Benchmarking the Semi-Supervised Naïve Bayes Classifier

Awat Saeed
School of Computing Sciences
University of East Anglia
Norwich, NR4 7TJ
United Kingdom
Email: Awat.Saeed@uea.ac.uk

Gavin C. Cawley
School of Computing Sciences
University of East Anglia
Norwich, NR4 7TJ
United Kingdom
Email: G.Cawley@uea.ac.uk

Anthony Bagnall
School of Computing Sciences
University of East Anglia
Norwich, NR4 7TJ
United Kingdom
Email:Anthony.Bagnall@uea.ac.uk

Abstract—**Semi-supervised learning involves constructing predictive models with both labelled and unlabelled training data. The need for semi-supervised learning is driven by the fact that unlabelled data are often easy and cheap to obtain, whereas labelling data requires costly and time consuming human intervention and expertise. Semi-supervised methods commonly use self training, which involves using the labelled data to predict the unlabelled data, then iteratively reconstructing classifiers using the predicted labels. Our aim is to determine whether self training classifiers actually improves performance. Expectation maximization is a commonly used self training scheme. We investigate whether an expectation maximization scheme improves a naïve Bayes classifier through experimentation with 30 discrete and 20 continuous real world benchmark UCI datasets. Rather surprisingly, we find that in practice the self training actually makes the classifier worse. The cause for this detrimental affect on performance could either be with the self training scheme itself, or how self training works in conjunction with the classifier. Our hypothesis is that it is the latter cause, and the violation of the naïve Bayes model assumption of independence of attributes means predictive errors propagate through the self training scheme. To test whether this is the case, we generate simulated data with the same attribute distribution as the UCI data, but where the attributes are independent. Experiments with this data demonstrate that semi-supervised learning does improve performance, leading to significantly more accurate classifiers. These results demonstrate that semi-supervised learning cannot be applied blindly without considering the nature of the classifier, because the assumptions implicit in the classifier may result in a degradation in performance.**

I. Introduction

Supervised machine learning methods usually require large amounts of labelled data to achieve good classification performance. However, situations where unlabelled data are easy and cheap to obtain, but labelling data is costly and time consuming are observed in many fields, such as social network and micro array analysis. The field of semi-supervised learning [1] involves developing techniques that can leverage useful predictive information from the unlabelled data.

Semi-supervised learning involves constructing predictive models with both labelled and unlabelled training data. Semi-supervised methods commonly use self training, which involves using the labelled data to predict the unlabelled data, then iteratively reconstructing classifiers using the predicted labels. Our aim is to determine whether self training classifiers actually improves performance in order to determine under what circumstances semi-supervised learning is useful. Previous research has claimed that using unlabeled data can improve naïve Bayes classification performance [2]. However, it has also been shown that unlabelled data can degrade classification performance [3], [4]. We experimentally evaluate the effect on a naïve Bayes classifier of using expectation maximization as a self training mechanism. We perform two sets of experiments with 30 discrete and 20 continuous real world benchmark UCI datasets. We find that self training significantly decreases the accuracy of a naïve Bayes classifier. This demonstrates that the blind application of semi-supervised learning is not guaranteed to improve performance, and that understanding the nature of the classifier is crucial in using it correctly. The reason for the detrimental affect on performance of semi-supervised learning on naïve Bayes could be caused by the self training scheme or the interaction of the expectation maximization algorithm and the classifier. Our hypothesis is that it is the latter cause. Naïve Bayes makes the assumption of independence of attributes, and we believe that this extreme assumption will means predictive errors caused by early iterations of expectation maximization will propagate through the self training scheme.

To test whether this is the case, we generate simulated data based on the UCI data. We do this by fitting a naïve Bayes classifier to the original data, then using the model attribute distribution estimates to generate simulated data. This means the simulated data maintains the characteristics of the original data and will fulfill the assumptions of the naïve Bayes classifier. Experiments with this data demonstrate that semi-supervised learning does improve performance, leading to significantly more accurate classifiers. These results demonstrate that semi-supervised learning cannot be applied blindly without considering the nature of the classifier, because the assumptions implicit in the classifier may result in a degradation in performance.

Following the discussion of learning methods in section II, we present the result of extensive experiments on different real and synthetic benchmark datasets in sections III and conclude in section IV.

II. Background

We start with an overview of Naïve Bayes (NB) [5] in section II-A, and then introduce Semi-Supervised Naïve Bayes (SSNB) that uses the Expectation-Maximization (EM) algorithm [6], in section II-B.

A. The Naïve Bayes Classifier

In a supervised classification setting, assume we are given labelled training data, $D = \{(x^{(i)}, y^{(i)})\}_{i=1}^{l}$, where $x^{(i)} \in \mathcal{X} \subseteq \mathcal{R}^d$ is a feature vector describing the ith example with class label $y^{(i)} \in \{1, 2, \ldots, C\}$. Each example $(x^i, y^{(i)})$ from the training data (that is assumed to be an independent, identically distributed (iid) sample drawn from a fixed distribution) are used to obtain estimates of the model parameters, denoted by $\hat{\theta}$.
The task of the NB classifier is the prediction of class labels (y) for a new pattern (x) by modelling the class conditional probability $p(x|y;\theta)$ and the prior probability $p(y;\theta)$, where θ are the model parameters, and then using Bayes' rule to estimate the posterior probability of class membership for all classes $p(y|x;\theta)$ after parameter estimation [7] [8].

$$p(y = c|x;\theta) \;=\; \frac{p(y;\theta)\; p(x|y;\theta)}{\sum_{k=1}^{C} p(y = k;\theta)\; p(x|y = k;\theta)}$$

The summation in the denominator is over all class labels k. The test pattern (x) is classified as a single class by selecting the maximum posterior probability of class membership according to the classification rule.

$$\hat{y} \;=\; \arg\max_{c}\; p(y = c|x;\theta)$$

One common way to find optimal model parameters, $\hat{\theta}$, is the maximum likelihood estimate (MLE)

$$\hat{\theta} \;=\; \arg\max_{\theta}\; \log p(\mathcal{D};\theta)$$

The NB assumption can be used to reduce complexity for learning the Bayesian classifier by making the strong assumption that the input attributes are independent of each other. The independence assumption is often unrealistic in the real world but it simplifies the estimation $p(x|y;\theta)$ from the training samples. Therefore, it is particularly suitable when the dimensionality of the input attributes is so high that a large number of parameters must be estimated [9].

