Sentiment Augmented Bayesian Network

Sylvester Olubolu Orimaye

School of Information Technology, MONASH University Malaysia
sylvester.orimaye@monash.edu

Abstract

Sentiment Classification has recently gained attention in the literature with different machine learning techniques performing moderately. However, the challenges that sentiment classification constitutes require a more effective approach for better results. In this study, we propose a logical approach that augments the popular Bayesian Network for a more effective sentiment classification task. 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 and MDL. The outcome of this technique is called Sentiment Augmented Bayesian Network. Empirical results on three product review datasets from different domains, suggest that a sentiment-augmented scoring mechanism for Bayesian Network classifier, has comparable performance, and in some cases outperform state-of-the-art sentiment classifiers.

Keywords: sentiment; classification; Bayesian network

1 Introduction

Sentiment Classification (SC) has recently gained a lot of attention in the research community. 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)(Cooper & Herskovits 1992) 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 (Cooper & Herskovits 1992, Friedman et al. 1997, Cheng & Greiner 2001, Chen et al. 2008). 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 such as Support Vectors Machines (SVM), Naïve Bayes (NB) and Maximum Entropy (ME) for SC tasks (Pang & Lee 2004, Boiy & Moens 2009), few research papers have proposed BN as a competitive alternative to other popular ML algorithms. Considering the huge amount 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 Su & Zhang (2006), a full Bayesian Network classifier (FBC) showed statistically significant improvement on state-of-the-art ML algorithms such as SVM-SMO, C4.5 and NB on 33 UCI datasets. In the case of SC, NB, which is a special case of BN (Cheng & Greiner 1999), and one of the leading ML algorithms for SC tasks (Pang & Lee 2004), 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 Augment Bayesian Network (SABN) has better or comparable performance with NB and SVM classifiers on popular review datasets such as Amazon product reviews.

Constructing a BN classifier requires learning a network structure with set of Conditional Probability Tables (CPTs)(Cooper & Herskovits 1992). 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 (Heckerman 2008). The challenge 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 (De Campos 2006). 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 (Chen et al. 2008). We believe these assumptions lead to many false positives in the classification results as

[1] https://twitter.com/

[2] https://amazon.com/

sentiment classes are better captured by conditional dependency between words, rather than independent word counts (Airoldi et al. 2006, Bai 2011).

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 simple but logical multi-class approach to compute the conditional mutual information between local variables in each class of instances. The conditional mutual information for all classes are then combined through a penalization process that uses the Minimum Description Length (MDL) principle. The final entropy score is further used to penalize the score from an existing BN scoring function. 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 (Esuli 2008)) and the observation of the variable in the selected class of instances and other classes in the dataset (e.g. positive, negative and neutral). The technique takes into account that the dependency score between two local variables x_i and x_j of a SC task would depend on two criteria:

- The posterior probability from multiple evidences that variables x_i and x_j have sentiment dependency.
- The sum of conditional mutual information between the variables for all classes.

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. Existing BN scoring functions uses the conditional *independence* given the data as a whole for determining dependencies between variables (De Campos 2006, Chen et al. 2008). Under such approach in SC, two independent words may occur with the same or similar frequencies in different classes. Thus, training BN classifier without penalizing such occurrences or dependencies, could affect the classifier decision to decide an appropriate sentiment class. Finally, the second criterion allows us to enforce strict *d-separation* policy between the network variables (Pearl 1988). Thus, only quality variables are used to form the dependency network for the BN classifier.

Section 2 of this paper discuss related work and additional motivations. In Section 3, we explain the problem background and then present the proposed sentiment augmentation 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 Pang et al. (2002) which employed supervised machine learning techniques to classify *positive* and *negative* sentiments in movie reviews. The significance of that work influenced the research community and created different research directions within the field of sentiment analysis and opinion mining (Liu 2012).

Turney & Littman (2002) uses unsupervised learning of semantic orientation to classify reviews based on the number of negative and positive phrases. They achieved an accuracy of 80% over an unlabeled corpus.

