Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 630176, 6 pages
http://dx.doi.org/10.1155/2015/630176

Research Article

A High Accurate Multiple Classifier System for Entity Resolution Using Resampling and Ensemble Selection

Zhou Xing, Diao Xingchun, and Cao Jianjun

PLA University of Science and Technology, Nanjing 210007, China

Correspondence should be addressed to Zhou Xing; zx0327@163.com

Received 27 July 2015; Revised 15 September 2015; Accepted 29 September 2015

Academic Editor: Julien Bruchon

Classifiers are often used in entity resolution to classify record pairs into matches, nonmatches, and possible matches, the performance of classifiers is directly related to the performance of entity resolution. In this paper, we develop a multiple classifier system using resampling and ensemble selection. We make full use of the characteristics of entity resolution to distinguish ambiguous instances before classification, so that the algorithm can focus on the ambiguous instances in parallel. Instead of developing an empirical optimal resampling ratio, we vary the ratio in a range to generate multiple resampled data. Further, we use the resampled data to train multiple classifiers and then use ensemble selection to select the best classifiers subset, which is also the best resampling ratio combination. Empirical study shows our method has a relatively high accuracy compared to other state-of-the-art multiple classifiers systems.

1. Introduction

Entity resolution, also called duplicate record detection, is the process of identifying different or multiple records that refer to one unique real world entity or object [1]. It is widely used in homeland security, custom relationship database, and fraud and crime detection [2]. Christen summarized the outline of the general entity resolution, showing that entity resolution mainly comprises three steps: the first is indexing, where similar records are grouped together, the second is record pair comparison, where each field is compared using similarity function and numeric similarity values are generated, and the last is similarity vector classification, where records are classified into matches, nonmatches, and possible matches [2]. The possible matches refer to ambiguous records, which often need experts' participation to manually assess and further classify into matches or nonmatches. In case classification algorithm is used in similarity vector classification, entity resolution becomes a typical classification problem; for instance, Bilenko et al. used SVM to conduct similarity vector classification [3]. In order to improve the resolution effectiveness, existing methods used in classification like multiple classifier system can also be applied; for example, Tejada et al. used multiple classifiers to detect ambiguous records and asked users for feedback to reach high accuracy [4], which shows the advantage of using multiple classifier system in entity resolution. However, the applications of multiple classifier system in entity resolution remain rare, and even few researches take the characteristic of entity resolution into account in developing multiple classifier system.

In this paper, we focus on the third step of entity resolution by constructing a multiple classifier system to improve resolution effectiveness; we made use of the characteristic of entity resolution in developing multiple classifier system too.

2. Related Work

Many kinds of multiple classifier systems have been developed, like Bagging, Boosting, and AdaBoost [5]. AdaBoost emphasizes the weight of ambiguous instances, to gain high accuracy, showing the effectiveness of emphasis on ambiguous data; the training is a sequential process.

Instead of selecting all classifiers in developing multiple classifier system, Zhou et al. showed that selecting a proper subset is superior to selecting all; his work also showed that it is better to select from parallel problems like Bagging than from sequential problems like Boosting [6].

The process of selecting a subset from a multiple classifier system is called ensemble selection, ensemble pruning, ensemble thinning, and so on. Many have been working on ensemble selection; the diversity among component classifiers is regarded as playing an important role in ensemble selection, but it is still an open problem on how to measure and evaluate diversity [7, 8].

Yu et al. managed the diversity in a deterministic mathematical programming framework; they conducted a theoretical analysis in a PAC learning framework to show that the diversity can effectively reduce the hypothesis space complexity, implying that the diversity control in ensemble selection plays a role of regularization as in statistical learning approaches; the solution is a quadratically constrained quadratic program (QCQP) and they used alternating optimization instead to improve efficiency [8].

Li et al. defined diversity based on the average of pairwise differences; they further used diversity and accuracy to conduct ensemble selection and achieve good results, and they also conducted a theoretical analysis and showed that encouraging diversity can reduce generalization error, thus enhancing accuracy; the solution is greedy forward pruning and is relatively efficient [9].

Rafal et al. defined the competence of classifiers based on the probability of correct classification and the pairwise diversity based on conditional probabilities of error and then constructed an ensemble selection model using competence and diversity; the solution of the model is a combinational optimization problem, and they solved it using simulated annealing [10].

