $N + 1$; is there a relation? If there is, we might be able to use Theorem 3.27 as a base to build up to the case for arbitrary N.

Furthermore, it may be worthwhile to investigate the quadratic-time greedy approach across various cases, as this approach yielded statistically promising results and perhaps

there is more progress to be made with this approach. Firstly, one could try to adapt the algorithm a little bit so that it performs better. For example, adaptations could be the way of assigning probabilities; we are using some assignment rule inspired by exponentiated decay, but other assignment rules are possible too. Secondly, since the distribution of the values produced by the quadratic time greedy approach was skewed towards the negative side of the reals, with relatively a very short tail to the positive side of the reals, boosting the found strategy towards the boundary of the simplex might yield an even better algorithm - as we would expect from Lemma 3.22. Third, more empirical research could be done like we did in the Data Analysis section to provide more insight into how the quadratic time approach performs under different circumstances: different N, different M, different hypothesis class, different loss functions, et cetera. Lastly, the formulation of relaxations that retain theoretical guarantees while reducing analytical burden, as partially addressed in the appendix.

6.3 Conclusion

In this thesis, we explored a pessimistic formulation of active learning through a theoretical and algorithmic lens. By framing the problem as a minimax optimisation over all possible labellings, we introduced a formal way to assess whether a given strategy is guaranteed to outperform random sampling in the worst case. Our analysis revealed both the potential and the limitations of this approach: while exact solutions exist and can be computed under certain conditions, their practical feasibility is often hindered by exponential time complexity. To address this, we developed and evaluated heuristic strategies, one of which demonstrated promising empirical performance in polynomial time. This opens the door to tractable yet robust active learning strategies that remain faithful to a worst-case rationale.

From a theoretical standpoint, we established foundational results on the structure of the optimisation problem, including the convexity of the objective, the location of optimal strategies being on the boundary of the simplex, and a closed-form solution in the case of two unlabelled data points. These results not only provide insight into the behaviour of pessimistic strategies but also guide the design of algorithms that can exploit this structure. In particular, the projected subgradient method was shown to converge to the exact solution, although its high computational cost limits its scalability.

To mitigate this, we introduced two greedy approximations. While one failed to consistently outperform random sampling, the other showed statistically significant improvements in expectation—providing a viable direction for further research and practical deployment. Moreover, we proposed a boundary-boosting technique to improve any found winning strategy by extending it along its path from the base strategy, thereby potentially increasing its margin over random sampling.

This thesis contributes both theoretical clarity and practical tools to the study of active learning under worst-case assumptions. While challenges remain, especially in terms of computational efficiency, the results presented here demonstrate that it is indeed possible to beat random sampling in expectation—even when preparing for the worst.

7 Appendix A

In this appendix, an adaptation of the pessimistic approach is discussed - which, in a sense, is even more pessimistic, since we give nature more power by allowing it to play mixed as well. Since we have not made significant progress with this approach, it is placed here in an appendix. Though, since this more pessimistic scenario also yields certain (im)possibility results about whether or not random sampling can be beat, we still include this section in the thesis.

7.1 A First Relaxation: from Pure to Mixed Labelling Strategies

Clearly, optimising over the discrete space of all labels, though possible in principle, is quite hopeless for even moderately sized u as the cardinality of the set it is taken from grows exponentially in N. Interpreting the choice of $v \in Y^N$ as the strategy that "nature" deploys, we can make use of a standard relaxation in game theory [Von Neumann] and rather consider the problem

$$\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} \sum_{v\in Y^N} P(v)[S(s|v) - S(\frac{1}{N}\mathbf{1}|v)], \tag{11}$$

where $\mathcal{P}$ denotes the space of all probability mass functions over the set of all possible labellings Y^N. By going from so-called pure strategies (deterministic labellings), to mixed strategies in which labellings are used with some probability, nature becomes stronger and it becomes harder (or just as hard) to find a sampling strategy that improves over random sampling. Therefore, if we do find a solution $s^\star$ whose resulting value of the difference in sampling risks is smaller than or equal to zero, then Equation (5) also evaluates to a value that is zero or smaller for any labelling with the same $s^\star$.

