Learning Sentiment Dependent Bayesian Network Classifier for Online Product Reviews

Sylvester Olubolu Orimaye, Zi Yang Pang and Alvino Mandala Putra Setiawan
School of Information Technology, Monash University Malaysia
E-mail: sylvester.orimaye@monash.edu

Keywords: sentiment classification, sentiment-dependent, Bayesian network, product reviews

Received: September 17, 2015

Analyzing sentiments for polarity classification has recently gained attention in the literature with different machine learning techniques performing moderately. The challenge is that sentiment-dependent information from multiple sources are not considered often in existing sentiment classification techniques. In this study, we propose a logical approach that maximizes the true sentiment class probabilities of the popular Bayesian Network for a more effective sentiment classification task using the individual word sentiment scores from SentiWordNet. We emphasize on creating dependency networks with quality variables by using a sentiment-dependent scoring technique that penalizes the existing Bayesian Network scoring functions such as K2, BDeu, Entropy, AIC and MDL. The outcome of this technique is called Sentiment Dependent Bayesian Network. Empirical results on eight product review datasets from different domains suggest that a sentiment-dependent scoring mechanism for Bayesian Network classifier could improve the accuracy of sentiment classification by 2% and achieve up to 86.7% accuracy on specific domains.

Povzetek: Razvit je nov Bayesov klasifikator na osnovi mnenjske odvisnosti, uporabljen za ocenjevanje spletnih produktov.

1 Introduction

Sentiment Classification (SC) has recently gained a lot of attention in the research community [1, 2, 3]. More recently, SC has moved from the commonly used bag-of-words models to bag-of-concepts models [4, 5]. This is due to its increasing demand for the analysis of consumer sentiments on products, topic and news related text from social media such as Twitter[1] and online product reviews such as Amazon[2]. In the same manner, Bayesian Network (BN)[6] also known as Bayesian Belief Network plays a major role in Machine Learning (ML) research for solving classification problems. Over the last decade, learning BNs has become an increasingly active area of ML research where the goal is to learn a network structure using dependence or independence information between set of variables [6, 7, 8, 9]. The resulting network is a directed acyclic graph (DAG), with a set of joint probability distributions, where each variable of the network is a node in the graph and the arcs between the nodes represent the probability distribution that signifies the level of dependency between the nodes.

While it is more common to use other ML algorithms for SC tasks [10, 11], few research papers have proposed BN as a competitive alternative to other popular ML algorithms. Considering the huge size of data available from social media and the level of difficulty attached with analysing sentiments from natural language texts, the ability of BN to learn dependencies between words and their corresponding sentiment classes, could undoubtedly produce a better classifier for the sentiment classification task. This paper focusses on constructing a BN classifier that uses sentiment information as one important factor for determining dependency between network variables.

BN has been successfully applied to solve different ML problems with its performance outweighing some of the popular ML algorithms. For example, in [12], a full Bayesian Network classifier (FBC) showed statistically significant improvement on state-of-the-art ML algorithms with the 33 UCI datasets. In the case of SC, Naïve Bayes (NB), which is a special case of BN [13], and one of the leading ML algorithms for SC tasks [10], has surprisingly and repeatedly shown improved performance on movie and product reviews despite its conditional independence assumption. By comparative study, we show that a Sentiment Dependent Bayesian Network (SDBN) classifier has improved performance on popular review datasets such as Amazon product reviews due to the ability of the Bayesian Network to construct a network structure of multiple dependencies [12].

Constructing a BN classifier requires learning a network structure with set of Conditional Probability Tables (CPTs) [6]. Basically, there are two combined steps involved in the BN construction process. The first is to perform variable search on a search space, and the other is to score each variable based on the degree of fitness [14]. The chal-

[1] https://twitter.com/
[2] https://amazon.com/

lenge however, is to ensure that good networks are learned with appropriate parameters using a scoring or fitness function to determine network variables from the given dataset. Thus, much of the research works on BN focus on developing scoring functions for the BN classifier [15]. We argue that such scoring functions rely on many assumptions that make them less effective for SC tasks. For example, K2 algorithm, which is based on Bayesian Scoring function relies on the assumptions of parameter independence and assigning a uniform prior distribution to the parameters, given the class [9]. We believe these assumptions lead to many false positives in the classification results as sentiment classes are better captured by conditional dependency between words, rather than independent word counts [16, 17].

We also suggest that *varying* prior distribution could be assigned to each variable since each word has a natural *prior* probability of belonging to a particular sentiment class, independent of the data. For example, the word "good" is naturally positive and "bad" is naturally negative. Thus, in this work, we propose a sentiment scoring function that leverage sentiment information between variables in the given data. The output of the sentiment scoring function is then used to augment existing BN scoring functions for better performance. Our aim is to ensure sentiment information form part of the fitness criteria for selecting network variables from sentiment-oriented datasets such as reviews.

The proposed scoring function uses a multi-class approach to compute the conditional mutual information using sentiment-dependent information between local variables in each class of instances. The conditional mutual information for all classes are then penalized using the Minimum Description Length (MDL) principle. The local probabilities used in computing the conditional mutual information is computed using the popular Bayesian probability that uses the prior probability of a variable belonging to a natural sentiment class (i.e. independent of the given data by using individual word sentiment score from SentiWordNet [18]) and the observation of the variable in the selected class of instances and other classes in the dataset (e.g. positive and negative). For example, the class probability of each word in a product review is augmented with the polarity probability of the same word from SentiWordNet. The technique takes into account that the network structure would depend on the following criteria:

- The posterior probability from multiple evidences that variables x_i and x_j have sentiment dependency;
- The conditional mutual information between the variables for all sentiment classes;
- The dependency threshold computed using the Minimum Description Length principle; and
- The representation of the network as a full Bayesian Network as proposed in [12].