Yin et al. defined a convex diversity measure based on ensemble ambiguity and presented a general ensemble selection framework with diversity and sparsity. With the convex measure, they converted the ensemble selection into a convex optimization problem; they further showed that sparsity will force some weights of classifiers to be zero, realizing selection [11].

Above all, it is concluded that diversity among component classifiers works like regularization in general statistical learning approaches; besides, those measures like accuracy, sparsity, competence, and so on focusing on the classification performance of component classifier and ensemble size can be used in ensemble selection too.

3. Resampling and Ensemble Selection

In this part, we develop a high accurate multiple classifier system, resampling and ensemble selection (RES), by first varying the resampling ratio in a range to resample the ambiguous instances to generate a group of new instances, then using the new instances to train multiple SVM classifiers, using diversity and sparsity to select the best classifiers subset, and then using weighted voting to make the final classification decision. The outline of the construction of multiple classifiers system is shown in Figure 1.

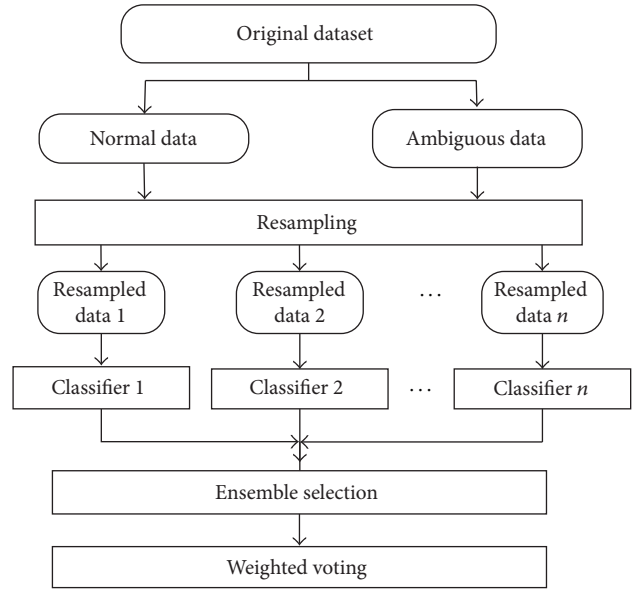

Figure 1: The outline of the construction of multiple classifiers system.

3.1. Resampling. In resampling, we make full use of the characteristic of entity resolution to distinguish ambiguous instances before classification and then use the idea of REA [12] to resample the ambiguous ones and use the resampled data to train a group of SVM.

Record similarity of each record pair is usually calculated in entity resolution; and those with high similarity are likely to be matches; those with low similarity are likely to be nonmatches; those with similarity neither too high nor too low can be either similar or distinct and can be assumed to be ambiguous.

The formal illustration of distinguishing ambiguous instances is as follows. Let **M** be a record similarity vector of duplicate record pairs, let **U** be a record similarity vector of distinct pairs, let E_M and E_U be the expectation of **M** and **U**, respectively, and let V_M and V_U be the variance of **M** and **U**, respectively. As the distribution of record similarity is approximately normal, then, since most values obeying normal distribution are within the region $(\mu_x - 3\sigma_x, \mu_x + 3\sigma_x)$, where μ_x is the expectation and σ_x is the variance, we can assume that those record pairs whose similarity is within $(E_M - 3V_M, 1)$ are likely to be duplicate, and those within $(0, E_U + 3V_U)$ are likely to be distinct; hence, those with similarity within the region $(E_U + 3V_U, E_M - 3V_M)$ can be regarded as ambiguous.

We give the pseudocode of resampling algorithm in Algorithm 1.

In Algorithm 1, we first calculate the upper bound and the lower bound (6–8), splitting the dataset into ambiguous and normal data (9–13), and then conduct resampling according to the resampling ratio (14–20).

