# SEIR Immune Strategy for Instance Weighted Naive Bayes Classification

Shan Xue[1,2,*], Jie Lu[1], Guangquan Zhang[1], and Li Xiong[2]

[1]Lab of Decision Systems & e-Service Intelligence (DeSI)
Centre for Quantum Computation & Intelligent Systems (QCIS)
Faculty of Engineering and Information Technology
University of Technology, Sydney, NSW 2007, Australia
`shan.xue@student.uts.edu.au,{jie.lu,guangquan.zhang}@uts.edu.au`
[2]School of Management, Shanghai University, Shanghai, 200436, China
`xiongli8@shu.edu.cn`

**Abstract.** Naive Bayes (NB) has been popularly applied in many classification tasks. However, in real-world applications, the pronounced advantage of NB is often challenged by insufficient training samples. Specifically, the high variance may occur with respect to the limited number of training samples. The estimated class distribution of a NB classier is inaccurate if the number of training instances is small. To handle this issue, in this paper, we proposed a SEIR (Susceptible, Exposed, Infectious and Recovered) immune-strategy-based instance weighting algorithm for naive Bayes classification, namely SWNB. The immune instance weighting allows the SWNB algorithm adjust itself to the data without explicit specification of functional or distributional forms of the underlying model. Experiments and comparisons on 20 benchmark datasets demonstrated that the proposed SWNB algorithm outperformed existing state-of-the-art instance weighted NB algorithm and other related computational intelligence methods.

**Keywords:** Naive Bayes, Classification, Immune Strategy, SEIR

## 1 Introduction

As a special case of a Bayesian network, Naive Bayes (NB) [3] has been popularly applied in many real-world learning tasks, such as text classification [7][17], web mining [19] and other computational approach [10]. Specifically, the high variance may occur with respect to the limited number of training samples [6][14], where the estimated class distribution of a NB classifier is inaccurate if the number of training instances is small.

To address this research problem, instance weighted naive Bayes (IWNB), as an effective solution, assign different weight values to instances for probability value estimation can improved the performance of NB. For example, an instance-cloned naive Bayes, which produces an expanded training set by cloning some training instances based on their similarities to the test instance is proposed in [5]. Moreover, Jiang [4] proposed to use instance weighting to improve the performance of Averaged One-Dependence Estimators [12], which is another Bayesian model. For this type of weighting method, each training instance is eagerly weighted according to the similarity with the "model" of training dataset. These instance weight setting methods have achieved good performance to solve domain specific problems [15]. However, for all these methods, the instance weights are determined without taking the NB objective function into consideration and the underlying sample distributions should be known in advance for the former approaches.

---

[*] Corresponding author.

In this paper we propose a SEIR Immune based algorithm, which automatically calculates the optimal instance weight values for IWNB, by directly working on IWNB's objective function based on SEIR immune strategy. Specifically, our method uses SEIR procedures to design an automated search strategy to find optimal instance weight for each dataset. The SEIR immune strategy, including initialization, clone, mutation and selection, ensures that our method can adjust itself to the data without any explicit specification of functional or distributional form for the underlying model.

In contrast to the conventional statistical probabilistic evaluation in NB, the SWNB algorithm is a self-learning algorithm by utilizing the immunological properties, such as memory property and clonal selection. The advantages of SWNB can be understood from the following three aspects: SWNB is a data-driven self-adaptive method because it does not requires explicit specification of functional or distributional form for the underlying model. The SWNB algorithm is a nonlinear model and is flexible in modeling complex real world relationships. It inherits the memory property of human immune mechanism and can recognize the same or similar antigen quickly at different times.

The rest of the paper is organized as follows. In Section 2, we present a new SEIR immune strategy and our SWNB algorithm. In Section 3, we describe the experimental conditions, process, and results in details. Section 4 concludes the paper and outlines several directions for future study.

