



AUTOMATED NEGOTIATION BASED ON ENSEMBLE LEARNING AND OPTIMAL COUNTER PROPOSAL

Peng Yanbin ^{*1}, Zheng Zhijun ²

^{*1,2} School of Information and Electronic Engineering, Zhejiang University of Science and Technology, China



Abstract:

Agent-mediated automated negotiation is a key form of interaction in the e-commerce environment. Agents reach an agreement through an iterative process of making offers. However, agents are prone to conceal their private negotiation information, which decreases the efficiency of negotiation. In this paper, an ensemble learning-based negotiation method is proposed. The new method labels the proposals automatically by mining the implicit information in negotiation history data. Then, the labeled proposals become the training samples of the ensemble learning algorithm, which generates the estimation of the opponent's utility function. At last, based on the utility function of both sides, a win-win negotiation counter-proposal is generated through a particle swarm optimization algorithm. The experimental results indicate the benefits and efficiency of the proposed method.

Keywords: E-Commerce; Automated Negotiation; Ensemble Learning; Utility Function; Optimization Algorithm.

Cite This Article: Peng Yanbin, and Zheng Zhijun. (2020). "AUTOMATED NEGOTIATION BASED ON ENSEMBLE LEARNING AND OPTIMAL COUNTER PROPOSAL." *International Journal of Engineering Technologies and Management Research*, 7(5), 1-10. DOI: <https://doi.org/10.29121/ijetmr.v7.i5.2020.628>.

1. Introduction

In an e-commerce environment, negotiation is a key form of interaction to reach an agreement [1-4]. With the rapid development of agent technology, an intelligent agent can decide for themselves what actions they might perform, at what time, and under what conditions. Therefore, the agent can negotiate with each other on behalf of its owner enterprises. In e-commerce, the most common form of negotiation is service-oriented negotiation. Wherein, a service provider and a service consumer have to come to a mutually acceptable agreement over the negotiation issues such as price, quality and service level, etc. In traditional negotiation, agents only consider the benefit of its owner, thus act competitively in a service-oriented negotiation process. Negotiation is both competition and cooperation relationship. For example, suppose the service consumer thinks that money is more important than quality. In the meantime, the service provider thinks that quality is more important than money. The final agreement should have high quality at a high price, which satisfies the utility of both sides to an extreme. Thus, the overall utility is maximized. In such cases, agents not only concern the welfare for themselves but also for their opponent, which leads to a

win-win negotiation. Many research works have been done to provide a win-win negotiation solution.

The difficulties of win-win negotiation solution are information uncertainty and resource limitations. In a competitive business environment, enterprises are prone to conceal their private negotiation information to prevent being maliciously used. Therefore, agents only know their negotiation information, which is impossible to make a win-win counter-proposal. There is a lot of literature on promoting negotiation [5-11]. Zheng [6] propose a tri-training based algorithm to learn the opponent's negotiation preference. Firstly, the process of negotiation was viewed as a proposal's sequence which can be mapped into bidding trajectory feature space to form a sample set. Then, tri-training was imported to increase the number of samples and improve the prediction accuracy of the opponent's negotiation preference learning. Finally, based on the negotiation preference of both sides, an optimization algorithm is conducted to compute a win-win counter-proposal. Cheng [7] labels negotiation history data and uses a support vector regression machine to train and estimate the opponent's negotiation utility. Then a genetic algorithm is used to calculate the counteroffer to achieve a win-win negotiation. Hindriks [9] presents a generic framework based on Bayesian learning to learn an opponent model. The opponent model includes issue preferences and issue priorities of the opponent. The proposed algorithm can effectively learn the opponent's preferences from bid exchanges by making some assumptions about the preference structure and rationality of the bidding process. Cheng [10] proposes a support vector machine-based method to learn the opponent's attitudes to solve the problem of bilateral automated negotiation in an agent-mediated application. The procedure of negotiation was transformed into multiple negotiation tracks. Then the opponent's attitude of each issue can be got by learning the negotiation tracks. A negotiation decision-making model was constructed by utilizing the opponent's attitude.

In this paper, an ensemble learning-based negotiation method is proposed. The new method labels the proposals automatically by mining the implicit information in negotiation history data. Then, the labeled proposals become the training samples of the ensemble learning algorithm which generates the estimation of the opponent's utility function. At last, based on the utility function of both sides, a win-win negotiation counter-proposal is generated through a particle swarm optimization algorithm. The experimental results indicate the benefits and efficiency of the proposed method.

