Self-Adaptive Attribute Weighting for Naive Bayes Classification

Jia Wu^{a,b}, Shirui Pan^b, Xingquan Zhu^c, Zhihua Cai^a, Peng Zhang^b, Chengqi Zhang^b

^aSchool of Computer Science, China University of Geosciences, Wuhan 430074, China.

^bQuantum Computation & Intelligent Systems (QCIS) Centre,
Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Australia.

^cDepartment of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431, USA.

Abstract

Naive Bayes (NB) is a popular machine learning tool for classification, due to its simplicity, high computational efficiency, and good classification accuracy, especially for high dimensional data such as texts. In reality, the pronounced advantage of NB is often challenged by the strong conditional independence assumption between attributes, which may deteriorate the classification performance. Accordingly, numerous efforts have been made to improve NB, by using approaches such as structure extension, attribute selection, attribute weighting, instance weighting, local learning and so on. In this paper, we propose a new Artificial Immune System (AIS) based self-adaptive attribute weighting method for Naive Bayes classification. The proposed method, namely AISWNB, uses immunity theory in artificial immune systems to search optimal attribute weight values, where self-adjusted weight values will alleviate the conditional independence assumption and help calculate the conditional probability in an accurate way. One noticeable advantage of AISWNB is that the unique immune system based evolutionary computation process, including initialization, clone, section, and mutation, ensures that AISWNB can adjust itself to the data without explicit specification of functional or distributional forms of the underlying model. As a result, AISWNB can obtain good attribute weight values during the learning process. Experiments and comparisons on 36 machine learning benchmark data sets and six image classification data sets demonstrate that AISWNB significantly outperforms its peers in classification accuracy, class probability estimation, and class ranking performance.

Keywords: Naive Bayes, Self-Adaptive, Attribute Weighting, Artificial Immune Systems, Evolutionary Computing

1. Introduction

Naive Bayes (NB) (Friedman et al., 1997), a special Bayesian network, is a Bayes' theorem oriented learning model particularly useful for learning tasks involving high dimensional data (Hernández-González et al., 2013), such as text classification (Kim et al., 2006; Chen et al., 2009) and web mining (Zhang et al., 2009). In general Bayesian models, the classification is derived by using the dependency (or conditional dependency) between random variables. This process is typically time consuming because examining relationships among all random variables is a combinatorial optimization task. Alternatively, NB relaxes the restriction of the dependency structures between attributes by simply assuming that attributes are conditionally independent, given the class label. As a result, examining relationships between attributes is no longer needed and the derivation of an NB model can linearly scale to the training data.

In reality, attributes in many learning tasks are correlated to each other, so NB's conditional independence assumption may impair its classification performance (Webb et al., 2012).

Email addresses: jia.wu@student.uts.edu.au (Jia Wu), shirui.pan@student.uts.edu.au (Shirui Pan), xzhu3@fau.edu (Xingquan Zhu), zhcai@cug.edu.cn (Zhihua Cai), peng.zhang@uts.edu.au (Peng Zhang), chengqi.zhang@uts.edu.au (Chengqi Zhang)

In order to relax the conditional independence assumption and simultaneously retain NB's efficiency, many approaches have been proposed by using solutions in five main categories: (1) structure extension (Liu et al., 2011; Jiang et al., 2012a); (2) attribute weighting (Zaidi et al., 2013; Wu et al., 2013a); (3) attribute selection; (4) instance weighting; and (5) instance selection. In this paper, we propose to use attribute weighting to mitigate NB's primary weakness (the conditional independence assumption of the attributes) by assigning a weight value to each individual attribute. Because weight values enforce attributes to play different roles in classification, the corresponding Weighted Naive Bayes (WNB) will help relax the conditional independence assumption and make NB efficient for data (Wu et al., 2007) with strong attribute correlations.

Indeed, the objective of assigning different weight values to attributes share striking similarity as feature selection, where the later intends to discover a subset set of features (with equal importance) to train a classification model. Assume weight values of all attributes are suitably determined, feature selection can be achieved by using a subset of features ranked in a preference order. Therefore, feature weighting can be considered as a generalization of feature selection, and many methods have been using feature selection to help improve NB classification. For example, Langley & Sage (1994) proposed the Selective Bayes Classifier (SBC), by using feature selection to accommodate redundant attributes in the prediction process

and to augment Naive Bayes with the ability to exclude attributes that introduce dependencies. Meanwhile, in order to discover proper weight values for weighted NB classification, researchers have proposed many useful methods to evaluate the importance of attributes, including gain ratio (Zhang & Sheng, 2004), correlation-based algorithm (Hall, 2000), mutual information (Jiang et al., 2009, 2012b), and ReliefF attribute ranking algorithm (Robnik-Šikonja & Kononenko, 2003). Zhang & Sheng (2004) investigated the gain ratio based weighting scheme and several wrapper based methods for finding attribute weights in order to improve the Area Under Curve (AUC), which is a common metric used to compare algorithms by comparing their performance with respect to different parameter settings. Hall (2007) proposed a new attribute weighting method to improve the AUC value, where the weights assigned to the attributes are inversely proportional to the minimum depth at which the attributes are first tested in an unpruned decision tree.

The above methods for weighted Naive Bayes classification have achieved good performance to solve domain specific problems, through the employment of some external criteria, such as gain ratio, to determine suitable weight values for attributes. By doing so, the assessment of the attributes and the derivation of the NB models are separated into two steps, with the attribute weights being determined without taking the NB objective function into consideration. In order to address the problem and seamlessly integrate attribute weighting and NB learning into an integrated process, in this paper we first carry out a systematic experimental analysis for the existing improved algorithms for naive Bayes via attribute weighting (WNB), and then propose a new method to automatically calculate optimal attribute weight values for WNB, by directly working on WNB's objective function. To this end, we employ an evolutionary computation based method, namely Artificial Immune System (AIS) (Er et al., 2012; Cuevas et al., 2012; Haktanirlar Ulutas & Kulturel-Konak, 2012), to assign proper weight values for NB classification. In our previous study (Wu et al., 2013b), we have successfully proposed an AIS based method to automatically and self-adaptively select optimal terms and values for probability estimation. By proposing evolution computation for attribute weighting, our method in this paper will further advance weighted Naive Bayes to ensure that attribute weighting can automatically adapt to different learning tasks.

In order to enable adaptive attribute weighting for NB classification, we will propose to use AIS mechanism to adaptively determine attribute weights, where an automated search strategy is used to find optimal attribute weight values for each data set. The unique immune system computation processes, including initialization, clone, mutation, and selection, ensure that our method can adjust itself to the data without any explicit specification of functional or distributional form of the underlying model. Experiments and comparisons, on 36 UCI machine learning benchmark data sets (Bache & Lichman, 2013) and six image classification data sets (Li & Wang, 2008), demonstrate that the proposed artificial immune systems based weighting scheme for Naive Bayes classification (AISWNB) can successfully find optimal weight combinations for different learn-

ing tasks, and its performance consistently outperforms other state-of-the-art NB algorithms. The corresponding superiority is demonstrated through three major performance metrics, including classification accuracy, class probability estimation, and class ranking performance (Zhang & Su, 2004; Jiang et al., 2009, 2012a).

AISWNB is a self-learning algorithm by utilizing the immunological properties, such as memory property and clonal selection. In contrast to the conventional statistical probabilistic evaluation in NB, the niche and advantages of AISWNB can be understood from the following four aspects:

- AISWNB is a data-driven self-adaptive method because it does not require explicit specification of functional or distributional form of the underlying model.
- AISWNB is a nonlinear model and is flexible in modeling complex real-world relationships.
- AISWNB inherits the memory property of human immune systems and can recognize the same or similar antigen quickly at different times.
- AISWNB can self-adaptively select suitable affinity functions to meet different types of learning tasks.

The remainder of the paper is organized as follows. Section 2 reviews related work. Preliminary concepts and problem statements are addressed in Section 3. Section 4 introduces our new AISWNB framework, followed by the experiments in Section 5. We conclude the paper in Section 6.

2. Related Work

By proposing to use artificial immune system based method to search optimal weight values for weighted naive Bayes classification, our method is related to attribute weighting in machine learning and AIS based evolutionary computation.

2.1. Attribute Weighted Methods

In real-world learning tasks, attributes often play different roles for classification. Therefore, assigning different weight values to attributes can potentially help improve the classification performance. During the whole process, the way of learning the attribute weights plays an essential role. In this subsection, we review existing work on attribute weighting by separating them into two categories: methods which consider each single attribute's correlation with the class, and methods which consider multiple attributes' joint correlations with the class.

2.1.1. Attribute Weighting via Single Attribute Correlation

Mutual Information (MI) between two random variables provides a quantity measure to evaluate the mutual dependence of two variables. A high MI value indicates a large reduction in uncertainty and low MI reflects a small reduction, and a zero MI value between two random variables means that the variables are independent. Friedman et al. (1997) provides a complete definition of mutual information between a pair of variables.

Mutual information has a long history of being used for measuring correlation between attributes and the class variable for classification. For instance, Jiang et al. (2012b) applied mutual information help to improve the accuracy of AODE (Averaged One-Dependence Estimators). Jiang et al. (2009) proposed a Hidden Naive Bayes (HNB) classifier, which uses MI based attribute weighting method to weight one-dependence estimators. Han et al. (2001) proposed a new algorithm called AWKNN (Attribute Weighted K-Nearest-Neighbor). In our experiments, we will apply mutual information to calculate the weight value between each attribute and the class attribute for WNB, and will use this approach, MIWNB, as a baseline for comparisons.

Information Gain (IG), originally used by Quinlan (1993) in the decision tree leaning algorithm, is a commonly used measure to evaluate the correlation of the attribute to the class. A notable drawback of IG is that the resulting score is biased to attributes with a large number of distinct values. Accordingly, information gain ratio (Quinlan, 1993) was proposed to solve the drawback by dividing each attribute's IG score by the information encoded in each attribute itself. Zhang & Sheng (2004) argued that an attribute with a higher gain ratio value deserves a larger weight in WNB. In their studies, they proposed a gain ratio weighted method that calculates the weight of an attribute from a data set.

2.1.2. Attribute Weighting via Multiple Attribute Correlation

Correlation-based Feature Selection (CFS) for attribute weighing uses a correlation-based heuristic evaluation function as an attribute quality measure (Hall, 2000) to calculate the weight value of each attribute. It uses a best-first search to traverse the feature space. CFS starts with an empty set and generates all possible single feature expansions. The subset with the highest evaluation is selected and expanded in the same manner by adding new features. If expanding a subset results in no improvement, the search drops back to the next best unexpanded subset and continues from there. The best subset found is returned after the search terminates.

The core of CFS is the heuristic process that evaluates the worth or "merit" of a feature subset. Hall (2007) employed this method to evaluate the importance of attributes according to the heuristic "merit" value.

Relief is a feature selection method based on attribute estimation (Kira & Rendell, 1992). Relief assigns a grade of relevance to each feature by examining the change of the feature values with respect to instances within the same class (*i.e.*, the nearest hit) and instances between classes (*i.e.*, the nearest miss). If a feature's values remain relatively stable for instances within the same class, the feature will receive a higher weight value. The original Relief only handles binary classification problems. Its extension, Relief-F, can be applied for multi-class classifications (Kononenko, 1994). Besides, Tucker et al. (2010) applied Relief-F attribute weighted approach to deal with top-down product, which is an engineering optimization problem.

Attribute Correlation-based Weighting is a method which explicitly considers the correlation of each attribute to all other attributes to calculate the attribute's weight value (Hall, 2007).

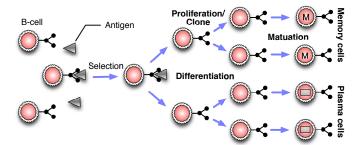


Figure 1: A conceptual view of immune response in immune systems: A B-cell contains the antibody (the middle rings on the left) that allows it to recognize the antigen (triangle), which denotes pathogenic materials invading to the system. The binding between B-cell and antigen can be evaluated by using certain affinity (i.e., degree of binding). In a learning system, this resembles to the assessment of how good a solution (i.e., antibody) recognizes/resolves the training data (i.e., antigen). After the recognition, the system will respond and result in proliferation, differentiation, and maturation process of the B-cell as secondary antibodies. The secondary antibodies with high affinity becomes a memory cell, and others become plasma cells. Memory cells are retrained in the system to allow faster response to the same (or similar) attacks in the future (if the body is re-infected by the same pathogenic materials).

