Classifying emotion in Twitter using Bayesian network

Muhammad Surya Asriadie, Mohamad Syahrul Mubarok and Adiwijaya
Fakultas Informatika, Telkom University, Jalan Telekomunikasi No. 1, Bandung, ID
E-mail: m.syahrulmubarok@telkomuniversity.ac.id, suryaa@student.telkomuniversity.ac.id, adiwijaya@telkomuniversity.ac.id

Abstract. Language is used to express not only facts, but also emotions. Emotions are noticeable from behavior up to the social media statuses written by a person. Analysis of emotions in a text is done in a variety of media such as Twitter. This paper studies classification of emotions on Twitter using Bayesian network because of its ability to model uncertainty and relationships between features. The result is two models based on Bayesian network which are Full Bayesian Network (FBN) and Bayesian Network with Mood Indicator (BNM). FBN is a massive Bayesian network where each word is treated as a node. The study shows the method used to train FBN is not very effective to create the best model and performs worse compared to Naive Bayes. F1-score for FBN is 53.71%, while for Naive Bayes is 54.07%. BNM is proposed as an alternative method which is based on the improvement of Multinomial Naive Bayes and has much lower computational complexity compared to FBN. Even though it’s not better compared to FBN, the resulting model successfully improves the performance of Multinomial Naive Bayes. F1-Score for Multinomial Naive Bayes model is 51.49%, while for BNM is 52.14%.

1. Introduction

Language is used to express not only facts, but also emotions. Emotions are noticeable from behavior up to the social media statuses written by a person. Classification of emotions in a text is done in a variety of media such as Twitter. Twitter is a popular social networking service where users can post and interact with messages called tweets. There are two types of users on Twitter, the user who tweets about themselves and the user who tweets to share information. In both cases, the tweets can provide information about emotion [4] [15]. Therefore, Twitter is considered to have suitable data for experiments related to emotion classification.

Classification of emotion has a wide range of applications. One of the example is for measuring the happiness index of people which is useful for political consideration for public well-being. In Indonesia, happiness index is measured through Happiness Index Measurement Survey held by Central Bureau of Statistics which is done only once every couple of years [2]. By applying emotion detection, the happiness index can be measured at anytime using any popular social media as the medium. Another example is to apply emotions detection in messaging application to check the message before sending the message to avoid misunderstanding. Emotions detection is also applicable to measure the popularity of a product or a brand [14].

Corresponding author.
Naive Bayes is one of the most commonly used methods for sentiment classification. Even though it usually has a good performance, it has a weakness to assume independencies between words [10]. Bayesian network is a probabilistic graphical model that can represent dependencies between variables, but there’s not so much research about sentiment classification using this method. The previous study of sentiment classification using Bayesian network is done by Wan and Gao [21] and Ortigosa-Hernández et al [16], but both of the studies did not focus on how Bayesian networks can be used to classify sentiment. In this paper, we aim to study emotion classification on Twitter using Bayesian network because of the ability to model uncertainty and relationships between features.

2. Related works

Sentiment analysis aims to extract subjective information from a text. In general, it can detect the writer’s attitude towards a topic. An example problem is to detect the polarity of sentiment in a text, whether it’s positive, negative or neutral. Sentiment analysis can also detect another type of aspect like emotional states.

Balabantaray, Mohammad, and Sharma studies emotion classification of Twitter data acquired from the web which are manually annotated by five judges [3]. The tweets are consists of seven classes, six Ekman emotion (joy, sadness, anger, disgust, surprise, and fear) and an addition of a neutral class. SVM then used to classify features that were constructed from the combination of Unigrams, Bigrams, Personal-pronouns, adjectives, Word-net Affect emotion lexicon, Word-net Affect emotion lexicon with left/right context, Word-net Affect emotion POS, POS, POS-Bigrams, Dependency-Parsing Feature, and Emoticons. The experiment was done using Leave-one-out cross-validation. The highest accuracy achieved is 73.24% that is accomplished using combination of all available features.

