Probabilistic Machine Learning: Methods of Inference

Author: Kheeran K. Naidu

Supervisor: Dr Carl Henrik Ek

11th April 2021

Acknowledgement of Sources

For all ideas taken from other sources (books, articles, internet), the source of the ideas is mentioned in the main text and fully referenced at the end of the report.

All material which is quoted essentially word-for-word from other sources is given in quotation marks and referenced.

Pictures and diagrams copied from the internet or other sources are labelled with a reference to the web page or book, article etc.

Signed 

Date 29/05/2020

Abstract

Probabilistic machine learning frames the problem of learning as finding the parameters of a model that best explain the data. This is done by the fundamental inference step of calculating the probability distribution of the hidden parameters given the observed data. In most cases, this is a difficult to compute probability distribution. The most common approach to solving this is with the Monte Carlo Markov Chain (MCMC). Other approaches include variational inference and its scalable counterpart, stochastic variational inference. MCMC is a sampling method which has few limitations on the models it can be used for. It is accurate but slow for large datasets. Variational inference is superseded by stochastic variational inference in all capacities. Stochastic variational inference frames the problem as an optimisation problem and efficiently finds an approximation using noisy natural gradient updates. Alhough it is outperformed by MCMC on smaller tasks, stochastic variational inference dominates on massive datasets.

Contents

1 Introduction

Probabilistic machine learning frames the problem of learning as finding the parameters of a model that best explain the data. This is done by the fundamental inference step of calculating the probability distribution of the hidden parameters given the observed data. In most cases, this is a difficult to compute probability distribution. The most common approach to solving this is with the Monte Carlo Markov Chain (MCMC). This is method samples directly from the distribution in question and with sufficiently many samples, provides an accurate approximation of the distribution. The problem with this method arises when the size of the dataset and the number of hidden parameters increases.

Variational inference is another method of solving this problem. It phrases the problem as an optimisation problem. This is a much more efficient algorithm over MCMC. However, the accuracy of the model takes a hit by virtue of the difficult to compute distribution being modelled by relaxed assumptions of the original model. Additionally, this method of inference scales better than MCMC, but is still inefficient for massive datasets.

On the other hand, stochastic variational inference provides a scalable alternative. Its assumptions are the same as that of variational inference, however, it is implemented using a method of stochastic optimisation. Additionally, the fundamental updates of the optimisation consider the natural gradient which proves to have more effective learning. This method is an overall improvement on variational inference. Although, it still limited by the relaxed assumption of the original model.

In recent years, these methods of machine learning have been overshadowed by deep neural networks. In particular, the gold standard in sub tasks of natural language processing have been set by deep neural networks. However, topic modelling is one sub task of natural language processing where probabilistic machine learning thrives. This project uses the Latent Dirichlet Allocation (LDA) topic model as the context for running and comparing the different methods of inference.

The results show that although MCMC outperforms the variational methods for models where there are fewer hidden variables however, the runtime of MCMC exponentially increases and becomes inefficient for larger datasets. Stochastic variational inference out performs variational inference in all settings and scales extremely well to large datasets.

Although these method of inference were run on the LDA topic model, the methods are generalised to a larger family of probabilistic machine learning models.

2 Overview of Natural Language Processing

2.1 What is Language?

Depending on the context in which it is used, *language* can be interpreted in several ways. The Oxford University Press defines it in the context of a *language of communication*, *language of a country*, *system of a language*, *language style* and even a *computing language*. These are all valid definitions and interpretations which constitute our understanding of what language truly is. Furthermore, language is deeply ingrained in culture, society and politics.

Going back 400 years to the Elizabethan Era, the most documented language of the time was Shakespearean English. Its grammar and pronunciation is considered similar to 21st century English; both of which fall under the classification of Modern English (Wolfe 1972). In this regard, English is considered to be fairly homogeneous. On the other hand, present day India has 22 officially recognised languages, each of which have their own vocabulary and grammar. Unofficially however, these numbers are estimated in the hundreds (Gadgil et al. 1998). Similarly in China, the official language is Standard Mandarin and there are several hundred unofficial languages which are scarcely known to us and have not been sufficiently recorded (Li 1973). At a lesser extent, there still exists a remnant of linguistic diversity in the North of France; the language known as Chti. That said, Chti is considered a 'seriously endangered' language by UNESCO (Duriez 2009). Countries like England and France have only one official language, though this was not always the case.

Noam Chomsky attributes the adoption of a single official language to nationalism. In an interview hosted by Al Page from the University of Washington in 1991 (Noam Chomsky 1991), he describes a time at the turn of the 20th century when villages in France spoke mutually unintelligible languages, and students would be taught French as the *lingua franca*[1]. It united people from different villages, allowed for general education on a larger scale and fuelled the development of society. On the other hand, this progression of the country led to the degradation of language diversity. He further asserts that the change in language is unpredictable. There are too many factors affecting it, and even with a complete understanding of the regional linguistics this would be a near impossible task.

It has been noted that prior to the first contact with Europeans, thousands of languages were spoken in the North and South Americas. Some of these indigenous languages - from Mayan civilisations (Robertson 1992) - have stood the test of time, but many have fallen to conquest. The most notable of these conquests was Christopher Columbus's 'discovery' of the West Indies (Williams 2020) in the late 15th century. Throughout the 15th century the Portuguese explored the world through Northern Africa, followed by the West Coast of Africa, then making their way to South Asia. By the early 16th century, the Portuguese general Alfonso D'Albuquerque had entered the great Straits of Malacca, invading the rich Malay port city of Malacca. The Portuguese occupied Malacca for over a century. In contrast to the little remnants of a once culturally and linguistically diverse people of the Americas (Bigelow et al. 1998), the

[1] A language used for communication between groups of people who natively speak different languages

influence that the Portuguese had on the Malay language can still be seen today (Adam 1991). For example, words corresponding to trade and navigation have origins from the Portuguese.

The loss in diversity of language is usually associated with war, suffering and destruction. Nevertheless, it also brings development, unity and widespread communication. The British colonisation of the Americas, Africa, Australasia, South, Southeast and parts of East Asia led to English becoming the lingua franca of the world. In countries such as USA, Australia, and India, where English is an official language, communication has no barriers. Additionally, many countries where English is not an official language, it is still taught in schools as the language for international business (Neeley 2012). No matter where in the world you learnt English, you will be able to communicate with people at an international scale - provided an adequate adaptation period to the local dialect.

Language is riddled with historical influences; it grows, evolves and even diminishes through no fault of its own. It is complex, fluid and is systematically ingrained in our development. From birth, children are greatly influenced by their exposure to the world around them. They start communicating almost immediately with the aid of a simple language[2] consisting of facial expressions, hand gestures, and sounds. Their language is slowly refined to include nuances of culture, society and politics learnt through the language of their parents[3]. By 12-18 months, they are uttering their first words, by 2 years they are forming sentences, and by 5 they are off with their friends playing, communicating, and learning at an unprecedented rate. As they develop, their language develops too. The question arises: *how have they learnt language?*

Many parents, philosophers and psychologists would say words are learnt in part by the process of association, imitation and reward. However, the process of learning the rest of the language doesn't follow such patterns; children do not receive consistent feedback on the grammaticality of what they say. In fact, the learning of words and language requires rich mental capacities that interact in complicated ways (Bloom 2002, Chapter 1). In the late 19th century, neuroscientists became aware of the specific areas of the brain responsible for speech. These areas, now known as the Wernicke (Wernicke 1874) and Broca (Dronkers et al. 2007) areas, are required for speech processing and production respectively. Perhaps, Gottfried Leibniz was right in saying that language is an imperfect mirror of the mind (Leibniz et al. 1996, Chapter 7), and that this imperfection stems from the complexities of the mind being processed and expressed by only two tiny areas of the brain.

[2]Language has verbal and non-verbal constituents.

[3]Parents here could be interchanged with carers, family, siblings or any person(s) who has the most contact with the child.

2.2 Natural Language Processing

Machines, on the other hand, process language very differently. In 1950, Alan Turing proposed a simple criterion for intelligence, now known as the *Turing Test*. He noted that in order to determine if a machine can think, the words *machine* and *think* need to be defined which is, in itself, a difficult task. Instead, he formulated a problem described in terms of a game, the Imitation Game. The goal of the game is for an interrogator to determine the machine from the human judged solely on their responses to the interrogator's questions. It provides a way for us to tell whether a machine exhibits intelligent behaviour indistinguishable from that of a human (Turing 1950). The Imitation Game may very well just be one of wits and deduction; however, it was the starting point and birth place of natural language processing.

The field of natural language processing encompasses three major aspects of speech: transcription, synthesis, and processing. Speech transcription and synthesis are involved in the conversion between vibrations[4] and text. In the past, acoustics and phonetics were major components to making strides in these tasks. For example, researchers would create acoustic models for clusters of similar sounding words (Gillick et al. 1990). In 1983, research at IBM combined acoustic models and probabilistic methods to develop an approach to speech transcription - using linguistics to maximise the likelihood of a transcribed sentence (Bahl et al. 1983). Similarly, before the turn of the 21st century, speech synthesis had advances by modifying the parameters of natural speech - such as pitch, duration and energy - to produce a 'good voice quality' by concatenating diphones[5] (Moulines et al. 1990). Though recently, these knowledge-based approaches have largely been superseded by data-driven ones (Zen et al. 2009; Seide et al. 2011).

The final aspect of speech which is highlighted in this project is text processing[6]. It comprises a large set of difficult sub-tasks, some of which are text translation, grammar/spelling correction, chatbots, textual entailment, topic modelling and general language understanding. Although all of these are solvable problems by the brain, a machine is yet to solve them - let alone a few of them - just as well as humans can. As a matter of fact, the *no-free-lunch theorem* states that a universal learner does not exist (Shalev-Shwartz et al. 2014, Chapter 5). If a machine works well on a particular sub-task, it must sacrifice competence in another. Even with this theoretical limitation, much progress has been made in these sub tasks.

In 1966, researchers from the MIT Artificial Intelligence Lab programmed the very first chatbot, called ELIZA (Weizenbaum 1966). It communicates with people by rephrasing their statements as questions and urging them to continue talking to simulate a psychotherapist. By 1971, a similar programme called PARRY was designed to simulate a paranoid patient (Colby et al. 1971). Using an altered Turing Test where the interrogator was an expert psychiatrist, the psychiatrist struggled to correctly identify the machine from the patient suffering from paranoia (Colby et al. 1972).

[4]Speech is simply articulate vibrations that travel through the air or another medium and can be heard when they reach a person's or animal's ear.

[5]A diphone is an adjacent pair of phones (sounds) in an utterance.

[6]From here the phrases natural language processing, text processing and speech processing will be used interchangeably.

Much like an automated irrigation system, these chatbots function by following a set of logical rules based on the grammar of the language. At the time, text processing was predominantly based on this kind of intelligence, known as *logic programming* (Dahl 1994). However, in recent years, the underlying chatbot technology has been shifting away from rule-based to data-driven ones - some of these can be seen in the (highly criticised) annual *Loebner Prize Contest*.

In fact, artificial intelligence has been shifting towards data-driven methods and algorithms, with a specific focus on deep neural networks. Put simply, a neural network is trained by providing it a task to accomplish and allowing itself to tune the internal parameters that define the network, improving itself on the task (LeCun et al. 2015). It is made up of layers and a deep neural network is one which has many of these layers. Typically, a neural network is thought of as developing a deterministic input-output mapping of a task - i.e. a mapping between a set of words and a set of topics.

In 2018, researchers at Google AI developed a deep neural network which obtained state-of-the-art results on 11 natural language processing tasks - including textual entailment and general language understanding (Devlin et al. 2018). It uses two training tasks when learning: masked language model (MLM) and next sentence prediction (NSP). The MLM task involves the prediction of blanked out words from a piece of text; and the NSP task involves the prediction of whether two (almost) randomly selected sentences from a piece of text are adjacent or not. Soon after, researchers at Baidu released a similar but updated network, called ERNIE, which outperformed BERT in various knowledge-driven tasks while still maintaining state-of-the-art performance in other common natural language processing tasks (Zhang et al. 2019).

The structure of Mandarin is very different from English; words are made up of 1 or, more often, 2 characters. A character alone has a very different meaning from when it is paired with another character. For example 'caution' is *xiǎo xīn*, whereas its direct translation is 'small heart'. This means that masking a single character, as per the BERT networks initial specifications, isn't suitable. Therefore, Baidu's adaptation included a whole word masking approach - masking all characters of the word - which has been shown to have improvements on English natural language processing tasks (Cui et al. 2019).

The theory of neural networks has been studied for several decades (Dechter 1986; Haykin 1994). However, effective practical applications of them only began very recently. In 2012 a convolutional neural network[7] set the then gold standard on the ImageNet training set (Krizhevsky et al. 2012). Significant advances in computing had allowed for this feat. Considerable applications of neural networks have since been published branching out into many fields of science, humanities and arts (Litjens et al. 2017; Plecháč 2019) while gaining significant media coverage along the way[8]. Despite all of its success, there are some fundamental issues with using neural networks for learning (Marcus 2018).

Data-driven methods, in general, work on the premise that more data means better results. Even so, neural networks require a significantly greater amount of data to train

[7]Sometimes called shift invariant or space invariant artificial neural networks.

[8]Visit `https://www.economist.com/topics/artificial-intelligence` and `https://www.nytimes.com/topic/subject/artificial-intelligence`.

itself well. In 2013 researchers at Google's Deep Mind developed a neural network which could learn to play Atari games (Mnih et al. 2013). However, it needed millions of frames of gameplay for training before being able to outperform human players. Training a neural network on data at this scale is expensive in terms on money, time and also the environment.