A large weight value will be assigned to the attributes with strong dependencies on other attributes. In order to estimate each attribute's dependence, an unpruned decision tree is constructed from the training instances with a minimum *depth*, which indicates the depth for testing the tree. The weight assigned to each attribute is inversely proportional to the minimum depth at which they are first tested in an unpruned decision tree. Attributes that do not appear in the tree receive zero weight values.

2.2. Artificial Immune Systems

2.2.1. Human Immune System

The human immune system contains two major parts: (1) humoral immunity, which deals with infectious agents in the blood and body tissues, and (2) cell-mediated immunity, which deals with body cells that have been infected. In general, the humoral system is managed by B-cells (with help from T-cells), and the cell-mediated system is managed by T-cells. Each cell (B or T) has a unique type of molecular receptor (location in shape space), which allows for the binding of the antigens (shown as triangles in Fig. 1). A higher affinity between the receptor and antigens indicates a stronger binding.

In immunology, immune system contains two types of lymphocyte cells (B- and T-cells), each of which has a unique type of molecular receptor allowing others to bind to them. When pathogens (*i.e.*, biological agents that may cause diseases or illness) invade the body, antibodies which are produced from B-cells are response for the detection/binding of a foreign protein or antigen (*i.e.*, pathogenic materials). Once the binding between B-cells and antigens are established, B-cells will undergo a series of process including proliferation, differentiation, and maturation, and eventually result in memory cells. The memory cells are retrained in the system to allow faster response to the same (or similar) attacks in the future (if the body is re-infected by the same pathogenic materials). This response process could

be explained by clonal selection theory, and the conceptual review is shown in Fig. 1. In this paper, humoral immunity is delegated to the natural immune system and the action of T-cells is not explained. The clonal selection followed by the B-cells of human immune system is the fundamental mechanism on which Artificial Immune Systems (AIS) is modeled.

2.2.2. AIS: Artificial Immune Systems

Artificial Immune Systems denotes a class of evolutionary computational methods which intend to exploit and simulate the functions and behaviors of mammalian immune systems' learning and memorization capability to solve a learning task. The theme of an AIS is to resemble a biological immune systems' ability to distinguish foreign molecules (or elements) which can attack/damage the body, and provide a learning system with capability of distinction between self *vs.* non-self. This capability eventually leads to the assessment of the fitness scores of candidates with respect to the underlying system.

More specifically, AIS consists of three major components, including representation, recognition, and clone selection when dealing with learning algorithms. The representation, known as shape-shape problem, focuses on how to model antibodies and antigens. When the immune system is attacked by antigen, antibodies try to neutralize the infection by binding to the antigen through the recognition process. Binding strength, also regarded as affinity, is used as a threshold for the immune system to respond to the antigen. The clone selection is corresponding to an affinity maturation process, which means immune individuals with high affinity will gradually increase during clone and mutation process. At the same time, some immune individuals will polarize into memory individuals.

Similar to the AIS, evolutionary algorithms (EAs), such as Genetic Algorithms (GA) (Park & Ryu, 2010), Evolution Strategies (ES) (Huang et al., 2011) and Differential evolution (DE) (Storn & Price, 1997) are all designed based on the basic idea of biological evolution to control, and optimize artificial systems. Evolutionary computation shares many concepts of AIS like a population, genotype phenotype mapping, and proliferation of the most fit. On the other hand, AIS models based on immune networks resemble the structures and interactions of neural network models. The key advantages of AIS over neural networks are the benefits of a population of solutions and the evolutionary selection pressure and mutation. Meanwhile, the underlying mechanisms are fundamentally different in many aspects. First and foremost, the immune system is highly distributed, highly adaptive, self-organising, maintains a memory of past encounters and has the ability to continuously learn about new encounters. AIS is the system developed around the current understanding of the immune system. Second, AIS is a general framework for a distributed adaptive system and could, in principle, be applied to many domains. Compared to most other evolutionary algorithms, AIS is much more simple and straightforward to be implemented, which is important for practitioners from other fields. In addition, because AIS is self-organizing, it requires much less system parameters than other evolutionary computation methods. Some works have also pointed out the similarities and the differences

between AIS and other heuristics (Zheng et al., 2010; Aickelin et al., 2013; Castro & Timmis, 2002).

In recent years, there has been considerable interests in exploring and exploiting the potential of AIS for applications in computer science and engineering including pattern recognition (Yuan et al., 2012), clustering (de Mello Honorio et al., 2012), optimization (Woldemariam & Yen, 2010), and Remote Sensing (Zhong & Zhang, 2012). However, the advantage of AIS for Bayesian classification has received very little attention. In this paper, we propose a new AIS based attribute weighting method for Naive Bayes classification. The performance of this design is validated through numerous performance metrics, including classification accuracy, class probability estimation, and class ranking performance. It is worth noting that some works exist to improve AIS for domain specific problems, such as an improved artificial immune system for seeking the Pareto front of land-use allocation problem in large areas (Huang et al., 2013). However, in this paper, we do not consider the improved AIS for WNB. This is mainly because that we aim at proposing a self-adaptive attribute weighting framework based on the immune system for WNB, and our designs can be easily generalized to any AIS based algorithms.

3. Preliminaries and Problem Definition

Given a training set $\mathcal{D} = \{\mathbf{x}_1, \cdots, \mathbf{x}_N\}$ with N instances, each of which contains n attribute values and a class label, we use $\mathbf{x}_i = \{x_{i,1}, \cdots x_{i,j}, \cdots x_{i,n}, y_i\}$ to denote the ith instance x_i in the data set \mathcal{D} . $x_{i,j}$ denotes the jth attribute value of x_i and y_i denotes the class label of x_i . The class space $\mathcal{Y} = \{c_1, \cdots, c_k, \cdots, c_L\}$ denotes the set of labels that each instance belongs to and c_k denotes the kth label of the class space. For ease of understanding, we use (x_i, y_i) as a shorthand to represent an instance and its class label, and use x_i as a shorthand of \mathbf{x}_i . We also use a_j as a shorthand to represent the jth attribute. For an instance (x_i, y_i) in the training set \mathcal{D} , its class label satisfies $y_i \in \mathcal{Y}$, whereas a test instance x_t only contains attribute values and its class label y_t needs to be predicted by a weighted naive Bayes classification model, which can be formally defined as

$$c(x_t) = \arg\max_{c_k \in \mathcal{Y}} P(c_k) \prod_{j=1}^n P(x_{t,j} | c_k)^{w_j}$$
 (1)

In Eq. (1), $p(c_k)$ represents the prior probability of class c_k in the whole training set. $P(x_{t,j}|c_k)$ denotes the conditional probability distribution of attribute $x_{t,j}$ conditioned by the given class c_k . w_j denotes the weight value of the jth attribute.

In this paper, we focus on the calculation of the conditional probability $p(x_{i,j}|c_k)^{w_j}$ by finding optimal attribute weight values w_j , $j=1,\cdots,n$. While all existing attribute weighting approaches define the weight without considering the uniqueness of the underlying training data, we intend to resolve the optimal w value selection problem as an optimization process. Assume that the calculation of each conditional probability value $p(x_{i,j}|c_k)^{w_j}$ has an optimal w_j value, there are n w_j vectors

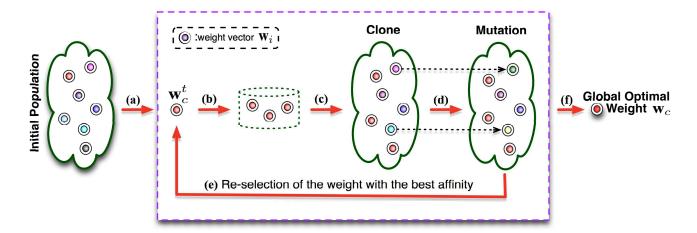


Figure 2: A conceptual view of self-adaptive weighting strategy for AISWNB: An initial population contains many antibodies (*i.e.*, weight vectors \mathbf{w}) that allow themselves to recognize antigens (*i.e.*, training instances \mathcal{D}^a) with certain affinity (*i.e.*, classification performance on \mathcal{D}^b). After recognition, the system will respond and select the weight vector \mathbf{w}_c^t (in the *t*th iteration) with the best affinity (a), and then clone it (b) to replace the weight vectors with low affinity (c). After that, the mutation strategy is adopted to maintain the diversity of the weight vectors (d). The mutation population will further replace the old population (e) to reselect the best weight vector as the memory antibody (a). Through the evolutionary process, the global optimal weigh vector \mathbf{w}_c will be obtained (f).

needed for NB classification. As a result, the WNB classification can be transferred to an optimization problem as follows.

$$\mathbf{w}^* = \underset{w_j \in \mathbf{w}}{\text{max}} f(x_t, \mathbf{w}) \quad s.t. \ 0 \le w_j \le 1$$
 (2)

where $\mathbf{w} = \{w_1, \dots, w_j, \dots, w_n\}$ denotes the attribute weight vector for WNB. And $f(x_t, \mathbf{w})$ is calculated by Eq. (1).

4. Self-Adaptive Attribute Weighted Naive Bayes

4.1. AIS Symbol Definitions and Overall Framework

4.1.1. AIS Symbol Definitions

In this paper, we propose to use AIS to learn optimal attribute weight values for NB classification. In our solution, antigens in AISWNB are simulated as training instances which are presented to the system during the training process. Antibodies represent attribute weight vector \mathbf{w} with different set of values (*i.e.*, candidates). The binding of the antibodies and antigens will resemble the fitness of a specific weight vector with respect to the given training data. This can be evaluated by using the affinity score.

During the learning process, the antibodies with good affinity will experience a form of clonal expansion after being presented with the training data sets (analogous to antigens). When antibodies are cloned they will undergo a mutation process, in which a specific mutation function will be designed (and deployed). The evolving optimization process of the AIS system will help discover optimal ${\bf w}$ vector with the best classification performance.

Before introducing algorithm details, we briefly define following key notations, which will help understand the learning of the weight values using AIS principle. In Table 1, we also summarize the mapping of the symbols between immune systems and AIS based weighting scheme for Naive Bayes.

- Antibodies: W represents the set of antibodies, $W = \{\mathbf{w}_1, \dots, \mathbf{w}_L\}$, where L represents the size of antibodies. $\mathbf{w}_i = \{w_{i,1}, \dots, w_{i,j}, \dots, w_{i,n}\}$ represents a single antibody (i.e., attribute weight vector). So $w_{i,j}$ will represent the jth value of the ith antibody \mathbf{w}_i .
- Antigens: \mathcal{D}^a represents the set of antigens, $\mathcal{D}^a = \{\mathbf{x}_1^a, \cdots, \mathbf{x}_{N_a}^a\}$, where N_a represents the size of antigens. \mathbf{x}_i^a represents a single antigen. In AISWNB, \mathbf{x}_i^a denotes an instance in the data set \mathcal{D}^a .
- Affinity: A measure of closeness between antibodies and antigens. In the current implementation, this value is calculated as accuracy (ACC), the area under the ROC curve (AUC), or conditional log likelihood (CLL) on a given data set, when dealing with classification accuracy, ranking, and probability estimation learning tasks, respectively.
- *Memory Cell:* \mathbf{w}_c represents the memory cell for the antibody which has the best affinity (*i.e.*, best classification performance on the test data sets $\mathcal{D}^b = \{\mathbf{x}_1^b, \dots, \mathbf{x}_{N_b}^b\}$.
- Clone rate: An integer value used to determine the number of mutated clones for a given antibody (weight vector).
 Specifically, a selected antibody is allowed to produce up to mutated clones with clone rate value after responding to a given antigen set.
- Mutation rate: A parameter between 0 and 1 that indicates
 the probability of an antibody being mutated. For a given
 antibody, its mutation rate is equal to 1 minus its affinity.
 By doing so, the antibody with high affinity will have a
 low probability of being mutated.

4.1.2. AISWNB Overall Framework

A conceptual view of the proposed self-adaptive weighting strategy for AISWNB is shown in Fig. 2. In our settings, we

Table 1: Symbol mapping between Immune System and AISWNB.