Roberts et al. studies emotion classification in twitter into seven classes which are anger, disgust, fear, joy, love, sadness, and surprise [20]. His dataset is acquired by selecting 14 specific subjects that are often believed to contain emotions. Seven Binary SVM are used to build the system; each of the Binary SVM is used to classify one class. Unigrams, Bigrams, Trigrams, Contains !, Contains ?, WordNet synsets, WordNet hypernyms, Topic scores, and Significant words are used as the features. The best feature combination of each class is selected using greedy additive feature selection. The experiment is done using 10-fold cross-validation with micro-averaged F1-Score of 66.8%.

Wan and Gao studies sentiment analysis of Twitter Data for Airline Services using ensemble classification strategy [21]. Majority vote principle is used to determine the result of multiple classification methods including Naive Bayes, SVM, Bayesian Network, C4.5 Decision Tree, and Random Forest. The study shows this method could improve the overall accuracy of the system. Bayesian Network can compete or perform even better than the other methods.

3. System design

In this study, we propose two models based on Bayesian Network called Full Bayesian Network and Bayesian Network with Mood Indicator. The following are the steps needed to build our models:

(i) Data

Data is prepared using k-fold cross-validation scheme with five folds. The data will be shuffled before divided using k-fold cross-validation.

(ii) Pre-processing

Pre-processing step is required to clean data and reduce the number of words used. The details of this step will be explained in the next section.
(iii) Binary Term Frequencies
Feature are represented using Binary Term Frequencies which treats every occurrence of words in a document as 1 and 0.

(iv) Feature selection
This step aims to reduce the number of features using chi-squared feature selection algorithm.

(v) Structure learning
Structure Learning aims to find the best structure of a Bayesian Network based on data. The structure of Full Bayesian Network is optimized using simulated annealing search and $f_{CLL}$ as the scoring function. Bayesian Network with Mood Indicator doesn’t require structure learning because the number of the possible structure is small.

(vi) Parameter learning
The probability distribution of each node is learned on train data using MAP-estimation.

(vii) Evaluation
The performance of the model is evaluated using F1-Score.

4. Dataset
In this experiment, we use #Emotional Tweets datasets by Saif Mohammad that is acquired by making use of hashtag feature on Twitter [13]. It consists of six classes with 21051 total tweets (Figure 1).

4.1. Pre-processing
Data gathered from the web is usually not clean and contains a lot of unnecessary information. Therefore, pre-processing steps are required to clean the data. The pre-processing steps are as follow:

(i) **HTML escape**
Data gathered from the web usually contain escaped character as a form of HTML code (&lt;, &gt; dan &amp;). This process aims to convert HTML code back to their original character.

![Class distribution](image-url)
Table 1: #Emotional Tweets’s dataset sample for each emotion classes.

| Tweet                                                                 | Emosi   |
|----------------------------------------------------------------------|---------|
| Soooo doowwwn!! Move on, get some sleep... Me deserve better.#forgetit | anger   |
| #yawning                                                             |         |
| Making art and viewing art are different at their core!              |         |
| Can parents teach kids how to cover their mouths when they cough?    | fear    |
| Ewah.                                                                |         |
| the moment when you get another follower and you cheer.               | disgust |
| Joy                                                                   |         |
| I said no to the same boy like 5 times last night!                   |         |
| Tomorrow I’m going to get @HerBraceFaceAss nd Jaela frmschool early nd takin my grls to lunch. L0L | sadness |
|                                                                    | surprise|

(ii) Apostrophe lookup

In this step, every word that contains apostrophes is converted into standard lexicons. For example I’ve become I have.

(iii) Username lookup

In twitter, every user can mention their friends in their tweets by prepending @ symbol before their friend’s username. This step aims to group usernames into one tag which is \{username\}.

(iv) URL lookup

It’s not rare to see that Twitter is used as a medium to share information or links to other websites. This step transforms links into one tag which is \{url\}.