Pang & Lee (2004) proposed a subjectivity summarization technique that is based on minimum cuts to classify sentiment polarities in movie reviews. The intuition is to identify and extract subjective portions of the review document using minimum cuts in graphs. The minimum cut approach takes into consideration, the pairwise proximity information via graph cuts that partitions sentences which are likely to be in the same class. This approach showed significant improvement from 82.8% to 86.4% on the subjective portion of the documents. The approach also shows equally good performance when only 60% portion of a review document is used compared to an entire review document.

Choi & Cardie (2008) proposed a *compositional semantics* approach to learn the polarity of sentiments from the sub-sentential level of opinionated expressions. The compositional semantic approach breaks the lexical constituents of an expression into different semantic components.

Wilson et al. (2005) use instances of polar words to detect contextual polarity of phrases from the MPQA corpus. Each phrase detected is verified to be either *polar* or *non-polar* phrase by using the presence of opinionated words from a polarity lexicon. A detailed review of other sentiment classification techniques on different datasets is provided in Liu (2012) and Tang et al. (2009).

2.2 BN Classifiers for Sentiment Classification

Airoldi et al. (2006) and Bai (2011) 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 (Glover et al. 1997), is then used to further prune the resulting Markov Blanket network for higher cross-validated accuracy. While the use of Markov Blanket has shown to be effective in avoiding *over-fitting* in BN classifiers (Friedman et al. 1997), 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, Chen et al. (2011) 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 Bai (2011) by using a three-phase (drafting, thickening and thinning) dependency search technique that was proposed in Cheng et al. (1997). 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 a graphical representation of a joint probability distribution between a set of random variables (Friedman & Yakhini 1996). The network consists of 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 (Cheng & Greiner 2001). The directed arc can also be seen to represent *causality* between one variable and the other (Aliferis et al. 2010), 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 $I(x_i, x_j|U)$. 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 $pa_G(x_i) = x_j \in R_n|x_{ji} \in M_r$, 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* (De Campos 2006). 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 (De Campos 2006). 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 augmented scoring function for the BN classifiers, hence the SABN proposed in this paper.

The local score metrics are of particular interest 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 (Friedman & Yakhini 1996)(De Campos 2006). 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.

3.2 BN Scoring Functions

We focus on the local score metrics functions, K2, BDeu, Entropy, AIC and MDL (De Campos 2006). 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. 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. Firstly, 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 Bouckaert (2004), 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 Cooper & Herskovits (1992). 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 (Airoldi et al. 2006). The K2 metric is defined as follows:

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

BDeu: The metric was proposed by Buntine (1991) as a generalization of K2. It resulted from Bayesian Drichlet (BD) and BDe which were proposed by Heckerman et al. (1995). 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 improve the performance of the sentiment classifier especially on sparse data distribution. We define the BDeu metric as follows:

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

Note that the function $\Gamma(.)$ is inherited from BD, and $\Gamma(c) = \int_0^\infty e^{-u}u^{c-1}du$ (De Campos 2006).

Entropy: Entropy metric measures the distance between the joint probability distributions of the network (De Campos 2006). 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 (Su & Zhang 2006)(Heckerman et al. 1995). 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 evidences could improve the metric in BN classifiers for the SC tasks. The metric is defined as follows:

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

AIC: The AIC metric adds a non-negative parameter penalization to the entropy method (De Campos 2006). The metric is specified as follows:

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

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 (De Campos 2006). Thus, the best BN is identified to minimize the sum of the description length for the data. The metric is defined as follows:

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

The use of MDL has been particularly effective for selecting dependency threshold between variables in BN. The study in Friedman & Yakhini (1996), 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 augmented score function is based on a straight forward 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 Friedman & Yakhini (1996).

4 Sentiment Augmented Score (SAS)

In this section, we will show how we derived the sentiment augmented score for BN. 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. Note that the process of creating the SASs is similar to the process of creating a CPT which contains the resulting network parameters from a particular search algorithm, given the data. Thus, we will create a SAS table (SAST) from the given data, and at the later stage, we will use the values in SAST to augment existing scores from the original CPT.

Creating an appropriate CPT or SAST is challenging, especially when there is a sheer number of variables in the given data (Cheng et al. 1997). 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 (Friedman et al. 1997). Thus, we do not intend to repeat the sophisticated local search process in our augmented 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 SAST. Equation 7 computes the CMI for a pair of variables. Note that this process is equivalent to the *drafting* phase proposed in Cheng et al. (1997) or the Chow and Liu algorithm in Chow & Liu (1968). 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.