The importance of the first criterion is that we are able to avoid the *independence assumption* made by the existing BN scoring functions. We capture local sentiment dependency between the variables as a joint probability of evidences from each variable and each class in the given data. Also, existing BN scoring functions uses the conditional *independence* given the data as a whole for determining dependencies between variables [15, 9]. Under such approach in SC, two co-occurring words may occur with the same or similar frequencies in different classes. We observed that training BN classifier without penalizing such occurrences or dependencies, could affect the classifier decision to decide an appropriate sentiment class. As such, our second criterion captures the conditional mutual information between the variables, while the third criterion ensures that a BN classifier uses quality variables that are above the computed threshold. The latter also allow us to enforce strict *d-separation* policy between the network variables, which formally defines the process of determining independence between variables [19]. Thus, only quality variables are used to form the dependency network for the BN classifier. Finally, we introduce the last criterion as an improvement over a network constructed based on the first three criteria. The full Bayesian Network technique ensures that an independent sentiment dependent network is constructed for each sentiment category (i.e. negative and positive).

Section 2 of this paper discusses related work and additional motivations. In Section 3, we explain the problem background and then present the proposed sentiment-dependent technique in Section 4. Our experiment is described in Section 5. Finally, Section 6 gives the conclusion to our study and some thoughts on future research directions.

2 Related work

2.1 Sentiment classification (SC)

The most prominent of SC work is perhaps [20] which used supervised machine learning techniques for the polarity classification of *positive* and *negative* sentiments in movie reviews. As a result of that work, different research directions within the field of sentiment analysis and opinion mining have been actively pursued [2, 3, 4, 5].

[10] proposed a subjectivity summarization technique, which uses minimum cuts to classify sentiment polarities in movie reviews. The technique identifies and extracts subjective portions of review documents using minimum cuts in graphs. The minimum cut approach takes into consideration, the pairwise proximity information supplied through graph cuts that partition sentences which are likely to be in the same sentiment class. This approach gave better improvement from 82.8% to 86.4% on the subjective portion of review documents. The approach also gave similar better improvement when only 60% portion of a product review document is used compared to an entire review. In our work, we propose a classification technique that uses

the entire portion of each product review.

[21] performed classification of approximately 200K product reviews by using different machine learning algorithms. More importantly, the work investigated the significance of higher order n-gram language model ($n \geq 3$) in classifying sentiments from product reviews with an F1 of 90%. While Cui's work was performed on random websites for online products with limited product domains, we performed experiments on Amazon product reviews with 8 different product domains.

Similarly, [22] performed experiment with higher order n-gram features to train a Neural Network model on Amazon and TripAdvisor datasets with the best average error rates of 7.12 and 7.37, respectively. In contrast, our work investigate the performance of a sentiment-dependent Bayesian Network classifier on the Amazon datasets with different product domains.

More recently, [23] proposed a concept-level sentiment analysis technique, which uses the knowledge-based sentic computing technique that has recently gained attention within the sentiment analysis domain [3, 4, 5]. Because the sentic computing knowledge base sometimes omits vital sentiment discourse, [23] combines low-level linguistic features with the sentic computing technique to train machine learning model for polarity detection. In our work, the only knowledge base employed is SentiWordNet [24], which computes the natural polarity values of words rather than concept. Thus, we captured the dependencies between words by constructing and learning a Bayesian Network for sentiment classification.

A detailed review of other recent sentiment classification techniques on different datasets can be found in [1, 2, 3, 4, 5].

2.2 BN classifiers for sentiment classification

[16] and [17] proposed a two-stage Markov Blanket Classifier (MBC) approach to extract sentiments from unstructured text such as movie reviews by using BN. The approach learns conditional dependencies between variables (words) in a network and finds the portion of the network that falls within the *Markov Blanket*. The *Tabu Search* algorithm [25], is then used to further prune the resulting Markov Blanket network for higher cross-validated accuracy. Although Markov Blanket has shown to be effective in avoiding *over-fitting* in BN classifiers [7], the MBC approach finds sentiment dependencies based on the ordinary *presence* or *absence* of words in their original sentiment class only. We identify sentiment dependencies by considering multiple sources of evidence. These include multiple sentiment classes in the data and the *natural* sentiment class of each variable which is independent of its sentiment class in the given data.

Similarly, [26] proposed a parallel BN learning algorithm using MapReduce for the purpose of capturing sentiments from unstructured text. The technique experimented on large scale blog data and captures dependencies among words using mutual information or entropy, with the hope of finding a vocabulary that could extract sentiments. The technique differs from [17] by using a three-phase (drafting, thickening and thinning) dependency search technique that was proposed in [27]. Other than using *mutual information* in the *drafting* phase of the search technique, the work did not capture additional sentiment dependencies using other source of evidence.

Again, we do not focus on developing a search algorithm but a scoring technique that considers multiple sentiment-dependent information as part of the existing state-of-the-art scoring functions.

3 Problem background

3.1 Bayesian network (BN)

A Bayesian Network N is represented as a graphical distribution of the joint probability between a set of random variables [28]. The network has two components: (1) a DAG $G = (R_n, M_r)$ that represents the structural arrangement of a set of variables (nodes) $R_n = \{x_1, ..., x_n\}$ and a corresponding set of dependence and independence assertions (arcs) M_r between the variables; (2) a set of conditional probability distributions $P = \{p_i, ..., p_n\}$ between the parent and the child nodes in the graph.