(v) Money lookup

In this step, every word that begins with money symbol (example: $) is transformed into one tag which is \{money\}.

(vi) Remove number

This step converts every numbers into one tag which is \{number\}.

(vii) Remove repeats

This step remove repeated characters of the text.

(viii) Stemming

This step aims to transform a word into its base or root form. Porter Stemmer algorithm is used in this step [18].

(ix) Remove punctuations

In this step, punctuations are removed of the text.

(x) Hashtag removal

Hashtag is a feature where users can tag tweets into a specific topic. It’s written by prepending “#” into a word which usually a topic. To reduce the number of features while retaining the information, any hashtag that rarely appears is grouped into a specific tag \{hashtag\}, while any hashtag that often appears remains.

(xi) Stop words removal

In this step, every word that doesn’t contain significant meaning is removed.

5. Bayesian Network

Bayesian Network is a graphical representation of probabilistic relationships between several random variables [19]. It’s very suitable for representing knowledge about an uncertain domain.
Each node in a Bayesian Network represents a random variable, while each edge represents the dependency between the random variables.

5.1. Parameter learning
Parameter learning aims to estimate the probability distribution of every node in Bayesian Network based on data. Some of the methods to learn parameters using complete data are Maximum Likelihood Estimate (ML-estimation) and Maximum A Posteriori Estimation (MAP-estimation) [19]. MAP-estimation can be seen as the general form of ML-estimation, where it adds information about prior knowledge to the estimation. MAP-estimation with equivalent sample size is formulated as follow:

\[
\theta_{ijk} = \frac{\frac{\alpha}{r_i q_i} + N_{ijk}}{\sum_k \frac{\alpha}{r_i q_i} + N_{ijk}}
\]  

(1)

The explanation of each symbol is in the next section.

5.2. Structure learning
Structure learning is used to learn a Bayesian Network structure from data. There are two kinds of structure learning algorithm, constrained-based method, and heuristic search. Hill climbing, tabu search, and simulated annealing are some of the methods based on heuristic search. Simulated Annealing is a heuristic algorithm inspired by the process of annealing in metallurgy. It's often used when the search space is discrete (e.g., Traveling Salesman Problem).
Algorithm 1 Simulated Annealing for Bayesian Network structure learning [5]

1: function SIMULATEDANNEALING($\tau_0, T_{ni}, ni$)
2:   for $i = 1, \ldots, n$ do
3:      $\pi_i \leftarrow \emptyset$
4:      $\pi_{best,i} \leftarrow \emptyset$
5:   end for
6:   $T \leftarrow \tau_0 \cdot N$
7:   $\alpha \leftarrow (T_{ni}/T_0)^{1/ni}$
8:   $k \leftarrow 0$
9: repeat
10:   repeat
11:      select two indices $i, j$ randomly
12:      until $v_j \in \pi_j$ or adding $v_j$ to $\pi_i$ does not introduce cycle
13:      if $v_j \in \pi_i$ then
14:         if $\exp(m(v_i, \pi_i \setminus v_j) \leftrightarrow m(v_i, \pi_i)) > \text{random}[0..1]$ then
15:            $\pi_i \leftarrow \pi_i \setminus v_j$
16:         end if
17:      else
18:         if $\exp(m(v_i, \pi_i v_j) \leftrightarrow m(v_i, \pi_i)) > \text{random}[0..1]$ then
19:            $\pi_i \leftarrow \pi_i v_j$
20:         end if
21:      end if
22:      if $\sum_{i=1}^{n} m(v_i, \pi_{best,i}) < \sum_{i=1}^{n} m(v_i, \pi_i)$ then
23:         $\forall i \in \{1, \ldots, n\} \pi_{best,i} \leftarrow \pi_i$
24:      end if
25:   until $k = ni$
26:   return $\pi_{best,1}, \ldots, \pi_{best,n}$
27: $k \leftarrow k + 1$
28: end function

Where $\pi_i$ is the parent node of $v_i$ and $\pi_{best,i}$ as its best parent so far. $T$ is temperature and $\tau_0$ is the initial factor for initial temperature $T_0$ based on total nodes $N$. $\alpha$ is temperature change factor based on final temperature $T_{ni}$, initial temperature $T$ and maximum iteration $ni$. $m$ is optimization function used to calculate fitness score.