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

4.1 Local probabilities for CMI

In order to compute the local probabilities $P(x_i)$ and $P(x_j)$, we adopt Bayesian probability (Lee 2012), 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 8. 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.

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)} \quad (8)$$

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. Pang et al. (2002) 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 1 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 a split probabilities between both classes. Our aim is to remove the contrasting variables such as *After* and *not* 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.

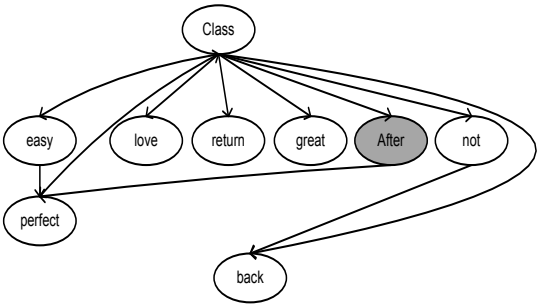

Figure 1: An example Bayesian network from product reviews.

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

The *prior* information is computed using the *natural sentiment or polarity scores* from SentiWordNet (Esuli & Sebastiani 2006). 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:

$$score_\phi(w) = \frac{1}{\epsilon}\sum_{i=1}^{\epsilon} E_c(e_i) \quad (9)$$

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:

$$POL_\phi(w) = \underset{c\in C}{\operatorname{argmax}}\, score_\phi(w) \quad (10)$$

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 and every other sentiment classes, $C_1...C_n$. Thus, the likelihood information is computed as follows:

$$p(x_1, ..., x_C|D) = \prod_{c=1}^{C} p(x_c|D) \quad (11)$$

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 Pang et al. (2002), thereby biasing the dependency structure. Common examples are *negation* and *objective* words such as *not* and *After* as illustrated with Figure 1. 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 10) 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 8, 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 augmented score table, SAST. Again, the SAST is similar to the conventional CPT. The obvious difference is that sentiment information have been used to generate SAST. 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 SAST. As mentioned earlier, Friedman & Yakhini (1996) 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:

$$\Theta_{x_i,x_j} = \frac{\log N_c}{2N_c} \tag{12}$$

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 SAST 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 SAST, 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 SAST.

Finally, we reiterate our ultimate goal to penalize the dependency score from any of the existing scoring functions described in Section 3.2. Scoring functions such as K2 identifies dependency relationships by computing a *parent-child* score for a pair of variables x_i and x_j and checks if it maximizes a *base score* calculated as the total influence of a variable x_i on other variables in the data. We suggest that, for a sentiment classification task, the base score has highly minimized entropy in its current state, due to *false positive* variables as highlighted earlier. Thus, we penalize the base score of the existing scoring function by the SAST's *sentiment dependency score* between a pair of variables x_i and x_j. Arguably, this method creates reliable dependency network structures for training a sentiment classifier. Hence, we refer to this dependency network as Sentiment Augmented Bayesian Network (SABN). The *sentiment dependency score* for the SABN is defined below and it is computed as the sum of the conditional mutual information scores for the pair of variables x_i and x_j over all the sentiment classes.

$$Score_{SD}(x_i, x_j) = \sum_{c=1}^{C} CMI(x_i, x_j|C) \tag{13}$$

where C is the set of sentiment classes and $CMI(x_i, x_j|C)$ is the conditional mutual information score defined in Equation 7.

4.3 Summary of the SABN Algorithm

Algorithm 1 and Algorithm 2 are the two main algorithms involved in SABN. We will give a summary of the two algorithms as follows.

The purpose of Algorithm 1 is to generate the SAST which contains the CMIs between pairs of variables in the dataset. More importantly, sentiment information have been used to compute the local probabilities for each CMI. The algorithm takes as input a dataset D containing a set of labelled instances that are partitioned into a subset of classes D_c. For each subset D_c, CMI is computed for each pair of variables. Note that each CMI is checked against a MDL threshold. CMIs that are above the threshold are stored into the $SAST_c$ for the corresponding subset D_c. Thus, the algorithm outputs a set of SAST to be used in Algorithm 2. Again, the SAST is similar to the conventional CPT but with encoded sentiment information as part of the local probabilities that compute the CMI.

