Parameter Optimization of Local-Concentration Model for Spam Detection by Using Fireworks Algorithm

Wenrui He, Guyue Mi, and Ying Tan

Key Laboratory of Machine Perception (Ministry of Education),
Department of Machine Intelligence, School of Electronics Engineering and Computer Science,
Peking University, Beijing, 100871 P.R. China
{wenrui.he,gymi,ytan}@pku.edu.cn

Abstract. This paper proposes a new framework that optimizes anti-spam model with heuristic swarm intelligence optimization algorithms, and this framework could integrate various classifiers and feature extraction methods. In this framework, a swarm intelligence algorithm is utilized to optimize a parameter vector, which is composed of parameters of a feature extraction method and parameters of a classifier, considering the spam detection problem as an optimization process which aims to achieve the lowest error rate. Also, 2 experimental strategies were designed to objectively reflect the performance of the framework. Then, experiments were conducted, using the Fireworks Algorithm (FWA) as the swarm intelligence algorithm, the Local Concentration (LC) approach as the feature extraction method, and SVM as the classifier. Experimental results demonstrate that the framework improves the performance on the corpora PU1, PU2, PU3 and PUA, while the computational efficiency is applicable in real world.

Keywords: Spam Detection, Fireworks Algorithm, Parameter Optimization, Local Concentration Approach.

1 Introduction

Spam, defined as Unsolicited Commercial E-mails (UCE) or Unsolicited Bulk E-mails (UBE), has become a significant problem for both recipients and Internet Service Providers (ISPs). For recipients, coping with spam is time-consuming; furthermore, spam frequently contains images that recipients find offensive, or attached malicious programs that attack recipients' computers. For ISPs, large scale of spam is a considerable burden on their systems. Commtouch reported that in Q4 2012, the average daily spam level was 90 billion messages per day, which is a slight increase over Q3 2012. [1] Ferris Research revealed that spam cost $130 billion worldwide in 2009, which was a 30% raise over the 2007 estimates. [2] Therefore, it is necessary to find an effective method for the spam detection.

Many approaches were proposed to handle the problem. In fact, Spam detection involves mainly three research fields, namely term selection, feature extraction, and classifier design. In the classifier design field, many machine learning (ML) methods

Y. Tan, Y. Shi, and H. Mo (Eds.): ICSI 2013, Part I, LNCS 7928, pp. 439–450, 2013.

were adopted to classify emails, such as Support Vector Machine (SVM) [3]–[6], k-Nearest Neighbor (k-NN), Naive Bayes (NB), Artificial Neural Network (ANN), Boosting, and Artificial Immune System (AIS). As the performance of an ML method depends on the extraction of discriminative feature vectors, feature extraction methods are crucial to the process of spam filtering. Commonly used feature extraction methods are, for example, Concentration-based Feature Construction (CFC) [4], Local concentration (LC) [7] and Bag-of-Words (BoW). The researches of term selection have also attracted much attention from researchers all over the world, widely utilized methods including Information Gain (IG) [8], Term Frequency Variance (TFV) [9] and Document Frequency (DF).

In previous research, parameters in the anti-spam process are set simply and manually. However, the manual setting might cause several problems. For instance, lack of prior knowledge may lead to improper parameter setting, repeated attempts of users cost overmuch human effort, and the inflexibility of the dataset-relevant parameters should also be taken into counted.

To solve the problems, this paper proposes a new framework that automatically optimizes parameters in anti-spam model with heuristic swarm intelligence optimization algorithms, and this framework could integrate various classifiers and feature extraction methods. 2 experimental strategies were designed to objectively reflect framework performance. Then, experiments are conducted, using the Fireworks Algorithm (FWA) as the Swarm Intelligence algorithm, the Local Concentration approach as the feature extraction method, and SVM as the classifier. Experimental results demonstrate that the framework improved the performance on the corpora PU1, PU2, PU3 and PUA, and the computational efficiency is applicable in real world.

