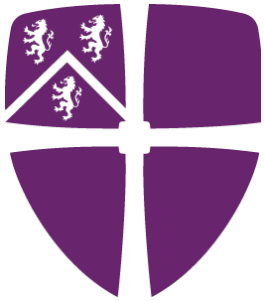


Probability Bounding Methods for Email Spam Filtering

Zaccheus P. Lines

Mathematics Department, Durham University

November 7, 2024

Abstract

This report addresses the challenges associated with email spam classification under uncertainty in prior knowledge by exploring and building on the work of Zaffalon [29] and his Naive Credal Classifier (NCC). The NCC accounts for the inherent uncertainties in spam detection by replacing point probabilities with credal sets. Through the derivation and implementation of novel bounds for interval dominance and through the integration of Zaffalon's [29] dominance criteria with the NCC_ϵ we will show through experimentation that the NCC significantly outperforms the traditional Naive Bayes Classifier (NBC), especially in indeterminate cases offering a promising robust approach to email spam filtering.

Acknowledgments

I extend gratitude to Prof. Matthias Troffaes, whose guidance and support was invaluable throughout the duration of this project. I also thank my parents for their support and my grandfather for instilling in me an early appreciation of mathematics.

Preface

Plagiarism Declaration

This piece of work is a result of my own work and I have complied with the Department's guidance on multiple submission and on the use of AI tools. Material from the work of others not involved in the project has been acknowledged, quotations and paraphrases suitably indicated, and all uses of AI tools have been declared.

Contents

Chapter 1

Introduction

Email has emerged as a fundamental method of communication in the modern day. This ubiquity has been tarnished by the pervasive problem of unsolicited (spam) emails. The problem can be dated to 1978 when Gary Thuerk sent over 400 people an email advertising a new product line from Digital Equipment Corporation, reportedly earning the company over $13 million in sales [31].

Since then, the volume and sophistication of such emails have grown tremendously [23], leading many people to spend a significant amount of time wading through inboxes filled with unwanted communications. This situation highlighted the necessity for an automated method to distinguish legitimate emails (ham) from spam.

Early systems required hand-built heuristics to detect spam [23], which not only demanded a substantial degree of expertise to construct but also lacked adaptability. An effective spam filtering mechanism should have the ability to adapt to changing spam tactics and language itself.

The problem of email spam filtering is fundamentally a classification problem. Classification involves allocating objects to groups based on certain attributes or features associated with those objects [8]. In this context, we say that an object can fall into one of two classes ham or spam.

While there are many different ways to select the features of an email, perhaps the simplest of these would be to consider each unique word in the content of the email as a feature. The classification is then performed by outputting the class that is most likely given the email content. Of course, this process can never be perfect; the very definition of what constitutes spam is a subjective notion, and at best, we can hope for a good approximate solution.

Although the problem of email spam filtering is a binary classification problem involving the two classes, spam and ham, for the sake of extensibility, in this report we will approach the problem from a general multi-category classification and only specialise to the binary case in our experimentation.

Bayesian methods, rooted in classical probability theory, prove highly effective in constructing the probabilistic models that underpin classifiers [3]. The Naive Bayes Classifier, referred to hereafter as the NBC, is among the simplest yet effective classifiers. It uses Bayes' theorem as well as a "naive" assumption of independence among attributes to drastically reduce the complexity of the classification. This assumption of conditional independence may seem highly implausible for many classification scenarios and often it is. However, it has been shown to yield an extremely high degree of efficacy even in situations where the assumption is substantially violated; we will discuss this further in Section 3.2.

Under this assumption, we create a mapping from each instance of the feature set to a class. When considering any classification problem from a purely Bayesian perspective, we are confronted with the problem of precision [15]. Namely, when using a probabilistic model, there is a need to define precisely any probabilities within such a model. In circumstances where not much information is present, specifically in email spam classification, this proves particularly challenging due to the ever-evolving nature of language, spam techniques, and the absence of extensive labelled datasets.

In this report, we will address this problem of imprecision by applying a more robust, conservative framework to the problem of spam filtering. The Naive Credal Classifier (NCC), first introduced by Zaffalon [30], is an extension of the NBC that attempts to account for this imprecision through the lens of imprecise probability theory by contemplating sets of probability distributions (credal sets). By contemplating sets of probabilities rather than the point probabilities of classical probability theory we can account for a range of prior beliefs in the underlying systems we are observing [27].

In terms of the structure of this report, we will first take a look at the Apache Software Foundation [2]'s email corpus on which we will train and test the classifiers we aim to explore throughout this report. We will explore methods posed by Zdziarski [31] behind feature selection and encoding of these features in the context of emails.

We will then follow the work of Bishop [3] to derive the NBC as an objective Bayesian approach to classification and look at various ways of modelling the unknown probabilities associated with it as motivation for our extension into imprecise probabilities. Here we will build on Zaffalon [29] by providing a derivation of his likelihood formula and by deriving the maximum likelihood estimates for certain parameters under the multinomial distribution using Lagrangian optimisation techniques.

The Imprecise Dirichlet Model (IDM), as introduced by Walley [27], is discussed as a method to rigorously account for prior ignorance in the classification process. Our contribution includes the derivation of Dirichlet posterior expectations using integration techniques and properties of the gamma function, which have not been fully addressed in previous work.

Following this, different ways to infer the dominance of one credal set over another will be explored. With the objective Bayesian approach, it is enough to simply compare probabilities p_1 and p_2 and make a classification decision accordingly; the simplicity of such a comparison is no longer afforded to us when dealing with credal sets. We will build on the work of Zaffalon [30] here by exploring interval/stochastic dominance in more detail.

The main result from this report is the derivation and implementation of a novel closed-form bound for the interval on which the probability of observing a class given an instances of a feature set must fall which provides a much simpler way of inferring a form of dominance that does not involve numerical optimisation as with credal dominance. While stochastic/interval dominance is mentioned by Zaffalon [28] a closed-form expression of this exact bound is never put forward.

Then we will tackle the so-called "Feature Problem" [5] of the NCC. The feature problem occurs as a result of zero counts of features when trying to classify. In other words, when encountering a feature that it hasn't encountered in its training. This problem is only relevant in certain domains, email being one of them as the chances of encountering a misspelled or slang word in testing that has not been encountered in training is relatively high. We tackle this problem with the NCC_ϵ introduced by Corani and Benavoli [5]. In doing this we disprove a claim made by Corani and Benavoli [5] on the interval on which ϵ must fall and provide a corrected solution.

Finally, we will conduct three experiments using a novel Python implementation of the NCC_ϵ. These tests aim to evaluate the effectiveness of the NCC_ϵ under both credal dominance and the interval dominance frameworks discussed in this report. The software used for these tests features a modular, object-oriented design, suitable for classification in any dimension. Comprehensive documentation and instructions are available in my dedicated GitHub repository's README file, accessible at `https://github.com/zaccheus-lines/EmailSpamCredalClassifier`. This implementation closely mirrors the functionality of `MultinomialNB` from `sklearn.naive_bayes`, which is utilised for the third experiment in our series.

In summary, our main contributions are highlighted through rigorous proofs, the derivation and implementation of novel bounds for use in interval dominance, and a series of experiments that demonstrate the effectiveness of our proposed modifications to traditional spam filtering techniques. This work not only contributes to the academic field of classification but also to its application in spam filtering.

Chapter 2

Email Data

2.1 The Dataset

An important consideration when building any classification algorithm is the sourcing, preprocessing and encoding of a comprehensive dataset for training and testing. A comprehensive dataset is essential in ensuring that the model is exposed to a wide variety of examples, variations, and edge cases [8].

This domain, characterised by its ever-evolving nature and the adaptability of spammers, necessitates a dataset that encapsulates not just the broad spectrum of legitimate communications but also the diverse tactics employed by spam. A dataset that thoroughly represents the linguistic nuances, cultural variations, and the myriad formats of both genuine and unsolicited emails is critical for training models that can accurately distinguish between spam and non-spam emails [31].

Furthermore bias considerations in email spam classification underscore the importance of a representative dataset [3]. A model trained on a dataset skewed towards certain linguistic or cultural markers may unjustly classify legitimate emails as spam. This not only diminishes the user experience by incorrectly filtering important communications but also raises ethical concerns about the equitable treatment of all users, regardless of their linguistic or cultural background.

It is with these considerations in mind that we have opted for the open source Apache SpamAssassin Public Corpus from Apache Software Foundation [2] as the set of pre-classified emails on which we will train our model. The corpus is a comprehensive collection of both spam and legitimate emails of varying degrees of "spamliness", ranging from easy to detect spam to spam that is extremely nuanced and difficult to classify, as well as legitimate emails that are very difficult to differentiate from spam. Specifically the corpus directory is split into 3 parts, as follows:

- **Spam**: 500 spam messages, all received from non-spam-trap sources[1].
- **Easy Ham**: 2500 non-spam messages. These are typically quite easy to differentiate from spam, since they frequently do not contain any spam signatures (like HTML etc).
- **Hard Ham**: 250 non-spam messages which are closer in many respects to typical spam: use of HTML, unusual HTML markup, coloured text, "spammy" phrases etc.

Total count: 2750 messages, with about an 18% spam ratio.

The structure of an email itself within the corpus can vary but typically comprises the sender's information (email address and name), recipient details, subject line, message body, and attachments. The sender's information identifies the email's origin, while recipient details specify who the email

[1]Non-spam-trap sources refer to legitimate channels or sources, such as personal email accounts, organisational email servers, public mailing lists, and online forums, where users receive emails without the intentional purpose of attracting or capturing spam messages (spam traps).

is for. The subject line summarises the email's content, and the message body contains the main information. Attachments, such as images or documents, can accompany the email. Each component contributes to the email's functionality and communication process. In our implementation we have only looked at the message body but there is scope to go beyond this.

2.2 Tokenisation

Tokenisation is a key aspect of spam filtering; it involves breaking down messages into constituent components or attributes, such as words or symbols [31]. While the process of identifying features may evolve over time, the initial establishment of how these features are parsed from an email is typically programmed by a human. Fortunately, language evolves slowly, requiring only minor adjustments to adapt tokenisation to combat spammer tactics. Tokenisation is usually a heuristic process often defined once at build time and rarely requiring further maintenance due to its simplicity [31].

Some filters may autonomously analyse messages, akin to DNA sequencing, to determine the optimal method of extracting data. Research by Ogundepo et al. [17] has utilised this approach to filter languages lacking spaces or other word delimiters, demonstrating the potential for enhanced efficiency and adaptability in spam filtering mechanisms.

For the purposes of our filtering we will stick to a heuristic approach. The simplest and perhaps most intuitive of these heuristic approaches is word-by-word tokenisation as follows:

Original Phrase	**Tokens**
Meeting scheduled at noon!	[Meeting, Scheduled, At, Noon]

Table 2.1: Example of Tokenisation

There is notable validity in the question: should "noon" have a different disposition than "noon!" or "Meeting" than "meeting"? Capitalisation and punctuation can significantly impact an email's perception and are commonly leveraged by spammers to attract attention or convey urgency [31]; however including too much punctuation could result in five or ten different permutations of a word in the tokenisation drastically reducing their utility.

Furthermore tokenising text into n-grams, rather than just individual words, can significantly enhance analysis. For instance, consider the phrases "Free money" and "Are you free this evening?". When tokenised into individual words, the unique context and meaning of "Free" in "Free money" as a single entity gets lost, making it indistinguishable from the "free" in "Are you free this evening?". However, by using bi-grams (2-word n-grams), the phrase "Free money" is preserved as a single token, maintaining its specific meaning. This approach allows algorithms to better capture the nuances of language, such as idiomatic expressions, proper nouns, and specific terminologies that lose their distinct meanings when broken down into individual words. Consequently, n-gram tokenisation can lead to more accurate and contextually aware spam filters.

For the sake of tractability we are going to use the white space as a delimiter and tokenise into words. While one could design an algorithm to do the tokenisation from scratch I have used the Python libraries `Email.Parser` [1] and `Beautiful Soup` [21] in our implementation [16].

2.3 Data Encoding

For now let us explore a method for encoding an email by contemplating a toy dataset, containing basic email content you might expect to receive in various types of email. The techniques we employ will be scalable to the actual dataset.

Toy Email	**Classification**
"Free cash now"	Spam
"Win a vacation!"	Spam
"Meeting scheduled at noon"	Non-Spam
"Your invoice"	Non-Spam
"Click to claim prize"	Spam
...	...

Table 2.2: Toy Dataset.