2. Negotiation Model

Negotiation model is defined as a six-tuple: $NM=(A,R,S,V,P,U)$. Wherein, $A=\{a_1,a_2,\dots,a_m\}$ denotes the set of Negotiation participants. In Bilateral negotiation, there are two participants: initiator and opponent. R denotes the negotiation rounds. In negotiation, agents reach an agreement through an iterative process of making offers. $S=\{s_1,s_2,\dots,s_n\}$ denotes the set of issues under negotiation. For example, in multi-issues negotiation, issues can be price, quality, and service level, etc. For each issue, there are minimum and maximum values, which correspond to the best and worst value the agents can accept. $V=\{v_1,v_2,\dots,v_n\}$ denotes the set of value range of issues. $v_i=[\min_i, \max_i]$ represents value range of issue s_i . $P=\{p_1,p_2,\dots\}$ denotes the set of proposals the negotiating agent offers. Proposal $p=\{x_1,x_2,\dots,x_n\}$ is a set of values for all issues. $x_i \in [\min_i, \max_i]$ is the value of issue s_i . U denotes the utility function of negotiating agents. Given a proposal

p , $U_i(p)$ is the utility value of proposal p for agent i . for each agent, there is a utility space which defines the maximum and minimum utility value the agent can accept. The final agreement should lie in the intersection of two agent's utility space.

When the initiator agent establishes and sends an initial proposal for negotiation, the negotiation starts. As can be seen from Fig 1, the initiator agent sends an initial proposal. After that, the opponent agent sends a counter-proposal. In the following, two agents iteratively send counter proposals to agree. At last, one of the agents will accept the last proposal being sent by the other agent or refuse negotiation when receiving an oppressive counter-proposal.

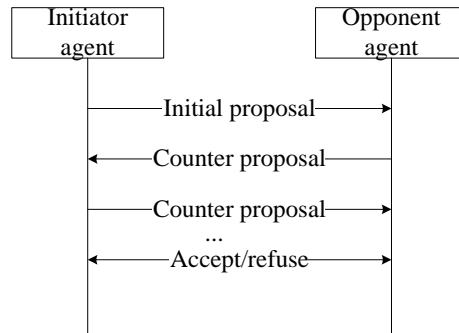


Figure 1: Flowchart of Negotiation

Negotiation decision model (NDM for short) is made up of six parts: message management module, negotiation history database, training sample generation, ensemble learning algorithm, counter proposal generation and negotiation decision. As is shown in Fig 2, the message management module is responsible for the agent's foreign communication. It receives the proposal sent by the opponent agent and stores the proposal into a negotiation history database. After internal processing, evaluation, and decision making, the module will send a counter-proposal to the opponent agent, accept or refuse negotiation.

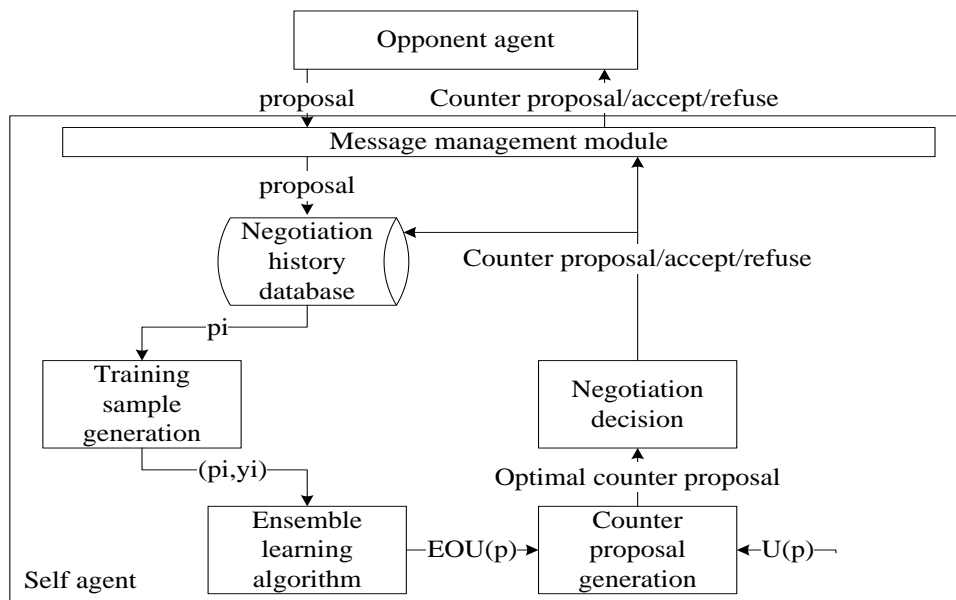


Figure 2: Negotiation decision model

Negotiation history database stores the historical negotiation information of both opponent and self agent. Historical negotiation information mainly contains a proposal list. The specific information includes negotiation round, proposal sender, proposal content, and the receiver's attitude. The data structure is shown in Table 1.