A study on the energy consumption of hardware built for training neural networks - specific to natural language processing, including the BERT network previously discussed - revealed that training one of these networks can emit as much carbon as the lifetime of five mid-sized sedans (Strubell et al. 2019). Even with these high costs, neural networks aren't as great as the media makes them out to be. It has been shown that neural networks are not robust to adversarial strategies as its extent and limitations are unknown. In the task of language comprehension and answering questions based on the text, neural networks performed strikingly worse on average against adversarial strategies (Jia et al. 2017).

The results of neural networks should be taken as approximations, with the caveat that the uncertainty of these approximations are not quantifiable. A neural network will classify a written review as either positive or negative, it is incapable of saying "I am 60% certain that this is a positive review". This makes distinguishing between a 90% certainty that it is a positive review and a 60% one. Furthermore, once something has been learnt, there is no way of provably verifying what has been learnt.

An alternative approach to data-driven neural networks is probabilistic machine learning. Although many sub-tasks of natural language processing have been conquered by the implementation of deep neural networks, topic modelling is one such sub task where probabilistic methods are used. The context based nature of topic assignments and abundance of unlabelled data make probabilistic methods the ideal choice for modelling the data.

The idea of topic modelling is that each word of each document of a corpus has an inherent topic it relates to. This was first seen in 1988, by finding textual information and statistical synonyms of a text using *Latent Semantic Analysis* (LSA) (Dumais et al. 1988). The initial approach was a purely deterministic one; whereas in 1999, *probabilistic LSA* (pLSA) was developed which modelled the topic assignment of words as hidden variables which could be learnt given some data (T. Hoffman 1999) - the more data provided, the more certain the learnt topic assignments become. Further progress in a similar direction led to the development of the *Latent Dirichlet Allocation* (LDA) method[9] (Blei et al. 2003). This model has further hidden variables for the topic assignment of words, the topic distribution of documents and the word distributions of each topic; all of which can be learnt given a corpus of articles. The basic LDA has since been developed to include topic hierarchy (Griffiths et al. 2004), temporal information of documents (Blei et al. 2006) and conditioning on the previous word (Wallach 2006); there has also been a Bayesian non-parametric counterpart (Teh et al. 2005).

Although expansions on LDA have been developed, the basic LDA is an excellent model (Chapter 4) to show the fundamental problem that arises when learning in a

[9]The details haven't been explicitly described, however, the literature often refers to these methods of machine learning/artificial intelligence as models.

probabilistic machine learning setting (Chapter 5) - i.e. *marginalisation*. Furthermore, it provides a test bed for comparison of the different methods of solving this problem (Chapter 6). However, the starting point of this project is to provide a firm understanding of the Bayesian framework (Chapter 3) which all machine learning is fundamentally based on.

3 Bayesian Framework

This chapter is extremely important for building the fundamental theory used in this project. Bayes theorem is developed from the ground up, the Bayesian assumption is explained and the law of total probability (marginalisation) is proved. These three concepts provide the Bayesian framework and the first step to understanding a fundamental problem encountered by it. If you are comfortable with these concepts, please feel free to skip this chapter. Any references made to the definitions or theorems in this chapter will be cross-referenced.

The theory covered in this chapter is the accepted view of probability thought by Márton Balázs in the University of Bristol MATH11300 Probability 1 unit, with supplementary details from the "Introduction to Probability" book by Grinstead and Snell.

The Bayesian framework is central to all machine learning. It provides the theory and assumptions needed to characterise observed data using a model in a Bayesian context. The idea of using a model to characterise and understand data has been around for many centuries, dating back to ancient Greece (Evans 1984; Toomer 1998). However, it was not till the 18th century that Thomas Bayes developed, what is now known as, *Bayes theorem* (Dale 2012, Chapter 2). The difference between these contexts of models is the Bayesian assumption. This states that the parameters of a model are *random variables*, known as *priors*.

This chapter develops the Bayesian framework from the basic axioms of probability, revealing nuances of probability theory that machine learning is founded on.

3.1 Axioms of Probability

> *"On voit que la théorie des probabilités n'est, au fond, que le bon sens réduit au calcul."* - Pierre-Simon Laplace (Laplace 1814).

As Laplace puts it, probability theory is, at its core, simply common sense reduced to calculation. It is a language which can be used to formalise the world around us, and what we know or don't know about it. Some basic knowledge in set notation and combinatorial counting is assumed for this chapter.

Take the following sentence:

"It is a good day to be a pirate."

There are 32 characters; 2 of which are an 'e', 4 are an 'a', 8 are spaces and so on. The idea is to find some textual understanding of the sentence. Imagine that each of these 32 characters were written on equally sized pieces of paper, and noting the position of the character in the 32 character long sentence. This is to differentiate between the character 'a' in position 7 and 15, for example. Similar to a game of scrabble, all 32 pieces of paper are then put into an opaque bag which is shaken around to mix up the pieces of paper, simulating a random environment. *What are the chances of picking a character 'a'?*

One way of approaching this would be to repeat the experiment a hundred times and count the number of times an 'a' was picked. The number of times that happened divided by a hundred would be the chances of picking an 'a'. This method is a sampling method which gives an approximation to the answer. As a matter of fact, sampling methods are widely used in machine learning when the problem is intractable[10]. However, in this case, it can be solved analytically giving a precise answer.

Definition 3.1.1. A *sample space* is the set of all possible outcomes of an experiment, denoted Ω.

Definition 3.1.2. An element of the sample space is called an *elementary event* $\omega \in \Omega$ if and only if $|\Omega| < \infty$ and each elementary event is equally likely.

Definition 3.1.3. An *event* $E \subseteq \Omega$ is an outcome of the experiment, which is simply a set of elements of the sample space.

The *sample space* is the set of all possible sheets of papers that could be chosen from the bag - note that different sheets of paper with the same character on them are considered different elements of the sample space. The sample space is the following: $\Omega_1 = \{$'I', 't', ' ', 'i', 's', ' ', 'a', ' ', 'g', 'o', 'o', 'd', ' ', 'd', 'a', 'y', ' ', 't', 'o', ' ', 'b', 'e', ' ', 'a', ' ', 'p', 'i', 'r', 'a', 't', 'e', '.'$\}$.

Each of these elements are equally likely and are therefore defined as *elementary events*; an example of one such event is "picking the sheet of paper with the character 'a' in position 15 of the sentence". A non-elementary *event*, on the other hand, could be something along the lines of "the character 'a' was picked" which constitutes 4 elementary events of the sample space.

A probability is defined over a sample space. As expected, it is a way of describing the chances of an event of the sample space occurring. Formally, it is defined by a set of axioms.

Definition 3.1.4 (axioms of probability)**.** A probability $\mathbf{P}$ on a sample space Ω assigns numbers to events of Ω in such a way, that:

1. the probability of an event is non-negative, $\mathbf{P}\{E\} \geq 0$;
2. the probability of a sample space is one; $\mathbf{P}\{\Omega\} = 1$;
3. for any finitely or countably infinitely[11] many mutually exclusive events E_1, E_2, ...,

$$\mathbf{P}\{\bigcup_i E_i\} = \sum_i \mathbf{P}\{E_i\}.$$

These axioms are a formalisation of something that feels pretty natural. The first one simply says that an event occurring with a negative probability is nonsense. The

[10] A model is intractable when it is analytically or computationally too difficult to solve.

[11] Countably infinite is a term mathematicians like to use to mean things that we could count on our fingers, but would need infinitely many fingers to count. An example of this it the set of integers $\mathbb{Z}$.

second states that the chances of any event from the sample space happening - not a specific one - is always going to one. In other words, the chance of something not in the sample space happening is zero, or impossible. The third is a bit more tricky. If the events in question aren't partly or wholly the same event then the chance of all the events occurring is the sum of their individual probabilities. For example, the events "picking an 'a' from the bag" and "picking an 'e' from the bag" are mutually exclusive. However, the events "picking an 'a' from the bag" and "picking the 'a' in position 15 of the sentence" are not. Another way for defining probability is the Kolomogrov system of probability (Jaynes 2003, Appendix A.1), however, it is not required for the purposes of this project. The above axioms of a probability are sufficient to develop the Bayesian framework and the field of probability theory in general.

`Answering the question:` The event in question is E_a = "the character 'a' is picked from the bag". Since $|\Omega_1| = 32$, the probability of each elementary event $\omega \in \Omega_1$ is $\mathbf{P}\{\omega\} = \frac{1}{32}$. It follows that E_a is made up of 4 elementary elements; therefore,

$$\mathbf{P}\{E_a\} = \sum_{i=1}^{4} \mathbf{P}\{\omega_i\} = \frac{1}{8}. \tag{1}$$

Definition 3.1.5. The *complement* of an event E, denoted E^c is defined as the set $\Omega \backslash E$. The set of all the events in the sample space not including event E.

Proposition 3.1.6. For any event, $\mathbf{P}\{E^c\} = 1 - \mathbf{P}\{E\}$.

Proof. We know that $E^c = \Omega \backslash E$, therefore, E^c and E are mutually exclusive and $E^c \cup E = \Omega$. Therefore by axioms 2 and 3:

$$\mathbf{P}\{E^c\} + \mathbf{P}\{E\} = \mathbf{P}\{E^c \cup E\} = \mathbf{P}\{\Omega\} = 1.$$

□

3.2 Joint Probability

Using only the axioms of probability requires carefully describing the sets of elementary events that make up each event. This is a tedious but necessary step in understanding probabilities. However, this naturally reveals what marginalisation fundamentally is in the context of the sample space.

Altering the question slightly, provides the basis for developing the joint probability: *What are the chances of picking a character 'a' followed by picking an 'e'?*

The sample space is now the set of all possible pairs of sheets of paper, each of which are elementary events, denoted as Ω_2. Using counting methods, $|\Omega_2| = 32 \times 31 = 992$. Starting with 32 choices in the bag followed by only 31 after the first pick. The outcome of the experiment is split into two sub-events. For simplicity, the events of the sample space are denoted as tuples. The first entry represents the first pick and the second entry the second pick. The outcome of each pick of an event is denoted by the position in the sentence that the character on the sheet of paper is from, where two subsequent picks cannot be the same sheet of paper. For example the elementary event of picking

the 'I' in position 1 followed by the 't' in position 2 is denoted by the tuple (s_1, s_2) whereas $\mathbf{P}\{(s_1, s_1)\} = 0$.

Events are represented by the set of elementary events which it corresponds to. Consider this example: the event of picking the sheet with the character 'o' in position 10 followed by picking any character 't' is the set of all the elementary events where the first entry of the tuple is s_{10} and the second is either s_2, s_{18} or s_{30}. This event is denoted using the following set notation: $\{(s_{10}, n) : n \in \{s_2, s_{18}, s_{30}\}\}$. Similarly, sub-events are represented in this notation. The sub-event where the second pick is a 't' is denoted as $\{(m, n) : m \in \Omega_1, n \in \{s_2, s_{18}, s_{30}\}\backslash m\}$. There is slight abuse of notation here; the backslash after the set is defined as the set not including the 'currently selected' element m. This explicitly excludes impossible events such as (s_1, s_1).

With these definitions in mind, the question is solved by a counting approach: first count the number of possible elementary events where the first pick is an 'a' and the second is an 'e', then divide it by the total number of elementary events. There are 4 possible sheets of paper with the character 'a' on it; s_7, s_{15}, s_{24} and s_{29}. Additionally, there are 2 possible sheets of paper with the character 'e' on it; s_{22} and s_{31}. Let the event $E_{ae} = \{(m, n) : m \in \{s_7, s_{15}, s_{24}, s_{29}\}, n \in \{s_{22}, s_{31}\}\backslash m\}$. By counting the possibilities, $|E_{ae}| = 4 \times 2 = 8$ and it follows that

$$\mathbf{P}\{E_{ae}\} = \frac{1}{124} \tag{2}$$

Another way to look at the question is using the joint probability. This approach considers the two sub events as separate events. Let E_{a1} be the event "picking an 'a' in the first pick" and E_{e2} be the event "picking an 'e' in the second pick" where $E_{a1} = \{(m, n) : m \in \{s_7, s_{15}, s_{24}, s_{29}\}, n \in \Omega_1\backslash m\}$ and $E_{e2} = \{(m, n) : m \in \Omega_1, n \in \{s_{22}, s_{31}\}\backslash m\}$. By counting we see that $|E_{a1}| = 4 \times 31 = 124$ and, with a bit more work, $|E_{e2}| = 62$. In this context the question would be: what is the probability of both E_{a1} and E_{e2}?

The event E_{a1} is the set of tuples from Ω_2 where the first entry is either a s_7, s_{15}, s_{24} or s_{29}, and, similarly, E_{e2} is the set of tuples from Ω_2 where the second entry is either a s_{22} or s_{31}. The goal is to find the set of elementary events where both of these events occur together; where the first entry is either a s_7, s_{15}, s_{24} or s_{29}, and the second entry is either a s_{22} or s_{31}. This is, very simply, the intersection of the two events, which is equivalent to E_{ae}. Therefore the joint probability of E_{a1} and E_{e2} is $\mathbf{P}\{E_{a1} \cap E_{e2}\} = \mathbf{P}\{E_{ae}\}$. Note that if the events are from different sample spaces, their intersection would be the null set making their joint probability zero.