Once we have a labelled dataset as above we need a way to encode both the email content and its respective classification. We take a feature set $\mathbf{X}$ to be a feature vector [24] with each dimension taking on the binary representing of either the presence or absence of a particular word. Namely let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a vector of binary random variables, where each X_i indicates the presence (1) or absence (0) of a particular word in the email,

$$X_i = \begin{cases} 1 & \text{if word } i \text{ is present in the email} \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

As such, we can encode each of our emails in a feature vector $\mathbf{x}$. This process begins with the construction of a vocabulary. There are multiple ways in which we could do this; one option is to pre-define a dictionary of words that we expect to see in the email content. The problem with this is that it does not account for incorrectly spelt or slang words that may not be included in the dictionary. Instead we will construct our vocabulary as a list of all unique words encountered across our entire email corpus. Each word in this vocabulary is then assigned a specific position in our feature vectors, leading to high-dimensional representations of each email as shown in table 2.3.

Toy Email	**Feature Vector**					
	Free	Win	Noon	Click	Prize	...
"Free cash now"	[0	1	0	0	1	...]
"Free holiday offer"	[1	0	0	0	0	...]
"Meeting scheduled at noon"	[0	0	1	0	0	...]
"Click to claim prize"	[0	0	0	1	1	...]

Table 2.3: Feature Vector construction.

Clearly in practice we will expect these feature vector to be extremely sparse with the vocabulary being of magnitude 10^5 where as some emails may only contain a few words. We can consider this when we come to a practical implementation as we can employ data compression techniques to save on memory usage.

Chapter 3

Naive Bayes Classifier

3.1 Bayes' Theorem

Now that we have a comprehensive dataset and a method to encode this data, we need a classification method. For a lot of what follows we will be loosely following the work of Bishop [3] to build the NBC. Formally, we define a classifier as follows:

Definition 3.1.1 (Classifier [3])**.** A **classifier** is a function $f : X \to C$, where X is a set of inputs and C is a set of class labels. The function $f(x) = c$ maps each input $x \in X$ to a class label $c \in C$.

There are many different ways to construct a classification algorithm, in particular we will first focus on Bayesian methods due to their unique ability of expressing uncertain knowledge, their capability to convey probabilities in detail, and their ability to learn incrementally by incorporating prior knowledge in the system; specifically we turn to the Naive Bayes Classifier (NBC). The NBC is among the most basic instances of a Bayesian network; a graphical framework used to represent a joint probability distribution. These graphs are at the forefront of probabilistic reasoning [18].

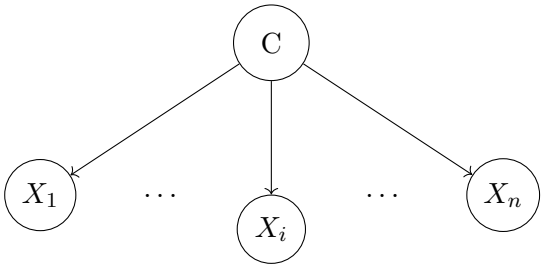


Figure 3.1: The Naive Bayes Classifier.

Given a particular instance of our feature vector $[X_1, ..., X_n] = [x_1, ..., x_n]$ a classification is made by computing the posterior probability of observing a class given that instance, $p(c|\mathbf{x}) = p(c|x_1, ..., x_n)$. As with most Bayesian method the key to this calculation lies in Bayes' theorem [3]. Given the class variable C and a feature set $\mathbf{X}$, the probability of class C being c given an instance of the feature set $\mathbf{x}$, is expressed by Bayes' theorem as:

$$p(c \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid c) \cdot p(c)}{p(\mathbf{x})} \tag{3.1}$$

In the context of email filtering we take our class variable C to be:

$$C = \begin{cases} c_1 = \text{Spam} \\ c_2 = \text{Ham} \end{cases} \tag{3.2}$$

3.2 Naive Assumption

Without assumptions of independence of each X_i from every other X_j for $i \neq j$, given the category C, computing the joint probability $p(\mathbf{x} \mid c) = p(x_1, ..., x_n \mid c)$ proves difficult. Below we can see the calculation needed to calculate the joint probability under normal circumstances.

$$p(x_1, \ldots, x_n|c) = p(x_1|c) \cdot \ldots \cdot p(x_n|x_1, \ldots, x_{n-1}, c) \tag{3.3}$$

In fact not only is the order of complexity of this calculation worst-case exponential $O(2^n)$ [22] but also, a lot of the interaction terms are practically difficult to predict. We therefore impose the naive assumption of conditional independence on our random variables X_i, an idea first introduced by Good [9] in his aptly named Naive Bayesian Classifier. Under the naive independence assumption, this joint probability simplifies to the product of individual probabilities:

$$p(\mathbf{x} \mid c) = \prod_{i=1}^{n} p(x_i \mid c) \tag{3.4}$$

It must not go unchecked that this assumption of feature independence is notably violated by the intrinsic linguistic dependencies between words fundamental to the structure of language. However, the NBC frequently demonstrates a surprising level of accuracy across various applications, in particular spam filtering. For some time, the reason behind this seemingly implausible efficacy under the naive assumption went unknown; subsequently theoretical insights by Zhang [32] have illuminated the underlying reasons for this performance. Zhang [32] demonstrated that the key factor lies in the distribution of dependencies. Specifically, how a node's local dependencies are spread across each class, whether uniformly or disproportionately, and how the local dependencies of all nodes collectively contribute, either coherently reinforcing a specific classification or conflicting and negating each other. Thus, the NBC can remain optimal regardless of the strength of attribute dependencies, provided these dependencies are either balanced across classes or neutralise one another. The advantages naive assumption affords in tractability and computational efficiency far outweigh the draw-backs associated with it.

3.3 MAP Decision Rule

So far we have clearly outlined Bayes' probability model for calculating the posterior $p(c|\mathbf{x})$. We then look to take our class estimator $\hat{c}$ to be the class maximising the posterior probability of being in a class given a certain feature set; this is suitably named *maximum a postiori* [7].

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, p(c \mid \mathbf{x}). \tag{3.5}$$

Intuitively here we are saying; given this email, is it more likely to be ham or spam?

We combine this with our findings in Bayes' theorem above to obtain an expression in terms of our prior probability and our likelihood. Notice our evidence, $p(\mathbf{x})$, on the denominator is omitted as this is a normalising constant and so does not depend on the class argument over which we are maximising.

$$\hat{c} = \underset{c}{\operatorname{argmax}}\, p(c) \prod_{i=1}^{n} p(x_i \mid c). \tag{3.6}$$

3.4 Multinomial Sampling

Now a key characteristic of the Bayesian framework is that we must know the prior $p(c)$ and likelihoods $p(x_i|c)$ in order to conduct inference. The issue here, as with many other problems, is that we do not necessarily know these probabilities, and so we must treat these as unknown parameters themselves. Let us consider the sample space C. Then assuming the standard multinomial, we have N observations drawn independently from C. Now let $n(c)$ and $n(x_i, c)$ denote the observed frequencies of the class $c \in C$ and $(x_i, c) \in (X_i \times C)$ in the N observations respectively. Let us define the parameters involved in a Bayesian model as follows: $\theta_{c,\mathbf{x}}$ denotes the chances of the multinomial distribution $(c, \mathbf{x}) \in (C \times X_1 \times ... \times X_k)$; θ_c the chance of a class c; $\theta_{x_i|c}$ as the chance of the instance of a word x_i conditional on the class c; and similarly $\theta_{\mathbf{x}|c}$ as the chance of the instance of an feature vector $\mathbf{x} = [x_1, ..., x_k]$ conditional on c. In doing this we obtain a new parameterised expression for the joint:

$$\theta_{c,\mathbf{x}} = \theta_c \prod_{i=1}^{k} \theta_{x_i|c}. \tag{3.7}$$

Let us now look to the underlying distribution of these parameters namely the data is distributed multinomially. So taking $\mathbf{n}$ to be the vector containing all the counts $n(c)$ and $n(x_i, c)$, and $\boldsymbol{\theta}$ as the vector whose elements are the chances of the multinomial distribution $\theta_{c,\mathbf{x}}$. Then the likelihood of observing $\boldsymbol{\theta}$ given $\boldsymbol{n}$ is given by Zaffalon [29] in his 2001 paper of credal classifiers, however he does not derive this. We build on Zaffalon by providing our own derivation as such:

$$\begin{aligned}
L(\boldsymbol{\theta}|\mathbf{n}) \propto \prod_{j=1}^{N} \theta_{c^{(j)},\mathbf{x}^{(j)}} &= \prod_{j=1}^{N} \left(\theta_{c^{(j)}} \prod_{i=1}^{k} \theta_{{x_i}^{(j)}|c^{(j)}} \right) \\
&= \prod_{c \in C} \left[\prod_{\substack{j=1 \\ c^{(j)}=c}}^{N} \theta_{c^{(j)}} \prod_{i=1}^{k} \theta_{{x_i}^{(j)}|c^{(j)}} \right] \\
&= \prod_{c \in C} \left[\theta_c^{n(c)} \prod_{\substack{j=1 \\ c^{(j)}=c}}^{N} \prod_{i=1}^{k} \theta_{{x_i}^{(j)}|c^{(j)}} \right] \\
&= \prod_{c \in C} \left[\theta_c^{n(c)} \prod_{\substack{j=1 \\ c^{(j)}=c}}^{N} \prod_{\substack{i=1 \\ x_i^{(j)}=x_i}}^{k} \left(\prod_{x_i \in X_i} \theta_{{x_i}^{(j)}|c^{(j)}} \right) \right] \\
&= \prod_{c \in C} \left[\theta_c^{n(c)} \prod_{i=1}^{k} \prod_{x_i \in X_i} \prod_{\substack{j=1 \\ x_i^{(j)}=x_i \\ c^{(j)}=c}}^{N} \theta_{{x_i}^{(j)}|c^{(j)}} \right] \\
&= \prod_{c \in C} \left[\theta_c^{n(c)} \prod_{i=1}^{k} \prod_{x_i \in X_i} \theta_{x_i|c}^{n(x_i,c)} \right]
\end{aligned} \tag{3.8}$$

3.5 Maximum Likelihood Estimate

With the likelihood just derived in mind an intuitive approach towards estimating these parameters could be to calculate the *maximum likelihood estimate* from our N observations. Optimising this product of priors directly would prove difficult. Since the log function is monotonically increasing [7] we can simplify the problem by optimising the log likelihood which allows us to take better advantage of the linearity of the partial differential operator. We get the *log-likelihood* as follows:

$$\begin{aligned}\log L &= \log\left(\prod_{c\in C}\left[\theta_c^{n(c)}\prod_{i=1}^{k}\prod_{x_i\in X_i}\theta_{x_i|c}^{n(x_i,c)}\right]\right)\\ &= \sum_{c\in C}\left[\log(\theta_c^{n(c)}) + \sum_{i=1}^{k}\sum_{x_i\in X_i}\log(\theta_{x_i|c}^{n(x_i,c)})\right]\\ &= \sum_{c\in C}\left[n(c)\log(\theta_c) + \sum_{i=1}^{k}\sum_{x_i\in X_i} n(x_i,c)\log(\theta_{x_i|c})\right]\end{aligned} \tag{3.9}$$

Now that we have the *log-likelihood* we need to maximise it with respect to all of our parameters $\theta_c, \theta_{x_i|c}$. In particular this is a constrained optimisation since we know $\sum_{c\in C}\theta_c = 1$ and $\sum_{x_i\in X_i}\theta_{x_i|c} = 1$ as each event is mutually exclusive and their sum exhaustive, so we have the following constraints.

$$g_c(\theta_c) = \sum_{c\in C}\theta_c - 1 = 0 \tag{3.10}$$

$$g_{x_i|c}(\theta_{x_i|c}) = \sum_{x_i\in X_i}\theta_{x_i|c} - 1 = 0 \quad i \geq 1 \tag{3.11}$$

Now we turn to the Lagrangian as a method for optimising under these constraints:

Definition 3.5.1 (Lagrangian [26])**.** Given a function $f : \mathbb{R}^n \to \mathbb{R}$ that is subject to optimisation and a constraint function $g : \mathbb{R}^n \to \mathbb{R}$ such that $g(\mathbf{x}) = c$, where $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n$, the *Lagrangian* $\mathcal{L}(\mathbf{x}, \lambda)$ is defined as:

$$\mathcal{L}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda\left(g(\mathbf{x}) - c\right),$$

where λ is the Lagrange multiplier and c is a constant.

