PROBABILITY OF CO-OCCURRENCE WITH IRREGULAR DEPENDENCY

Ashirbad Mishra, Monalisha Pattnaik

1 Introduction

In this era, we are surrounded by an unprecedented abundance of information. The overwhelming volume of textual data on online platforms such as social media and e-Commerce sites, needs to be processed to get valuable insights such as the preferences and sentiments of users of those platforms. Recommendation systems have emerged as indispensable guides, steering users through the labyrinth of information overloaded by presenting them with tailored and pertinent content suggestions. The heart of recommendation systems lies in the vital task of text classification, which forms the foundation for the processing of contextual information. Over the years, various text classification techniques [1] have been developed, ranging from traditional statistical methods to deep learning algorithms.

Short text classification refers to the categorization of short textual input, typically comprising one or two sentences, such as tweets on Twitter, product titles, or user ratings on platforms like Amazon. This particular form of classification holds great value in discerning user sentiment, evaluating product demand, and even detecting fraudulent activities. This process is a bit challenging as unlike document classification there isn't enough contextual information for each input. One of the earliest strategies was to use probability methods for classification to assign class labels to text documents. These methods rely on statistical models that estimate the probability of a document belonging to a particular class or category.

While deep neural methods [2] often excel in capturing complex patterns and achieving high accuracy, probabilistic methods offer advantages such as interpretability, computational efficiency, and ease of implementation [3]. In scenarios where computational resources are limited or explainability is crucial, simpler and lightweight probabilistic methods provide a practical and effective alternative. Here, we show a conventional probabilistic approach to classifying short texts utilizing Bayesian network modeling [4] with n-gram language model. Additionally, we discuss the latest developments in this domain and explore potential avenues for future enhancements.

2 Probability of the Co-occurrence and Bayesian Inference

Probabilistic methods in an n-gram language model primarily depends on the co-occurrence of the n-grams or words in the input text. The label associated with each input is considered agnostic and associations between the label and combinations of the words in the text are crucial for modeling. We first describe the notations required for understanding the method and explain the terms involved.

2.1 Terms and Notation

We consider two sets of data, training T based on which the model is determined and test V on which inference is performed. Let's consider each input text as a t belonging to a training set T. Each input t consists of x1, x2, ..., xl, where the xi's are words tokenized from the text t. The unique set of all words is X and unique set of all labels is Y . Each t ∈ T, V is associated with a y ∈ Y . The goal is to use the generated model M to determine relevant y's associated with the input text in V .

2.2 Bayesian Formulation with Independence Assumption

Let's consider a table based on the Cartesian product X ×Y . This table relates all unique words across all input texts to all unique labels. Each cell (x, y) indicates the frequency of co-occurrence (Cx,y) of a word x ∈ X and a label y ∈ Y . We define the probability of co-occurrence of the word x given the label y as,

$$P(x|y) = \frac{C_{xy}}{C_{*y}} \qquad (1)$$

where $C_{*y} = \sum_{\forall x \in X} C_{xy}$ is the sum of column y. For a list of co-occurring words $\{x_1, x_2, ..., x_l\} \in t$ the probability that a label y is associated with those words can be formulated using the Bayes method as,

$$P(y|x_1, x_2, ..., x_l) = \frac{P(x_1, x_2, ..., x_l|y)P(y)}{P(x_1, x_2, ..., x_l)} \quad (2)$$

The above equation 2 gives us a way to order the different y' s co-occurring with the words in t in the training set T. The higher probability labels are more likely to co-occur along with those group of words and thus more relevant to them. However, computing P(x1, ..., xl|y) in equation 2 requires storing all combinations of words co-occur with a label for all the labels in the training set. As this is infeasible in this scenario so we can substitute it with the simplified form of the Bayes equation, the Na¨ıve Bayes formulation. Na¨ıve Bayes assumes independence among the xi' s, as in words co-occur independently from each other. This assumption transforms the equation 2 as,

$$P(y|x_1, x_2, ..., x_l) = P(x_1|y)...P(x_l|y) \cdot P(y) \quad (3)$$

Note that we have removed the denominator. This is done as the denominator will be common for the same group of words and thus removing it would not impact the comparison between the different labels. For a simpler computation we can reorganize the equation 3 by substituting equation 1 in it to get

$$P(y|x_1, x_2, ..., x_l) = P(y) \cdot \prod_{i=1}^{l} \frac{C_{x_i y}}{C_{*y}} \quad (4)$$

