University of Glasgow | School of Computing Science

Infinite Vocabulary Topic Modelling on Metabolomics Data

Joe Frew
2089249f

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

Level 4 Project — March 21, 2017

Abstract

Metabolomics is the study of the detection, identification and analysis of light weight molecules which comprise the metabolism [16]. A major issue within this field is the lack of tools and labelled datasets which are required to extract useful biochemically relevant information from samples using mass spectrometry. For this reason the MS2LDA tool was created which adapts a machine learning algorithm called Latent Dirichlet Allocation (LDA), originally designed for discovering topics in text corpora, for use in metabolomics. MS2LDA examines mass spectrometry data and allows for the identification of common molecular substructures based on the co-occurrence of mass fragments between different molecules. One limitation of the tool is that input data currently undergoes a preprocessing phase to eliminate noise, which may also result in the elimination of some relevant data. This eliminated data could be used to differentiate between molecule fragments of very similar masses.

This report demonstrates a proof of concept extension of the LDA model with an "infinite vocabulary" of molecule fragments, along with a set of visualisation tools which can be used by researchers to evaluate the model. A summary of the implementation is provided, along with the presentation and evaluation of two experiments — one run on synthetic data, and another on a portion of real world mass spectrometry data. The hope is that this extension will allow the system to uncover some additional insights into underlying biochemical processes which may previously have been obscured due to information loss.

Acknowledgements

I would like to thank my supervisor, Dr Simon Rogers, for his generosity with his time, encouragement and for providing the guidance that I needed when attempting to understand the very tip of the metabolomics iceberg.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: _____ Signature: _____

Contents

Chapter 1

Introduction

1.1 Aims

This project aims to implement a proof of concept variation on the LDA model, used in MS2LDA, which will feature an infinite vocabulary of molecule fragments. The implementation should include a set of tools to aid in the running of experiments, and the subsequent evaluation of the model's ability to extract useful biochemically relevant information from the input data.

The advantages of the infinite vocabulary model are expected to be two-fold. Firstly, the algorithm may be able to identify and group identical molecule fragments which have different observed masses (due to noise in the instrumentation) without requiring a preprocessing phase. Secondly, the current implementation eliminates some useful information by grouping observed fragments of similar mass into a single "vocabulary item". The infinite vocabulary extension may allow the algorithm to detect that, although they have the same mass, these observations are in fact different, meaning they will be treated as two distinct groups rather than one. This can be achieved by observing that they regularly co-occur with different fragments in other fragmentation spectra. The algorithm could determine that two observed fragment masses which were previously assumed to be noisy readings, are actually different fragments.

1.2 Motivation

1.2.1 MS2LDA

A major issue in the field of metabolomics is the lack of both tools and comprehensive labelled data, in the form of reference spectra, which are required to extract useful biochemical information [18]. Additionally, much of the data produced in metabolomics experiments is not leveraged to the fullest extent since most current tools compare spectra from individual molecules to reference spectra, rather than sharing information between fragmentation spectra.

The development of MS2LDA was motivated by the current lack of tools and datasets and the desire to extract key biochemically relevant data which could be used to further the life sciences. The choice of unsupervised learning was motivated in part due to hand identification of metabolites being very labour intensive [12]. Unsupervised learning is highly advantageous since it eliminates the need for human intervention. This is achieved by comparing fragmentation spectra from many molecules in order to uncover shared structure which can later be analysed and labelled by users with expert knowledge.

1.2.2 Infinite Vocabulary HDP

As mentioned in the aims section, the preprocessing step in MS2LDA eliminates some data which may be useful when differentiating between very similar molecule fragments. This project is motivated by the effort to determine whether an infinite vocabulary model would be able to utilise the raw data to produce additional biochemically relevant insights.

Some of the advantages of infinite vocabulary HDP are discussed in greater detail in section 3.2.2.

Chapter 2

Background

2.1 Mass Spectrometry

Mass spectrometry is a technique used to characterise chemical species by observing their mass to charge ratio (*m/z*) [18]. A mass spectrometer typically operates by bombarding molecules with electrons in order to ionise them. The ionised molecules are then accelerated into an electric or magnetic field, which causes them to deflect by different amounts depending on their mass-to-charge ratio. Molecules with similar *m/z* will be deflected the same amount and will be incident on the detector in the same location. By plotting the frequency with which particles hit different parts of the detector, a mass spectrum can be constructed which shows the abundance of particles with different mass-to-charge ratios. Ionisation is necessary since particles must be charged in order to be deflected by a magnetic or electric field.

Organic compounds are often difficult to characterise in a mass spectrometer using their *m/z* alone [20]. One technique used to overcome this is to fragment the molecule during ionisation, before it is accelerated in the magnetic field [21]. Using this method, the resulting spectra is of fragments rather than the original molecules. Fragmentation spectra are traditionally used to characterise the molecule by comparing the spectra to a set of reference spectra [20] in order to identify the fragments. Unfortunately databases of reference spectra are often very incomplete [18]. To circumvent this issue, MS2LDA identifies shared structure between molecules by identifying similarities in the fragments they break into. This allows the LDA algorithm to uncover useful insights about the molecules without having to identify molecules or fragments using reference spectra [20].

2.1.1 Noise in Mass Spectrometry Data

One of the challenges faced by mass spectrometrists is that mass spectrometers produce inherently noisy data [9, 18]. Identical molecules or fragments can produce slightly different *m/z* readings within the mass spectrometer, or false data points can be detected where there are none. A common technique to remedy this issue, which is used currently in MS2LDA, is to preprocess the dataset by grouping observations of very similar *m/z* values together. Although this often works, it results in the inevitable loss of some information about the underlying spectra. A situation could arise where two fragments are so similar in *m/z* that noise could cause the readings of each of the fragments to overlap. When this happens, a preprocessing phase may erroneously group these two distinct fragments into a single *m/z* reading. By avoiding the preprocessing step to group fragments, and allowing the LDA algorithm to do the grouping, the system may be able to identify unique fragments with very similar masses which it would have previously grouped together.

2.2 Mass Spectrometry in Metabolomics

Metabolism is the set of chemical processes involved in sustaining living organisms [16]. Within the metabolism, metabolites are small molecules which are intermediaries and products of those chemical processes. Metabolites often serve a variety of functions such as signalling, providing fuel, or inhibiting or facilitating reactions [18].

As mentioned previously, organic compounds such as metabolites are often difficult to characterise in a mass spectrometer using their *m/z* alone, since they are often made up of similar constituents such as Oxygen, Carbon, Hydrogen, Nitrogen, etc. They can also be isomers of one another [12], meaning they are constructed of the same number of each atom, but these atoms are arranged in different ways. Fragmentation of the molecule happens randomly, however, due to the shape and strength of chemical bonds within the molecule, different molecules are more likely to fragment in different ways.

It can be seen in figure 2.1 that it is unlikely that the stable bonds within the benzene ring will break, and much more likely that it will fragment along less stable bonds between the ring and the functional groups.

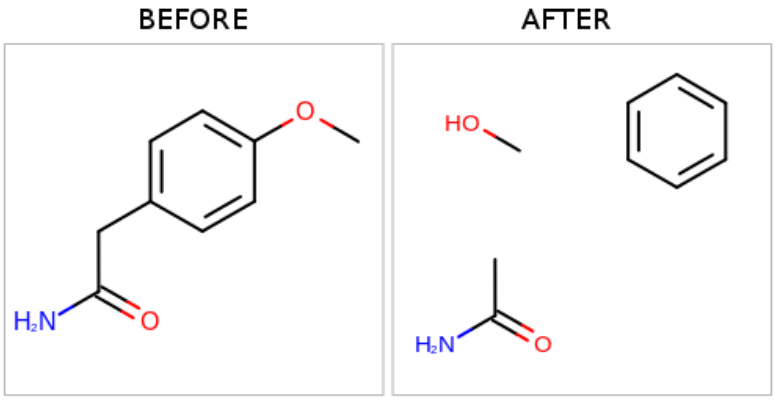

Figure 2.1: (left) A benzene ring with an attached oxygen atom and a functional group containing nitrogen. (right) the same molecule after fragmentation. Image from Eyesopen [13]

Fragmentation is especially useful for metabolites as they are often composed of common substructures, and hence are likely to fragment into these building blocks. This allows for fragmentation spectra to be used to differentiate between two isomers of identical *m/z*, since those isomers are likely to fragment in different ways [12]. Additionally, using fragmentation some underlying chemistry can be exposed, since metabolites which fragment into similar substructures may share some chemical properties or may have similar effects on metabolic processes [20].

2.2.1 Example Mass Spectrometry Data

A mass spectrometry experiment can be run on a sample containing many molecules. The resulting data is in the form of many fragmentation spectra, each of which consists of numerous individual peaks.

It can be seen in figure 2.2 that individual peaks have both a *m/z* and an relative intensity, which can be used as features in machine learning models.

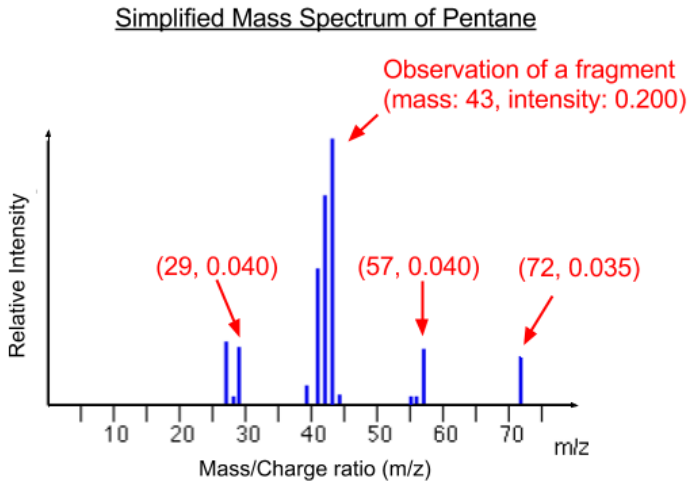

Figure 2.2: An example of a single fragmentation spectrum of Pentane from an experiment. Peaks are observations made by the mass spectrometer — each peak has an associated *m/z* and intensity. Image from Chemguide [11].

2.3 MS2LDA

MS2LDA is an application of topic modelling to the analysis of fragmentation spectra. It allows for the identification of common substructures based on the co-occurrence of mass fragments between different molecules. An advantage of this technique is that it can compare many fragmentation spectra of different metabolites [20] in an unsupervised way, in order to find common underlying characteristics known as "motifs".

Since learning is unsupervised, MS2LDA can identify similarities between molecules without the need for a user to specify which features to look for, and without the need for reference spectra from existing databases. This presents a major advantage over current supervised learning efforts in metabolomics due to the lack of available reference spectra and labelled datasets [16, 18, 20].

Currently, experts are required to do the highly labour intensive task of labelling many spectra manually [12], this is often left undone in studied [17] which is a limiting factor in the rate of progression in the field. MS2LDA help alleviate this issue with features which aid researchers in annotative work. In an experimental run of MS2LDA on beer samples, around 30 motifs of the molecules were identified automatically and then structurally annotated by researchers. Since many molecules share these motifs, this resulted in 70% of the spectra in the sample being annotated. New spectra introduced to the system later can also be annotated automatically in this way, since MS2LDA shares learning across spectra [20]. Figure 2.3 depicts the way words in documents are analogous to fragments in MS2LDA.

MS2LDA provides a web interface for browsing spectra and motifs, which was used extensively during this project to gather testing data.

2.4 Dirichlet Process (DP)

A Dirichlet process (DP) is a stochastic process which is a distribution over distributions [24, 22]. Being stochastic refers to the fact that the samples are drawn from the process are random. Being a distribution over distributions, a Dirichlet process has a base distribution from which samples can be drawn. Each time a sample is drawn from the DP, it chooses whether to return a value that was sampled previously, or a new value drawn from

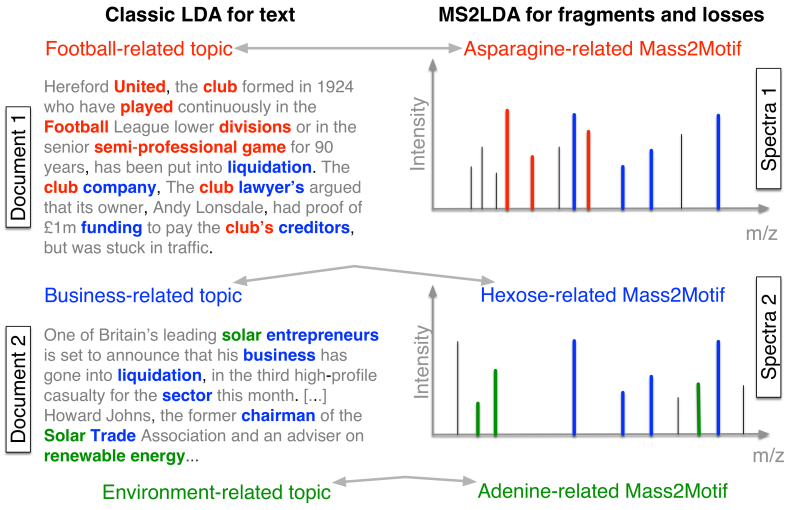

Figure 2.3: A depiction of the way MS2LDA applies a text processing algorithm to mass spectrometry data. On the right documents containing words from a variety of topics are shown, while on the left, spectra containing fragments from a variety of motifs are shown. Image from [20].

the underlying distribution. Values which have been sampled previously do not have equal probability of being sampled again. Instead, their probability of being sampled is proportional to the number of times they have been sampled in the past.

The way the DP produces samples inevitably leads to "clustering" behaviour, whereby many of the samples are of the same few values. This phenomenon is sometimes referred to as "rich-get-richer" [24], and gives the Dirichlet its distinctively "peaky" shape.

2.4.1 Concentration Parameter

The Dirichlet process features a concentration parameter α which specifies the probability it will sample from the underlying distribution, instead of returning a value it has sampled previously [24]. The higher the concentration parameter, the more the Dirichlet process resembles the underlying distribution. A very low concentration parameter results in a distribution which is very concentrated around certain values, since values which have been sampled previously have a much higher probability of being sampled again.

2.4.2 Formal Definition

To describe a Dirichlet process more formally, let us introduce the following definitions:

- H — the base distribution from which samples can be taken
- $X_1, X_2, X_3 ... X_n$ — a set of values which have been sampled from the Dirichlet.
- n_x — the number of times a particular X value has been sampled.
- n — the total number of samples taken from the Dirichlet process.
- α — the concentration parameter for the Dirichlet process.