The Lagrangian facilitates the solution to constrained optimisation problems by converting them into unconstrained optimisation problems in a higher-dimensional space [26]. Given a constraints g the Lagrangian $\mathcal{L}$ for the MLE optimisation is:

$$\mathcal{L} = \log L - \lambda(g(\theta)) \tag{3.12}$$

Differentiating once with respect to θ and equating to zero in order to find points of local extrema we obtain the following equality:

$$\frac{\partial\mathcal{L}}{\partial\theta} = 0 \implies \frac{\partial \log L}{\partial\theta} = \lambda\frac{\partial g}{\partial\theta} \tag{3.13}$$

Calculating the derivatives of $\log L$ for θ_c and $\theta_{x_i|c}$ respectively we obtain:

$$\frac{\partial \log L}{\partial\theta_c} = \frac{n(c)}{\theta_c} \tag{3.14}$$

$$\frac{\partial \log L}{\partial\theta_{x_i|c}} = \frac{n(x_i,c)}{\theta_{x_i|c}} \tag{3.15}$$

Furthermore the derivatives of the constraints g_c and $g_{x_i|c}$ are:

$$\frac{\partial g_c}{\partial \theta_c} = \frac{\partial}{\partial \theta_c}\left(\sum_{c\in C}\theta_c - 1\right) = 1 \tag{3.16}$$

$$\frac{\partial g_{x_i|c}}{\partial \theta_{x_i|c}} = \frac{\partial}{\partial \theta_{x_i|c}}\left(\sum_{x_i\in X_i}\theta_{x_i|c} - 1\right) = 1 \tag{3.17}$$

Plugging these derivatives back into the equality eq. (3.13) we obtain the following estimate for θ_c:

$$\frac{n(c)}{\hat{\theta}_c} = \lambda_c \tag{3.18}$$

$$\implies \hat{\theta}_c = \frac{n(c)}{\lambda_c} \tag{3.19}$$

Similarly we obtain the following estimates for $\theta_{x_i|c}$:

$$\frac{n(x_i, c)}{\hat{\theta}_{x_i|c}} = \lambda_{(x_i,c)} \tag{3.20}$$

$$\implies \hat{\theta}_{x_i|c} = \frac{n(x_i, c)}{\lambda_{(x_i,c)}} \tag{3.21}$$

Plugging the estimate $\hat{\theta}_c$ back into the first constraint eq. (3.10) we obtain λ_c

$$\sum_{c\in C}\theta_c = \sum_{c\in C}\frac{n(c)}{\lambda_c} = 1 \tag{3.22}$$

$$\implies \lambda_c = \sum_{c\in C} n(c) = N \tag{3.23}$$

Plugging the estimate $\hat{\theta}_{x_i|c}$ back into the second constraint eq. (3.11) we obtain $\lambda_{(x_i,c)}$

$$\sum_{x_i\in X}\theta_{x_i|c} = \sum_{x_i\in X_i}\frac{n(x_i, c)}{\lambda_c} = 1 \tag{3.24}$$

$$\implies \lambda_{(x_i,c)} = \sum_{x_i\in X_i} n(x_i, c) = n(c) \tag{3.25}$$

In addition to this we can verify that these are both maxima and not minima since at our estimate

$$\frac{\partial^2 \log L}{\partial \theta_c^2} = -\frac{n(c)}{\hat{\theta}_c^2} = -\frac{N^2}{n(c)} \leq 0 \tag{3.26}$$

$$\frac{\partial^2 \log L}{\partial \theta_{x_i|c}^2} = -\frac{n(x_i, c)}{\hat{\theta}_{x_i|c}^2} = -\frac{n(c)^2}{n(x_i, c)} \leq 0 \tag{3.27}$$

So these are shown to be concave at $\hat{\theta}_c$ $\hat{\theta}_{x_i|c}$ as such it must be a maximum and so the *maximum likelihood estimates* for θ_c and $\theta_{x_i|c}$ are shown to be:

$$\hat{\theta}_c = \frac{n(c)}{N} \tag{3.28}$$

$$\hat{\theta}_{x_i|c} = \frac{n(x_i, c)}{n(c)} \tag{3.29}$$

□

Chapter 4

The Imprecise Dirichlet Model

4.1 Conjugate Prior

We now turn to the conjugate prior [19] as an algebraic convenience in giving a closed-form expression for the posterior. It also offers us insight into how a likelihood function updates a prior distribution. Namely we will motivate the multinomial case with the binomial. For a binomial likelihood [7] with parameter θ, the likelihood function given k successes in n trials is:

$$L(k|\theta, n) = \binom{n}{k}\theta^k(1-\theta)^{n-k} \tag{4.1}$$

We then look to combine the likelihood with the conjugate prior to the binomial distribution, the Beta distribution:

$$\text{Beta}(\alpha, \beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)} \tag{4.2}$$

With the normalisation constant given by the Beta function ensuring the posterior integrates to 1:

$$B(\alpha', \beta') = \int_0^1 t^{\alpha'-1}(1-t)^{\beta'-1}dt \tag{4.3}$$

In doing this we obtain the following expression obtain the posterior:

$$p(\theta|k, n) \propto \theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1} \tag{4.4}$$

So, the complete posterior is:

$$p(\theta|k, n) = \frac{\theta^{k+\alpha-1}(1-\theta)^{n-k+\beta-1}}{B(k+\alpha, n-k+\beta)} \tag{4.5}$$

4.2 Extension to multinomial sampling

For a multinomial likelihood with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)$ representing the probabilities of different classes $c \in C$ the likelihood function given $\mathbf{n}$ successes in N trials across these classes is:

$$L(\mathbf{n}|\boldsymbol{\theta}) \propto \prod_{c \in C} \theta_c^{n(c)} \tag{4.6}$$

where $\sum_{c \in C} n(c) = N$ and $\sum_{c \in C} \theta_c = 1$. We combine our likelihood with a Dirichlet prior distribution,

$$\text{Dir}(s, \boldsymbol{t}) \propto \prod_{i=1}^{k} \theta_i^{st_i-1} \tag{4.7}$$

with the normalisation constant ensuring the integral of the posterior over the simplex of possible values of θ is 1. In an effort to define this normalisation constant let us first define the Gamma function as follows:

Definition 4.2.1 (Gamma Function [11])**.** For any $z \in \mathbb{C}$ such that $Re(z) > 0$ we define the gamma function of z as such:

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx$$

Theorem 4.2.1 (Recursive Relation of Γ [11])**.**

$$\Gamma(z+1) = z\Gamma(z)$$

Proof [1]

$$\begin{aligned}
\Gamma(z+1) &= \int_0^\infty x^z e^{-x} dx && (4.8)\\
&= \left[-x^z e^{-x}\right]_0^\infty + \int_0^\infty z x^{z-1} e^{-x} dx && (4.9)\\
&= \lim_{x\to\infty}(-x^z e^{-x}) + z\int_0^\infty x^{z-1} e^{-x} dx && (4.10)\\
&= z\int_0^\infty x^{z-1} e^{-x} dx && (4.11)\\
&= z\Gamma(z) && (4.12)
\end{aligned}$$

□

The normalisation condition requires that:

$$\int_{\boldsymbol{\theta}} f(\theta|s,\boldsymbol{t}) d\boldsymbol{\theta} = 1 \tag{4.13}$$

where the integration is carried out over the $(k-1)$-dimensional probability simplex. Substituting the density of the Dirichlet distribution into the normalisation condition, we obtain:

$$\frac{1}{B(s,\boldsymbol{t})} \int_{\boldsymbol{\theta}} \prod_{i=1}^k \theta_c^{st_i - 1} d\boldsymbol{\theta} = 1 \tag{4.14}$$

Then solving for $B(s,\boldsymbol{t})$ we get,

$$B(s,\boldsymbol{t}) = \int_{\boldsymbol{\theta}} \prod_{i=1}^k \theta_c^{st_i - 1} d\boldsymbol{\theta} \tag{4.15}$$

With the Beta function defined as such:

Definition 4.2.2 (Beta Function)**.**

$$B(s,\boldsymbol{t}) = \frac{\prod_{i=1}^k \Gamma(st_i)}{\Gamma\left(\sum_{i=1}^k st_i\right)}$$

[1] Hogg and Craig state this theorem without proof in [11] and so I have verified this with a simple integration by parts.

The posterior distribution then becomes:

$$p(\boldsymbol{\theta}|\boldsymbol{n}) \propto \prod_{c \in C} \theta_c^{n_i + st_i - 1} \tag{4.16}$$

Notice this is an unconventional parameterisation of the Dirichlet distribution. The direct interpretation of the conventional parameters, $\boldsymbol{\alpha}$, can sometimes be non-intuitive, especially when dealing with vague or weak prior information. To address this, Walley [27] has introduced a re-parameterisation using s and $\mathbf{t}$ with:

$$\boldsymbol{\alpha} = s\mathbf{t} \tag{4.17}$$

Here, s represents the total weight or strength of prior information. A larger value of s indicates stronger belief in the prior, while a smaller value indicates weaker or vaguer prior knowledge, more on this later. On the other hand, $\mathbf{t}$ captures the prior probability of success, lying between 0 and 1. This direct interpretation allows for a more intuitive understanding:

1. s answers the question: "How strong is our prior belief?"
2. t answers the question: "What is our prior probability of success?"

We refer to these as the hyper-parameters so as to not confuse them with the original $\boldsymbol{\theta}$ parameters of the multinomial. We will look at methods in setting these hyper-parameters in a later section. Now implementing this approach onto the likelihood obtained previously:

$$f(\mathbf{n}|\boldsymbol{\theta}) \propto \prod_{c \in C} \left[\theta_c^{n(c)} \prod_{i=1}^{k} \prod_{x_i \in X_i} \theta_{x_i|c}^{n(x_i,c)} \right] \tag{4.18}$$

Then the conjugate prior densities to the likelihood is given as the following product of Dirichlet distributions.

$$f(\boldsymbol{\theta}|s,\mathbf{t}) \propto \prod_{c \in C} \left[\theta_c^{st(c)-1} \prod_{i=1}^{k} \prod_{x_i \in X_i} \theta_{x_i|c}^{st(x_i,c)-1} \right] \tag{4.19}$$

Thus the posterior density is also a product of independent Dirichlet densities:

$$f(\boldsymbol{\theta}|\mathbf{n},s,\mathbf{t}) \propto \prod_{c \in C} \left[\theta_c^{n(c)+st(c)-1} \prod_{i=1}^{k} \prod_{x_i \in X_i} \theta_{x_i|c}^{n(x_i,c)+st(x_i,c)-1} \right] \tag{4.20}$$

4.3 Posterior Expectations

Now given this set of posteriors, we can compute the posterior expectations. As with the likelihood derivation, these expectations are often quoted in the literature specifically by Zaffalon [29] but a derivation is not present. I have provided my own here using the integral definition of expectation.

$$\mathbb{E}[\theta_c|\mathbf{n}] = \int_{\boldsymbol{\theta}} \theta_c f(\theta|s,\boldsymbol{t}) d\boldsymbol{\theta} \tag{4.21}$$

$$= \int_{\boldsymbol{\theta}} \theta_c \left(\frac{1}{B(\boldsymbol{n}+s\boldsymbol{t})} \prod_{\tilde{c}\in C} \theta_i^{n(\tilde{c})+st(\tilde{c})-1} \right) d\boldsymbol{\theta}$$

$$= \frac{1}{B(\boldsymbol{n}+s\boldsymbol{t})} \int_{\boldsymbol{\theta}} \theta_c \prod_{\tilde{c}\in C} \theta_i^{n(\tilde{c})+st(\tilde{c})-1} d\boldsymbol{\theta}$$

$$= \frac{1}{B(\boldsymbol{n}+s\boldsymbol{t})} \int_{\boldsymbol{\theta}} \theta_c \theta_c^{n(c)+st(c)-1} \prod_{\tilde{c}\in C|\tilde{c}\neq c} \theta_i^{n(\tilde{c})+st(\tilde{c})-1} d\boldsymbol{\theta}$$

$$= \frac{1}{B(\boldsymbol{n}+s\boldsymbol{t})} \int_{\boldsymbol{\theta}} \theta_c^{n(c)+st(c)} \prod_{\tilde{c}\in C|\tilde{c}\neq c} \theta_i^{n(\tilde{c})+st(\tilde{c})-1} d\boldsymbol{\theta} \tag{4.22}$$

This is simply an integral of a Dirichlet Distribution with the $n(c) + st(c)$ parameter increased by one, which we know has to integrate to one after normalisation under the Beta function with these parameters and so:

$$\mathbb{E}[\theta_c|\mathbf{n}] = \frac{B\left(n_1 + st_1, ..., n(c) + st(c) + 1, n_k + st_k\right)}{B(\boldsymbol{n}+s\boldsymbol{t})} \tag{4.23}$$