$$\begin{aligned} p(x|y;\theta) &= p(x_1, x_2, ..., x_d|y;\theta) \\ &= p(x_1|y;\theta)\;.\;p(x_2|y;\theta)\;...\;p(x_d|y;\theta) \\ &= \prod_{j=1}^{d} p(x_j|y;\theta) \end{aligned}$$

As we assumed the training data comprise an iid sample, the likelihood given as follows

$$\begin{aligned} p(\mathcal{D};\theta) &= \prod_{i=1}^{l} p(x_j^{(i)}, y^{(i)};\theta) \\ &= \prod_{i=1}^{l} \left(p(y^{(i)};\theta) \prod_{j=1}^{d} p(x_j^{(i)}|y^{(i)};\theta) \right) \end{aligned}$$

Instead of maximising the likelihood $p(\mathcal{D};\theta)$ we work with the log-likelihood $(log\, p(\mathcal{D};\theta))$.

$$\begin{aligned} log\, p(\mathcal{D};\theta) &= \sum_{i=1}^{l} log \left(p(y^{(i)};\theta) \prod_{j=1}^{d} p(x_j^{(i)}|y^{(i)};\theta) \right) \\ &= \sum_{i=1}^{l} log\, p(y^{(i)};\theta) + \sum_{i=1}^{l}\sum_{j=1}^{d} log\, p(x_j^{(i)}|y^{(i)};\theta) \end{aligned} \tag{1}$$

1) Maximum likelihood Estimation for Categorical Distribution: The categorical distribution is the discrete distribution for handling nominal data. Suppose the attributes come from categorical distribution that each attribute, x_j, has (S_j) possible values (states), $x_j \in \{1, 2, \ldots, S_j\}$. So, the i^{th} example indicates one of the (S_j) value $x_j^{(i)} = s$.
The likelihood of observing a state $x_j^{(i)} = s$ is denoted by $\theta_{sc}^{j} = p(x_j^{(i)} = s|y^{(i)} = c)$ which is the probability of attribute value $x_j^{(i)} = s$ in class c where $\sum_{s=1}^{S} \theta_{sc}^{j} = 1$ and $\pi_c = p(y = c)$ is the class prior probabilty for class c where $\sum_{c=1}^{C} \pi_c = 1$. If $x_j \sim cat(\theta)$ then equation (1) can be written more explicitly in terms of the parameters.

$$\begin{aligned} \log p(\mathcal{D};\pi,\theta) &= \sum_{i=1}^{l}\sum_{c=1}^{C} \phi(y^{(i)} = c) \log \pi_c \\ &+ \sum_{i=1}^{l}\sum_{j=1}^{d}\sum_{s=1}^{S}\sum_{c=1}^{C} \log cat(x_j^{(i)}|y^{(i)};\theta_{sc}^{j}) \\ &= \sum_{i=1}^{l}\sum_{c=1}^{C} \phi(y^{(i)} = c) \log \pi_c \\ &+ \sum_{i=1}^{l}\sum_{j=1}^{d}\sum_{s=1}^{S}\sum_{c=1}^{C} \phi(x_j^{(i)} = s \wedge y^{(i)} = c) \log \theta_{sc}^{j} \end{aligned}$$

$\phi(z) = 1$ where z is true, and $\phi(z) = 0$ otherwise.
The log-likelihood can be maximised with respect to the parameters (θ_{sc}^{j}, π_c) using Lagrange multiplers (α, β_c^{j}) to enforce the constraints that the class priors and class-conditional probabilities must sum to one [10]. The log-likelihood with Lagrangian terms is given as follows.

$$\begin{aligned} \Lambda(\pi,\theta,\alpha,\beta) &= \sum_{i=1}^{l}\sum_{c=1}^{C} \phi(y^{(i)} = c) \log \pi_c \\ &+ \sum_{i=1}^{l}\sum_{j=1}^{d}\sum_{s=1}^{S}\sum_{c=1}^{C} \phi(x_j^{(i)} = s \wedge y^{(i)} = c) \log \theta_{sc}^{j} \\ &- \alpha\left(\sum_{c=1}^{C} \pi_c - 1\right) - \sum_{c=1}^{C}\sum_{j=1}^{d} \beta_c^{j}\left(\sum_{s=1}^{S} \theta_{sc}^{j} - 1\right) \end{aligned} \tag{2}$$

In order to obtain the maximum likelihood solution for the parameters, the partial derivatives can be computed for equation (2) with respect to all the parameters, and setting each partial derivative to zero.

$$\begin{aligned}
\frac{\partial \Lambda}{\partial \alpha} &= 0 \Rightarrow \sum_{c=1}^{C} \pi_c = 1 \\
\frac{\partial \Lambda}{\partial \beta_c^j} &= 0 \Rightarrow \sum_{s=1}^{S} \theta_{sc}^j = 1 \\
\frac{\partial \Lambda}{\partial \pi_c} &= 0 \Rightarrow \\
\pi_c &= \frac{\sum_{i=1}^{l} \phi(y^{(i)} = c)}{\sum_{k=1}^{C} \sum_{i=1}^{l} \phi(y^{(i)} = k)} \\
\frac{\partial \Lambda}{\partial \theta_{sc}^j} &= 0 \Rightarrow \\
\theta_{sc}^j &= \frac{\sum_{i=1}^{l} \phi(x_j^{(i)} = s \wedge y^{(i)} = c)}{\sum_{m=1}^{S} \sum_{i=1}^{l} \phi(x_j^{(i)} = m \wedge y^{(i)} = c)}
\end{aligned}$$

In some cases the probability estimation suffers from zero probability values when there are not enough training samples. So, a small-sample corrections can be added into all probabilities to prevent zero probability values. This technique is know as Laplace correction [11].

$$\begin{aligned}
\pi_c &= \frac{\sum_{i=1}^{l} \phi(y^{(i)} = c) \; + \; 1}{\sum_{k=1}^{C} \sum_{i=1}^{l} \phi(y^{(i)} = k) \; + \; C} \\
\theta_{sc}^j &= \frac{\sum_{i=1}^{l} \phi(x_j^{(i)} = s \wedge y^{(i)} = c) \; + \; 1}{\sum_{m=1}^{S} \sum_{i=1}^{l} \phi(x_j^{(i)} = m \wedge y^{(i)} = c) \; + \; S_j}
\end{aligned}$$

2) Maximum likelihood Estimation for Gaussian Distribution: Suppose x_j are drawn from a Gaussian distrbution, $x_j \sim \mathcal{N}(\mu, \sigma^2)$, with unkown model parameters (mean μ and variance σ^2). The difference between Gaussian log-likelihood and Categorical log-likelihood distrbution is in the $p(x_j^{(i)}|y^{(i)};\theta)$, because the estimating of the class prior for all distribution are same. Then the log-likelihood equation (1) without class prior probabilty for Gaussian distrbution can be written as follows.