Definition 3.2.1. Let $E_1, E_2, ...$ be events of a sample space Ω. Their joint probability is the probability of the intersection of all the events: $\forall i = \{1, 2, ...\}$,

$$\mathbf{P}\{E_1, E_2, ...\} = \mathbf{P}\{\bigcap_i E_i\}$$

3.3 Marginalisation

The set notation in the previous section is difficult. However, it is a complete way of looking at probabilities from the perspective of the sample space - albeit a rather simple one. In any case, a fundamental concept in machine learning has risen very naturally. It is the idea of *marginalisation*.

In the example, the experiment is made up of two parts; the first and second pick. The question asks to find the joint probability of E_{a1} and E_{e2}. The step of separating out these sub events and treating them as individual events is precisely marginalisation where in either case, E_{a1} or E_{e2}, the other pick has been marginalised out from the joint probability.

$$E_{e2} = \{(m, n) : m \in \Omega_1, n \in \{s_{22}, s_{31}\} \backslash m\} \tag{3}$$

can be described as marginalising out the first pick by considering all its possibilities.

Definition 3.3.1. Let $E_1, E_2, ...$ be finitely or countably infinitely many events of a sample space Ω. These events form a *partition* of Ω if and only if $E_i \cap E_j = \emptyset$ and $\bigcup_i E_i = \Omega$. The simplest partition of Ω would be $E, E^c \in \Omega$.

An elegant nuance here, which is very easy to overlook, is the fact that all the possibilities being considered in equation 3 are a *partition* of the sample space Ω_1. This is because each possibility $m \in \Omega_1$ is an elementary event. By definition, this implies that they are mutually exclusive. This concept is formalised by the *law of total probability*.

Theorem 3.3.2 (law of total probability)**.** *Let E be an event of a sample space Ω and $F_1, F_2, ...$ be a partition of Ω. It follows that*

$$\mathbf{P}\{E\} = \sum_i \mathbf{P}\{E \cap F_i\}.$$

Proof. The events $F_1, F_2, ...$ make up a partition of Ω, therefore by definition they are mutually exclusive. This implies that the events $E \cap F_1, E \cap F_2, ...$ are also mutually exclusive. Therefore by the third axiom of probability (Definition 3.1.4)

$$\sum_i \mathbf{P}\{E \cap F_i\} = \mathbf{P}\{\bigcup_i E \cap F_i\}$$

. The distributivity property of sets followed by Definition 3.3.1 completes the proof by the following:

$$\bigcup_i E \cap F_i = E \cup (\bigcap_i F_i) = E \cup \emptyset = E$$

.

□

This is a powerful tool used in machine learning that 'explains away' variables that are not included in the final model but also need to be considered when building the model. In equation 3, the first pick has been 'explained away'. This looks at only at the outcomes of the second pick without disregarding the affects of the first pick.

3.4 Conditional Probability

The conditional probability plays an important role in machine learning. It provides the language to consider *partial information* of the model. This could be the observed data or the parameters of the model.

In the previous section, the experiment was split up into parts and the joint probability of these parts was found. However, knowing only the outcome of the first pick, how would the probability of the outcome of the second pick change?

Imagine that the first pick was the character 'a'. What is the probability of picking an 'e' in the second pick? After having picked an 'a' from the bag in the first pick, the total number of sheets of paper left in the bag would have reduced by one, however, the number of sheets with 'e' characters in the bag would still be two. From here, the probability of picking an 'e' is $\frac{2}{31}$.

In this case, the partial information given is that the first pick is an 'a', call this event E_{a1}. This notion is incorporated in probability theory using the vertical bar symbol (|); it denotes that some probabilistic information is known. Given E_{a1}, the sample space of the experiment reduces to only the tuples where the first entry is either s_7, s_{15}, s_{24} or s_{29}. This is denoted as $\Omega_2|E_{a1}$ and read as "Ω_2 given E_{a1}". It follows that $|\Omega_2|E_{a1}| = 4 \times 31 = 124$ and $\Omega_2|E_{a1} = \{(m,n) : m \in \{s_7, s_{15}, s_{24}, s_{29}\}, n \in \Omega_1 \backslash m\}$. Picking an 'e' in the second pick, E_{e2}, only has 8 outcomes in the new sample space. Therefore, the probability of picking an 'e' given E_{a1} is $\frac{2}{31}$.

$$\mathbf{P}\{E_{e2}|E_{a1}\} = \frac{2}{31}. \tag{4}$$

This can be generalised; find the number of elementary events where both events E_{a1} and E_{e2} occur, $|E_{a1} \cap E_{e2}|$, and divided it by the number of elementary events in the given event, $|E_{a1}|$. Since the size of the sample space $|\Omega_2|$ will cancel itself out, the conditional probability is indeed the joint probability divided by the probability of the given event.

Verify this by finding the answer to the revised question using this fact and checking the answer; the probability of E_{ae} (eq 2) is the probability of E_{e2} given E_{a1} (eq 4) multiplied by the probability of E_{a1} (eq 1). The events E_a and E_{a1} are equivalent.

$$\mathbf{P}\{E_{ae}\} = \mathbf{P}\{E_{a1}, E_{e2}\} = \mathbf{P}\{E_{e2}|E_{a1}\} \cdot \mathbf{P}\{E_{a1}\}.$$

Definition 3.4.1. Let E and F be events of a sample space Ω such that $\mathbf{P}\{F\} > 0$. This is an assumption that we always make. Then the conditional probability of E given F is defined as:

$$\mathbf{P}\{E|F\} := \frac{\mathbf{P}\{E \cap F\}}{\mathbf{P}\{F\}}.$$

The conditional probability is seen as well-behaved. Once the sample space is reduced using the partial information, the conditional probability satisfies the axioms of probability in Definition 3.1.4.

Proposition 3.4.2. Let $E_1, E_2, ..., E_n$ be events of a sample space Ω. The *multiplication rule* states the following:

$$\mathbf{P}\{\bigcap_{i=1}^{n} E_i\} = \mathbf{P}\{E_1\} \cdot \mathbf{P}\{E_2|E_1\} \cdot \mathbf{P}\{E_3|E_1 \cap E_2\} \cdot ... \cdot \mathbf{P}\{E_n|\bigcap_{j=1}^{n-1} E_j\}.$$

Proof. This can be seen by using Definition 3.4.1 to write out the conditionals. □

Partial information doesn't always provide information which affects the likelihood of an event. In these cases, the outcome of one event has no affect on the outcome of the other. As such,they are called *independent* events.

Definition 3.4.3. Let E and F be events of a sample space Ω. Events E and F are considered *independent* if and only if $\mathbf{P}\{E|F\} = \mathbf{P}\{E\}$.

3.5 Bayes Theorem

Bayes theorem describes the fundamental learning step of machine learning. Put simply, this inference step enables the parameters of a model to be found given the data. This is commonly described as the posterior of the model.

The example is described as a sequence of events and follows a chronological order of events; the first pick followed by the second pick. It may be difficult to think about, however, the following question is a good motivation for Bayes theorem: *What are the chances of picking a character 'a' in the first pick given the second pick is the character 'e'?*

Theorem 3.5.1. *Let E and F be events of a sample space Ω such that $\mathbf{P}\{E\} > 0$. The conditional probability of F given E is as follows:*

$$\mathbf{P}\{F|E\} = \frac{\mathbf{P}\{E|F\} \cdot \mathbf{P}\{F\}}{\mathbf{P}\{E\}}.$$

Proof. By proposition 3.4.2 followed by definition 3.4.1,

$$\frac{\mathbf{P}\{E|F\} \cdot \mathbf{P}\{F\}}{\mathbf{P}\{E\}} = \frac{\mathbf{P}\{E \cap F\}}{\mathbf{P}\{E\}} = \mathbf{P}\{F|E\}$$

□

Without thinking too much about the implications of an event in the future having already happened, the answer can be calculated by the following:

$$\mathbf{P}\{E_{a1}|E_{e2}\} = \frac{\mathbf{P}\{E_{a1} \cap E_{e2}\}}{\mathbf{P}\{E_{e2}\}} = \frac{\frac{1}{124}}{\frac{1}{16}} = \frac{4}{31}.$$

3.6 Random Variables

An event of a sample space is an abstract concept. It is difficult to generalise and deal with mathematically. We managed to answer the questions in the experiments without an issue, using some basic probability tools. However, the events in our examples are concepts like "the character 'a' is picked" which are very non-mathematical. As such, typical behaviour of experiments is difficult to quantify. Nonetheless, these concepts can be handled mathematically using *random variables*. Put simply, it is another way of representing events.

Definition 3.6.1. A *random variable* X is function $X : \Omega \to \mathbb{R}$ which maps the sample space to some measurable space, which, for simplicity, is the real numbers.

Let's look at the same experiment we have been dealing with, but from a different perspective. Let X_{a1} be the random variable that indicates if event $E_{a1} \in \Omega_2$ has occurred. This is the simplest random variable, called the indicator variable, and is defined for some event $E \in \Omega_2$ as

$$X_{a1}(E) = \mathbb{I}\{E \subseteq E_{a1}\} := \begin{cases} 1 & \text{if } E \subseteq E_{a1} \\ 0 & \text{otherwise.} \end{cases}$$

Definition 3.6.2. A random variable X that can take on finitely or countably infinitely many possible values is called *discrete*. Similarly, one that can take on uncountably infinitely many possible values is called *continuous*.

The probability of the random variable X_{a1} being one is the same as the probability of the event E_{a1} occuring; $\mathbf{P}\{X_{a1} = 1\} = \mathbf{P}\{E_{a1}\}$. And, similarly, by how an indicator variable is defined, $\mathbf{P}\{X_{a1} = 0\} = \mathbf{P}\{(E_{a1})^c\}$. Indicator variables only take on two possible values. Therefore, by Definition 3.6.2, they are *discrete random variables*.

Let X_{e2} and X_{ae} be the indicator random variables for event E_{e2} and E_{ae} respectively. The solutions to the three different experiments we've carried out can be phrased in terms of these random variables:

1. $\mathbf{P}\{E_{a1}\} = \mathbf{P}\{X_{a1} = 1\} = \frac{1}{8}$,
2. $\mathbf{P}\{E_{ae}\} = \mathbf{P}\{X_{ae} = 1\} = \frac{1}{124}$, and
3. $\mathbf{P}\{E_{e1}|E_{e2}\} = \mathbf{P}\{X_{e1} = 1|X_{e2} = 1\} = \frac{4}{31}$.

Random variables behave very similarly to events of a sample space, however, we can deal with them mathematically. Random variables can take on real values and the likeliness of it happening can be represented by a function called the *probability distribution*.

Definition 3.6.3. Let X be a discrete random variable with possible values x_1, x_2, ... The *probability distribution* or *probability mass function* (pmf) of a random variable indicates the probabilities of these possible values: $\forall i = \{1, 2, ...\}$,

$$p_X(x_i) = \mathbf{P}\{X = x_i\}.$$

The random variable X_{a1} can be represented by the the probability distribution, $p_{X_{a1}}$ which, in turn, can be visualised as seen in Figure 1. Once we have the random variable and its corresponding probability distribution, its typical behaviour can be quantified. The most frequently used quantities for this are the *expectation* and *variance* of the random variable.

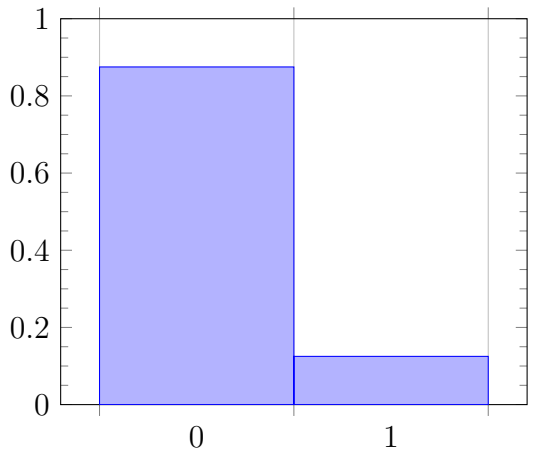


Figure 1: A graphical view of the probability mass function of X_{a1}.

Definition 3.6.4. The *expectation*, *mean* or *expected* value of a random variable X is defined as:

- for a discrete X;
$$\sum_i x_i \cdot p(x_i)$$
- for a continuous X;
$$\int_{\mathbb{R}} x \cdot p(x) dx.$$

The *expected value* of a random variable is the weighted average of the possible values, weighted by their likelihood. It provides a best guess at what the outcome of the experiment would be. Take the random variable X_{a1}. It only has two possibilities, $X_{a1} = 0$ or 1. Therefore, the expected value of an indicator variable is the probability of the event it is indicating:

$$\mathbb{E}[X_{a1}] = 0 \cdot p(0) + 1 \cdot p(1) = p(1) = \frac{1}{8}.$$

Definition 3.6.5. The variance (squared standard deviation) of a random variable is defined as:

$$Var(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2.$$

The *variance* of a random variable calculates the spread of the distribution around its expected value. It provides a sense of how much to trust the expected value. It is defined as the squared deviation of the random variable from its expected value. In

the case of the random variable X_{a1}, its variance is:

$$\mathbb{E}[(X_{a1})^2] - (\mathbb{E}[X_{a1}])^2 = \frac{1}{8} - \frac{1}{64} = \frac{7}{64}.$$

Let's compare this to a random variable Y which is an indicator variable for some event in the sample space Ω_y which occurs half of the time. This is a random variable where all its values are equally likely, with the expected value $\mathbb{E}[Y] = \frac{1}{2}$. The variance of such a variable is, $Var(Y) = \frac{16}{64}$. This is more than double $Var(X_{a1})$.