In the DAG component, the existence of an directed arc between a pair of variables x_i and x_j asserts a conditional dependency between the two variables [8]. The directed arc can also be seen to represent *causality* between one variable and the other [29], that is, variable x_y is an existential cause of variable x_z, hence $x_y \rightarrow x_z$. The absence of an directed arc between a pair of variables, however, represents a conditional independence, such that, given a subset U of variables from R_n, the degree of information about variable x_i does not change by knowing x_j, thus $I(x_i, x_j|\mathbb{U})$. This also implies that $p(x_i|x_j, U) = p(x_i|U)$. The parent(s) of variable $x_i \in R_n$ is denoted by a set $pa_G(x_i) = x_j \in R_n|x_j \rightarrow X_i \in M_r$, and $pa_G(x_i) = \emptyset$ for the root node.

The conditional probability distributions of the DAG G is represented by its CPT, which contains a set of numerical parameters for each variable $x_i \in R_n$. These numeric parameters are computed as the probability of each variable given the set of parents, $p(x_i|pa_G(X_i))$. Over the set of variables in R_n, the joint probability for the BN is therefore obtained as follows:

$$p(x_1, ..., x_n) = \prod_{X_i \in R_n} p(x_i|pa_G(x_i)) \qquad (1)$$

Thus, for a typical classification task, the BN classifier would learn the numerical parameters of a CPT from the DAG structure G, by estimating some statistical information from the given data. Such information include, *mutual information* (MI) between the variables and *chi-square distribution* [15]. The former is based on the *local score*

metrics approach and the latter exhibits *conditional independence tests* (CI) approach. For both approaches, different *search* algorithms are used to identify the network structure. The goal is to ascertain, according to one or more search criteria, the best BN that fits the given data by evaluating the weight of the arc between the variables. The criteria for evaluating the fitness of the nodes (variables), and the arcs (parameters) in the BN search algorithms, are expressed as fitting or scoring functions within the BN classifier [15]. Our goal is to ensure that those criteria include *sentiment-dependent* information between the variables. We will focus on penalizing existing *local score metrics* with our sentiment-dependent scoring function for the BN classifiers, hence the SDBN proposed in this paper.

The local score metrics are of particular interest to our sentiment classification task because they exhibit a practical characteristic that ensures the joint probability of the BN is *decomposable* to the sum (or product) of the individual probability of each node [28][15]. To the best of our knowledge, very few research papers have considered sentiment-dependent information, as part of the fitness criteria for capturing dependency between the variables, especially for product reviews on different domains.

3.2 BN scoring functions

We focus on the local score metrics functions, K2, BDeu, Entropy, AIC and MDL [15]. The functions define a fitness score, and a specified search algorithm searches for the best network that maximizes the score. Each of these functions identifies frequencies of occurrence of each variable x_i in the data D and a network structure N. Although the performance of these scoring functions may vary on different datasets [15], in this paper, we assume that the scores generated by the scoring functions are somehow naïve, thus, we attempt to mitigate its effect on SC tasks. First, we will define the parameters that are common to all the functions. We will then describe each of the functions with their associated formula and specify their limitations to the SC tasks.

Similar to [30], we use $r_i (1 \leq i \leq n)$ to denote the size or cardinality of x_i. $pa(x_i)$ represents the parents of x_i and the cardinality of the parent set is represented by $q_i = \prod_{x_j \in pa(x_i)} r_j$. If $pa(x_i)$ is empty (i.e. $pa(x_i) = \emptyset$), then $q_i = 1$. The number of instances in a dataset D, where $pa(x_i)$ gets its jth value is represented by $N_{ij} (1 \leq i \leq n, 1 \leq j \leq q_i)$. Similarly, $N_{ijk} (1 \leq i \leq n, 1 \leq j \leq q_i, 1 \leq k \leq r_i)$ represents the portion of D where $pa(x_i)$ gets its jth value and x_i gets its kth value such that $N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$. Obviously, N represents the size of D.

K2: This metric is a type of Bayesian scoring function proposed by [6]. The function relies on series of assumptions such as parameter independence and uniform prior probability for the network. We reiterate that instead of independent word counts, the sentiments expressed in a given data are better captured using conditional dependency between words and their related sentiment classes [16]. The K2 metric is defined as follows:

$$S_{k2}(N, D) = P(N) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(r_i - 1 + N_{ij})!} \prod_{k=1}^{r_i} N_{ijk}! \tag{2}$$

BDeu: The metric was proposed by [31] as a generalization of K2. It resulted from Bayesian Drichlet (BD) and BDe which were proposed by [32]. The BD is based on hyperparameters η_{ijk} and the BDe is a result of BD with additional assumptions. BDeu relies on the sample size η as the single parameter. Since BDeu is a generalization of K2, it carries some of our concerns expressed on K2 earlier. Most importantly, the uniform prior probability assigned to each variable $x_i \in pa(x_i)$ could be replaced by the probability of the variable belonging to a *natural* sentiment class as stated earlier. We suggest that this is likely to increase the accuracy of the sentiment classifier especially on sparse data distribution. We define the BDeu metric as follows:

$$S_{BDeu}(N, D) = P(N) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{(\Gamma(\frac{\eta}{q_i})}{\Gamma(N_{ij} + \frac{\eta}{q_i})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \frac{\eta}{r_i q_i})}{\Gamma(\frac{\eta}{r_i q_i})} \tag{3}$$

Note that the function $\Gamma(.)$ is inherited from BD, and $\Gamma(c) = \int_0^\infty e^{-u} u^{c-1} du$ [15].

Entropy: Entropy metric measures the distance between the joint probability distributions of the network [15]. This allows dependency information to be identified by computing the mutual information (or entropy) between pair of variables. Thus, a minimized entropy between a pair of variables denotes dependency relationship, otherwise, a large entropy implies conditional independence between the variables [12][32]. While the entropy metric has been successful in measuring dependency information for BN classifiers, the local probabilities involved in the metric is largely computed based on *conditional independence* assumption given the data (i.e. using frequency counts for independent variables). We suggest that a joint probability of multiple sentiment evidences could improve the metric in BN classifiers for the SC tasks. The metric is defined as follows:

$$H(N, D) = -N \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N} \log \frac{N_{ijk}}{N_{ij}} \tag{4}$$

AIC: The AIC metric adds a non-negative parameter penalization to the entropy method [15], which could also be improved by multiple sentiment evidences as in the case of the entropy method. The metric is specified as follows:

$$S_{AIC}(N, D) = H(N, D) + K \tag{5}$$

Where K is the number of parameters, such that $K = \sum_{i=1}^{n}(r_i - 1).q_i$.