Immune systems	AISWNB
Antibody	Attribute weight vector w
Antigens	Training instances in \mathcal{D}^a
Shape-space	Possible values of the data vectors
Affinity	The fitness of the weight vector w
	on the testing datasets
Clonal Expansion	Reproduction of weight vectors that
	are well matched with antigens
Affinity Maturation	Specific mutation of w vector and
	removal of lowest stimulated weight vectors.
Immune Memory	Memory set of mutated weight vectors

use antibody to simulate weight vector \mathbf{w} of the naive Bayes models, so an initial population of random antibodies, which correspond to a set of random weight value vectors \mathbf{W} , are selected. The antibodies will recognize the antigens (which correspond to the training instances \mathcal{D}^a) with certain affinity (*i.e.*, classification performance on \mathcal{D}^b). This recognition process resembles to the assessment of evaluating how good the weigh value solutions fit the underlying training data.

After the recognition process, the system will respond and select the weight vector \mathbf{w}_c^t with a good affinity, which corresponds to step (a) in Fig. 2, and then clone some high promising weight vectors to replace some weight vectors with low affinity values. After that, the mutation strategy is carried out to maintain the diversity of the weight vectors, as shown on step (d) of Fig. 2. The mutation population will further replace the old population to reselect the best weight vector as the memory antibody. Through the repetitive evolutionary process, the final optimal weigh vector \mathbf{w}_c will be obtained, as show on step (f) in Fig. 2.

The performance of a classifier is often measured by classification accuracy (ACC). So one can use ACC to calculate the affinity as shown in the above process. In reality, many data mining applications also require the calculation of the class distributions and ranking of the classes, in addition to the classification accuracy (Zhang & Su, 2004; Jiang et al., 2009; Wu & Cai, 2014). In recent years, the area under the ROC curve (AUC) has been used by machine learning and data mining community, and researchers believe (Ling et al., 2003) that AUC is a more discriminant evaluation method than error rate for learning algorithms that also produce class probability estimations. For class probability estimation, conditional log likelihood (CLL) has also been used to evaluate the quality of class probabilities from a classifier (Grossman & Domingos, 2004). Meanwhile, some existing research works (Zhang & Sheng, 2004; Hall, 2007) have proposed to use attribute weighting to improve the AUC performance of NB. However, attribute weighting for Naive Bayes on accuracy performance or class probability estimation has received very little attention. According to the experimental analysis in Section 5.2.1, most existing attribute weighting approaches cannot work well on all the above mentioned three learning tasks. This is mainly because that the attribute subset evaluation used in the traditional

attribute weighted Naive Bayes (*e.g.*, SBC and CFS), intends to maximize the classification accuracy, which may lead to mismatching between the learning process and the other learning goals (Jiang et al., 2012a). In order to address this challenge, the affinity function in AISWNB can be dynamically adjusted to match the learning process and the learning goal. The details related to the affinity function will be addressed in Section 4.2.2.

4.2. AISWNB: AIS based Attribute Weighted Naive Bayes

The proposed AISWNB is achieved through the following two major steps: (1) use AIS algorithm to train models from the training instances, with the purpose of obtaining optimal attribute weight values; and (2) the test instances are classified by the AISWNB classifiers with the learned attribute weight values. Algorithm 1 reports the details of the proposed AISWNB framework, which is described as follows:

4.2.1. Initialization

During the initialization process, we generate a set of L weight candidates: $\mathbf{W} = \{\mathbf{w}_1, \cdots, \mathbf{w}_L\}$, where each individual $\mathbf{w}_i = \{w_{i,1}, \cdots w_{i,j}, \cdots w_{i,n}\}$ represents an antibody (*i.e.* a weight value vector with $w_{i,j}$ representing the weight value for the jth attribute). To generate random weight values for all candidates, we set each $w_{i,j}$ value as a uniformly distributed random variable within range (0, 1]. In our experiments (detailed in Section 5), we use 80% of instances in a given data set \mathcal{D} as the antigens set \mathcal{D}^a to learn optimal weight values \mathbf{w}_c , and the remaining instances are used as the test set \mathcal{D}^b .

4.2.2. AISWNB Evaluation

The AISWNB evaluation process intends to resemble the recognition and the evolution process of the immune systems to find good antibodies (*i.e.*, weight vectors) as shown in Fig. 2. In a weighted NB learning context, the above process corresponds to finding and selecting good weight vectors, and then applying clone and mutation actions to the selected weight vectors to generate new candidates. Some newly generated good candidates will further be retained to train weighted naive Bayes networks.

In the following, we briefly explain the actions in each individual step:

• Calculation of affinity function: For the learning task concerning about the maximization of the classification accuracy (ACC), the affinity of the *i*th individual of the *t*th generation \mathbf{w}_i^t can be obtained by applying the current attribute weight vector \mathbf{w}_i^t to the WNB model, and then evaluate its affinity function as follows,

$$f[\mathbf{w}_i^t] = \frac{1}{N^b} \sum_{i=1}^{N^b} \delta[\mathbf{c}(\mathbf{x}_i^b), y_i^b]$$
 (3)

In Eq. (3), $c(\mathbf{x}_i^b)$ is the classification result of the *i*th instance in a test data set \mathcal{D}^b with N_b instances, by using an AISWNB classifier with attribute weight values \mathbf{w}_i^t . y_i^b is

Algorithm 1 AISWNB (Weighted Naive Bayes by AIS)

Input:

Clone Factor c; Threshold T;

Maximum Generation MaxGen; Antibody Population W;

Antigen Population \mathcal{D}^a ; Test affinity set \mathcal{D}^b ;

Output:

The target class label $c(\mathbf{x}_t)$ of test instance \mathbf{x}_t ;

- W ← The w_{i,j} value of w_i for each individual is initialized using a random number distributed between (0, 1].
- 2: while $t \leq MaxGen$ and $f[\mathbf{w}_c^{t+1}] f[\mathbf{w}_c^t] \leq T$ do
- 3: $f[\mathbf{w}_i^t] \leftarrow \text{Apply antigen population } \mathcal{D}^t$, test affinity set \mathcal{D}^b w.r.t. antibody \mathbf{w}_i^t , and calculate the affinity of \mathbf{w}_i^t .
- w^t_c ← Apply the sequence of each f[w^t_i] to the whole antibody population W^t and find the w^t_c with the best affinity.
- 5: $(W^r)^t \leftarrow$ Select the temporary antibodies set with the lowest affinity with clone factor c.
- 6: $(W^c)^t \leftarrow \text{Clone } \mathbf{w}_c^t \text{ with clone factor } c \text{ and obtain clone antibody set.}$
- 7: $W^t \leftarrow [W^t (W^r)^t] \cup (W^c)^t$;
- 8: **for all** each \mathbf{w}_{i}^{t} in \mathbf{W}^{t} **do**
- 9: $\mathbf{v}_i^{t+1} \leftarrow \text{Apply } \mathbf{w}_c^t$ and a normally distributed random variable N(0,1) to \mathbf{w}_i^t and obtain the mutation individual.
- 10: $\mathbf{w}_i^{t+1} \leftarrow \text{Apply } \mathbf{v}_i^{t+1} \text{ to } \mathbf{w}_i^t \text{ and obtain the new individual in } t + 1 \text{th generation.}$
- 11: end for
- 12: end while
- 13: $c(\mathbf{x}_t) \leftarrow \text{Apply } \mathbf{w}_c$ to instance \mathbf{x}_t to predict the underlying class label .

the actual class label of the *i*th instance. $\delta[c(\mathbf{x}_i^b), y_i^b]$ is one if $c(\mathbf{x}_i^b) = y_i^b$ and zero otherwise.

For learning tasks concerning about other types of learning goals, such as the ranking of the classes or the estimation of class probability distributions, the affinity function can be changed by using the corresponding evaluation criteria, AUC or CLL. So the attribute weight values can be learned in line with the underlying learning goals. The corresponding details are addressed in Section 5.1.3.

- Antibody Selection: We sort the individuals in the initial
 antibody population according to the affinity of each individual, and choose the individual w^t_c with the best affinity
 performance in the tth generation as the memory antibody.
- **Antibody Clone:** To ensure that the population size of every generation is fixed, the best individual \mathbf{w}_c^t will be cloned under the clone factor c. After that, we use the clone set to replace the individuals with low affinity according to the same rate c.
- Antibody Mutation: Using the mutation operation to treat the individuals in tth generation W^t . It means that we obtain the middle generation composed with the new variation individuals from the parent generation. For any individual \mathbf{w}_i^t from the tth generation, the new variation individual \mathbf{v}_i^{t+1} can be generated as follows:

$$\mathbf{v}_{i}^{t+1} = \mathbf{w}_{i}^{t} + F * N(0,1) * (\mathbf{w}_{c}^{t} - \mathbf{w}_{i}^{t})$$
 (4)

Among them, N(0,1) is a normally distributed random variable within the range [0,1]. F, as the variation factor during the process of evolution, can be adaptively obtained according to the different clones (Zhong & Zhang, 2012).

$$F = 1 - f[\mathbf{w}_i^t] \tag{5}$$

where $f[\mathbf{w}_i^t]$ denotes the affinity of the *i*th individual from the *t*th generation.

4.2.3. AISWNB Updating

To determine whether a variation individual \mathbf{v}_i^{t+1} can replace a target individual vector \mathbf{w}_i^t , as a new individual \mathbf{w}_i^{t+1} for the t+1th generation, AISWNB adopts a greedy search strategy. The target individual \mathbf{w}_i^t is replaced by \mathbf{v}_i^{t+1} , if and only if \mathbf{v}_i^{t+1} 's affinity is better than that of \mathbf{w}_i^t . In addition, the system also chooses the individual \mathbf{w}_c^{t+1} with the best affinity performance in the t+1th generation as a new memory antibody.

An unabridged evolutionary process for the population includes **Evaluation** and **Update**, which continuously repeats until (1) the algorithm surpasses the pre-set maximum number MaxGen, or (2) the results obtained from two consecutive iterations are less than the threshold (*i.e.*, T). After obtaining the best individual \mathbf{w}_c (*i.e.*, attribute weight value vector), we use the weight values to build a WNB classifier to classify test data.

4.3. Time Complexity

The time complexity of AISWNB is mainly attributed to the following two processes: (1) evaluation of AISWNB, and (2) updating of the weight values.

Prior to the evaluation of AISWNB model, an NB classifier needs to be trained from \mathcal{D}^a with N_a instances, which will take $O(N_a \cdot n)$, where n is the number of attribute (NB needs to scan the whole training set and build prior probabilities for all classes and conditional probabilities for all n attributes). For the weight population \mathcal{W} in each generation, the calculation of affinity function for each weight individual $\mathbf{w} \in \mathcal{W}$ is similar to testing an NB classifier on a test set \mathcal{D}^b with N_b instances, which will take $O(N_b \cdot n \cdot L)$, where L is the size of the weight populations (*i.e.* the number of weight vectors). The rest four operations (*e.g.*, selection, clone, mutation, and update) are all based on weight vectors. The corresponding time complexity is $O(L \cdot log L)$. Assume the average number of evolution generations is M, the total time complexity U is given by Eq. (6).

$$U = O(N_a \cdot n) + M \times [O(N_b \cdot n \cdot L) + O(L \cdot log L)]$$
 (6)

Because $N_a + N_b = N$, where N is the total number of training data, Eq. (6) can be rewritten as

$$U = O[(N - N_b) \cdot n] + O(N_b \cdot n \cdot L \cdot M) + O(L \cdot logL \cdot M)$$

$$\leq O(N \cdot n) + O(N_b \cdot n \cdot L \cdot M) + O(L \cdot logL \cdot M)$$

$$\leq O(N \cdot n \cdot L \cdot M) + O(L \cdot logL \cdot M)$$

$$\leq O(N \cdot n \cdot L^2 \cdot M)$$
(7)

Eq. (7) shows that the total time complexity of AISWNB is bounded by four important factors: (1) the total number of training samples N; (2) the number of the attribute n; (3) the size of

weight pollution L; and (4) the average number of evolution generations M. In our experiments, we use a threshold T to automatically determine the termination process by following the principe that if the results obtained from two consecutive iterations are less than T, the algorithm will terminate. This process will further reduce the number of iterations and save computational costs.