Table 1: Negotiation history database

Negotiation round	Sender	Content	Receiver's attitude
1	Initiator	$\{x_1, x_2, \dots, x_n\}$	Counter proposal
1	Opponent	$\{x_1, x_2, \dots, x_n\}$	Counter proposal
2	Initiator	$\{x_1, x_2, \dots, x_n\}$	Accept/Refuse

In an e-commerce environment, due to fierce competition, agents are prone to conceal their private negotiation information, especially their utility function. Therefore, it is impossible to gain that information directly. However, utility function to some extent implied in the proposals sent in the negotiation process. We can learn utility functions from the negotiation history database through a machine learning method. To improve the accuracy of learning, an ensemble learning algorithm is used to learn and predict the opponent's utility value. The role of training sample generation is to generate a training sample set for the ensemble learning algorithm. The training sample is composed of a proposal p_i and the estimation of utility value y_i . The output of the ensemble learning algorithm is an estimation of the opponent's utility function $UF()$. Based on the utility function of both sides, a win-win optimal negotiation counter-proposal is generated through particle a swarm optimization algorithm, this task is taken by a counter-proposal generation module. At last, the negotiation decision module will send counter-proposal to the opponent agent, accept or refuse negotiation. Suppose p_{oppo} is proposal sent by the opponent agent, p_{self} is the optimal counter-proposal just generated. If $u_{self}(p_{oppo}) \geq u_{self}(p_{self})$, accept the negotiation; if $u_{self}(p_{oppo}) < u_{self}(p_{self})$ send a counter-proposal; if time is over, refuse the negotiation.

3. Training Sample Generation

Negotiation is a process of both competition and cooperation. On the one hand, the negotiation participants hope to maximize their benefit. On the other hand, they also want a quick agreement. Fortunately, negotiation participant usually prefers different negotiation issues, which gives space for the success of negotiation. As is shown in Fig 3, the left eclipse represents the proposal space of negotiator A. Each point in the space represents a negotiation proposal. The whole eclipse is the acceptable domain of negotiator A. From left to right, the utility of the proposal decreases. The right eclipse represents the proposal space of negotiator B. Different from negotiator A, the proposal on the right side has a bigger utility. The intersection of two eclipses represents negotiation space, where each point (proposal) is acceptable by both sides. In Fig 3, $U_{A,max}$ represents the max utility value of negotiator A; $U_{A,min}$ represents the min utility value of negotiator A; $U_{A,unacc}$ represents the most unacceptable utility value of negotiator A, which negotiator B can offer. We call $[U_{A,min}, U_{A,max}]$ acceptable utility domain of negotiator A, $[U_{A,unacc}, U_{A,min}]$ unacceptable utility domain of negotiator A.

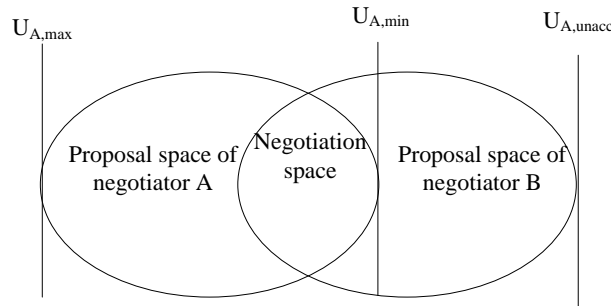


Figure 3: Negotiation space

In an e-commerce environment, the negotiator’s utility value isn’t public information. However, the utility value information is to some extent implied in proposals sent in the negotiation process. Firstly, any proposal is certainly acceptable by its sender. Further to say, because negotiators will gradually make concessions in the negotiation process, therefore, the utility of the proposal decreases with the increase of negotiation rounds. Secondly, if the receiver refuses a proposal, the current proposal and all the former received proposals are unacceptable to the receiver negotiator. Under these circumstances, the utility of the proposal increases with the increase of negotiation rounds.