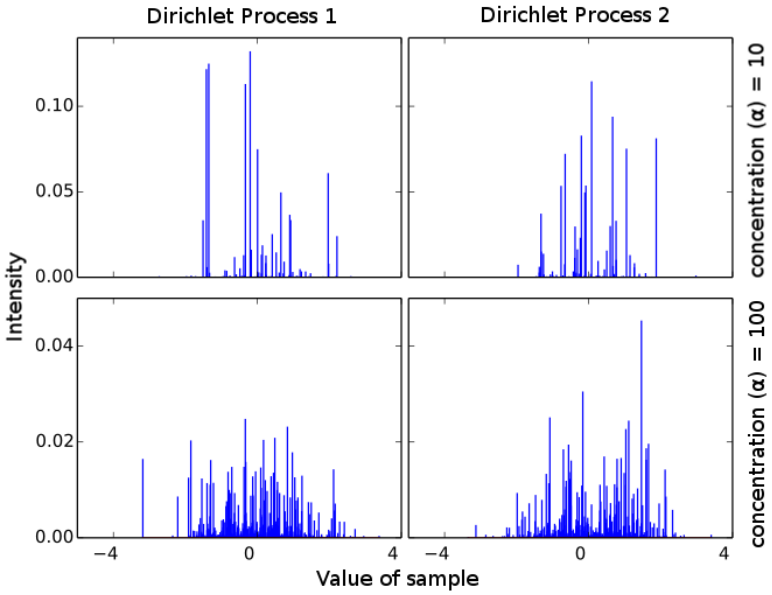

Figure 2.4: Four Dirichlet processes with an underlying normal distribution. Each row is a pair of distributions with different concentration parameters. Each column is two different instances of the same process. [23]

Assuming $n > 1$, the probability of sampling a new value from H is:

$$\frac{\alpha}{\alpha + n - 1}$$

The chance of sampling an existing value X from the process:

$$\frac{n_x}{\alpha + n - 1}$$

Chapter 3

Machine Learning Models

3.1 Latent Dirichlet Allocation (LDA)

Latent Dirichlet Allocation (LDA) is a generative probabilistic machine learning model which has been shown to work well for topic modelling in text corpora [8, 10]. LDA models documents within the corpus as distributions over topics, and topics as distributions over the words which comprise the documents. Being generative, LDA models how it believes the dataset was generated by approximating the set of hidden parameters used to produce the data [8]. Given a word, LDA can then infer which topic the word should belong in based on the probability that it was generated by each of the topics.

When inference is performed in LDA, the model approximates the hidden parameters, allowing it to learn:

- The distribution of topics-over-words (ie. which words different topics are composed of).
- The distribution of documents-over-topics (ie. which topics different documents are composed of).

3.1.1 Dirichlet Prior

LDA gets its name in part from the fact that it uses a Dirichlet prior to describe the dataset [15]. This means that it makes the assumption, before it has seen any data, that the distribution of documents over topics will be in the "shape" of a Dirichlet. This is a useful assumption to make because the DP's clustering behaviour limits each document to being comprised of only a few main topics, which is often the case in the real world. For example, a news article is never normally about "fishing", "sports", "psychology", "finance", and "metabolomics" — it would only ever be about of 2 or 3 of these.

3.1.2 Definition of Terms in a Text Processing Context

There are three main types of object that exist in an LDA model. For the sake of this explanation, let's assume we are modelling a corpus of text documents. The three types of object are the following:

- *Words* — A word is an individual point of discrete data which is generated for, and belongs to, a specific document. There can be many of the same word within a document. Two of the same word within a document could have been generated from different topics. The likelihood that a word was generated from

a particular topic depends on whether other instances of the same word belong to that topic, and whether the document the word belongs to has many other words from that topic.

- *Documents* — Each document contains words. The document is a 'bag of words' meaning the words are not generated or observed in any particular order, and there can be multiple copies of the same word present. Each document is made up of a mixture of topics that influence which words are most likely to be generated for the document.
- *Topics* — In the same way that documents are a mixture of topics, topics are a mixture of words. Topics contain a set of probabilities that they will generate any word in the vocabulary - since the distribution over words is modelled as a Dirichlet, most words will have an almost 0 probability, while others will have a very high probability.

Figure 3.1 Illustrates the way in which documents are comprised of words which belong to different topics, and hence documents have an associated set of topic proportions. It also illustrates how topics are comprised of different words, and words can be more or less likely to appear in a given topic than others.

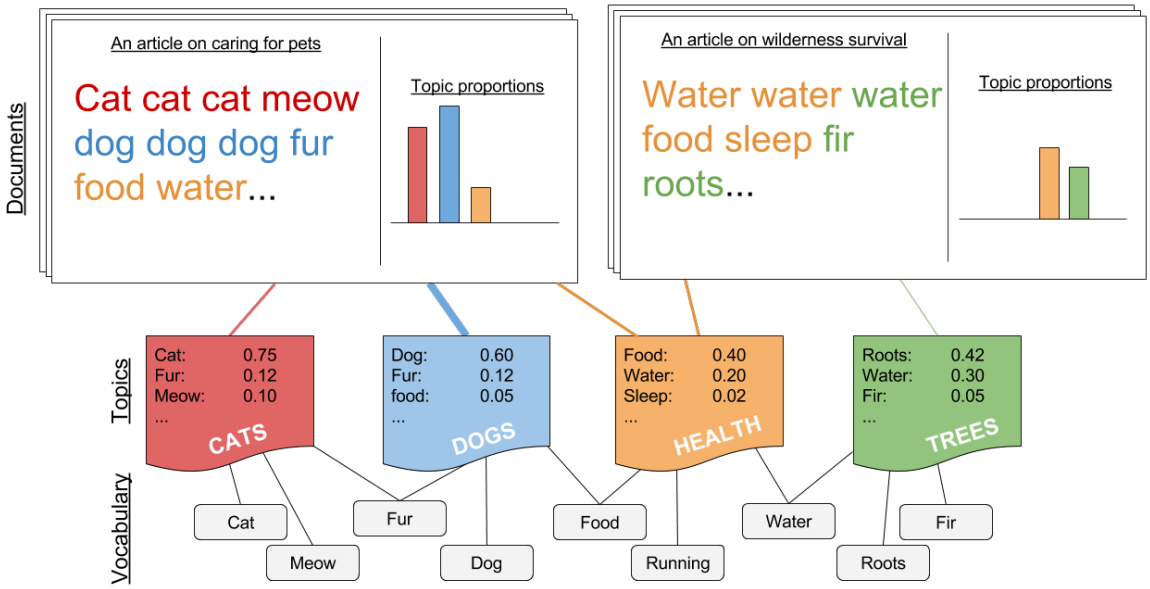

Figure 3.1: Two documents, comprised of different mixtures of 4 different topics. Each of the topics are mixtures over a vocabulary of 9 different words. The graph titled topic proportions represents the Dirichlet process which generates topics within that document.

As mentioned earlier in this section, topics are mixtures over words. Since in real text corpora, some words are more likely to appear in certain topics than others, topics have different probabilities of "generating" each word — these probabilities sum to 1 across the entire vocabulary of words. It is possible that multiple topics could have a high probability of generating the same word. In the diagram, we see that both the topic on health and the topic on trees have a high probability of generating the word "water". The LDA algorithm infers which topic it believes generated the word based on how frequently other words from that topic appeared in the document. In this case, the health topic is more popular in the document on the left than the trees topic, so the LDA algorithm decided that the health topic was most likely to have generated the word "water".

Similarly, documents are mixtures over topics. In real text corpora, a document is only likely to be about a small number of topics. For this reason, the Dirichlet prior is useful for clustering the topic proportions of a document to be comprised of only a few main topics.

3.1.3 Generating a Word in LDA

LDA is generative meaning it models how it believes the text corpus was generated. For this reason, understanding the way a data point (word) is generated is intrinsic to being able to perform inference on some data and discover the parameters which generated it.

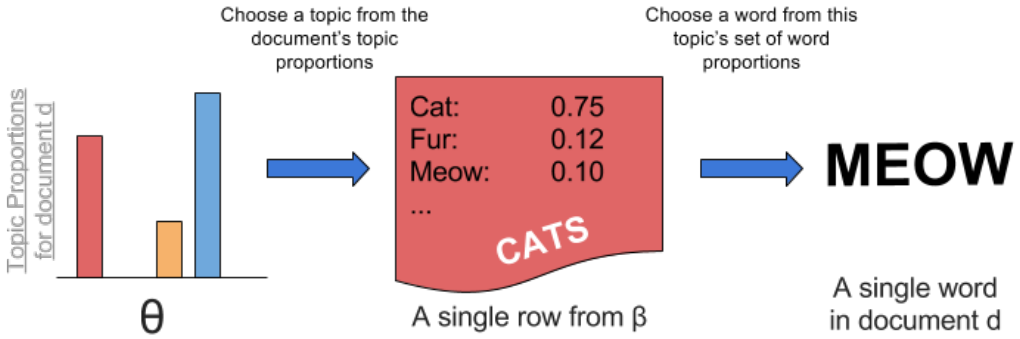

Figure 3.2: A depiction of the process of generating a word for a given document. Explanation of the annotations underneath each stage in the formal definition of LDA 3.1.4.

It can be seen in figure 3.2 that the process takes place in 2 stages:

- A topic is selected for the word by sampling from the "topic proportions" Dirichlet process within the document. This word is then added to the topic proportions, influencing the topic choice for future words. This is why the Dirichlet is named a "process" rather than just a "distribution".
- A word is selected by sampling from the "word proportions" process within the topic. Like the "topic proportions" for the document, the word being generated from the "word proportions" process influences which words will be generated in the future.

Note that although the topic proportions DP is likely to choose a topic which has been chosen many time previously, this is not always the case. As we saw in section 2.4.2, there is a chance of $\frac{\alpha}{\alpha+n-1}$ that a Dirichlet process will draw a new sample from the base distribution H. This means there is always a small chance that the model will sample a new topic within a document, or a new word within a topic.

3.1.4 Formal Definition of LDA

Figure 3.3 presents the LDA algorithm more formally as a "plate diagram", where the plates are parts of the system which are replicated many times. The hyperparameter α can be seen on the left — this is the concentration parameter for all of the θ Dirichlet processes, of which there is one for each document. θ is a vector, with a length equal to the number of topics, which represents the topic composition of document M.

The parameter z, within the N plate, represents a single topic choice for the word w. z is sampled from the θ Dirichlet process and is used to lookup the corresponding row in the β matrix. β is a matrix which represents the probability of generating any word given any topic. From this matrix, a single row is extracted containing a probability for every word. A word is then sampled from this probability distribution, and becomes the new word within document M.

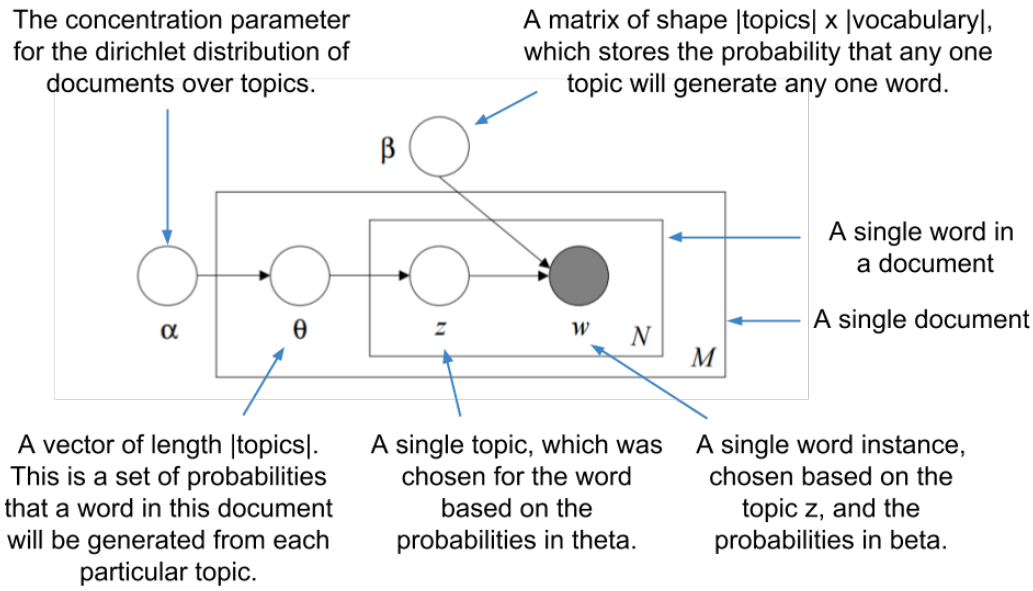

Figure 3.3: A more formal definition of the LDA algorithm in a "plate diagram". Plate N represents one of many words, and plate M represents one of many documents. Image from [8].

3.1.5 inference in LDA Using Gibbs Sampling

We have now seen how modelling a corpus of text in LDA allows us to generate a realistic set of words that might occur based on 1) Which topics feature most strongly in the document, and 2) which words feature most strongly in the topics. We have seen how the distribution of documents-over-topics is governed by a Dirichlet process θ for each document. The distribution of topics-over-words is governed by the rows in the matrix β, each of which is a Dirichlet process for a topic. In a real world scenario when the LDA model receives a corpus of text, these parameters are "hidden", meaning that they are not directly observable. In order to approximate them, we need to perform "inference" which can be done in a variety of ways.

A popular inference method in Bayesian problems is Gibbs Sampling (Gibbs) which works by making repeated guesses about the values of the hidden parameter, refining its guess with each iteration [15]. This class of algorithm is know as a "Markov Chain Monte Carlo" method.

Gibbs sampling works in a very simple way. Since it has no prior knowledge of which word should belong to which topic, words begin assigned to random topics. Gibbs then iterates over each document, and within each document it iterates over each word. Each word is unassigned from its current topic, and a new topic is sampled for the word.

A small visualisation of the way Gibbs works can be seen in figure 3.4 where the unassigned word has a strong incentive to join the red topic, since there are many other words in the document in the red topic. However, it also has a strong incentive to join the blue topic since many of the same word (word "b") are assigned to this topic.

This incentive that a word experiences can be quantified using two probabilities. These probabilities are dependent on the hidden parameters θ and β, which are the parameters that Gibbs is continually refining with each guess. Some definitions of terms which were outlined by Steyvers and Griffiths [14] are as follows:

- $n_{-i,j}^{(w_i)}$ — Number of uses of the word w_i in topic j, excluding the current instance of wi.