5. Experiments

5.1. Experimental Settings

5.1.1. Benchmark Data and Parameters

We implement the proposed method using WEKA (Witten & Frank, 2005) data mining tool and validate its performance on 36 benchmark data sets from UCI data repository (Bache & Lichman, 2013) and six image classification data sets from Corel Image repository (Li & Wang, 2008). Because Bayesian classifiers are designed for categorical attributes, in our experiments, we first replace all missing attribute values using the unsupervised attribute filter *ReplaceMissingValues* in WEKA. Then, we apply unsupervised filter *Discretize* in WEKA to discretize numeric attributes into nominal attributes. The similar data preprocessing could also be found in previous works (Jiang et al., 2009, 2012a).

The three parameters L, M and T in Algorithm 1 are set to 50, 50, and 0.001, respectively. All results are obtained via 10 runs of 10-fold cross validation, and our algorithm is carried out on the same training data sets and evaluated on the same testing data. Moreover, all experiments are conducted on a Linux cluster node with an Interl(R) Xeon(R) @3.33GHZ CPU and 3GB fixed memory size.

5.1.2. Baseline Methods

For comparison purposes, we compare AISWNB with the following baseline methods:

- NB: A standard Naive Bayes classifier with conditional attribute independence assumption (Friedman et al., 1997);
- **CFSWNB**: An attribute weighted Naive Bayes based on correlation-based feature selection (Hall, 2000);
- **GRWNB**: An attribute weighted Naive Bayes using gain ratio based feature selection (Zhang & Sheng, 2004);
- MIWNB: An attribute weighted Naive Bayes using mutual information weighted method for feature selection (Jiang et al., 2012b);
- ReFWNB: An attribute weighted Naive Bayes using a feature selection method based on attribute estimation (Robnik-Šikonja & Kononenko, 2003);
- **TreeWNB**: An attribute weighted Naive Bayes with the weighting method according to the degree to which they depend on the values of other attributes (Hall, 2007);
- SBC: A bagged decision-tree based attribute selection filter for Naive Bayes (Langley & Sage, 1994);

Table 2: Detailed information of the 36 UCI benchmark data sets

Data set	Instances	Attributes	Classes	Missing	Numeric
anneal	898	39	6	Y	Y
anneal.ORIG	898	39	6	Y	Y
audiology	226	70	24	Y	N
autos	205	26	7	Y	Y
balance-scale	625	5	3	N	Y
breast-cancer	286	10	3 2 2 2 2 2 2 2	Y	N
breast-w	699	10	2	Y	N
colic	368	23	2	Y	Y
colic.ORIG	368	28	2	Y	Y
credit-a	690	16	2	Y	Y
credit-g	1000	21	2	N	Y
diabetes	768	9	2	N	Y
Glass	214	10	7	N	Y
heart-c	303	14	5	Y	Y
heart-h	294	14	5	Y	Y
heart-statlog	270	14	2 2	N	Y
hepatitis	155	20		Y	Y
hypothyroid	3772	30	4	Y	Y
ionosphere	351	35	2	N	Y
iris	150	5	3	N	Y
kr-vs-kp	3196	37	2	N	N
labor	57	17	2	Y	Y
letter	20000	17	26	N	Y
lymph	148	19	4	N	Y
mushroom	8124	23	2	Y	N
primary-tumor	339	18	21	Y	N
segment	2310	20	7	N	Y
sick	3772	30	2	Y	Y
sonar	208	61	2	N	Y
soybean	683	36	19	Y	N
splice	3190	62	3	N	N
vehicle	846	19	4	N	Y
vote	435	17	2	Y	N
vowel	990	14	11	N	Y
waveform-5000	5000	41	3	N	Y
Z00	101	18	7	N	Y

• **RMWNB**: An attribute weighted Naive Bayes with the attribute weights randomly selected from (0, 1];

5.1.3. Evaluation Criterion

In our experiments, the selected algorithms are evaluated using three performance metrics, including classification accuracy (measured by ACC), class ranking performance (measured by AUC), and class probability estimation (measured by CLL). The ACC of each method is calculated by the percentage of correctly predicted samples in the test set.

In some data mining applications, learning a classifier with accurate class ranking or class probability distributions is also desirable (Zhang & Su, 2004). For example, in direct marketing, the limitation of the resources only allows promotion of the top x% customers during gradual roll-out, or different promotion strategies are deployed for customers with different likelihood of buying certain products. To accomplish these learning tasks, ranking customers according to their likelihood of buying is more useful than simply classifying customers as: buyer or non-buyer (Jiang et al., 2012a). To evaluate the classifier performance in terms of class ranking and class probability distributions, we use AUC and CLL, where AUC of the classifier is calculated as follows:

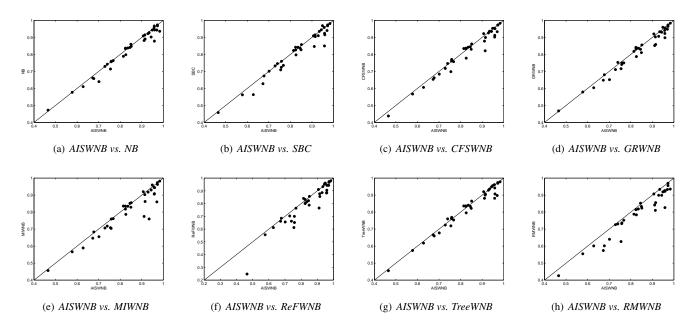


Figure 3: AISWNB vs. competing algorithms: classification accuracy (ACC).

Table 3: Detailed experimental results on classification accuracy (ACC) and standard deviation %.

Data set	AISWNB	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB	RMWNB
anneal	96.99±2.02	94.32±2.23 •	94.03±2.37 •	93.35±2.19 •	94.78±2.29 •	86.01±3.97 •	94.28±1.99 •	89.73±1.94 •	93.31±2.74 •
anneal.ORIG	91.08±2.92	88.16±3.06 •	84.66±3.74 •	87.81±2.55 •	85.21±4.23 •	77.45±4.33 •	76.54±0.84 •	88.11±2.53 •	84.16±3.63 •
audiology	75.49 ± 8.00	71.40±6.37	70.91±7.09	69.84±6.46 •	71.58±6.76	70.82±6.99	61.22±6.63 •	71.92±6.67	62.67±9.22 •
autos	70.13±11.42	63.97±11.35 •	70.10±9.38	68.41±10.53	65.14±10.92	65.52±11.44	65.58±9.57	67.76±10.51	64.01±11.24 •
balance-scale	91.41±1.32	91.44±1.30	91.44±1.30	82.13±3.32 •	90.27±1.89 •	90.27±1.91 •	87.71±3.11 •	90.03±1.99 •	81.01±7.58 •
breast-cancer	72.91±7.98	72.94±7.71	73.25±7.60	71.73±7.40	71.07±6.30	70.63 ± 8.76	70.30 ± 1.37	72.39±7.47	72.60±6.61
breast-w	97.24±1.68	97.30±1.75	97.30±1.75	97.21±1.84	97.28±1.78	97.33±1.77	97.25±1.91	97.34±1.81	96.94±2.00
colic	81.46±6.07	78.86±6.05	82.28±5.86	83.45±5.35	81.77±6.12	83.56±5.79	84.07±5.27	83.64±5.47	78.78±6.22
colic.ORIG	73.99±7.22	74.21±7.09	74.57±5.85	74.61 ± 6.62	75.29 ± 5.92	71.78 ± 6.70	66.20±1.37 •	76.00 ± 6.53	73.04±6.39
credit-a	85.13±3.90	84.74±3.83	85.75±4.16	86.14±4.06	85.51±3.96	85.51±3.96	85.67±3.97	86.46±3.85	82.32±5.85
credit-g	75.80±3.59	75.93±3.87	72.43±3.61 •	76.13±3.64	74.06±2.85	70.43±4.55 •	70.00±0.00 •	76.14±3.62	73.23±3.63
diabetes	75.86±4.87	75.68 ± 4.85	75.93±5.07	77.02±4.87	75.03±3.95	75.94±5.49	65.16±0.47 •	76.91±5.07	73.70±5.30
glass	57.74±10.16	57.69±10.07	56.22±10.36	56.70±9.79	57.87±9.28	56.63±9.84	55.56±8.67	57.44±9.37	55.49±9.63
heart-c	82.41±6.66	83.44±6.27	84.20±6.37	83.64±6.37	84.14±6.19	83.50±6.60	83.20±6.83	83.57±6.04	81.48±7.15
heart-h	82.42±5.98	83.64±5.85	82.59±6.40	83.31±6.46	83.15±6.77	81.75±6.58	80.04±5.91	83.34±6.28	81.47±7.66
heart-statlog	83.52±6.19	83.78±5.41	84.19±6.07	84.22±5.99	83.59±6.52	83.41±6.08	81.44±5.97	84.04±5.90	81.85±6.24
hepatitis	84.52±9.61	84.06±9.91	83.60±9.77	84.52±9.22	82.97±9.89	85.15±9.45	82.28±4.54	83.35±8.24	85.22±8.87
hypothyroid	93.42±0.62	92.79±0.73 •	93.52±0.48	93.60±0.51	93.33±0.45	75.98±2.04 •	93.30±0.44	93.58±0.50	93.37±0.53
ionosphere	90.69±4.05	90.86±4.33	90.89±4.72	92.20±3.94	92.02±4.19	92.00±4.08	91.08±4.18	92.00±4.06	90.66±4.74
iris	94.87±6.28	94.33±6.79	96.87±4.29	95.33±5.40	95.93±4.73	95.93±4.73	96.07±4.65	95.53±5.19	91.87±7.38
kr-vs-kp	95.84±1.55	87.79±1.91 •	92.38±1.56 •	91.22±1.45 •	89.79±1.63 •	90.83±1.72 •	90.51±1.60 •	94.21±1.29 •	82.63±4.99
labor	95.80±8.73	96.70±7.27	84.97±12.91 •	90.20±11.28	92.30±9.98	90.57±11.18	88.63±12.10	88.10±12.66	92.17±11.15
letter	67.75±2.17	65.80±2.04 •	67.32±2.22	66.23±2.07 •	68.03±2.10	68.33±2.15	68.50±1.99	65.99±2.08 •	60.17±3.54
lymph	84.94±8.42	85.97±8.88	82.72±9.39	83.00±9.35	81.03±9.15	82.85±8.96	78.78±8.66 •	82.20±10.01	83.33±9.51
mushroom	98.32±0.99	93.58±2.03 •	98.12±1.14	98.28±1.04	98.33±0.98	98.21±1.01	97.83±1.25	97.82±1.10	93.54±3.18 •
primary-tumor	46.52±6.28	47.20±6.02	45.78±6.84	43.78±5.05	46.77±6.08	45.64±6.93	24.78±1.47 •	45.54±5.39	42.60±6.44
segment	91.54±1.82	89.03±1.66 •	90.46±2.10 •	89.92±1.76 •	85.82±2.06 •	86.23±1.89 •	85.57±2.03 •	90.48±1.56 •	85.56±3.95
sick	97.27±0.79	96.78±0.91 •	96.81±0.89	97.30±0.88	96.47±0.93 •	96.26±0.95 •	96.64±0.71 •	96.94±0.92	95.58±1.38
sonar	76.76±10.78	76.35±9.94	73.54±9.45	75.71±9.58	74.99±9.54	76.14±9.55	76.39±9.45	75.36±8.81	75.22±9.93
soybean	92.94±2.54	92.20±3.23	91.00±3.31 •	92.61±2.82	90.61±3.40 •	91.74±3.28	88.58±3.46 •	92.85±2.90	89.90±3.56
splice	95.71±1.07	95.42±1.14	95.84±1.03	95.39±1.28	94.08±1.28 •	94.50±1.21 •	88.30±2.44 •	96.14±1.03	92.61±2.17
vehicle	62.81±3.81	61.03±3.48	56.32±4.01 •	60.59±3.57	60.39±3.35	58.91±3.35 •	61.16±3.31	61.75±3.44	60.11±3.42
vote	94.35±3.67	90.21±3.95 •	94.46±2.81	93.49±3.55	93.03±3.57	93.01±3.54	95.24±2.91	94.83±3.01	89.86±4.56
vowel	67.26±4.86	66.09±4.78	62.75±5.10 •	65.32±4.59	64.82±4.63	64.65±4.66	66.14±4.67	66.65±4.73	57.41±7.79
waveform-5000	82.60±3.52	79.80±2.97 •	79.71±2.97	77.78±2.92 •	78.71±3.05 •	78.65±3.05 •	80.43±3.38	79.56±3.00 •	78.38±3.52
ZOO	95.95±5.62	94.37±6.79	91.51±7.68 •	95.15±4.98	95.55±5.30	94.39±5.96	94.25±4.93	90.65±7.29 •	92.58±7.25

 $[\]bullet: Statistically \ significant \ degradation.$

$$E = \frac{P_0 - t_0(t_0 + 1)/2}{t_0 t_1} \tag{8}$$

where t_0 and t_1 are the number of negative and positive instances, repressively. $P_0 = \sum r_i$, with r_i denoting the rank of the *i*th negative instance in the ranked list. It is clear that AUC

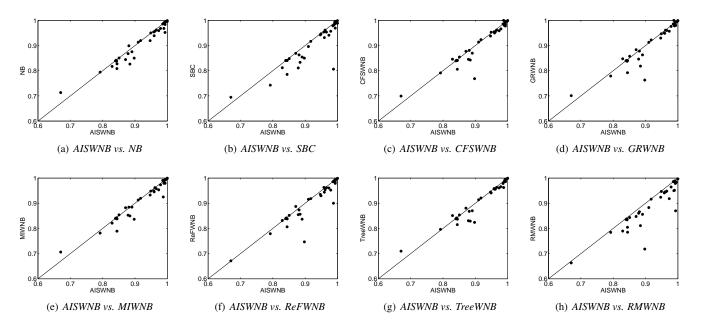


Figure 4: AISWNB vs. competing algorithms: area under the ROC curve (AUC).