Theorem 7.1. *The value of*

$$\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} \sum_{v\in Y^N} P(v)[S(s|v) - S(\frac{1}{N}\mathbf{1}|v)]$$

exists. Moreover,

$$\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} \sum_{v\in Y^N} P(v)[S(s|v) - S(\frac{1}{N}\mathbf{1}|v)] = \max_{P\in\mathcal{P}} \min_{s\in\Delta_{N-1}} \sum_{v\in Y^N} P(v)[S(s|v) - S(\frac{1}{N}\mathbf{1}|v)].$$

Proof. Let $(n_i)_{i=1}^{C^N}$ be an enumeration of the elements of Y^N, and $P \in \mathcal{P}$ is viewed as a vector such that P_j is the probability of nature using strategy n_j. Now let $L \in \mathrm{Mat}_{N\times C^N}(\mathbb{R})$ be defined by letting $L_{ij} := S(e_i|n_j) - S(\frac{1}{N}\cdot\mathbf{1}|n_j)$ and take any $s \in \Delta_{N-1}$

and any $P \in \mathcal{P}$. Then,

$$
\begin{aligned}
s^T LP &= s^T \begin{pmatrix} S(e_1|n_1) - S\left(\frac{1}{N}\mathbf{1}|n_1\right) & \cdots & S(e_1|n_{C^N}) - S\left(\frac{1}{N}\mathbf{1}|n_{C^N}\right) \\ \vdots & \ddots & \vdots \\ S(e_N|n_1) - S\left(\frac{1}{N}\mathbf{1}|n_1\right) & \cdots & S(e_N|n_{C^N}) - S\left(\frac{1}{N}\mathbf{1}|n_{C^N}\right) \end{pmatrix} P \\
&= s^T \begin{pmatrix} \sum_{i=1}^{C^N} P_i\big[S(e_1|n_i) - S\left(\frac{1}{N}\mathbf{1}|n_i\right)\big] \\ \vdots \\ \sum_{i=1}^{C^N} P_i\big[S(e_N|n_i) - S\left(\frac{1}{N}\mathbf{1}|n_i\right)\big] \end{pmatrix} \\
&= \sum_{k=1}^{N} s_k \sum_{i=1}^{C^N} P_i\big[S(e_k|n_i) - S(\frac{1}{N}\mathbf{1}|n_i)\big] \\
&= \sum_{i=1}^{C^N} P_i \sum_{k=1}^{N} s_k\big[S(e_k|n_i) - S(\frac{1}{N}\mathbf{1}|n_i)\big] \\
&= \sum_{i=1}^{C^N} P_i \sum_{k=1}^{N} s_k\big[R(h_k^{n_{ik}}) - S\left(\frac{1}{N}\mathbf{1} \mid n_i\right)\big] \\
&= \sum_{i=1}^{C^N} P_i \left(\sum_{k=1}^{N} s_k R(h_k^{n_{ik}}) - \sum_{k=1}^{N} s_k S(\frac{1}{N}\mathbf{1}|n_i)\right) \\
&= \sum_{i=1}^{C^N} P_i \left(S(s|n_i) - S(\frac{1}{N}\mathbf{1}|n_i)\right) \\
&= \sum_{v \in Y^N} P(v)[S(s|v) - S(\frac{1}{N}\mathbf{1}|v)]
\end{aligned}
$$

Therefore,

$$
\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} \sum_{v\in Y^N} P(v)[S(s|v) - S(\frac{1}{N}\mathbf{1}|v)] = \min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} s^T LP,
$$

and the Von Neumann minimax theorem now gives that $\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} s^T LP$ exists. Moreover, it gives that

$$
\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} s^T LP = \max_{P\in\mathcal{P}} \min_{s\in\Delta_{N-1}} s^T LP,
$$

which is the desired result. □

7.2 Solving the Minimax Problem with Linear Programming

To estimate the value of Equation (5), we focus on finding

$$
s^\star := \arg \min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} \sum_{v\in Y^N} P(v)\big[S(s|v) - S\big(\frac{1}{N}\mathbf{1} \mid v\big)\big].
$$

Our goal is to transform this minimax formulation into a suitable form for linear programming (LP).