Optimization function is used to find the best structure on each iteration based on their score. Factorized Conditional Log Likelihood ($\hat{f}CLL$) is a discriminative and decomposable optimization function based on an approximation of conditional log likelihood [8].

\[
\hat{f}CLL(G|D) = (\alpha + \beta)\hat{L}(B|D) - \beta \lambda \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \sum_{c=0}^{1} N_{ijk}(\log(\frac{N_{ijck}}{N_{ij+k}}) - \log(\frac{N_{ijc}}{N_{ij}})) - \beta \lambda \sum_{c=0}^{1} N_c \log(\frac{N_c}{N}) - \beta N \rho
\]  

(2)

Where $n$ is the number of nodes, $r_i$ is the number of possible random variable values, $q_i$ is the number of possible parent nodes configuration, $q_i^*$ is $q_i$ without class node as its parent and $C$ is the total number of classes.
\( \alpha, \beta, \) dan \( \Lambda \) is a constant that minimalize the mean squared error (MSE) of \( \hat{f}(U_t, V_t) = \alpha \log(U_t) + \beta \log(V_t) + \lambda \) to approximate \( f = \log(U_t / U_t + V_t) \). Where \( \hat{f} \) is an approximation of \( f \).

\[
\alpha = \frac{\pi^2 + 6}{24} \tag{3}
\]

\[
\beta = \frac{\pi^2 - 18}{24} \tag{4}
\]

\[
\lambda = \frac{\pi^2}{12 \ln 2} - (2 + \frac{(\pi^2 - 6) \log p}{12}) \tag{5}
\]

\( \hat{LL} \) is Log-Likelihood defined as follow [7]:

\[
\hat{LL}(G|D) = \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijk} \log \left( \frac{N_{ijk}}{N_{ij}} \right) \tag{6}
\]

The observed parameters are defined as follow:

\[
N_{ij} = \sum_{k=1}^{r_i} N_{ijk} \tag{7}
\]

\[
N_{ijck} = \sum_{c=1}^{s} N_{ijck} \tag{8}
\]

\[
N_{ijc} = \sum_{k=1}^{r_i} N_{ijck} \tag{9}
\]

\[
N_{ij*} = \sum_{c=1}^{s} \sum_{k=1}^{r_i} N_{ijck} \tag{10}
\]

\[
N_c = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} N_{ijck} \tag{11}
\]

Where \( N \) is the documents in the datasets, \( X_i \) is observed variable, \( PA_i \) is parent node of \( X_i \) and \( C_i \) is the label of \( X_i \). \( N_{ijk} \) is the number of appearances when \( X_i = k \) and \( PA_i = j \). \( N_{ijck} \) is the number of appearances when \( X_i = k \), \( PA_i = j \) and \( C_i = c \).

6. Bayesian Network models
This section describes Full Bayesian Network and Bayesian Network with Mood Indicator model in more detail.

6.1. Full Bayesian Network (FBN)
Full Bayesian Network uses words as nodes with the addition of one class node. The result is a massive Bayesian Network structure consist of more than one thousand nodes. Chi-squared feature selections method is used to reduce the number of nodes that are independent towards the document’s class.

A metaheuristic algorithm, simulated annealing is used to approximate the best Bayesian Network structure in the solution space. Fitness score is calculated using \( \hat{f} \text{CLL} \) because it’s discriminative and decomposable. The initial structure of Bayesian Network is set to Naive Bayes to allow a faster optimization. To avoid overfitting because of \( \hat{f} \text{CLL} \), the number of maximum parent nodes is limited.
6.2. Bayesian Network with Mood Indicator (BNM)
Bayesian Network with Mood Indicator is an improvement of Multinomial Naive Bayes by connecting indicator nodes to the class node. It affects the class prior probability, resulting in a better inference performance. There are two indicator nodes used by the model, a node for positive mood indicator and a node for negative mood indicator. We use emoticons in tweets to define the tweet’s mood, it’s positive when any emoticons that show a happy expression exist and negative mood otherwise.