Now by rewriting the Beta function as Gamma functions we can take advantage of theorem 4.2.1. For the sake of simplicity let us define $\mu(x) := n(x) + st(x)$ then:

$$\mathbb{E}[\theta_c|\mathbf{n}] = \frac{\Gamma\Big(\mu(c)+1\Big) \prod_{\tilde{c}\in C|\tilde{c}\neq c} \Gamma\Big(\mu(\tilde{c})\Big)}{\Gamma\Big(\sum_{\tilde{c}\in C} \mu(\tilde{c}) + 1\Big)} \cdot \frac{\Gamma\Big(\sum_{\tilde{c}\in C} \mu(\tilde{c})\Big)}{\prod_{\tilde{c}\in C} \Gamma\Big(\mu(\tilde{c})\Big)\cdot} \tag{4.24}$$

$$= \frac{\mu(c) \prod_{\tilde{c}\in C} \Gamma\Big(\mu(\tilde{c})\Big)}{\sum_{\tilde{c}\in C} \Big(\mu(\tilde{c})\Big) \Gamma\Big(\sum_{\tilde{c}\in C} \mu(\tilde{c})\Big)} \cdot \frac{\Gamma\Big(\sum_{\tilde{c}\in C} \mu(\tilde{c})\Big)}{\prod_{\tilde{c}\in C} \Gamma\Big(\mu(\tilde{c})\Big)} \tag{4.25}$$

$$= \frac{\mu(c)}{\sum_{\tilde{c}\in C} \Big(\mu(\tilde{c})\Big)} \tag{4.26}$$

Now plugging back in $n(x) + st(x) = \mu(x)$ and utilising the knowledge that $\sum_{c\in C} n(c) = N$ and $\sum_{c\in C} t(c) = 1$ we obtain the following closed-form expression for the expectation of θ_c

$$\mathbb{E}[\theta_c|\mathbf{n}] = \frac{n(c) + st(c)}{\sum_{\tilde{c}\in C} n(\tilde{c}) + s \sum_{\tilde{c}\in C} t(\tilde{c})}$$

$$= \frac{n(c) + st(c)}{N + s} \tag{4.27}$$

The derivation is analagous for $\theta_{x_i|c}$ however this time we take advantage of the fact that $\sum_{x_i\in X_i} n(x_i, c) = n(c)$ and $\sum_{c\in C} t(x_i, c) = t(c)$ and end up with the closed-form expression:

$$\mathbb{E}[\theta_{x_i|c}|\mathbf{n}] = \frac{n(x_i, c) + st(x_i, c)}{\sum_{\tilde{x}_i\in X_i} n(\tilde{x}_i, c) + s \sum_{\tilde{x}_i\in X} t(\tilde{x}_i, c)}$$

$$= \frac{n(x_i, c) + st(x_i, c)}{n(c) + st(c)} \tag{4.28}$$

4.4 Imprecise Dirichlet Model

Up until now we have been following classical probability theory where we are dealing with known distributions producing real-valued point probabilities. Here we will alternatively consider probabilities as intervals. The notion of probabilities beyond real numbers is not new; Keynes [12] in his 1921 Treatise of Probability gives a philosophical interpretation of this notion. Using interval representations for probabilistic measures often proves more effective than point values, particularly when derived from extensive data. The approach acknowledges the inherent uncertainties or imprecisions in probabilistic outcomes, suggesting that our knowledge extends only to a parameter's probable range, not its exact value. We now turn to the credal-set [15, p. 38 - 40] as a way of representing probabilistic uncertainty.

Definition 4.4.1. A **credal set** ψ is a set of probability distributions.

The credal set is comprehensive as it produces inference that of it's convex hull. Furthermore, it has been shown by Dalal and Hall [6] that every prior distribution can be expressed as a finite combination of Dirichlet distributions meaning the credal set itself can be used to represent the prior.

In particular the Imprecise Dirichlet Model, introduced by Walley [27], is proposed as an alternative source of inference to the Bayesian approach towards multinomial data. We aim to contemplate a credal-set of prior distributions, culminating in a similar set of posteriors. This offers a more robust and conservative framework, especially when there is uncertainty about the prior distribution or when data is sparse. The IDM's mathematical formulation leads to interval-valued summaries, encapsulating a wider potential outcome range than mere point estimates.

Formally we will define the Imprecise Dirichlet Model as the set of all $Dirichlet(s, \mathbf{t})$ distributions under which we fix s as a pre-specified constant independent of the sample space and we allow the hyper-parameter $\mathbf{t}$ to vary with respect to the following constraints proposed by Zaffalon [29]:

- $t(x_i, c) > 0$
- $\sum_c t(c) = 1$
- $\sum_{x_i \in X_i} t(x_i, c) = t(c)$

We now seek to optimise the expectations eqs. (4.27) and (4.28) over these constraints in order to obtain posterior predictive intervals: For the purposes of this optimisation we are simply looking at the minimum and maximum values under the $\mathbf{t}$ parameter. Since the extrema will occur at a static value for $t(x_i, c)$ we no longer need to take the constraint (4.4) that $\sum_i t(x_i, c) = t(c)$ since we could theoretically take $t(x_i, c)$ arbitrarily close to $t(c)$ or arbitrarily close to 0 and still have this constraint met. Similarly for an instance of $t(c)$ we could have $t(c)$ arbitrarily close to either 0 or 1 and still have the constraint (4.4) $\sum_c t(c) = 1$ met and so it is enough to take the constraints for the optimisation as [30]:

- $0 < t(x_i, c) < t(c)$
- $0 < t(c) < 1$

It is then simple to see eq. (4.27) falls over the following interval, since the denominator is independent of $\mathbf{t}$ and so it is enough to take $t(c) = 0$ for the minimum and $t(c) = 1$ for the maximum:

$$[\underline{p(c)}, \overline{p(c)}] = \left[\frac{n(c)}{N+s}, \frac{n(c)+s}{N+s}\right] \tag{4.29}$$

Optimising eq. (4.28) is slightly more complicated and while the literature often presents this previous interval, an obvious approach to the optimisation of eq. (4.27) is not currently presented, I have provided such a solution. Here we will implement our own approach by optimising sequentially. We

will start by taking $t(c)$ constant and optimising over $t(x_i, c)$. We will then optimise over $t(c)$. Namely for $t(c)$ constant $\mathbb{E}[\theta_{x_i|c}|\mathbf{n}]$ is monotonically increasing in $t(x_i, c)$ and so the minimum and maximum of the expectation are achieved at the lower and upper bounds on $t(x_i, c)$, 0 and $t(c)$ respectively.

$$\min_{t(x_i,c)} \mathbb{E}[\theta_{x_i|c}|\mathbf{n}] = \frac{n(x_i, c)}{n(c) + st(c)} \tag{4.30}$$

$$\max_{t(x_i,c)} \mathbb{E}[\theta_{x_i|c}|\mathbf{n}] = \frac{n(x_i, c) + st(c)}{n(c) + st(c)} \tag{4.31}$$

Clearly eq. (4.30) is decreasing and is minimised with $t(c) = 1$. However eq. (4.31) requires a little more work. Differentiating with respect to $t(c)$ as follows we see that this is an increasing function in $t(c)$:

$$\frac{\partial}{\partial t(c)} \left(\max_{t(x_i,c)} \mathbb{E}[\theta_{x_i|c}|\mathbf{n}] \right) = \frac{-s\left(n(x_i, c) + st(c)\right)}{(n(c) + st(c))^2} + \frac{s}{n(c) + st(c)} \tag{4.32}$$

$$= s\frac{n(c) - n(x_i, c)}{(n(c) + st(c))^2} > 0 \quad \forall \quad n(c) > n(x_i, c) \tag{4.33}$$

As such $\max_{t(x_i,c)} \mathbb{E}[\theta_{x_i|c}|\mathbf{n}]$ is increasing in $t(c)$ and so the maximum is achieved at $t(c) = 1$. So we have shown the minimum to be achieved at $t(c) = 1, t(x_i, c) = 0$ and the maximum at $t(c) = 1, t(x_i, c) = 1$ obtaining

$$\underline{\mathbb{E}}[\theta_{x_i|c}|\mathbf{n}] = \frac{n(x_i, c)}{n(c) + s} \tag{4.34}$$

$$\overline{\mathbb{E}}[\theta_{x_i|c}|\mathbf{n}] = \frac{n(x_i, c) + s}{n(c) + s} \tag{4.35}$$

In other words the posterior predictive probability for the posterior expectation falls on the following interval:

$$[\underline{p}(x_i|c), \overline{p}(x_i|c)] = \left[\frac{n(x_i, c)}{n(c) + s}, \frac{n(x_i, c) + s}{n(c) + s}\right] \tag{4.36}$$

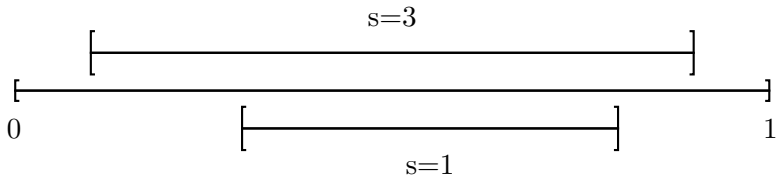
4.5 Choosing the hyper-parameter s

It now falls on us to determine a suitable value for the hyper-parameter s. As mentioned previously s can be thought of as representing the weighting of the effect that the prior has on the posterior. In particular, in the context of the IDM s is the number of observations needed to reduced the size of the imprecision of the interval, $\overline{p}(c|x_i) - \underline{p}(c|x_i)$, by a half. In other words it determines how quickly the posterior predictive intervals converge as the size of our dataset increases. Opting for a smaller value of s will accelerate the convergence making for more decisive decision-making, however this comes at the risk of over-fitting to more recent data, possibly at the expense of extensibility. Alternatively a higher value of s results in a slower more conservative inference.

Definition 4.5.1 (Representation Invariance Principle (RIP))**.** The posterior upper and lower probabilities assigned to an observable event ω should not depend on the sample space Ω in which ω and the previous events have been sampled.

This RIP is proposed by Walley [27, p. 5] as a key characteristic of the IDM. The reasoning behind this is that the choice of the sample space Ω is somewhat arbitrary and so inference should not depend on this, unlike standard Bayesian inference where this is often violated. It is in coherence with this principle that we require s to not depend on the size of the sample space.

Notice Dirichlet models with different values of s are always consistent with each other. Namely, the intervals created by smaller values of s will be contained within the larger interval the same cannot be said in standard Bayesian inference.



As we can see taking higher values for s will encompass a wider range of priors in the credal set. Some common priors we may want to consider when choosing value for s are:

- Haldane's Improper Prior:[10]: This is the single-product Dirichlet density with $s = 0$ which is the limit of IDM as $s \to 0$. This is analogous to the maximum likelihood estimates obtained before as it results in no smoothing term and so results in the same parameter estimates as before. $p(c) = \frac{n(c)}{N}$ and $p(x_i|c) = \frac{n(x_i, c)}{n(c)}$.

- Uniform Prior [14]: The concept of uniform priors, especially through the Dirichlet distribution, is traditionally employed to reflect complete non-informativeness. This results in a distribution where all $|C|$ dimensional simplex are equally likely. In the traditional sense we would like to take the hyper-parameters $\alpha_i = 1 \quad \forall i$ [14]. A direct application of this onto Zaffalon [29]'s model is not obvious. For a fixed s we can achieve uniformity on the class taking $s = |C|$ and $t(c) = 1/|C|$, or we can achieve uniformity of the attributes taking $s = |C||X_i|$ and $t(x_i, c) = \frac{1}{|C||X_i|}$, ensuring $\sum_i t(x_i, c) = t(c) = 1/|C|$. However it is not possible to achieve both for a fixed s. In the case of our binary classification problem we would require $s = 2$ over the class and $s = 4$ over the attributes.

The literature, notably Zaffalon [29], has not fully addressed adapting uniform priors to scenarios where equal likelihood across multiple dimensions is required. This oversight highlights a potential research avenue: developing a model with an adjustable s parameter, allowing for uniformity across various dimensions. Such advancements would significantly enhance the applicability of uniform priors in the NCC.