Table 4: The detailed experimental results on area under the ROC curve (AUC) and standard deviation %.

Data set	AISWNB	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB	RMWNB
anneal	98.88±1.67	98.76±1.84	98.27±2.54	98.66±1.89	98.63±1.90	98.82±1.73	98.77±1.77	98.58±1.96	98.74±1.81
anneal.ORIG	97.90±3.22	96.79±5.42	95.68±6.71	96.61±6.55	97.59±3.91	97.32±4.28	95.21±7.68	96.58±6.09	96.46±4.59
audiology	84.32±1.56	83.85±1.44	84.05±1.61	84.06±1.59	83.75±1.45	84.26±1.56	83.98±1.58	83.99±1.59	83.43±1.37 •
autos	94.68±3.27	91.96±3.32 •	94.26±3.13	93.78±3.40	92.98±3.68	93.27±3.73	93.50±3.48	94.44±3.70	92.39±3.85
balance-scale	89.76±4.24	85.00±4.03 •	85.00±4.03 •	76.84±4.78 •	76.29±3.84 •	83.64±3.79 •	74.67±3.75 •	82.40± 4.07 •	71.81±6.47
breast-cancer	67.02±14.05	71.32±13.81 o	69.45±14.71	69.93±13.90	70.10±14.49	70.61±13.08	67.13±12.81	70.98±13.89 o	66.31±15.36
breast-w	99.13±0.94	99.23±0.83	99.23±0.83	99.19±0.87	99.20±0.88	99.16±0.89	99.19±0.90	99.29±0.76	98.89±1.21
colic	87.08±5.05	84.42±5.45 •	86.91±7.08	87.72±5.92	88.35±5.47	88.23±6.13	88.72±5.32	88.06±5.56	84.75±4.97
colic.ORIG	82.93±7.23	81.70±7.23	81.19±5.32	84.58±5.29	84.75±4.77	82.10±4.97	83.15±4.33	85.11±5.78	78.96 ± 6.56
credit-a	91.74±3.72	91.97±3.14	91.66±3.45	92.34±3.28	92.22±3.64	91.95±3.31	91.83±3.12	92.07±3.33	91.62±3.43
credit-g	79.22±5.20	79.42±4.52	74.26±5.55 •	79.13±5.01	77.90±5.62	78.15±5.84	77.91±6.15	79.64±4.85	78.45 ± 4.12
diabetes	84.44±4.68	82.74±4.94 •	84.08±5.04	84.10±4.66	84.00±4.80	83.81±4.80	83.79±4.64	83.67±4.81	78.49±4.81
glass	88.43±3.18	82.63±6.07 •	83.33±7.29	84.25±4.60 •	81.84±5.59 •	84.97±6.12	85.63±4.36 •	82.90±5.23 •	81.11±5.19
heart-c	84.17±0.66	84.17±0.50	84.05±0.61	84.13±0.54	84.07±0.57	84.05±0.57	84.15±0.56	84.14±0.55	83.79±0.67
heart-h	83.97±0.51	83.92±0.63	84.07±0.68	83.91±0.68	83.97±0.72	83.99±0.72	83.91±0.65	83.82±0.72	83.56±0.81
heart-statlog	91.00±4.80	91.33±5.15	89.61±5.01	91.39±4.17	91.28±4.84	91.28±4.13	91.56±3.94	91.39±4.26	88.28 ± 6.60
hepatitis	88.11±10.12	89.90±8.24	86.29±11.02	88.07±9.17	87.79±9.55	88.46±9.07	87.35±8.74	86.79±9.78	86.61±10.60
hypothyroid	89.05±9.74	87.53±9.20 •	85.43±8.19 •	86.96±8.70 •	86.30±9.11 •	88.46±9.98	83.68±7.65 •	87.03±8.80 •	85.60±8.07 •
ionosphere	95.93±2.75	93.90±3.21	93.14±4.49	94.99±2.82	94.86±2.97	94.54±3.05	94.39±3.31	95.55±2.51	94.44± 2.89
iris	99.20±1.80	98.93±2.16	99.40±1.27	99.20±1.80	99.20±1.80	99.20±1.80	99.20±1.80	99.20±1.80	97.87±4.08
kr-vs-kp	99.29±0.36	95.20±1.28 •	96.94±0.83 •	97.75±0.92 •	98.10±0.77 •	97.95±0.81 •	98.48±0.69 •	98.72±0.67 •	87.00±2.22 •
labor	98.75±3.95	98.75±3.95	80.63±33.13	100.00±0.00	100.00 ± 0.00	92.50±23.72	90.00±27.51	96.25±8.44	95.00±12.08
letter	96.18±0.62	95.68±0.73 •	96.13±0.57	95.44±0.71 •	96.10±0.61	96.20±0.59	95.41±0.67 •	95.60±0.71 •	94.47±0.81
lymph	94.88±4.92	95.01±4.87	94.64±4.87	95.30±5.32	94.67±4.71	94.81±4.92	92.99±4.15	94.10±4.65	94.68±4.74
mushroom	99.98±0.23	99.59±0.17 •	98.88±0.80 •	99.82±0.15 •	99.81±0.16 •	99.91±0.07 •	99.87±0.11	99.92±0.10	99.67±0.28 •
primary-tumor	85.07±2.67	85.05±2.96	84.81±2.92	85.44±2.17	85.76±2.21	85.40±2.53	85.20 ± 1.82	85.34±2.54	84.33 ± 2.64
segment	99.30±0.27	98.37±0.52 •	98.72±0.55 •	98.51±0.55 •	97.97±0.62 •	97.95±0.64 •	97.90±0.63 •	98.59±0.46 •	98.16±0.53
sick	97.33±1.41	95.92±2.48	94.10±2.89 •	95.88±2.66	95.67±2.72	95.32±3.09	96.04±2.66	96.11±2.66	91.82±2.61
sonar	87.86±10.08	86.79±9.83	81.11±11.83 •	84.52±10.52	84.61±10.17	85.27±10.16	85.47±10.10	83.05±9.80	86.03±10.79
soybean	99.97±0.07	99.90±0.07 •	99.87±0.11	99.93±0.07	99.91±0.07	99.92±0.06	99.89±0.07	99.92±0.06	99.68±0.19
splice	99.50±0.21	99.41±0.22 •	99.45±0.26	99.45±0.27	99.27±0.29 •	99.32±0.27 •	99.40±0.27	99.54±0.22	98.34±0.34
vehicle	84.41±3.74	80.85±3.73 •	78.53±4.19 •	80.57±4.24 •	79.21±4.19 •	78.91±3.95 •	80.64±4.60 •	81.50±3.92 •	80.46±3.73
vote	98.93±1.10	96.79±1.95 •	98.25±1.64	98.02±1.44	97.82±1.58	97.94±1.47	98.53±1.35	98.56±1.12	95.15±2.45
vowel	96.57±0.63	96.19±0.72 •	95.33±0.83 •	96.18±0.90	95.99±0.99	95.80±1.08	96.20±0.92	96.31±0.78	94.81±0.95
waveform-5000	95.79±1.50	95.41±1.36	95.38±1.43	95.44±1.45	94.97±1.46	94.94±1.50	96.28±1.16	95.85±1.28	94.12±1.86
ZOO	98.57±1.66	98.57±1.66	97.86±2.37	98.57±1.66	98.57±1.66	99.05±1.23	98.57±1.66	97.86±2.37	98.57±1.66

 $[\]circ, \bullet \text{:}$ Statistically significant upgradation and degradation, respectively.

is essentially a measure of the quality of ranking. The above measure can only deal with two-class problem. For multiple classes, Hand & Till (2001) proposes an improved AUC calcu-

lating measure:

$$E' = \frac{2}{g(g-1)} \sum_{i < j < L} E(c_i, c_j)$$
 (9)

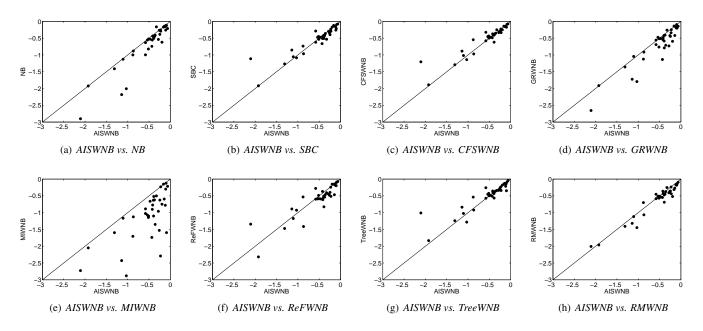


Figure 5: AISWNB vs. competing algorithms: averaged conditional log likelihood (CLL).

Table 5: The detailed experimental results on averaged conditional log likelihood (CLL) and standard deviation %.