$$\begin{aligned}
\log p(\mathcal{D}; \mu, \sigma^2) &= \sum_{i=1}^{l} \sum_{c=1}^{C} \sum_{j=1}^{d} \log \mathcal{N}\left(x_j^{(i)} | y^{(i)}; \mu_{jc}, \sigma_{jc}^2\right) \\
= \sum_{i=1}^{l} \sum_{c=1}^{C} \sum_{j=1}^{d} &\log \left(\frac{1}{(2\pi)^{\frac{1}{2}} |\sigma_{jc}^2|^{\frac{1}{2}}} \exp\big(-\frac{1}{2} \frac{(x_j^{(i)} - \mu_{jc})^2}{(\sigma_{jc}^2)} \big) \right)
\end{aligned} \tag{3}$$

To obtain the maximum likelihood estimate in closed form the partial derivatives can be computed for equation (3), with respect to all the parameters$(\mu_{jc}, \sigma_{jc}^2)$, and then each partial derivative to zero is set to zero.

$$\begin{aligned}
\frac{\partial \Lambda}{\partial \mu_{jc}} &= 0 \Rightarrow \mu_{jc} = \frac{\sum_{i=1}^{l} x_j^{(i)}}{l_c} \\
\frac{\partial \Lambda}{\partial \sigma_{jc}^2} &= 0 \Rightarrow \sigma_{jc}^2 = \frac{\sum_{i=1}^{l} (x_j^{(i)} - \mu_{jc})^2}{l_c}
\end{aligned}$$

where $l_c = \sum_{i=1}^{l} \phi(y^{(i)} = c)$ is number of patterns in class (c).

B. The Semi-supervised Naïve Bayes Classifier

In section II-A, the supervised NB classifier with fully labelled data was described. However, in some cases the training data D consists of both labelled, D_l, and unlabelled, D_u instances, $D = D_l \cup D_u$. Applying NB with both types of data is called semi-supervised learning. Consider that the labelled data $D_l = \left\{(x^{(i)}, y^{(i)})\right\}_{i=1}^{l}$ and unlabelled data is $D_u = \left\{(x^{(i)})\right\}_{i=l+1}^{l+u}$ where $x^{(i)} \in \mathcal{X} \subseteq \mathcal{R}^d$ represents a feature vector describing the ith example and it corresponding class label $y^{(i)} \in \{1, 2, \ldots, C\}$ in the labelled data. Then, the likelihood function is defined as:

$$\begin{aligned}
p(\mathcal{D}; \theta) &= p(\mathcal{D}_l; \theta) \; \times \; p(\mathcal{D}_u; \theta) \\
&= \prod_{i=1}^{l} \left(p(y^{(i)}; \theta) \prod_{j=1}^{d} p(x_j^{(i)} | y^{(i)}; \theta) \right) \\
&\quad \times \prod_{i=l+1}^{l+u} \prod_{j=1}^{d} p(x_j^{(i)}; \theta)
\end{aligned}$$

The likelihood for unlabeled data is the essential difference between supervised and semi-supervised log likelihood. The likelihood for unlabeled data is the marginal probability $p(x_j^{(i)}; \theta)$ as we do not know which classes they belong to. Here, to address this problem we add the latent variable $z^{(i)}$ where $i = l+1, l+2, ..., l+u$ for unlabelled data and try to maximise semi-supervised likelihood.

$$\begin{aligned}
p(\mathcal{D}; \theta) &= \prod_{i=1}^{l} \left(p(y^{(i)} | \theta) \prod_{j=1}^{d} p(x_j^{(i)} | y^{(i)}; \theta) \right) \\
&\times \prod_{i=l+1}^{l+u} \left(\sum_{c=1}^{C} p(z^{(i)} = c; \theta) \; \prod_{j=1}^{d} p(x_j^{(i)} | z^{(i)} = c; \theta) \right)
\end{aligned}$$

Instead of maximising the likelihood $p(\mathcal{D}; \theta)$ we work with log-likelihood $log \; p(\mathcal{D}; \theta)$.

$$\begin{aligned}
\log p(\mathcal{D}; \theta) &= \sum_{i=1}^{l} log \bigg(\; p(y^{(i)}; \theta) \prod_{j=1}^{d} \; p(x_j^{(i)} | y^{(i)}; \theta) \bigg) \\
+ \sum_{i=l+1}^{l+u} &\log \left(\sum_{c=1}^{C} p(z^{(i)} = c; \theta) \; \prod_{j=1}^{d} p(x_j^{(i)} | z^{(i)} = c; \theta) \right)
\end{aligned} \tag{4}$$

When there are latent variables (no labels for unlabelled data) in the training data, it is no longer possible to find a closed form solution for the MLE, because the summation inside the *log* which hard to maximise by setting partial derivatives to zero. Therefore, we use an iterative statistical technique known as the Expectation Maximisation (EM) algorithm. This algorithm overcomes this problem; it can find a local maximum of the likelihood by maximizing a lower bound on the likelihood for unlabeled data instead of maximizing likelihood itself.
The EM algorithm starts with an estimate for the initial vector of parameters, using the labelled data only, via the standard

NB, and then iterates over the following two steps until it converges to a stable solution and set of labels for the data. The EM algorithm first estimates the expectations of the missing labels (latent variables) for the unlabelled instances in the E-step $q_{ic} = p(z^{(i)} = c|x_j^{(i)}; \theta)$ where $i = (l+1, ..., l+u)$ and $0 \leq q_{ic} \leq 1$ and assigns probabilistic labels to the unlabelled data.

$$q_{ic} = \frac{p(z^{(i)} = c; \theta) \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta)}{\sum_{k=1}^{C} p(z^{(i)} = k; \theta) \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = k; \theta)}$$

Note that for the labeled data we already know to which class each pattern belongs then $q_{ic} = 1$ if $y^{(i)} = c$ and $q_{ic} = 0$ otherwise. In addition, the q_{ic} satisfy the summation constraint $\sum_{c=1}^{C} q_{ic} = 1$.