## 2      SEIR Immune Strategy based Instance Weighted Bayes

### 2.1      Instance Weighted Naive Bayes

Given a training set $\mathcal{D} = \{\mathbf{x}_i\}$ with $N$ instances, each instance contains $n$ attribute values and corresponds a class label. We use $\mathbf{x}_i = \{x_{i1}, \cdots, x_{ij}, \cdots, x_{in}, y_i\}$ to stand for the $i$th instance, with $x_{ij}$ denoting the $j$th attribute value and $y_i$ denoting the class label of the instance. Meanwhile, 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 $k$th label of the class space. We use $\mathbf{a} = \{a_j\}$ to include all the attributes of all instances, with $a_j$ representing the $j$th attribute.

For a training instance $\mathbf{x}_i \in \mathcal{D}$ with its class label satisfies $y_i \in \mathcal{Y}$, the training method is based on the IWNB model, which is formally defined as

$$c(\mathbf{x}_i) = \arg\max_{c_k \in \mathcal{Y}} P_{\mathbf{w}}(c_k) \prod_{j=1}^{n} P_{\mathbf{w}}(x_{ij}|c_k). \tag{1}$$

In Eq. (1), $P_{\mathbf{w}}(c_k)$ denotes the probability of class $c_k$ with a certain weights set $\mathbf{w} = \{w_i, i \in [1, \cdots, N]\}$. $P_{\mathbf{w}}(x_{tj}|c_k)$ denotes the joint distribution of $\mathbf{x}_i$ conditioned by the given class $c_k$ based on $\mathbf{w}$.

In this paper, we focus on the calculation of the priori probability $P_{\mathbf{w}}(c_k)$ and the conditional probability $P_{\mathbf{w}}(x_{ij}|c_k)$ by using optimal instance weight value $\mathbf{w}$. The Laplace-estimate instance weighted strategy is introduced and shown in Eq. (2).

$$\begin{cases} P_{\mathbf{w}}(c_k) = \dfrac{n_k + 1}{N_k + L}, \quad P_{\mathbf{w}}(x_{ij}|c_k) = \dfrac{n_k^{ij} + 1}{n_k + |a_j|}, \\ n_k = \displaystyle\sum_{\mathbf{x}_i \in \mathcal{D}, y_i = c_k} w_i, N_k = \sum_{\mathbf{x}_i \in \mathcal{D}} w_i, \quad n_k^{ij} = \sum_{\mathbf{x}_i \in \mathcal{D}, y_j = c_k, x_{pq} \in \mathbf{x}_p \in \mathcal{D}, x_{pq} = x_{ij}} w_p \end{cases}, \tag{2}$$

where $|a_j|$ is the number of distinct values of attribute $a_j$, and $L$ is the number of classes.
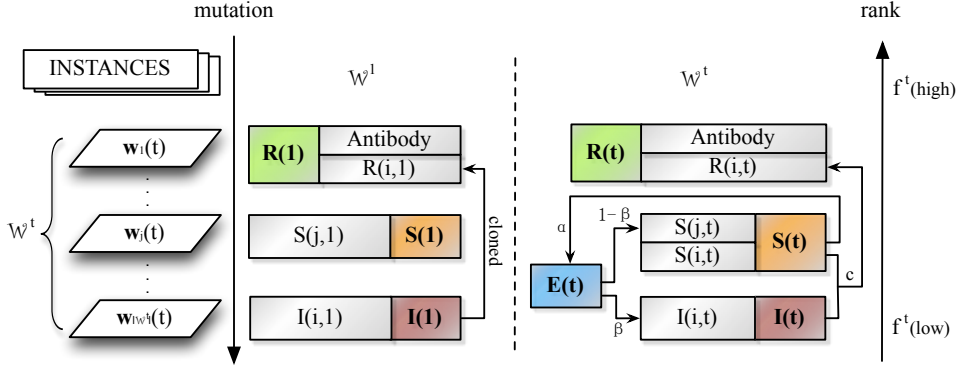
**Fig. 1.** The SEIR immune procedure for IWNB