The remainder of the paper proceeds as follows. To begin with, we will provide a brief background on the LC approach and the FWA in Section II. The proposed framework for anti-spam is presented in Section III. In Section IV, the corpora, the criteria and the experimental setup are described, and experiments results are analyzed in detail. Section V concludes the paper.

2 Related Works

2.1 Local Concentration (LC) Based Feature Extraction Approach for Anti-spam

In an anti-spam model, feature extraction is an essential step. The feature extraction method decides spatial distribution characteristics of email sample points, influencing construction of a specific email classification model and the final classification performance. An effective feature extraction method is able to extract extinguishing features of emails, endowing different kinds of emails possessing obvious spatial distribution difference. Moreover, it should be capable of reducing the complexity and difficulty of classification, so as to improve overall performance of the anti-spam model. The Local-concentration (LC) approach is proved to meet both of the requirements mentioned above. It not only greatly reduces feature dimensionality by

remaining the position-correlated information of emails, but also performs better in terms of both accuracy and measure compared to the BoW approach and the GC approach.

Inspired from the biological immune system, the LC feature extraction approach is able to extract position-correlated information from messages by transforming each area of a message to a corresponding LC feature effectively. Two implementation strategies of the LC approach were designed by using a fixed-length sliding window and a variable-length sliding window. To incorporate the LC approach into the whole process of spam filtering, a generic LC model is designed. In the LC model, two types of detector sets are at first generated by using term selection methods and a well-defined tendency threshold. Then a sliding window is adopted to divide the message into individual areas. After segmentation of the message, the concentration of detectors is calculated and taken as the feature for each local area. Finally, all the features of local areas are combined as a feature vector of the message.

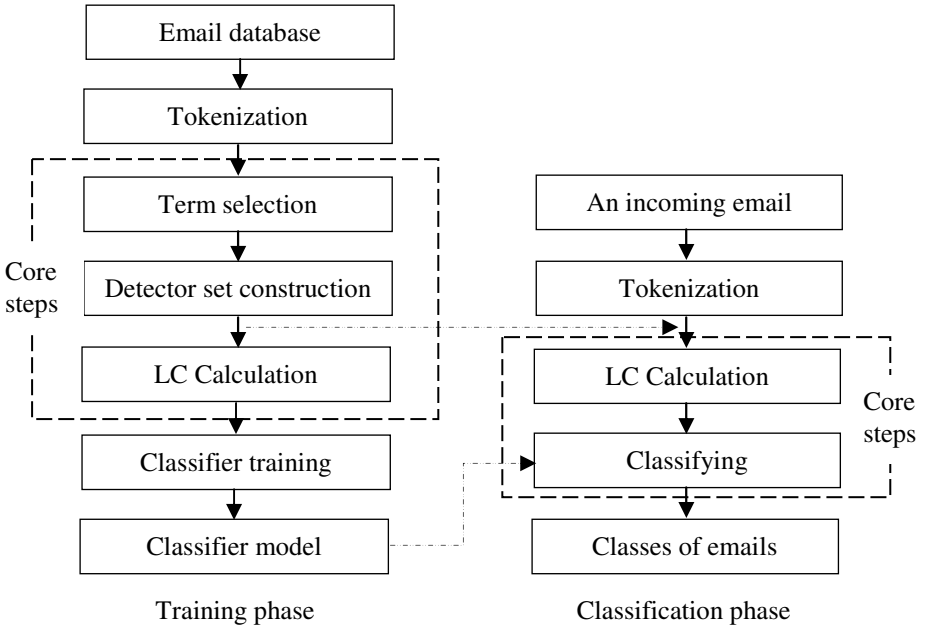

Fig. 1. Training and classification phases of the LC model

The generic structure of the LC model is shown in Fig. 1. The tokenization is a simple step, where messages are tokenized into words (terms), while term selection, detector set construction and LC calculation are quite essential to the model.

In the term selection step, terms are sorted in the order of importance and the top m% of the terms are selected to form the gene library. The term selection rate parameter, m%, decides the size of the gene library, influencing the computational complexity of the detector construction algorithm and distinguishability of detectors

in the next step. An optimal value of m% is supposed to effectively screen out noise terms, while guarantee the existence of the informative terms.