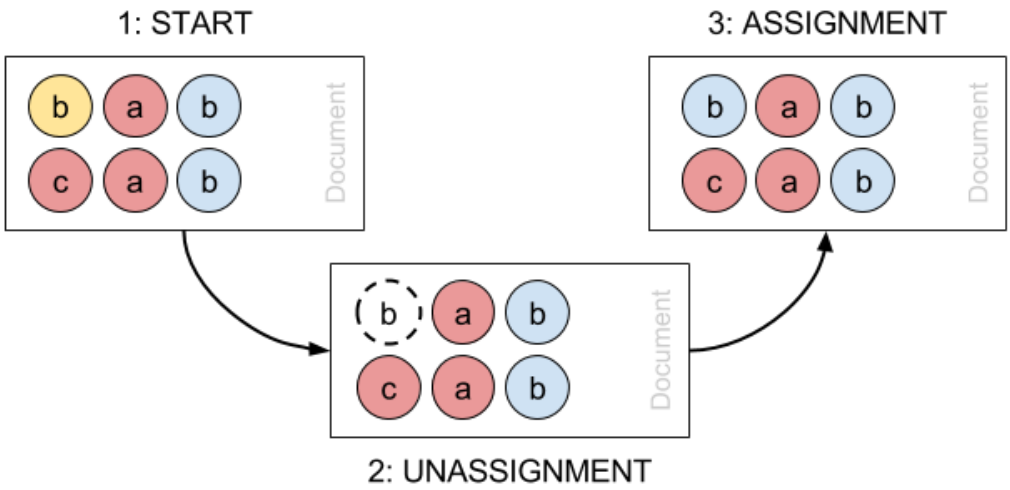

Figure 3.4: A single iteration of a Gibbs sampler. Topics are represented by colour and words are represented by letters. A word in the yellow topic is unassigned and a new topic is chosen for it.

- $n_{-i,j}^{(\cdot)}$ — Number of uses of this topic for all words in all documents, excluding the word wi current instance.
- $n_{-i,j}^{(d_i)}$ — Number of uses of the topic j in document d_i, excluding the current word i.
- $n_{-i,\cdot}^{(d_i)}$ — Number of uses of this topic for all words in all documents, excluding the word wi current instance.
- δ — the hyperparameter on the Dirichlet process that controls topic distributions over words.
- α — The hyperparameter on the Dirichlet process that controls documents-over-topics.
- W — Number of words in the vocabulary.
- T — Number of topics in the corpus.
- z_{-i} — Topic assignments for all words excluding the current word.
- w — All words in the corpus.

Using these terms, we can express the probability that word i in document d will choose topic j in the following way:

$$P(\textit{choosing topic } j \textit{ for word } w_i \textit{ in document } d)$$

$$= \textit{how common is word } i \textit{ in topic } j \cdot \textit{how common is topic } j \textit{ in document } d$$

$$= \frac{\textit{uses of word } i \textit{ in topic } j + \delta}{\textit{total uses of all words in topic } j + W\delta} \cdot \frac{\textit{uses of topic } j \textit{ in doc } d + \alpha}{\textit{total uses of all topics in doc } d + T\alpha}$$

$$= \frac{n_{-i,j}^{(w_i)} + \delta}{n_{-i,j}^{(\cdot)} + W\delta} \cdot \frac{n_{-i,j}^{(d_i)} + \alpha}{n_{-i,\cdot}^{(d_i)} + T\alpha}$$

We can see that the probability a word will be assigned to a topic is governed by two "forces": how common the word is in the prospective topic, and how common the prospective topic is in the document. Although at first the choices made by the Gibbs sampler are poor, since it has a poor approximation of the hidden parameters,

after many iterations these forces allow words to be grouped into suitable topics. Note that whenever "uses of _____ in _____" is mentioned, this excludes the current item in question. For example, when "uses of word i in topic j" is mentioned, this excludes the current word in question from the count.

It can also be seen that extra δ and α are used on the top and bottom of the fractions. This has the effect of ensuring that even when the "uses of _____ in _____" is equal to 0, there is still a small probability that this topic will be chosen, since in a Dirichlet process there is always a small chance that a new sample will be drawn from the base distribution.

3.1.6 Parameters & Hyperparameters in LDA

One of the issues that can be encountered using LDA is that the concentration of the Dirichlet processes governing documents-over-topics and topics-over-words may not be suited to the corpus of text. To remedy this, the two hyperparameters for the LDA model can be adjusted along with parameters for the underlying distribution, which are the following [8]:

- α - The concentration parameter on the Dirichlet prior for the distribution of documents-over-topics. A lower α means documents are comprised of fewer, more concentrated topics.
- δ - The concentration parameter on the Dirichlet prior for the distribution of topics-over-words. A lower δ means documents are comprised of fewer, more concentrated words.
- Mean & variance of the underlying distribution.

3.2 Hierarchical Dirichlet Process (HDP)

So far we have seen how LDA can be applied in a text processing context. One limitation of this model in its current incarnation is that it can only recognise words in a finite vocabulary. This finite vocabulary can cause problems in other fields where the data points (words) are continuous rather than a set of discrete items. It also limits our model to words that it knows exists, and prevents it from learning a new word should one come into existence. This limitation necessitates that we implement an alternate model called a Hierarchical Dirichlet Process (HDP). The HDP is not a direct variation on LDA, as it was developed separately, however it works in a very similar way and performs many of the same functions.

3.2.1 Words as Distributions

In order to extend our model to incorporate an infinite vocabulary, imagine that words are actually distributions in a continuous space. At the mean of the distribution is the correct spelling, and surrounding it are possible misspellings of the word. Each word seen in a document is no longer a word in its own right, but is now an "observation" of a word from a distribution. It can be seen in figure 3.5 that the further from the mean the sample is taken, the more uncommon the misspellings of the word become.

3.2.2 Advatages of Infinite Vocabulary

Robust to Noise This way of modelling words as distributions has a number of major advantages. Firstly, the model is able to group "observations" of words into a single "conceptual" word. This allows it to identify

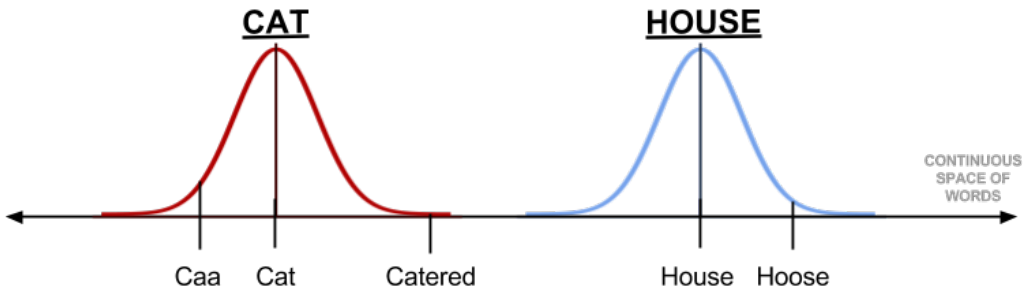

Figure 3.5: A visualisation of two words in the English language surrounded by misspellings. The words are distributions in a continuous space in which every possible combination of letters is present.

misspellings and group them along with the word it believes they are a misspelling of, rather than treating them as unique words. If misspellings were extremely common, a normal LDA model would be completely unable to draw correlations between words and topics because each word would likely only occur once.

Can Learn New Words and Their Topic Associations Another advantage of this model is the potential for it to learn to recognise new words. It can be seen in figure 3.6 that until the year 2012, the word brexit did not exist in the English vocabulary. If ever the word brexit occurred, it might have been assumed to be an unlikely misspelling of the word "exit". However, as more occurrences of the word appear, the model detects that the probability of this misspelling occurring so often is too low, and that it must be its own unique word.

The model also makes use of the knowledge of which words "brexit" regularly occurs alongside. In this example, the word "exit" might commonly occur in architectural articles, whereas "brexit" would occur in political documents.

Not only was the model able to determine that the word brexit was in fact likely not a misspelling of the word "exit" — it was also able to determine that it was likely generated by the "politics" topic. It was able to determine this due to its co-occurrence with other words like "EU" and "parliament" which are often associated with politics.

3.2.3 Definition of the Hierarchical Dirichlet Process

One of the most attractive features of the Bayesian paradigm for machine learning is the concept of "hierarchy" in models. One of the primary allowances of a hierarchical model is that it can share information between learning tasks [22]. Hierarchy within the model allows it to solve multiple learning tasks simultaneously and also to have the processes within "submodels" represent different levels of detail to the higher levels of the hierarchy [6]. A popular analogy for describing this model is the "Chinese Restaurant Process" [7].

In order to augment our previous LDA model with the ability to group observations into words, we need to add a level of hierarchy which will handle observation clustering. The Hierarchical Dirichlet Process (HDP) incorporates the concept of a hierarchy of submodels, and allows for multiple learning tasks to happen simultaneously.

The HDP works perfectly for the scenario outlined above, where words are a continuous distribution over spellings, since it has the following characteristics:

- The scenario is a case where multiple problems are present. One of the problems is to cluster observations into words while the other is to discover the distribution of topics over these words and the distribution

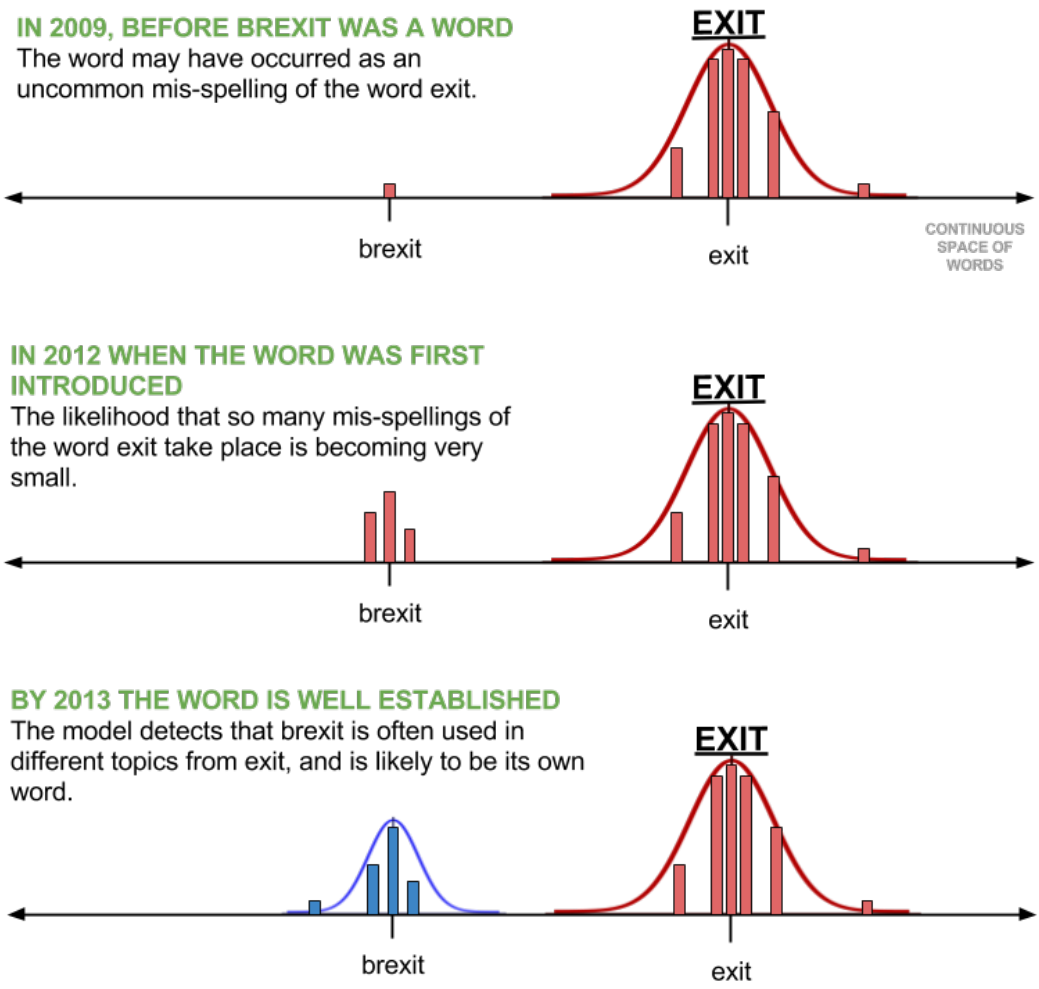

Figure 3.6: A scenario where a new word is introduced into a corpus. There is an acquisition period wherein until there are enough observations of the word, the model is still not confident enough that it should be treated as a real word.

of documents-over-topics. The HDP allows us to incorporate both of these learning tasks into the same model as different submodels.

- The learning task of discovering the distribution of topics-over-words should not have to concern itself with the details of clustering observations into words. A hierarchical process allows the word clustering task to be separated into a submodel, meaning the topic distribution learning task never has to concern itself with the fact that observations are not real words.
- The knowledge about observation clusters can be shared between documents, and all documents can contribute towards the learning of words.

A very important distinction to make in the HDP is between an observation and a word. After a word has been sampled from a topic, the HDP includes the additional step of sampling an observation from that word. This observation may be a variation on the word such as a misspelling of it.