Let $L \in \mathbb{R}^{N\times C^N}$ denote the matrix introduced earlier in Theorem 7.1, where each entry L_{ij} is defined as:

$$
L_{ij} = S(e_i \mid n_j) - S\big(\frac{1}{N}\mathbf{1} \mid n_j\big),
$$

where n_j enumerates the elements of Y^N. The matrix L encodes the differences in sampling risks in all possible labelling strategies.

Note that

$$\min_{s \in \Delta_{N-1}} \max_{P \in \mathcal{P}} s^T L P = \min_{s \in \Delta_{N-1}} \max_{P \in \mathcal{P}} P^T L^T s,$$

since $s^T LP$ is a scalar. Now, let $A := L^T$, and write

$$\min_s \max_P P^T A s$$

instead of

$$\min_{s \in \Delta_{N-1}} \max_{P \in \mathcal{P}} P^T A s.$$

Define $\rho(s) = \max_P P^T As$ and note that $\rho(s) = \max_i (As)_i$. Moreover, note that $w = \max_i (As)_i$ is equivalent to minimizing $w \in \mathbb{R}$ with the conditions that $w \geq (As)_i$ for all i. This gives rise to the following LP-formulation:

minimize $w \in \mathbb{R}$ under the boundary conditions $s \geq 0$ - that is, $s_j \geq 0$ for all j - and

$$\begin{pmatrix} 0 & \mathbf{1}_N^T \\ \mathbf{1}_{C^N} & -A \end{pmatrix} \begin{pmatrix} w \\ s \end{pmatrix} \begin{matrix} = \\ \geq \end{matrix} \begin{pmatrix} 1 \\ \mathbf{0}_{C^N} \end{pmatrix}$$

In order to solve this, we can use an LP-solver like the Simplex algorithm. Though, this seems to only solve for the value of w, but since s is arbitrary, we actually get that $w = \min_s(\max_i (As)_i)$ is embedded in this LP-problem. To see this, we can reformulate our target function that we want to minimize to $c^T x$, where we define $c = \begin{pmatrix} 1 & \mathbf{0}_N \end{pmatrix}$ and $x = \begin{pmatrix} w & s^T \end{pmatrix}^T$. From this, it is clear that minimizing w is equivalent to minimizing $c^T x$ and so we find the value of x that obtains this minimum, if it exists. Now, our solution x gives the value of the game, w, by the Von Neumann minimax theorem, and we find the s that is part of the Nash equilibrium for this game. Therefore, we now know that we can solve (11) by using linear programming.

The problem, with all of this, is that this algorithm too takes exponential time to compute. This is because of the construction of the pay-off matrix for the zero-sum game; this matrix contains an exponential amount of columns and in every column, work is performed, causing the matrix to take exponential time to compute. Not only this, but also the fact that this algorithm only provides an upper bound on the value that we can attain makes this method a somewhat wishful approach. Though, some exploration of this setting has been done, which is shown in the next section.

7.3 Other results

Now that we have found a method to determine the strategy that is part of the Nash equilibrium of our minimax problem, one might wonder: when is our strategy better than random sampling? In this section, we will be exploring the (im)possibilities of active learning when nature is equiped with mixed strategies. Does the pay-off matrix we saw in the previous section have any structure? If so, does it have any implications? As it turns out, the answer to both of these questions is yes.