In the detector construction step, the tendency of each detector, namely the difference between a term's posterior probability of presence in normal emails and that in spams, is calculated. If the tendency of a term exceeds θ, the term will be added into the detector set. This parameter, θ, as the standard of detector set construction, is capable of controlling significance of detector matching, yet can't be set too high, so as not to cause loss of information.

In the LC calculation step, the number of sliding windows, N, no matter in fix-length LC approach or in variable-length LC approach, is an important parameter, since it decides the size of a single sliding window and defines the local region,. As a result, it has a great impact on the dimensionality of LC feature vectors, and also performance of the algorithm.

The above three parameters, as well as the parameters of classifiers in the classification step, are fairly essential in LC approach. They, as a whole, heavily influence the performance of the anti-spam model.

In the previous research, these parameters in LC approach were set simply and manually. However, the manual setting might cause several problems. For instance, lack of prior knowledge may lead to improper parameter setting, repeated attempts of users cost overmuch human effort, and the inflexibility of the dataset-relevant parameters should also be taken into counted. To solve these difficulties, a parameter-optimized LC approach using Fireworks Algorithm is proposed in this paper.

2.2 Fireworks Algorithm

In recent years, swarm intelligence (SI) algorithms have been popular among researchers who are working on optimization problems. SI algorithms, e.g. Fireworks Algorithm (FWA) [10], Particle Swarm Optimization (PSO), Ant System, Clonal Selection Algorithm, and Swarm Robots, etc., have advantages in solving many optimization problems. Among all the SI algorithms, FWA is one of the most popular algorithms for searching optimal locations in a D-dimensional space.

Like most swarm intelligence algorithms, FWA is inspired by some intelligent colony behaviors in nature. Specifically, the framework of FWA is mimicking the process of setting off fireworks. The explosion process of a firework can be viewed as a search in the local space around a specific point where the parent firework is set off through the offspring sparks generated in the explosion.

Assume the population size of fireworks is N and the population size of generated spark is M. Each fireworks $i(i = 1, 2, \cdots, N)$ in a population has the following properties: a current position x_i, a current explosion amplitude A_i and the amount of the generated sparks s_i. Each firework generates a number of sparks within a fixed explosion amplitude. In each generation, N fireworks set off within a feasible bounds within explosion amplitude A_i and spark size s_i, then the spark are generated. In addition, the fireworks algorithm also takes Gaussian mutation operators to enhance local search capability.

The best firework is kept for the next generation, and the other $N-1$ fireworks for the next generation are selected based on their distance to other fireworks or randomly as to keep the diversity in the set, which includes the N fireworks, the generated sparks and Gaussian mutation fireworks. The fireworks algorithm continues conducting these operations till the termination criteria is satisfied.

As to the optimization problem f, a point with better fitness is considered as a potential solution, which the optima locate nearby with high chance, vice versa. Suppose FWA is utilized to solve a general optimization problem:

$$Minimize\ f(x) \in R, \qquad x \in R^n \tag{1}$$

where $\mathrm{x} = x_1, x_2, \cdots, x_d$ denotes a location in the potential space, $f(x)$ is an objective function, and R^n denotes the potential space. Then the FWA is implemented to find a point $x \in R^n$, which has the minimal fitness value. This is also how the optimization of the anti-spam process is implemented.

3 Parameter Optimization of Local-Concentration Model for Spam Detection by Using Fireworks Algorithm

The classification problem that whether an email is spam or a normal email, is here considered as an optimization problem, that is, to achieve the lowest error rate by finding the optimal parameter vector in the potential search space.