Data set	AISWNB	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB	RMWNB
anneal	-11.73±8.27	-15.69± 9.70	• -15.73±7.33	• -14.67±4.88	-17.17±10.48	-78.00±39.77			• -16.39±5.44 •
anneal.ORIG	-25.95±9.51	-26.55 ± 8.07	-34.30±7.60	• -29.77±5.65		-152.86±51.66		• -29.11±6.19	-36.35±4.96
audiology	-210.64±61.79				∘ -265.53±85.38 •			∘ -100.85±18.51	
autos	-114.18±51.49	-218.20±102.17	• -85.52±29.92	-88.69±32.47	-172.38±85.19	-242.85±136.90	 -89.03±15.96 	-83.84±30.97	-131.85±59.60
balance-scale	-38.92±4.81	-50.83 ± 2.27	 -50.83±2.27 	• -62.01±2.28		▶ -51.87± 2.24		• -56.55±2.16	• -68.96±1.97 •
breast-cancer	-58.21±14.01	-63.30±20.73	-62.27±18.57	-57.88 ± 14.47	-55.79 ± 7.11	-88.72±35.20	• -59.81±1.75	-57.16±13.78	-58.29 ± 10.45
breast-w	-10.93±8.17	-26.10±23.29	-26.10±23.29	-16.06±12.78	• -19.42±16.26	-29.87±25.75	-11.75±8.98	-14.52±12.15	-17.54±14.12
colic	-51.43±15.04	-82.31±26.04	• -45.03±15.68	-43.71±14.50	-76.49±33.29	→ -114.24±59.00	 -48.55±2.67 	-42.76±12.86	-56.87±16.17
colic.ORIG	-53.74±15.12	-55.54±15.53	-47.46 ± 7.72	-45.33±7.31	-47.45±11.37	-107.18±33.73	-58.99±1.54	-45.18 ± 8.06	-51.61±7.10
credit-a	-35.52 ± 7.02	-41.34±11.26	 -36.20±8.35 	-35.36±8.19	-71.13±24.35	-135.06±44.97	 -46.36±2.28 	• -35.73±8.14	-37.44±8.19
credit-g	-51.02±6.80	-52.42 ± 7.29	-53.45±5.47	-49.80 ± 5.24	-51.48±3.74	-110.29±33.21	 -58.28±0.45 	-49.67±5.52	-51.05±3.23
diabetes	-47.40 ± 7.06	-53.18±10.46	 -47.43±6.96 	-47.53 ± 6.59	-51.03 ± 2.71	-66.08 ± 15.95	-58.98±0.99	 -48.71±7.58 	-53.29±5.43
glass	-110.74±16.18	-112.89±19.39	-105.31±23.28	-101.69±14.84	-104.39 ± 18.75	-116.55±24.63	-117.45±6.44	-102.39±13.45	∘ -111.84±14.83
heart-c	-37.25±11.66	-45.10±16.99	• -42.37±13.85	-37.98±10.62	-40.89 ± 13.51	-61.93±27.50	-53.89±3.23	• -37.57±11.28	-45.11±12.67
heart-h	-39.29±10.79	-46.47±17.46	• -41.49±18.32	-41.16±15.97	-58.45 ± 32.43	-110.01±61.17	• -50.58±3.53	• -43.24±17.01	-47.18±18.08
heart-statlog	-39.77±11.61	-45.09±16.73	-45.77±12.75	-38.98±9.43	-42.12±12.19	-63.07±24.21	• -54.17±2.20	• -38.55±9.52	-45.27±13.86
hepatitis	-42.24±19.93	-55.52±26.52	• -51.14±24.18	-37.18±15.13	-54.35±25.26	-90.12±47.24	• -36.72±5.87	-35.63±12.40	-36.06±16.12
hypothyroid	-23.06±4.09	-25.83 ± 5.10	• -23.45±2.47	-22.77±3.48	-33.92±6.80	-229.14±15.95	-24.28±1.92	-22.94±3.71	-23.68±2.90
ionosphere	-58.28±27.66	-99.71±39.08	• -28.78±11.61	-32.56±10.80	-69.17±25.88	-102.56±38.33	-27.78±8.98	∘ -25.73±9.04	∘ -45.20±15.87
iris	-16.05±21.20	-17.09±18.44	-12.21±11.73	-14.31±13.53	-13.95±11.59	-15.09±19.29	-15.47±9.07	-15.03±10.38	-31.23±11.86
kr-vs-kp	-22.44±3.44	-29.17± 2.61	• -29.42±1.56	• -33.64±1.67	• -24.86±2.72	-23.07 ± 5.52	-51.42±0.90	• -33.97±1.40	• -43.70±2.89
labor	-32.62±58.47	-16.33±18.27	-53.33±55.16	-25.05±22.81	-26.12±26.51	-50.12±86.71	-40.81±12.46	-34.04±21.59	-25.18±25.41
letter	-131.06±16.50	-141.18±13.19	• -126.77±10.75	-129.18±8.46	-135.78±11.98	-159.59±15.95	• -147.16±5.25	• -124.01±9.14	-140.77±10.48
lymph	-41.50±29.51	-43.67±21.80	-44.23±23.18	-40.17±16.98	-48.92 ± 24.70	-77.21±46.06	• -61.21±7.59	-40.45±14.42	-43.34±14.44
mushroom	-6.47±4.47	-21.35±10.02	-7.94±3.93	-7.54±4.35	-14.55±9.96	-21.49±15.63	• -7.47±3.38	-4.61±1.98	-9.45±5.63
primary-tumor	-192.48±29.10	-192.48±29.10	-191.85±26.09	-189.01±17.16	-191.34±28.77	-205.01±35.39	-231.81±8.00	• -183.30±18.38	-196.23±20.31
segment	-28.01±7.28	-53.79±16.55	-29.68±7.00	-33.61±7.60	• -73.20±22.92	-95.99±31.85	-44.91±9.49	• -33.68±7.74	• -40.80±7.24 •
sick	-8.94 ± 2.02	-12.13± 3.07	• -9.37±1.74	-9.05±2.35	-20.08±5.08	-159.49±38.79	• -9.04±1.51	-9.47±2.34	-14.36±1.00
sonar	-87.88±49.02	-99.84±57.80	-73.80±31.79	-54.72±23.58	-112.51±61.06	-170.49±99.29	• -53.32±5.03	-53.80±17.28	-70.07±36.18
soybean	-24.61±7.16	-38.71±14.36	• -39.72±15.49	• -28.24±6.95		-62.61±26.64		• -25.84±5.37	-35.62±8.96
splice	-14.32±2.97	-14.63 ± 2.70	-13.59±2.86	-15.11±1.79	-42.06±9.42	-59.55±13.56	• -46.68±2.66	• -12.85±1.84	-26.96±1.78
vehicle	-103.14±20.56		• -108.54±15.01	-113.99±17.20	-179.37±32.99		• -92.71±7.80		• -144.67±21.50 •
vote	-19.57±11.92	-62.16±29.81	• -17.21±10.75	-25.04±14.94		-74.89±47.32	• -15.25±8.06	-18.01±11.83	-51.92±18.60
vowel	-86.72±9.97	-88.30 ± 9.00	-96.55±8.84	• -97.03±7.38	• -91.60±13.72	-112.48±22.63			• -106.41±6.30
waveform-5000	-43.30±8.36	-74.37±17.55	• -66.35±15.03	• -48.84±7.92	-113.49±28.22			• -51.10±10.63	-54.77±12.34
zoo	-9.96±10.34	-11.82±10.79	-21.17±18.37	• -17.44±10.34	-9.65±9.97	-12.35±13.26			• -18.46±13.37

 $[\]circ, \bullet \colon Statistically significant upgradation and degradation, respectively.$

where g is the number of classes and $E(c_i, c_j)$ is the AUC of each pair of classes c_i and c_j .

In our experiments, CLL value of a classifier h on data set \mathcal{D}

with N instances is evaluated by using Eq. (10).

$$CLL(h|\mathcal{D}) = \sum_{i=1}^{N} \sum_{j=1}^{n} log P_h(y_i|x_{t,j})$$
 (10)

Table 6: Two-tailed *t*-test on classification accuracy (ACC).

Table 8: Two-tailed *t*-test on averaged conditional log likelihood (CLL).

	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB	RMWNB
SBC	6/25/5							
CFSWNB	8/25/3	4/28/4						
GRWNB	5/25/6	2/30/4	2/27/7					
MIWNB	5/22/9	2/26/8	2/26/8	2/28/6				
ReFWNB	5/19/12	3/23/10	4/19/13	2/24/10	5/23/8			
TreeWNB	7/26/3	6/27/3	4/30/2	8/24/4	10/25/1	11/23/2		
RMWNB	1/23/12	1/25/10	0/22/14	0/28/8	3/25/8	6/24/6	1/20/15	
AISWNB	11/25/0	10/26/0	8/28/0	9/27/0	11/25/0	14/22/0	8/28/0	15/21/0

	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB	RMWNB
SBC	16/16/4							-
CFSWNB	23/8/5	12/18/6						
GRWNB	11/15/10	6/9/21	3/7/26					
MIWNB	1/8/27	2/2/32	2/3/31	1/5/30				
ReFWNB	12/13/11	3/13/20	3/9/24	14/9/13	24/8/4			
TreeWNB	25/4/7	12/17/7	11/16/9	22/10/4	31/2/3	21/11/4		
RMWNB	14/13/9	2/18/16	0/7/29	17/13/6	28/5/3	17/9/10	3/5/28	
AISWNB	21/15/0	8/27/1	6/29/1	17/19/0	29/7/0	18/16/2	7/26/3	16/20/0

Table 7: Two-tailed *t*-test on area under the ROC curve (AUC).

	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB	RMWNB
SBC	4/25/7							
CFSWNB	9/24/3	10/23/3						
GRWNB	8/24/4	9/23/4	2/29/5					
MIWNB	8/25/3	5/27/4	4/24/8	6/26/4				
ReFWNB	5/27/4	8/24/4	3/30/3	7/25/4	6/26/4			
TreeWNB	13/19/4	9/24/3	7/28/1	9/26/1	9/23/4	8/27/1		
RMWNB	0/19/17	3/19/14	0/16/20	2/18/16	2/16/18	2/20/14	0/15/21	
AISWNB	15/20/1	10/26/0	8/28/0	8/28/0	6/30/0	7/29/0	7/28/1	17/19/0

where h is a learning model. In Friedman et al. (1997), maximizing Eq. (10) amounts to best approximate the conditional probability of \mathcal{Y} given each text instance x_t , and is equivalent to minimizing the conditional cross-entropy.

5.2. UCI Benchmark Learning Tasks

We first test the performance of the proposed method on 36 UCI benchmark data sets (Bache & Lichman, 2013), which include a wide range of domains with data characteristics described in Table 2. These 36 UCI data sets have always been considered as benchmark data sets for data mining and machine learning related classification tasks (Jiang et al., 2009), in order to validate the effectiveness of new algorithm. In General, an algorithm is effective if it can significantly outperform its peers on at least 8 UCI data sets (Wu et al., 2013a,b; Jiang & Zhang, 2005; Zhang & Sheng, 2004).

In order to demonstrate the performance of weighted NB, compared to the generic NB classifiers, our experiments will first report the performance of NB and WNB classifiers with respect to classification accuracy measured by ACC, ranking performance measured by AUC, and probability estimation measured by CLL. More specifically, we compare an NB classifier (Friedman et al., 1997) with SBC (Langley & Sage, 1994), CFSWNB (Hall, 2000), GRWNB (Zhang & Sheng, 2004), MIWNB (Jiang et al., 2012b), RefWNB (Robnik-Šikonja & Kononenko, 2003) and TreeWNB Hall (2007).

The purpose of the second experiment is to compare the proposed attribute weighted Naive Bayes, namely AISWNB, with each attribute weighted approach in literature. Finally, we compare related algorithms via two-tailed *t*-test with a 95% confidence level. Based on the statistical theory, the difference is statistically significant only if the probability of significant dif-

ference is at least 95 percent, *i.e.*, the *p*-value for a *t*-test between two algorithms is less than 0.05.

5.2.1. Attribute Weighted NB vs. Standard NB

Tables 3, 4, and 5 report the detailed results (the average ACC, AUC, and CLL values and the respective standard deviation values) of AISWNB and other baseline algorithms, respectively. In these three tables, the symbols \circ and \bullet represent statistically significant upgradation and degradation over the proposed AISWNB with the p-value less than 0.05. In addition, Tables 6, 7, and 8 illustrate the compared results of two-tailed t-test, in which each entry w/t/l means that the algorithm in the corresponding row wins in w data sets, ties in t data sets, and loses in t data sets on the 36 UCI data sets, compared to the algorithm in the corresponding column. Overall, the results can be summarized as follows:

- 1. TreeWNB shows the best performance. Compared to NB, TreeWNB wins 7 data sets, ties 26 data sets, loses 3 data sets on ACC, and shows great superiority to NB on AUC (13 wins and 4 losses) and CLL (25 wins and 7 losses).
- 2. CFSWNB outperforms NB on ACC and AUC. Compared to NB, CFSWNB wins 8 data sets, ties 25 data sets and loses 3 data sets on ACC, with the better performance (9 wins and 3 losses) on AUC. But CFSWNB significantly outperforms NB on CLL (23 wins and 5 losses).
- 3. GRWNB slightly outperforms NB. The results show that GRWNB has a slightly better performance on AUC (8 wins and 4 losses), and almost ties NB both on ACC (5 wins and 6 losses) and CLL (11 wins and 10 losses).
- 4. SBC's performance is comparable with NB. More specifically, SBC almost ties NB (6 wins and 5 losses) on ACC, and is sightly worse than NB (4 wins and 7 losses) on AUC. On the contrary, SBC shows better performance on CLL (16 wins and 4 losses) compared with NB.
- 5. MIWNB is inferior to NB. Because, although MIWNB performs a little better than NB on AUC (8 wins and 3 losses), it shows bad performance on ACC (5 wins and 9 losses), and is significantly inferior to NB on CLL (1 wins and 27 losses).
- 6. ReFWNB is an ineffective attribute weighting method for improving accuracy performance. In terms of ACC, it is

inferior to NB (5 wins and 12 losses), SBC (3 wins and 10 losses) and all other attribute weighted method: CF-SWNB (4 wins and 13 losses), GRWNB (2 wins and 10 losses), MIWNB (5 wins and 8 losses) and TreeWNB (2 wins and 11 losses). Besides, ReFWNB almost ties NB both on ACC (5 wins and 4 losses) and CLL(12 wins and 11 losses).