Chapter 5

Naive Credal Classification

5.1 Posterior Probability

We infer the Naive Credal Classifier (NCC) [28] framework by incorporating Walley's Imprecise Dirichlet Model (IDM) [27] with the logic of the Naive Bayes Classifier. We first obtain a closed-form expression for the posterior probability $p(c|\mathbf{x})$ under the Dirichlet prior by combining the parameterised form of the posterior probability under Naive Bayes eq. (3.7) and the posterior predictive probabilities eqs. (4.27) and (4.28) as such:

$$\begin{aligned} p(c|\mathbf{x}) &= p(c)\prod_{i=1}^{k} p(x_i|c) \\ &= \frac{n(c)+st(c)}{N+s}\prod_{i=1}^{k}\frac{n(x_i,c)+st(x_i,c)}{n(c)+st(c)} \\ &= \frac{1}{(N+s)(n(c)+st(c))^{k-1}}\prod_{i=1}^{k}(n(x_i,c)+st(x_i,c)) \end{aligned} \tag{5.1}$$

We then infer an interval for the posterior from the IDM by varying the parameter $\mathbf{t}$ to obtain the credal set of Dirchlet priors.

5.2 Interval Dominance

Now there are multiple ways in which we can obtain an estimate for class preference. One of the simplest none the less effective approaches to this is interval dominance. Here we propose two differing levels of interval dominance which we will informally refer to as: approximate interval dominance and exact interval dominance. The exact intervals obtained here are a novel contribution of this report. We will define interval dominance as such:

Definition 5.2.1. [13] Let $c_1, c_2 \in C$ be two states of the class variable. Consider the distribution $\delta \in \psi_C^E$ where E represents what is already known and ψ_C^E is a non-empty credal set of distributions. Then $p(c_1|E)$ and $p(c_2|E)$ under ψ_C^E can be represented by the intervals $I_1 = [\underline{p}(c_1|E), \overline{p}(c_1|E)]$ and $I_2 = [\underline{p}(c_2|E), \overline{p}(c_2|E)]$ respectively. Interval I_1 is said to dominate I_2 if $\underline{p}(c_1|E) > \overline{p}(c_2|E)$. In this case we say c_1 is interval dominant to c_2.

The approximate approach to implementing this is the obtain bounds on these intervals by taking the product of the lower bounds on eq. (4.36) and eq. (4.29) to obtain the lower bound on eq. (5.1) and similarly for the upper bounds. I am referring to this as an approximate approach as the lower bounds

for eq. (4.36) are achieved for static values on $t(x_i, c)$ and so when taking a product over them, we are violating the constraint (4.4). Furthermore the upper bound on eq. (4.29) is achieved at $t(c) = 1$ and the lower bound for eq. (4.29) is achieved at $t(c) = 0$ and so while this lower probability will be a bound on the posterior it will be far from optimal. We will improve on this in our exact approach. For now let us plug in the values to obtain approximate intervals.

$$\begin{aligned}\underline{p}(c|\mathbf{x}) &= \frac{n(c)}{N+s}\prod_{i=1}^{k}\frac{n(x_i,c)}{n(c)+s} \\ &= \frac{n(c)}{(N+s)(n(c)+s)^k}\prod_{i=1}^{k} n(x_i,c)\end{aligned} \tag{5.2}$$

$$\begin{aligned}\overline{p}(c|\mathbf{x}) &= \frac{n(c)+s}{N+s}\prod_{i=1}^{k}\frac{n(x_i,c)+s}{n(c)+s} \\ &= \frac{n(c)+s}{(N+s)(n(c)+s)^k}\prod_{i=1}^{k}(n(x_i,c)+s) \\ &= \frac{1}{(N+s)(n(c)+s)^{k-1}}\prod_{i=1}^{k}(n(x_i,c)+s)\end{aligned} \tag{5.3}$$

Obtaining

$$I = \left[\frac{n(c)}{(N+s)(n(c)+s)^k}\prod_{i=1}^{k} n(x_i,c), \frac{1}{(N+s)(n(c)+s)^{k-1}}\prod_{i=1}^{k}(n(x_i,c)+s)\right] \tag{5.4}$$

This interval will be a lot wider than necessary and so while it will be more accurate, it will be overly cautious resulting in an indeterminate result more often than necessary. Throughout my research there were no references to a closed-form interval on which we can determine exact interval dominance and so I have derived a novel bound on this posterior probability that we will see has a surprising degree of efficacy when implemented in our experimentation. Let us optimise the posterior eq. (5.1) directly. We are optimising the following:

$$p(c|\mathbf{x}) \propto \frac{1}{(n(c)+st(c))^{k-1}}\prod_{i=1}^{k}(n(x_i,c)+st(x_i,c)) \tag{5.5}$$

- $0 < t(x_i, c) < t(c)$
- $0 < t(c) < 1$

Firstly we notice that the minimum is clearly achieved for $t(x_i, c) = 0$ and $t(c) = 1$ since this minimises and maximises the numerator and the denominator respectively. Obtaining

$$\underline{p}(c|\mathbf{x}) = \frac{1}{(N+s)(n(c)+s)^{k-1}}\prod_{i=1}^{k} n(x_i,c) \tag{5.6}$$

Furthermore the maximum will be achieved for $t(x_i, c) = t(c)$ and so we can reduce this k dimensional optimsation problem to 1 dimension. Namely applying this $st(c) = \zeta$

$$\tilde{p}(c|\mathbf{x}) \propto \frac{1}{(n(c)+\zeta)^{k-1}}\prod_{i=1}^{k}(n(x_i,c)+\zeta) \tag{5.7}$$

In an effort to simplify the optimisation we look to optimise $\log p(c|\mathbf{x})$ as taking the logarithm of a function preserves its extrema.

$$\log \tilde{p}(c|\mathbf{x}) \propto -(k-1)\log(n(c)+\zeta) + \sum_{i=1}^{k} \log(n(x_i,c)+\zeta) \tag{5.8}$$

Now we will differentiate with respect to ζ in an effort to show monotonicity:

$$\frac{\partial \log \tilde{p}(c|\mathbf{x})}{\partial \zeta} \propto \frac{-(k-1)}{n(c)+\zeta} + \sum_{i=1}^{k} \frac{1}{n(x_i,c)+\zeta} \tag{5.9}$$

Now using the property that $n(x_i,c) \leq n(c)$:

$$\implies \frac{1}{n(x_i,c)+\zeta} \geq \frac{1}{n(c)+\zeta} \tag{5.10}$$

$$\implies \sum_{i=1}^{k} \frac{1}{n(x_i,c)+\zeta} \geq \sum_{i=1}^{k} \frac{1}{n(c)+\zeta} = \frac{k}{n(c)+\zeta} > \frac{k-1}{n(c)+\zeta} \tag{5.11}$$

$$\implies \sum_{i=1}^{k} \frac{1}{n(x_i,c)+\zeta} - \frac{k-1}{n(c)+\zeta} > 0 \tag{5.12}$$

$$\implies \frac{\partial \log \tilde{p}(c|\mathbf{x})}{\partial \zeta} > 0 \tag{5.13}$$

So $\log \tilde{p}(c|\mathbf{x})$ is monotonically increasing on all ζ and so in particular $\tilde{p}(c|\mathbf{x})$ is increasing on $\zeta \in [0,s]$. Therefore $\overline{p}(c|\mathbf{x}) = \max_\zeta \tilde{p}(c|\mathbf{x})$ must be achieved at $\zeta = s$, and so

$$\overline{p}(c|\mathbf{x}) = \frac{1}{(N+s)(n(c)+s)^{k-1}} \prod_{i=1}^{k} (n(x_i,c)+s) \tag{5.14}$$

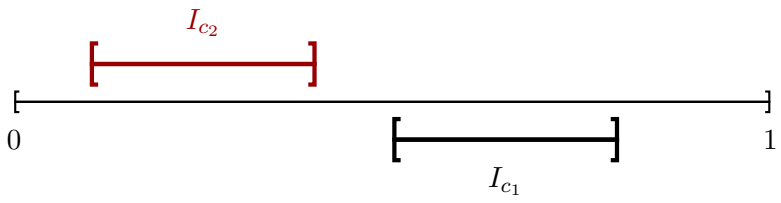
□

This upper bound on the probability is actually in agreement with our approximate upper bound but we have now formally proven this to be optimal. We now have a closed-form expression for the interval I_c on which the probabilities must fall:

$$I_c = \left[\frac{1}{(N+s)(n(c)+s)^{k-1}} \prod_{i=1}^{k} n(x_i,c), \frac{1}{(N+s)(n(c)+s)^{k-1}} \prod_{i=1}^{k} (n(x_i,c)+s) \right] \tag{5.15}$$

Once we have obtained these optimal intervals on which the posterior probaility will fall we can make our decision as described in definition 5.2.1. In other words in order for c_1 to be interval dominant to c_2 we require:

$$\frac{1}{(n(c_2)+s)^{k-1}} \prod_{i=1}^{k} (n(x_i,c_2)+s) < \frac{1}{(n(c_1)+s)^{k-1}} \prod_{i=1}^{k} n(x_i,c_1) \tag{5.16}$$



5.3 Credal Dominance

It is worth noting that the information potentially available through credal sets exceeds that offered by the intervals above. Specifically, the credal set associated with $p(C|E)$, denoted ψ_C^E, provides a richer knowledge base compared to individual intervals. This is because credal sets are capable of encapsulating additional constraints that are not apparent when adopting a purely interval-based perspective. Consequently, it prompts the question as to whether a more effective dominance criteria can be utilised that takes advantage of all the information encoded in the credal set. To illustrate let us consider the following example that I have adapted from Walley [27].

Example 5.3.1. *Consider the outcome of a football match X. Let $\Omega_X = W, D, L$ be a set of possible outcomes, where W wins, D draws and L loses. We are given the following information:*

- *W^c is at least as likely as W,*
- *W is at least as likely as D,*
- *D is at least as likely as L.*

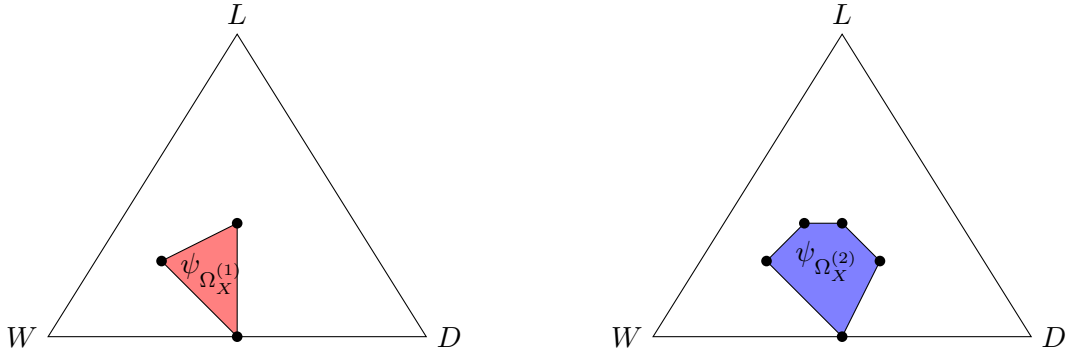
Then it follows naturally from classical probability theory that these pieces of information allow us to consider the following set distribution on Ω_X:

$$\psi_{\Omega_X}^{(1)} = \left\{ p(W) \leq \frac{1}{2}, \quad p(W) \geq p(D), \quad p(D) \geq p(L) \right\} \tag{5.17}$$

If we instead maximise and minimise the probabilities under the constraints to obtain interval on which the probabilities fall we get the following set:

$$\psi_{\Omega_X}^{(2)} = \left\{ \frac{1}{3} \leq p(W) \leq \frac{1}{2}, \quad \frac{1}{4} \leq p(L) \leq \frac{1}{2}, \quad 0 \leq p(D) \leq \frac{1}{3} \right\} \tag{5.18}$$

So we have two different representations for the same information using precise and imprecise probabilities. The credal sets $\psi_{\Omega_X}^{(1)}$ and $\psi_{\Omega_X}^{(2)}$ are shown below, where we use a simplex bi-dimensional representation of a credal set on $\mathbb{R}^3$, where one point in a triangle of height one represents a probability distribution $(p1; p2; p3) \in \mathbb{R}^3$ where p_i is the distance of this point to the side opposite the corresponding vertex.



$\psi_{\Omega_X^{(1)}}$ is the set of convex combinations of the distributions

$$\psi_{\Omega_X}^{(1)} = Conv\left\{ \left(\frac{1}{2}, \frac{1}{2}, 0\right); \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right); \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right) \right\} \tag{5.19}$$

$\psi_{\Omega_X^{(2)}}$ is the set of convex combinations of the distributions

$$\psi_{\Omega_X}^{(2)} = Conv\left\{ \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right); \left(\frac{5}{12}, \frac{1}{4}, \frac{1}{3}\right); \left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right); \left(\frac{1}{2}, \frac{1}{2}, 0\right); \left(\frac{1}{3}, \frac{1}{2}, \frac{1}{6}\right) \right\} \tag{5.20}$$

Here we can see in $\psi_{\Omega_X^{(1)}}$ that $p(W) \geq p(D)$ for every distribution, yet we get $\underline{p}(W) = \frac{1}{3}$ and $\overline{p}(D) = \frac{1}{2}$, therefore $\overline{p}(D) > \underline{p}(W)$ and we do not have interval-dominance; yet W is clearly probabilistically superior to D.