Input:
(1) the dataset to be resampled: **D**
% each instance is the field similarity vector of a record pair, and the class label indicates whether the corresponding record pair is match or non-match
(2) the ratio of resampling: r
Initialization:
(3) **A** = [], **N** = [], **DO** = []
Splitting the dataset into ambiguous and normal:
(4) get the instance number N
(5) Split dataset **D** into duplicate data **DM** and distinct data **DU**
(6) Average each instance to get record similarity vector $\mathbf{S_M}$ and $\mathbf{S_U}$
(7) Calculate expectation and variance of $\mathbf{S_M}$ and $\mathbf{S_U}$ as E_M, E_U and V_M, V_U respectively
(8) Calculate lower bound LB of **DM** as LB = $E_M - 3V_M$ and the upper bound UB of **DU** as UB = $E_U + 3V_U$
(9) for $i = 1, \ldots, N$
(10) If UB $\le S_i \le$ LB %S_i is the similarity of the ith instance
(11) $\mathbf{A} = \mathbf{A} \cup \{\mathbf{D}_i\}$ % ambiguous instances
(12) Else
(13) $\mathbf{N} = \mathbf{N} \cup \{\mathbf{D}_i\}$ % normal instances
Resampling:
(14) For $i = 1, \ldots,$ round$(r \times N)$
(15) Randomly select an instance $\mathbf{A}_r$ from **A**
(16) $\mathbf{DO} = \mathbf{DO} \cup \{\mathbf{A}_r\}$
(17) For $j = 1, \ldots, N -$ round$(r \times N)$
(18) Randomly select an instance $\mathbf{N}_r$ from **N**
(19) $\mathbf{DO} = \mathbf{DO} \cup \{\mathbf{N}_r\}$
(20) Order **DO** in random order
Output:
(21) the resampled dataset: **DO**

Algorithm 1: Resampling algorithm.

The overall time complexity of resampling is linear.

3.2. Ensemble Selection. While conducting selection, we also use diversity and sparsity; the general formula is [11]

$$\begin{aligned} \min_{\mathbf{w}} \quad & f_{\text{loss}}(\mathbf{w}) \\ \text{s.t.} \quad & \text{sparsity}(\mathbf{w}) \le t_1 \\ & \text{diversity}(\mathbf{w}) \ge t_2, \end{aligned} \tag{1}$$

where t_1 is the sparsity control parameter and t_2 is the diversity control parameter.

For loss function, we use the least square error, as

$$f_{\text{loss}} = \sum_{j=1}^{m} \frac{1}{2} \left(\mathbf{w}\mathbf{h}_j - y_j\right)^2, \tag{2}$$

which is also used in [11], where $\mathbf{w}$ is the weight vector with $\mathbf{w} = (w_1, \ldots, w_n)$, $\mathbf{h_j}$ is the output vector of all classifiers on the jth instance, and $y_\mathbf{j}$ is the target label of the jth instance.

In case of binary classification with class label as 1 and −1,

$$f_{\text{loss}}(\mathbf{w}) = \sum_{j=1}^{m} \frac{1}{2} \left(\mathbf{w}\mathbf{h}_j - \mathbf{y}_j\right)^2 \le \sum_{j=1}^{m} \frac{1}{2} \times 2^2 = 2m. \tag{3}$$

So we can use $2m$ to normalize the loss function; then (3) can be written as

$$f_{\text{loss}} = \frac{1}{4m} (\mathbf{w}\mathbf{H} - \mathbf{Y}) (\mathbf{w}\mathbf{H} - \mathbf{Y})^{\mathbf{T}}, \tag{4}$$

where $\mathbf{H}$ is the prediction matrix of all classifiers, with size $n \times m$, n is the number of classifier sets, m is the number of test instances, and $\mathbf{Y} = (y_1, \ldots, y_m)$.

As to diversity measure, we use the diversity measure in [6], as

$$\begin{aligned} \text{div}(\mathbf{H}) &= 1 - \frac{1}{\sum_{1 \le i \ne j \le n} 1} \sum_{1 \le i \ne j \le n} \text{diff}\left(\mathbf{h}_i, \mathbf{h}_j\right), \\ \text{diff}\left(\mathbf{h}_i, \mathbf{h}_j\right) &= \frac{1}{m} \sum_{k=1}^{m} \mathbf{h}_i(x_k) \mathbf{h}_j(x_k) = \frac{1}{m} \mathbf{h}_i \mathbf{h}_j{}^{\mathbf{T}}. \end{aligned} \tag{5}$$