Therefore, for the estimation of the utility of the opponent agent, we can divide the proposals into two categories: 1) the acceptable proposal, indicating the proposals sent by the opponent; 2) the unacceptable proposal, including the refused proposal and its former received proposals. According to the analysis above, we estimate the opponent’s utility value by the following formula:

$$y_i = \begin{cases} u_{opp,min} + (1 - \frac{R(p_i)}{AR})(u_{opp,max} - u_{opp,min}) \\ u_{opp,min} - (1 - \frac{R(p_i)}{AR})(u_{opp,min} - u_{opp,unacc}) \end{cases} \quad (1)$$

Wherein, $R(p_i)$ indicates the negotiation round the proposal p_i exists. AR indicates the number of negotiation round. $u_{opp,max}$, $u_{opp,min}$ and $u_{opp,unacc}$ is unknown information to the self agent. However, in the process of calculating the counter-proposal, we only need the relative value of $u_{opp,max}$, $u_{opp,min}$ and $u_{opp,unacc}$. Therefore, we use $u_{self,max}$, $u_{self,min}$ and $u_{self,unacc}$ to estimate the opponent’s utility parameters, which will not influence the calculating of counter-proposal. After the replacement, we get the following formula:

$$y_i = \begin{cases} u_{self,min} + (1 - \frac{R(p_i)}{AR})(u_{self,max} - u_{self,min}) \\ u_{self,min} - (1 - \frac{R(p_i)}{AR})(u_{self,min} - u_{self,unacc}) \end{cases} \quad (2)$$

Through formula (2), the training sample set is labeled, laying a solid foundation for the following ensemble learning algorithm.

4. Ensemble Learning Algorithm

In section 3, we have made a training sample set $\{(p_i, y_i), i=1,2,3,\dots,m\}$. To improve the classification effect, the ensemble learning method is imported. In the field of machine learning, ensemble learning algorithm uses multiple base learners to obtain better predictive performance than could be obtained from any of the constituent base learners [12-15]. Because the estimation of the opponent's utility value is a regression problem. Therefore, the relevance vector machine (RVM) [16-20] is imported as a base learner to solve this regression problem. Here we adopt the most widely used form of ensemble learning algorithm called AdaBoost. At each stage of the AdaBoost algorithm, it trains a new learner using a sample set in which the weighting coefficients are adjusted according to the performance of the previously trained learner so as to give greater weight to the misclassified sample points. Finally, when the desired number of base learners has been trained, they are combined to form a committee using coefficients that give different weights to different base learners [21-22]. Aiming to the characteristics of the problem in this paper, we design an AdaboostUtility algorithm. The precise form of the AdaBoostUtility algorithm is given below:

Algorithm 1: AdaboostUtility

Inputs: training sample set $\{(p_i, y_i), i=1,2,3,\dots,k\}$ where $p_i \in R^n$, $y_i \in R$; parameter β .

(1) Initialize the data weighting coefficients $\{WC(i)\}$ by setting $WC_1(i) = 1/k$ for $i=1,\dots,k$.

(2) For $t = 1,\dots,T$:

{ 1) Calculate sample probability: $pp_i^t = \frac{WC_t(i)}{\sum_{j=1}^k WC_t(j)}$, $i=1,\dots,k$

2) Train weak learner $RVM_t: p_i \rightarrow y_i$ on training sample set with distribution WC_t .

3) Calculate weighted error of sample set $\varepsilon_t = \sum_{i=1}^k pp_i^t [|y_i - RVM_t(p_i)| > \beta]$

4) Calculate parameter: $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$

5) Update the data weighting coefficients:

$$WC_{t+1}(i) = \frac{WC_t(i)}{N_t} \times \begin{cases} e^{-\alpha_t} & \text{if } |y_i - RVM_t(p_i)| \leq \beta \\ e^{\alpha_t} & \text{if } |y_i - RVM_t(p_i)| > \beta \end{cases}$$

Output: the final prediction function: $UF(p) = \frac{\sum_{t=1}^T \alpha_t RVM_t(p)}{\sum_{t=1}^T \alpha_t}$

Wherein, parameter β is a threshold, which is greater than zero. N_t is a normalization factor. $UF(p)$ is the estimation of the opponent's utility function.