## 2.2 SEIR Immune Strategy for Instance Weighting

The classic model for microparasite dynamics is the flow of hosts between Susceptible (S), Exposed (E), Infectious (I) and Recovered (R) compartments [20]. However, the SEIR immune strategy could not directly applied for the IWNB classification problem. In this paper, we introduce SEIR model to describe the immune strategy proposed for instance weight (*i.e.,* antibody) optimization. By rating the performance of instance weights, the low scorekeepers are defined in Infectious state $I$, which are planned to be cloned with a set clone factor $c$ and then move to Recovered state $R$. Through mutation, weights are driven into Susceptible state $S$, with a $\alpha$ probability to move into Exposed state $E$, where is in an extremely unstable state causing movements to state $I$ in probability $\beta$, and otherwise kept in state $S$.

Assume in generation $t$, in the proposed SWNB, the number of weight groups in $S$, $E$, $I$ and $R$ are defined as $|(\mathcal{W}^s)^t|$, $|(\mathcal{W}^e)^t|$, $|(\mathcal{W}^i)^t|$ and $|(\mathcal{W}^r)^t|$, we have the SEIR immune procedure in Fig. 1 and corresponding formulations shown in Eq. (3).

$$
\begin{cases}
|\mathcal{W}^t| = |(\mathcal{W}^s)^t| + |(\mathcal{W}^i)^t| + 1, \quad |(\mathcal{W}^r)^t| = 1 + c[|(\mathcal{W}^s)^t| + |(\mathcal{W}^i)^t|], \\
|(\mathcal{W}^e)^t| = \alpha|(\mathcal{W}^s)^t|, \quad |(\mathcal{W}^i)^t| = \alpha\beta|(\mathcal{W}^s)^t| = c[|(\mathcal{W}^s)^t| + |(\mathcal{W}^i)^t|], , \\
|(\mathcal{W}^{s(j)})^t| = \alpha(1-\beta)|(\mathcal{W}^s)^t|, \quad |\mathcal{W}^t| = |(\mathcal{W}^{s(j)})^t| + |(\mathcal{W}^i)^t| + 1
\end{cases}
\tag{3}
$$

where $|\mathcal{W}^t|$ is the total amount of weight groups in the $t$th immune generation. From Eq. (3), we have $\alpha = \frac{1}{1-c}$, $\beta = c$.

## 2.3 SWNB Classifier

In this section, we first introduce some important notations and definitions, then propose our solutions.

**DEFINITION 1 (Calculation of affinity function)** The affinity of the $j$th individual in the $t$th generation $\mathbf{w}_j^t$ is the classification accuracy that is obtained by SWNB using the $\mathbf{w}_j^t$ to carry out the probability estimation. The calculation of affinity function is defined as

$$
f(\mathbf{w}_j^t) = \frac{1}{N^S} \sum_{i=1}^{N^S} s(c^t(\mathbf{x}_i), y_i),
\tag{4}
$$