Also, in case of binary classification with class label as 1 and −1, (5) can be written as

$$\begin{aligned} \text{div}(\mathbf{H}) = 1 &- \frac{1}{\sum_{1 \le i \ne j \le n} 1} \sum_{1 \le i \ne j \le n} \frac{1}{m} \mathbf{h_i} \mathbf{h_j}^{\mathbf{T}} = 1 \\ &- \frac{1}{mn(n-1)} (1, \ldots, 1) \\ &\cdot \begin{pmatrix} 0 & \mathbf{h}_1\mathbf{h}_2{}^{\mathbf{T}} & \mathbf{h}_1\mathbf{h}_n{}^{\mathbf{T}} \\ \mathbf{h}_2\mathbf{h}_1{}^{\mathbf{T}} & 0 & \mathbf{h}_2\mathbf{h}_n{}^{\mathbf{T}} \\ & \ddots & \\ \mathbf{h}_n\mathbf{h}_1{}^{\mathbf{T}} & \mathbf{h}_n\mathbf{h}_2{}^{\mathbf{T}} & 0 \end{pmatrix} (1, \ldots, 1)^{\mathbf{T}}. \end{aligned} \tag{6}$$

As

$$\begin{aligned} &\begin{pmatrix} 0 & \mathbf{h}_1\mathbf{h}_2{}^{\mathbf{T}} & \mathbf{h}_1\mathbf{h}_n{}^{\mathbf{T}} \\ \mathbf{h}_2\mathbf{h}_1{}^{\mathbf{T}} & 0 & \mathbf{h}_2\mathbf{h}_n{}^{\mathbf{T}} \\ & \ddots & \\ \mathbf{h}_n\mathbf{h}_1{}^{\mathbf{T}} & \mathbf{h}_n\mathbf{h}_2{}^{\mathbf{T}} & 0 \end{pmatrix} \\ &= \begin{pmatrix} \mathbf{h}_1\mathbf{h}_1{}^{\mathbf{T}} & \mathbf{h}_1\mathbf{h}_2{}^{\mathbf{T}} & \mathbf{h}_1\mathbf{h}_n{}^{\mathbf{T}} \\ \mathbf{h}_2\mathbf{h}_1{}^{\mathbf{T}} & \mathbf{h}_2\mathbf{h}_2{}^{\mathbf{T}} & \mathbf{h}_2\mathbf{h}_n{}^{\mathbf{T}} \\ & \ddots & \\ \mathbf{h}_n\mathbf{h}_1{}^{\mathbf{T}} & \mathbf{h}_n\mathbf{h}_2{}^{\mathbf{T}} & \mathbf{h}_n\mathbf{h}_n{}^{\mathbf{T}} \end{pmatrix} \\ &- \begin{pmatrix} \mathbf{h}_1\mathbf{h}_1{}^{\mathbf{T}} & 0 & 0 \\ 0 & \mathbf{h}_2\mathbf{h}_2{}^{\mathbf{T}} & 0 \\ & \ddots & \\ 0 & 0 & \mathbf{h}_n\mathbf{h}_n{}^{\mathbf{T}} \end{pmatrix}, \end{aligned} \tag{7}$$

Input:
(1) the prediction matrix of all classifiers **H**
(2) the target label **Y**
(3) the control parameters: α and β
Initialization:
(4) $\mathbf{w} = \mathbf{0}$, $\mathbf{OP} = []$
Ensemble selection:
(5) get the instance number m
(6) apply an optimization tool to solve (10) to get **w**
(7) $\mathbf{OP} = \mathbf{wH}$
(8) $\mathbf{OP} = \text{sgn}(\mathbf{OP})$ % sgn is sign function
Output:
(9) the weighted predict: **OP**

Algorithm 2: Ensemble selection.

we have

$$\begin{aligned}\text{div}(\mathbf{H}) &= 1 - \frac{1}{\sum_{1\le i\ne j\le n} 1} \sum_{1\le i\ne j\le n} \frac{1}{m}\mathbf{h_i}\mathbf{h_j}^{\mathbf{T}} \\ &= 1 - \frac{1}{mn(n-1)}\left(\mathbf{DHH}^{\mathbf{T}}\mathbf{D}^{\mathbf{T}} - nm\right),\end{aligned} \tag{8}$$

where **D** is an all-1 n-dimension row vector.

As to sparsity, we just set $0 \le \mathbf{w} \le t_1$.