In this example we can see that the observation “hlep” was sampled from the word “help”. A few points of note about this example are:

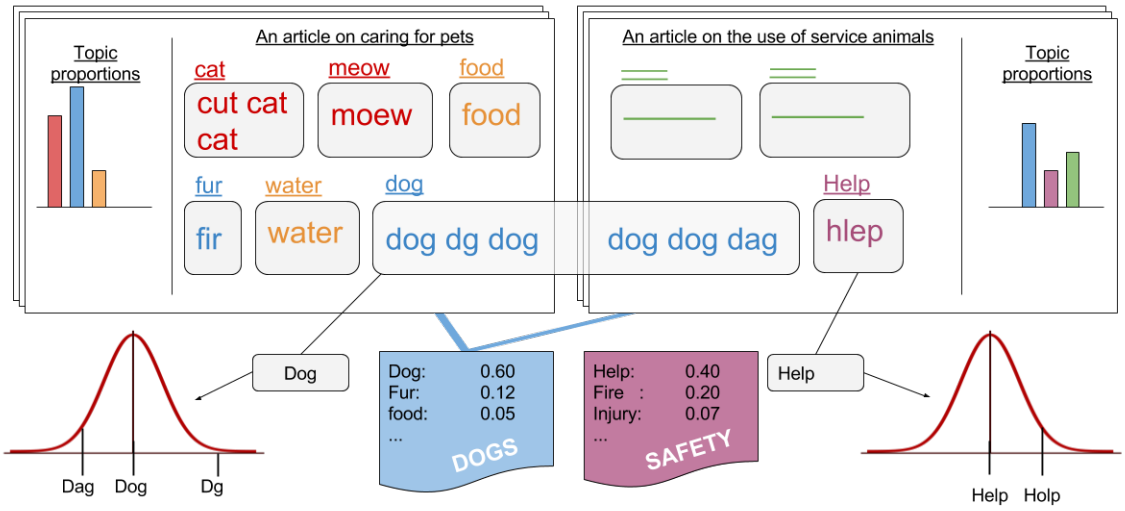

Figure 3.7: A visualisation of two documents in an HDP model. Grey boxes within the documents indicate the words present in that document. Observations of that word (sometimes misspelled) are seen within that word. Below are two topics, from which the documents sample words.

- A word is selected from a topic before an observation of that word is made. This means that the topic which the word was generated from is unaware of whether the observation turns out to be a misspelling or not.
- The word "dog" is shared between the two documents, as indicated by the grey box which bridges the gap between them. Documents share and contribute to a common understanding of which words exist in the vocabulary, although obviously not every document needs to make use of every word in the vocabulary.
- The word "fur" in the leftmost document has been misspelled as the word "fir". In figure 2.4 we have seen that "fir" might be a real word in the "trees" topic. However, the trees topic is not present in the document, and the dog and cat topics are heavily present. For this reason, the model has inferred that it was more likely that the word "fir" was a misspelling of a "fur" from the dog topic than it was a correct spelling of "fir" from the trees topic.

3.2.4 Generating Words in an Infinite Vocabulary

In a HDP, words are generated for a document in a similar way to LDA. Since words are no longer single data points, but are instead distributions, after a word is chosen it must be sampled from to get an "observation" of that word.

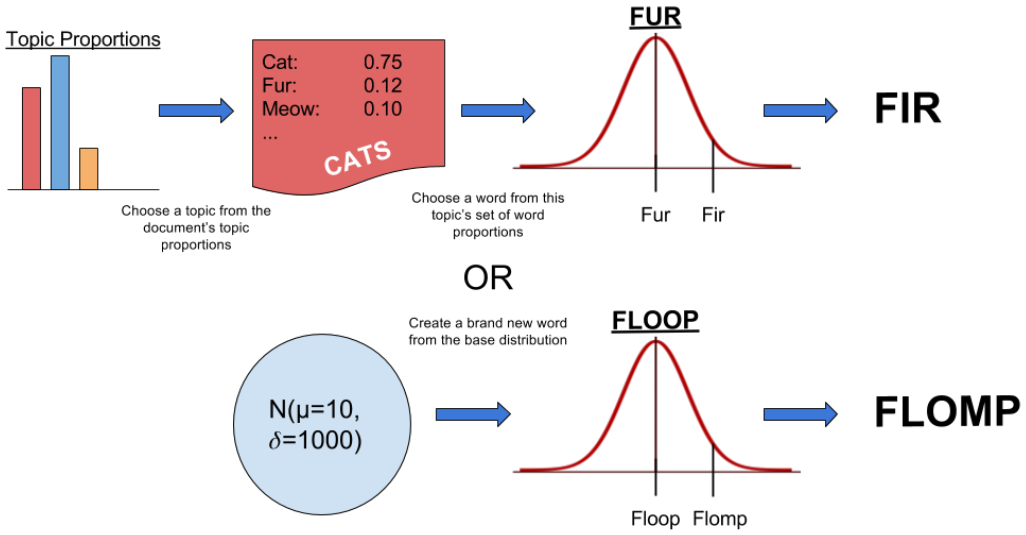

Figure 3.8: The process of generating an observation in HDP. Then top half shows the process for an existing word, and the bottom half shows an entirely new word being generated from the base distribution.

The top half of the figure 3.8 depicts the word "fur" being chosen from a topic about cats before being sampled to produce the observation "fir". However, in order to allow for an infinite vocabulary of words, the model needs a way to sample a new word from the base distribution. The bottom half of the figure depicts how each time a word selected from a topic, there is a small chance that the topic will sample an entirely new word from the base distribution. In this example, instead of choosing an existing word from the CATS topic, the topic samples a new word "floop" from the base distribution, from which an observation "flomp" is then sampled.

3.2.5 Inference in HDP using Gibbs Sampling

Gibbs sampling is the only feasible tool to use for inference in this scenario, since it allows for the easy growing and shrinking of the vocabulary. In the LDA model we saw how a Gibbs sampler worked classifying words into topics. In the HDP, the Gibbs sampler works in a relatively similar way. However, since words are now distributions rather than individual data points, the HDP is actually clustering observations into words, and simultaneously discovering the distribution of topics over these words.

We can see from figure 3.7 from a previous section, that observations are clustered together into a single conceptual word. Let us define this cluster of observations as a "grouping" which "expresses" a certain word. A grouping is present within a *topic* and can be shared between multiple documents. Some notes about groupings are:

- A grouping is strictly contained within a topic. If a word appears in multiple topics, there will be a separate groupings for that word in each topic.
- It is very possible for multiple groupings to express the same word in different topics. For example, there may be a grouping for the word "food" in the restaurants topic and also in the health topic.
- There can be multiple groupings within a topic which express the same word. Rather counterintuitively, there can be two "food" groupings within the restaurants topic, which is a side effect of the way the Gibbs sampler decides on which grouping to put an observation in.

- Groupings can be shared between documents. As we have seen in the diagram, the grouping for "dog" originates from a single topic (since this is a requirement of the grouping) but it is shared between two documents.

As with LDA, the Gibbs sampler unassigns and reassigns data points in an iterative manner. The difference is that in HDP, the Gibbs sampler is reassigning observations to groupings rather than words to topics. Since a grouping belongs to a single topic and expresses a single word, the Gibbs sampler implicitly makes the decision about which topic and word the observation is sampled from. In order to understand the probability that the Gibbs sampler will choose any particular grouping for an observation, or create a new grouping, some terms are defined as follows:

- $groupingcount_{[topic|word|corpus]}$ — The number of groupings which are associated with the current, the current word, of all of the groupings in the entire corpus (depending on the subscript).
- $observationcount_{[grouping|topic|word|corpus]}$ — The number of observations in the current grouping, topic, word or the entire corpus (depending on the subscript).
- $\alpha_{[topic|base]}$ — The concentration parameter for the current topic, or the base concentration parameter (depending on the subscript).
- $\delta_{[word|base]}$ — The standard deviation for the current word or the base distribution (depending on the subscript).
- μ_{base} — The mean for the base distribution.
- x — The current observation we are choosing a grouping for.

Choosing an Existing Grouping

The probability that a particular existing grouping g expressing the word w, will be chosen for an observation x can be summarised as follows:

$$GroupingProbability_g = GroupingPrior_g \cdot GroupingLikelihood_g$$

Where the prior and likelihood terms are:

$$GroupingPrior_g = \frac{observationcount_g}{observationcount_{topic} + \alpha_{topic}}$$

$$GroupingLikelihood_g = N(x|a_w, {b_w}^2 + {\delta_w}^2)$$

Where a_w and ${b_w}^2 + {\delta_w}^2$ are the mean and variance respectively, and are defined as the following:

$${b_w}^2 = \left(\frac{1}{{\delta_{base}}^2} + \frac{observationcount_w}{{\delta_w}^2}\right)^{-1}$$

$$a_w = {b_w}^2 \left(\frac{\mu_{base}}{{\delta_{base}}^2} + \frac{\Sigma x}{{\delta_w}^2}\right)$$

Where Σx is the sum over all of the observations current in groupings which express the word w.

Choosing to create a New Group

One of the mechanisms that allows HDP to have an infinite vocabulary is that each time an observation is reassigned, there is a chance it will create a new grouping, which in turn may create a new word to express. The probability that an observation will create a new grouping is:

$$GroupingProbability_{new} = GroupingPrior_{new} \cdot GroupingLikelihood_{new}$$

Where the prior and likelihood term are:

$$GroupingPrior_{new} = \frac{\alpha_{topic}}{observationcount_{topic} + \alpha_{topic}}$$

$$GroupingLikelihood_{new} = WordPrior_{newword} \cdot WordLikelihood_{newword} + \sum_{w} WordPrior_{w} \cdot WordLikelihood_{w}$$

Where $\sum_w$ is the sum over all of the words in the corpus. In order to understand the likelihood term for a new grouping, we need to understand the probabilities associated with choosing a word for the grouping. These probabilities are outlined in the following section.

Choosing an Existing Word

The probability of choosing any existing word "w" to be expressed by a new group is:

$$WordProbability_{w} = WordPrior_{w} \cdot WordLikelihood_{w}$$

Prior for each existing word in the franchise:

$$WordPrior_{w} = \frac{groupingcount_{w}}{groupingcount_{corpus} + \alpha_{base}}$$

$$WordLikelihood_{w} = N(x|a_{w}, {b_{w}}^{2} + {\delta_{w}}^{2})$$

Where a_w and ${b_w}^2 + {\delta_w}^2$ are the mean and variance respectively, and are defined as before.

Choosing a New Word for a New Grouping

As mentioned previously, the mechanism that allows the HDP to have an infinite vocabulary is the chance that when a new grouping of observations is created, the grouping can choose to express a new word which is not part of the existing vocabulary. The probability of choosing a new word for a new grouping is:

$$WordProbability_{new} = WordPrior_{new} \cdot WordLikelihood_{new}$$

Prior for each existing word in the corpus:

$$WordPrior_{new} = \frac{\alpha_{base}}{groupingcount_{corpus} + \alpha_{base}}$$

Likelihood for each existing word in the corpus:

$$WordLikelihood_{new} = N(x|\mu_{base}, {\delta_{base}}^2 + {\delta_w}^2), \text{ where } {\delta_{base}}^2 + {\delta_w}^2 \text{ is the variance.}$$

3.2.6 Additional Hyperparameters in HDP

Since HDP is a hierarchical model, it features more parameters and hyperparameters that allow the behaviour of the models to be tuned. Some of these parameters are as follows:

- concentration α_t for each topic — unlike the LDA model which has a single concentration parameter for the Dirichlet process governing document distribution over topics, the HDP model has a concentration parameter for each topic, since each topic is its own discrete submodel.
- Variance δ_w for words — Whereas in LDA words were discrete data points, in HDP words are distributions, so they have an associated variance term which defines how likely misspellings are to occur.

Chapter 4

Implementation

4.1 Mass Spectrometry Data Format

MS2LDA preprocesses raw mass spectrometry data before it is used to train the LDA model. By implementing the infinite vocabulary extension to LDA, some of the preprocessing stages were eliminated. For this reason, the unprocessed mass spectrometry data was used in the HDP.

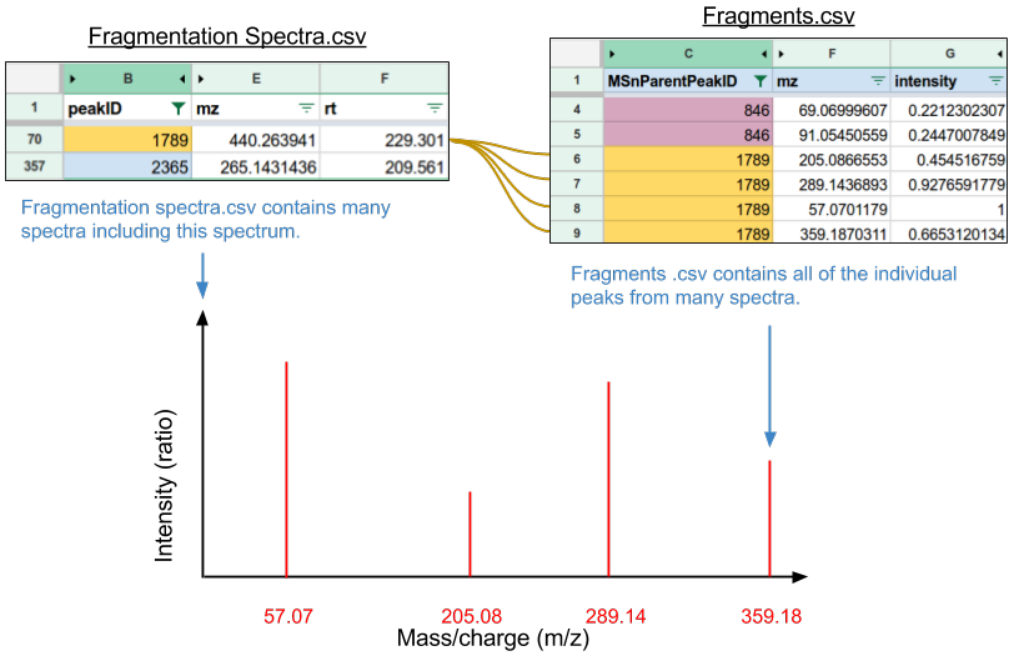

Figure 4.1: A visualisation of the unprocessed data files, and how they map onto a mass spectrum.

The data from mass spectrometry experiments is presented in the form of two .csv files: a "fragments" file and a "fragmentation spectra" file. The fragments file contains rows, each of which represent a single peak (fragment) inside a single fragmentation spectrum. These rows contain three columns, which are the following:

- *MSnParentPeakID* — This peak ID allows us to reference the parent fragmentation spectra in the other .csv file. There will be many peaks in the file with the same ParentPeakID.
- *mz* — The mass to charge ratio of the fragment.

- *intensity* — The intensity of the observation relative to other peaks in the fragmentation spectrum. After normalisation, the greatest peak in the fragmentation spectrum has an intensity of 1, and the others have an intensity relative to this.

In the fragmentation spectra file, each row represents an entire fragmentation spectrum for a molecule. The rows also contain three columns:

- *peakID* — This allows individual fragments in the fragments file to reference the spectra they are a part of. For this reason, each row in this file has a unique peakID.
- *mz* — The mass to charge ratio of the entire molecule.
- *rt* — The mass spectrometer is often used in conjunction with a chromatograph. The chromatograph features a procedure which separates analytes in a column based on their chemical properties. The analytes then reach the mass spectrometer at different times. Although retention time is not used as training data in this model, it is used in conjunction with mz to differentiate between different molecules which could be of the same *m/z*.