The optimal vector $P^* =< {F_1}^*, {F_2}^*, \cdots, {F_n}^*, {C_1}^*, {C_2}^*, \cdots, {C_m}^* >$, composes of 2 parts: the first part is the feature calculation relevant parameters ${F_1}^*, {F_2}^*, \cdots, {F_n}^*$, and the second part is the classifier relevant parameters ${C_1}^*, {C_2}^*, \cdots, {C_m}^*$. The optimal vector P^* is the vector whose cost function $CF(P)$ associated with classification achieves the lowest value, with

$$CF(P) = Err(P) \tag{2}$$

where $Err(P)$ is the classification error measured by 10-fold cross validation on the training set. Input vector P consists of two parts – parameters ${F_1}^*, {F_2}^*, \cdots, {F_n}^*$ associated with a certain feature extraction method and ${C_1}^*, {C_2}^*, \cdots, {C_m}^*$ associated with a certain classifier. ${F_1}^*, {F_2}^*, \cdots, {F_n}^*$ uniquely determine the performance of feature construction, while ${C_1}^*, {C_2}^*, \cdots, {C_m}^*$ influence the performance of a certain classifier. Different feature extraction methods hold different parameters and lead to different performance. For LC approach, specifically, m, the Term Selection Rate, helps select the top m % terms with descending importance in term set, which determines the term pool size. θ, the Proclivity Threshold, the minimal difference of a term's frequency in non-spam e-mails minus that in spam e-mails, has an assistant function in screening out terms with greater discrimination. N, the number of sliding windows, determines the dimensionality of the feature vector of emails. Different classifiers hold different parameters and also lead to different performance. Parameters associated with neural network, which determine the structure of the network, include number of layers, number of nodes within a layer and each connection weight between two nodes. SVM-related parameters that determine the

position of optimal hyper-plane in feature space, include cost parameter C and kernel parameters, just to name a few.

The vector P is the optimization objective whose performance is measured by $CF(P)$. Therefore, the optimization of concentrations can be formulated as follows.

Finding $P^* = < {F_1}^*, {F_2}^*, \cdots, {F_n}^*, {C_1}^*, {C_2}^*, \cdots, {C_m}^* >$, so that

$$CF(P^*) = \min_{\{F_1, F_2, \cdots, F_m, C_1, C_2, \cdots, C_m\}} CF(P) \tag{3}$$

Several optimization approaches not demanding an analytical expression of the objective function such as particle swarm optimization (PSO), genetic algorithms (GA) and so forth can be employed for the optimization process. Fireworks Algorithm was used to design concentrations.

Figure 2 shows the optimization process of Parameter Optimization of Local-Concentration Model for Spam Detection Using Fireworks Algorithm.