The probability P(y) can be computed directly from the frequency of occurrences of y. The inference is quite simple, for a given input text t ∈ V , its words are tokenized and mapped to the words in set X. Then probability in equation 3 is computed based on the values in the table.

The independence assumption allows us to create the simplistic table as mentioned above. Although the sizes of the set X and Y can be quite large for real-world datasets, the table size is relatively small. This is because many words and label combinations simply don' t occur in practical applications. This makes it more efficient to store the table in memory thus enabling faster computation. The figure 1 shows a Bayesian network employing the independence assumption Here we use the same formulation as in section 2.1. The vertices represent the elements in the Bayesian formulation and the edges represent the dependency between the elements. In this case the words x1 to xl seem to be dependent or co-occur with the label y. Due to the independence assumption co-occurrence or edges between the words are not present.

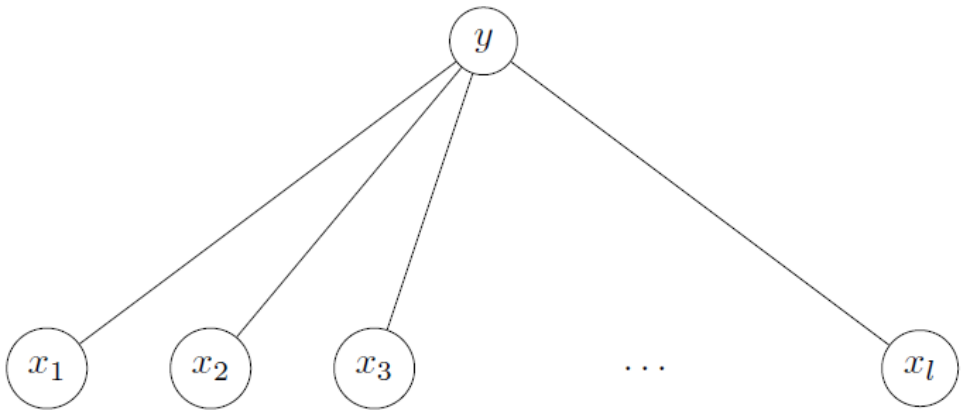

Figure 1: Bayesian network with Independence assumption

2.3 Bayes Method with Markov assumption

Bayesian networks provide a visualization of the probabilistic model and yield insights that are conducive to modeling. Similar to the figure 1 if we remove the independence assumption the Bayesian network will look like the one in figure 2. The complexity of the probabilistic evident from the equation 2 by expanding it into,avoiding the denominator for conciseness which is also irrelevant for comparison.

$$P(x1, x2, \ldots, xl, y) = P(y)P(x1|y)P(x2|x1, y)\ldots P(xl|x1, x2, \ldots, xl-1, y) \quad (5)$$