It is in this vein we therefore wonder if there is a criteria for class dominance that can better exploit the full knowledge of the credal set.

Definition 5.3.1. [28] Let $c_1, c_2 \in C$ be two states of the class variable. Consider the distribution $\delta \in \psi_C^E$ where E represents what is already known and ψ_C^E is a non-empty credal set of distributions. Then the class c_1 is said to be **credal-dominant** to c_2 if for every distribution $\delta \in \psi_C^E$, then $p(c_1|E) \geq p(c_2|E)$ under δ.

Remark 5.3.1. We can trivially induce from these definitions that

$$\text{interval-dominance} \Rightarrow \text{credal-dominance}$$

However as shown through counterexample in the football example

$$\text{credal-dominance} \not\Rightarrow \text{interval-dominance}$$

Now the following formulation of credal dominance is following that proposed by Zaffalon [29]. We can see clearly that this defining inequality of credal-dominance is equivalent to:

$$p(c_1|\mathbf{x}) \geq p(c_2|\mathbf{x}) \iff \frac{p(c_1|\mathbf{x})}{p(c_2|\mathbf{x})} \geq 1 \tag{5.21}$$

In this way checking for credal-dominance can be rewritten as the following optimisation problem.

$$\inf_{p(C|\mathbf{X}) \in \psi_C^{\mathbf{X}}} \frac{p(c_1|\mathbf{x})}{p(c_2|\mathbf{x})} \geq 1 \tag{5.22}$$

$$\sum_{c \in C} t(c) = 1 \tag{5.23}$$

$$0 < t(x_i, c) < t(c) \tag{5.24}$$

We can substitute eq. (5.1) into this infimum to get

$$\inf_{p(C|\mathbf{X}) \in \psi_C^{\mathbf{X}}} \frac{p(c_1|\mathbf{x})}{p(c_2|\mathbf{x})} = \inf \left\{ \frac{(n(c_2) + st(c_2))^{k-1}}{(n(c_1) + st(c_1))^{k-1}} \prod_{i=1}^{k} \frac{n(x_i, c_1) + st(x_i, c_1)}{n(x_i, c_2) + st(x_i, c_2)} \right\} \tag{5.25}$$

Based on our constraints we can minimise the product with $t(x_i, c_1) = 0$ and $t(x_i, c_2) = t(c_2)$. Furthermore assume $t(c_1) + t(c_2) < 1$ then we can fix $t(c_1)$ and increase $t(c_2)$ up to $1 - t(c_1)$ thus decreasing the infimum. So by necessity $t(c_1) + t(c_2) = 1$ and the above optimisation can be once again be rewritten as such:

$$\inf \left\{ \left[\frac{n(c_2) + st(c_2)}{n(c_1) + st(c_1)} \right]^{k-1} \prod_{i=1}^{k} \frac{n(x_i, c_1)}{n(x_i, c_2) + st(c_2)} \right\} \geq 1 \tag{5.26}$$

$$t(c_1) + t(c_2) = 1 \tag{5.27}$$

$$t(c_1), t(c_2) > 0 \tag{5.28}$$

Now in an effort to find this infimum let us first prove that there exists one such point of local infimum on the interval open (0,s). We do this by showing convexity of the quotient. Let us substitute $\zeta = st(c_2)$ for simplicity and then plug eq. (5.27) into eq. (5.22) and call it $f(\zeta)$. Since f is strictly positive by construction, we can then take its logarithm without loosing information with respect to the optimsation:

$$\log f(\zeta) = \log \frac{p(c_1|\mathbf{x})}{p(c_2|\mathbf{x})} = (k-1)\log(n(c_2)+\zeta) - (k-1)\log(n(c_1)+s-\zeta)) + \sum_{i=1}^{k} \log(n(x_i,c_1)) - \log(n(x_i,c_2)+\zeta) \tag{5.29}$$

Differentiating this using the chain rule and $\frac{dlog(x)}{dx} = \frac{1}{x}$ we get:

$$\frac{d\log f(\zeta)}{d\zeta} = \frac{k-1}{n(c_2)+\zeta} + \frac{k-1}{n(c_1)+s-\zeta} - \sum_{i=1}^{k} \frac{1}{n(x_i,c_2)+\zeta} \tag{5.30}$$

Differentiating once more:

$$\frac{d^2\log f(\zeta)}{d\zeta^2} = -\frac{k-1}{(n(c_2)+\zeta)^2} + \frac{k-1}{(n(c_1)+s-\zeta)^2} + \sum_{i=1}^{k} \frac{1}{(n(x_i,c_2)+\zeta)^2} \tag{5.31}$$

Now once again using the property that $n(x_i,c_2) < n(c_2)$:

$$\frac{k-1}{(n(c_2)+\zeta)^2} < \frac{k}{(n(c_2)+\zeta)^2} = \sum_{i=1}^{k} \frac{1}{(n(c_2)+\zeta)^2} \leq \sum_{i=1}^{k} \frac{1}{(n(x_i,c_2)+\zeta)^2} \tag{5.32}$$

Which implies positivity of the following term in the second order derivative:

$$\sum_{i=1}^{k} \frac{1}{(n(x_i,c_2)+\zeta)^2} - \frac{k-1}{(n(c_2)+\zeta)^2} > 0 \tag{5.33}$$

And so we have:

$$\frac{d^2\log f(\zeta)}{d\zeta^2} > 0 \iff \log f(\zeta) \quad \text{convex.}$$

Then it follows through basic properties of the exponential function that $f(\zeta)$ is also convex and so on $(0,s)$ must have a single infimum. □

Now we have shown uniqueness of such an infimum on $(0,s)$ it falls on us to construct an algorithm to check for credal-dominance. First notice that if $n(x_i,c_2) = 0$

$$\lim_{\zeta\to 0} f(\zeta) = \frac{(n(c_2))^{k-1}}{(n(c_1)+s)^{k-1}} \prod_{i=1}^{k} \lim_{\zeta\to 0} \frac{n(x_i,c_1)}{\zeta} = +\infty \tag{5.34}$$

$$\lim_{\zeta\to 0} \frac{d\log f(\zeta)}{d\zeta} = \frac{k-1}{n(c_2)} + \frac{k-1}{n(c_1)+s} - \sum_{i=1}^{k} \lim_{\zeta\to 0} \frac{1}{\zeta} = -\infty \tag{5.35}$$

So the algorithm to determine the infimum of f and so determine whether or there is credal domaince is as below. The first two if statements are taking into account the possibility of an unbounded $f(\zeta)$. If all other conditions fails we must fall to a numerical method to approximate this root. The Newton

Raphson [4] method is a suitable choice as we have already calculated the first and second order derivatives of $\log f$ and we can ensure convergence by implementing bracketing [4].

Algorithm 1: Algorithm to determine infimum of $f(\zeta)$ on $\zeta \in (0, s)$

Result: $\inf f(\zeta)$.
if $\exists\ i\ s.t.\ n(x_i, c_1) = 0$ **then**
  $\inf f(\zeta) = 0$. Stop
end
if $\exists\ i\ s.t.\ n(x_i, c_2) = 0$ **then**
  if $n(x_i; c_2') = 0$ **then**
    $(\log f(0))' = -\infty$.
  else
    Compute $(\log f(0))'$.
  end
end
Compute $(\log f(s))'$.
if $(\log f(0))' \geq 0$ **then**
  $\inf f(\zeta) = f(0)$. Stop
else
  if $(\log f(s))' \leq 0$ **then**
    $\inf f(\zeta) = f(s)$. Stop
  else
    Approximate the minimum numerically. Stop
  end
end

5.4 The Feature Problems

While this may seem an intuitive approach it suffers a substantial setback. An important consideration is that of data incompleteness. As we have seen the credal set we are using to infer the NCC here excludes the extremes of the set corresponding to the improper densities. However in the above algorithm we have taken $t(x_i|c_1) \to 0$ and $t(x_i, c_2) \to t(c_2)$. This leaves us with the so called "feature problem". Namely if $n(x_i, c_1) = 0$ then the infimum of eq. (5.37) is always zero and so credal dominance is impossible. Let us demonstrate this through an example.

Example: The Feature Problem

Take the toy email

```
From: scammer@sender.com
To: you@example.com
Subject: Exclusive Offer!
Content:
  Collect winningz now !!!
```

If the training data's vocabulary were to not include the word "winningz" then $n(\text{"winningz"}, \text{Spam})$ and subsequently $n(x_i, c_1) = n(x_i, c_2) = 0$. Clearly this is a problem as this results in

$$\inf \left\{ \left[\frac{n(c_2) + st(c_2)}{n(c_1) + st(c_1)} \right]^{k-1} \prod_{i=1}^{k} \frac{n(x_i, c_1)}{n(x_i, c_2) + st(c_2)} \right\} = 0 \tag{5.36}$$

The infimum is zero and so the classifier is indeterminate. This is actually a tactic often employed by spammers to try and by-pass less sophisticated Bayesian filters. They will purposefully misspell or shorten words to avoid detection on such filters. We call this problem the feature problem and it is pervasive in lots of classification problems not just that of emails. To deal with this problem we will turn to the NCC$_\epsilon$ [5] which mitigates against this zero infimum by redefining the constraints on $\mathbf{t}$ in terms of $\epsilon > 0$ as follows:

- $\epsilon \leq t(x_i, c_1) \leq t(c_1)$
- $\epsilon \leq t(x_i, c_2) \leq t(c_2)$
- $t(c_1) + t(c_2) = 1$

This approach guarantees that the infimum is always non-zero since we now have $st(x_i, c_1) \to s\epsilon$ on the numerator.

$$\inf_{p(C|\mathbf{X}) \in \psi_C^{\mathbf{X}}} \frac{p(c_1|\mathbf{x})}{p(c_2|\mathbf{x})} = \inf \left\{ \frac{(n(c_2) + st(c_2))^{k-1}}{(n(c_1) + st(c_1))^{k-1}} \prod_{i=1}^{k} \frac{n(x_i, c_1) + s\epsilon}{n(x_i, c_2) + st(c_2)} \right\} \tag{5.37}$$

Naturally the question is raised as to a suitable choice for ϵ. While Corani and Benavoli [5] state that ϵ should fall on the interval $\epsilon \in (0, 0.5]$, this is an apparent oversight on their behalf. My findings highlight an amendment to the existing literature and propose a more accurate guideline for the selection of ϵ in Corani and Benavoli [5]'s NCC_ϵ. Namely if we take $\epsilon > 0.25$ then

$$t(c_1) = t(x_1, c_1) + t(x_2, c_1) \geq 2\epsilon > 0.5 \tag{5.38}$$

$$t(c_2) = t(x_1, c_2) + t(x_2, c_2) \geq 2\epsilon > 0.5 \tag{5.39}$$

$$\implies t(c_1) + t(c_2) > 1 \tag{5.40}$$

This notably violates the $t(c_1) + t(c_2) = 1$ condition in our model, as such we have shown that necessarily $\epsilon \in (0, 0.25]$ in order that we maintain coherence with Zaffalon [29].

We desingate the resultant classifier NCC$_\epsilon$ as named by Corani and Benavoli [5]. In particular from the above constraints we see $\epsilon \in (0, 0.25]$. This effectively mitigates the feature problem by enforcing $t(x_i, c_1) > \epsilon$, thereby eliminating zeros in the numerator of eq. (5.37). Moreover, the credal set here adheres to the Representation Invariance Principle (RIP) definition 4.5.1 exhibiting independence from the class count. To minimise disruptions to the credal set and address the feature problem effectively, Corani and Benavoli [5] recommend using modest ϵ values, typically between 0.01 and 0.1. We will explore this claim in more detail in the following chapter. Moreover, since our procedures for credal dominance relied on convexity arguments, these procedures remain unchanged. We simply replace the instances where we took $t(x_i, c) \to 0$ with $t(x_i, c) \to \epsilon$. A similar modification is made to our bounds we derived for interval dominance eq. (5.16). Our argument here relied on the monotonicity and so once again we simply minimise for $t(x_i, c) = \epsilon$ obtaining:

$$\frac{1}{(n(c_2) + s)^{k-1}} \prod_{i=1}^{k} (n(x_i, c_2) + s) < \frac{1}{(n(c_1) + s)^{k-1}} \prod_{i=1}^{k} (n(x_i, c_1) + s\epsilon) \tag{5.41}$$

It is worth noting that in our implementation of this bound k is often very large and so evaluating the inequality in it's current form is often infeasible due to under-flow error. To get around this we take the log, preserving the inequality and collecting similar terms we get the interval dominance condition eq. (5.16) equivalent to:

$$(k-1) \log \left(\frac{n(c_1) + s}{n(c_2) + s} \right) < \sum_{i=1}^{k} \log \left(\frac{n(x_i, c_1) + s\epsilon}{n(x_i, c_2) + s} \right) \tag{5.42}$$

This is the condition for c_1 being interval dominant over c_2 that we will use in our implementation.