In order to obtain the lower bound for unlabeled data we multiply and divide $\log p(\mathcal{D}_u; \theta)$ by q_{ic}.

$$\begin{aligned}
&\log p(\mathcal{D}_u; \theta) = \\
&\sum_{i=l+1}^{l+u} \log \left(p(z^{(i)} = c; \theta) \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta) \right) \frac{q_{ic}}{q_{ic}} \\
= &\sum_{i=l+1}^{l+u} \log \sum_{c=1}^{C} q_{ic} \left(\frac{p(z^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta)}{q_{ic}} \right) \\
= &\sum_{i=l+1}^{l+u} \log E_{q_{ic}} \left\{ \frac{p(z^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta)}{q_{ic}} \right\}
\end{aligned}$$

The lower bound for unlabeled data is obtained via Jensens inequality [12] $E[\log(X)] \leq \log(E[X])$.

$$\begin{aligned}
&\sum_{i=l+1}^{l+u} \log E_{q_{ic}} \left\{ \frac{p(z^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta)}{q_{ic}} \right\} \\
\geq &\sum_{i=l+1}^{l+u} E_{q_{ic}} \left\{ \log \frac{p(z^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta)}{q_{ic}} \right\}
\end{aligned} \tag{5}$$

we substitute the right hand side for above expression 5 instead the second term in the equation 4 and denote by $\psi(\theta)$

$$\begin{aligned}
\psi(\theta) = &\sum_{i=1}^{l} \log \left(\ p(y^{(i)}; \theta) \prod_{j=1}^{d} \ p(x_j^{(i)}|y^{(i)}; \theta)) \right) \\
+ &\sum_{i=l+1}^{l+u} E_{q_{ic}} \left\{ \log \frac{p(z^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta)}{q_{ic}} \right\} \\
= &\sum_{i=1}^{l} q_{ic} \log \left(\ p(y^{(i)}; \theta) \prod_{j=1}^{d} \ p(x_j^{(i)}|y^{(i)}; \theta) \right) \\
+ &\sum_{i=l+1}^{l+u} \sum_{c=1}^{C} q_{ic} \log \left(\frac{p(z^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta)}{q_{ic}} \right) \\
= &\sum_{i=1}^{l} q_{ic} \log \left(\ p(y^{(i)}; \theta) \prod_{j=1}^{d} \ p(x_j^{(i)}|y^{(i)}; \theta) \right) \\
+ &\sum_{i=l+1}^{l+u} \sum_{c=1}^{C} q_{ic} \log \left(p(z^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|z^{(i)} = c; \theta) \right) \\
- &\sum_{i=l+1}^{l+u} \sum_{c=1}^{C} q_{ic} \log q_{ic} \\
= &\sum_{i=1}^{l+u} \sum_{c=1}^{C} q_{ic} \log \left(p(y^{(i)} = c; \theta) \ \prod_{j=1}^{d} p(x_j^{(i)}|y^{(i)} = c; \theta) \right) \\
- &\sum_{i=l+1}^{l+u} \sum_{c=1}^{C} q_{ic} \log q_{ic}
\end{aligned} \tag{6}$$

where

$$y^{(i)} = \begin{cases} y^{(i)} & : i = 1, ..., l \\ z^{(i)} & : i = l+1, ..., l+u \end{cases}$$

The M-step estimates the new model parameters by partial derivatives for equation 6 using all of the labelled and unlabelled data, and treats the expected values of the latent variable that calculated in the E-step as the true class labels for the unlabelled data. We can show how estimate the new model parameters as follows.

1) Maximum likelihood Estimation for Categorical Distribution: If $x_j \sim cat(\theta)$ then equation (6) can be written in terms of the parameters with Lagrangian term.

$$\begin{aligned}
\Lambda(\pi, \theta, \alpha, \beta) = &\sum_{i=1}^{l+u} \sum_{c=1}^{C} q_{ic} \log \pi_c \\
+ &\sum_{i=1}^{l+u} \sum_{j=1}^{d} \sum_{s=1}^{S} \sum_{c=1}^{C} q_{ic} \phi(x_j^{(i)} = s) \log \theta_{sc}^{j} \\
- &\sum_{i=l+1}^{l+u} \sum_{c=1}^{C} q_{ic} \log q_{ic} - \alpha \left(\sum_{c=1}^{C} \pi_c - 1 \right) \\
- &\sum_{c=1}^{C} \sum_{j=1}^{d} \beta_c^j \left(\sum_{s=1}^{S} \theta_{sc}^{j} - 1 \right)
\end{aligned} \tag{7}$$

To obtain the maximum likelihood estimate the partial derivatives can be computed for equation (7) with respect to all the parameters $(\pi_c, \alpha, \beta_c^j)$ and set to zero. For α same as supervised NB.

$$\begin{aligned}
\frac{\partial \Lambda}{\partial \pi_c} &= 0 \Rightarrow \\
\pi_c &= \frac{\sum_{i=1}^{l+u} q_{ic}}{\sum_{k=1}^{C} \sum_{i=1}^{l+u} q_{ik}} \\
\frac{\partial \Lambda}{\partial \theta_{sc}^{j}} &= 0 \Rightarrow \\
\theta_{sc}^{j} &= \frac{\sum_{i=1}^{l+u} q_{ic} \phi(x_j^{(i)} = s)}{\sum_{m=1}^{S} \sum_{i=1}^{l+u} q_{ic} \phi(x_j^{(i)} = m)}
\end{aligned}$$

Where the summation in the denominator is over all possible values (states) m for each attribute x_j. The Laplace correction for the parameters, (θ^j_{sc}, π_c), are shown as follows:

$$\begin{aligned} \pi_c &= \frac{\sum_{i=1}^{l+u} q_{ic} \ + \ 1}{\sum_{k=1}^{C} \sum_{i=1}^{l+u} q_{ik} \ + \ C} \\ \theta^j_{sc} &= \frac{\sum_{i=1}^{l+u} q_{ic} \phi(x_j^{(i)} = s) \ + \ 1}{\sum_{m=1}^{S} \sum_{i=1}^{l+u} q_{ic} \phi(x_j^{(i)} = m) \ + \ S_j} \end{aligned}$$

2) Maximum likelihood Estimation for Gaussian Distribution: Then the log-likelihood equation (6) without class prior probability for Gaussian distribution in SSNB can be written as follows if $\mathcal{X} \sim \mathcal{N}(\mu, \sigma^2)$, because the only difference with Categorical log-likelihood in $p(x_j^{(i)}|y^{(i)}; \theta)$.

$$\begin{aligned} \log p(\mathcal{D}; \mu, \sigma^2) &= \sum_{i=l+1}^{l+u} \sum_{c=1}^{C} \sum_{j=1}^{d} q_{ic} \log \Bigg(\frac{1}{(2\pi)^{\frac{1}{2}} |\sigma^2_{jc}|^{\frac{1}{2}}} \exp \\ &\quad \Big(-\frac{1}{2} (x_j^{(i)} - \mu_{jc})^2 (\sigma^2_{jc})^{-1} \Big) \Bigg) \\ &\quad - \sum_{i=l+1}^{l+u} \sum_{c=1}^{C} q_{ic} \log q_{ic} \end{aligned} \tag{8}$$