Lemma 7.2. *When $N = 2$, we have $L_{1j} = -L_{2j}$ for every $j \in \{1, \ldots, C^N\}$.*

Proof.

$$\begin{aligned} R(h_1^{n_{j1}}) + R(h_2^{n_{j2}}) &= 2(\frac{1}{2}R(h_1^{n_{j1}}) + \frac{1}{2}R(h_2^{n_{j2}})) \\ S(e_1|n_j) + S(e_2|n_j) &= 2S(\frac{1}{2}\mathbf{1}|n_j) \\ S(e_1|n_j) - S(\frac{1}{2}\mathbf{1}|n_j) &= S(\frac{1}{2}\mathbf{1}|n_j) - S(e_2|n_j) \\ L_{1j} &= -L_{2j} \end{aligned}$$

□

Lemma 7.3. *For $N = 2$, if both rows of L contain both non-negative and negative elements, then we have*

$$\min_{s\in\Delta_1} \max_{P\in\mathcal{P}} s^T LP = 0.$$

Proof. We have seen earlier in Lemma 3.1 that $\min_{s\in\Delta_1} \max_{P\in\mathcal{P}} s^T LP \leq 0$. Using Lemma 7.2, we see that

$$\begin{aligned} s^T LP &= s^T \begin{pmatrix} L_{1-} \cdot P \\ L_{2-} \cdot P \end{pmatrix} \\ &= s^T \begin{pmatrix} L_{1-} \cdot P \\ -L_{1-} \cdot P \end{pmatrix} \\ &= s_1(L_{1-} \cdot P) - s_2(L_{1-} \cdot P) \\ &= s_1(L_{1-} \cdot P) - (1 - s_1)(L_{1-} \cdot P) \\ &= (2s_1 - 1)(L_{1-} \cdot P) \end{aligned}$$

So using that $s^T L = ((2s_1 - 1)L_{11} \quad (2s_1 - 1)L_{12}) \cdots (2s_1 - 1)L_{1C^N})$, and the fact that $\{L_{1j} \mid 1 \leq j \leq C^N\}$ contains both negative and non-negative elements, we also see that $s^T L$ contains both negative and non-negative entries, from which we conclude that $\max_{P\in\mathcal{P}} s^T LP \geq 0$, by using a pure strategy for P. This implies $\min_{s\in\Delta_1} \max_{P\in\mathcal{P}} s^T LP \geq 0$, which closes the proof. □

Lemma 7.4. *When $N = 2$, if one row is all negative or all non-negative, we find that $\arg\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} s^T LP$ contains only pure strategies.*

Proof. We denote with L_1 the first row of L. Suppose that L_1 is either all negative or all non-negative. Let $[-1, 1] \ni \alpha := 2s_1 - 1$, and note that

$$\max_{P\in\mathcal{P}} s^T LP = \begin{cases} \alpha \max_j L_{1j} \text{ if } \alpha > 0 \\ 0 \text{ if } \alpha = 0 \\ \alpha \min_{1j} \text{ if } \alpha < 0 \end{cases}$$

We now split into two cases. If L_1 contains only non-negative elements, it is easy to see that $\max_{P\in\mathcal{P}} s^T LP$ is minimized for $\alpha = -1$, and likewise if L_1 contains only negative elements, we find that $\max_{P\in\mathcal{P}} s^T LP$ is minimized for $\alpha = 1$. From this it follows that we must have that $s_1 \in \{0, 1\}$, by $s_1 = \frac{\alpha+1}{2}$. Therefore, we may conclude that $\arg\min_{s\in\Delta_{N-1}} \max_{P\in\mathcal{P}} s^T LP$ can only contain pure strategies. □

As it turns out, we will have to focus on nature employing pure strategies only, in the case of having two unlabelled data points and an additional assumption. This is illustrated by the following lemma.