This model doesn’t require a structure learning algorithm because the maximum number of nodes are three. We obtain the best structure by iterating all possible structures.

7. Experiment and analysis
This section describes the result and analysis of the experiment for Full Bayesian Network and Bayesian Network with Mood Indicator.

7.1. Full Bayesian Network
Performance analysis on test and train Data

Table 2: The performance of Naive Bayes (NB) and Full Bayesian Network (FBN) using maximum two and three parent nodes on train data for every significant level.

| Model         | Significant Level | Precision | Recall  | F1-Score |
|---------------|-------------------|-----------|---------|----------|
| NB            | 0.001             | 62.76%    | 61.45%  | 60.28%   |
| NB            | 0.01              | 65.09%    | 63.88%  | 62.96%   |
| NB            | 0.1               | 72.27%    | 70.92%  | 70.37%   |
| FBN 2 Parents | 0.001             | 63.02%    | 61.69%  | 60.5%    |
| FBN 2 Parents | 0.01              | 65.5%     | 64.28%  | 63.35%   |
| FBN 2 Parents | 0.1               | 72.67%    | 71.29%  | 70.73%   |
| FBN 3 Parents | 0.001             | 63.2%     | 61.85%  | 60.66%   |
| FBN 3 Parents | 0.01              | 65.69%    | 64.45%  | 63.52%   |
| FBN 3 Parents | 0.1               | 72.89%    | 71.48%  | 70.92%   |

Table 2 shows the performance of Naive Bayes and Full Bayesian Network on train data. Full Bayesian Network has better performance over Naive Bayes for every significant level. This shows that optimizing conditional log-likelihood (using \(\hat{f}_{CLL}\)) could affect the performance of Bayesian Network for classification.
Table 3: The performance of Naive Bayes (NB) and Full Bayesian Network (FBN) using maximum two and three parent nodes on test data for every significant level.

| Model         | Significant Level | Precision | Recall  | F1-Score |
|---------------|-------------------|-----------|---------|----------|
| NB            | 0.001             | 54.74%    | 54.56%  | 52.75%   |
| NB            | 0.01              | 55.58%    | 55.46%  | 53.89%   |
| NB            | 0.1               | 55.46%    | 55.74%  | 54.07%   |
| FBN 2 Parents | 0.001             | 54.60%    | 54.50%  | 52.68%   |
| FBN 2 Parents | 0.01              | 55.32%    | 55.30%  | 53.71%   |
| FBN 2 Parents | 0.1               | 54.37%    | 54.89%  | 53.24%   |
| FBN 3 Parents | 0.001             | 54.48%    | 54.42%  | 52.60%   |
| FBN 3 Parents | 0.01              | 55.00%    | 55.09%  | 53.49%   |
| FBN 3 Parents | 0.1               | 54.11%    | 54.65%  | 53.01%   |

Table 3 shows the performance of Naive Bayes and Full Bayesian Network on test data. Full Bayesian Network with maximum two parent nodes is better compared to Full Bayesian Network with maximum three parent nodes. However, Full Bayesian Network has worse performance compared to Naive Bayes. This shows the improvement of the conditional log-likelihood score doesn’t affect the performance of model directly.

**Performance analysis on each Iteration**

To analyze the cause of poor performance of Full Bayesian Network compared to Naive Bayes, we compare the performance of the best FBN model on each iteration towards the train and test data.