---

**Algorithm 1** SWNB Classifier

---
**Input:**
   Training dataset $\mathcal{D}$; Clone Factor $c$; Threshold $\varepsilon$; Maximum Evolution Generation $MaxGen$.
**Output:**
   The target class label $c(\mathbf{x}_t)$ of a test instance $\mathbf{x}_t$.
 1: $\mathcal{W}^0 \leftarrow$ Initial population, which is a set of random number distributed between $(0, 1]$.
 2: **while** $t \leq MaxGen$ and $f(\mathbf{w}_c^{t+1}) - f(\mathbf{w}_c^t) \leq \varepsilon$ **do**
 3:     $f(\mathbf{w}_j^t) \leftarrow$ Apply Eq. (4) to calculate affinity of $\mathbf{w}_j^t$ for the $t$th generation.
 4:     $\mathbf{w}_r^t \leftarrow$ The selected antibody with highest $f(\mathbf{w}_j^t)$.
 5:     $\mathcal{W}^t \leftarrow$ The population of the $t$th generation.
 6:     $(\mathcal{W}^i)^t \leftarrow$ The individual sets with low $f(\mathbf{w}_j^t)$ selected as $I(t)$ on clone factor $c$ within $\mathcal{W}^t$ but excluding $\mathbf{w}_r^t$.
 7:     $(\mathcal{W}^r)^t \leftarrow$ The $R(t)$ individuals consist of the antibody and the cloned $(\mathcal{W}^i)^t$ by $\mathbf{w}_r^t$.
 8:     **for all** each $\mathbf{w}_j^t$ in $\mathcal{W}^t$ **do**
 9:         $\mathbf{v}_j^{t+1} \leftarrow$ Apply Eq. (5) to do mutation on each $\mathbf{w}_j^t$.
10:         $\mathbf{w}_j^{t+1} \leftarrow$ Apply Eq. (6) to obtain the new weights in the $t+1$th generation.
11:     **end for**
12:     $(\mathcal{W}^s)^{t+1} \leftarrow$ The $S(t+1)$ individuals.
13:     $(\mathcal{W}^e)^{t+1} \leftarrow$ The $E(t+1)$ individuals contribute to $(\mathcal{W}^i)^{t+1}$ on $\beta$ and $(\mathcal{W}^{s(j)})^{t+1}$ on $1 - \beta$.
14:     $\mathcal{W}^{t+1} \leftarrow (\mathcal{W}^s)^{t+1} \cup (\mathcal{W}^e)^{t+1} \cup (\mathcal{W}^i)^{t+1} \cup (\mathcal{W}^r)^{t+1}$.
15: **end while**
16: $c(\mathbf{x}_t) \leftarrow$ Apply $\mathbf{w}_t$ to instance $\mathbf{x}_t$ to predict the underlying class label via Eqs. (1) and (2).

---

where $c^t(\mathbf{x}_i)$ is the classification result of each $i$th instance in the $t$th training generation using the SWNB classifier based on each individual $\mathbf{w}_j^t$, $i \in [1, \cdots, N]$. $s(c^t(\mathbf{x}_i), y_i)$ defines the similarity between $c^t(\mathbf{x}_i)$ and $y_i$, where $s(c^t(\mathbf{x}_i), y_i)$ is 1 if $c^t(\mathbf{x}_i) = y_i$ and 0 otherwise.

**DEFINITION 2 (Antibody clone)** We select $\mathbf{w}_r^t$ as the antibody of the $t$th generation with the best affinity performance of $f(\mathbf{w}_j^t)$ sorting from $\mathcal{W}^t$. After that, we use the antibody to replace the weight groups $(\mathcal{W}^i)^t$ with low affinity according to the same rate of a set clone factor $c$. As a result, $\mathbf{w}_i^t \in (\mathcal{W}^i)^t$ are cloned by $\mathbf{w}_r^t$ and move to $(\mathcal{W}^r)^t$.

**DEFINITION 3 (Antibody mutation)** The mutate operation is used to treat all individuals in the $t$th population $\mathcal{W}^t$ and for training preparation on $\mathcal{W}^{t+1}$ in the $t+1$th generation. For any individual $\mathbf{w}_j^t$, the new variation $\mathbf{v}_j^{t+1}$ is generated as

$$\mathbf{v}_j^{t+1} = \mathbf{w}_j^t + [1 - f(\mathbf{w}_j^t)] * N(0, 1) * (\mathbf{w}_r^t - \mathbf{w}_j^t), \qquad (5)$$

where $N(0, 1)$ is a normally distributed random variable within the range of $[0, 1]$, and

$$\mathbf{w}_j^{t+1} = \mathbf{v}_j^{t+1}, \ f(\mathbf{v}_j^{t+1}) > f(\mathbf{w}_j^t); \quad or \quad \mathbf{w}_j^t, \quad f(\mathbf{v}_j^{t+1}) \leq f(\mathbf{w}_j^t). \qquad (6)$$