These methods of quantifying random variables are very important as their distributions are usually very dissimilar, making it, otherwise, difficult to draw any comparisons between them.

3.7 Typical Distributions

Probability is a mathematical tool which allows us to deal with uncertainty. It is widespread and used in all domains, down to the fundamental building blocks of the Universe, such as quarks. The distributions of these probabilities can vary drastically, however, there are typical distributions of probabilities which help us analytically deal with them. Here, we will describe the typical distributions required for this project. Each of these typical distributions will be described by their support $x \in \chi$, parameters and probability distribution function. Additionally, we will state some important quantities of them.

Bernoulli Distribution

This is of the simplest distributions, which we have already seen - in the form of an indicator variable. It is a discrete probability distribution which has support, $\chi = \{0, 1\}$. For parameter $\alpha \in [0, 1]$ the probability mass function is defined as:

$$p(x; \alpha) := \begin{cases} 1 - \alpha & \text{if } x = 0 \\ \alpha & \text{if } x = 1. \end{cases}$$

The expectation and variance of such a probability distribution are:

$$\mathbb{E}[X] = \alpha$$
$$Var(X) = \alpha(1 - \alpha).$$

Beta Distribution

This is a continuous distribution best described by a stick-breaking analogy: Imagine you have a stick, breaking it into two pieces would give you two smaller sticks whose lengths, added together, make up the original length. The support of this distribution, represents all possible ways of breaking a stick in the form of the corresponding proportion of the original length for one of the sticks - the second is not necessary as the two proportions would sum to one. Therefore the support is $\chi = [0, 1]$. Its parameters

are the real numbers $\alpha, \beta > 0$ with a probability density function defined as:

$$p(x; \alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

where $\Gamma(n) = (n-1)!$. The expectation and variance of such a probability distribution are:

$$\mathbb{E}[X] = \frac{\alpha}{\alpha + \beta}$$
$$Var(X) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}$$

The beta distribution is very versatile, and depending on the choice of parameters α and β, it can model the behaviour of other typical distributions. In particular, setting $\alpha = \beta = 1$ models a continuous distribution with a support that has uniform probability.

3.8 Bayesian Framework

The final component of the Bayesian framework which this project is fundamentally based on is the *Bayesian assumption*. It is the idea that parameters of a probability distribution can be modelled as a probability distribution in itself. Take the beta distribution. Its support - i.e. the kind of variables sampled from it - is exactly the set of parameters of a Bernoulli distribution. By the Bayesian assumption, the parameters of the Bernoulli distribution can be modelled by a beta distributed random variable.

Let D_n be a collection of N Bernoulli random variables each with parameter θ_n such that $D_n \sim Bern(\theta_n)\ \forall n = 1, ..., N$. Let each θ_n be a beta distribution with parameters $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ such that $\theta_n \sim beta(\boldsymbol{\alpha})\ \forall n = 1, ..., N$. Under the Bayesian assumption, the model can be phrased as $p(\mathbf{D}|\boldsymbol{\theta})p(\boldsymbol{\theta}; \boldsymbol{\alpha})$. As such, each random variable D_n is dependent on θ_n. This dependence between random variables can be described using a graphical model, as seen in Figure 2. The nodes with circles represent the random variables, the shaded node represents the observed random variable, and the non-shaded one represents the hidden variable, whereas, the non-shaded and non-circled node represents the fixed parameter of the model - which is pre-specified. The edge between nodes represents dependence between nodes; the directed edge points from the node that the receiving node is dependent on. Finally, the plate surrounding the nodes indicates that there are N such nodes in the model all of which share the same dependence, but do not share any inter-dependence.

In the case of probabilistic machine learning, the random variables D_n represent the observed data and each θ_n represents a setting of the model. The goal of any machine learning model is to find the settings of the model that best explain the data. The Bayesian assumption, allows us to incorporate our knowledge of the problem by virtue of setting the fixed parameter $\boldsymbol{\alpha}$.

In a probabilistic setting; the goal is to find the probability distribution of the settings of the model given the observed data as partial information, $p(\boldsymbol{\theta}|\mathbf{D})$. The

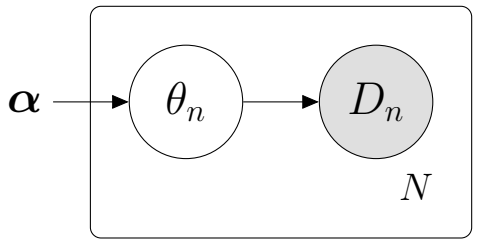


Figure 2: The graphical representation of a simple Bayesian model.

Bayesian framework is then completed by applying Bayes theorem (Theorem 3.5.1) to get the following;

$$\overbrace{p(\boldsymbol{\theta}|\mathbf{D})}^{posterior} = \frac{\overbrace{p(\mathbf{D}|\boldsymbol{\theta})}^{likelihood}\overbrace{p(\boldsymbol{\theta};\boldsymbol{\alpha})}^{prior}}{\underbrace{p(\mathbf{D})}_{evidence}}$$

where we have labelled the usual terminology for the components of the Bayesian framework. This is the foundation of machine learning and in this project we will see the difficulty in finding the required posterior and study methods of approximating it.

4 Latent Dirichlet Allocation

Topic modelling is a sub-task of natural language processing that tackles the problem which arises from the massive digitisation of information - in the form of books, magazines, journals, sound and images. It works as a tool to organise and search for relevant information in the sea of data. In the context of a text corpus, the main assumption of a topic model is that each word of each document of the corpus has a hidden topic assignment.

> *"The difference between capitalism per se and an economic system merely involving private ownership of material wealth is, rather curiously, provided by the Hindu tale of Ganesha and Kubera. For those who may not be aware, Ganesha, typically portrayed with an elephant head and pot belly, is the lord of wealth in Hinduism, he is traditionally associated with all things related to business and commerce and is the favourite god of all traders and merchants."*

Figure 3: An extract of an article written by Harsh Tiwari from The Pangean. Highlighted showing some topic assignments - labelled by hand and meant for illustrative purposes only.

Take the extract of an article titled *"American Psycho, Capitalism and the Hindu Gods"* shown in Figure 3. It is an article about the parallels between the movie American Psycho, capitalism and a traditional Hindu tale. Some of the words have been highlighted by hand to illustrate the potential[12] hidden topic assignments of the extract. Three topics have been indicated; politics (yelllow), trade (green) and religion (red). Labelling these words by hand isn't a difficult task, however, it is one that is dependent on knowledge and understanding of language. Therefore, these assignments may vary depending on the person who labels it.

Topic modelling has the goal of grouping words of the same topic without the need for pre-labelling data - this is an unsupervised learning task; whereas a supervised learning approach would require labelling the topic assignment of each word of the training data by hand. This means that massive datasets can be easily prepared for topic modelling. This has the caveat that, once words are grouped, the label of the group (topic) needs to be manually assigned; most of the time, this is fairly straightforward. For example, the grouping of the words `Hindu`, `Ganesha`, `Kubera` and `Hinduism` is very easily labelled *religion*; on the other hand, `material, wealth, business, commerce, traders` and `merchants` is labelled *trade*, but could have quite easily been labelled *economics*. The task of grouping these words in a text may seem straightforward, especially given our knowledge and experience using language, however, language is riddled with historical nuances and how we process it is still very unclear. Topic modelling tries to learn these groupings, not through understanding the nuances of language, but through statistical modelling of the textual information.

[12]These topic assignments are done for illustrative purposes and are by no means the output of some machine learning algorithm.

Latent Dirichlet Allocation (LDA) is one such topic model. The overall assumptions of the model is that for a fixed vocabulary and pre-determined number of topics, each topic asserts a different distribution of words over the vocabulary. For example, words like law, legal and legislation are more probable under the topic corresponding to politics than they are under the topic corresponding to economics. This is not to say that it is impossible for the word 'legislation' to be drawn from a topic assignment corresponding to economics, it is just very improbable. Furthermore, the model assumes that within the corpus, each word of each document has some topic assignment which follows the proportion of topics assigned to the document; therefore, if a document is made up of three topics with equal proportions, the topic assignment of the words that make up the document would reflect that.

Given that there are K topics in the corpus which has a vocabulary of size V, the assumptions of the model are best formalised using the generative stochastic process of LDA (Hoffman et al. 2013) outlined in Figure 4.

1. For each topic $k \in \{1, ..., K\}$,
   (a) sample a distribution over the vocabulary $\beta_k \sim Dir_V(\eta)$.
2. For each document $d \in \{1, ..., D\}$
   (a) sample a distribution over topics $\theta_d \sim Dir_K(\alpha)$.
   (b) For each word of the document $n \in \{1, ..., N_d\}$,
      i. sample a topic assignment $z_{d,n} \sim Cat(\theta_d)$, then
      ii. sample a word $w_{d,n} \sim Cat(\beta_{z_{d,n}})$.

Figure 4: The generative process of LDA.

The model is made up of $(K \times V + D \times K + \sum_{d=1}^{D} N_d \times K)$ variables. The interdependence of these variables on one another is fully described using the graphical model in Figure 5. The word and topic proportions, β_k and θ_d, require V and K positive values that sum to one respectively. Therefore, they are each modelled by a Dirichlet distribution (Appendix A.3). The topic and word assignments, $z_{d,n}$ and $w_{d,n}$ then use the corresponding proportions to select from K and V discrete values respectively. It follows that they are each modelled by a categorical distribution (Appendix A.1). Finally, α acts as our initial assumptions of how the topics are distributed in the documents, and η is our assumptions of how the vocabulary is distributed within the topics. These act as prior knowledge, a way of adding external information to the model.

The input of the LDA model is data from a corpus of documents. Before learning, the word distributions for each topic, the topic proportions of each document and the topic assignment of each word of each document is unknown - these are the hidden variables. The only information known is the generated word - the observed variable. This means that the model has $(K \times V + D \times K + \sum_{d=1}^{D} N_d \times K)$ variables that need to be learnt. We can phrase the problem as finding the conditional joint probability distribution of the hidden variables given the observed variables and prior knowledge - this is commonly known as the posterior of the model. Once we have this distribution, an analysis on it would yield the best setting of the hidden variables which generated the data, and how good the setting actually is. The difficult part is the inference of the

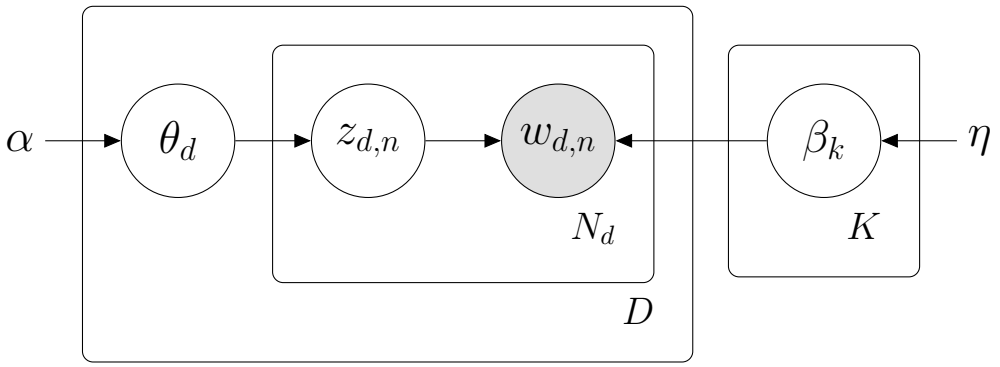


Figure 5: The graphical model representation of LDA.

posterior, $p(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}|\mathbf{w}; \alpha, \eta)$. We make a quick note that this notation for the posterior is equivalent to the following fully expanded one:

$$p(\beta_1, ..., \beta_k, \theta_1, ..., \theta_D, z_{1,1}, ..., z_{1,N_1}, ..., z_{D,N_D}|w_{1,1}, ..., w_{1,N_1}, ..., w_{D,N_D}; \alpha, \eta).$$

Using our knowledge of probability (in particular, definition 3.4.1), and the inter-dependence of the random variables (as seen in Figure 5), the posterior can be re-written as the following:

$$p(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}|\mathbf{w}; \alpha, \eta) = \frac{p(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w}; \alpha, \eta)}{p(\mathbf{w}; \alpha, \eta)} = \frac{p(\mathbf{w}|\mathbf{z}, \boldsymbol{\beta})p(\mathbf{z}|\boldsymbol{\theta})p(\boldsymbol{\beta}; \eta)p(\boldsymbol{\theta}; \alpha)}{p(\mathbf{w}; \alpha, \eta)}. \tag{5}$$

The difficulty in finding the posterior comes specifically from the denominator of equation 5 - commonly called the evidence of the model. It is found by marginalising out $\boldsymbol{\beta}, \boldsymbol{\theta}$ and $\mathbf{z}$ from the numerator using the *law of total probability* (Theorem 3.3.2), such that:

$$p(\mathbf{w}; \alpha, \eta) = \int \int \sum_{\mathbf{z}} p(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}, \mathbf{w}; \alpha, \eta) d\boldsymbol{\beta} d\boldsymbol{\theta}. \tag{6}$$

Computing this evidence requires considering - by summing or integrating over - all the different combinations of all the possible values of each of the $(K \times V + D \times K + \sum_{d=1}^{D} N_d \times K)$ hidden variables of the LDA model. This considers (1) all possible word distributions for each topic, (2) all possible topic proportions for each document, and (3) all possible topic assignments for each word for each document. Furthermore, the posterior cannot be analytically determined, due to the lack of a conjugate prior. Therefore, the posterior of the LDA model (equation 6) is computationally intractable. That said, there are methods of approximate inference of the posterior, which we will see in the next chapter.