4.2 Application of LDA to Metabolomics (MS2LDA)

4.2.1 Preprocessing

When LDA is used in a text processing context, topics are a distribution over a finite number of discrete words, since there is only a finite vocabulary of valid words in any one language. In the metabolomics context, MS2LDA preprocesses the fragmentation spectra in order to group observations into a finite "vocabulary" of fragments. This prevents the issue of each small variation in *m/z* (due to noise) being treated as a unique feature, which would make it more difficult for the model to identify any useful correlations between spectra. However, some researchers believe a lack of standardisation in preprocessing steps are a current limitation in the field [17].

The preprocessing step clusters any observations which fall within 7 parts-per-million (PPM) of one another into the same fragment. This means that any two observations more than 7 PPM apart are likely to be interpreted as unique fragments.

In figure 4.2 it can be seen that two sets of observations have been grouped into two discrete peaks. To decide on the *m/z* of the fragment, a mean is calculated over all observations. These peaks' *m/z* values form the "vocabulary" which the LDA algorithm will use to produce a set of topics.

We can see that if two fragments with very similar *m/z* values had observations under 7 PPM apart, they might be erroneously clustered into a single feature by the preprocessing step.

4.2.2 Converting Text Processing Terminology for Metabolomics

In order to apply the LDA algorithm to metabolomics, a set of terms other than "words", "documents" and "topics" should be introduced. Table 4.1 shows the correspondence between terms in the text processing and metabolomics domains.

In the metabolomics context the aim is to observe common underlying characteristics shared between multiple molecules, in the same way that topics identify common characteristics between text documents. A collection

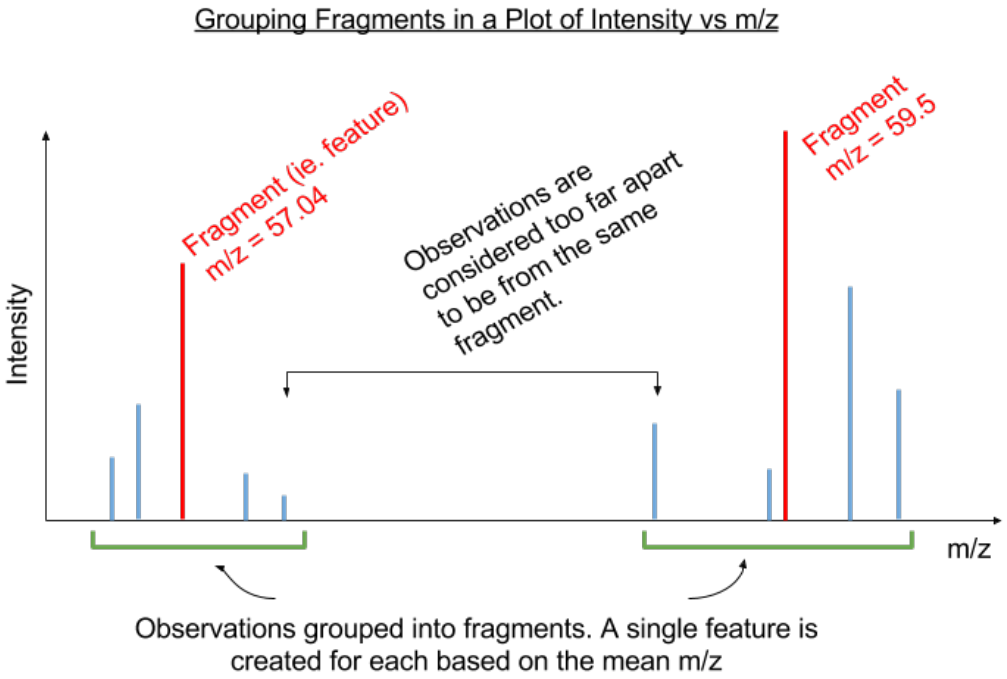

Figure 4.2: A set of observations over a continuous range of *m/z* values (in blue) have been grouped into a finite number of discrete peaks (in red).

Text Processing Context	Metabolomics Context
Word	Fragment
Document	Molecule (single fragmentation spectrum)
Corpus	Sample in MS (many fragmentation spectra)
Topic	Motif

Table 4.1: Showing the correspondence between terms when LDA is applied in a text processing vs. a metabolomics context. MS = Mass spectrometer

of fragmentation spectra from a mass spectrometry experiment is analogous to a corpus of text, where an individual fragmentation spectrum of a single molecule is analogous to a single text document.

In the same way that a text document is constructed from words, a molecule's fragmentation spectrum is constructed from fragments. In a text document, a word might appear many times; for example, in a document about football, the word "team" might occur 15 times, whereas the word "striker" might appear twice. In a metabolomics context the word itself corresponds to the *m/z* of the peak, and the frequency with which it appears corresponds to the intensity of the measurement.

Depending on the inference method, a peak may be "quantised" into a number of discrete observations of a particular *m/z*. For example, in variational inference peaks can be represented by a single *m/z* and a single intensity value, whereas when using Gibbs sampling for inference, it is more suitable to break a peak of intensity 0.8 into 8 discrete objects, and a peak of intensity 0.2 into 2.

4.3 Application of HDP to Metabolomics

4.3.1 Converting LDA Vocabulary for HDP

We have now seen how an extension of the LDA algorithm can be implemented as a HDP to feature an infinite vocabulary of "words". We have also seen how the LDA algorithm can be translated from use in text corpora to being directly applicable to metabolomics.

Since the HDP is similar to LDA, most of the terminology remains consistent when translating between a text processing and a metabolomics context, as can be seen in table 4.2.

Text Processing Context	Metabolomics Context
Observation	Observation
Grouping	Grouping
Word	Fragment
Document	Molecule (single fragmentation spectrum)
Corpus	Sample in MS (many fragmentation spectra)
Topic	Motif

Table 4.2: All of the terms present in an HDP text processing context, and how they translate into metabolomics terms

It can be seen that this terminology is consistent with LDA, except from the addition of the "observation" and "grouping" terms. A translation of figure 3.6, where the HDP decides to break the word "brexit" into its own unique word, can be seen in figure 4.3. The same principles from test data apply for fragmentation spectra data: where when enough points are observed in a cluster, the HDP decides that they are likely to represent a unique fragment.

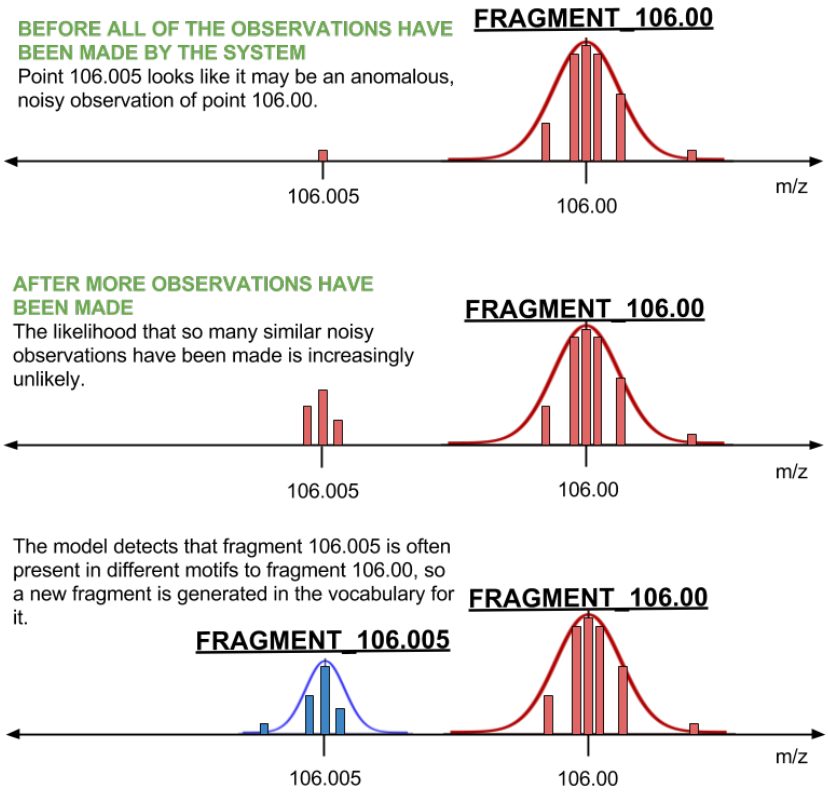

Figure 4.3: a translation of figure 3.6 into a mass spectrometry context.

4.4 Implementation of HDP and Visualisation Tools

A python library was created named *hdplab* where the main body of the HDP was implemented in an object oriented way. The "numpy" and "scipy" [2, 5] libraries were utilised regularly for their common statistical tools. A small segment of numpy code can be seen in the figure 4.4.

```
def _calculate_likelihood_single(self, observation):
    b_squared = self._calculate_b_squared_term()
    a = self._calculate_a_term(b_squared = b_squared)

    return norm.pdf(observation.v, loc=a, scale=np.sqrt(b_squared + np.power(self.delta, 2)))
```

Figure 4.4: A small section of code which utilises numpy as scipy.

A Jupyter notebook [1] was then used to import the model from the python library, run the experiment and visualise the outcome.

4.4.1 Experimental Framework

In order to make it easier to run experiments and extract the results, three python classes were created:

- *ExperimentSettings* - Responsible for storing all of the settings of the experiment. Some of the settings include the number of iterations to take, the parameters and hyperparameters for the model, and param-

```
In [16]: experiment = Experiment(settings, documents)

In [17]: from datetime import datetime
         start_time = datetime.now()
         print "Time now:", start_time.strftime("%Y-%m-%d %H:%M:%S")

         for i in range(settings.burn_iters):
             experiment.iter(burn=True)

         for i in range(settings.iters):
             experiment.iter()

         end_time = datetime.now()
         print "Elapsed time:", (end_time - start_time)
         """""

Time now: 2017-03-12 10:42:37

iter 0
dishes: 67   restaurant_n_tables: 2 11 8 15 8 11 28 21 2 5   total tables: 111
restaurant_n_points 6 36 72 159 73 44 152 161 11 45 rest_tot_points 759 tab_tot_points 759 dish_tot_points 759
. . . . . .
iter 10
dishes: 67   restaurant_n_tables: 3 11 7 14 9 5 25 9 4 8   total tables: 95
restaurant_n_points 14 32 136 149 79 38 125 56 51 79 rest_tot_points 759 tab_tot_points 759 dish_tot_points 759
. . . . . .
iter 20
dishes: 66   restaurant_n_tables: 3 6 5 11 14 7 23 13 3 6   total tables: 91
restaurant_n_points 14 21 156 150 84 43 80 68 51 92 rest_tot_points 759 tab_tot_points 759 dish_tot_points 759
. . . . . .
iter 30
dishes: 66   restaurant_n_tables: 4 7 4 11 13 7 25 10 5 5   total tables: 91
restaurant_n_points 16 23 146 150 87 27 87 60 83 80 rest_tot_points 759 tab_tot_points 759 dish_tot_points 759
. . . . . .
```

Figure 4.5: A screenshot of the Jupyter notebook used for experiments. In the screenshot, the Experiment object is initialised and iterations are performed with a live experiment log.

eters defining the input spectra such as the names of the spectrum files and the intensity threshold for observations.

- *ExperimentState* - This class stores the state of the experiment including the number of iterations that have been made. It also stores data which might be useful when evaluating the output such as the mean mass to charge ratios of all fragments for each iteration.
- *Experiment* - This class acts as a container for the *ExperimentSettings, ExperimentState* and the main model. It contains an "iter()" method allowing the researcher to run iterations without having to interact directly with the model itself.

Since experiments took many hours to run, it was important not to discard the resulting data. For this reason the Experiment class can be passed to a method which saves it into a specified file using python pickle [3]. The experiment can be loaded later in order to run more iterations or perform further analyses. This was useful on multiple occasions where an experiment which took many hours to run could be saved, and could be loaded a week later to analyse the data.

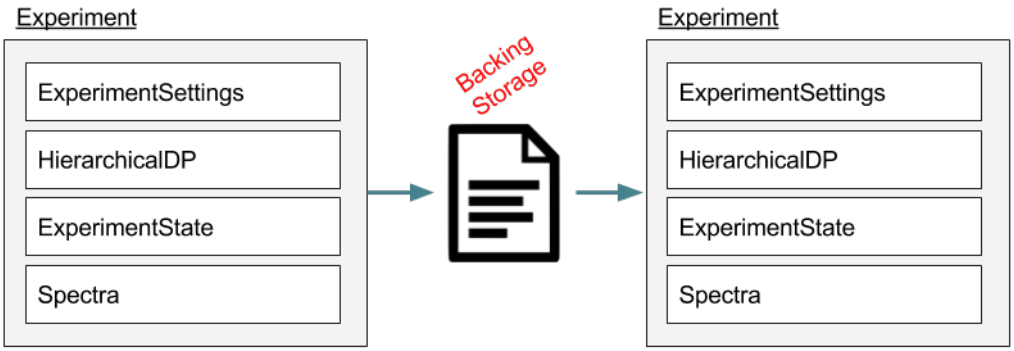

Figure 4.6: A visualisation of the class structure along with the loading and saving functionality.

4.4.2 Visualisations and Reports

One of the aims of this project was to create a set of visualisation and reporting tools to aid researchers when evaluating the HDP model. This is necessary since the HDP undertakes multiple learning tasks simultaneously, making the output data complex and difficult for researchers to comprehend. A variety of visualisations and reports were included in the hdplab library, which can be run within a Jupyter notebook to help understand the results of an experiment.

All visualisations can be produced in the Jupyter notebook by passing the Experiment object to methods in the *hdpanalyser.py* file. Some of the most effective visualisations are as follows:

Textual dreakdowns and live experiment log are available before during, and after an experiment. These include a breakdown of all observations in a dataset pre-experiment, fragments detected post-experiment, and information on the variation on grouping and fragment presence between iterations.

A time series visualisation of the mean *m/z* of each fragment can be seen in 4.7. Lines are plotted in different colours to improve readability, where the opacity represents the popularity of that fragment compared to the others.

