Chapter 1

Background Theory

1.1 Introduction

Automatic Text Classification or Text Categorisation is a rapidly growing field in Machine Learning and Natural Language Processing. This is mainly due to the amount of electronic data we currently generate. The main task is to assign one or more classes to a given text document. Some applications of text classification include *Email Spam Detection* and *Language Detection*. The former involves trying to distinguish spam emails from legitimate ones while the latter involves the identification of the language a document was written in.

However, this study makes use of classification techniques for data filtration which involves removing irrelevant documents from a list of documents(similar to spam filtering) and topic modelling (extracting topics from a list of documents) and sentiment analysis (predicting the sentiment of the author of a document). This chapter explains a few background concepts and reviews some relevant research previously done in this area.

1.2 Naïve Bayes Classifier

The Naïve Bayes classifier is one of the simplest classifier that can be used and this is due to the fact that it is based on simple Bayes Theorem. It is a probabilistic classifier which assumes that all features of the documents are independent of each other. This means that if a document

has features $f1$ and $f2$ (could be length of document, occurrence of words e.t.c), the existence of $f1$ has nothing to do with the existence of $f2$ and vice versa. This also means that it makes assumptions that may or may not be correct, hence the "Naïve" in its name.

Bayes theorem states that the probability of A given B is the probability of B given A times the probability of A divided by the probability of B. Mathematically, this is written as:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)} \tag{1.1}$$

Applying this logic to text classification, the probability that a document $d_i \in D$ belongs to a class c is denoted as:

$$p(c|d_i) = \frac{p(d_i|c)p(c)}{p(d_i)} \tag{1.2}$$

Although other techniques like Maximum Entropy, Random Forests or Support Vector Machines tend to perform better, a naive Bayes classifier will require less memory and CPU cycles. Furthermore, it is computationally less complex and simpler to implement. With regard to performance, Huang *et al.* (2003) showed using multiple datasets from Blake & Merz (1998) that the naive Bayes classifier in many cases performs as good as other complex classifiers and Zhang (2004) goes further to explain why it performs well. Other studies have also found Bayesian classifiers to be effective without being affected by its simple independence assumption (Langley *et al.* , 1992; Manning *et al.* , 2008).

The Naïve Bayes classifier has been used in many text classification problems but one of its common applications which is relevant to tweet classification is email spam[1] filtering. A spam filter is a system that takes in text and decides whether or not it is spam. Androutsopoulos *et al.* (2000) addressed this issue using a naive Bayes classifier. They trained the model using a predefined set of manually labelled messages. They were able to show that the naive Bayes classifier was capable of classifying messages with impressive accuracy and precision compared to the then common keyword based approach to classification. Deshpande *et al.* (2007) also carried out a similar research and the results were equally impressive and similar.

[1] irrelevant or unsolicited messages. They are typically to large numbers of users

1.3 Topic Modelling

Topic Modelling is a process by which abstract topics/themes are extracted from a collection of documents. This process is usually carried out with the aid of topic models, a suite of algorithms used for topic modelling. It has been applied in a variety of fields like Software Analysis where Linstead *et al.* (2009) used topic modelling to find topics embedded in code and Gethers & Poshyvanyk (2010) used topic modelling to capture coupling among classes. Kireyev *et al.* (2009) applied topic models on disaster related data from Twitter in an effort to determine what topics were discussed within the time span of a natural disaster. Hospedales *et al.* (2009) introduced a new topic model that can be used to analyze videos with complex and crowded scenes in other to discover regularities in the videos. A system built on such model will be able to answer a question like "What interesting events happened in the last 5 hours". Other fields include Audio Analysis (Smaragdis *et al.* , 2009), Influence modelling (Gerrish & Blei, 2009), Finance (Doyle & Elkan, 2009), Writer Identification (Bhardwaj *et al.* , 2009) and many more.

There are a number of topic models but the two main ones are ***Latent Semantic Indexing*** (LSI) and ***Latent Dirichlet Allocation*** (LDA) and we discuss them further in the following sections.

1.3.1 Latent Semantic Indexing

Latent Semantic Indexing(LSI) (Hofmann, 1999), sometimes referred to as *Latent Semantic Analysis*, is an indexing technique that leverages matrix-algebra computations[2] to identify any patterns in relationships between a collection of text documents. It works based on the assumption that words used in the same context tend to have homogeneous meanings (Deerwester *et al.* , 1990; Dumais, 2004; Landauer, 2006). LSI, has been used mostly in Information Retrieval and Search Engine Optimisation where it tries to figure out what words in a web page are relevant to the web page even though they might not be used in that page. One of the main drawbacks the LSI model suffers from is ambiguity.