5 Methods of Inference

The problem of an intractable evidence faced in building the LDA model is one that is encountered in many probabilistic machine learning models. In fact, we can generalise the problem and describe the state-of-the-art tools used to solve them.

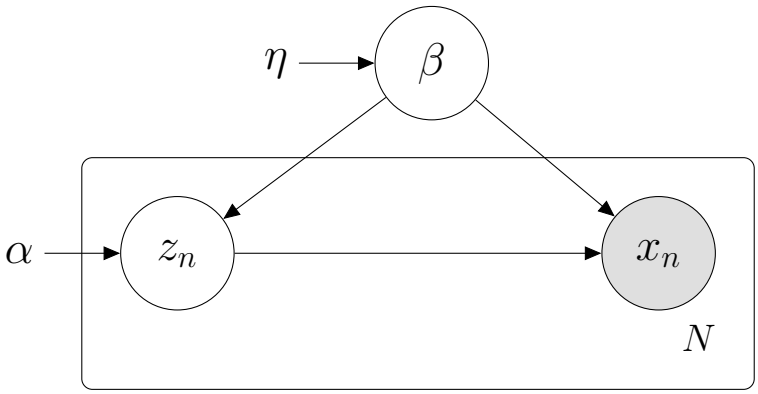


Figure 6: The graphical model representation of a generalised probabilistic model.

Probabilistic machine learning models can be described as the class of models which have *local* and *global* hidden variables, along with their fixed parameters, in addition to the *observed* variables, acquired from the data. This is described by the graphical model in Figure 6. The graphical model has a minor simplification where $\boldsymbol{\beta}, \mathbf{z_n}$ and $\mathbf{x_n}$ may be a collection of multiple random variables. This simplification does not affect the generalisation of the model.

We consider that there are N observations, $x_1, ..., x_N$; N local hidden variables, $z_1, ..., z_N$ with a corresponding vector of local fixed parameters α; and, a vector β of global hidden variables with fixed parameter vector η. The goal of this model is to find the setting of the parameters that best explain the data. This is done by finding the probability distribution of the hidden variables given the observations and fixed parameters, described by the following posterior:

$$p(\beta, \mathbf{z}|\mathbf{x}; \alpha, \eta) = \frac{p(\beta, \mathbf{z}, \mathbf{x}; \alpha, \eta)}{p(\mathbf{x}; \alpha, \eta)} = \frac{p(\mathbf{x}|\mathbf{z}, \beta)p(\mathbf{z}|\beta; \alpha)p(\beta; \eta)}{p(\mathbf{x}; \alpha, \eta)}.$$

In the cases where the model does not have a conjugate prior (Diaconis et al. 1979), the evidence of the posterior is intractable in high-dimensions, as we've seen similarly in the case of the LDA model, because it is found by marginalising out the hidden variables of the numerator - the joint distribution of the model - as follows;

$$p(\mathbf{x}; \alpha, \eta) = \int \int p(\beta, \mathbf{z}, \mathbf{x}; \alpha, \eta) d\beta d\mathbf{z}$$

where we have assumed, without loss of generality, continuous distributions for all the hidden random variables.

We discuss three methods for solving this problem; *Markov Chain Monte Carlo* (in particular, *Gibbs Sampling*), *Variational Inference* and *Stochastic Variational Inference*. Markov Chain Monte Carlo methods are the most commonly used and well

researched tool for finding an intractable posterior (Gamerman et al. 2006, Chapter 7). On the other hand, Variational Inference and its scalable counterpart, Stochastic Variational Inference, are not as widely used or deeply researched. This project aims to shine light on these methods and show their advantages and limitations.

5.1 Gibbs Sampling

Markov Chain Monte Carlo (MCMC) is a computationally driven method that characterises a probability distribution without knowing the distribution's full mathematical properties. It gives an asymptotically exact expectation of the statistics[13] of a distribution by means of simulation (Gilks 2005). It is essentially a Monte Carlo integration using Markov chains. Put simply, Monte Carlo integration is a method of estimating definite integrals by randomly sampling points for evaluation and taking the average (Grinstead et al. 2012). And, a Markov chain is a sequence of samples, where the next sample depends only on the current sample - it is *memoryless*. It is fully described by its *transition probabilities* which, for MCMC, we assume is fixed for all samples - a *homogeneous* Markov chain (Bishop 2006).

MCMC is Monte Carlo integration which draws samples to be evaluated following a well constructed Markov chain, which is achieved by the Metropolis-Hastings algorithm (Hastings 1970). MCMC uses a proposal distribution based on the previous sample, which can be any distribution with the same support as the posterior, to draw samples throughout the support in correct proportions. The *memoryless* property of the sequence of samples means that the starting point of the sequence becomes irrelevant after sufficiently many iterations, and the average of the sequence - disregarding an adequate *burn in* period - approximates the expected value of the posterior.

Before describing the algorithm, we note that, the goal of using MCMC is to find the hidden parameter settings of the model which best explain the data; i.e. finding the posterior. MCMC works at equivalently solving this by finding an estimate of the expected value of some function- which is often the identity map - of the posterior;

$$\mathbb{E}[f(\beta, \mathbf{z})|\mathbf{x}; \alpha, \eta] = \frac{\int f(\beta, \mathbf{z}) p(\mathbf{x}|\mathbf{z}, \beta) p(\mathbf{z}|\beta; \alpha) p(\beta; \eta)}{\int \int p(\beta, \mathbf{z}, \mathbf{x}; \alpha, \eta) d\beta d\mathbf{z}}.$$

This is sufficient when finding the parameters of the model as it provides the expected value of the parameters, which we know is the most probable setting of the model given the data.

The general form of MCMC is given by the Metropolis-Hastings algorithm (Hastings 1970). The sequence of sampled events is often denoted by t and is referred to as time steps. The initial state of the algorithm is denoted X_0. At each time step t, the next state X_{t+1} is determined by first sampling a candidate point Y from a proposal distribution q, $Y \sim q(.|X_t)$, whereby the candidate point Y is accepted with probability

[13]The statistics of a distribution is simply some function of the support of the distribution.

$a(X_t, Y)$ where;

$$a(X, Y) = min(1, \frac{p_{\beta,\mathbf{z},\mathbf{x}}(Y)q(X|Y)}{p_{\beta,\mathbf{z},\mathbf{x}}(X)q(Y|X)}. \tag{7}$$

Here $p_{\beta,\mathbf{z},\mathbf{x}}(.)$ is the joint distribution of the model. If the candidate point is accepted the next state becomes $X_{t+1} = Y$. If it is rejected, $X_{t+1} = X_t$. The pseudocode for Metropolis-Hastings is shown in Algorithm 1.

Algorithm 1 Metropolis-Hastings algorithm

```
Initialise X_t and set t = 0
repeat
    Sample a point Y ~ q(.|X_t)
    Sample a uniform random variable U ~ Uniform(0, 1)
    if U ≤ a(X_t, Y)
        X_{t+1} = Y
    else
        X_{t+1} = X_t
    Increment t
until forever
```

The Gibbs sampler (Geman et al. 1984) is a special case of the Metropolis-Hastings algorithm which updates each component of the the sample X_i separately. Additionally, the proposal distribution used for the update is the complete conditional[14] of the joint distribution;

$$q(Y_i|X) = p_{\beta,\mathbf{z},\mathbf{x}}(Y_i|X_i, X_{\setminus i}).$$

Here we use the notation $X_{\setminus i}$ to mean all components of X except X_i (Bishop 2006, Chapter 11). Substituting this into equation 7 gives an acceptance probability which is always one; as such, the Gibbs sampler candidates are always accepted. Therefore, the Metropolis-Hastings algorithm is simplified to subsequently sampling from the complete conditionals of the joint (Gilks et al. 1996).

5.2 Variational Inference

MCMC, as we've just seen, is a sampling method for finding the expected value of the posterior by providing asymptotically exact samples from the posterior. Variational Inference takes a different approach. It phrases the problem of finding an intractable posterior as an optimisation problem (Blei et al. 2003).

Even though variational inference is a relatively new method for finding the posterior, optimisation problems have been studied since as early as the 17th century, most notably by Pierre de Fermat. The basic idea of these types of problems is to find stationary points of a function where the slope of the curve, its gradient, is zero.

[14] A complete conditional distribution is the distribution of a hidden variable given all other hidden and observed variables of the model.

Fermat showed that the first derivative of a function represents its gradient which can be used to find these stationary points.

Variational inference is an algorithm where the problem of finding the posterior is altered to finding a similar distribution $q(.)$ which belongs to a family of probability distributions over the support of the posterior, $q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi}) \in \mathcal{D}$. The distributions based on the different combinations of the parameters make up the family of distributions. This is turned into an optimisation problem by finding the parameters of the family of distributions which corresponds to the distribution that is most similar to the posterior. This, however, requires a way of comparing probability distributions.

Relative Entropy

Information theory, fortunately, provides the tool for comparing two probability distributions for similarity. This is the *relative entropy* or *Kullback–Leibler divergence* between two distributions. In 1948, Claude Shannon developed this field of mathematics, now known as information theory. It provides a framework for measuring how much information there is in any outcome of an experiment (Shannon 1948). Events which are less likely to occur yield more information when they do occur, whereas, events that are extremely likely are somewhat expected and yield little information when they occur. In particular, he developed a way of finding the average information gain of a probability distribution, known as *entropy*. The relative entropy, on the other hand, was presented by Kullback and Leibler in 1951 (Kullback et al. 1951). It finds the average information gain of a probability distribution, $q(.)$, from the perspective of another, $p(.)$. This means that two identically distributed distributions would, on average, gain no information relative to one another. However, in any other case, the relative entropy would be greater than zero. This is proved by *Gibbs' inequality* (MacKay et al. 2003, Chapter 2). The relative entropy is defined as follows[15]:

$$D_{KL}(q(\mathbf{x})||p(\mathbf{x})) = \int q(\mathbf{x}) \log \frac{q(\mathbf{x})}{p(\mathbf{x})} d\mathbf{x}. \tag{8}$$

Using this, the problem can be phrased as an optimisation problem where the parameters λ and $\boldsymbol{\phi}$ of the family of distributions, $\mathcal{D}$, can be found by minimising the relative entropy to the posterior with respect to the parameters:

$$q(\beta, \mathbf{z}; \lambda^*, \boldsymbol{\phi}^*) = \underset{\lambda, \boldsymbol{\phi}}{\operatorname{argmin}} \; D_{KL}(q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi})|p(\beta, \mathbf{z}|\mathbf{x}; \alpha, \eta).$$

Evidence Lower Bound

This would provide the best approximate of the posterior within the family of distributions $\mathcal{D}$. The problem is that the calculation requires the log evidence which is intractable - if this wasn't the case, there would be no need for variational inference in the first place. The log evidence arises in the following expansion of the relative

[15]The standard practice of information theory is to use the base two logarithm as it was founded in the context of communication theory, however, any base logarithm can be used as long as it is standardised for all calculations.

entropy:

$$\begin{aligned} D_{KL}(q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi}) | p(\beta, \mathbf{z}|\mathbf{x}; \alpha, \eta) =& \mathbb{E}_q[\ln q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi})] - \mathbb{E}_q[\ln p(\beta, \mathbf{z}|\mathbf{x}; \alpha, \eta] \\ =& \mathbb{E}_q[\ln q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi})] - \mathbb{E}_q[\ln p(\beta, \mathbf{z}, \mathbf{x}; \alpha, \eta] \\ &+ \underbrace{\ln p(\mathbf{x}; \alpha, \eta)}_{\log evidence}. \end{aligned}$$

Re-arranging this equation shows an interesting property of the log evidence:

$$\ln p(\mathbf{x}; \alpha, \eta) = D_{KL}(q||p) + \underbrace{\mathbb{E}_q[\ln p(\beta, \mathbf{z}, \mathbf{x}; \alpha, \eta] - \mathbb{E}_q[\ln q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi})]}_{elbo}.$$

The log evidence is lower bounded by the indicated *elbo* term, appropriately called the *evidence lower bound*. This holds because the evidence is the normalising term of the posterior and is, therefore, constant. The goal of variational inference is to minimise the relative entropy between $q(.) \in \mathcal{D}$ and the posterior. Since the log evidence is constant, minimising the relative entropy is equivalent to maximising the *elbo* (Blei et al. 2003).

Naive Mean Field Variational Family

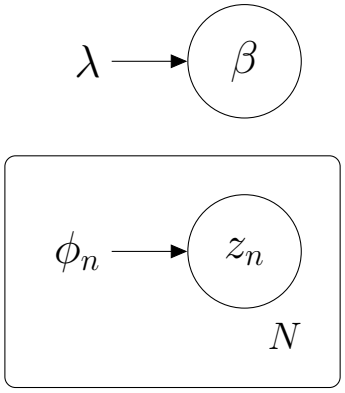


Figure 7: The graphical model representation of the naive mean field assumption.

The difficulty in maximising the *elbo* boils down to the choice of family of distributions $\mathcal{D}$. The difficulty is determined by the complexity of the family. Generally, the *naive mean field* variational family serves well. This is the family of distributions that approximates the model to one where the local and global hidden variables are *mutually independent* and are each conditioned on their own distinct parameter, called the *variational factor* (Wainwright et al. 2008, Chapter 5), as seen in Figure 7. This approximates the posterior with the joint distribution of the naive mean field assumption model:

$$q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi}) = q(\beta; \lambda) \prod_{n=1}^{N} q(z_n; \phi_n).$$

This mean field approximation is by no means a model of the data, in fact, the observed data is not included in the model. The link between the data is established by maximising the *elbo*. In turn, this minimises the dissimilarity between the approx-

imation and the posterior of the model by virtue of minimising the relative entropy.