Overall, our experiments show that most existing attribute weighting approaches cannot achieve good performance with respect to the measures studied in our experiments, including AUC, CLL, and ACC. What is more, the classification accuracy (ACC) is a fundamental evaluation criteria in many real-world applications (*e.g.*, image retrieval). In this case, it is important to design a novel attribute weighting approach for WNB that could improve the classification performance for different learning tasks.

5.2.2. Accuracy Comparisons Between AISWNB and Baselines

In Figs. 3, 4, and 5, we report the performance of the proposed AISWNB, which uses artificial immune system principles to calculate attribute weight values for NB classification in term of classification accuracy (ACC), and area under the ROC curve (AUC), averaged conditional log likelihood (CLL), respectively. In these figures, data points below the y=x diagonal line are data sets on which AISWNB achieves better results than the rival algorithm. The detailed results are further presented in Tables 3, 4, and 5. Our experimental results indicate that AISWNB has very significant gain compared to other attribute weighting methods on the above three evaluation criteria. In summary, our experimental results in Tables 6, 7, and 8 show:

- 1. AISWNB not only significantly outperforms NB (11 wins and 0 loss) in ACC, but is also in AUC (15 wins and 1 loss) and CLL (21 wins and 0 loss).
- 2. AISWNB significantly outperforms SBC (10 wins and 0 loss) in ACC and AUC, and also outperforms SBC in CLL with 8 wins and 1 loss.
- 3. AISWNB outperforms CFSWNB (8 wins and 0 losses), GRWNB (9 wins and 0 loss), MIWNB (11 wins and 0 losses), ReFWNB (14 wins and 0 losses) and TreeWNB (8 wins and 0 losses) in ACC. The similar superiority for AISWNB could also been found in other two evaluation criteria (AUC and CLL).

5.3. Image Retrieval Learning Tasks

In image classification, an image is classified according to its visual content. An important application of image classification is image retrieval: searching through an image data set to obtain (or retrieve) those images with particular visual content. For example, finding pictures containing a car.

In this part of experiment, we report AISWNB's performance for automatic image retrieval tasks. The original data are color images from Corel benchmark data set (Li & Wang, 2008). For



Figure 6: Example images used in the experiment are from the COREL image categorization database. Each row shows images in one category, and the selected images in our data set are from six categories ("people", "elephant", "car", "fox", "tiger", and "bus").

each image, four sets of visual features are extracted, including color histogram (Wu et al., 2014), color histogram layout (Liu & Yang, 2013), color moments (Chen et al., 2013), and co-occurrence texture (Luo et al., 2013). We choose the color histogram approach in the HSV (Hong et al., 2013) color space as color features. The HSV color space is divided into 32 subspaces (32 colors with 8 ranges of *H* and 4 ranges of *S*). For each image, the value in each dimension of the color histogram represents the density of the corresponding color in the entire image. So the whole image is represented as 32-dimensional color histogram features.

For the color histogram layout, each image is divided into 4 sub-images (one horizontal split and one vertical split), in which 4×2 color histogram for each subimage is computed. In this case, we can obtain another 32-dimensional features. In addition, the color moment feature has 9 dimensions, in which one (mean, or standard deviation, or skewness) for each of H, S, and V in HSV color space. At last, for texture feature, images are converted to 16 gray-scale images, then co-occurrence in 4 directions is computed (horizontal, vertical, and two diagonal directions). The corresponding 16 texture feature values are: one for each direction, second angular moment, contrast, inverse difference moment, and entropy. The similar image feature exploration approach could also been found in Ortega et al. (1998).

Some sample images from the benchmark data sets are shown in Fig. 6. In our experiment, we use six different categories ("people", "elephant", "car", "fox", "tiger", and "bus") to form six binary learning problems (one for each category). To obtain negative classes for each of the six positive classes, we randomly selected 100 images from the remaining classes. Therefore, we form six binary image data sets, each of which is evenly balanced and contains exactly 200 images.

Tables 9, 10, and 11 report the detailed experimental results in terms of ACC, AUC, and CLL, respectively. Overall, the proposed AISWNB can achieve the best performance in all cases, which demonstrate that AISWNB can outperform other baseline methods for visual feature based image classification.

Table 9: Experimental results on the image classification data sets with respect to the clarification accuracy (ACC) %.

Data set	AISWNB	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB
bike	80.03	71.46	74.66	74.28	74.03	73.94	73.90	72.66
car	75.05	66.47	66.74	67.36	66.14	68.03	59.25	67.25
elephant	82.48	73.50	77.15	75.95	73.00	76.40	75.05	74.25
fox	70.34	56.25	53.10	55.90	53.15	56.75	54.35	55.95
people	86.67	75.67	72.89	73.85	73.3	74.14	73.40	74.96
tiger	89.87	78.60	78.85	78.15	77.3	80.10	81.05	80.85

Table 10: Experimental results on the image classification data sets with respect to the area under the ROC curve (AUC) %.

Data set	AISWNB	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB
bike	89.76	79.80	81.51	81.53	80.96	81.63	80.96	80.42
car	82.54	72.33	72.21	73.35	72.55	73.10	72.14	73.20
elephant	90.62	84.96	86.48	86.62	86.56	85.56	84.41	85.04
fox	72.07	57.14	54.62	58.53	56.46	59.07	56.34	57.74
people	88.54	83.48	81.13	84.28	83.25	83.56	83.84	84.28
tiger	91.23	86.18	85.23	85.43	85.94	85.95	86.85	86.41

Table 11: Experimental results on the image classification data sets with respect to the averaged conditional log likelihood (CLL) %.

Data set	AISWNB	NB	SBC	CFSWNB	GRWNB	MIWNB	ReFWNB	TreeWNB
bike	-43.78	-159.58	-75.70	-63.64	-129.31	-387.03	-61.66	-72.12
car	-32.54	-116.60	-74.20	-63.51	-78.12	-237.54	-67.75	-66.45
elephant	-44.12	-60.95	-48.83	-48.22	-137.56	-742.52	-64.49	-51.20
fox	-28.40	-94.12	-77.35	-69.83	-130.99	-601.20	-69.20	-69.88
people	-37.90	-124.87	-57.82	-60.13	-200.71	-428.08	-57.44	-58.12
tiger	-30.88	-66.62	-55.99	-49.50	-133.67	-603.65	-62.52	-48.48

5.4. Convergence and Learning Curves

In order to investigate the convergence of the AISWNB algorithm, we report the relationship between the number of iterations and one of the evaluation criteria (*i.e.*, classification accuracy) on 12 UCI data sets (half of them are from data sets with a relatively large number of instances, the remaining half represent data sets with a relatively large number of attributes).

The results are shown in Figs. 7 and 8, where each point in the curves corresponds to the mean accuracy from 10-fold cross validation under the underlying iteration with the current optimal attribute weight values.

The results in Figs. 7 and 8 show that AISWNB has a quick convergence to achieve a higher classification accuracy than other algorithms. To further study the convergence of the algorithm, let's take the "kr-vs-kp" data set, which is a high-dimensional data set (37 attributes) with 3196 instances, as an example. Existing research by Kohavi (1996) has shown that "kr-vs-kp" data set has strong attribute dependencies. Our results show that AISWNB achieves 96.9% classification accuracy, which is significantly higher than NB's accuracy 88.1% on the same data set. The accuracy of the final convergence

is better than CFSWNB (91.3%), GRWNB (90%), MIWNB (90.8%), ReFWNB (90.4%), and TreeWNB (94.4%). Similar levels of improvement can also be observed from other data sets

In some situations, the proposed AISWNB can achieve a better accuracy than a number of attribute weighted baseline methods within only one iteration. This demonstrates the good convergence performance of AISWNB. Meanwhile, in order to determine whether the improvement of AISWNB is attributed to random attribute weights or simple random weight values can also have a high classification performance, we use random weight values and denote the results by RMWNB in the last column of Tables 3 (ACC), 4 (AUC) and 5 (CLL). The results on 36 UCI benchmark data sets clearly show that random attribute weight selection is ineffective to improve the performance of WNB.

According to the *t*-test results in Tables 6, 7, and 8, some detailed explanations can be discussed as follows:

1. RMWNB is severely inferior to AISWNB (0 win and 15 losses) in ACC, (0 win and 17 losses) in AUC, and (0 win

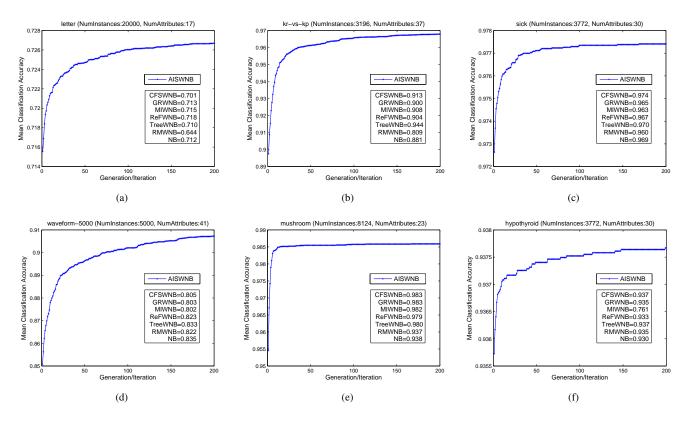


Figure 7: Convergence learning curves of AISWNB for 6 data sets with large number of instances. (a) letter: **20000** instances. (b) kr-vs-kp: **3196** instances. (c) sick: **3772** instances. (d) waveform-5000: **5000** instances. (e) mushroom: **8124** instances. (f) hypothyroid: **3772** instances.

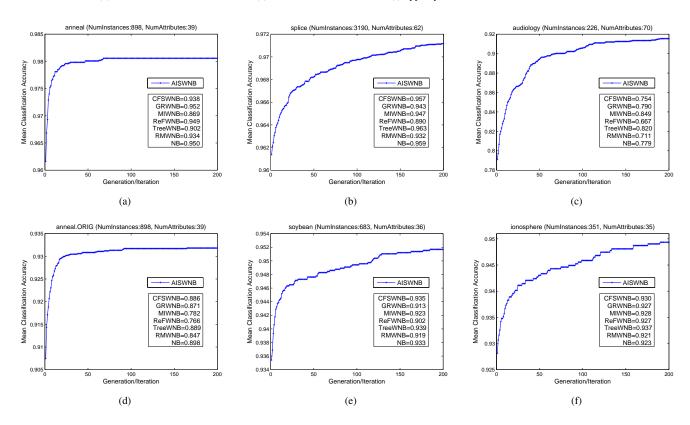


Figure 8: Convergence learning curves of AISWNB for 6 data sets with large number of attributes. (a) anneal: **39** attributes. (b) splice: **62** attributes. (c) audiology: **70** attributes. (d) anneal.ORIG: **39** attributes. (e) soybean: **36** attributes. (f) ionosphere: **35** attributes.

and 16 losses) in CLL.

- 2. RMWNB is inferior to SBC, CFSWNB, and TreeWNB in all above three evaluation criteria. Specifically, for ACC it is around (0 wins and 13 losses), AUC around (0 wins and 20 losses), and CLL around (2 wins and 28 losses).
- 3. Although RMWNB shows a better performance than GRWNB (17 win and 6 losses) and MIWNB (28 win and 3 losses) in terms of CLL values, its performs is inferior to GRWNB and MIWNB on the other two evaluation criteria ACC and AUC.
- 4. Compared with ReFWNB, RMWNB seems competitive with ReFWNB. This is due to the fact that RMWNB ties ReFWNB (6 wins and 6 losses) in ACC, and shows better CLL performance with (17 wins and 10 losses). On the contrary, RMWNB is worse than ReFWNB in AUC (2 wins and 14 losses). It is worth noting that in Section 5.2.1 we have investigated accuracy performance for a variety of attribute weighting methods for WNB in the literature. ReFWNB has shown to be an inappropriate method to improve the ACC of WNB.

6. Conclusions

In this paper, we proposed to improve Naive Bayes classification by relaxing the attribute conditional independence assumption. Because attributes may play different roles in realwork applications, many existing works have proposed to use attribute weighting, which is an expansion of feature selection, to improve NB classification. In the paper, we argued that existing attribute weighing methods separate the attribute weighting and the Naive Bayes learning into two separated steps, without taking the Naive Bayes objective function into consideration. As a result, the selected attribute weights may not directly correspond to the final classification performance.