![Figure 2: The performance of first fold’s FBN 2 parents with significant level 0.01 on each iteration towards train data on the first cross-validation’s fold.](image-url)
Figure 2 and 3 shows the performance of Bayesian Network on each iteration. The performance on train data is drastically improved in the beginning of the iteration before declined and became stable in the later iteration. The performance on test data declined drastically in the beginning of the iteration before it improves and becomes stable in the later iteration. This shows the improvement of \( \hat{f}_{CLL} \) doesn’t directly affect the performance of the Bayesian Network which makes it possible for the model to perform worse when the \( \hat{f}_{CLL} \) score is improved. It’s caused by the characteristics of \( \hat{f}_{CLL} \) which is an approximation of conditional log-likelihood. The improvement of conditional log-likelihood not only able to improve the performance but can also decline the performance. The performance declined when conditional log-likelihood is improved by improving the ability to recognize a class while sacrificing the ability to recognize the other class.

**Performance analysis for each emotion**

To analyze the cause of poor performance of Full Bayesian Network compared to Naive Bayes, we analyze the performance of the best FBN model on each iteration towards the train and test data.

Table 4: The performance of first fold’s FBN using maximum two parent nodes with significant level of 0.01 for each emotion.

| Emotion | Precision | Recall | F1-Score | Precision | Recall | F1-Score |
|---------|-----------|--------|----------|-----------|--------|----------|
| anger   | 62.25%    | 48.93% | 54.79%   | 44.64%    | 39.30% | 39.17%   |
| disgust | 86.74%    | 72.71% | 79.11%   | 29.79%    | 18.79% | 23.05%   |
| fear    | 79.16%    | 55.82% | 65.47%   | 64.92%    | 44.93% | 53.11%   |
| joy     | 63.28%    | 83.98% | 72.18%   | 60.54%    | 79.11% | 68.59%   |
| sadness | 51.99%    | 52.44% | 52.21%   | 43.86%    | 45.40% | 44.61%   |
| surprise| 67.11%    | 42.68% | 52.17%   | 63.62%    | 41.56% | 50.27%   |

Table 4 shows the performance of Full Bayesian Network for each emotion. The overall
performance of the model on train data is better compared to test data with each emotion scored at least 50% F1-score, meanwhile the performance on test data is worse compared to train data, especially for anger, disgust, and fear emotions which have lesser number of samples compared to other emotions. It’s likely caused by the class imbalance of the data which led to miss classification.

**Inference complexity analysis**

There are only two possible values (*true* when the word exist in a document and *false* otherwise) for each word node because the data is represented using Binary Term Frequency. Because of that, the value of each node become known and marginalization can be avoided. Therefore the inference of complexity of Full Bayesian Network is the same as Naive Bayes, which is $O(CN)$. Where $C$ is the number of possible class and $N$ is the number of nodes.

7.2. **Bayesian Network with Mood Indicator**

This section analyzes the performance of Bayesian Network with Mood Indicator. Figure 4 and table 5 shows the structure and the performance respectively.

Table 5: The performance of Bayesian Network with Mood Indicator for every possible structures (Figure 4).

| Struktur | Precision | Recall | F1-Score |
|----------|-----------|--------|----------|
| 1        | 54.97%    | 54.98% | 51.49%   |
| 2        | 55.08%    | 55.0%  | 51.64%   |
| 3        | 55.56%    | 55.44% | 52.13%   |
| 4        | 55.56%    | 55.44% | 52.13%   |
| 5        | **55.57%**| **55.44%**| **52.14%**|
| 6        | 55.57%    | 55.44% | 52.14%   |
| 7        | **55.57%**| **55.44%**| **52.14%**|
| 8        | 55.08%    | 55.0%  | 51.64%   |
| 9        | **55.57%**| **55.44%**| **52.14%**|
| 10       | 55.56%    | 55.44% | 52.13%   |
| 11       | **55.57%**| **55.44%**| **52.14%**|
| 12       | **55.57%**| **55.44%**| **52.14%**|
| 13       | **55.57%**| **55.44%**| **52.14%**|
| 14       | 55.08%    | 55.0%  | 51.64%   |
| 15       | 55.08%    | 55.0%  | 51.64%   |
| 16       | **55.57%**| **55.44%**| **52.14%**|
| 17       | **55.57%**| **55.44%**| **52.14%**|
| 18       | **55.57%**| **55.44%**| **52.14%**|
| 19       | 55.08%    | 55.0%  | 51.64%   |
| 20       | 55.08%    | 55.0%  | 51.64%   |
| 21       | 55.49%    | 55.3%  | 51.97%   |
| 22       | 55.49%    | 55.3%  | 51.97%   |
| 23       | 55.49%    | 55.3%  | 51.97%   |
| 24       | 55.49%    | 55.3%  | 51.97%   |
| 25       | 55.5%     | 55.32% | 51.99%   |
| 26       | 55.49     | 55.3%  | 51.97%   |