## 3   Experiments

### 3.1   Experimental Conditions and Baselines

We validate the performance of the proposed method on 20 benchmark datasets from UCI data repository [2]. Because naive Bayes based classifiers are designed for categorical attributes, in our experiments, we first replace all missing attribute values using unsupervised attribute filter *ReplaceMissingValues* in WEKA [13]. Then, we apply unsupervised filter *Discretize* in WEKA to discretize numeric attributes into nominal attributes. In our experiments, the algorithms are

**Table 1.** Experimental results for SWNB vs. baselines: classification accuracy %

| Dataset | **SWNB** | IWNB [4] | NB [3] | SBC [9] | C4.4 [11] | KNN [1] |
|---|---|---|---|---|---|---|
| anneal | **97.04** | 95.93 | 93.70 | 91.48 | 80.37 | 92.96 |
| anneal.ORIG | **91.11** | 88.52 | 90.37 | 84.07 | 88.89 | 84.07 |
| balance-scale | **92.91** | 89.84 | 89.84 | 89.84 | 64.17 | 83.42 |
| breast-cancer | **81.76** | 71.76 | 70.59 | 70.59 | 69.41 | 75.29 |
| colic.ORIG | **84.09** | 74.55 | 79.09 | 71.82 | 73.64 | 70.00 |
| credit-a | **88.54** | 84.54 | 84.54 | 85.02 | 85.02 | 84.54 |
| credit-g | **78.67** | 74.67 | 74.33 | 72.00 | 70.00 | 72.00 |
| diabetes | **85.65** | 76.09 | 77.39 | 76.96 | 71.30 | 70.87 |
| heart-c | **92.42** | 86.81 | 86.81 | 85.71 | 80.22 | 82.42 |
| heart-h | **93.76** | 91.01 | 86.52 | 85.39 | 78.65 | 85.39 |
| heart-statlog | **83.95** | 83.95 | 83.95 | 82.72 | 71.60 | 81.48 |
| hepatitis | **91.11** | 85.11 | 89.36 | 78.72 | 80.85 | 82.98 |
| ionosphere | **88.67** | 86.67 | 88.57 | 84.76 | 83.81 | 88.57 |
| kr-vs-kp | **98.67** | 81.67 | 81.67 | 94.00 | 93.00 | 88.00 |
| labor | **94.12** | 88.24 | 87.34 | 88.24 | 88.24 | 82.35 |
| letter | **69.56** | 61.00 | 57.67 | 60.67 | 50.00 | 45.67 |
| lymph | **88.64** | 84.09 | 84.09 | 81.82 | 79.55 | 79.55 |
| segment | **89.74** | 87.42 | 84.11 | 85.43 | 83.44 | 80.46 |
| soybean | **93.20** | 92.68 | 91.71 | 88.78 | 89.76 | 87.80 |
| waveform | **85.67** | 79.67 | 80.33 | 80.33 | 65.33 | 72.00 |

evaluated in terms of classification accuracy via 10 runs of 10-fold cross validation. Besides, the three parameters maximum iteration $MaxGen$, threshold $\varepsilon$, and the clone factor $c$ in Algorithm 1 are set to 50, 0.001 and 0.1 respectively. Moreover, all experiments are conducted on a Linux cluster node with an Intel(R) Xeon(R) @3.33GHZ CPU and 3GB fixed memory size.

For comparison purposes, we use the following baseline algorithms in our experiments.

1) $NB$: The standard naive Bayes classifier with conditional attribute independence assumption [3].
2) $IWNB$: Instance weighted naive Bayes with the weighting method based on the instance similarity [4].
3) $SBC$: A bagged decision-tree based attribute selection filter for naive Bayes [9].
4) $C4.4$: A specially designed tree to improve C4.5 performance [11] on classification ranking.
5) $KNN$: The k-Nearest Neighbors algorithm [1] with $k$ value been set to 10.