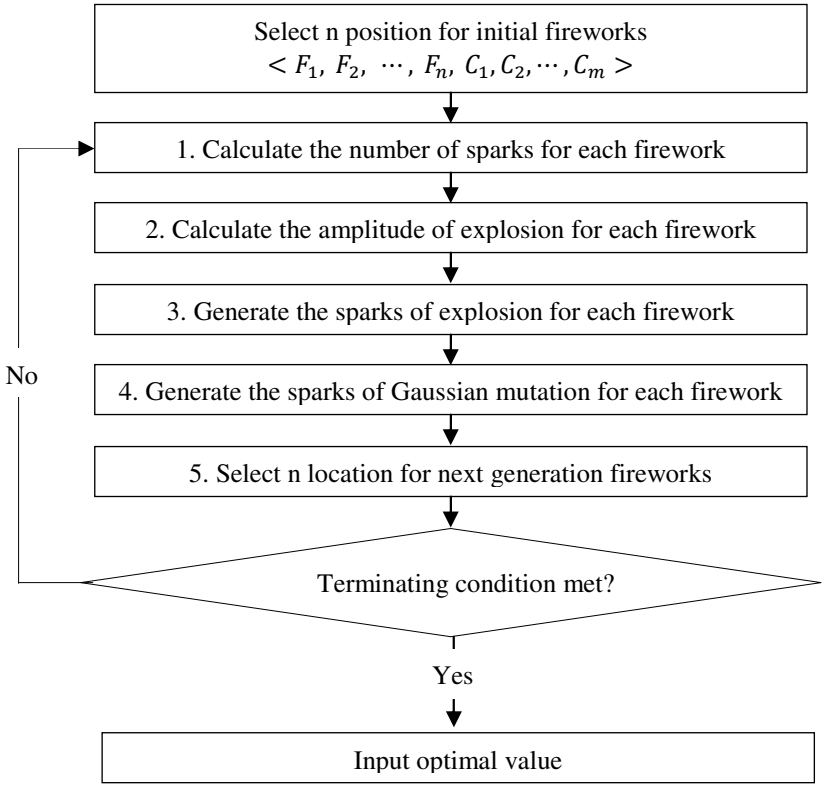

Fig. 2. Process of the Parameter Optimization of Local-Concentration Model for Spam Detection by Using Fireworks Algorithm

This framework utilizes the Fireworks Algorithm to optimize parameters in the Local Concentration approach. Not only the essential parameters in the LC approach, but also the classifier-relevant parameters are optimized in this framework, so that the whole anti-spam process gets optimized.

This framework optimizes anti-spam model with heuristic Swarm Intelligence optimization algorithms, which could integrate various classifiers and feature extraction methods.

4 Experiments

4.1 Experimental Corpora

The experiments were conducted on four benchmark corpora PU1, PU2, PU3, and PUA, using 10-fold cross validation. The corpora have been preprocessed with removal of attachments, HTML tags, and header fields except for the subject. The duplicates were removed from the corpora in that they may lead to over-optimistic conclusions in experiments. In PU1 and PU2, only the duplicate spam, which arrived on the same day, are deleted. While in PU3 and PUA, all duplicates (both spam and legitimate e-mail) are removed, even if they arrived on different days. Different from the former PU1 corpus (the one released in 2000), the corpora are not processed with removal of stop words, and no lemmatization method is adopted. The details of the corpora are given as follows.

1) **PU1:** The corpus includes 1099 messages, 481 messages of which are spam. The ratio of legitimate e-mail to spam is 1.28. The preprocessed legitimate messages and spam are all English messages, received over 36 months and 22 months, respectively.
2) **PU2:** The corpus includes 721 messages, 142 messages of which are spam. The ratio of legitimate e-mail to spam is 4.01. Similar to PU1, the preprocessed legitimate messages and spam are all English messages, received for over 22 months.
3) **PU3:** The corpus includes 4139 messages, 1826 messages of which are spam. The ratio of legitimate e-mail to spam is1.27. Unlike PU1 and PU2, the legitimate messages contain both English and non-English ones. While spam are derived from PU1, Spam Assassin corpus and other sources.
4) **PUA:** The corpus includes 1142 messages, 572 messages of which are spam. The ratio of legitimate e-mail to spam is 1. Similar to PU3, the legitimate e-mail contain both English and non-English messages, and spam is also derived from the same sources.

4.2 Evaluation Criteria

In spam filtering, many evaluation methods or criteria have been designed for comparing performance of different algorithms [12], [13]. We adopted four evaluation criteria, which were spam recall, spam precision, accuracy, and F_β measure, in all our experiments to do a before-and-after comparison. Among the criteria, accuracy and F_β measure are more important, for accuracy measures the total number of messages correctly classified, and F_β is a combination of spam recall and spam precision.

1) **Spam recall:** It measures the percentage of spam that can be filtered by an algorithm or model. High spam recall ensures that the filter can protect the users from spam effectively. It is defined as follows:

$$R_S = \frac{n_{s\to s}}{n_{s\to s} + n_{s\to l}} \tag{4}$$

where $n_{s\to s}$ is the number of spam correctly classified, and $n_{s\to l}$ is the number of spam mistakenly classified as legitimate e-mail.

2) **Spam precision:** It measures how many messages, classified as spam, are truly spam. This also reflects the amount of legitimate e-mail mistakenly classified as spam. The higher the spam precision is, the fewer legitimate e-mail have been mistakenly filtered. It is defined as follows:

$$P_S = \frac{n_{s\to s}}{n_{s\to s} + n_{l\to s}} \tag{5}$$

where $n_{l\to s}$ is the number of legitimate e-mail mistakenly classified as spam, and $n_{s\to s}$ has the same definition as in (4).