5. Optimal Counter Proposal

AdaBoostUtility algorithm gains the estimation of the opponent's utility function $UF()$. Suppose its own utility function is $U()$. We can construct a compositive utility function $CU(p) = \xi U(p) + (1 - \xi)UF(p)$, wherein ξ is a weighting factor. The bigger the factor, the more weight is given to self utility. Vice versa, the more weight is given to the opponent's utility. The counter-proposal should maximize $CU()$, forming the following optimization problem:

$$\begin{cases} \max_p CU(p) = \xi U(p) + (1 - \xi)UF(p) \\ s.t. p \in \Omega \end{cases} \quad (3)$$

Wherein, Ω is value space of proposal p . A particle swarm optimization algorithm (PSO) is used to solve this optimization problem. The new algorithm is called CU_PSO. The precise form of the CU_PSO algorithm is given below:

Algorithm 2: CU_PSO

Set $k=1$;

Initialize particles swarm: $particles(k) = \{(p_1, V_1), (p_2, V_2), \dots, (p_d, V_d)\}$

Do{ for each particle{ calculate $CU(p_i)$;

If ($CU(p_i) > CU(pBest_i)$) $pBest_i = p_i$;

If ($CU(pBest_i) > CU(gBest)$) $gBest = pBest_i$;

for each particle{

$V_i = \omega \times V_i + c_1 \times rand() \times (pBest - p_i) + c_2 \times rand() \times (gBest - p_i)$

$p_i = p_i + V_i$ } $k++$;

}while maximum iterations or minimum criteria is not attained

Wherein, k represents iterations. p_i represents a negotiation proposal. V_i represents the speed of particle i . $pBest$ represents the local best value. $gBest$ represents the global best value. c_1 and c_2 represent learning factors. $Rand()$ represents the random number between $[0,1]$.

6. Results and discussions

A series of experimental tests have been undertaken to verify the performance of the ensemble learning-based negotiation algorithm (ELN for short). In the experiment, we take "induction cooker" trading as an example. The negotiation participants include cooker sellers and cooker buyers. Negotiation issues include price, power, quality, and warranty period. We analyze the performance of the negotiation model from two aspects: 1) the number of negotiation rounds. In a negotiation round, negotiators exchange a proposal. The number of negotiation rounds is an important performance indicator. Negotiators want to make agreements in as few negotiation rounds as possible. 2) Total negotiation utility. Total negotiation utility is the sum of the utility of both negotiators. In win-win negotiation, it is better to gain higher total negotiation utility. Two negotiation models are selected for comparison: 1) Coaching based negotiation model [1]; 2) Tri-training based negotiation model [6].

Experiment 1: In this experiment, we compare the average total negotiation utility between ELN, Coaching, and Tri-training based negotiation model. In each negotiation model, negotiation executes 100 times. The average total negotiation utility is shown in Table 2.

Table 2: Data comparison of ELN, Coaching and Tri-training based negotiation model

The number of training samples	Average total negotiation utility		
	ELN	Coaching	Tri-training
100	1.333	1.088	0.951
200	1.367	1.146	1.011
300	1.398	1.203	1.058
400	1.443	1.252	1.106
500	1.499	1.297	1.139

As shown in Table 2, with the number of training samples increase, the total negotiation utility of all negotiation model increase too. While the number of sample points is small, the Tri-training model achieves the lowest negotiation utility. ELN model achieves the best performance. Its success is based on two points. Firstly, the base classifier is RVM, which have the characteristics of both generative and discriminative algorithm. Secondly, ensemble learning further improves the performance of the RVM algorithm. Therefore, it is obvious the ELN algorithm outperforms the other model.

Experiment 2: In this experiment, we compare the average negotiation round and success ratio between Random, Tri-training, Coaching, and ELN based negotiation model. The random model uses a random concession strategy.

Table 3: Data comparison of Random, Tri-training, Coaching and ELN based negotiation model

Negotiation model	Average negotiation rounds	Negotiation success ratio
Random	40.2	60.2%
Tri-training	18.6	82.7%
Coaching	19.1	83%
ELN	15.3	88.3%

In the Random model, the agent didn't consider the opponent's utility in the process of generating a counter-proposal. Therefore, it will negotiate in the long run, and the success ratio is low. On the contrary, the other three models achieve better negotiation rounds and success ratio. It is because they all consider the opponent's utility. The mindset of cooperation promotes negotiation, achieves better performance. Among them, ELN achieves the best performance. It is because ELN has the most precise estimation of the opponent's utility.

7. Conclusions

This paper presented a formal model and relative machine learning algorithm for performing tradeoff in automated negotiation. Based on our former experiences in real-world negotiation, the negotiation algorithm had to be designed in a setting in which the negotiating agents have uncertain information about the utility function of their opponent. An ensemble learning-based negotiation method is presented. Firstly, the Training sample set is taken from the negotiation history database.