### 3.2   UCI Standard Classification Task

The initial important task is to analyze the performance between the IWNB with the related instance weighting strategy in literature and NB, in terms of classification accuracy, which is calculated by the percentage of successful predictions on domain specific problems [16][18]. Besides, some other types of algorithms that have been well used in real-world applications have also been used for comparison. Specifically, we compare the effect of IWNB [4] with the standard NB [3], SBC [9], C4.4 [11], and KNN [1]. The purpose of the second experiment is to compare the proposed self-adaptive instance weighted Naive Bayes, namely SWNB, with each other types of baseline approaches in literature.

**Instance Weighted NB vs. Standard NB** Table 1 reports the detailed results of SWNB and other baseline algorithms, respectively. Besides, Table 2 illustrates the compared results about

**Table 2.** Winning or losing statistical analysis on 20 UCI datasets

|            | KNN [1]  | C4.4 [11] | SBC [9]  | NB [3]   | IWNB [4] |
|------------|----------|-----------|----------|----------|----------|
| C4.4 [11]  | 9/1/10   |           |          |          |          |
| SBC [9]    | 13/3/4   | 14/2/4    |          |          |          |
| NB [3]     | 16/2/2   | 17/0/3    | 12/3/5   |          |          |
| IWNB [4]   | 16/1/3   | 16/1/3    | 14/2/4   | 8/6/6    |          |
| **SWNB**   | **20/0/0** | **20/0/0** | **20/0/0** | **19/1/0** | **19/1/0** |

∗ All analyses are under two-tailed $t$-test with a 95% confidence level.

the winning-or-losing statistical analysis (*i.e.,* two-tailed $t$-test with a 95% confidence level) on those benchmark datasets. Based on the statistical theory, the difference is statistically significant only if the probability of significant difference is at least 95 percent, *i.e.,* the $p$-value for a $t$-test between two algorithms is less than 0.05. In Table 2, each entry $w/t/l$ means that the algorithm in the corresponding row wins in $w$ datasets, ties in $t$ datasets and loses in $l$ datasets on the 20 UCI datasets, compared to the algorithm in the corresponding column. Overall, the results can be summarized as follows:

1) Instance weighting IWNB outperforms NB (8 wins and 6 losses). In particular, for the dataset "letter" with 20000 instances and 26 classes, the classification accuracy for IWNB (61.00%) is higher than NB (57.67%). Because the 26 classes make the classification task particularly difficult, the 3.33% superiority on 20000 samples become significant.
2) Instance weighting IWNB greatly outperforms SBC with (14 wins and 4 losses) on the 20 UCI benchmark datasets.
3) Instance weighting IWNB significantly outperforms decision tree C4.4 and lazy learning approach KNN both with (16 wins and 3 losses).

**SWNB vs. Baselines** Our experimental results from Tables 1 and 2 indicate that SWNB has very significant gain compared to the state-of-art instance weighting strategy and other types of methods. In summary, our experimental results can be listed as:

1) SWNB significantly outperforms existing instance weighted IWNB and unweighted NB both with (19 wins and 0 losses). For the dataset "letter" with 26 classes, the SWNB can achieve a high classification accuracy (69.56%), which is 8.56% and 11.89% higher than IWNB and NB, respectively.
2) SWNB greatly outperforms selected naive Bayes SBC with (20 wins and 0 losses) on the 20 UCI benchmark datasets.
3) SWNB also significantly outperforms decision tree C4.4 and lazy learning approach KNN both with (20 wins and 0 losses).