Lemma 7.5. *If $N = 2$, and*

$$\arg\min_{s\in\Delta_1}\max_{P\in\mathcal{P}} s^T LP$$

contains a strictly mixed strategy, then

$$\min_{s\in\Delta_1}\max_{P\in\mathcal{P}} s^T LP = 0$$

Proof. By Lemma 7.4 it follows that the first row of L contains both negative and non-negative elements. By applying Lemma 7.3, we now get that $\min_{s\in\Delta_1}\max_{P\in\mathcal{P}} s^T LP = 0$, which yields the desired result. □

Essentially, what the previous lemma says is that we cannot beat nature with mixed strategies when it is employed with mixed strategies if the set of best performing strategies contain a strategy that is strictly mixed - in the case $N = 2$. This means that we cannot outperform nature with strictly mixed strategies. Therefore, we should do more research on the problem in which nature is only as powerful as using pure strategies. However, one might wonder: can we beat nature with mixed strategies when nature can only use hard labellings? We first document some helpful results to answer this question.

Lemma 7.6. *For every $N \in \mathbb{N}$, and for every $j \in \{1, \dots, N\}$, we have*

$$\sum_{i=1}^{N} L_{ij} = 0$$

Proof. Take $N \in \mathbb{N}$ and $j \in \{1, \dots, N\}$ arbitrary, then

$$\begin{aligned}
\sum_{i=1}^{N} L_{ij} &= \sum_{i=1}^{N} S(e_i|n_j) - S(\frac{1}{N}\mathbf{1}|n_j) \\
&= \sum_{i=1}^{N}\sum_{k=1}^{N}(e_{ik} - \frac{1}{N})R(h_k^{n_{jk}}) \\
&= \sum_{k=1}^{N} R(h_k^{n_{jk}})\sum_{i=1}^{N}(e_{ik} - \frac{1}{N}) \\
&= \sum_{k=1}^{N} R(h_k^{n_{jk}})((\sum_{i=1}^{N} e_{ik}) - 1) \\
&= \sum_{k=1}^{N} R(h_k^{n_{jk}})(1-1) = 0
\end{aligned}$$

□

The lemma tells us that there is an inherent structure in our pay-off matrix. That is, every column in the pay-off matrix sums up to zero, which heavily influences the possible values of the games induced by matrices of the form of L, depending on our data, loss function, and hypothesis class.

Though, it is not clear how this structure of the pay-off matrix plays a role in the behaviour of the LP-approach for the general setting. Only for the case where we are dealing with two unlabelled data points we know how this structure plays a role, as was illustrated in lemmas 7.3 and 7.4.

References

[1] Boyd, Stephen, John Duchi, Mert Pilanci, and Lieven Vandenberghe: *Subgradients*, Apr 2022. `https://web.stanford.edu/class/ee364b/lectures.html`, Notes for EE364b, Stanford University, Spring 2021–22.

[2] Boyd, Stephen and Jaehyun Park: *Subgradient methods*. Lecture Notes, May 2014. `https://web.stanford.edu/class/ee364b/lectures.html`, Notes for EE364b, Stanford University, Spring 2013–14.

[3] Chen, Yunmei and Xiaojing Ye: *Projection onto a simplex*. arXiv preprint arXiv:1101.6081v2, Feb 2011. `http://arxiv.org/abs/1101.6081v2`, arXiv:1101.6081v2 [math.OC].

[4] Lewis, David D. and William A. Gale: *A sequential algorithm for training text classifiers*. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)*, pages 3–12. Springer, 1994. `https://link.springer.com/chapter/10.1007/978-1-4471-2099-5_1`.

[5] Settles, Burr: *Active Learning Literature Survey*. University of Wisconsin-Madison, 2010. `https://minds.wisconsin.edu/handle/1793/64523`.

[6] Tong, Simon and Daphne Koller: *Support vector machine active learning with applications to text classification*. In *Proceedings of the 17th International Conference on Machine Learning (ICML)*, pages 999–1006. Morgan Kaufmann, 2001. `https://www.aaai.org/Papers/ICML/2001/ICML01-143.pdf`.

[7] Wang, Weiran and Miguel A. Carreira-Perpiñán: *Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application*. arXiv preprint arXiv:1309.1541, 2013. `https://arxiv.org/abs/1309.1541`, Version 1, 6 Sep 2013.

[8] Yang, Yifan and Marco Loog: *A benchmark and comparison of active learning for logistic regression*. Pattern Recognition, 83:401–415, 2018.