3) **Accuracy:** To some extent, it can reflect the overall erformance of filters. It measures the percentage of messages (including both spam and legitimate e-mail) correctly classified. It is defined as follows:

$$\mathrm{A} = \frac{n_{l\to l} + n_{s\to s}}{n_l + n_s} \tag{6}$$

where $n_{l\to l}$ is the number of legitimate e-mail correctly classified, $n_{s\to s}$ has the same definition as in (4), and n_l and n_s are, respectively, the number of legitimate e-mail and the number of spam in the corpus.

4) $\boldsymbol{F_\beta}$ **measure**: It is a combination of R_s and P_s, assigning a weight β to P_s. It reflects the overall performance in another aspect. F_β measure is defined as follows:

$$F_\beta = (1 + \beta^3)\frac{R_s + P_s}{\beta^2 P_s + R_s} \tag{7}$$

In our experiments, we adopted $\beta = 1$ as done in most approaches [12]. In this case, it is referred to as F_1 measure. In the experiments, the values of the four measures were all calculated. However, only accuracy and F_1 measure are used for parameter selection and comparison of different approaches. Because they can reflect overall performance of different approaches, and F_1 combines both R_s and P_s. In addition, R_s and P_s, respectively, reflect different aspects of the performance, and they cannot reflect the overall performances of approaches, separately. That is also the reason why the F_β is proposed. We calculated them just to show the components of F_1 in detail.

4.3 Experimental Setup

All the experiments were conducted on a PC with Intel Core i5-2300 CPU and 4G RAM. The LC-based model with variable-length sliding window was optimized and

the term selection method utilized was information gain. SVM was employed as classifier and LIBSVM was applied for the implementation of the SVM.10-fold cross validation was utilized on each corpora. Since FWA is a stochastic algorithm, the experimental results we present are average results under ten independent runs. Accuracy, recall, precision and F1 measure were selected as evaluation criteria, in which accuracy and F1 measure are main ones since they can reflect the overall performance of spam filtering.

4.4 Experimental Results and Analysis

Two strategies for experiments were designed to investigate the effectiveness of the proposed optimization process of LC model. In both strategies, optimization of the LC model is conducted on the training set and finally examined on the testing set in each fold. In this case, the original training set is further divided into a new training set and a testing set for computing the fitness to evaluate the LC model that the current spark is corresponding to.

For the consideration of efficiency, the first strategy (strategy-1) is designed by defining a validation set on the original training set and making it independent from the original training set, e.g. the original training set is divided into a new training set and a validation set. The fitness of each spark is independently computed on the validation set after a corresponding classifier is trained on the new training set. The optimal model that corresponding to the optimal spark achieved and trained on the new training set is finally examined on the testing set in each fold. In this strategy, fitness of each spark is evaluated on an independent validation set in each fold, thus the computational complexity is relatively low and the optimization process of the LC model could be finished quickly.

Table 1. Performance comparison of LC before and after optimization with strategy-1

Corpus	Approach	Precision (%)	Recall (%)	Accuracy (%)	F1 (%)
PU1	LC	94.85	95.63	95.87	95.21
	Strategy-1	96.55	95.21	**96.33**	**95.81**
PU2	LC	95.74	77.86	94.79	85.16
	Strategy-1	95.15	80.71	**95.35**	**86.65**
PU3	LC	96.68	94.34	96.03	95.45
	Strategy-1	95.81	95.71	**96.18**	**95.69**
PUA	LC	95.60	94.56	94.91	94.94
	Strategy-1	96.63	94.56	**95.53**	**95.49**

Experiments were conducted on the original PU1, PU2, PU3 and PUA corpus to verify the effectiveness of strategy-1. Table 1 shows the optimization results with strategy-1 as well as the performance of the original LC model. It is clear that the performance of the LC model is improved with the optimization process defined by strategy-1, indicating that strategy-1, e.g. the FWA-based optimization process, is

effective to improve the performance of the original LC model. On the other hand, as shown in Table 1, the performance improvement of the LC model with strategy-1 is limited due to that the validation set cannot well reflect the data distribution of the testing set all the time.