Chapter 6

Experiments

6.1 Experimental Setup

In this chapter we implement an empirical evaluation of the Naive Credal Classifier on spam filtering. The experiments have been designed to assess the efficacy of the NCC_ϵ in comparison to the traditional NBC. In particular we will be evaluating the performance of the NCC_ϵ using both the credal dominance from Zaffalon [29] and interval dominance utilising the novel bounds derived previously. In order to do this we will be looking at two main metrics, determinacy and single accuracy. Where the determinacy is the percentage of times the classifier is decisive and makes a classification and the single accuracy is the percentage of determinate predictions in which the model correctly classifiers an email

The experiments comprise three: one varying ϵ; one varying s; and one comparing the performance of the NCC_ϵ using both forms of dominance criteria against the NBC. For the latter, as well as accuracy and determinacy we will also look at how accurate the NBC is in cases where the NCC_ϵ was indeterminate under credal dominance. The dataset was randomly divided using stratified sampling into a training and testing. Now in an effort to ensure the results from these experiments are representative we turn to the following definition:

Definition 6.1.1 (k-fold Cross Validation)**.** [20]: The data is partitioned into k segments or folds. We then perform k iterations of training and validation such that in each iteration a different fold is used for validation and the remaining $k-1$ folds are used for learning. Then for a metric μ we can estimate its mean $\overline{\mu}$ and standard deviation σ across the k folds:

$$\overline{\mu} = \frac{1}{k}\sum_{i=1}^{k}\mu_i \tag{6.1}$$

$$\sigma = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(\mu_i - \overline{\mu})^2} \tag{6.2}$$

where μ_i is the metric for the ith iteration of the validation

In each of the following experiments each of the results is obtained from by performing a k-fold cross validation of the data set setting $k = 5$. To do this we have used the `KFold` method from `scikit-learn` library [25]. For each of the experiments I have provided the random state or "seed" used in the k-fold segmentation: in order to allow for complete transparency and reproducibility of the results.

6.2 Experiment 1: Determining ϵ

Earlier we turned to NCC_ϵ as a solution to the feature problem, however we were vague on the choice of ϵ other than specifiying the interval $\epsilon \in (0, 0, 25]$. For our first experiment we are going to look at the effect the choice of epsilon (ϵ) has on both the single accuracy and the determinacy of the classifier under both credal and interval dominance. Here we are taking $s = 1$ and we are testing each ϵ on a k-fold cross validation ($k = 5$) with a random seed of 5. Below table 6.1 shows the trade off between determinacy and accuracy as we vary ϵ.

Table 6.1: Model Performance for Different Epsilon Values (seed = 5)

Epsilon (ϵ)	Single Accuracy (%)		Determinacy (%)	
	Credal	Interval	Credal	Interval
0.025	99.91 ± 0.12	99.96 ± 0.07	77.99 ± 1.48	64.13 ± 1.48
0.05	99.88 ± 0.15	99.90 ± 0.14	80.90 ± 0.91	68.94 ± 1.78
0.075	99.80 ± 0.15	99.90 ± 0.13	82.85 ± 0.73	73.30 ± 1.69
0.1	99.77 ± 0.17	99.87 ± 0.18	84.59 ± 0.75	76.61 ± 1.46
0.125	99.64 ± 0.26	99.70 ± 0.25	86.26 ± 0.46	79.51 ± 1.19
0.15	99.48 ± 0.38	99.65 ± 0.30	87.30 ± 0.51	81.71 ± 1.03
0.175	99.41 ± 0.45	99.54 ± 0.35	88.57 ± 0.31	83.66 ± 0.60
0.2	99.36 ± 0.44	99.53 ± 0.38	89.40 ± 0.44	86.18 ± 0.41
0.225	99.18 ± 0.33	99.46 ± 0.43	90.31 ± 0.56	87.92 ± 0.41
0.25	99.09 ± 0.34	99.23 ± 0.31	91.26 ± 0.46	89.61 ± 0.29

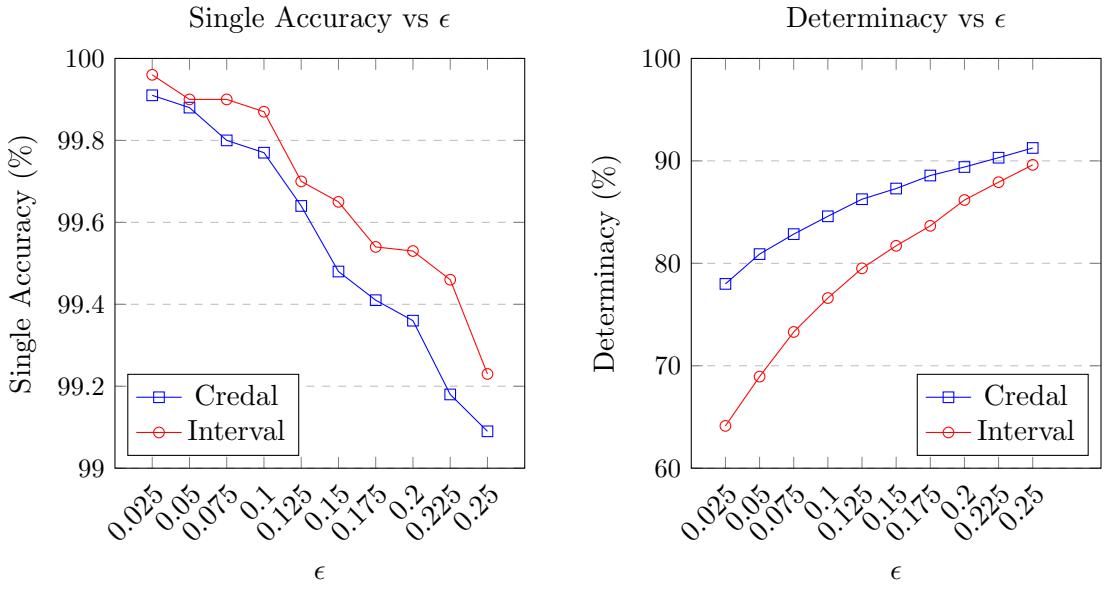


Figure 6.1: Performance metrics showing Accuracy and Determinacy for various ϵ values.

Under both dominance criteria, we see that raising the value of ϵ consistently enhances determinacy. This effect can be explained by considering that increasing ϵ decreases the interval over which t varies, thereby narrowing down the credal set and diminishing the likelihood that an instance's classification relies heavily on prior assumptions.

Our results are coherent with the fact that interval dominance implies credal dominance but not necessarily the contrary. In other words, in any instance where the NCC_ϵ is determinate under interval dominance, it must also be under credal dominance. Therefore, for a given value of ϵ, the interval determinacy should always be less than the credal determinacy. This is demonstrated through the results. However, increasing ϵ comes at the expense of single accuracy. We observe a gradual, nonetheless significant, decrease in single accuracy as we increase ϵ under both forms of dominance.

When comparing interval dominance with credal dominance, we note that the single accuracy is slightly worse under credal dominance, which is expected due to its reduced determinacy. In terms of a suitable choice for ϵ, this is dependent on the domain and specific requirements of the classification. For instance, in cancer screening, where accuracy is of the utmost importance, one may opt for a lower value of ϵ, trading determinacy for single accuracy. However, in the context of spam filtering, where single accuracy is not as critical, a higher value of ϵ might be preferable.

I would argue that having 40% versus 10% of my emails unfiltered is more significant than an extra 1 out of every 100 emails being incorrectly classified, as implied by the data. To this end, in the following experiment, we take $\epsilon = 0.25$ so as to maximise determinacy and to make the effect of varying s on the single accuracy more apparent.

6.3 Experiment 2: The s parameter

Previously we talked about the implications of varying s, here we are going to perform a similar experiment as with ϵ but this time varying the hyper-parameter s and seeing the effect this has on the single accuracy and deteminancy of the classification. Here we are taking $\epsilon = 0.25$ and we are testing each s on a k-fold cross validation ($k = 5$) with a random seed of 5. Below table 6.2 shows the trade off between determinacy and accuracy as we vary s.

Table 6.2: Model Performance for Different Epsilon Values (seed = 50)

	Single Accuracy (%)		Determinacy (%)	
s parameter	Credal	Interval	Credal	Interval
0.5	98.72 ± 0.48	99.11 ± 0.38	94.66 ± 0.60	91.31 ± 0.46
1.0	99.09 ± 0.34	99.70 ± 0.25	91.26 ± 0.46	79.51 ± 1.19
1.5	99.50 ± 0.31	99.88 ± 0.16	86.28 ± 0.67	58.50 ± 2.65
2.0	99.76 ± 0.18	99.93 ± 0.13	79.92 ± 0.96	36.30 ± 2.13
2.5	99.73 ± 0.21	100.00 ± 0.00	70.58 ± 1.73	19.70 ± 0.65
3.0	99.75 ± 0.24	100.00 ± 0.00	59.08 ± 2.71	10.31 ± 0.61
3.5	99.71 ± 0.28	100.00 ± 0.00	49.55 ± 2.40	5.41 ± 0.46
4.0	99.65 ± 0.35	100.00 ± 0.00	40.83 ± 1.49	2.60 ± 0.35
4.5	99.61 ± 0.36	100.00 ± 0.00	30.90 ± 1.34	1.41 ± 0.29
5.0	99.49 ± 0.45	100.00 ± 0.00	23.44 ± 0.57	1.05 ± 0.29

As we can see the determinacy of the classifier drops off as s increases with a less significant effect on the single accuracy. This pattern suggests a fundamental trade-off when adjusting the s parameter: while a higher s might be seen to enhance the model's flexibility in handling uncertain data, it substantially weakens the classifier's decisiveness in making clear, definitive classifications. The stark reduction in determinacy can be attributed to the classifier's increasing hesitation to commit

to a single class as s increases, thereby expanding the credal set which in turn reflects a broader spectrum of potential classes. We see that the NCC$_\epsilon$ under interval dominance quickly reaches 100% accuracy which is expected due to the significantly low determinacy. The decrease in accuracy under credal dominance after $s = 2$ is likely due to the model under-fitting to the data. Under-fitting comes from a model failing to capture the underlying patterns of the data. This results in worse performance on both training and testing sets due to the model's high bias and its overly generalised assumptions about data relationships, however further analysis would be required to verify this. As such an appropriate value for s would apparently be $s = 2$ as this maximises the accuracy while still maintaining an effective level of determinacy. The graphs below show these relationships.

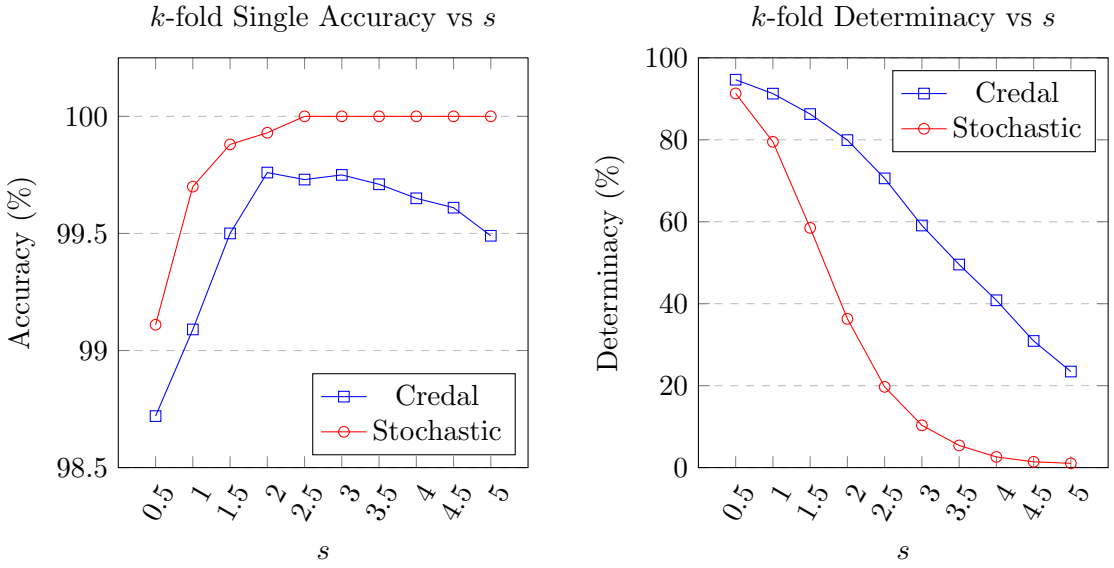


Figure 6.2: k-fold performance metrics showing changes in Accuracy and Determinacy for various s values.