Complete Conditionals in the Exponential Family

The final assumption of variational inference restricts the algorithm to the class of models where the complete conditionals of the hidden variables are in the exponential family of distributions (Hoffman et al. 2013). This is a broad family of distributions whose members include the uniform distribution, the Dirichlet distribution, the multinomial distribution and the Gaussian distribution (Jordan 2009). The requirement for a distribution to be part of the exponential family is that its probability distribution function can be written in the following form:

$$p(\mathbf{x}; \boldsymbol{\theta}) = h(\mathbf{x}) \exp\{\Phi(\boldsymbol{\theta}) \cdot T(\mathbf{x}) - A(\Phi(\boldsymbol{\theta})\}.$$

Appendix A.4 describes each component in further detail. By the independence assumption of the generalised probabilistic model (see Figure 6) the complete conditionals of the local and global variables are the following:

$$p(z_n|\beta, \mathbf{z}_{\setminus n}, \mathbf{x}; \alpha, \eta) = p(z_n|\beta, x_n; \alpha) \tag{9}$$
$$p(\beta|\mathbf{z}, \mathbf{x}; \alpha, \eta) = p(\beta|\mathbf{z}, \mathbf{x}; \eta) \tag{10}$$

This asserts that the complete conditionals of the local, $\mathbf{z}$, and global, β, hidden variables must be of the form:

$$p(z_n|\beta, x_n; \alpha) = h_l(z_n) \exp\{\Phi_l(\beta, x_n, \alpha) \cdot T_l(z_n) - A_l(\Phi_l(\beta, x_n, \alpha))\}, \tag{11}$$
$$p(\beta|\mathbf{z}, \mathbf{x}; \eta) = h_g(\beta) \exp\{\Phi_g(\mathbf{z}, \mathbf{x}, \eta) \cdot T_g(\beta) - A_g(\Phi_g(\mathbf{z}, \mathbf{x}, \eta)\}. \tag{12}$$

where the functions $h(.)$, $\Phi(.)$, $T(.)$ and $A(.)$ are significantly different for the local and global contexts, denoted by the subscripts l and g respectively. Additionally, the corresponding mean field approximation of the local and global parameters is set to the same distributions respectively:

$$q(z_n; \phi_n) = h_l(z_n) \exp\{\Phi_l(\phi_n) \cdot T_l(z_n) - A_l(\Phi_l(\phi_n))\}, \tag{13}$$
$$q(\beta; \lambda) = h_g(\beta) \exp\{\Phi_g(\lambda) \cdot T_g(\beta) - A_g(\Phi_g(\lambda))\}. \tag{14}$$

Note that there is an overload on notation of the functions that describe the exponential family of distributions. For example, even though $\Phi_{l_1}(\beta, x_n, \alpha)$ and $\Phi_{l_2}(\phi_n)$ map to the same natural parameter space, the domain of the function is different which makes them different functions. That said, the functions are one-to-one mappings to the same natural parameter space, this means that there exists some function that maps these two separate domains onto each other. One such function is $\Phi_{l_1}^{-1}(\Phi_{l_2}(\phi_n))$.

Coordinate Ascent Update

The problem of finding the posterior has been rephrased as finding the optimal variational factors, λ^* and $\boldsymbol{\phi}^*$, of the *naive mean field variational family* which maximise the *evidence lower bound*:

$$q(\beta, \mathbf{z}; \lambda^*, \boldsymbol{\phi}^*) = \underset{\lambda, \boldsymbol{\phi}}{\operatorname{argmax}}\ \mathbb{E}_q[\ln p(\beta, \mathbf{z}, \mathbf{x}; \alpha, \eta] - \mathbb{E}_q[\ln q(\beta, \mathbf{z}; \lambda, \boldsymbol{\phi})]$$

This optimisation problem can be solved by the coordinate ascent update algorithm (Bishop 2006, Chapter 10). Much like the Gibbs sampler, each variational factor of the *elbo* is updated individually, assuming that all the other variational parameters are fixed. For each local and global variational parameter, or coordinate, the *elbo* can be re-written to group all variables which do not change with respect to it as a constant:

$$\begin{aligned} elbo_{\phi_n} &= \mathbb{E}_{q(z_n;\phi_n)}[\ln p(z_n|\beta;\alpha)] - \mathbb{E}_{q(z_n;\phi_n)}[\ln q(z_n;\phi_n)] + const \\ elbo_{\lambda} &= \mathbb{E}_{q(\beta;\lambda)}[\ln p(\beta|\mathbf{z},\mathbf{x};\eta)] - \mathbb{E}_{q(\beta;\lambda)}[\ln q(\beta;\lambda)]] + const. \end{aligned}$$

Re-writing the *elbo* using the exponential family assumption gives the following:

$$\begin{aligned} elbo_{\phi_n} &= (\mathbb{E}_q[\Phi_l(\beta, x_n, \alpha)] - \Phi_l(\phi_n)) \cdot \nabla A_l(\Phi(\phi_n)) + A_l(\Phi(\phi_n)) + const, \\ elbo_{\lambda} &= (\mathbb{E}_q[\Phi_g(\mathbf{z}, \mathbf{x}, \eta)] - \Phi_g(\lambda)) \cdot \nabla A_g(\phi_g(\lambda)) + A_g(\Phi(\lambda)) + const. \end{aligned}$$

Finally, finding the first derivative of the *elbo* with respect to a coordinate reveals its parameter update:

$$\nabla_{\phi_n} elbo_{\phi_n} = \nabla^2_{\phi_n} A_l(\Phi_l(\phi_n))(\mathbb{E}_q[\Phi_l(\beta, x_n, \alpha)] - \Phi_l(\phi_n)) \tag{15}$$

$$\nabla_{\lambda} elbo_{\lambda} = \nabla^2_{\lambda} A_g(\Phi_g(\lambda))(\mathbb{E}_q[\Phi_g(\mathbf{z}, \mathbf{x}, \eta)] - \Phi_g(\lambda)). \tag{16}$$

Equating the first derivative to zero shows that the update for the local (resp. global) context is equivalent to setting the natural parameter of equation 13 (resp. 14) to the expected natural parameter of equation 11 (resp. 12). Therefore the update for the local and global variational parameters are simply the following:

$$\begin{aligned} \Phi_l(\phi_n) &= \mathbb{E}_q[\Phi_l(\beta, x_n, \alpha)], \\ \Phi_g(\lambda) &= \mathbb{E}_q[\Phi_g(\mathbf{z}, \mathbf{x}, \eta)] \end{aligned}$$

The literature presents this as an update of the variational parameter (Hoffman et al. 2013), not requiring that the update is for the natural variational parameter. This is sufficient for all practical purposes, however without it, the statement is false. The local update can be done independent of the global update and of each other. It follows to update the global variational parameter, λ, when the local variational parameters, $\boldsymbol{\phi}$, converge, and repeat until the *elbo* converges. The pseudocode for coordinate ascent variational inference is shown in Algorithm 2.

Algorithm 2 Coordinate Ascent Variational Inference

Initialise $\lambda^{(0)}$
repeat
    repeat
        for each ϕ_n **do**
            Update $\Phi_l(\phi_n) = \mathbb{E}_q[\Phi_l(\beta, x_n, \alpha)]$
    until $\boldsymbol{\phi}$ converges
    Update $\Phi_g(\lambda) = \mathbb{E}_q[\Phi_g(\mathbf{z}, \mathbf{x}, \eta)]$
until *elbo* converges

5.3 Stochastic Variational Inference

Stochastic variational inference is the scalable counterpart of variational inference. It provides a method of sub-sampling noisy estimates of the natural gradient of the elbo. This method is scalable to large corpora.

The coordinate ascent variational inference algorithm requires randomly initialising λ^0 and iterating through every word of the corpus before updating the global parameter λ for the first time. This makes it inefficient for large corpora (Hoffman et al. 2013). However, this inefficiency is solved by considering the stochastic optimisation method of sub-sampling from a noisy estimate of the gradient (Robbins et al. 1951). This method updates the global parameter λ without having to iterate through all the words of the corpus.

The problem with this is that the gradient points in the direction of steepest ascent in the standard parameter space. This is an issue because closeness in the standard parameter space for parameters $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ is measured by the euclidean distance metric:

$$||\mathbf{a} - \mathbf{b}|| = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}.$$

The closeness of the parameters does not correspond to the closeness of the distributions which they are parameters for.

Natural Gradients

Fortunately, the similarity between probability distributions is measured by the relative entropy, as seen in equation 8. Using a Riemannian metric whereby the closeness between parameters is calculated by the symmetric relative entropy (Kullback et al. 1951),

$$D_{KL}^{sym}(p(\mathbf{x}), q(\mathbf{x})) = D_{KL}^{sym}(q(\mathbf{x}), p(\mathbf{x})) = D_{KL}(p(\mathbf{x})||q(\mathbf{x})) + D_{KL}(q(\mathbf{x})||p(\mathbf{x})), \quad (17)$$

a Riemannian paramater space can be defined which accounts for the information geometry of the parameters. In this space, the direction of steepest ascent is that of the natural gradient.

Computing the natural gradient turns out to be simpler than the standard counterpart for stochastic optimisation. The natural gradient of can be found by premultiplying the gradient by the inverse Riemannian metric $G(\phi_n)$ and $G(\lambda)^{-1}$ for the local and global contexts respectively (Amari 1998). G for the local and global contexts are the Fisher information matrix of $q(z_n; \phi_n)$ and $q(\beta; \lambda)$ respectively. (Kullback et al. 1951):

$$G(\phi_n) = \mathbb{E}_q[\nabla_{\phi_n} \ln q(z_n; \phi_n) \cdot \nabla_{\phi_n} \ln q(z_n; \phi_n)] = \nabla^2_{\phi_n} A_l(\Phi_l(\phi_n)),$$
$$G(\lambda) = \mathbb{E}_q[\nabla_{\lambda} \ln q(\beta; \lambda) \cdot \nabla_{\lambda} \ln q(\beta; \lambda)] = \nabla^2_{\lambda} A_g(\Phi_g(\lambda)),$$

where this simplification is by virtue of the probability distributions being in the exponential family (Hoffman et al. 2013). It follows that pre-multiplying this by the gradient of the *elbo* in equations 15 and 16 gives a straightforward natural gradient of

the *elbo*:

$$\nabla_{\phi_n}^{nat} = \mathbb{E}_q[\Phi_l(\beta, x_n, \alpha)] - \Phi_l(\phi_n),$$
$$\nabla_{\lambda}^{nat} = \mathbb{E}_q[\Phi_g(\mathbf{z}, \mathbf{x}, \eta)] - \Phi_g(\lambda).$$

In order to determine the natural gradient, uniform randomly chosen data point (or subset of dataset) is selected and the local updates calculated. The global parameter is updated as though the sampled point(s) was N points. This provides the noisy natural gradient (Blei et al. 2017).

Updating Parameters

Stochastic variational inference approaches variational inference as a stochastic optimisation. This means that once the direction of the noisy natural gradient is found, steps in the direction of the natural gradient is taken. These noisy gradients are seen as approximates of the true natural gradient which after sufficiently many noisy gradients taken in the right proportions, converges to the local optimum.

These right proportions are known as the step sizes in the direction of the noisy natural gradient. It is taken from the theory of stochastic optimisation which states that in order for a stochastic optimisation to converge to a local optima, the step sizes, ρ_t, must satisfy the following criteria (Robbins et al. 1951):

$$\sum_t \rho_t = \infty,$$
$$\sum_t \rho_t^2 < \infty.$$

These criteria are satisfied by setting:

$$\rho_t = (t + \tau)^{-\kappa},$$

where $\tau \geq 0$ and $\kappa \in (0.5, 1]$ (Hoffman et al. 2013).

The overall pseudocode is shown in Algorithm

Algorithm 3 Stochastic Variational Inference

Initialise $\lambda^{(0)}$
repeat
    Sample S data points
    for each of the sampled data point(s) **do**
        Compute $\Phi_l(\phi_s) = \mathbb{E}_q[\Phi_l(\beta, x_s, \alpha)]$
    Compute $\Phi_g(\lambda^{(int)}) = \mathbb{E}_q[\Phi_g(z_s^{(\frac{N}{S})}, x_s^{(\frac{N}{S})}, \eta)]$
    Update $\Phi_g(\lambda^{(t)}) = (1 - \rho_t)\Phi_g(\lambda^{(t-1)}) + \rho_t \Phi_g(\lambda^{(int)})$
until forever

6 Implementation

Chapter 4 described a probabilistic machine learning approach to topic modelling, known as LDA. However, the problem with LDA is that the posterior, $p(\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{z}|\mathbf{w}; \alpha, \eta)$, is intractable by virtue of its intractable evidence seen in equation 6. Chapter 5 elaborates on several inference methods for approximating the posterior; namely, MCMC, Variational Inference and Stochastic Variational Inference. This chapter implements these algorithms in the context of LDA to compare and contrast the advantages/disadvantages of the different methods.

6.1 Data

LDA on its own is just a model. It is a model that needs to be applied to some context to generate meaningful results. This is added to the model through the observed variables. The observed variables for LDA, also referred to as the data, are the words of every document of a corpus. In this chapter, the data used for testing comes from thepangean.com. The corpus consists of 244 articles with a vocabulary of 21085 words after pre-processing. The data is provided by the founding partners of *The Pangean*, who have agreed for the data to be used in this project.