MDL: The MDL metric is based on the minimum description length principle which selects a minimum representative portion of the network variables through coding [15]. The best BN is identified to minimize the sum of the description length for the data. The metric is defined as follows:

$$S_{MDL}(N, D) = H(N, D) + \frac{K}{2} \log N \tag{6}$$

The use of MDL has not been investigated for sentiment classification on its own except for selecting dependency threshold between variables in BN. The study in [28], suggests that the mean of the total cross-entropy error is asymptotically proportional to $\frac{\log N}{2N}$, which is why the entropy metric is penalized in Equation 6.

In this paper, the proposed sentiment-dependent score function is based on the Information Theory approach. The approach uses the entropy-based conditional mutual information (CMI) technique to measure the dependencies between the variables. The local probabilities for computing the CMI between two variables are derived as joint probability resulting from multiple evidences of both variables belonging to the same sentiment class. This is achieved by using a multiclass approach that measures the CMI in each sentiment class. The sum of the CMIs over the data is thereafter penalized using the MDL principle as suggested in [28].

4 Sentiment-dependent BN

As emphasized earlier, our motivation is to include sentiment information as part of the dependency criteria between the network variables. Similar to [12], we constructed a multi-class Full Bayesian Network and encode the sentiment information within the CMIs that determine the dependencies within the network.

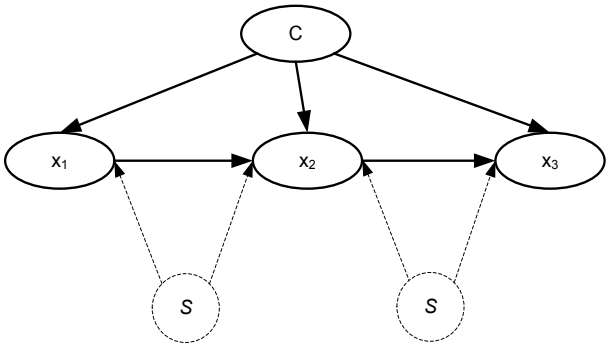

Figure 1: Structural overview of SDBN.

Figure 1 shows the structure of a SDBN. In this figure, C denotes the class node and the parent to all the variable nodes x_1, x_2 and x_3. The edges between variable x_1 and x_2 and x_3 represents dependencies between the variables. The dashed directed lines from each sentiment component S show the contribution of the sentiment information to the dependencies.

The proposed SDBN is created from a sentiment dependent score table (SDST) similar to the conventional CPT which contains network parameters from the data. Given a dataset D containing two or more sentiment classes, we divide D into c subsets, where $D_1 ... D_c$ represent the sentiment classes which are present in D. Thus, for each subset D_c, we create a SDST from the given data, and at the later stage, we use the values in SDST to learn a full Bayesian classifier.

Creating an appropriate CPT or SDST is challenging, especially when there is a sheer number of variables in the given data [27]. In fact, local search algorithms such as *K2*, *Hill Climbing*, *TAN*, *Simulated annealing*, *Tabu search* and *Genetic search* have been developed to address this challenge [7]. Thus, we do not intend to repeat the sophisticated local search process in our scoring technique. We use a straight forward approach that computes CMI as the dependency between a pair of variables, given a subset D_c. The resulting scores for each pair of variables is stored into the SDST. Equation 7 computes the CMI for a pair of variables. Note that this process is equivalent to the *drafting* phase proposed in [27] or the Chow and Liu algorithm in [33]. We can therefore focus on computing the local probabilities $P(x_i)$ and $P(x_j)$ for the CMI. In this work, each local probability encodes the sentiment dependency information as a *joint probability* of multiple sentiment evidences. We suggest that the joint probability is better than using the ordinary variable presence or single frequency count.

$$CMI(x_i, x_j|C) = \sum_{x_i, x_j, c} P(x_i, x_j, c) \log \frac{P(x_i, x_j, c)}{P(x_i|c), P(x_j|c)} \tag{7}$$

Alternatively, the CMI in Equation 7 can be computed with Equation 7 for penalizing the default CMI for a pair of variables, where $S_\lambda(x_i, x_j)$ is the sentiment *prior* computed from the multiple sentiment evidences for each variable x_i and x_j.

$$CMI(x_i, x_j|C) = \sum_{x_i, x_j, c} P(x_i, x_j, c) \log \frac{P(x_i, x_j, c)}{P(x_i|c), P(x_j|c)} S_\lambda(x_i, x_j) \tag{8}$$

4.1 Local probabilities for CMI

In order to compute the local probabilities $P(x_i)$ and $P(x_j)$, we adopt Bayesian probability [34], to calculate the joint probability from multiple sentiment evidences. Bayesian probability encodes a *generative model* or *likelihood* $p(D|\theta)$ of the dataset with a *prior* belief $p(\theta)$ to infer

a *posterior* distribution $p(\theta|D)$, see Equation 9. The idea is to determine a favourable posterior information of a particular variable belonging to its observed class, such that, the conditional mutual information between two dependent variables x_i and x_j increases when the posterior information for both variables in the same class is large.