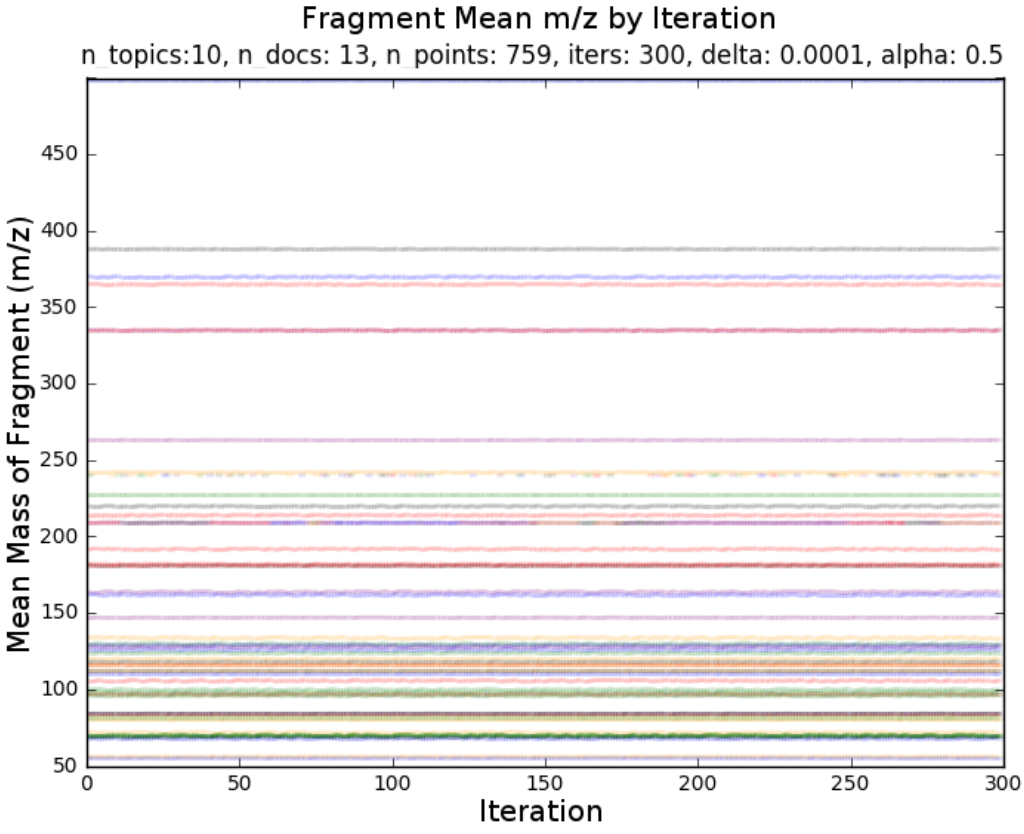

Figure 4.7: The "fragment time series" visualisation output. Along the X axis is the iteration number and on the Y axis is the *m/z* of each fragment. Each coloured line represents a different fragment.

It can be seen that due to the way fragments are initialised, the mean *m/z* of each varies very little resulting in mostly straight lines across the plot. Where some lines are "blotchy", a fragment is being created and then deleted repeatedly.

Interactive plots of observations are useful when analysing the way in which observations from different spectra were grouped into fragments and motifs. Observations can be colour coded by fragment, spectrum or motif membership.

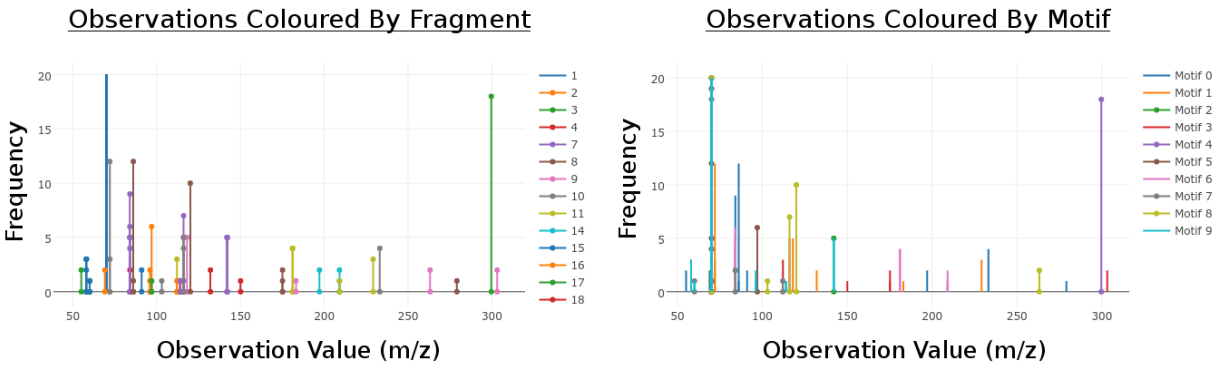

Figure 4.8: Two of the available visualisations showing all observations in a dataset coloured by fragment (left) and motif (right).

The plots seen in figure 4.8 are produced using the plotly library [4]. A considerable advantage of using plotly is the ability to interactively pan and zoom on the plot. This allows the researcher an overview of all fragments in a wide *m/z* range before zooming in to distinguish between two observations that are as little as 0.00005 *m/z* apart.

Finally, in order to gain an overview of the association between spectra and motifs, plots of the spectral makeup of each motif and the motif makeup of each spectrum can be produced. Figure 4.9 shows the set of spectra which compose a motif. Motif 2 is common, since it is shared amongst many spectra. The annotations on the "116.0703" fragment are part of the interactive plotly visualisation, available on hover.

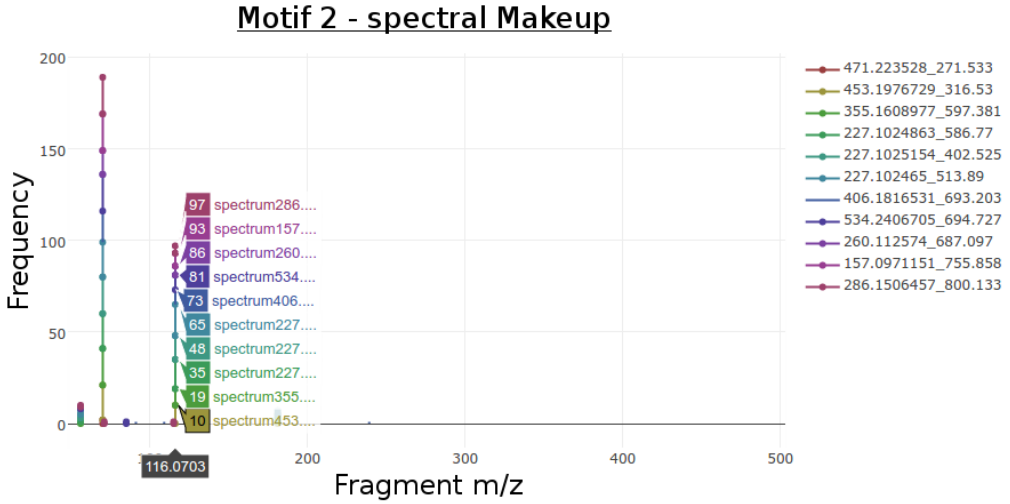

Figure 4.9: A visualisation of a single motif which is present in many spectra.

4.4.3 Object Oriented Design

During the development of the initial LDA implementation, it became clear that an object oriented approach to all of the different components within the model was needed. For this reason, the LDA and HDP models were

designed to share a common interface which allowed for migration between them by simply changing the class used.

The HDP is constructed out of 4 classes to make the interactions within the model easier to understand for researchers, along with facilitating the extraction and visualisation of data within the model:

- *Hdp* — The highest level class which contains all of the other subclasses. Responsible for storing all of the fragments defined in the vocabulary, sampling new fragments from the base distribution and storing all of the motifs.
- *Motif* — A single motif, responsible for assigning an observation to a grouping and sampling fragments from the HDP to assign to groupings.
- *Grouping* — A single grouping, responsible for tracking which observations are assigned to it, along with which motif it belongs to and which fragment it expresses.
- *Fragment* — A single fragment, responsible for tracking which observations are assigned to it via the groupings which express it.
- *Observation* — A single observation, which stores information on its *m/z* value, the spectrum it belongs to, and other optional information pertaining to how it was generated.

All classes other than the Observation class implement an *add_observation()* and *remove_observation()* function which simplifies the process of connecting the different layers in the hierarchy.

4.4.4 Continuous Testing Using Synthetic Data

Throughout the development process, the LDA and HDP models were tested continuously on synthetic data. The HDP was tested on trivial problems such as identifying that 5 fragments exist in a dataset, along with the values of these fragments. Interestingly, in figure 4.10 it can be seen that after some iterations the model on the left identified that the fragment of *m/z* ∼10 was actually two fragments of *m/z* 0 and 20.

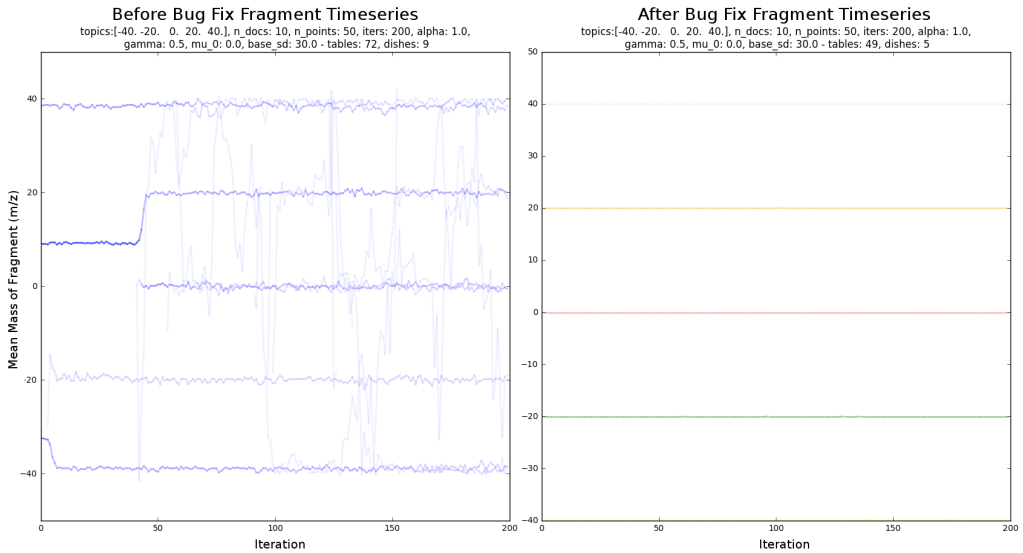

Figure 4.10: Two fragment time series plots for the model: one in which bug causes erratic behaviour (left), and one after the bug is fixed (right).

The code used to produce the synthetic data was refined into a set of classes and functions which generate relatively realistic mass spectrometry data based on a variety of parameters. This became an important tool for evaluating the HDP, was included in the hdplab library.

Chapter 5

Testing and Evaluation

5.1 Experiment with Synthetic Data

Aims An experiment was run on synthetically generated data with two aims. The first was to test the capabilities of the data generator. The second was to assess the HDP's ability to identify that observations originate from two distinct fragments due to their occurrence in different motifs.

Implementation The data for the experiment was designed to have a vocabulary of 8 fragments and 3 motifs. Using the synthetic data generator in *hdpgenerator.py* 15 spectra were generated with around 40 observations in each for a total of around 640 observations. A high alpha value was used for the model and data generation meaning that spectra were likely to be comprised of a mixture of the 3 topics rather than just 1 or 2.

```
from hdplab.hdpgenerator import generate_spectrum, Motif

# Vocabulary of fragments to generate observations around.
vocabulary = np.array([41.56, 57.02, 57.1, 105.20, 105.2007, 123.211, 201.8, 333.02])

motifs = []
# Generating some motifs with a vocabulary, fragment proportions and a label.
# The fragment proportions should sum to 1.
motifs.append(Motif(vocabulary, [0.40, 0.05, 0.00, 0.45, 0.0, 0.05, 0.02, 0.03], 0))
motifs.append(Motif(vocabulary, [0.00, 0.23, 0.27, 0.00, 0.40, 0.05, 0.01, 0.04], 1))
motifs.append(Motif(vocabulary, [0.05, 0.05, 0.00, 0.00, 0.00, 0.20, 0.35, 0.35], 2))

n_spectra = len(settings.spectrum_names)
n_obs = 40 # Number of observations per spectrum

# Iteratively generating the spectra from the topics, along with a number of fragments (words)
# per spectrum and some parmeters.
spectra = [generate_spectrum(motifs, min_obs=n_obs, max_obs=n_obs+5, delta=settings.delta,
                             alpha=settings.alpha, label = i) for i in range(n_spectra)]
```

Figure 5.1: The segment of code that allows for easy generation of synthetic data using the *hdplab* library.

It can be seen in figure 5.1 that the vocabulary contains fragments of *m/z* 105.20 and 105.2007. These fragments were particularly close together intentionally, so as to make it a challenge for the model to identify that they are unique fragments. It can also be seen when generating motifs 1 and 2 that these fragments do not co-occur in the same motif. This was done with the intention that it would allow the model to differentiate between the fragments.

A visualisation of all observations in the dataset can be seen in figure 5.2. Note that although the coloured lines look like single peaks, they are actually clusters of observations with very similar *m/z*. The variety of colours within in each peak indicate that it is present in many different spectra.

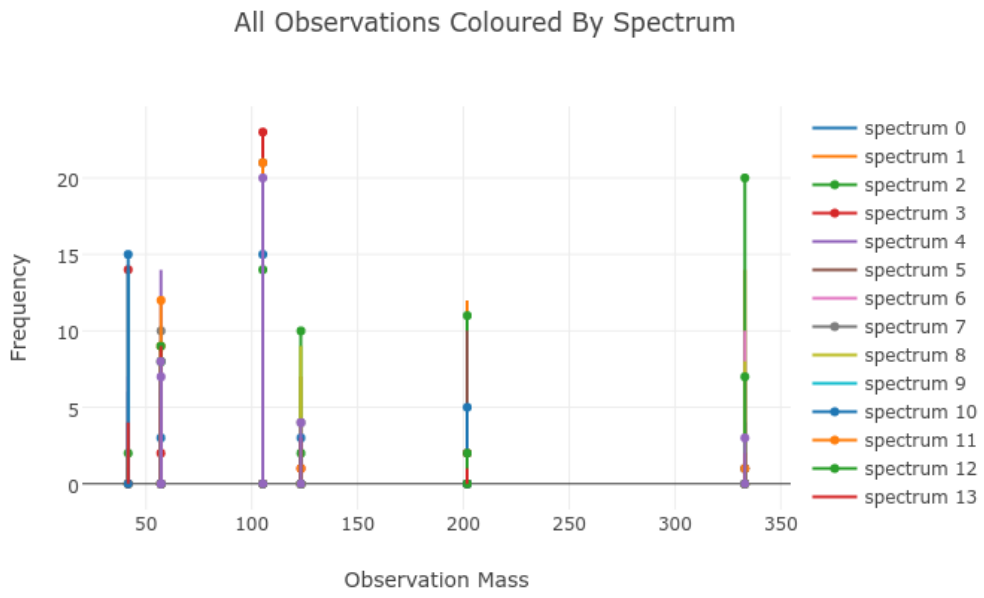

Figure 5.2: A visualisation of all observations in the synthetic data, coloured by the spectrum they belong to.