Since sparsity will force some weights to be zero, we extend the diversity measure to include weight as a parameter, so that only those classifiers with weight above zero will be selected in calculating diversity. The diversity measure thus becomes

$$\begin{aligned}\text{div}(\mathbf{H}, \mathbf{w}) = 1 - \frac{1}{mn'(n'-1)} \\ \left(\text{sgn}(\mathbf{w})\,\mathbf{HH}^{\mathbf{T}}\,\text{sgn}(\mathbf{w})^{\mathbf{T}} - n' * m\right),\end{aligned} \tag{9}$$

where sgn is a sign function and n' is the number of classifiers whose weights are above zero; it satisfies $n' = \text{sgn}(\mathbf{w}) * \text{sgn}(\mathbf{w})^{\mathbf{T}}$.

For solution, we convert formula (1) to

$$\begin{aligned}\min_{\mathbf{w}} \quad & f(\mathbf{w}) \\ & = \frac{1}{4m}(\mathbf{wH} - \mathbf{Y})(\mathbf{wH} - \mathbf{Y})^{\mathbf{T}} \\ & \quad + \alpha\frac{1}{mn'(n'-1)}\left(\text{sgn}(\mathbf{w})\,\mathbf{HH}^{\mathbf{T}}\,\text{sgn}(\mathbf{w})^{\mathbf{T}} - n' * m\right) \\ \text{s.t.} \quad & 0 < \mathbf{w} < \beta \\ & \|\mathbf{w}\| = 1.\end{aligned} \tag{10}$$

α and β are two control parameters, and (10) is a typical nonlinear programming problem and can be solved using existing optimization tool.

We give the pseudocode of ensemble selection in Algorithm 2.

In Algorithm 2, the main process is to use an optimization tool to solve (10); the overall time complexity is determined by the number of classifiers n, also related to the specific optimization tool used.

4. Experiment

4.1. Settings. We use 10 synthetic datasets and 5 real datasets. The synthetic datasets are abalone, dermatology, innosphere, breast cancer, seismic, ILPD, vote, biodeg, glass, and diabets from UCI machine learning repository; we then use a duplication generation tool to generate duplication. For convenience, we only choose numeric and nominal fields.

As to real world dataset, we use the datasets abt_buy, amazon_gp, dblp_acm, and dblp_scholar formally used in [13] and cora formally used in [14].

When calculating the similarity of each field, we use Jaccard similarity for characteristic, and $s(a,b) = 1 - ||a| - |b||/\max(|a|,|b|)$ for numeric data; as to nominal data, we use $s(a,b) = \begin{cases} 1 & a=b \\ 0 & \text{others} \end{cases}$.

For each dataset, we conduct 10 runs of 5-fold cross validation, by randomly selecting 4/5 pieces of data as training data and 1/5 as test data. On each dataset, each experiment is run for 10 times. We use the average and variance of the 10 runs to evaluate the classification performance.

We compare our algorithm with Gentle AdaBoost [5] (a sequential multiple classifier system), Bagging [1] (no resampling nor ensemble selection), DREP [6] (only ensemble selection), and REA [12] (only resampling).

For Gentle AdaBoost, we use the MATLAB code implemented by Alexander Vezhnevets; its base classifier is CART and it carries out 100 iterations.

For Bagging, we conduct Bootstrap sampling on the training data to train 21 SVM classifiers and then use majority voting to get the final prediction.

For DREP, we first conduct 1 run of Bagging and use the output of Bagging as the input of DREP; we vary the tradeoff parameter of DREP from 0.05 to 0.5 with step 0.05 to get 10 results and use the average as the final result.

For REA, we use its resampling equations to calculate the empirical resampling ratio and conduct resampling on the training data with the ratio to train SVM classifier.

For RES, we vary the resampling ratio from 0.4 to 0.6 or from 0.4 to 0.8 with step 0.01 to resample the training data and train SVM classifiers, then use the trained classifiers to predict on the test data to generate a prediction matrix, and use ensemble selection to get the final weighted prediction; for parameters in (10), we set α to be 1 and β to be 0.7 for simplicity.

We use rbf as the kernel function of SVM, and the width of rbf is 0.4; the tradeoff parameter is 100. We use fmincon in MATLAB optimization toolbox to solve (10).

4.2. Results. The overall accuracy comparison is shown in Table 1. On each dataset, an entry is marked with bullet "•" (or circle "○") if it is significantly better (or worse) than Bagging based on t-test at the significance 0.05; the win/tie/loss counts are summarized in the last row and the entry with the best performance of each dataset is marked in bold font.