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \tag{9}$$

However, in sentiment oriented documents such as product reviews, it is very common to observe variables that belong to different classes in one sentiment class. [20] referred to such scenario as *thwarted expectation*. For example, a "positive" review document may contain certain "negative" words used to express dissatisfaction about an aspect of a product despite some level of satisfaction that the product might offer. With this kind of problem, it is much probable that a dependency network that is learned with ordinary frequency counts of each variable (regardless of the sentiment class) would no doubt leads to inaccurate sentiment classifiers.

Figure 2 shows a sample BN resulting from a product review dataset upon performing attribute selection. In that network, variable "After" has a 1.0 probability of belonging to the *negative* and *positive* classes, respectively. Similarly, variable "not" has a 0.723 probability of belonging to a "positive" class rather than "negative". Every other variables in the network, has split probabilities between both classes. Our aim is to remove such variables from the dependency network or at least minimize its influence in the network such that the quality of the network is improved for sentiment classification.

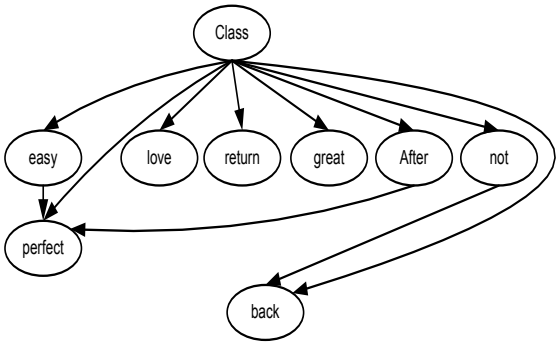

Figure 2: An example Bayesian network from product reviews.

In this work, we compute the posterior information for each variable by considering its *prior* information and joint *likelihood* or *observation* from all the sentiment classes available in the data.

The *prior* information is computed using the *natural sentiment or polarity scores* from SentiWordNet [24]. SentiWordNet gives the polarity scores of corresponding synsets for each English word. However, the polarity scores are often different for each of the synset entries. A synset contains multiple semantic or polarity interpretation of a given word. Each interpretation has three different polarities values. That is, a synset entry (word) would have a *positive*, *negative*, and *neutral* polarity scores which varies depending on the semantic interpretation of the word. An example of such words is "great". Its fourth synset entry in SentiWordNet has 0.25 *positive*, 0.625 *negative*, and 0.125 *neutral* polarity scores, respectively.

In this work, we focus on the "positive" and "negative" sentiments, thus we will only consider positive and negative polarity scores from SentiWordNet. The challenge however, is to compute an absolute and single polarity score for each word from its multiple synset entries. First, we compute the score for each polarity independently and then find the polarity that maximizes the other. The score for the positive or negative polarity of all synset entries for a given word is computed as follows:

$$score_\phi(w) = \frac{1}{\epsilon}\sum_{i=1}^{\epsilon} E_c(e_i) \tag{10}$$

where $score_\phi(w)$ is the score for each polarity of the given word w, ϵ is the number of synset entries E for the word, c is the polarity or category (i.e. positive or negative) and e_i is each synset entry. Thus, the *prior* or *absolute polarity score* for w is computed as follows:

$$POL_\phi(w) = \arg\max_{c \in C} score_\phi(w) \tag{11}$$

where $POL_\phi(w)$ is the maximum polarity score computed with respect to either *positive* or *negative* category c from all the syset entries.

We compute the *likelihood* information using a multi-class approach. Given a set of sentiment classes C, the probability of a variable belonging to its "first" observed sentiment class, is calculated as a joint probability of independently observing the variable in its first observed sentiment class (i.e.negative and positive) and every other sentiment classes, $C_1...C_n$. Thus, the likelihood information is computed as follows:

$$p(x_1, ..., x_C|D) = \prod_{c=1}^{C} p(x_c|D) \tag{12}$$

Where $p(x_c|D)$ is the probability of a variable x belonging to a class c given the data D.

Given the data, our aim is to minimise the effect of the variables which might have appeared in a wrong (false positive) class as a result of *thwarted expectation* that was suggested in [20], thereby biasing the dependency structure. Common examples are *negation* and *objective* words such as *not* and *After* as illustrated with Figure 2. If the word "not" for example, has a probability of 0.723 in a first observed "positive" class and a probability of 0.496 in the other negative class, then its *likelihood* of actually belonging to the "positive" class would be 0.359. Note that each probability is independent in this case as both probabilities do not sum to 1.

In addition, the *prior* or *natural sentiment score* (see Equation 11) obtained from SentiWordNet regulates the *likelihood* further, ensuring that the probability of a variable belonging to its first observed class is also conditioned on the natural sentiment class of the word which is independent of the data. With variable *not* having a probability of 0.625 *negative* from SentiWordNet, the *posterior* Bayesian probability is 0.149. This means the probability of the variable belonging to the *negative* class is higher (i.e. 0.85), and thus, should not be allowed to have strong dependency on a "true positive" variable. We suggest that this technique is more reliable than using the highest probability from both classes at the expense of accuracy (e.g. using only 0.723 and without the *prior*).

Thus, using the Bayesian probability defined in Equation 9, we substitute the *likelihood* information $p(x_1, ..., x_C|D)$ to $p(D|\theta)$ and the *prior* information $POL_\phi(w)$ to $p(\theta)$. Note that $P(D)$ is the sum of the two independent probabilities used in the likelihood (i.e. 0.723 and 0.496).