Assuming we have two documents, one talking about Microsoft Office and the other talking about actual physical office space. How can the model differentiate between the two? Unfor-

[2]Specifically, it uses Singular Value Decomposition which is a factorization of a complex matrix. See `http://en.wikipedia.org/wiki/Singular_value_decomposition`

tunately, it is unable to differentiate between such topics and a significant step to solve this problem was made by Hofmann (1999) who presented the probabilistic LSI model. Blei *et al.* (2003) argues that while Hoffman's work is a very useful step towards using probabilistic models to model text, it is incomplete.

1.3.2 Latent Dirichlet Allocation

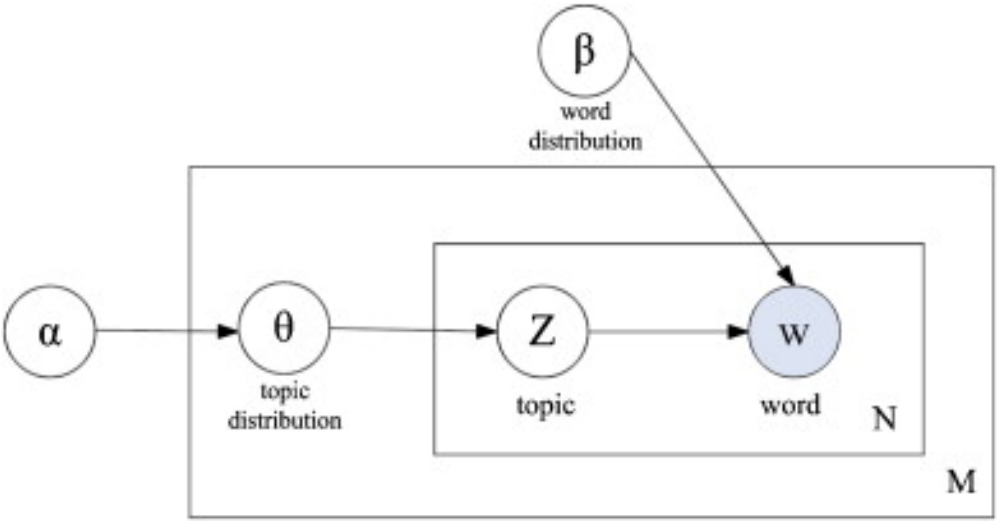

Figure 1.1: A graphical model representation of LDA (Courtesy: `http://victorfang.wordpress.com/2012/03/11/latent-dirichlet-allocation`)

Latent Dirichlet Allocation(LDA) is a generative[3] and probabilistic model that can be used to automatically group words into topics and documents into a mixture of topics (Blei *et al.* , 2003). It works based on the assumption that each document contains one or more topics. Words can also exist in multiple topics as they actually do in natural language. In other to tackle the problem of ambiguity LSI suffers from, LDA takes a combination of all topics that seem relevant to a document in a corpora[4] and compares that document to the topics in an effort to determine which topic is closer to the document. Figure 1.1 shows a graphical model representation of Latent Dirichlet Allocation. The inner boxes represents the choice of topics and words within a document while the outer box represents the actual documents.

Hoffman *et al.* (2010) developed a variant of LDA called Online LDA which uses variational Bayes as its posterior inference algorithm as opposed to Gibbs Sampling. It also allows the

[3]See `http://en.wikipedia.org/wiki/Generative_model`
[4]Corpora is simply a large collection of documents

model to be updated with more data after initial training. During initial training, the entire corpora is observed/trained in batches rather than at once. Asuncion *et al.* (2009) shows that although this model uses constant memory and it converges quicker, it still requires a full pass through the entire corpora. This makes it very slow when applied to large datasets.

An oversimplified version of the algorithm is:

while *model is yet to converge* **do**
  Data: B = randomly selected mini-batch of documents;
  for $b \in B$ **do**
    Estimate approximate posterior over what topics each word in each document came from;
    Update posterior over topic distributions based on what words are believed to have come from what topics;
  end
end

Most of the research done on social media data, especially Twitter, has been to detect usage and communities (Java *et al.* , 2007). Nonetheless, recent research has started to look into the detection of topics in social media. Kireyev *et al.* (2009) used LDA to extract topics/themes from a collection of disaster related tweets. Zhao *et al.* (2011) used LDA to compare news related tweets on Twitter with topics in The New York Times. They were also able to show that the standard LDA might not always work well on tweets and so they proposed a new model which is a slight variant of LDA. Weng *et al.* (2010) proposed an algorithm that leverages LDA to find topic-sensitive influential twitter users. Lau *et al.* (2012) presented an LDA-based model for detecting and tracking emerging trends/events on microblogs like Twitter.

References

Androutsopoulos, Ion, Paliouras, Georgios, Karkaletsis, Vangelis, Sakkis, Georgios, Spyropoulos, Constantine D, & Stamatopoulos, Panagiotis. 2000. Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *arXiv preprint cs/0009009*.