### 3.3   Convergence and Learning Curves

In order to investigate the convergence of the SWNB algorithm, we report the relationship between the number of iterations and error rate on the 9 datasets, and the results are shown in Fig. 2. Each point in the curves corresponds to the accuracy under the underlying iteration with the current optimal instance weight values. Fig. 2 shows that SWNB converges quickly. Although the curves are not quite smooth, they converge well, which accords with the immunization strategy in SWNB.
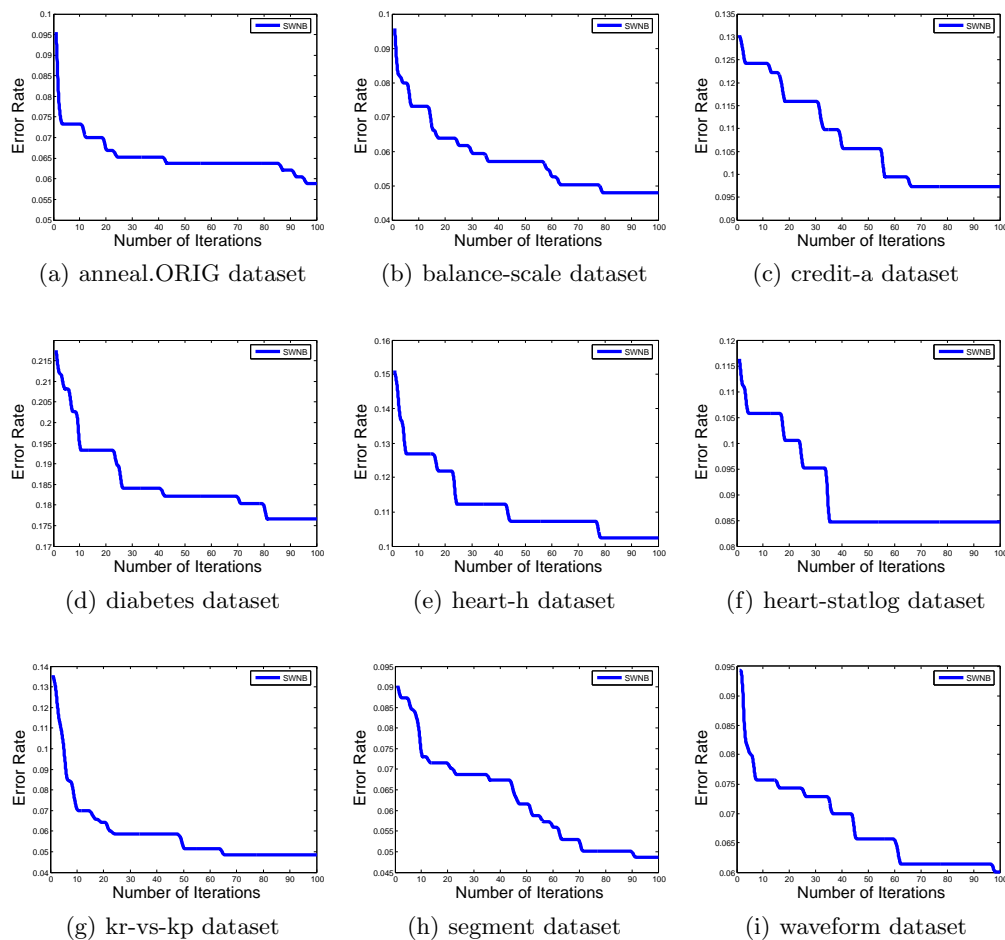
**Fig. 2.** Convergence learning curves by error rate of SWNB for 9 datasets

For additional insight into our experiment, we observe the "kr-vs-kp" for example. It is a high-dimensional dataset (37 attributes) with 3196 instances. In addition, strong attribute dependencies have been found in this dataset by Kohavi [8]. Our results show that SWNB achieves 98.67% classification accuracy, which is significantly higher accuracy than instance weighted and unweighted NB on the same dataset (81.67%). The accuracy of the final convergence is also much better than selected naive Bayes SBC (94.00%), decision tree C4.4 (93.00%), and lazy learning KNN (88.00%). Similar levels of improvement can also be observed from other datasets.