4.2 Sentiment dependency score

Having computed the local probabilities $P(x_i)$ and $P(x_j)$ using the Bayesian probability approach, we compute the conditional mutual information as the dependency information between pair of variables in each class. Thus, we store the dependency information in the sentiment dependent score table, SDST. Again, the SDST is similar to the conventional CPT. The obvious difference is that sentiment information have been used to generate SDST. However, since we are using conditional mutual information to compute dependencies between variables, certain dependency threshold needs to be met in order to generate a reliable sentiment dependencies between each pair of variables in the SDST. As mentioned earlier, [28] suggested that the mean of the total cross-entropy (mutual information) error is asymptotically proportional to $\frac{\log N}{2N}$. Using that MDL principle, we defined the threshold value as follows:

$$\Theta_{x_i,x_j} = \frac{\log N_c}{2N_c} \tag{13}$$

where Θ_{x_i,x_j} is the sentiment dependency threshold between a pair of variables x_i and x_j, N_c is the size of the data for a particular training class. Note that we generated individual SDST for each sentiment class in our data. In this work, a pair of variables x_i and x_j have strong sentiment dependency and get stored into the appropriate SDST, if and only if, the conditional mutual information $CMI(X_i, X_j|C) > \Theta_{x_i,x_j}$. Otherwise, we store a zero value to the corresponding slot in the SDST. CMI values greater than zero in the SDST is then used to build the resulting sentiment-dependent network structure for the ML task. The process of generating SDST is shown in Algorithm 1.

Algorithm 4 SDST(D)

Input : A set of labelled instances D.
Output : A set of Sentiment Dependent Score Tables for all pairs of variables x_i and x_j.
Steps

1: Create a multi-class structure that partitions instances D into subsets of classes D_c.
2: $SDST_{1,...,C}$ = empty.
3: **for each** subset D_c in D **do**
4: Compute the local probabilities $P(x_i)$ and $P(x_j)$ with sentiment dependent information as in Equation 9.
5: Use the local probabilities to compute CMI for each pair of variables x_i and x_j using Equation 7.
6: Compute the MDL threshold Θ with Equation 13.
7: **if** CMI $>$ MDL threshold Θ_{x_i,x_j} **then**
8: Store the CMI into $SDST_c$ columns x_i, x_j and x_j, x_i, respectively.
9: **else**
10: Store 0 into $SDST_c$ columns x_i, x_j and x_j, x_i, respectively.
11: **end if**
12: **end for**
13: Return $SDST_{1,...,C}$ for a full Bayesian Network classifier

5 Experiments and results

We conducted set of experiments using the proposed SDBN algorithm on 8 different product review domains. We then compared the accuracy with a state-of-the-art sentiment classification technique.

5.1 Datasets and baselines

Our datasets consist of Amazon online product reviews [3] that were manually crawled by [35]. These include *Health*, *Kitchen*, *Software*, *Video*, *Books*, *DVD*, *Electronics*, and *Grocery* reviews. Each product domain consists of **1000** *positive* reviews and **1000** *negative* reviews, hence each domain has **2000** balanced set of instances. Note that 60% training and 40% testing sets were used on all domains.

As baseline, we implemented the popular sentiment classification technique in [10] as a traditional sentiment classification benchmark on product reviews. The baseline technique produced subjective portions of reviews from our datasets and were used with NB and the ordinary BN classifiers. Baseline-NB denotes the baseline using NB classifier and Baseline-BN represents the baseline with ordinary BN classifier on the subjective portions of reviews, respectively. We performed a grid search with 10-fold cross-validation for the three algorithms and observed that both SDBN and Baseline-BN gave best accuracies using SimpleEstimator with $\alpha = 0.5$ and K2 search algorithm with

[3] http://www.cs.jhu.edu/ mdredze/datasets/sentiment/

the Bayes/K2 scoring function, while NB performed better with the Kernel Estimator.

5.2 Data preparation

We implemented our algorithm within the *weka.classifiers.bayes* package of the WEKA[4] data mining framework. The SentiWordNet library[5] including the lexicon file were also incorporated into the same WEKA directory. Further, we prepared our datasets according to the WEKA's ARFF format by using the term frequency-inverse document frequency (TF-IDF) from the positive and negative reviews for each domain. Our implementation also uses the discretization algorithm within WEKA in order to reduce the classification error. This technique also correct the missing values within the training data.

5.3 Attribute selection

We evaluated the performance of the SDBN with reduced attribute sets since attribute selection tends to improve BN's accuracy [16]. Thus, we ranked and reduced the set of attributes for each of our dataset by using the "Attribute-Selection" filter in Weka. Specifically, we used the *InfoGainAttributeEval* evaluator with the *ranker* search algorithm to select from the top-10 ranked attributes to the top-1000 ranked attributes for each domain. Our experiment showed better result with the top-ranked 100 attributes.

5.4 Results

As emphasised in Table 1, we observed the proposed SDBN to have improved and sometimes comparable performance with the baseline classifiers. SDBN recorded better improvements on the Health and Kitchen domains. We also note that the accuracies on the Amazon video reviews seems to be lower than the accuracies that were reported on the IMDb video reviews by [10]. We suggest that this is a trade-off in sentiment classification on different datasets and/or domains as could be observed in our experiment on different Amazon domains. This could be investigated further in our future work. We believe that increased size of dataset, that is beyond the limited 1000 Amazon reviews, could further improve the accuracy of the SDBN classifier.

We also performed experiment using the SDBN for top-10, top-20, top-30, top-50, and top-100 attributes alone as shown in Table 2. The results showed that the performance of the SDBN increased steadily up to the best accuracy given by the top-100 attributes. We believe this is an indication that the performance of the SDBN could increase with much larger datasets. In addition, we compared between the performance of the top-10 to top-100 attributes for SDBN and the two baselines on the top three domains with better accuracy: Health, Kitchen, and Video. Figures 3, 4, 5,and 6 show consistent improvement with increasing attributes for the SDBN compared to the baselines.

[4]http://www.cs.waikato.ac.nz/ml/weka/

[5]http://sentiwordnet.isti.cnr.it/download.php

Table 1: Accuracies of SDBN and baseline classifiers on Amazon product reviews.