Results The experiment was run for 100 iterations and a log was produced as it progressed. Initially, the model identified only 7 of the 8 fragments, which continued for around 50 to 60 iterations. It can be seen in the experiment log in listing 5.1 that at around iteration 60, an additional fragment was added to the model's vocabulary, changing the number from 7 to 8. This indicates that the model detected that two sets of observations may belong to two separate fragments.

Listing 5.1: A portion of the experiment log for the synthetic data experiment, which gives the researcher live updates on the model's progress.

```
iter 30
fragments: 7   motif_n_groupings: 6 8 4   total groupings: 18
motif_n_obs 237 330 73 motif_tot_obs 640 group_tot_obs 640 f_tot_points 640
. . . . . .
iter 40
fragments: 7   motif_n_groupings: 6 7 4   total groupings: 17
motif_n_obs 232 284 124 mot_tot_obs 640 group_tot_obs 640 f_tot_points 640
. . . . . .
iter 50
fragments: 7   motif_n_groupings: 5 9 6   total groupings: 20
motif_n_obs 295 197 148 mot_tot_obs 640 group_tot_obs 640 f_tot_points 640
. . . . . .
iter 60
fragments: 8   motif_n_groupings: 6 11 5   total groupings: 22
motif_n_obs 277 238 125 mot_tot_obs 640 group_tot_obs 640 f_tot_points 640
. . . . . .
iter 70
fragments: 8   motif_n_groupings: 5 10 7   total groupings: 22
motif_n_obs 287 192 161 mot_tot_obs 640 group_tot_obs 640 f_tot_points 640
. . . . . .
```

Fragment 8 remained present for the remainder of the experiment. When the experiment had finished, multiple reports and visualisations were produced. The fragment breakdown in listing 5.2 shows that two 105.2∼ fragments were been identified (where the "tilde" character represents any number of additional decimal places). Additionally, the fragment time series visualisation displayed an additional coloured line which appeared in iteration 58.

Listing 5.2: The fragment breakdown generated after running the HDP on synthetic data.

```
fragment breakdown
==============
fragment 41.5601818899     pop: 71
fragment 57.0199478546     pop: 80
fragment 57.0999250206     pop: 81
fragment 105.200020189     pop: 73
fragment 105.200897755     pop: 141
fragment 123.210962757     pop: 54
fragment 201.799823845     pop: 57
fragment 333.019975974     pop: 83
```

Interactive plots of the observations coloured by fragment and motif were generated. The area of interest (around fragment 105.2∼) was zoomed in upon, as can be seen in figure 5.3.

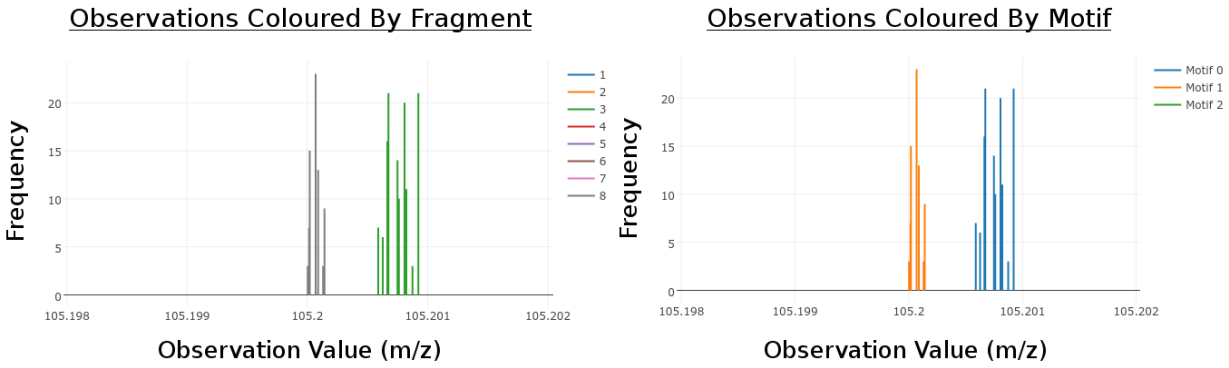

Figure 5.3: A highly zoomed in section from the visualisations of observations in the synthetic data experiment. Coloured by fragment (left) and motif (right) membership.

Evaluation It can be seen from the plots of observations coloured by fragments and motifs in figure 5.2 that not only were the points correctly divided between the two fragments, but they also ended up in a different motif.

This creation of fragment 8 after so many iterations provides a strong indication that the points began in a single fragment, and that they were split into two fragments since they occurred in separate motifs. This demonstrates the HDP model's ability to split observations into fragments based on motifs, which is not present in LDA since the preprocessing stage is executed before the model is applied. If this behaviour can be observed when HDP is applied to real world data, and if the fragments it finds turn out to be real molecules, it would mean that HDP uncovers different biochemically relevant information to LDA.

Additional points of interest during this experiment include a fragment 41.56, which was present in two motifs. When the synthetic motifs were created, fragment 41.56 was split between motif 1 and motif 3 with a probability of 0.40 and 0.05 respectively. In figure 5.4 observations coloured by motif can be seen post-experiment. Observations are split with 63 points in motif 1 and 8 points in motif 2. This ratio matches very

closely with the ratio of 0.40 to 0.05, which is a good indication that the model has identified motifs correctly and classified observations into them accurately. Note that the motif labels 1 and 3 in the synthetic data generator, and 1 and 2 in the posterior do not match. This is expected, since the model has no knowledge of the motifs used to generate the data, and has no concept of the ordering or labelling of motifs.

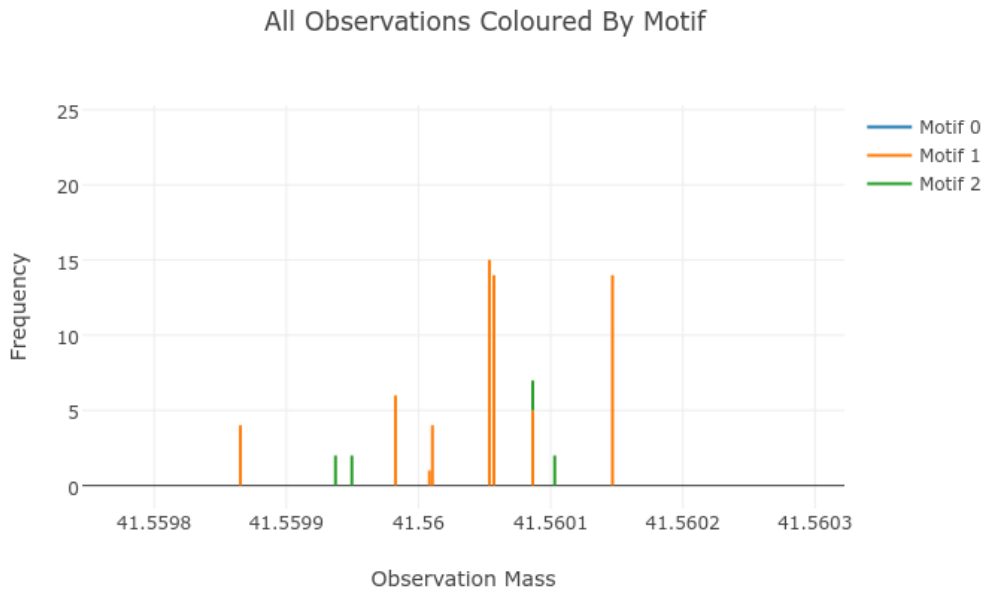

Figure 5.4: A highly zoomed in visualisation of some observations in the synthetic data. All of these points were generated by a single fragment, however, they are shared between two motifs.

Limitations While this experiment provided convincing evidence that the model is able to classify observations into fragments based on their co-occurrence with other observations, there are a few caveats.

Firstly, although the synthetic data is visually similar to real data when zoomed in, it is considerably less noisy, with fewer spurious data points than real mass spectrometry data. Secondly, the motifs which were chosen are simpler than real world motifs, and have considerably less fragments which overlap multiple motifs.

Finally, and importantly, the difference of 0.0007 between fragments 205.2000 and 205.2007 is larger than the grouping threshold of 7 parts-per-million necessary for the MS2LDA preprocessor. If the MS2LDA system were presented with this same data, it would automatically identify both of the fragments without having to discover it via machine learning like HDP does. However, as mentioned previously, most data presented to MS2LDA is noisier, with more observations interspersed. For this reason, observations may bridge the gap between 205.2000 and 205.2007, causing MS2LDA to group them into a single fragment, where HDP may still be able to discover that two fragments exist.

5.2 Experiment with Real Data

A real mass spectrometry dataset from samples of beer was analysed using the HDP. The data was provided in the format as outlined in section 4.1 on the mass spectrometry data format. Some of the challenges associated with analysing real world data were:

- The entire dataset was vast, and due to the limited computing resources and unoptimised nature of the HDP, subsets of the data had to be carefully chosen for use in experiments.
- When a subset of the dataset was chosen, some important information that would have aided in the discovery of motifs was inevitably left out. This meant that the motifs from the HDP did not exactly match those from MS2LDA.
- The hyperparameters for real world data are not known. For this reason, reasonable estimates were used. Defaults within the tool are reasonable estimates, and the data importing functionality automatically calculates some parameters such as μ_{base} and the base standard deviation. This is in alignment with best principles for metabolomics tool development [18].

Aim An experiment was devised for real world data in an attempt to split observations of *m/z* 116.07∼ into two separate fragments, and to show that these fragments are likely to be unique molecules. Fragment 116.07 was chosen since it had been previously split in other experiments, and on the MS2LDA website there exists only 1 fragment, suggesting it may have been erroneously grouped by the preprocessing stage.

Implementation By examining the .csv files, spectra were selected which had a high intensity of observations around a *m/z* of 116.07 and the experiment was run for 300 iterations.

Results The visualisations in figure 5.5 show all observations coloured by fragment and motif, which were used to assess how the observations had been grouped. By zooming in on *m/z* 116.07, it could be seen that the observations centred around *m/z* 116.0704 had been grouped into one fragment while observations centred around *m/z* 116.0711 had been grouped into another.

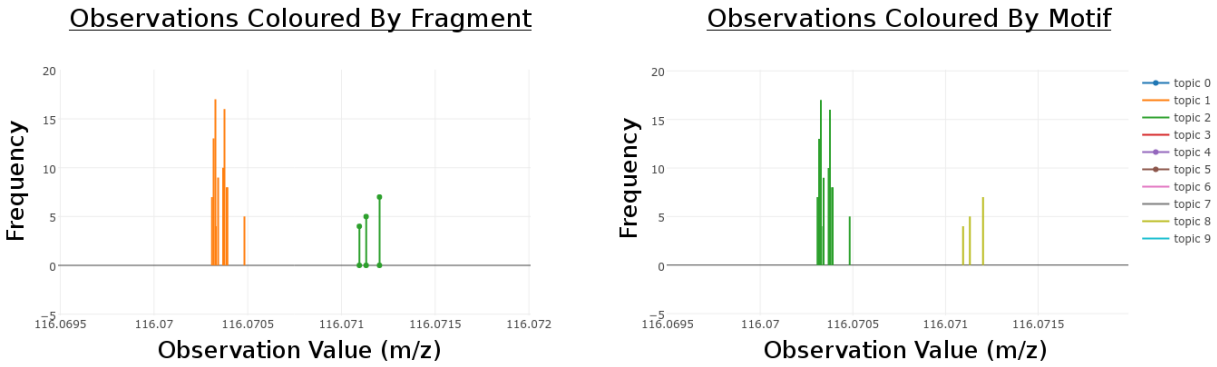

Figure 5.5: The visualisation of all real world observations coloured by fragment (left) and motif (right).

It could be seen from the "observations coloured by motif" visualisation that the observations which were grouped into separate fragments were also associated with different motifs.

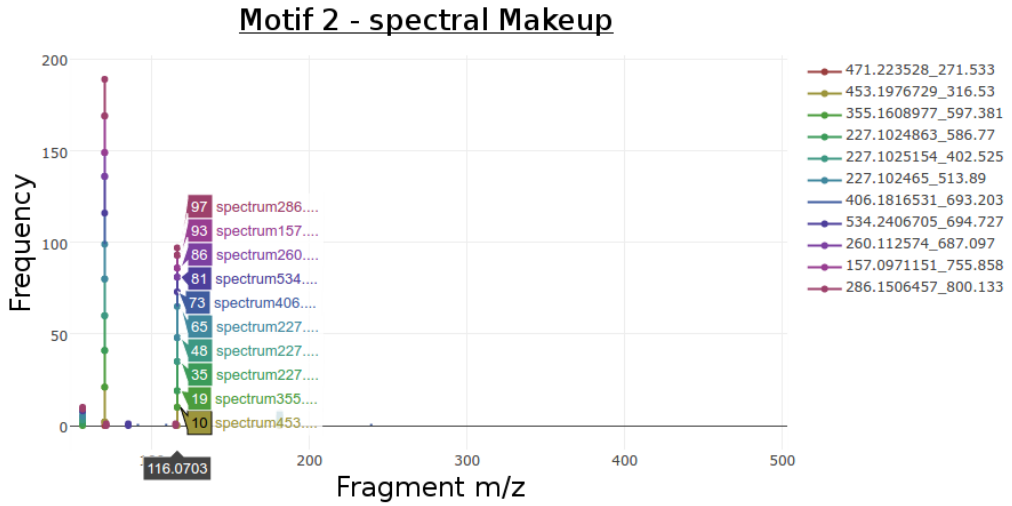

Figure 5.6: A figure showing the observations in motif 2. Portions of the peaks are coloured according to which spectrum the observations belong to.