The closed form maximum likelihood estimate can be obtained by computing the partial derivatives for equation (8) with respect to all the parameters $(\mu_{jc}, \sigma^2_{jc})$, and then setting each partial derivative to zero.

$$\begin{aligned} \frac{\partial \Lambda}{\partial \mu_{jc}} &= 0 \Rightarrow \mu_{jc} = \frac{\sum_{i=1}^{l+u} q_{ic} \ x_j^{(i)}}{\sum_{i=1}^{l+u} q_{ic}} \\ \frac{\partial \Lambda}{\partial \sigma^2_{jc}} &= 0 \Rightarrow \sigma^2_{jc} = \frac{\sum_{i=1}^{l+u} q_{ic} (x_j^{(i)} - \mu_{jc})^2}{\sum_{i=1}^{l+u} q_{ic}} \end{aligned}$$

III. EXPERIMENTS

A. UCI benchmark datasets

1) Datasets and experimental design: In order to evaluate the performance of NB compared to the SSNB classifier, we performed two sets of experiments for discrete and continuous attributes respectively. The first experiment used 30 discrete data sets, the second experiment used the 20 continuous benchmark datasets. All datasets were taken from the UCI machine-learning repository [13], and the SGI[1] repository. Across both of the experiments, the following steps were taken for all datasets at the pre-processing stage: The categorical and ordinal variables were encoded using discrete values from 1-to-n. For each feature, whether discrete or continuous, the instances where any attribute value is missing are discarded.

All experiments consisted of 100 trials, with random partitioning of the datasets into training and test sets in each trial. For each dataset, 75% was used for training and 25% was held-out as a test set, used only to evaluate the classification error rate during the experiments. The number of labelled instances is gradually increased up to the number of training sample on a logarithmic scale, and the remaining training data were used as unlabelled data. This procedure was repeated until all the training instances were used as labelled data. During the training stage, at least two training patterns were selected from each class for Gaussian NB in order to avoid having zero variance. If any attribute has zero variance, it is omitted from the analysis.

The learning curve provides from the error rate value for all the examples. The error rate was calculated as the mean error rate measured over the 100 replications and the corresponding standard error. The area under learning curve error rate (AULC) plot was computed in each replication to evaluate the error rate performance. The ranking score was obtained by normalising the AULC, to give the global score.

$$globalscore \ = \ \frac{AULC \ - \ min(A)}{max(A) \ - \ min(A)} \tag{9}$$

where $max(A)$ is the maximum possible area under curve and $min(A)$ is the minimum possible area under curve that always zero. The idea of computing global score is similar [14], but the only difference is that they found a global score for the Area Under ROC curve (AUC) while we found a global score for error rate learning curve. The average global score over 100 error rate learning curves was calculated in order to compare the prediction performance between classifiers for each dataset. The Wilcoxon signed rank test [15] was used to determine the statistical significance of the difference between the SSNB and the NB over multiple datasets in terms of the global score.

2) Results for UCI benchmark datasets: Our first experiments found that use of the unlabeled dataset does not generally reduce the classification error rate. Table I shows the results for 36 discrete benchmark datasets. NB was best on 25 out of 36 benchmark datasets, the SSNB best on only 10. The result for the Wilcoxon signed rank test shows that the NB is statistically superior at the 95% level of significance.

From table II it can be seen that the global score for AULC of the SSNB is statistically better than the NB only for the (`iris`, `new-thyroid`, and, `wine`) continuous datasets. However, the global score for AULC for NB was best on most of the datasets. There is statistical significant difference according to Wilcoxon signed rank test at the 95% level of significance over all datasets.

In these experiments, we concluded that the performance of the SSNB was inferior to that the NB for both discrete and continuous input attributes.

B. Why is the naïve Bayes classifier significantly better on average than the semi-supervised naïve Bayes classifier?

The most obvious explanation for this result is that NB is unable to utilise the unlabelled data correctly. The key characteristic of NB is that it makes the assumption of independence between attributes. This assumption is usually false and NB often produces inaccurate probability estimates, but fairly good classifications. EM relies on the probability estimates, so may be over compensating. To test this hypothesis, we generate simulated data that satisfies the NB assumption. A simple synthetic dataset is generated from two classes with univariate Gaussian distributions when an infinite amount of labelled and unlabelled data is available for training and testing. The model parameters mean and variance for the two Gaussian is (μ_1=-1, μ_2=+1, σ_1 = σ_2 =1) respectively as shown in figure 1.

[1] https://www.sgi.com/tech/mlc/db/

TABLE I. GLOBAL SCORE FOR AULC FOR THE NB AND SSNB OVER 36 DISCRETE DATASETS FROM UCI REPOSITORY. THE RESULTS FOR EACH AULC CLASSIFIER ARE PRESENTED IN THE FORM OF THE MEAN AND STANDARD ERROR OVER TEST DATA FOR 100 REALISATIONS OF EACH DATASET. THE BOLDFACE FONT INDICATES THAT THE GLOBAL SCORE FOR ONE OF THE CLASSIFIERS IS BETTER THAN THE OTHER CLASSIFIER.