Finally, Algorithm 2 creates the sentiment dependency network as the BN. The algorithm takes as inputs the generated $SAST_{1,...,C}$ for the set of classes and the dataset D. For each variable in D, a *base score* is calculated using a specified base score function from the existing scoring function. The *parent-child* dependency score is also computed between each pair of variables using the specified parent-child dependency score function. Further, we compute our *sentiment dependency* score using the sum of CMIs for a specified pair of variables over the set of SASTs. The sentiment dependency score is then used to penalize the base score of the specified scoring function. If a parent-child dependency score is larger than the penalized base score, then a dependency exists between the selected pair of variables and then stored in the network. Thus, the output of the algorithm is the sentiment augmented Bayesian network that is used to build the sentiment classifier.

Algorithm 1 SAST(D)

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

Steps

1: Partition instances D into subsets of classes D_c.
2: $SAST_{1,...,C}$ = empty.
3: **for each** subset D_c in D **do**
4: Compute the local probabilities $P(x_i)$ and $P(x_j)$ with Equation 8.
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 12.
7: **if** CMI > MDL threshold Θ_{x_i,x_j} **then**
8: Store the CMI into $SAST_c$ columns x_i, x_j and x_j, x_i, respectively.
9: **else**
10: Store 0 into $SAST_c$ columns x_i, x_j and x_j, x_i, respectively.
11: **end if**
12: **end for**
13: Return $SAST_{1,...,C}$

5 Experiments and Results

We conducted set of experiments using the proposed SABN algorithm on different product reviews. We then compared the accuracy with the ordinary BN classifier and a state-of-the-art sentiment classification technique.

Algorithm 2 SABN($SAST_{1,...,C}$, D)

Input : A set of $SAST_{1,...,C}$, training instances D.
Output : Sentiment Augmented Bayesian Network.

Steps

1: SABN = empty
2: **for each** variable x_i and x_j in D **do**
3: Get BaseScore(x_i) from a specified base score function in the search algorithm.
4: Get ParentChild(x_i, x_j) from a specified parent-child score function in the search algorithm.
5: $Score_{SD} = 0$.
6: **for each** subset $SAST_c$ in $SAST_{1,...,C}$ **do**
7: $Score_{SD} = Score_{SD} + SAST_c(x_i, x_j)$
8: **end for**
9: Penalize BaseScore(x_i) with $Score_{SD}$
10: **if** ParentChild(x_i, x_j) > BaseScore(x_i) **then**
11: Add dependency between x_i and x_j in SABN.
12: **end if**
13: **end for**
14: Return SABN

5.1 Datasets and Baselines

Our datasets consist of Amazon online reviews from three different product domains[3] that were manually crawled by Blitzer et al. (2007). These include *video*, *music*, and *kitchen* appliances. Each product domain consists of **1000** *positive* reviews and **1000** *negative* reviews, hence each domain has **2000** balanced set of instances. According to Blitzer et al. (2007), positive reviews were selected using a star rating of greater than 3 and negative reviews used a star rating of less than 3. Other ratings were discarded due to the ambiguity of their polarities. Note that 60% training and 40% testing sets were used on all domains. Table 1 shows details of the three datasets.

Table 1: Details of the three review datasets.

Dataset	Instances	Neg/Pos	Attributes
Kitchen	2000	1000/1000	1290
Music	2000	1000/1000	1292
Video	2000	1000/1000	1326

As our baseline, we implemented the popular sentiment classification technique in Pang & Lee (2004) using NB and SVM classifiers on our datasets with the same testing-to-training ratio. We also included additional baseline by using the ordinary BN without our proposed algorithm.

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 concatenating the positive and negative reviews for each domain and created a string data file in ARFF format. The string data file was then converted to TFIDF data file in ARFF format using StringToWordVector filter with default settings. Note that the TFIDF format is just a processable numerical representation of the text variables that is supported by the *bayes* package. Arguably, the representation still maintains the dependency relationship between the words (variables) as in the original string format.