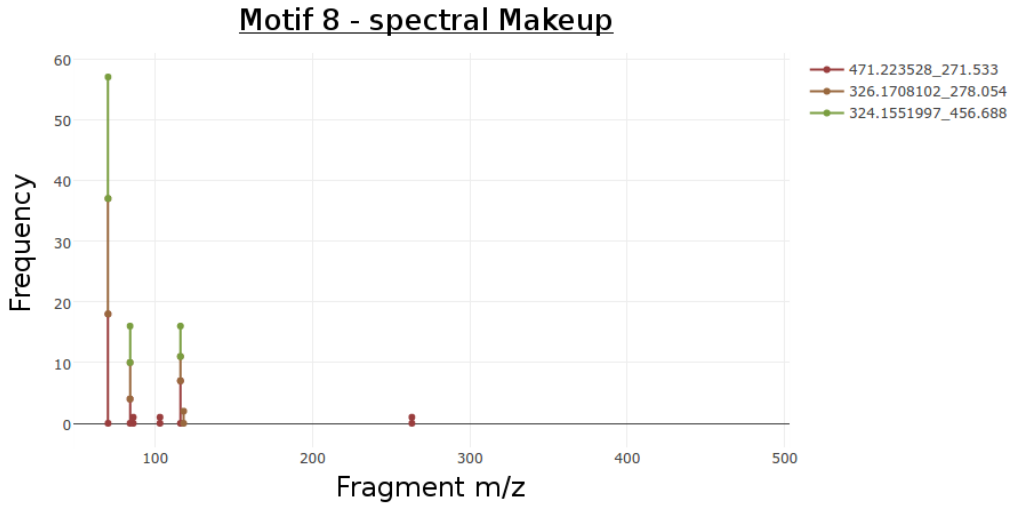

Figure 5.7: A figure showing the observations in motif 8. Portions of the peaks are coloured according to which spectrum the observations belong to.

Figures 5.6 and 5.6 show the makeup of motifs 2 and 8, which fragments 116.0704 and 116.0711 were associated with. Interestingly, in motif 2 fragment 116.0704 co-occurred regularly with fragment 181.09, whereas in motif 8 fragment 116.0711 co-occurred with 84.04. Fragment 70.06 was present in both motifs since within the dataset fragment 70.06 was in the vast majority. Motifs 2 and 8 also have only one document in common. Evaluation In figure 5.5, where observations in separate fragments were also in separate topics, is interesting since it suggests that the HDP did not find a trivial solution. An example of a trivial solution might be that the observations were separated into two fragments, but the fragments always co-occur in the same motifs, essentially acting as a single fragment.

The fact that the spectra within motifs 2 and 8 have little overlap is of note, since it indicates that they may share common substructure which the other spectra do not. This is also an indicator that the HDP did not find a trivial solution whereby it creates multiple motifs which are all associated with the same set of spectra. Some evidence that the HDP finds useful motifs is that some are comparable to ones found using the LDA model

within MS2LDA. By browsing motifs associated with the spectra used in the experiment, MS2LDA motif 257 was found to be similar to the HDP motif 2. Both were comprised mainly of fragments 70.06, 116.07 and 181.09. The MS2LDA model estimated the probability of these three fragments within the motif were 0.60, 0.15 and 0.11 respectively, whereas the HDP estimated them to be 64, 0.32 and 0.02. The bias towards fragment 116.07 rather than 181.09 can be partly accounted for by the fact that only a subset of the total data was chosen specifically to contain many observations of *m/z* 116.07∼.

However, there is reason to believe that the "newly discovered" fragment at *m/z* 116.0711 is in fact not a real fragment, and instead an anomaly of the HDP. A script designed to find plausible chemical formulae for sets of *m/z* values was run on the dataset. The program found no feasible matches for fragment 116.0711, but did find one at *m/z* 116.0704. Although this is not definitive proof, since the search run by the script is not exhaustive, it may be an indication that observations of *m/z* 116.0711 may not be a fully fledged fragment.

5.3 Evaluation of Experiments

As can be seen in the results sections of both experiments, no conclusive evidence was found that HDP can identify previously undiscovered fragments. However, these experiments did provide some useful insights into the other capabilities of HDP.

The experiment with synthetic data showed that HDP was able to identify fragments due to patterns in co-occurrences between observations, rather than simply by grouping observations together based on proximity. This is a promising sign that the model could be used to identify complex underlying structure in real world data. It also displayed the ability to divide observations of fragments between motifs in the correct ratio, according to which other fragments they occur alongside.

The experiment with real data showed that the HDP found two possible fragments which had previously been grouped together by the preprocessing step in MS2LDA. The analyses of the experiment revealed that the fragments were associated with different motifs which was a strong indication that the HDP had found a non-trivial solution. Additionally, motif 2 in the HDP was similar in content to motif 257 in MS2LDA, which is an indication that the HDP is working as expected, and finds useful solutions. It may also be an indication that both LDA and the HDP are robust enough to find correlations between fragmentation spectra, even when observations are grouped into too many or too few fragments.

5.4 Future Improvements

Although the results of the experiments were inconclusive, they have demonstrated the capability of the HDP machine learning model along with the surrounding experimentation and visualisation framework. I believe these tools would be sufficient, given additional time for experimentation, to answer the question of whether the infinite vocabulary HDP can extract additional biochemically relevant information, due to its observation clustering abilities. There are some additional pieces of functionality I believe would make the experiment more comprehensible and usable for a researcher.

Considerable performance optimisations could be made to the underlying python code. There are several places where calculations performed 100s of times per iteration could be cached, and therefore only performed once per iteration. Another significant performance optimisation would be to allow multiple observations to be reassigned to a group simultaneously, allowing for the more efficient discovery of new fragments. These

performance enhancements would make experiments faster to run, and would therefore increase the usability of the system since researchers could iterate more rapidly on ideas/experiments.

The ability to measure the distance between observations or fragments in terms of PPM is a standard used regularly amongst researchers in the field [19] and would be a useful addition to the suite of visualisation tools. An additional useful visualisation would be an interactive graph where motifs and documents are represented as interconnected nodes. With this visualisation, researchers could more easily compare the results of LDA and HDP, allowing them to identify similarities in the motifs each of the models finds.

Dynamic numbers of motifs would be a useful addition for the analysis of large real world datasets. A current limitation of the HDP is that a finite number of motifs needs to be defined before running the model on a dataset. This is not an issue for known datasets since a larger number can be specified, and the HDP can leave any unneeded motifs empty. However, in larger datasets with an unknown number of motifs, the only feasible solution would be to allow the number of motifs to expand and contract in the same way that the number of fragments can. This would be relatively simple to implement due to the modular, object-oriented structure of the project.

5.5 Conclusion

In conclusion, the aim of this project was to implement a proof of concept infinite vocabulary HDP and assess its ability to group observations in the mass spectrometer into fragments. This report has given a very brief overview of the field of metabolomics and outlined the way in which the LDA machine learning model can be applied using MS2LDA to discover shared substructure between molecules. It also offered an explanation of the closely related HDP model, and discussed the implementation of the HDP model using python and numpy.

An overview was given of how an experimental framework was built around the HDP and integrated with jupyter notebooks, making the model more easily testable for researchers. Two sets of experimental results were presented and discussed. Although no conclusive evidence was found that HDP can produce more useful insights into the metabolomics data than LDA, it was shown that it does possess the ability to separate observations into fragments based on their motif membership — a feature not present in LDA. It was also shown that both LDA and HDP find similar motifs within the dataset even when observations are grouped into fragments differently. This may indicate that both algorithms are robust enough to do topic modelling in spite of erroneous observation grouping, and that the LDA algorithm can recover from bad feature selection.

The presentation and analysis of the experimental results demonstrated the set of visualisations and reports which can be used to gain insight into the way the HDP model performs. Some future improvements for the experiment toolkit and the HDP model were also outlined. I believe that with these future improvements, the performance of the HDP model can be definitively assessed. This will allow researchers to answer the important question of whether HDP unveils additional insights into the metabolomics data, or whether LDA alone is robust enough to overcome poor grouping.

Appendices

Appendix A

Readme

A short user manual, with guidance on setup and how to use the toolkit, is provided as a Readme.md file along with the code. A printout of this ”readme” to PDF format can be seen in figure A.1.

HDP Lab

An implementation of a Hierarchical Dirichlet Process model designed for the processing and subsequent analysis of metabolomics data (in the form of fragmentation spectra).

Files

The files and folder, along with their purposes are listed below:

- *data/* - stores the mass sectrometry data which can be loaded and analysed.
- *experiment saves/* - contains pickled experiment objects which can be loaded to run more iterations or analyse.
- *Test reports/* - contains some hand-gathered dumps of data from previous experiments. not of much interest.
- *README.md* - this readme file.
- *EXPERIMENT TEMPLATE - clone me.ipynb* - a python notebook containing an experiment template which can be clones, and the experiment settings can be changed.
- *EXAMPLE - generating and analysing synthetic data.ipynb* - an example of the synthetic data generating functionality (discussed in dissertation).
- *TEST 1 - attempting to split frag 70.06~.ipynb* - A test run.
- *TEST 2 - attempting to split frag 116.07~.ipynb* - A test run (discussed in dissertation).
- *TEST 3 - splitting frag 209.09~.ipynb* - A test run.

Usage

To use this software, start a jupyter server in the current directory. For a guide on how to do this, visit: `http://jupyter.org/install.html` (an installation via "conda" is recommended). This will give you access to the notebooks. The notebooks rely on some standard maths, statistics and visualisation libraries including `numpy`, `scipy`, `matplotlib`, and `plotly`. Most should come preinstalled with an installation of jupyter. the plotly library can be installed via `pip` or `conda` by typing `pip install plotly` or `pip install conda` respectively.

As mentioned above, the "EXPERIMENT TEMPLATE - clone me" notebook can be cloned to produce your own experiments. Instructions on how to change the experiment settings can be found within.

For an example of an experiment generating and analysing synthetic data, refer to the "EXAMPLE - analysing synthetic data" notebook.

What to Try Out

To get an overview of the functionality of the HDP Lab, try viewing one of the existing experiments such as *TEST 2 - attempting to split frag 116.07~.ipynb*. This experiment was featured in the dissertation. In this notebook, find the "Observations coloured by fragments" plot, and zoom in on the m/z range of 116.07. You should be able to see that there are actually many peaks situated very closely together.

Additional things to try, might be to run an experiment. To do this, as mentioned in the "Usage" section, clone the template notebook. Run an experiment with a very small number of iterations (maybe 10 or 20) to start with, and with very few spectra (maybe 2 or 3). This should take under 10 minutes to process on a regular laptop.

To change settings such as the number of iterations, or the set of spectra used, change the parameters in the ExperimentSettings object constructor. When changing parameters of the model such as alpha and delta, *ensure that the number is a float datatype* (ie. has a .0 on the end), to prevent issues with division by integers.

To run the experiment within the notebook, go to the "kernel" menu and select "restart & run all".

Figure A.1: A PDF version of the readme.md user guide which accompanies the source code for the project.

Bibliography

[1] Jupyter notebook. jupyter.org/about.html.

[2] Python numpy library. www.numpy.org/.

[3] Python pickle library. docs.python.org/2/library/pickle.html.

[4] Python plotly library. plot.ly/python/getting-started/.

[5] Python scipy library. www.scipy.org/about.hmtl.

[6] Mcculloch R Allenby G, Rossi P. Hierarchical bayes models: A practitioners guide hierarchical bayes models. *SSRN*, pages 3, 6, 2005.

[7] Jordan M Tenenbaum J Blei D, Griffiths T. Hierarchical topic models and the nested chinese restaurant process. *NIPS*, 2003.

[8] Ng A Edu A Jordan M Edu J Blei D, Edu B. Latent dirichlet allocation. *Journal of Machine Learning Research volume 3*, pages 995, 996, 997, 1003, 2003.

[9] Cutillas P Heck A Van Breukelen B Cappadona S, Baker P. Current challenges in software solutions for mass spectrometry-based quantitative proteomics. *Amino Acids*, pages 1096, 1098, 2012.

[10] Shen X Rosen G Chen X, Hu X. Probabilistic topic modeling for genomic data interpretation. *BIBM 2010 IEEE International*, page 2, 2010.

[11] Jim Clark. Frag spectrum of pentane. www.chemguide.co.uk/analysis/masspec/fragment.html, 2000.

[12] Weber R Creek D Brown M Breitling R Hankemeier T-Goodacre R Neumann S Kopka J Viant M Dunn W, Erban A. Mass appeal: Metabolite identification in mass spectrometry-focused untargeted metabolomics. *Metabolomics*, 2013.

[13] Eyesopen. Eyesopen fragmentation image. docs.eyesopen.com/.

[14] Steyvers M Griffiths T. Finding scientific topics. *PNAS*, page 2, 2004.

[15] Bach F Hoffman M, Blei D. Online learning for latent dirichlet allocation. pages 2, 6, 2010.

[16] Hufsky F Scheubert K Böcker S. Computational mass spectrometry for small-molecule fragmentation. *TrAC - Trends in Analytical Chemistry*, pages 1, 47, 2014.

[17] Fiehn O Hankemeier T Kristal B van Ommen B Pujos-Guillot E Verheij E Wishart D Wopereis S Scalbert A, Brennan L. Mass-spectrometry-based metabolomics: Limitations and recommendations for future progress with particular focus on nutrition research. *Metabolomics*, pages 442, 445, 2009.

[18] Ventura D Prince J Smith R, Mathis A. Proteomics, lipidomics, metabolomics: a mass spectrometry tutorial from a computer scientist's point of view. *From The 10th Annual Biotechnology and Bioinformatics Symposium*, pages 2, 5, 7, 8, 12, 2013.

[19] Engineering Toolbox. Parts per million (ppm). www.engineeringtoolbox.com/ppm-d_1039.html.

[20] Barrett M Burgess K Rogers S Van Der Hooft J, Wandy J. Topic modeling for untargeted substructure exploration in metabolomics. *PNAS*, pages 1–5, 2016.

[21] Saghatelian A Vinayavekhin N. Untargeted metabolomics. *Current Protocols in Molecular Biology*, 2010.

[22] Beal M Blei D Whye Teh Y, Jordan M. Sharing clusters among related groups: Hierarchical dirichlet processes. *NIPS*, pages 3, 4, 2004.

[23] Wikipedia. Dirichlet processes. https://en.wikipedia.org/wiki/Dirichlet_process, 2014.

[24] Whye Teh Y. Dirichlet process.