DREP does not perform as well, which lies in the fact that its result depends on the input of Bagging, and it requires proper tradeoff parameter chosen.

Gentle AdaBoost wins 11 times and achieves the best performance on 8 datasets, and it can sometimes get a quite good

Table 1: Accuracy comparison.

	Dataset	Bagging	DREP	Gentle AdaBoost	REA	RES
Synthetic data	breast cancer	0.8996 ± 0.0002	0.9088 ± 0.0002●	0.9215 ± 0.0003●	0.9045 ± 0.0001●	**0.9327** ± 0.0002●
	innosphere	0.9647 ± 0.0001	0.9732 ± 0●	**0.9946** ± 0.0001●	0.9832 ± 0○	0.9911 ± 0●
	dermatology	0.9541 ± 0.0003	0.9676 ± 0.0002●	0.9649 ± 0.0002●	0.9588 ± 0.0001	**0.9825** ± 0.0001●
	ILPD	0.9619 ± 0.0001	0.9721 ± 0●	**0.9964** ± 0●	0.9579 ± 0.0001○	0.9929 ± 0.0002●
	seismic	0.9779 ± 0	0.9756 ± 0○	**0.9984** ± 0●	0.9878 ± 0●	0.9942 ± 0●
	abalone	0.9812 ± 0	0.9812 ± 0.0001	0.9880 ± 0●	0.9833 ± 0	**0.9940** ± 0●
	vote	0.7592 ± 0.0001	0.7495 ± 0.0005	**0.9515** ± 0.0001●	0.8155 ± 0.0004●	0.8447 ± 0.0003●
	biodeg	0.8036 ± 0.0001	0.7643 ± 0.0002○	**0.9643** ± 0.0001●	0.8155 ± 0.0004●	0.8214 ± 0.0002●
	glass	0.9255 ± 0.0002	0.9275 ± 0.0004●	**0.9804** ± 0.0001●	0.9216 ± 0.0006○	0.9412 ± 0.0003●
	diabets	0.9429 ± 0.0002	0.9457 ± 0.0008	**0.9837** ± 0●	0.8891 ± 0.0005○	0.9457 ± 0
Real data	abt_buy	0.9417 ± 0	0.9444 ± 0●	0.9372 ± 0.0001○	0.9465 ± 0●	**0.9535** ± 0●
	amazon_gp	0.9669 ± 0	0.9640 ± 0○	0.9525 ± 0.0001○	0.9729 ± 0●	**0.9796** ± 0●
	dblp_acm	0.9938 ± 0	0.9975 ± 0●	**0.9994** ± 0●	0.9966 ± 0●	0.9966 ± 0●
	dblp_scholar	0.9809 ± 0	0.9750 ± 0○	0.9783 ± 0○	0.9817 ± 0	**0.9848** ± 0●
	cora	0.9407 ± 0.0001	**0.9650** ± 0●	0.9550 ± 0●	0.9465 ± 0●	0.9600 ± 0●
	win/tie/loss		9/2/4	12/0/3	8/3/4	14/1/0

Table 2: Runtime comparison.

	Dataset	Bagging	DREP	Gentle AdaBoost	REA	RES
Synthetic data	breast cancer	989	0.011	1.3	54.3	0.25
	innosphere	178	0.006	0.73	8.5	0.09
	dermatology	533	0.006	0.62	25.3	0.06
	ILPD	336	0.008	0.92	16.0	0.38
	seismic	16527	0.006	1.83	806.8	0.24
	abalone	580	0.005	0.99	27.6	0.18
	vote	83	0.005	3.9	4.0	0.15
	biodeg	50	0.006	6.3	2.4	0.16
	glass	21	0.006	0.2	1.0	0.10
	diabets	17	0.003	1.1	1.5	0.26
Real data	abt_buy	8165	0.006	2.0	391.8	0.27
	amazon_gp	9408	0.006	2.2	437.7	0.27
	dblp_acm	1092	0.014	1.2	51.9	1.0
	dblp_scholar	2235	0.009	1.3	106.6	0.16
	cora	3594	0.006	2.9	135.6	0.2

result, but when focusing on real dataset, Gentle AdaBoost only wins twice and only achieves the best performance once; what is more, Gentle AdaBoost is sometimes inferior to Bagging; it especially loses on 3 real datasets.

REA performs not that well on synthetic data, but, on real dataset, it wins 4 times with slight improvement.