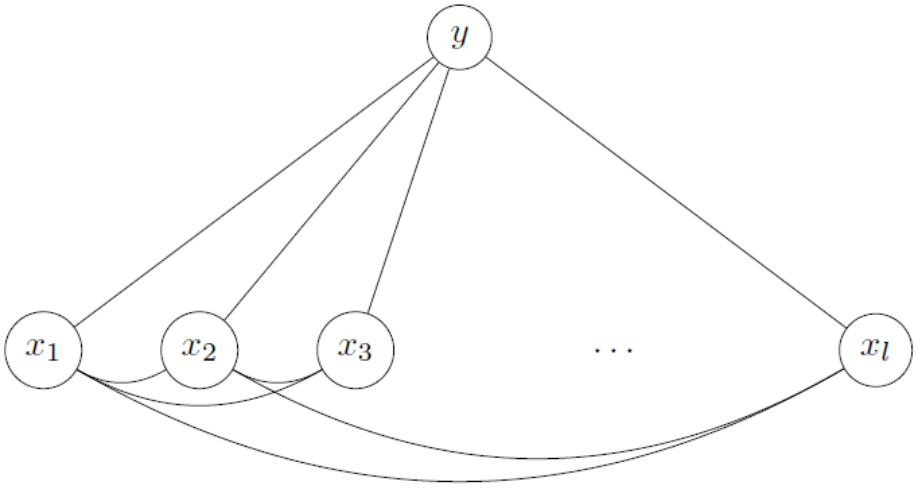

Figure 2: Complete Bayesian network

Looking at both figures, we can say a lot pertaining to the model' s complexity and the information about the underlying data. The more edges the network the more co-occurrence that the model requires to be stored. In graph terminology the vertices x1, x2, ..., xl will have a clique. The problem with a complete network as in figure 2 is it would require storing the counts of all possible combinations of words x1, x2, ..., xl where l can be from 2 up to maximum words in the input. And it is also required to store counts for the combinations related to each label y. This will be infeasible and can' t be stored in memory, in fact for large data sizes the space complexity is exponential. Each formulation of a Bayesian network on this data will be a subgraph of the figure 2.

The dilemma of using Na¨ıve Bayes to overlooking the complete probabilistic model of Bayes was a prickle for text classification. However, with Markov's assumption, the formulation can be improved. Considering the order in which the words appeared in the input text as x1, x2, ..., xl, the first order Markov principle assumes that each word is dependent only on the word that appears before it and thus,

$$\forall_{i=2}^{l} P(x_i|x_1, x_2, \ldots x_{i-1}) = P(x_i|x_{i-1}) \quad (6)$$

The CAN model [5] uses the assumption in equation 6 to modify equation 5 to get

$$P(x_1, x_2, \ldots, x_l, y) = P(x_l|x_{l-1}, y)\ldots P(x_2|x_1, y)P(x_1|y)P(y) \quad (7)$$

This reduces the storage overhead contrast to a complete Bayesian network by only storing the relevant co-occurrence counts for each label. Figure 3 shows the corresponding Bayesian network derived from figure 2 by considering successive edges between the "word" vertices. From the figure, it is evident that the dependency is between two n-grams or words, so their co-occurrences are stored per unique label y ∈ Y .

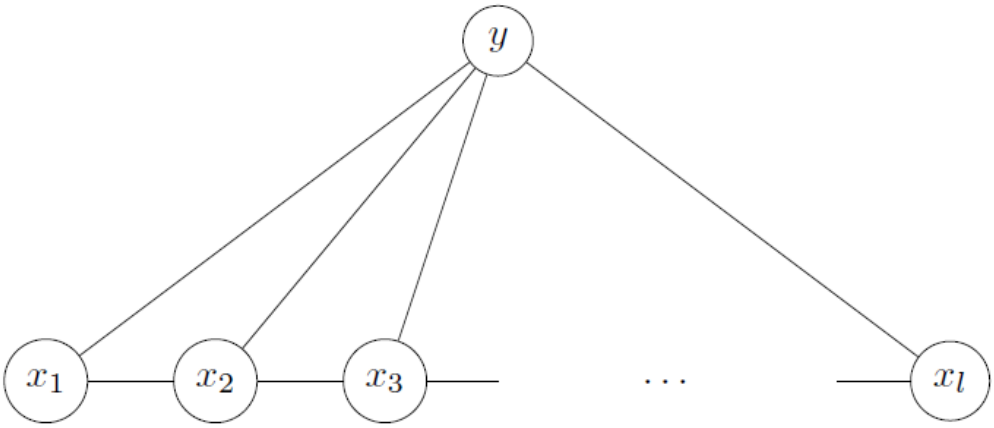

Figure 3: Bayesian network with Markov assumption.

This is equivalent to creating a three-dimensional table that stores tri- cooccurrences between two words and each label. This format of storage seems big but for moderately sized data the three-dimensional structure is sparse for the same reason as in section 3. Thus it' s quite effective for small-scale text classification.