Table 5 shows structure 5, 6, 7, 9, 11, 12, 13, 16, 17 and 18 has the better performance compared to the other structures. The addition of indicator node also improves the performance
Figure 4: All possible structures of Bayesian Network with Mood Indicator. Words node is shown using plate notation which illustrates a Multinomial Naive Bayes relationship. The emotion’s node is the class node while the positive and the negative node is the indicator nodes. Only nodes within the Markov Blanket of the class node are displayed.
of the model compared to Multinomial Naive Bayes without indicator node. Additional information provided by indicator node affects the prior probability of the class, resulting in better posterior probability towards a class.

**Inference complexity analysis**

The inference complexity of Multinomial Naive Bayes depends on the input data. The complexity is $O(CW)$, where $C$ is the number of classes and $W$ is the number of terms (input data). Bayesian Network with Mood Indicator has the similar complexity because of the modification on prior calculation. The complexity is $O((2+C)W)$, where 2 comes from the number of nodes that affects class node, which is positive and negative nodes.

8. Conclusion

We have presented the experiment result of Full Bayesian Network and Bayesian Network with Mood Indicator for emotion classification on twitter data. Based on the result of the experiment, it shows that the performance of both model is not good enough for classifying emotion from twitter data due to the simplicity for the model.

Full Bayesian Network trained using Simulated Annealing and $\hat{f}_{CLL}$ performs better compared to Naive Bayes on train data. However, Full Bayesian Network has worse performance compared to Naive Bayes on test data. It’s caused by the characteristics of $\hat{f}_{CLL}$ which is an approximation of conditional log-likelihood. In future work, we suggest trying different search algorithm and the score function that directly affects the performance (e.g., classification rate).

Even though it’s not better compared to Full Bayesian Network, Bayesian Network with Mood Indicator successfully improves the performance of Multinomial Naive Bayes with much lower computational complexity. The addition of indicator node on Multinomial Naive Bayes could improve the performance of Multinomial Naive Bayes. For future work, we suggest trying different indicator nodes such as the result of K-means clustering on the text and experimenting with n-grams for the multinomial naive bayes features.

**References**

[1] Grimace. http://www.grimace-project.net/. Accessed: 2016-11-10.
[2] Indeks kebahagiaan indonesia tahun 2017. Badan Pusat Statistik Indonesia, August 2017.
[3] Balabantaray, R. C., Mohammad, M., and Sharma, N. Multi-class twitter emotion classification: A new approach. International Journal of Applied Information Systems 4, 1 (September 2012), 48–53. Published by Foundation of Computer Science, New York, USA.
[4] Bollen, J., Mao, H., and Pepe, A. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena, 2011.
[5] Bouckaert, R. Bayesian Belief Networks: from Construction to Inference. PhD thesis, Utrecht, Netherlands, 1995.
[6] Bougie, R., Pieters, R., and Zeeelenberg, M. Angry customers don’t come back, they get back: The experience and behavioral implications of anger and dissatisfaction in services. Journal of the Academy of Marketing Science 31, 4 (2003), 377–393.
[7] Carvalho, A. M. Scoring functions for learning bayesian networks, 2009.
[8] Carvalho, A. M., Roos, T., Oliveira, A. L., and Myllymäki, P. Discriminative learning of bayesian networks via factorized conditional log-likelihood. J. Mach. Learn. Res. 12 (July 2011), 2181–2210.
[9] Ekman, P. Facial expression and emotion. American Psychologist 48, 4 (1993), 384–392.
[10] Gamallo, P., and Garcia, M. Citius: A naive-bayes strategy for sentiment analysis on english tweets. pp. 171–175.
[11] Korb, K. B., and Nicholson, A. E. Bayesian Artificial Intelligence, Second Edition, 2nd ed. CRC Press, Inc., Boca Raton, FL, USA, 2010.
[12] Manning, C. D., Raghavan, P., and Schütze, H. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA, 2008.
[13] Mohammad, S. M. #emotional tweets. In Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the Main Conference and the Shared Task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (Stroudsburg, PA, USA, 2012), SemEval ’12, Association for Computational Linguistics, pp. 246–255.
[14] Mohammad, S. M., and Yang, T. W. Tracking sentiment in mail: How genders differ on emotional axes. In Proceedings of the 2Nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (Stroudsburg, PA, USA, 2011), WASSA ’11, Association for Computational Linguistics, pp. 70–79.