The raw data (The Pangean 2020) consists of 244 files. Each file has a header with the details of the article (i.e. title, author, date) followed by the body of content which is formatted for markdown with the occasional HTML tag. The pre-processing stage involved compiling the articles into a `pandas` dataframe with separate columns for the details in the header and a column for the body of text. The text was then parsed using regular expressions to remove all html tags, blank spaces, punctuation and all words shorter than four letters. Additionally, a total of 179 most commonly used words in the English dictionary were removed. The processed text was then added as a new column to the dataframe.

The dataset was partitioned into training and test data with a 9:1 split. The posterior of the LDA model is found with respect to the training data using any of the inference methods described in Chapter 5. Subsequently the model is tested using the average log predictive probability which is calculated by dividing each test document into observed words, w_{obs}, and held out words, w_{ho}. The idea is to find the probability of the held out words given the model learnt from the training data and the observed words of the test document, $p(w_{ho}|w_{obs}, \mathbf{D})$. This is calculated by the following (Hoffman et al. 2013):

$$\begin{aligned} p(w_{ho}|w_{obs}, \mathbf{D}) &= \int\int (\sum_{k=1}^{K} \theta_k \beta_{k,w_{ho}}) p(\theta|w_{obs}, \beta) p(\beta|\mathbf{D}) d\theta d\beta \\ &= \sum_{k=1}^{K} \mathbb{E}_p[\theta_k] \mathbb{E}_p[\beta_{k,w_{ho}}], \end{aligned}$$

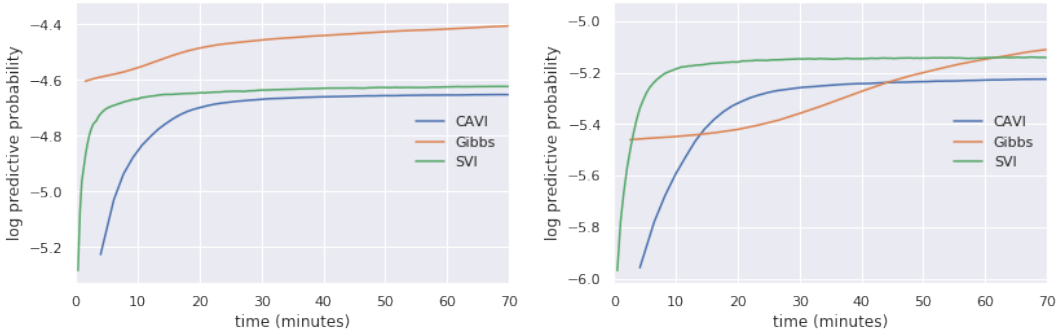
and in the case of variatonal inference and stochastic variational inference:

$$p(w_{ho}|w_{obs}, \mathbf{D}) \approx \sum_{k=1}^{K} \mathbb{E}_q[\theta_k]\mathbb{E}_q[\beta_{k,w_{ho}}].$$

6.2 Results

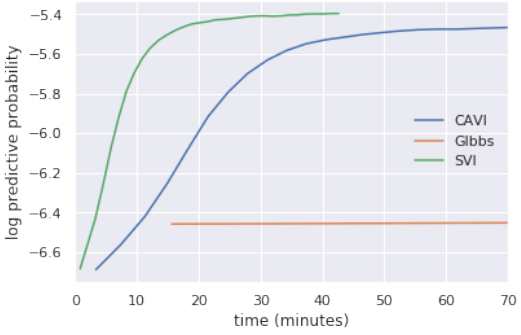
The results in this section are from inference using the three methods discussed in Chapter 5; Gibbs sampling, variational inference and stochastic variational inference. For each of these methods a uniform prior is used. This is achieved by setting $\alpha = \eta = 1$. Results for $K = 3, 10$ and 50 are compared. For stochastic variational inference the hyperparameters are set to the folllowing: $\kappa = 0.9$ and $\tau = 1$.

The corpus of articles from `thepangean.com` is artificially inflated by duplicating each article 10 times. This increases the size of the corpus to 2440 articles but still maintains the size of the vocabulary at 21085 words. This is done to test the limitations and advantages of the different methods of inference. The data is split into a 2196 training articles and 244 test articles.



(a) LDA model with $K = 3$.

(b) LDA model with $K = 10$.



(c) LDA model with $K = 50$.

Figure 8: The average log predictive probability of LDA for a different number of topics and different inference methods.

The results are shown in Figure 8. For $K = 3$, the Gibbs sampler immediately outperformed both variational inference and stochastic variational inference. For $K = 10$, the Gibbs sampler initially outperforms variational inference which improves much more quickly and pass Gibbs. However, it reaches a local optima and plateaus. Gibbs

continues to get better slowly. On the other hand stochastic variational inference very quickly gets good results, by 500 seconds it is already outperforming Gibbs at 3000 seconds. For $K = 50$ the shortcomings of Gibbs are vere prevalent.

The trend follows that as the number of topic K increases, Gibbs fails to perform whereas the variational methods maintain effectiveness. In particular, stochastic variational inference always finds a local optima which is slightly better than the normal variational inference, at a quicker rate. Therefore, variational inference never outperforms stochastic variational inference.

7 Conclusion and Future Works

MCMC proves to be the most accurate approximation of the posterior. However, it doesn't hold up to an increasing number of parameters to sample from and its runtime exponentially increases. Variational inference in outperformed by stochastic variational inference on all aspects. Therefore, variational inference is obsolete. Stochastic variational inference scales extremely well to massive datasets. It's problem stems from the mean field variational family. The approximation of the algorithm is limited by how close the mean field family is to the original model. As a result, stochastic variational inference will not perform well on complex models.

An area of open research is the study of the types of families of distributions that can be applied to stochastic variational inference. Finding a family of models which doesn't relax too many dependence assumptions of the original model would serve to better approximate the posterior.

Stochastic variational inference has not been widely applied to probabilistic models. Future works would also involved deriving the parameter update in cases where the complete conditional assumptions are not satisfied. This would provide a wider scope of models which this algorithm could be applied to.

A Probability Distributions

A.1 Categorical Distribution

This is the generalisation of the Bernoulli to a finite number of values or categories, denoted k. It is a discrete distribution which has support as the set of vectors $\chi = \{(x_1, x_2, ..., x_k) : \sum_{i=1}^{k} x_i = 1, x_i \in \{0, 1\}\}$ representing the category that was picked. The category probabilities are represented by parameters $p_1, ..., p_k$ such that $p_i \geq 0$ and $\sum_{i=1}^{k} p_i = 1$, which have a mass function defined as:

$$p(\mathbf{x}; \mathbf{p}) = p_1^{x_1} p_2^{x_2} ... p_k^{x_k}.$$

The expectation and variance of each category of such a probability distribution are:

$$\begin{aligned} \mathbb{E}[X_i] &= p_i \\ Var(X_i) &= p_i(1 - p_i). \end{aligned}$$

A.2 Multinomial Distribution

This is a generalisation to n samples of a categorical distribution for k categories. Each sample can take any of the k categories and so the support is a vector representing the number of samples in which each category was picked; $\chi = \{x_1, x_2, ..., x_k\}$ where $\sum_i x_i = n$. Similarly to the categorical distribution, the parameters are the probabilities $p_1, ..., p_k$ such that $p_i \geq 0$ and $\sum_{i=1}^{k} p_i = 1$, however, the probability mass function is defined as:

$$p(\mathbf{x}; n, \mathbf{p}) = \frac{n!}{x_1! x_2! ... x_k!} p_1^{x_1} p_2^{x_2} ... p_k^{x_k}.$$

The expectation and variance of each category over n samples are:

$$\begin{aligned} \mathbb{E}[X_i] &= np_i \\ Var(X_i) &= np_i(1 - p_i). \end{aligned}$$

A.3 Dirichlet Distribution

This is is a generalisation of the beta distribution where the stick is broken into $k \geq 2$ pieces. This continuous distribution has support $\chi = \{\mathbf{x} = (x_1, ..., x_k) : x_i \geq 0, \sum_i x_i = 1\}$. It is important to note that this is the set of possible parameters for a categorical distribution. Its parameters are $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ..., \alpha_k)$ where $\alpha_i > 0$ with a probability density function defined as:

$$p(\mathbf{x}; k, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(\sum_i \alpha_i)}{\prod_i \Gamma(\alpha_i)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}$$

where $\Gamma(n) = (n-1)!$. The expectation and variance - we include the expectation for the natural log of the random variable as it will be useful later - of such a probability

distribution are:

$$\mathbb{E}[X_i] = \frac{\alpha_i}{\sum_j \alpha_j}$$
$$Var(X_i) = \frac{\mathbb{E}[X_i](1 - \mathbb{E}[X_i])}{\sum_j \alpha_j + 1}$$

Similarly to the beta distribution, the Dirichlet distribution is extremely versatile. It is often uses as a prior for a categorical distribution, since samples from a Dirichlet distribution can be used an the parameters for a categorical distribution. The Dirichlet parameters are often set to one, $\alpha_i = 1$, to use as a uniform prior.

A.4 Exponential Family

The multinomial and Dirichlet distributions are generalised distributions. In the case of the multinomial, setting either $n = 1$ or both $n = 1$ and $k = 2$ will retrieve the categorical or Bernoulli distributions respectively. Whereas setting $k = 2$ in the Dirichlet distribution will retrieve the beta distribution. Therefore we consider only these generalised distributions. Their probability distribution functions, in this form, are at first glance, fairly different, especially seeing as they are a discrete and continuous distribution respectively. However, they are very similar.

A.4.1 Multinomial

We consider the fact that, for a fixed n, $x_k = n - \sum_{i=1}^{k-1} x_i$ and $p_k = 1 - \sum_{i=1}^{k-1} p_i$. It then follows:

$$
\begin{aligned}
p(\mathbf{x}; n, \mathbf{p}) &= \frac{n!}{x_1!...x_k!} \prod_{i=1}^{k} p_i^{x_i} \\
&= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!} \exp\{\ln \prod_{i=1}^{k} p_i^{x_i}\} \\
&= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!} \exp\{\sum_{i=1}^{k} x_i \ln p_i\} \\
&= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!} \exp\{\sum_{i=1}^{k-1} x_i \ln p_i + x_k \ln p_k\} \\
&= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!} \exp\{\sum_{i=1}^{k-1} x_i \ln p_i + (n - \sum_{i=1}^{k-1} x_i) \ln(1 - \sum_{i=1}^{k-1} p_i)\} \\
&= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!} \exp\{\sum_{i=1}^{k-1} x_i \ln p_i + n \ln(1 - \sum_{i=1}^{k-1} p_i) - \sum_{i=1}^{k-1} x_i \ln(1 - \sum_{i=1}^{k-1} p_i)\} \\
&= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!} \exp\{\sum_{i=1}^{k-1} x_i(\ln p_i - \ln(1 - \sum_{i=1}^{k-1} p_i)) + n \ln(1 - \sum_{i=1}^{k-1} p_i)\} \\
&= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!} \exp\{\sum_{i=1}^{k-1} x_i \ln \frac{p_i}{1 - \sum_{i=1}^{k-1} p_i} + n \ln(1 - \sum_{i=1}^{k-1} p_i)\}
\end{aligned}
$$

A.4.2 Dirichlet

$$
\begin{aligned}
p(\mathbf{x}; k, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} x_i^{\alpha_i - 1} \\
&= \exp\{\ln \frac{\Gamma(\sum_{i=1}^{k} \alpha_i)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} x_i^{\alpha_i - 1}\} \\
&= \exp\{\ln \Gamma(\sum_{i=1}^{k} \alpha_i) - \ln \prod_{i=1}^{k} \Gamma(\alpha_i) + \ln \prod_{i=1}^{k} x_i^{\alpha_i - 1}\} \\
&= \exp\{\ln \Gamma(\sum_{i=1}^{k} \alpha_i) - \sum_{i=1}^{k} \ln \Gamma(\alpha_i) + \sum_{i=1}^{k} (\alpha_i - 1) \ln x_i\} \\
&= \exp\{\sum_{i=1}^{k} (\alpha_i - 1) \ln x_i - [\sum_{i=1}^{k} \ln \Gamma(\alpha_i) - \ln \Gamma(\sum_{i=1}^{k} \alpha_i)]\}
\end{aligned}
$$

A.4.3 Exponential Form

It may be difficult to notice it immediately, but there is a nice elegance to this way of representing the probability distribution functions. Both these distributions, and by extension all the distributions they generalise, fall under the exponential family of distributions (Jordan 2009). This class of distributions can be written in the following general form:

$$p(\mathbf{x};\boldsymbol{\theta}) = h(\mathbf{x})exp\{\Phi(\boldsymbol{\theta}) \cdot T(\mathbf{x}) - A(\Phi(\boldsymbol{\theta})\}$$

where $h(.)$ and $A(.)$ are scalar functions; on the other hand, $\Phi(.)$ and $T(.)$ are vector function. All these functions are known for a particular distribution of the family. The function $h(.)$ represents a simple *base measure* and $T(.)$ is the *sufficient statistics*. The sufficient statistics represent the necessary information needed to estimate the parameters. $A(.)$ represents the *log normaliser*. As its name implies, it is the logarithm of the normalising constant of the probability distribution; it asserts that the sum of all the probabilities of the distribution is one and is defined as the following:

$$A(\Phi(\boldsymbol{\theta})) = log \int h(\mathbf{x}) \exp\{\Phi(\boldsymbol{\theta}) \cdot T(\mathbf{x})\} dx.$$

$\Phi(.)$ is slightly more complicated but very important to note. Put simply, it ensures that the normalising constant of the distribution doesn't give undefined probabilities. Formally, it is a mapping of the parameters to the natural parameter space, which is the set of parameters where the normalising constant of the distribution is finite, $\mathcal{N} = \{\boldsymbol{\theta} : \exp\{A(\Phi(\boldsymbol{\theta}))\} < \infty$.