[3] http://www.cs.jhu.edu/ mdredze/datasets/sentiment/
[4] http://www.cs.waikato.ac.nz/ml/weka/
[5] http://sentiwordnet.isti.cnr.it/download.php

5.3 Results

Table 2: Accuracies of SABN and baseline classifiers on Amazon product reviews.

Dataset	SABN	Baseline-BN	Baseline-NB	Baseline-SVM
Kitchen	75.5%	70.7%	75.3%	76.3%
Music	**74.4%**	68.4%	73.9%	71.5%
Video	**81.5%**	72.6%	80.0%	77.1%

Table 2 shows the accuracies of the proposed SABN alongside other baseline classifiers. Baseline-BN represents the ordinary BN classifier in Weka. Baseline-NB and Baseline-SVM denote the implemented baseline technique on NB and SVM-SMO classifiers respectively. For both NB and SVM, we use the presence and absence of *unigram* features as suggested in Pang & Lee (2004). Note that both SABN and Baseline-BN used the SimpleEstimator with $\alpha = 0.5$ and K2 search algorithm with the Bayes/K2 scoring function.

As emphasised in Table 2, we observed the proposed SABN to have similar and in some cases improved performance compared to the baseline classifiers. For example, SABN recorded better improvements with average of 3.1% and 5.3% over the three baselines on Music and Video domains, respectively. 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 Pang & Lee (2004). 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. In addition, we believe that increased size of dataset, that is beyond the limited 1000 Amazon reviews, could further improve the accuracy of the SABN classifier.

In further experiments, we evaluated the performance of the SABN with reduced attribute sets since attribute selection tends to improve BN's accuracy (Airoldi et al. 2006). Thus, we ranked and reduced the set of attributes for each of our dataset by using the "AttributeSelection" filter in Weka. Specifically, we used the *InfoGainAttributeEval* evaluator with the *ranker* search algorithm. We used up to top-ranked 50 attributes for each domain and we performed classification with SABN and the Baseline-BN using 10-folds cross validation. For each domain, we report the number of attributes with the best accuracy. Table 3 shows the accuracies of the two classifiers on the three datasets.

With the attribute selection, we see that the accuracies of both SABN and Baseline-BN increased except for the Video domain. Again, we suggest that the accuracy of the SABN on the video domain could be improved with large dataset that may contain more representative attributes. Nevertheless, the accuracy of the SABN is still better than the Baseline-BN on the Video domain with the reduced attributes. We also performed experiment by using SABN with other

Table 3: Accuracies of SABN and Baseline-BN with the best ranked attribute sets.

Dataset	Ranked Attributes	SABN	Baseline-BN
Kitchen	50	75.7%	71.4%
Music	30	74.6%	69.7%
Video	50	77.8%	72.9%

scoring functions reported in Section 3.2 using the reduced attributes. The result in Table 4, shows that those scoring functions did not improve the result for SABN beyond the Bayes/K2 scoring function used in the earlier experiments. This is consistent with the comparative study conducted in De Campos (2006) on BN scoring functions.

Table 4: Experimental results using SABN with different scoring functions.

Function	Kitchen	Music	Video
K2/Bayes	75.7%	74.6%	77.8%
MDL	75.2%	71.7%	73.5%
BDeu	75.3%	71.8%	73.5%
Entropy	73.4%	70.0%	74.4%
AIC	75.2%	71.7%	72.8%

In terms of computational complexity, the SABN classifier has a training time complexity of $O(n^2.D)$ and a testing time complexity of $O(n)$, where n represents the count of the variables in the dataset and D denotes the size of the dataset. We believe this complexity is comparable with those of popular state-of-the-art classifiers, such as reported in Su & Zhang (2006). Overall, we have observed the SABN classifier to have reasonable performance that shows a promising research pathway for using Bayesian Network as a competitive alternative classifier for sentiment classification tasks.

6 Conclusion

In this study, we have proposed a sentiment augmented Bayesian network (SABN) classifier. The proposed SABN 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 three different datasets with the resulting network structure. Experimental results show that the proposed SABN has comparable, and in some cases, improved calcification accuracy with state-of-the-art sentiment classifiers. In future, we will experiment with SABN on cross-domain datasets and large scale sentiment datasets.