#	Dataset	NB	SSNB
1	audiology	**0.480±0.0050**	0.534± 0.0052
2	balance-scale	**0.160±0.0022**	0.167± 0.0023
3	blogger	**0.305±0.0060**	0.337±0.0057
4	breast-cancer	**0.298±0.0033**	0.306± 0.0037
5	breastw	0.033±0.0010	**0.026± 0.0010**
6	car	**0.177±0.0019**	0.211± 0.0021
7	DNA	0.077±0.0008	**0.076± 0.0009**
8	flare1	**0.371±0.0042**	0.382± 0.0042
9	flare2	**0.272±0.0021**	0.279± 0.0022
10	hayes-roth	**0.320±0.0057**	0.373± 0.0055
11	house-votes	**0.106±0.0023**	0.113± 0.0022
12	kr-vs-kp	**0.140±0.0013**	0.273± 0.0016
13	led7	0.271±0.0012	**0.270± 0.0012**
14	led24	0.299±0.0011	**0.292± 0.0012**
15	lung-cancer	0.541±0.0101	**0.512± 0.0101**
16	lymphography	**0.211±0.0050**	0.234 ± 0.0048
17	marketing	**0.686±0.0008**	0.714± 0.0007
18	monk1-corrupt	**0.346±0.0032**	0.347± 0.0032
19	monk1-cross	**0.034±0.0010**	0.046± 0.0015
20	monk1-local	**0.299±0.0033**	0.316± 0.0039
21	monk1	0.287±0.0029	**0.279± 0.0031**
22	monk3-cross	**0.055±0.0017**	0.156± 0.0031
23	monk3-local	**0.054±0.0014**	0.114± 0.0025
24	monk3	**0.052±0.0016**	0.065± 0.0017
25	mushroom	**0.055±0.0006**	0.089± 0.0005
26	nursery	**0.103±0.0006**	0.119± 0.0007
27	primary-tumor	**0.586±0.0034**	0.588± 0.0036
28	promoters	0.168±0.0047	**0.141± 0.0049**
29	shuttle-l&ing-control	0.083±0.0025	**0.079± 0.0025**
30	soybean-small	0.085±0.0052	**0.060± 0.0045**
31	soybean-large	**0.554±0.0024**	0.577± 0.0025
32	splice	0.066±0.0007	**0.064± 0.0007**
33	threeOf9	**0.225±0.0026**	0.231± 0.0025
34	titanic	**0.225±0.0016**	0.228± 0.0017
35	xd6	**0.224±0.0026**	0.229± 0.0029
36	zoo	0.134±0.0051	0.134± 0.0053

TABLE II. GLOBAL SCORE FOR AULC FOR THE NB AND SSNB OVER 20 CONTINUOUS DATASETS FROM UCI REPOSITORY. THE RESULTS FOR EACH AULC CLASSIFIER ARE PRESENTED IN THE FORM OF THE MEAN AND STANDARD ERROR OVER TEST DATA FOR 100 REALISATIONS OF EACH DATASET.THE BOLDFACE FONT INDICATES THAT THE GLOBAL SCORE FOR ONE OF THE CLASSIFIERS IS BETTER THAN THE OTHER CLASSIFIER.

#	Dataset	NB	SSNB
1	banknote	**0.156±0.0021**	0.244± 0.0020
2	Blood-transfusion	**0.252±0.0025**	0.290±0.0035
3	breast-cancerw-continuous	0.040±0.0013	0.040±0.0014
4	Climate-Mode-Simulation-Crashes	**0.067±0.0017**	0.070±0.0017
5	glass	**0.483±0.0049**	0.554±0.0041
6	haberman	**0.259±0.0040**	0.279±0.0052
7	ionosphere	**0.180±0.0044**	0.237±0.0042
8	iris	0.063±0.0032	**0.061±0.0031**
9	letter	**0.370±0.0006**	0.508±0.0006
10	liver-disorder	**0.442±0.0040**	0.490±0.0043
11	magic04	**0.274±0.0006**	0.334±0.0006
12	musk1	**0.283±0.0041**	0.395±0.0035
13	new-thyroid	0.044±0.0022	**0.034±0.0022**
14	pendigits	**0.155±0.0006**	0.180±0.0008
15	sleep	**0.341±0.0003**	0.438± 0.0004
16	vehicle	**0.539±0.0027**	0.600±0.0022
17	vowel	**0.396±0.0031**	0.488±0.0026
18	waveform-noise	**0.202±0.0011**	0.226±0.0010
19	waveform	**0.232±0.0010**	0.265±0.0011
20	wine	0.063±0.0023	**0.041±0.0024**

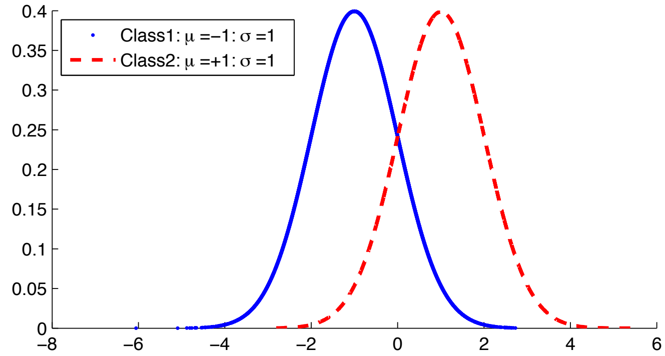

Fig. 1. Two class classification problem for the synthetic dataset

This experiment consisted of 10,000 trials of random partitioning of the datasets (67584 patterns) into training and test sets, that 2048 patterns were used for training and 65536 patterns were held-out as a test set used to evaluate the classification error rate performance during the experiments. The experimental design exactly same as section III-A1. Figure 2 shows the error rate learning curve for both the semi-supervised Gaussian classifier and the Gaussian classifier. It is clearly seen that the semi-supervised Gaussian classifier performs better than the Gaussian classifier, especially when very few labeled data were used for training, and rapidly converges in the number of labelled samples.

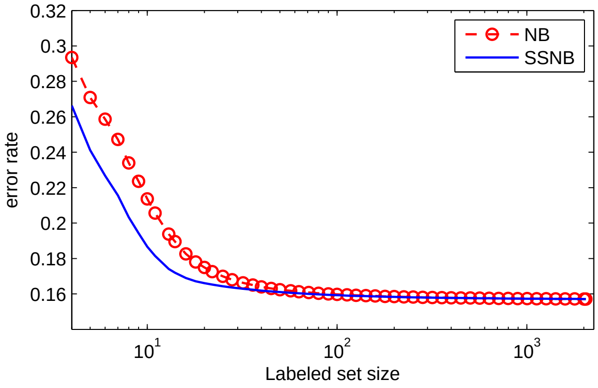

Fig. 2. The learning curve for two class classification problem in Gaussian distribution for synthetic dataset

C. Synthetic benchmark dataset

1) Generating synthetic dataset and experimental design: The results from the previous section III-B indicate that the violation of the independence assumption of NB mode, might be a reason for the lack of increased performance for the SSNB. It is possible that the SSNB is sensitive to the correctness of the model's assumptions. To investigate this further, we generate simulated data from the UCI sets that will satisfy the independence assumption. To do this, we first fit a NB model to each data set, then use the model estimates of the attribute distributions to generate simulated data with independent features. The synthetic datasets are similar in character to the original datasets but the model's assumption of the independence of attributes is valid.

TABLE III. GLOBAL SCORE FOR AULC FOR THE NB AND SSNB OVER 36 SYNTHETIC DATASETS FROM UCI. THE RESULTS FOR EACH AULC CLASSIFIER ARE PRESENTED IN THE FORM OF THE MEAN AND STANDARD ERROR OVER TEST DATA FOR 100 REALISATIONS OF EACH DATASET.THE BOLDFACE FONT INDICATES THAT THE GLOBAL SCORE FOR ONE OF THE CLASSIFIERS IS BETTER THAN THE OTHER CLASSIFIER.