The elegance of this family of distributions comes from the unassumingly simple calculations when quantifying their typical behaviour. The expectation and variance turn out to be the first and second derivative of the log normaliser respectively.

$$\begin{aligned}
\frac{dA}{d\boldsymbol{\theta}^T} &= \frac{\int T(\mathbf{x}) \exp\{\Phi(\boldsymbol{\theta}) \cdot T(\mathbf{x})\} h(x) dx}{\int \exp\{\Phi(\boldsymbol{\theta}) \cdot T(\mathbf{x})\} h(x) dx} \\
&= \int T(\mathbf{x}) \exp\{\Phi(\boldsymbol{\theta}) \cdot T(\mathbf{x}) - A(\Phi(\boldsymbol{\theta}))\} \\
&= \mathbb{E}[T(\mathbf{x})].
\end{aligned}$$

This distribution is quite difficult to digest, however, as a re-iteration, it is very elegant. Coming back to our multinomial and Dirichlet distributions. Their re-written forms exactly match the form of the exponential family, this means they are part of the exponential family. This fact is extremely important for this project as it simplifies many of the calculations that need to be done when deriving *Variational Inference* and *Stochastic Variational Inference*.

Multinomial:

$$
\begin{aligned}
h(\mathbf{x}) &= \frac{(\sum_{i=1}^{k} x_i)!}{\prod_{i=1}^{k} x_i!}, \\
\Phi(\theta_i) &= \ln \frac{p_i}{1 - \sum_{i=1}^{k-1} p_i}, \\
T(x_i) &= x_i, \\
A(\Phi(\boldsymbol{\theta})) &= -n \ln(1 - \sum_{i=1}^{k-1} p_i) = n \ln(1 + \sum_{i=1}^{k-1} \exp\{\Phi(\theta_i)\}), \\
E[T(x_i)] &= \frac{dA}{d\theta_i} = np_i
\end{aligned}
$$

Dirichlet:

$$
\begin{aligned}
h(\mathbf{x}) &= 1, \\
\Phi(\theta_i) &= \alpha_i - 1, \\
T(x_i) &= \ln x_i, \\
A(\Phi(\boldsymbol{\theta})) &= \sum_{i=1}^{k} \ln \Gamma(\Phi(\theta_i) + 1) - \ln \Gamma(\sum_{i=1}^{k} \Phi(\theta_i) + 1) \\
E[T(x_i)] &= \frac{dA}{d\theta_i} = \psi(\alpha_i) - \psi(\sum_{j} \alpha_j)
\end{aligned}
$$

where $\psi(n)$ is the digamma function which is defined as the logarithmic derivative of the gamma function; $\frac{d}{dx} \ln(\Gamma(n))$.

B Latent Dirichlet Allocation

Each of the inference methods work very closely with the complete conditionals of the hidden variables of the model. The complete conditionals of the LDA model are the following:

$$
\begin{aligned}
p(z_{d,n} = k | \beta_k, \theta_{dk}, w) &\propto \exp(\log \theta_{dk} + \log \beta_{k,w_{d,n}}), \\
p(\theta_d | z_d; \alpha) &= Dir_K(\alpha + \sum_{n=1}^{N_d} z_{d,n}), \\
p(\beta_k | \mathbf{z}, \mathbf{w}; \eta) &= Dir_V(\eta + \sum_{d=1}^{D} \sum_{n=1}^{N_d} \mathbb{I}\{z_{d,n} = k\} w_{d,n})
\end{aligned}
$$

References

Adam, Ahmat (1991). 'Portuguese Words in the Malay Language'. In: *International Seminar on Silk Roads: Roads of Dialogue, UNESCO*.

Amari, Shun-Ichi (1998). 'Natural gradient works efficiently in learning'. In: *Neural computation* 10.2, pp. 251–276.

Bahl, Lalit R, Frederick Jelinek and Robert L Mercer (1983). 'A maximum likelihood approach to continuous speech recognition'. In: *IEEE transactions on pattern analysis and machine intelligence* 2, pp. 179–190.

Bigelow, Bill and Bob Peterson (1998). *Rethinking Columbus: the next 500 years*. Rethinking Schools.

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer.

Blei, David M, Alp Kucukelbir and Jon D McAuliffe (2017). 'Variational inference: A review for statisticians'. In: *Journal of the American statistical Association* 112.518, pp. 859–877.

Blei, David M and John D Lafferty (2006). 'Dynamic topic models'. In: *Proceedings of the 23rd international conference on Machine learning*, pp. 113–120.

Blei, David M, Andrew Y Ng and Michael I Jordan (2003). 'Latent dirichlet allocation'. In: *Journal of machine Learning research* 3.Jan, pp. 993–1022.

Bloom, Paul (2002). *How children learn the meanings of words*. MIT press.

Colby, Kenneth Mark, Sylvia Weber and Franklin Dennis Hilf (1971). 'Artificial paranoia'. In: *Artificial Intelligence* 2.1, pp. 1–25.

Colby, Kenneth Mark et al. (1972). 'Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes'. In: *Artificial Intelligence* 3, pp. 199–221.

Cui, Yiming et al. (2019). 'Pre-training with whole word masking for chinese bert'. In: *arXiv preprint arXiv:1906.08101*.

Dahl, Veronica (1994). 'Natural language processing and logic programming'. In: *The Journal of Logic Programming* 19, pp. 681–714.

Dale, Andrew I (2012). *A history of inverse probability: From Thomas Bayes to Karl Pearson*. Springer Science & Business Media.

Dechter, Rina (1986). *Learning while searching in constraint-satisfaction problems*. University of California, Computer Science Department, Cognitive Systems ...

Devlin, Jacob et al. (2018). 'Bert: Pre-training of deep bidirectional transformers for language understanding'. In: *arXiv preprint arXiv:1810.04805*.

Diaconis, Persi, Donald Ylvisaker et al. (1979). 'Conjugate priors for exponential families'. In: *The Annals of statistics* 7.2, pp. 269–281.

Dronkers, Nina F et al. (2007). 'Paul Broca's historic cases: high resolution MR imaging of the brains of Leborgne and Lelong'. In: *Brain* 130.5, pp. 1432–1441.

Dumais, Susan T et al. (1988). 'Using latent semantic analysis to improve access to textual information'. In: *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 281–285.

Duriez, Isabelle (2009). 'Seriously speaking: what is Ch'ti?' In: *The UNESCO Courier* 2, p. 8.

Evans, James (1984). 'On the function and the probable origin of Ptolemy's equant'. In: *American journal of physics* 52.12, pp. 1080–1089.

Gadgil, Madhav et al. (1998). 'Peopling of India'. In: *The Indian human heritage*, pp. 100–129.

Gamerman, Dani and Hedibert F Lopes (2006). *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. CRC Press.

Geman, Stuart and Donald Geman (1984). 'Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images'. In: *IEEE Transactions on pattern analysis and machine intelligence* 6, pp. 721–741.

Gilks, Walter R (2005). 'Markov Chain Monte Carlo'. In: *Encyclopedia of biostatistics* 4.

Gilks, Walter R, Sylvia Richardson and David J Spiegelhalter (1996). 'Introducing markov chain monte carlo'. In: *Markov chain Monte Carlo in practice* 1, p. 19.

Gillick, Laurence et al. (Feb. 1990). *Method for representing word models for use in speech recognition*. US Patent 4,903,305.

Griffiths, Thomas L et al. (2004). 'Hierarchical topic models and the nested chinese restaurant process'. In: *Advances in neural information processing systems*, pp. 17–24.

Grinstead, Charles Miller and James Laurie Snell (2012). *Introduction to probability*. American Mathematical Soc.

Hastings, W Keith (1970). 'Monte Carlo sampling methods using Markov chains and their applications'. In:

Haykin, Simon (1994). *Neural networks: a comprehensive foundation*. Prentice Hall PTR.

Hoffman, Thomas (1999). 'Probabilistic latent semantic analysis'. In: *proc. of the 15th Conference on Uncertainty in AI, 1999*.

Hoffman et al. (2013). 'Stochastic variational inference'. In: *The Journal of Machine Learning Research* 14.1, pp. 1303–1347.

Jaynes, Edwin T (2003). *Probability theory: The logic of science*. Cambridge university press.

Jia, Robin and Percy Liang (2017). 'Adversarial examples for evaluating reading comprehension systems'. In: *arXiv preprint arXiv:1707.07328*.

Jordan, Michael I (2009). *The Exponential Family: Basics*. URL: `https://people.eecs.berkeley.edu/~jordan/courses/260-spring10/other-readings/chapter8.pdf`.

Krizhevsky, Alex, Ilya Sutskever and Geoffrey E Hinton (2012). 'Imagenet classification with deep convolutional neural networks'. In: *Advances in neural information processing systems*, pp. 1097–1105.

Kullback, Solomon and Richard A Leibler (1951). 'On information and sufficiency'. In: *The annals of mathematical statistics* 22.1, pp. 79–86.

Laplace, Pierre-Simon (1814). 'Essai philosophique sur les probabilités (1814)'. In: *Printed as a preface to Théorie analytique des probabilités in the Oeuvres Complètes edition* 7.

LeCun, Yann, Yoshua Bengio and Geoffrey Hinton (2015). 'Deep learning'. In: *nature* 521.7553, pp. 436–444.

Leibniz, Gottfried Wilhelm and Gottfried Wilhelm Freiherr von Leibniz (1996). *Leibniz: New essays on human understanding*. Cambridge University Press.

Li, Fang Kuei (1973). 'Languages and dialects of China'. In: *Journal of Chinese Linguistics*, pp. 1–13.

Litjens, Geert et al. (2017). 'A survey on deep learning in medical image analysis'. In: *Medical image analysis* 42, pp. 60–88.

MacKay, David JC and David JC Mac Kay (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Marcus, Gary (2018). 'Deep learning: A critical appraisal'. In: *arXiv preprint arXiv:1801.00631*.

Mnih, Volodymyr et al. (2013). 'Playing atari with deep reinforcement learning'. In: *arXiv preprint arXiv:1312.5602*.

Moulines, Eric and Francis Charpentier (1990). 'Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones'. In: *Speech communication* 9.5-6, pp. 453–467.

Neeley, Tsedal (2012). 'Global business speaks English'. In: *Harvard business review* 90.5, pp. 116–124.

Noam Chomsky, Al Page (1991). 'The Concept of Language'. In: *University of Washington TV*. URL: https://www.youtube.com/watch?v=hdUbIlwHRkY.

Plecháč, Petr (2019). 'Relative contributions of Shakespeare and Fletcher in Henry VIII: An Analysis Based on Most Frequent Words and Most Frequent Rhythmic Patterns'. In: *arXiv preprint arXiv:1911.05652*.

Robbins, Herbert and Sutton Monro (1951). 'A stochastic approximation method'. In: *The annals of mathematical statistics*, pp. 400–407.

Robertson, John S (1992). *The history of tense/aspect/mood/voice in the Mayan verbal complex*. University of Texas Press.

Seide, Frank, Gang Li and Dong Yu (2011). 'Conversational speech transcription using context-dependent deep neural networks'. In: *Twelfth annual conference of the international speech communication association*.

Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Shannon, Claude E (1948). 'A mathematical theory of communication'. In: *Bell system technical journal* 27.3, pp. 379–423.

Strubell, Emma, Ananya Ganesh and Andrew McCallum (2019). 'Energy and policy considerations for deep learning in NLP'. In: *arXiv preprint arXiv:1906.02243*.

Teh, Yee W et al. (2005). 'Sharing clusters among related groups: Hierarchical Dirichlet processes'. In: *Advances in neural information processing systems*, pp. 1385–1392.

The Pangean (2020). *Dataset: Corpus of Articles*. [Online; accessed 16-Apr-2020]. URL: https://github.com/thepangean/thepangean.github.io/tree/dev/_posts.

Toomer, Gerald J (1998). *Ptolemy's almagest*. Princeton University Press.

Turing, A. M. (1950). 'Computing Machinery and Intelligence'. In: *Mind* LIX.236, pp. 433–460.

Wainwright, Martin J and Michael I Jordan (2008). 'Graphical models, exponential families, and variational inference'. In: *Foundations and Trends® in Machine Learning* 1.1-2, pp. 1–305.

Wallach, Hanna M (2006). 'Topic modeling: beyond bag-of-words'. In: *Proceedings of the 23rd international conference on Machine learning*, pp. 977–984.

Weizenbaum, Joseph (1966). 'ELIZA—a computer program for the study of natural language communication between man and machine'. In: *Communications of the ACM* 9.1, pp. 36–45.

Wernicke, Carl (1874). *Der aphasische Symptomencomplex: eine psychologische Studie auf anatomischer Basis*. Cohn.

Williams, Eric (2020). *From Columbus to Castro: the history of the Caribbean*. Lulu Press, Inc.

Wolfe, Patricia M (1972). *Linguistic change and the great vowel shift in English*. Univ of California Press.

Zen, Heiga, Keiichi Tokuda and Alan W Black (2009). 'Statistical parametric speech synthesis'. In: *speech communication* 51.11, pp. 1039–1064.

Zhang, Zhengyan et al. (2019). 'ERNIE: Enhanced language representation with informative entities'. In: *arXiv preprint arXiv:1905.07129*.