Asuncion, Arthur, Welling, Max, Smyth, Padhraic, & Teh, Yee Whye. 2009. On smoothing and inference for topic models. *Pages 27–34 of: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Bhardwaj, Anurag, Malgireddy, Manavender, Setlur, Srirangaraj, Govindaraju, Venu, & Ramachandrula, S. 2009. Writer identification in offline handwriting using topic models. *In: Proceedings of the NIPS 2009 Workshop on Applications of Topic Models: Text and Beyond*.

Blake, Catherine L, & Merz, Christopher J. 1998. UCI Repository of machine learning databases [http://www.ics.uci.edu/˜mlearn/MLRepository.html]. Irvine, CA: University of California. *Department of Information and Computer Science*, **460**.

Blei, David M, Ng, Andrew Y, & Jordan, Michael I. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, **3**, 993–1022.

Deerwester, Scott C., Dumais, Susan T, Landauer, Thomas K., Furnas, George W., & Harshman, Richard A. 1990. Indexing by latent semantic analysis. *JASIS*, **41**(6), 391–407.

Deshpande, Vikas P, Erbacher, Robert F, & Harris, Chris. 2007. An evaluation of Naive Bayesian anti-spam filtering techniques. *Pages 333–340 of: Proceedings of the 2007 IEEE Workshop on Information Assurance United States Military Academy, West Point, NY*.

Doyle, Gabriel, & Elkan, Charles. 2009. Financial topic models. *In: NIPS 2009 Workshop on Applications of Topic Models: Text and Beyond*.

Dumais, Susan T. 2004. Latent semantic analysis. *Annual review of information science and technology*, **38**(1), 188–230.

Gerrish, Sean, & Blei, David. 2009. Modeling Influence in Text Corpora.

Gethers, Malcom, & Poshyvanyk, Denys. 2010. Using relational topic models to capture coupling among classes in object-oriented software systems. *Pages 1–10 of: Software Maintenance (ICSM), 2010 IEEE International Conference on*. IEEE.

Hoffman, Matthew D, Blei, David M, & Bach, Francis R. 2010. Online Learning for Latent Dirichlet Allocation. *Page 5 of: NIPS*, vol. 2.

Hofmann, Thomas. 1999. Probabilistic latent semantic indexing. *Pages 50–57 of: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM.

Hospedales, Timothy, Gong, Shaogang, & Xiang, Tao. 2009. A markov clustering topic model for mining behaviour in video. *Pages 1165–1172 of: Computer Vision, 2009 IEEE 12th International Conference on*. IEEE.

Huang, Jin, Lu, Jingjing, & Ling, Charles X. 2003. Comparing naive Bayes, decision trees, and SVM with AUC and accuracy. *Pages 553–556 of: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE.

Java, Akshay, Song, Xiaodan, Finin, Tim, & Tseng, Belle. 2007. Why we twitter: understanding microblogging usage and communities. *Pages 56–65 of: Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*. ACM.

Kireyev, Kirill, Palen, Leysia, & Anderson, K. 2009. Applications of topics models to analysis of disaster-related twitter data. *In: NIPS Workshop on Applications for Topic Models: Text and Beyond*, vol. 1.

Landauer, Thomas K. 2006. Latent semantic analysis. *Encyclopedia of Cognitive Science*.

Langley, Pat, Iba, Wayne, & Thompson, Kevin. 1992. An analysis of Bayesian classifiers. *Pages 223–228 of: AAAI*, vol. 90.

Lau, Jey Han, Collier, Nigel, & Baldwin, Timothy. 2012. On-line Trend Analysis with Topic Models:\# twitter Trends Detection Topic Model Online.

Linstead, Erik, Hughes, Lindsey, Lopes, Cristina, & Baldi, Pierre. 2009. Software analysis with unsupervised topic models. *Page 52 of: NIPS Workshop on Application of Topic Models: Text and Beyond*, vol. 50.

Manning, Christopher D, Raghavan, Prabhakar, & Schütze, Hinrich. 2008. *Introduction to information retrieval*. Vol. 1. Cambridge University Press Cambridge.

Smaragdis, Paris, Shashanka, Madhusudana, & Raj, Bhiksha. 2009. Topic Models for Audio Mixture Analysis. *Applications for Topic Models: Text and Beyond, Whistler*.

Weng, Jianshu, Lim, Ee-Peng, Jiang, Jing, & He, Qi. 2010. Twitterrank: finding topic-sensitive influential twitterers. *Pages 261–270 of: Proceedings of the third ACM international conference on Web search and data mining*. ACM.

Zhang, Harry. 2004. The optimality of naive Bayes. *A A*, **1**(2), 3.

Zhao, Wayne Xin, Jiang, Jing, Weng, Jianshu, He, Jing, Lim, Ee-Peng, Yan, Hongfei, & Li, Xiaoming. 2011. Comparing twitter and traditional media using topic models. *Pages 338–349 of: Advances in Information Retrieval*. Springer.