#	Dataset	NB	SSNB
1	audiology	**0.436±0.0055**	0.437±0.0049
2	balance-scale	0.270±0.0026	**0.268±0.0028**
3	blogger	0.287±0.0062	**0.284±0.0058**
4	breast-cancer	**0.267±0.0037**	0.269±0.0037
5	breastw	0.004±0.0002	**0.000±0.0000**
6	car	0.247±0.0016	**0.244±0.0017**
7	DNA	0.033±0.0005	**0.022±0.0005**
8	flare1	0.298±0.0040	**0.283±0.0039**
9	flare2	0.201±0.0020	**0.198±0.0020**
10	hayes-roth	0.329±0.0054	**0.314±0.0051**
11	house-votes	0.001±0.0001	**0.000±0.0000**
12	kr-vs-kp	0.175±0.0013	**0.168±0.0013**
13	led7	0.271±0.0015	**0.269±0.0015**
14	led24	0.288±0.0012	**0.277±0.0012**
15	lung-cancer	0.187±0.0085	**0.167±0.0090**
16	lymphography	0.104±0.0032	**0.066±0.0030**
17	marketing	0.589±0.0007	**0.586±0.0007**
18	monk1-corrupt	0.287±0.0029	**0.280±0.0031**
19	monk1-cross	0.107± 0.0021	**0.086±0.0021**
20	monk1-local	0.318±0.0031	**0.310±0.0035**
21	monk1	0.320±0.0030	0.320±0.0031
22	monk3-cross	0.090±0.0022	**0.076±0.0023**
23	monk3-local	0.122±0.0021	**0.107±0.0020**
24	monk3	0.116±0.0022	**0.106±0.0024**
25	mushroom	0.001±0.0001	0.001±0.0001
26	nursery	0.156±0.0006	**0.15 ±0.0006**
27	primary-tumor	0.560±0.0037	0.560±0.0038
28	promoters	0.063±0.0024	**0.009±0.0015**
29	shuttle-landing-control	0.101±0.0028	**0.093±0.0029**
30	soybean-small	0.067 ±0.0046	**0.049±0.0045**
31	soybean-large	**0.451±0.0027**	0.460±0.0027
32	splice	0.048±0.0005	**0.032 ±0.0006**
33	threeOf9	0.280±0.0032	**0.270±0.0033**
34	titanic	0.227±0.0015	**0.226±0.0015**
35	xd6	0.229±0.0029	**0.219±0.0029**
36	zoo	0.087±0.0042	**0.081±0.0044**

TABLE IV. GLOBAL SCORE FOR AULC FOR THE NB AND SSNB OVER 20 SYNTHETIC DATASETS FROM UCI. THE RESULTS FOR EACH AULC CLASSIFIER ARE PRESENTED IN THE FORM OF THE MEAN AND STANDARD ERROR OVER TEST DATA FOR 100 REALISATIONS OF EACH DATASET. THE BOLDFACE FONT INDICATES THAT THE GLOBAL SCORE FOR ONE OF THE CLASSIFIERS IS BETTER THAN THE OTHER CLASSIFIER.

#	Dataset	NB	SSNB
1	banknote	0.201 ±0.0016	**0.197± 0.0018**
2	Blood-transfusion	0.093 ±0.0017	**0.087±0.0018**
3	breast-cancerw-continuous	0.002 ±0.0002	**0.000±0.0001**
4	Climate-Mode-Simulation-Crashes	0.008 ±0.0006	**0.004±0.0004**
5	glass	0.126±0.0033	**0.105±0.0033**
6	haberman	0.184±0.0041	**0.175±0.0042**
7	ionosphere	0.008±0.0005	**0.002±0.0003**
8	iris	0.004±0.0008	**0.001±0.0005**
9	letter	0.236±0.0005	**0.223±0.0006**
10	liver-disorder	0.217±0.0033	**0.198±0.0035**
11	magic04	0.006±0.0001	0.006±0.0001
12	musk1	0.014±0.0008	**0.007±0.0006**
13	new-thyroid	0.006±0.0006	**0.001±0.0005**
14	pendigits	0.065±0.0003	**0.059±0.0003**
15	sleep	0.128±0.0002	**0.127±0.0002**
16	vehicle	0.247±0.0020	**0.241±0.0020**
17	vowel	0.108±0.0017	**0.071±0.0017**
18	waveform-noise	0.020±0.0003	**0.017±0.0003**
19	waveform	0.052±0.0005	**0.049±0.0005**
20	wine	0.010±0.0008	**0.003±0.0006**

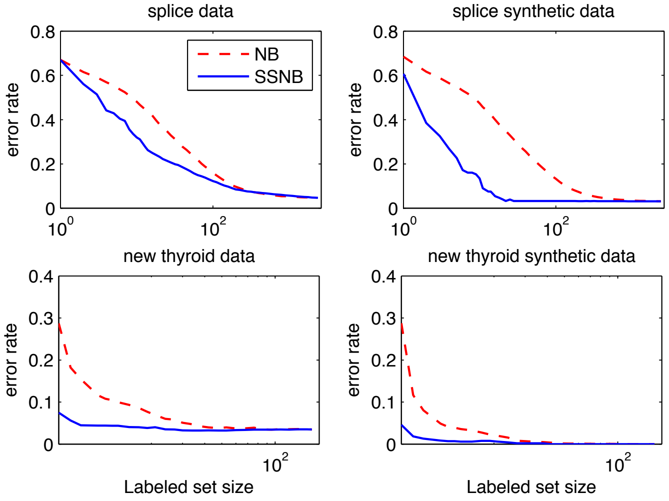

Fig. 3. The average learning curve for NB and SSNB of the UCI and synthetic (splice, newthyroid) datasets

2) Result for synthetic benchmark datasets: Table III shows that the SSNB performs well compared to the NB for the 30 synthetic benchmark datasets. The SSNB also was best on 19 of the continuous synthetic datasets, shown in Table IV,thus the SSNB better than the NB in the current experiment.There is a statistical significant difference between the average rank of the SSNB and NB global scores for AULC according to the Wilcoxon signed rank test at the 95% level of confidence over multiple synthetic benchmark datasets. This suggest that SSNB is sensitive to conformance to its assumption of the independence between attributes.