Alternatively, our research intends to design an integrated process with attribute weighting being seamlessly integrated into the Naive Bayes learning objective. To this end, we proposed to improve the Naive Bayes classification by self-adaptively assigning proper weight values to the attributes. Our method, namely AISWNB, uses immunity theory in artificial immune systems, including initialization, clone, section, and mutation, to self-adaptively search optimal weight values for weighted Naive Bayes classification.

In each generation (*i.e.*, iteration), AISWNB will choose the appropriate affinity function to meet the different learning tasks. After that, each individual (*i.e.*, weight vector) will be sorted by the affinity in order to select the best individual as the memory antibody. By cloning the best individual to replace the individuals with low affinity, the performance of the current population will be enhanced. Meanwhile, in order to maintain the diversity, the mutation operation is adopted. The iterative process will continue until the algorithm converges, through which, the selected attribute weight values can adapt to the underlying training data to ensure good performance gain.

Experiments and comparisons on 36 UCI benchmark data sets and six image classification data sets demonstrated that AISWNB outperforms state-of-the-art attribute weighted Naive Bayes approaches in terms of three performance metrics, including classification accuracy (measured by ACC), class ranking performance (measured by AUC), and class probability estimation (measured by CLL). The convergence study also shows that the proposed AISWNB has good convergence speed.

Overall, this paper provided an effective approach to self-adaptively calculate the attribute weight for Naive Bayes classifiers. Our principle of using evolutionary machine learning can be applied/extended to other weight optimization problems, such as weighted one-dependence estimators (Wu & Cai, 2014) in Bayesian networks. In addition, the affinity function in AISWNB may be changed when facing different evaluation criteria. In this case, further research can consider integrating affinity functions to achieve a trade-off between different evaluation criteria for optimal performance gain.

Acknowledgments

The work was supported by the Key Project of the Natural Science Foundation of Hubei Province, China (Grant No. 2013CFA004), and the National Scholarship for Building High Level Universities, China Scholarship Council (No. 201206410056), and National Natural Science Foundation of China (Grant No. 61403351 and 61370025). It is also partially supported by the Australian Research Council Discovery Projects under Grant No. DP140100545 and DP140102206. This research was also partially done when the first author visited Sa-Shixuan International Research Centre for Big Data Management and Analytics hosted in Renmin University of China. This Center is partially funded by a Chinese National "111" Project "Attracting International Talents in Data Engineering and Knowledge Engineering Research".

References

Aickelin, U., Dasgupta, D., & Gu, F. (2013). Artificial immune systems (intros 2). *CoRR*, *abs/1308.5138*.

Bache, K., & Lichman, M. (2013). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Castro, L. N. D., & Timmis, J. (2002). Artificial immune systems: A novel paradigm to pattern recognition. In *Artificial Neural Networks in Pattern Recognition* (pp. 67–84). Springer Verlag, University of Paisley, UK.

Chen, J., Huang, H., Tian, S., & Qu, Y. (2009). Feature selection for text classification with naive bayes. *Expert Systems with Applications*, *36*, 5432–5435.

Chen, Y., Sampathkumar, H., Luo, B., & wen Chen, X. (2013). ilike: Bridging the semantic gap in vertical image search by integrating text and visual features. *IEEE Transactions on Knowledge and Data Engineering*, 25, 2257– 2270.

Cuevas, E., Osuna-Enciso, V., Wario, F., Zaldívar, D., & Pérez-Cisneros, M. (2012). Automatic multiple circle detection based on artificial immune systems. *Expert Systems with Applications*, 39, 713–722.

Er, O., Yumusak, N., & Temurtas, F. (2012). Diagnosis of chest diseases using artificial immune system. *Expert Systems with Applications*, 39, 1862–1868.
Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, 29, 131–163.

- Grossman, D., & Domingos, P. (2004). Learning bayesian network classifiers by maximizing conditional likelihood. In *Proceedings of the Twenty-first In*ternational Conference on Machine Learning ICML'04 (pp. 361–368). New York, NY, USA.
- Haktanirlar Ulutas, B., & Kulturel-Konak, S. (2012). An artificial immune system based algorithm to solve unequal area facility layout problem. Expert Systems with Applications, 39, 5384–5395.
- Hall, M. (2007). A decision tree-based attribute weighting filter for naive bayes. Knowledge-Based Systems, 20, 120–126.
- Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning* ICML '00 (pp. 359–366). San Francisco, CA, USA.
- Han, E.-H., Karypis, G., & Kumar, V. (2001). Text categorization using weight adjusted k-nearest neighbor classification. In *Proceedings of the 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD'01* (pp. 53–65). London, UK, UK.
- Hand, D. J., & Till, R. J. (2001). A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45, 171–186.
- Hernández-González, J., Inza, I. n., & Lozano, J. A. (2013). Learning bayesian network classifiers from label proportions. *Pattern Recogn.*, 46, 3425–3440.
- Hong, Z., Mei, X., Prokhorov, D., & Tao, D. (2013). Tracking via robust multitask multi-view joint sparse representation. In *International Conference on Computer Vision* ICCV'13 (pp. 649–656). Sydney, Australia.
- Huang, K., Liu, X., Li, X., Liang, J., & He, S. (2013). An improved artificial immune system for seeking the pareto front of land-use allocation problem in large areas. *International Journal of Geographical Information Science*, 27, 922–946.
- Huang, Y.-P., Chang, Y.-T., Hsieh, S.-L., & Sandnes, F. E. (2011). An adaptive knowledge evolution strategy for finding near-optimal solutions of specific problems. *Expert Systems with Applications*, 38, 3806–3818.
- Jiang, L., Cai, Z., Zhang, H., & Wang, D. (2012a). Not so greedy: Randomly selected naive bayes. Expert Systems with Applications, 39, 11022–11028.
- Jiang, L., & Zhang, H. (2005). Learning instance greedily cloning naive bayes for ranking. In *Proceedings of the Fifth IEEE International Conference on Data Mining* ICDM'05 (pp. 202–209). Washington, DC, USA.
- Jiang, L., Zhang, H., & Cai, Z. (2009). A novel bayes model: Hidden naive bayes. *IEEE Transactions on Knowledge and Data Engineering*, 21, 1361– 1371.
- Jiang, L., Zhang, H., Cai, Z., & Wang, D. (2012b). Weighted average of one-dependence estimators?. *Journal of Experimental and Theoretical Artificial Intelligence*, 24, 219–230.
- Kim, S.-B., Han, K.-S., Rim, H.-C., & Myaeng, S.-H. (2006). Some effective techniques for naive bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, 18, 1457–1466.
- Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. In Proceedings of the ninth international workshop on Machine learning (pp. 249–256). San Francisco, CA, USA.
- Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers:a decision-tree hybrid. In *Proceedings of Second International Conference on Knowledge Discovery and Data Mining KDD'96* (pp. 202–207). Protland, Oregon, USA.
- Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. In *Proceedings of the 7th European Conference on Machine Learning* ECML'94 (pp. 171–182). Secaucus, NJ, USA.
- Langley, P., & Sage, S. (1994). Induction of selective bayesian classifiers. In Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence UAI'94 (pp. 339–406). Morgan Kaufmann.
- Li, J., & Wang, J. Z. (2008). Real-time computerized annotation of pictures. IEEE Transactions on Pattern Analysis and Machine Intelligence, 30, 985– 1002.
- Ling, C. X., Huang, J., & Zhang, H. (2003). Auc: A statistically consistent and more discriminating measure than accuracy. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence IJCAI'03* (pp. 519–524). San Francisco, CA, USA.
- Liu, G.-H., & Yang, J.-Y. (2013). Content-based image retrieval using color difference histogram. *Pattern Recogn.*, 46, 188–198.
- Liu, W.-Y., Yue, K., & Li, W.-H. (2011). Constructing the bayesian network structure from dependencies implied in multiple relational schemas. *Expert* Systems with Applications, 38, 7123–7134.

- Luo, Y., Tao, D., Xu, C., Xu, C., Liu, H., & Wen, Y. (2013). Multiview vectorvalued manifold regularization for multilabel image classification. *Neural Networks and Learning Systems, IEEE Transactions on*, 24, 709–722.
- de Mello Honorio, L., Leite da Silva, A., & Barbosa, D. (2012). A cluster and gradient-based artificial immune system applied in optimization scenarios. *IEEE Transactions on Evolutionary Computation*, *16*, 301–318.
- Ortega, M., Rui, Y., Chakrabarti, K., Porkaew, K., Mehrotra, S., & Huang, T. (1998). Supporting ranked boolean similarity queries in mars. *IEEE Transactions on Knowledge and Data Engineering*, 10, 905–925.
- Park, T., & Ryu, K. R. (2010). A dual-population genetic algorithm for adaptive diversity control. *IEEE Transactions on Evolutionary Computation*, 14, 865–884.
- Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Robnik-Šikonja, M., & Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning*, 53, 23–69.
- Storn, R., & Price, K. (1997). Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. *Journal* of Global Optimization, 11, 341–359.
- Tucker, C., Kim, H., Barker, D., & Zhang, Y. (2010). A relieff attribute weighting and x-means clustering methodology for top-down product family optimization. *Engineering Optimization*, 42, 593–616.
- Webb, G. I., Boughton, J. R., Zheng, F., Ting, K. M., & Salem, H. (2012). Learning by extrapolation from marginal to full-multivariate probability distributions: Decreasingly naive bayesian classification. *Machine Learning*, 86, 233–272.
- Witten, I. H., & Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series in Data Management Systems (2nd ed.). San Francisco, CA: Morgan Kaufmann Publishers. URL: http://www.cs.waikato.ac.nz/ml/weka/.
- Woldemariam, K. M., & Yen, G. G. (2010). Vaccine-enhanced artificial immune system for multimodal function optimization. *IEEE Transactions on Systems, Man, and Cybernetics–Part B: Cybernetics*, 40, 218–228.
- Wu, J., & Cai, Z. (2014). Learning attribute weighted aode for roc area ranking. *International Journal of Information and Communication Technology*, 6, 23–38.
- Wu, J., Cai, Z., Zeng, S., & Zhu, X. (2013a). Artificial immune system for attribute weighted naive bayes classification. In *Proceedings of the Inter*national Joint Conference on Neural Networks IJCNN'13 (pp. 798–805). Dallas, TX, USA.
- Wu, J., Cai, Z., & Zhu, X. (2013b). Self-adaptive probability estimation for naive bayes classification. In *Proceedings of the International Joint Confer*ence on Neural Networks IJCNN'13 (pp. 2303–2310). Dallas, TX, USA.
- Wu, J., Hong, Z., Pan, S., Zhu, X., Zhang, C., & Cai, Z. (2014). Multi-graph learning with positive and unlabeled bags. In *Proceedings of SIAM Inter*national Conference on Data Mining SDM'14 (pp. 217–225). Philadelphia, Pennsylvania, USA.
- Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., Zhou, Z.-H., Steinbach, M., Hand, D. J., & Steinberg, D. (2007). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14, 1–37.
- Yuan, J., Zhang, L., Zhao, C., Li, Z., & Zhang, Y. (2012). An improved self-organization antibody network for pattern recognition and its performance study. *Pattern Recognition*, 321, 96–103.
- Zaidi, N. A., Cerquides, J., Carman, M. J., & Webb, G. I. (2013). Alleviating naive bayes attribute independence assumption by attribute weighting. *Journal of Machine Learning Research*, 14, 1947–1988.
- Zhang, C., Xue, G.-R., Yu, Y., & Zha, H. (2009). Web-scale classification with naive bayes. In *Proceedings of the 18th International Conference on World Wide Web* WWW '09 (pp. 1083–1084).
- Zhang, H., & Sheng, S. (2004). Learning weighted naive bayes with accurate ranking. In *Proceedings of the Fourth IEEE International Conference on Data Mining* ICDM'04 (pp. 567–570). Washington, DC, USA.
- Zhang, H., & Su, J. (2004). Naive bayesian classifiers for ranking. In *Proceedings of the 15th European Conference on Machine Learning ECML*'04 (pp. 501–512). Pisa, Italy.
- Zheng, J., Chen, Y., & Zhang, W. (2010). A survey of artificial immune applications. Artificial Intelligence Review, 34, 19–34.
- Zhong, Y., & Zhang, L. (2012). An adaptive artificial immune network for supervised classification of multi-/hyperspectral remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 50, 894–909.