6.4 Experiment 3: NCC$_\epsilon$ vs. NBC

In our final experiment we will compare on the efficacy of the NCC$_\epsilon$ with that of the NBC. In this experiment we will take $\epsilon = 0.25$ and $s = 1$. Aswell as comparing directly to the NBC we will also look at the accuracy of the NBC specifically in cases where the NCC$_\epsilon$ under credal dominance has been indeterminate. It is worth noting the determinacy of the NBC is trivially 100% and so is omitted from the table below. Once again each of these metrics has been computed using a k-fold cross validation ($k = 5$) and a random seed of 5.

Table 6.3: Comparison of Classifier Performance

Classifier	Single Accuracy (%)	Determinacy (%)
NCC (Credal Dominance)	99.09 ± 0.34	91.26 ± 0.46
NCC (Stochastic Dominance)	99.70 ± 0.25	79.51 ± 1.19
NBC	98.21 ± 0.43	-
NBC (Credal Indeterminate Cases)	56.38 ± 7.97	-

As we can see both the NCC$_\epsilon$ under credal dominance and interval dominance demonstrated superior accuracy to the NBC in instances where they made determinate predictions. Approximately

8.74% of the time, the NCC$_\epsilon$ reserved judgement under credal dominance and 19.49% under interval, highlighting instances with insufficient information to make a confident classification. Under these circumstances, the Naive Bayes Classifier (NBC) tends to underperform, with its accuracy on these indeterminate cases falling significantly to 56.38%. This is short not only of the NBC's overall accuracy of 98.21% but also, of both the NCC$_\epsilon$'s single accuracies. This phenomenon highlights the NCC$_\epsilon$'s capability to selectively target a subset of data where it can make strong, reliable predictions, less affected by the overarching uncertainties in prior knowledge. The NBC's performance on the subset where the NCC$_\epsilon$ withholds judgment attains a 56.38% accuracy which still surpasses the 50% we would expect to see when randomly guessing. This outcome may imply a breach of the independence assumption among attributes given the class in the dataset, prompting the exploration of more sophisticated credal classifiers to better accommodate such dependencies.

Reflecting on the football analogy mentioned earlier (see example 5.3.1), the increase in determinacy from interval dominance to credal dominance aligns with expectations. By treating the credal set strictly as an interval, the classifier fails to utilise all possible information within the set. This approach renders the classifier less decisive. Essentially, predictions made under interval dominance are more cautious than that of credal dominance.

Having said this, depending on the specific requirements of the classification problem our interval dominance could prove more effective. Especially in cases where speed and accuracy is valued highly. Credal dominance often involves a numerical optimisation, where as interval dominance is simply a comparison of two bounds and so while in this paper we have not delved into the computational complexities of these algorithms we can note that interval dominance is orders of magnitude times quicker than credal dominance and so in instances where speed is paramount this may serve as a superior option.

These experiments underscored the capability of the NCC$_\epsilon$ to provide robust and determinate predictions in the domain of email classification, outperforming the NBC, particularly in instances of high uncertainty. The ability to identify and withhold judgement on ambiguous instances presents a significant advantage, enabling more informed decision-making in the real-world application of spam filtering.

6.5 Object-Oriented Design

My implementation of the NCC$_\epsilon$ is accessible via a dedicated GitHub repository [16], providing transparency and reproducibility of all results. The NCC$_\epsilon$ was implemented in Python using an object-oriented approach, encapsulating its functionality within a modular and reusable structure. This design not only enhances the readability and maintainability of the code but also facilitates the integration of the NCC as an independent module.

Comprehensive documentation and usage instructions are provided in the GitHub repository's README file, which can be found at `https://github.com/zaccheus-lines/EmailSpamCredalClassifier`. This README explains how to install, configure, and utilise the classifier, ensuring that users can easily adopt and integrate the NCC into their own projects. Below is the code required to run the each of these experiments in the `main()` function of the file `experiments.py`.

```
def main():
  X, y = prepare_data()
  epsilon_test(X,y, seed = 5)
  s_test(X,y,seed=5)
  comparison(X,y, seed=5)
```

Chapter 7

Conclusion

In this report, we have explored various solutions to the challenges associated with classification and their applications in email spam filtering. Specifically, we have explored the NCC, that extends the conventional NBC through the application of imprecise probability theory to better handle uncertainty in spam detection.

Our main contributions comprise, us building on the work of Zaffalon [29] by deriving closed-form expressions for the upper and lower bounds of probabilities of observing a class given an instance of a feature vector, $p(c|\mathbf{x})$. We have integrated both interval and credal dominance with the work of Corani and Benavoli [5] to get two versions of the NCC_ϵ that we have implemented and tested in an open-source Python repository.

Our investigations began with an analysis of feature selection and encoding using the Apache SpamAssassin [2] email corpus. The work of Zdziarski [31] provided the foundations for this exploration. While we only opted for simple word-by-word tokenisation in our implementation the insights of Zdziarski [31] into n-gram tokenisation and much more covered in his 2005 book "Ending Spam" could prove instrumental in improving the efficacy of the classifier in later research.

This was followed by a rigorous derivation of the NBC following the work of Bishop [3]. Here we provide our derivations of the maximum likelihood estimates for the unknown parameters associated with the NBC using Lagrangian optimisation techniques [26]; having this grounding allowed us to understand the fundamentals of Bayesian classification.

This then motivated our exploration of Walley [27]'s IDM and its application in the NCC [29]. We used properties of the gamma function to provide derivations for the posterior expectations under the Dirichlet distribution, previously stated without derivation in the literature [27]. A potential avenue for further research is to look at integrating the uniform prior into Zaffalon's [29] model by allowing for multiple values for the s parameter, one for the class and one the attribute.

Moreover, our research delved into the methods for determining the dominance of one credal set over another, a critical consideration when the direct comparison of probabilities, as advocated by the Bayesian approach, is insufficient. Inspired by the contributions of Zaffalon [29], we conducted an in-depth exploration of interval dominance, providing a nuanced approach to decision-making under uncertainty within the framework of credal sets.

The practical application of the NCC_ϵ was demonstrated through three distinct experiments using our custom implementation. These experiments not only validated our models against the NBC but also highlighted the flexibility and robustness of our approach under varying conditions. The results from these tests underscore the potential of extending traditional classification techniques with imprecise probability models to improve the accuracy and reliability of spam filtering systems.

Ultimately, our work contributes to both the theoretical landscape and practical applications of classification. This not only assists in advancing academic knowledge but also in implementing more effective and adaptable tools for combating spam.

Bibliography

[1] Python Standard Library: email.parser. `https://docs.python.org/3/library/email.parser.html`. Accessed: [Insert date here].

[2] Apache Software Foundation. Apache spamassassin public corpus. `https://spamassassin.apache.org/old/publiccorpus/`.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. ISBN 978-0-387-31073-2.

[4] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. Cengage Learning, 10 edition, 2015. ISBN 978-1305253667.

[5] Giorgio Corani and Alessio Benavoli. Restricting the idm for classification. *Communications in computer and information science*, pages 328–337, 01 2010. doi:10.1007/978-3-642-14055-6_34.

[6] S. R. Dalal and W. J. Hall. Approximating priors by mixtures of natural conjugate priors. *Journal of the Royal Statistical Society. Series B (Methodological)*, 45:278–286, 1983. URL `https://www.jstor.org/stable/2345533`.

[7] Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics*. Addison Wesley, New York, 3 edition, 2002.

[8] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. Wiley, second edition, 2001.

[9] I. J. Good. *The estimation of probabilities*. MIT Press, Cambridge (MA), 1965.

[10] S Haldane. On a method of estimating frequencies. *Biometrika*, 33:222–222, 11 1945. doi:10.2307/2332299.

[11] Robert V. Hogg and Allen T. Craig. *Introduction to Mathematical Statistics*. The MacMillan Comapny, London, 3 edition, 1959.

[12] J. M. Keynes. *A Treatise on Probability*. Macmillan, London, 1921.

[13] Henry E. Kyburg. Bayesian and non-bayesian evidential updating. *Artificial Intelligence*, 31: 271–293, 03 1987. doi:10.1016/0004-3702(87)90068-3.

[14] P.S. Laplace. *Théorie analytique des probabilités*. Mme. Ve Courcier, Paris, 1812.

[15] Isaac Levi. On indeterminate probabilities. *Journal of Philosophy*, 71:391–418, 1974.

[16] Zaccheus P. Lines. EmailSpamCredalClassifier: Implementing a credal approach for email spam classification. `https://github.com/zaccheus-lines/EmailSpamCredalClassifier`, 2023. Accessed: 2023-04-13.

[17] Odunayo Ogundepo, Xinyu Zhang, and Jimmy Lin. Better than whitespace: Information retrieval for languages without custom tokenizers. *arXiv (Cornell University)*, 10 2022. doi:10.48550/arxiv.2210.05481.

[18] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, California, 1988.

[19] Howard Raiffa and Robert Schlaifer. *Applied Statistical Decision Theory*. MIT Press, 1961.

[20] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of Database Systems*, pages 532–538, 2009. doi:10.1007/978-0-387-39940-9_565. URL `https://link.springer.com/referenceworkentry/10.1007%2F978-0-387-39940-9_565`.

[21] Leonard Richardson. *Beautiful Soup Documentation*, 2020. URL `https://www.crummy.com/software/BeautifulSoup/bs4/doc/`. Accessed: [Insert date here].

[22] Sheldon M. Ross. *Introduction to Probability Models*. Academic Press, 11 edition, 2014. ISBN 978-0-12-407948-9. URL `https://www.elsevier.com/books/introduction-to-probability-models/ross/978-0-12-407948-9`.

[23] Mehran Sahami, Susan Dumais, David Heckerman, and Eric Horvitz. A bayesian approach to filtering junk e-mail. Technical report, Microsoft Research, Stanford University, Stanford, CA and Redmond, WA, 1998.

[24] Gerard Salton and Michael J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill Computer Science Series. 1983.

[25] Scikit-Learn Developers. Cross-validation: evaluating estimator performance. `https://scikit-learn.org/stable/modules/cross_validation.html`, 2024. Accessed: 2024-04-20.

[26] James Stewart. *Calculus: Early Transcendentals*. Cengage Learning, 8 edition, 2015. ISBN 978-1285741550.

[27] Peter Walley. Inferences from multinomial data: Learning about a bag of marbles. *Journal of the Royal Statistical Society, Series B*, 58(1):3–34, 1996. URL `http://www.jstor.org/stable/2346164`.

[28] Marco Zaffalon. A credal approach to naive classification. In G. de Cooman, F. G. Cozman, S. Moral, and P. Walley, editors, *ISIPTA '99: Proceedings of the First International Symposium on Imprecise Probabilities and Their Applications*, pages 405–414, Ghent, 1999. Imprecise Probabilities Project.

[29] Marco Zaffalon. Statistical inference of the naive credal classifier. In G. de Cooman, T. L. Fine, and T. Seidenfeld, editors, *ISIPTA '01: Proceedings of the Second International Symposium on Imprecise Probabilities and Their Applications*, pages 384–393. Shaker Publishing, Maastricht, 2001.

[30] Marco Zaffalon. The naive credal classifier. *Journal of Statistical Planning and Inference*, 105(1): 5–21, 2002. ISSN 0378-3758. doi:https://doi.org/10.1016/S0378-3758(01)00201-4. URL `https://www.sciencedirect.com/science/article/pii/S0378375801002014`. Imprecise Probability Models and their Applications.

[31] Jonathan A. Zdziarski. *Ending Spam, Bayesian Content Filtering and the Art of Statistical Language Classification*. No Starch Press, San Francisco, 2005.

[32] Harry Zhang. The optimality of naive bayes. In *FLAIRS2004 Conference*, 2004.