## 4   Conclusion and Future Work

In this paper, we proposed a novel algorithm to train weighted instances for naive Bayes classification, namely SWNB, by extending the classical SEIR immune strategy. The SWNB algorithm calculates the probability values by using the adaptively instance weighting approach. Considering real-world applicabilities, experiments and comparisons taken on the 20 benchmark UCI datasets, with respect to the classification accuracy performance, show that SWNB outperforms

existing NB instance weighting models and other related algorithms, such as classification ranking tree, k-Nearest Neighbors, etc.

The proposed immune strategy based instance weighting for naive Bayes can also be extended to Bayesian networks and applied to other dynamic social networks. Our further study will focus on dynamic Bayesian network applications.

# References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. 6(1), 37–66 (1991)
2. Bache, K., Lichman, M.: UCI machine learning repository (2013), `http://archive.ics.uci.edu/ml`
3. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Mach. Learn. 29(2-3), 131–163 (1997)
4. Jiang, L., Cai, Z., Wang, D.: Learning averaged one-dependence estimators by instance weighting. Journal of Computational Information Systems 4, 2753–2760 (2008)
5. Jiang, L., Zhang, H.: Learning instance greedily cloning naive bayes for ranking. In: Proc. of ICDM. pp. 202–209 (2005)
6. Jiang, L., Zhang, H., Cai, Z.: A novel bayes model: Hidden naive bayes. IEEE Trans. on Knowl. and Data Eng. 21(10), 1361–1371 (2009)
7. Kim, S.B., Han, K.S., Rim, H.C., Myaeng, S.H.: Some effective techniques for naive bayes text classification. Knowledge and Data Engineering, IEEE Transactions on 18(11), 1457–1466 (2006)
8. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers:a decision-tree hybrid. In: Proc. of KDD. pp. 202–207 (1996)
9. Langley, P., Sage, S.: Induction of selective bayesian classifiers. In: Proc. of UAI. pp. 339–406 (1994)
10. Naderpour, M., Lu, J., Zhang, G.: A fuzzy dynamic bayesian network-based situation assessment approach. In: Proc. of IEEE FUZZ. pp. 1–8 (2013)
11. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1993)
12. Webb, G.I., Boughton, J.R., Wang, Z.: Not so naive bayes: Aggregating one-dependence estimators. Mach. Learn. 58(1), 5–24 (2005)
13. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques. The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann Publishers, San Francisco, CA, 2nd edn. (2005), `http://www.cs.waikato.ac.nz/ml/weka/`
14. Wu, J., Cai, Z.: A naive bayes probability estimation model based on self-adaptive differential evolution. Journal of Intelligent Information Systems 42(3), 671–694 (2014)
15. Wu, J., Pan, S., Cai, Z., Zhu, X., Zhang, C.: Dual instance and attribute weighting for naive bayes classification. In: Neural Networks (IJCNN), 2014 International Joint Conference on. pp. 1675–1679. IEEE (2014)
16. Wu, J., Pan, S., Zhu, X., Cai, Z.: Boosting for multi-graph classification. Cybernetics, IEEE Transactions on 45(3), 430–443 (2015)
17. Wu, J., Pan, S., Zhu, X., Cai, Z., Zhang, P., Zhang, C.: Self-adaptive attribute weighting for naive bayes classification. Expert Systems with Applications 42(3), 1487–1502 (2015)
18. Wu, J., Zhu, X., Zhang, C., Yu, P.S.: Bag constrained structure pattern mining for multi-graph classification. Knowledge and Data Engineering, IEEE Transactions on 26(10), 2382–2396 (2014)
19. Zhang, C., Xue, G.R., Yu, Y., Zha, H.: Web-scale classification with naive bayes. In: Proc. of WWW. pp. 1083–1084 (2009)
20. Zhang, T., Liu, J., Teng, Z.: Existence of positive periodic solutions of an seir model with periodic coefficients. Applications of Mathematics 57(6), 601–616 (2012)