[15] Naaman, M., Boase, J., and Lai, C.-H. Is it really about me?: Message content in social awareness streams. In Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work (New York, NY, USA, 2010), CSCW ’10, ACM, pp. 189–192.

[16] Ortigosa-Hernández, J., Rodríguez, J. D., Alzate, L., Lucania, M., Inza, I., and Lozano, J. A. Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. Neurocomputing 92 (2012), 98 – 115. Data Mining Applications and Case Study.

[17] Pang, B., and Lee, L. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval 2, 12 (2008), 1–135.

[18] Porter, M. F. Readings in information retrieval. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997, ch. An Algorithm for Suffix Stripping, pp. 313–316.

[19] Rish, I. A tutorial on inference and learning in bayesian networks. IBM T.J.Watson Research Center, 2000.

[20] Roberts, K., Roach, M. A., Johnson, J., Guthrie, J., and Harabagiu, S. M. Empatweet: Annotating and detecting emotions on twitter. In Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC’12) (Istanbul, Turkey, may 2012), N. C. C. Chair), K. Choukri, T. Declerck, M. U. Doan, B. Maegaard, J. Mariani, A. Moreno, J. Odijk, and S. Piperidis, Eds., European Language Resources Association (ELRA).

[21] Wan, Y., and Gao, Q. An ensemble sentiment classification system of twitter data for airline services analysis. In 2015 IEEE International Conference on Data Mining Workshop (ICDMW) (Nov 2015), pp. 1318–1325.

[22] Adiwijaya. Aplikasi Matriks dan Ruang Vektor. In Graha Ilmu (Yogyakarta, 2014).

[23] Adiwijaya. Matematika Diskrit dan Aplikasinya. In Alfabeta (Bandung, 2014).

[24] MS Mubarok, Adiwijaya, MD Aldhi. Aspect-based sentiment analysis to review products using Nave Bayes. In AIP Conference Proceedings 1867, 020060 (2017).

[25] Aziz, R.A., Mubarok, M.S. and Adiwijaya. Klasifikasi Topik pada Lirik Lagu dengan Metode Multinomial Naive Bayes. In Indonesia Symposium on Computing (IndoSC) (2016).

[26] Arifin, A.H.R.Z., Mubarok, M.S. and Adiwijaya. Learning Struktur Bayesian Networks menggunakan Novel Modified Binary Differential Evolution pada Klasifikasi Data. In In Indonesia Symposium on Computing (IndoSC) (2016).

[27] Yulita, I.N., Houw Liong The and Adiwijaya. Fuzzy Hidden Markov Models for Indonesian Speech Classification. In JACIII, 16(3), pp.381-387 (2012).

[28] Pratiwi, A.I., Adiwijaya. On The Feature Selection and Classification Based on Information Gain for Document Sentiment Analysis. In Applied Computational Intelligence and Soft Computing, (2017).