Dataset	SDBN	Baseline-BN	Baseline-NB
Health	**80.2%**	78.9%	78.6%
Kitchen	**82.3%**	80.5%	81.8%
Software	78.7%	78.6%	78.7%
Video	**75.4%**	75.2%	74.9%
Books	**77.7%**	77.5%	77.3%
DVD	77.6%	77.6%	77.5%
Electronic	79.9%	79.8%	79.7%
Grocery	86.7%	86.7%	86.5%

Table 2: Accuracies of SDBN on Top-10 to Top-100 attributes.

Dataset	Top-10	Top-20	Top-30	Top-50	Top-100
Health	67.4%	70.7%	73.4%	76.9%	80.2%
Kitchen	71.4%	74.8%	77.2%	79.0%	82.3%
Software	69.7%	73.4%	75.4%	76.1%	78.7%
Video	63.5%	68.6%	72.3%	75.1%	75.4%
Books	68.4%	71.1%	73.6%	75.3%	77.7%
DVD	68.1%	72.3%	73.8%	75.7%	77.6%
Electronic	67.0%	70.1%	70.5%	75.8%	79.9%
Grocery	69.9%	75.9%	80.0%	83.0%	86.7%

As shown in Table 3, we also performed experiment by using SDBN with other scoring functions reported in Section 3.2 using the top-100 attributes, which gave better accuracy. Our observation shows that those scoring functions did not improve the result for SDBN beyond the Bayes/K2 scoring function used in the earlier experiments. This is consistent with the comparative study conducted in [15] on BN scoring functions. Overall, we have observed the SDBN classifier to have reasonable performance that shows a promising research pathway for using Bayesian Network as a competitive alternative classifier for sentiment classification tasks.

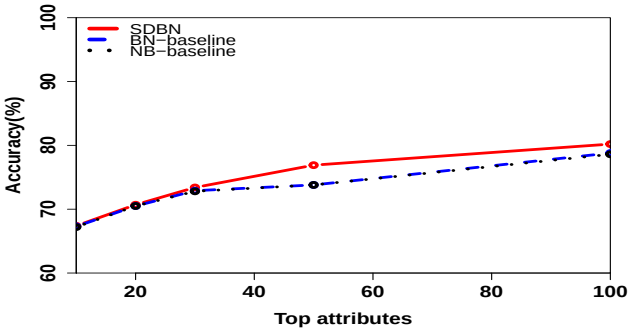

Figure 3: SDBN vs. baselines for top-10 to top-100 attributes on Health domain.

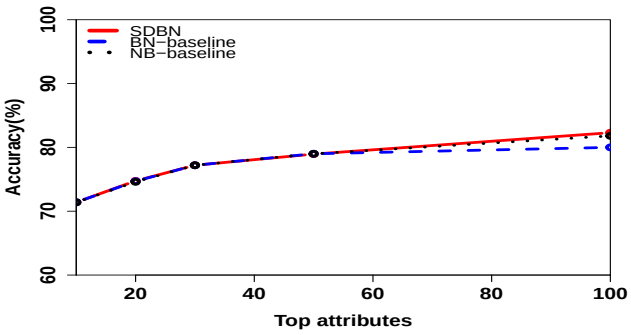

Figure 4: SDBN vs. baselines for top-10 to top-100 attributes on Kitchen domain.

Table 3: Accuracies of SDBN with different scoring functions on Amazon product reviews.

Dataset	K2	BDeu	Entropy	AIC	MDL
Health	80.2%	76.3%	76.3%	76.1%	76.1%
Kitchen	82.3%	75.3%	73.4%	75.2%	75.2%
Software	78.7%	73.1%	73.1%	73.1%	73.1%
Video	75.4%	73.5%	74.4%	72.8%	73.5%
Books	77.7%	70.9%	70.1%	71.1%	71.1%
DVD	77.6%	71.8%	70.0%	71.7%	71.7%
Electronic	79.9%	77.0%	76.2%	76.2%	77.2%
Grocery	86.7%	79.2%	80.0%	80.2%	79.3%

There are some limitations in the use of SentiWordNet though. For example, because there are many domain specific or technical terms (e.g. brand names) that were used in the product reviews, sentiment priors of those terms returned zero (i.e. neutral) as they are neither negative nor positive. This might have affected the sentiment dependencies within the network structure.

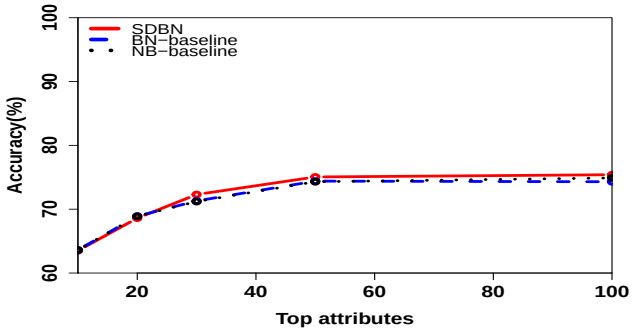

Figure 5: SDBN vs. baselines for top-10 to top-100 attributes on Video domain.

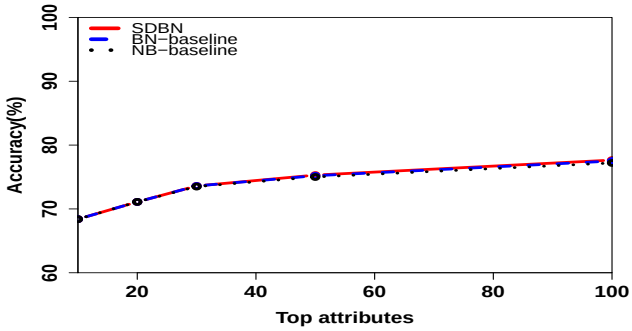

Figure 6: SDBN vs. baselines for top-10 to top-100 attributes on Books domain.