2.4 Modified Bayesian Dependency Network

The method described in section 2.3 would provide better performance than Na¨ıve Bayes method in section 2.2 for moderate-size data. But, can we do something about large-scale data? The important part is whether we use the assumption in equation 6 in a more efficient manner! Comparing the graph structures in figure 1 and figure 3 we can see that the more edges a "word" vertex is incident to the more co-occurrence information that is needed to compute the formulation of the Bayesian network.

So, we can derive a simplified Bayesian network similar to one in figure 3 from the subgraph in figure 2 that doesn' t require storing the tri-occurrences in a three-dimensional structure. Depending on the connections between the words, they can either form a single connected component or have multiple connected components. If the subgraph formed by the words is not disconnected, then we can generate a Bayesian network as shown in figure 4. In this case, the subsequent vertices will have edges between them and the network will result in a valid Bayesian formulation.

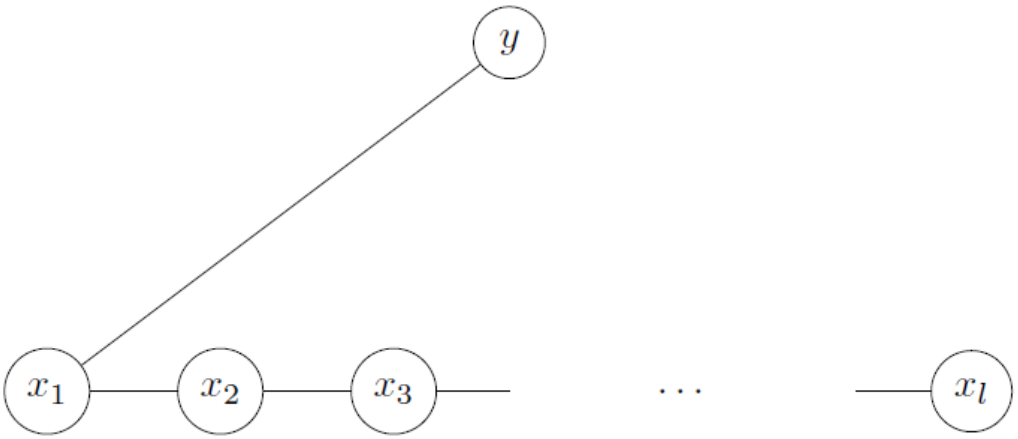

Figure 4: Bayesian network with modified assumption.

The modified Bayesian formulation will be look like,

$$P(x1, x2, \ldots, xl, y) = P(xl|xl-1)\ldots P(x2|x1)P(x1|y)P(y) \quad (8)$$

A disconnected network would occur when the combination of words from the input in the test set wasn't encountered in the training set. The words might co-occur in the training set but not in the specific order. A disconnected network wouldn't result in a valid Bayesian formulation and special techniques need to be developed to enable the same.

2.5 Conclusion and Future Work

We have described here the various ways that the Bayesian Probability method can be used to perform short text classification. The simplest Na¨ıve Bayes method is the benchmark for the performance of classifiers. There have been some improvements suggested recently and we believe that future improvements are still possible in a method that has been studied well. The modified approach that we described here serves as a barometer for further advancements that can be done in this direction. As an example disconnected Bayesian networks are something that can be utilitarian to the applications that require faster processing.

References

[1] Charu C. Aggarwal and ChengXiang Zhai. A Survey of Text Classification Algorithms, pages 163-222. Springer US, Boston, MA, 2012.

[2] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. Proceedings of the AAAI Conference on Artificial Intelligence, 29(1), Feb. 2015.

[3] Andrew McCallum, Kamal Nigam, et al. A comparison of event models for naive bayes text classification. In AAAI-98 workshop on learning for text categorization, volume 752, pages 41-48. Madison, WI, 1998.

[4] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. Machine learning, 29:131-163, 1997.

[5] Fuchun Peng and Dale Schuurmans. Combining naive bayes and n-gram language models for text classification. In European Conference on Information Retrieval, pages 335-350. Springer, 2003.