Secondly, an ensemble learning algorithm is trained to generate the estimation of the opponent's utility function. At last, a win-win negotiation counter-proposal is generated based on the utility function of both sides. The experimental results indicate the benefits and efficiency of the proposed method. In the next study, we will apply the ensemble learning algorithm to the setting of multi-parts negotiation.

Acknowledgements

Project supported by the National Natural Science Foundation of China (61803337)

References

- [1] Zhou Y, He H, Black A W, A Dynamic Strategy Coach for Effective Negotiation, Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue. 2019.
- [2] Zhichao L, Xuefeng Y. Adaptive Selective Ensemble-Independent Component Analysis Models for Process Monitoring, Industrial & Engineering Chemistry Research, 2018.
- [3] Darooei A, Khayyambashi M R. Design and Implementation of an Agent-based Trading Mechanism, Information Technology Journal, 2010, 9(2).
- [4] Faratin P, Sierra C, Jennings N. Using Similarity Criteria to Make Negotiation Trade-offs. Proceedings Fourth International Conference on MultiAgent Systems ICMAS-00, 2000.
- [5] Nava A, Natalie B, Corinne L, Negotiating a Better Future: How Interpersonal Skills Facilitate Intergenerational Investment, Quarterly Journal of Economics, 2017.
- [6] Zheng Z, Peng Y. Tri-Training based Bilateral Multi-Issue Negotiation Framework, Journal of Software, 2014, 9(5), 1129-1134.
- [7] YU C, JI G, HUAMAO G, Automated Negotiation Decision Model Based on Machine Learning, Journal of Software, 2009, 20(8), 2160-2169.
- [8] Sambuddha Ghosh, Gabriele Gratton, Caixia Shen. Intimidation: Linking Negotiation And Conflict, International Economic Review, 2019.
- [9] HINDRIKS K, TYKHONOV D. Opponent Modelling in Automated Multi-issue Negotiation Using Bayesian Learning, Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems Richland, USA, 2008, 331-338.
- [10] YU C, JI G, HUAMAO G, Negotiation Decision Model based on Learning of Opponent's Attitudes, Journal of Zhejiang University (engineering science), 2008, 42(10), 1676-1680.
- [11] Rahmawati Y, Utomo C, Zawawi N A W A . BIM and E-Negotiation Practices in AEC Consulting Businesses, Sustainability, 2019.
- [12] Sarker I H, Kayes A S M, Furhad M H, E-MIIM: An Ensemble-learning-based Context-aware Mobile Telephony Model for Intelligent Interruption Management, AI & Society, 2019(7).
- [13] Luque-Fernandez M A. ELTMLE: Stata Module to Provide Ensemble Learning Targeted Maximum Likelihood Estimation, Statistical Software Components, 2019.
- [14] Dong X, Zhiwen Y U, Cao W, A survey on Ensemble Learning, Frontiers of Computer Science, 2019, 14(2).
- [15] Zhichao Li, Xuefeng Yan, Adaptive Selective Ensemble-Independent Component Analysis Models for Process Monitoring, Industrial & Engineering Chemistry Research, 2018.
- [16] Yaguo L, Wu C, Naipeng L I, A Relevance Vector Machine Prediction Method Based on Adaptive Multi-kernel Combination and Its Application to Remaining Useful Life Prediction of Machinery, Journal of Mechanical Engineering, 2016.
- [17] Wu Y, Breaz E V, Gao F, Nonlinear Performance Degradation Prediction of Proton Exchange Membrane Fuel Cells Using Relevance Vector Machine, IEEE Transactions on Energy Conversion, 2016, 31(4).

- [18] Rui L I, Xiaodan W, Lei L, Ballistic Target HRRP Fusion Recognition Combining Multi-class Relevance Vector Machine and DS, Information and Control, 2017.
- [19] Zhao G, Zhang G, Liu Y, Lithium-ion Battery Remaining Useful Life Prediction with Deep Belief Network and Relevance Vector Machine, 2017 IEEE International Conference on Prognostics and Health Management (ICPHM). IEEE, 2017.
- [20] Zhang X, Luo G, He G, A Multi-Scale Residential Areas Matching Method Using Relevance Vector Machine and Active Learning, International Journal of Geo Information, 2017, 6(3).
- [21] Bishop C M. Pattern Recognition and Machine Learning, springer, 2006.
- [22] Kankanala, Padmavathy, Machine Learning Methods for the Estimation of Weather and Animal-related Power Outages on Overhead Distribution Feeders, Dissertations & Theses Gradworks, 2013.

*Corresponding author.

E-mail address: pyb2010@126.com