For the consideration of robustness, the second strategy (strategy-2) is designed based on strategy-1. Different from strategy-1, the fitness of each spark in this strategy is not simply computed on an independent validation set. Instead, 10-fold cross validation mechanism is employed in the process of computing fitness of each spark, where the original training set is divided into ten parts and one of them is defined as the validation set and others are defined as the new training set in each fold. The current spark is evaluated by training a corresponding model on the new training set and computing fitness on the validation set ten times. In this case, each spark is comprehensively evaluated by the performance on 10 folds. The optimal model that is corresponding to the optimal spark achieved and trained on the original training set is finally examined on the testing set. In this strategy, fitness of each spark is evaluated on the training set by 10-fold cross validation, overcoming the shortage of strategy-1 that the performance improvement of LC model is totally dependent on the consistency of data distribution in validation set and testing test. Strategy-2 enhances the robustness of the optimization process and is considered to achieve the improvements, with great performance, of the LC model.

Table 2. Performance comparison of LC before and after optimization with strategy-2

Corpus	Approach	Precision (%)	Recall (%)	Accuracy (%)	F1 (%)
PU1s	LC	100	92.36	96.67	95.88
	Strategy-2	100	96.64	**98.57**	**98.22**
PU2s	LC	100	64.00	90.71	74.62
	Strategy-2	100	94.17	**98.57**	**96.57**
PU3s	LC	97.84	91.30	95.37	94.34
	Strategy-2	98.25	95.91	**97.56**	**97.02**
PUAs	LC	95.78	90.72	93.64	92.68
	Strategy-2	98.75	96.44	**97.73**	**97.42**

Considering the efficiency of experiments, we randomly selected part of each corpora instead of the original corpus to investigate the effectiveness of strategy-2, e.g. 20% samples of PU1, PU2 and PUA were selected to form PU1s, PU2s and PUAs, and 10% samples of PU3 were selected to form PU3s. Table 2 presents the comparison of LC model before and after the optimization with strategy-2. It is notable that strategy-2 indeed brings a great improvement to the performance of the LC model, validating the effectiveness (taken the precision, recall, accuracy and F1 into account) of this strategy as well as the FWA-based optimization process. But the drawback of strategy-2 is that employing 10-fold cross validating in computing the fitness of sparks is time consuming. However, in fact, the usual offline training of the spam filters in the real world endows this strategy with usability.

5 Conclusion

This paper proposes a new framework that optimizes anti-spam model with heuristic swarm intelligence optimization algorithms, and this framework could integrate various classifiers and feature extraction methods. 2 experimental strategies were designed to objectively reflect the performance of the framework. Then, experiments are conducted, using the Fireworks Algorithm (FWA) as the Swarm Intelligence algorithm, the Local Concentration approach as the feature extraction method, and SVM as the classifier. During the experiments, 3 core parameters of the LC approach and 2 core parameters of SVM were optimized by using FWA. Experimental results demonstrated that the framework improved the performance on the corpora PU1, PU2, PU3 and PUA, and the computational efficiency is applicable in real world.

In future work, we intend to incorporate other swarm intelligence algorithms, feature extraction methods and classifiers into the framework, and investigate their performance under these configurations.

Acknowledgements. This work is supported by the National Natural Science Foundation of China (NSFC), under grant number 61170057 and 60875080.