6 Conclusion

In this study, we have proposed a sentiment-dependent Bayesian network (SDBN) classifier. The proposed SDBN uses a multi-class approach to compute sentiment dependencies between pairs of variables by using a joint probability from different sentiment evidences. Thus, we calculated a sentiment dependency score that penalizes existing BN scoring functions and derived sentiment dependency network structure using the conditional mutual information between each pair of variables in a dataset. We performed sentiment classification on eight different Amazon product domains with the resulting network structure. Experimental results show that the proposed SDBN has comparable, and in some cases, improved accuracy than the state-of-the-art sentiment classifiers. In the future, we will experiment with SDBN on large scale Amazon SNAP datasets and the Hadoop platform.

Acknowledgement

The authors would like to thank the anonymous reviewers for the constructive comments.

References

[1] H. Tang, S. Tan, and X. Cheng, "A survey on sentiment detection of reviews," *Expert Systems with Applications*, vol. 36, no. 7, pp. 10 760–10 773, 2009.

[2] B. Liu, "Sentiment analysis and opinion mining," *Synthesis Lectures on Human Language Technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[3] E. Cambria and A. Hussain, *Sentic computing: Techniques, tools, and applications*. Springer Science & Business Media, 2012, vol. 2.

[4] E. Cambria, B. Schuller, Y. Xia, and C. Havasi, "New avenues in opinion mining and sentiment analysis," *IEEE Intelligent Systems*, no. 2, pp. 15–21, 2013.

[5] E. Cambria, D. Olsher, and D. Rajagopal, "Senticnet 3: a common and common-sense knowledge base for cognition-driven sentiment analysis," in *Twenty-eighth AAAI conference on artificial intelligence*, 2014, pp. 1515–1521.

[6] G. F. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine learning*, vol. 9, no. 4, pp. 309–347, 1992.

[7] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, 1997, 10.1023/A:1007465528199. [Online]. Available: http://dx.doi.org/10.1023/A:1007465528199

[8] J. Cheng and R. Greiner, "Learning bayesian belief network classifiers: Algorithms and system," in *Advances in Artificial Intelligence*. Springer, 2001, pp. 141–151.

[9] X.-W. Chen, G. Anantha, and X. Lin, "Improving bayesian network structure learning with mutual information-based node ordering in the k2 algorithm," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 5, pp. 628–640, 2008.

[10] B. Pang and L. Lee, "A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*. Barcelona, Spain: Association for Computational Linguistics, 2004, p. 271.

[11] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual web texts," *Information Retrieval*, vol. 12, no. 5, pp. 526–558, 2009.

[12] J. Su and H. Zhang, "Full bayesian network classifiers," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 897–904.

[13] J. Cheng and R. Greiner, "Comparing bayesian network classifiers," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 101–108.

[14] D. Heckerman, *A tutorial on learning with Bayesian networks*. Springer, 2008.

[15] L. M. De Campos, "A scoring function for learning bayesian networks based on mutual information and conditional independence tests," *The Journal of Machine Learning Research*, vol. 7, pp. 2149–2187, 2006.

[16] E. Airoldi, X. Bai, and R. Padman, "Markov blankets and meta-heuristics search: Sentiment extraction from unstructured texts," in *Advances in Web Mining and Web Usage Analysis*. Springer, 2006, pp. 167–187.

[17] X. Bai, "Predicting consumer sentiments from online text," *Decision Support Systems*, vol. 50, no. 4, pp. 732–742, 2011.

[18] A. Esuli, "Automatic generation of lexical resources for opinion mining: models, algorithms and applications," *SIGIR Forum*, vol. 42, no. 2, pp. 105–106, 2008.

[19] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[20] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques," in *Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10*. Association for Computational Linguistics, 2002, pp. 79–86.

[21] H. Cui, V. Mittal, and M. Datar, "Comparative experiments on sentiment classification for online product reviews," American Association for Artificial Intelligence (AAAI), 2006.

[22] D. Bespalov, B. Bai, Y. Qi, and A. Shokoufandeh, "Sentiment classification based on supervised latent n-gram analysis," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 375–382.

[23] S. Poria, E. Cambria, G. Winterstein, and G.-B. Huang, "Sentic patterns: Dependency-based rules for concept-level sentiment analysis," *Knowledge-Based Systems*, vol. 69, pp. 45–63, 2014.

[24] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining," *Proceedings of LREC*, 2006.

[25] F. Glover, M. Laguna *et al.*, *Tabu search*. Springer, 1997, vol. 22.

[26] W. Chen, L. Zong, W. Huang, G. Ou, Y. Wang, and D. Yang, "An empirical study of massively parallel bayesian networks learning for sentiment extraction from unstructured text," in *Web Technologies and Applications*. Springer, 2011, pp. 424–435.

[27] J. Cheng, D. A. Bell, and W. Liu, "Learning belief networks from data: An information theory based approach," in *Proceedings of the sixth international conference on Information and knowledge management*. ACM, 1997, pp. 325–331.

[28] N. Friedman and Z. Yakhini, "On the sample complexity of learning bayesian networks," in *Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1996, pp. 274–282.

[29] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos, "Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation," *The Journal of Machine Learning Research*, vol. 11, pp. 171–234, 2010.

[30] R. R. Bouckaert, *Bayesian network classifiers in weka*. Department of Computer Science, University of Waikato, 2004.

[31] W. Buntine, "Theory refinement on bayesian networks," in *Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 1991, pp. 52–60.

[32] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.

[33] C. Chow and C. Liu, "Approximating discrete probability distributions with dependence trees," *Information Theory, IEEE Transactions on*, vol. 14, no. 3, pp. 462–467, 1968.

[34] P. M. Lee, *Bayesian statistics: an introduction*. John Wiley & Sons, 2012.

[35] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," Association of Computational Linguistics (ACL), 2007.