RES wins 14 times and achieves the best performance on 6 datasets; it especially achieves the best performance on 3 real datasets and is very close to the best performance on the remaining 2 datasets. It clearly shows that the proposed RES is superior to simple ensemble selection (DREP), simple resampling (REA), and Gentle AdaBoost in accuracy, as it can always achieve better performance than Bagging, and it can achieve the best performance sometimes.

The runtime comparison is shown in Table 2; each entry is the average of the 10 runs, and the unit is second.

The time complexity of Bagging mainly depends on the number of classifiers and the training of single classifier. It appears to be quite inefficient in this case, which is because it has 21 component classifiers, and the training of SVM is a little time consuming.

It is easy to see that DREP is quite efficient, as it only aims at ensemble selection procedure and uses greedy forward pruning.

Gentle AdaBoost is also efficient, because its base classifier is CART, which is relatively more efficient than SVM in this case.

The resampling of REA itself is linear, which is very efficient, though it appears to be time consuming; the main time consumed is during the training of SVM.

RES is relatively efficient too, as the resampling is linear, and, in case of medium scale nonlinear programming problem, the solution of the fmincon in MATLAB optimization toolbox switches to linear search, and the search is efficient too.

The time consumed by Bagging is almost 21 times that of REA, reflecting the efficiency of resampling. Besides, Bagging is the basis of DREP and RES.

RES is not much sensitive to parameter chosen, as the most key parameter in resampling is the resampling ratio, and it is replaced with a range; besides, the control parameters in (10) do not matter that much too.

5. Conclusion

By making full use of the characteristics of entity resolution, we emphasize the ambiguous instances through resampling; besides, we construct a parallel multiple classifiers system by varying the resampling ratio to form multiple classifiers and using ensemble selection to select the best classifier subset. The empirical study shows our system has relatively high accuracy compared to other state-of-the-art multiple classifier systems.

RES works in situation where accuracy is more emphasized, and it proves the effectiveness of resampling and ensemble selection in entity resolution.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007.

[2] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1537–1555, 2012.

[3] M. Bilenko, R. J. Mooney, W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "Adaptive name matching in information integration," *IEEE Intelligent Systems*, vol. 18, no. 5, pp. 16–23, 2003.

[4] S. Tejada, C. A. Knoblock, and S. Minton, "Learning domain-independent string transformation weights for high accuracy object identification," in *Proceedings of the 8ht ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '02)*, pp. 350–359, ACM, Edmonton, Canada, July 2002.

[5] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–374, 2000.

[6] Z.-H. Zhou, J.-X. Wu, Y. Jiang, and S.-F. Chen, "Genetic algorithm based selective neural network ensemble," in *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI '01)*, vol. 2, pp. 797–802, Seattle, Wash, USA, August 2001.

[7] Z.-H. Zhou and N. Li, "Multi-information ensemble diversity," in *Proceedings of the 9th International Workshop on Multiple Classifier System*, pp. 134–144, Cairo, Egypt, April 2010.

[8] Y. Yu, Y.-F. Li, and Z.-H. Zhou, "Diversity regularized machine," in *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*, pp. 1603–1608, Barcelona, Spain, July 2011.

[9] N. Li, Y. Yu, and Z.-H. Zhou, "Diversity regularized ensemble pruning," in *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, Bristol, UK, September 2012.

[10] L. Rafal, K. Marek, and W. Tomasz, "Optimal selection of ensemble classifiers using measures of competence and diversity of base classifies," *Neurocomputing*, vol. 126, pp. 29–35, 2014.

[11] X.-C. Yin, K. Huang, C. Yang, and H.-W. Hao, "Convex ensemble learning with sparsity and diversity," *Information Fusion*, vol. 20, no. 1, pp. 49–59, 2014.

[12] Y. Qian, Y. Liang, M. Li, G. Feng, and X. Shi, "A resampling ensemble algorithm for classification of imbalance problems," *Neurocomputing*, vol. 143, pp. 57–67, 2014.

[13] G. Papadakis, G. Koutrika, T. Palpanas, and W. Nejdl, "Meta-blocking: taking entity resolutionto the next level," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1946–1960, 2014.

[14] L. Leitão, P. Calado, and M. Herschel, "Efficient and effective duplicate detection in hierarchical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 5, pp. 1028–1041, 2013.