D. Exploratory Data Analysis

The results for UCI benchmark datasets experiments suggest that a few datasets are always likely to have better performance for SSNB, such as (`breastw`, `DNA`, `led7`, `led24`, `lung-cancer`, `promoters`, `shuttle-landing-control`, `splice`, `soybean-small`) in discrete and (`iris`, `new-thyroid`, `wine`) in continuous benchmark datasets. The details of this performance can be seen by examing the learning curve. We can show learning curve only for a few datasets due to space limitation. Figure 3 shows the learning curve for one of the discrete datasets `splice` and `new-thyroid` which is the continuous datasets.

Interestingly, most of the benchmark datasets show improved classification performance for the SSNB in the synthetic datasets. Figure 4 shows the learning curve for `nursery` and `waveform` datasets which are discrete and continuous respectively. However, learning curve for (`audiology`, `breast-cancer`, `monk1`, `mushroom`, `primary-tumor`, `soybean-large`) in discrete datasets and (`magic04`) in continuous datasets show that SSNB does not help in all experiments, as we can see learning curve for one of them in figure 5.

The learning curve results across all experiments shows that if the model assumption is correct the unlabeled data might help to improving performance, especially when a few lableled data used as a training set, but if the model assumption is violated, the classififcation performance could degrade as adding more unlableled to the training set.

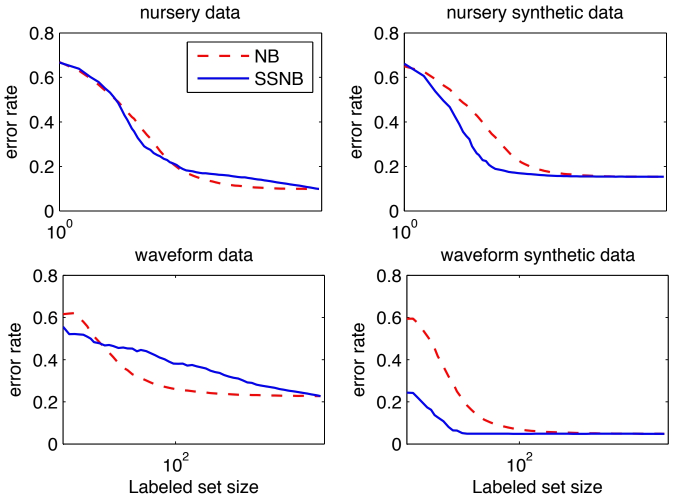

Fig. 4. The average learning curve for the NB and SSNB of the UCI and synthetic (nursery, waveform) datasets

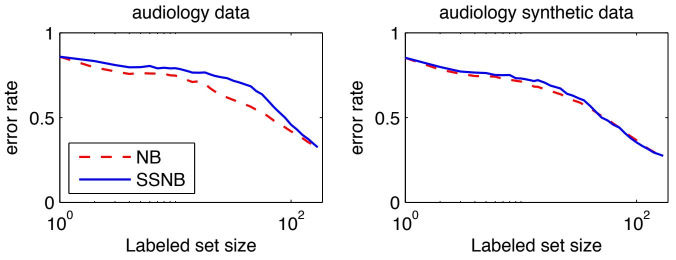

Fig. 5. The average learning curve for the NB and SSNB of the UCI and synthetic audiology datasets

IV. Conclusion

The contribution of this paper is an empirical evaluation of NB and SSNB on binary and multi-class classification problems with continuous and discrete attributes. We wish to address the question of whether using unlabelled data will improve classification accuracy. This will clearly be dictated by our choice of classifier and semi-supervised learning scheme. We evaluate a naïve Bayes classifier used in conjunction with an Expectation-Maximization algorithm that iteratively uses NB to predict the unlabelled instances. We found that using the unlabelled data made the classifier significantly less accurate. To understand why this may be so, we assessed the performance of NB and SSNB on synthetic data for which the NB assumption of independent attributes is true. We found that SSNB was significantly more accurate on these data. We conclude that if a classifier is not suitable for a data set, then using unlabelled data in a self training scheme is likely to make it worse. This implies that effort should be applied in finding a classifier suitable for a problem before using unlabelled data to self train.

References

[1] X. Zhu, “Semi-supervised learning literature survey,” 2006.

[2] K. Nigam, A. McCallum, and T. Mitchell, “Semi-supervised text classification using EM,” *Semi-Supervised Learning*, pp. 33–56, 2006.

[3] F. G. Cozman and I. Cohen, “Unlabeled data can degrade classification performance of generative classifiers,” in *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference, May 14-16, 2002, Pensacola Beach, Florida, USA*, 2002, pp. 327–331. [Online]. Available: http://www.aaai.org/Library/FLAIRS/2002/flairs02-065.php

[4] Y. Guo, X. Niu, and H. Zhang, “An extensive empirical study on semi-supervised learning,” in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 186–195.

[5] R. O. Duda and P. E. Hart, “Pattern Recognition and Scene Analysis,” 1973.

[6] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[7] C.-H. Lee, F. Gutierrez, and D. Dou, “Calculating feature weights in naive Bayes with Kullback-Leibler measure,” in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 1146–1151.

[8] L. Jiang, Z. Cai, H. Zhang, and D. Wang, “Not so greedy: Randomly selected naive Bayes,” *Expert Systems with Applications*, vol. 39, no. 12, pp. 11 022–11 028, 2012.

[9] P. Langley, W. Iba, and K. Thompson, “An Analysis of Bayesian Classifiers.” in *Proceedings of the 10th National Conference on Artificial Intelligence. San Jose, CA, July 12-16, 1992.*, 1992, pp. 223–228. [Online]. Available: http://www.aaai.org/Library/AAAI/1992/aaai92-035.php

[10] D. Klein, “Lagrange multipliers without permanent scarring,” *University of California at Berkeley, Computer Science Division*, 2001.

[11] J. Wu and Z. Cai, “A naive Bayes probability estimation model based on self-adaptive differential evolution.” *Journal of Intelligent Information Systems*, vol. 42, no. 3, pp. 671–694, 2014. [Online]. Available: http://dx.doi.org/10.1007/s10844-013-0279-y

[12] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, “Text classification from labeled and unlabeled documents using EM,” *Machine learning*, vol. 39, no. 2-3, pp. 103–134, 2000.

[13] K. Bache and M. Lichman, “UCI Machine Learning Repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[14] I. Guyon, G. Cawley, G. Dror, and V. Lemaire, “Results of the active learning challenge,” *Active Learning Challenge Challenges in Machine Learning, Volume 6*, p. 21.

[15] J. Demšar, “Statistical comparisons of classifiers over multiple data sets,” *The Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.