References

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

Aliferis, C. F., Statnikov, A., Tsamardinos, I., Mani, S. & Koutsoukos, X. D. (2010), '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* **11**, 171–234.

Bai, X. (2011), 'Predicting consumer sentiments from online text', *Decision Support Systems* **50**(4), 732–742.

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

Boiy, E. & Moens, M.-F. (2009), 'A machine learning approach to sentiment analysis in multilingual web texts', *Information Retrieval* **12**(5), 526–558.

Bouckaert, R. R. (2004), *Bayesian network classifiers in weka*, Department of Computer Science, University of Waikato.

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

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

Chen, X.-W., Anantha, G. & Lin, X. (2008), 'Improving bayesian network structure learning with mutual information-based node ordering in the k2 algorithm', *Knowledge and Data Engineering, IEEE Transactions on* **20**(5), 628–640.

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

Cheng, J. & Greiner, R. (1999), Comparing bayesian network classifiers, *in* 'Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence', Morgan Kaufmann Publishers Inc., pp. 101–108.

Cheng, J. & Greiner, R. (2001), Learning bayesian belief network classifiers: Algorithms and system, *in* 'Advances in Artificial Intelligence', Springer, pp. 141–151.

Choi, Y. & Cardie, C. (2008), Learning with compositional semantics as structural inference for subsentential sentiment analysis, *in* 'Proceedings of the Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Honolulu, Hawaii, pp. 793–801.

Chow, C. & Liu, C. (1968), 'Approximating discrete probability distributions with dependence trees', *Information Theory, IEEE Transactions on* **14**(3), 462–467.

Cooper, G. F. & Herskovits, E. (1992), 'A bayesian method for the induction of probabilistic networks from data', *Machine learning* **9**(4), 309–347.

De Campos, L. M. (2006), 'A scoring function for learning bayesian networks based on mutual information and conditional independence tests', *The Journal of Machine Learning Research* **7**, 2149–2187.

Esuli, A. (2008), 'Automatic generation of lexical resources for opinion mining: models, algorithms and applications', *SIGIR Forum* **42**(2), 105–106.

Esuli, A. & Sebastiani, F. (2006), 'Sentiwordnet: A publicly available lexical resource for opinion mining', *Proceedings of LREC* .

Friedman, N., Geiger, D. & Goldszmidt, M. (1997), 'Bayesian network classifiers', *Machine Learning* **29**, 131–163. 10.1023/A:1007465528199.
URL: *http://dx.doi.org/10.1023/A:1007465528199*

Friedman, N. & Yakhini, Z. (1996), On the sample complexity of learning bayesian networks, *in* 'Proceedings of the Twelfth international conference on Uncertainty in artificial intelligence', Morgan Kaufmann Publishers Inc., pp. 274–282.

Glover, F., Laguna, M. et al. (1997), *Tabu search*, Vol. 22, Springer.

Heckerman, D. (2008), *A tutorial on learning with Bayesian networks*, Springer.

Heckerman, D., Geiger, D. & Chickering, D. M. (1995), 'Learning bayesian networks: The combination of knowledge and statistical data', *Machine learning* **20**(3), 197–243.

Lee, P. M. (2012), *Bayesian statistics: an introduction*, John Wiley & Sons.

Liu, B. (2012), 'Sentiment analysis and opinion mining', *Synthesis Lectures on Human Language Technologies* **5**(1), 1–167.

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

Pang, B., Lee, L. & Vaithyanathan, S. (2002), 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, pp. 79–86.

Pearl, J. (1988), *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann.

Su, J. & Zhang, H. (2006), Full bayesian network classifiers, *in* 'Proceedings of the 23rd international conference on Machine learning', ACM, pp. 897–904.

Tang, H., Tan, S. & Cheng, X. (2009), 'A survey on sentiment detection of reviews', *Expert Systems with Applications* **36**(7), 10760–10773.

Turney, P. & Littman, M. L. (2002), Unsupervised learning of semantic orientation from a hundred-billion-word corpus, Technical report.

Wilson, T., Wiebe, J. & Hoffmann, P. (2005), Recognizing contextual polarity in phrase-level sentiment analysis, *in* 'Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Vancouver, British Columbia, Canada, pp. 347–354.