References

1. Commtouch,: Internet threats trend report-February 2013. Tech. rep. (2013)
2. Cost of spam, http://www.ferris.com/2009/01/28/cost-of-spam-is-flattening-our-2009-predictions/
3. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Transactions on Neural Network 10, 1048–1054 (1999)
4. Ruan, G., Tan, Y.: Intelligent detection approaches for spam. In: Proceedings of International Conference on Natural Computation, pp. 1–7 (2007)
5. Bickel, S., Scheffer, T.: Dirichlet-enhanced spam filtering based on biased samples. Adv. Neural Inf. Process. Syst. 19, 161–168 (2007)
6. Kanaris, I., Kanaris, K., Houvardas, I., Stamatatos, E.: Words versus character N-grams for anti-spam filtering. Int. J. Artif. Intell. T. 16(6), 1047–1067 (2007)
7. Zhu, Y.C., Tan, Y.: A local-concentration-based feature extraction approach for spam filtering. IEEE Transactions on Information Forensics and Security 6(2), 1–12 (2011)
8. Information gain, http://en.wikipedia.org/wiki/Information_gain
9. Koprinska, I., Poon, J., Clark, J., Chan, J.: Learning to classify e-mail. Inform. Sci. 177, 2167–2187 (2007)
10. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) ICSI 2010, Part I. LNCS, vol. 6145, pp. 355–364. Springer, Heidelberg (2010)
11. Dasgupta, D.: Advances in artificial immune systems. IEEE Computational Intelligence Magazine, 40–49 (2006)
12. Guzella, T.S., Caminhas, M.: A review of machine learning approaches to spam filtering. Expert Syst. Appl. 36, 10206–10222 (2009)
13. Blanzieri, E., Bryl, A.: A Survey of Learning-Based Techniques of e-mail Spam Filtering. Tech. Rep. 1 DIT-06-065 (2008)

14. Timmis, J.: Artificial immune systems—today and tomorrow. Nat. Comput., 1–18 (2007)
15. Oda, T., White, T.: Developing an immunity to spam. In: Cantú-Paz, E., et al. (eds.) GECCO 2003. LNCS, vol. 2723, pp. 231–242. Springer, Heidelberg (2003)
16. Tan, Y., Deng, C., Ruan, G.: Concentration based feature construction approach for spam detection. In: Proceedings of International Joint Conference on Neural Networks, pp. 3088–3093 (2009)
17. Ruan, G., Tan, Y.: Intelligent detection approaches for spam. In: Proceedings of International Conference on Natural Computation, pp. 1–7 (2007)
18. Tan, Y.: Multiple-point bit mutation method of detector generation for SNSD model . In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3973, pp. 340–345. Springer, Heidelberg (2006)
19. Tan, Y., Xiao, Z.: Clonal particle swarm optimization and its applications. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 2303–2309 (2007)
20. Tan, Y., Wang, J.: A support vector network with hybrid kernel and minimal vapnik-chervonenkis dimension. IEEE Trans. Knowl. Data Eng. 26, 385–395 (2004)
21. Stuart, I., Cha, S.-H., Tappert, C.C.: A neural network classifier for junk E-mail. In: Marinai, S., Dengel, A.R. (eds.) DAS 2004. LNCS, vol. 3163, pp. 442–450. Springer, Heidelberg (2004)
22. Zhu, Y., Tan, Y.: A danger theory inspired learning model and its application to spam detection. In: Tan, Y., Shi, Y., Chai, Y., Wang, G. (eds.) ICSI 2011, Part I. LNCS, vol. 6728, pp. 382–389. Springer, Heidelberg (2011)
23. Ruan, G., Tan, Y.: A three-layer back-propagation neural network for spam detection using artificial immune concentration. Soft Comput. 14, 139–150 (2010)
24. Zhu, Y., Tan, Y.: Extracting discriminative information from E-mail for spam detection inspired by immune system. In: Proceedings of IEEE Congress on Evolutionary Computation, pp. 2491–2497 (2010)
25. Wu, C.-H.: Behavior-based spam detection using a hybrid method of rule-based techniques and neural networks. Expert Syst. Appl. 36, 4321–4330 (2009)
26. Siefkes, C., Assis, F., Chhabra, S., Yerazunis, W.S.: Combining winnow and orthogonal sparse bigrams for incremental spam filtering. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) PKDD 2004. LNCS (LNAI), vol. 3202, pp. 410–421. Springer, Heidelberg (2004)