*AMSE JOURNALS-AMSE IIETA publication-2017-Series: Advances D; Vol. 60; N°4; pp 657-672*
*Submitted Feb. 2017; Revised March 15, 2017, Accepted April 15, 2017*

# Agent Simulation Model Based on Similarity Measure

Hongchen Wei, Xiaobin Li

Bao Ji University of Arts Of Sciences, BaoJi, Shanxi Province, 721016, China
(happypix @163.com, 343955450@qq.com)

## Abstract

Under the situation of scarcity of data in the target domain, the performance of the traditional agent simulation model tends to decrease. In this scenario, the useful knowledge in the source domain is extracted to guide the target domain learning to obtain more appropriate class information and agent simulation performance is an effective learning strategy. Based on the similarity measure, this paper proposes a Biased Agent Model (TIM) algorithm, which is similar to the source and target domain data distribution by introducing a biased learning mechanism (IM) algorithm to improve the simulation performance of the intelligent simulation (IM) algorithm in the data scarcity scenario. In order to ensure the validity of the bias, the TIM improves the performance of IM by considering the statistical and geometric characteristics of the source and target domains, the message passing mechanism in the algorithm makes it possible to achieve the goal of assisting the target domain learning. In addition, the factor graph of TIM can also show that the IM can be similar to IM in the case of lack of data in the target domain. The simulation results of the simulated data set and the real data set show that the proposed algorithm is more efficient than the classical IM algorithm in dealing with the non-sufficient data agent simulation task with better performance.

## Keywords

Biased learning, Statistical feature, Geometric structure, Similarity measure, Agent simulation method.

## 1. Introduction

The similarity measure (IM) algorithm [1] has been attracting the attention of many researchers because of its unique principle of intelligent simulation, which was introduced by Frey in 2007. According to the introduction in [1], the nature of IM algorithm is a belief propagation and maximization algorithm based on factor graph [2], which has the following advantages compared with other classical agent simulation methods: 1) IM agent simulation does not need to specify K (classical K-Means [3]) or other parameters describing the number of agents (SOM [4]) and 2) the most representative point in the simulation of an agent is called IM Different from the other simulation centers, the agent simulation point is the data point which exists in the original data, not the virtual point. 3) The simulation model of the IM agent is executed many times. The results are exactly the same and do not need to randomly select the initial step, that IM algorithm is not sensitive to initialization, 4) IM agent simulation than other methods of error squared and low [5, 6]. IM [7], IM [8], semi-supervised IM [9] and so on, which are based on IM.

The successful point of IM algorithm is the ability to automatically generate a reasonable amount of simulation of the number of agents, in the case of sufficient data, IM can accurately identify the representative of the agent simulation points, and the resulting intelligent body simulation results are often perfect. However, in practice, the data collected due to the high confidentiality of some production process data or the low cost of high-cost industries are usually very limited, resulting in scenes of scarcity of data often appearing. In the case of the agent simulation under the scenario, the simulation results of the IM algorithm are usually sensitive to the geometric distribution of the data. This is mainly because IM is designed to maximize the data points in each category to its agent simulation representative point. The sum of the energy and the real geometric distribution in the data scarcity scenario are often neglected, so it is difficult to meet any requirement except for the minimum energy, which leads to the inaccuracy of the representative point and the distribution matrix of the obtained agent. Therefore, if we continue to use the energy minimization principle of the IM algorithm and ignore the important information in the relevant field will affect the final simulation results of the intelligent body. Therefore, how to make the IM algorithm in the face of data scarcity scene can still have better class identification. The ability and high performance of agent simulation are currently awaiting solution. The Biased Learning Framework [10-12] is similar to human cognitive processes in that learning can be efficiently used to guide new things. It has been shown to be effective in solving machine learning problems in the context of data scarcity [13]. In recent years, the bias learning framework has been widely used in pattern classification, regression modeling and agent simulation, among which representative work includes:

1) In the field of pattern classification, the theory of bias learning is applied to the classification of unlabeled data [17]. In [18], the TPLSA algorithm based on bias learning theory

is proposed and applied to text categorization. In [19], the biased learning method of domain adaptation is successfully applied to large-scale emotion data classification. A bias learning method based on boosting algorithm [16] is proposed to solve the classification problem in the case of interference; recently proposed a multi-source adaptive bias learning strategy [15] for image classification.

2) In the area of regression modeling, a fuzzy system based on knowledge utilization and an enhanced version are proposed to solve the problem of fuzzy regression modeling in data scarcity scenarios [13, 14].

3) In the field of unsupervised agent simulation, the current research is still less. In 2012, a biased agent simulation model based on spectral method is proposed to solve the problem of biased agent simulation in text data.

Based on the above research results, we can find that the current bias learning strategy aims at abstracting the relevant effective knowledge from the source domain data to guide the learning of the target domain. The main learning strategies can be summarized as follows: 1) That is maximizing the embedded variance or minimizing the reconstruction error [13-20]; 2) maintaining the geometry such that similar data in the target domain remains in a similar expression to the samples in the source domain, although the above studies work on different, but only in the past, they chose only one learning mode and neglected the role of other structural information. In order to solve the problem, the IM algorithm is not able to simulate the performance of the intelligent agent when the data is lacking. In this paper, the biased learning mechanism is introduced into the IM algorithm to obtain the biased IM agent simulation model with knowledge biased ability. In addition, the statistical characteristics and geometrical structure of the data are considered in the process of knowledge bias to ensure the quality of bias. The proposed bias IM algorithm can be viewed as a generalization of the IM algorithm, which can effectively share information and bias information between similar domains or tasks while maintaining the classical form of IM. This paper will focus on the introduction and analysis of two IM algorithms, biased strategy and biased agent simulation model. Firstly, the IM algorithm and two bias strategies are introduced, and then the biased agent simulation method TIM based on similarity measure is introduced to integrate these two strategies. When the algorithm converges or reaches a certain number of iterations, the label distribution vector of each data point $c = \left[c_1, \dots\dots, c_N\right]$ is calculated by (1)

$$c_i = \arg\max_j \left[ a(i,j) + r(i,j) \right] \qquad (1)$$

It can be seen that the IM algorithm can efficiently find the best representative simulation points and distribution matrixes without any external intervention when the data volume is sufficient, but if there is no data in the absence of any extra data, it is easy to make the simulation center of intelligent body deviate greatly, which leads to the failure of the agent simulation. In this paper, we will give a new bias in the next chapter, Agent simulation method.

## 2. Bias similarity measure (TIM) agent simulation model

In this paper, we introduce the following two techniques to improve the current method based on the statistic features and geometry structure of the IM Agent simulation method described in the previous section. We propose a new method which is suitable for the IM algorithm Biased learning framework, the framework will make full use of the statistical characteristics of the source domain data (distribution matching bias strategy) and the geometric features (instance retention bias strategy) between the source domain data and the target domain data to improve the biased agent simulation results biased learning quality and enhanced IM algorithm in the face of data scarcity scenario of the agent simulation results, the specific method is shown in the following two sections.

### 2.1 Domain distribution approximation strategy

According to the theory of bias learning, the higher the similarity of domain distribution is, the closer the distribution of target domain is to the source domain, the more the knowledge abstracted by the source domain can guide the target domain data learning, the theoretical correctness should be improved using the similarity principle of data distribution among agents, we randomly extract some data from the source domain. If the distribution of the source domain and the target domain are closer, this data belongs to the energy of the respective class representative point of the source domain. Part of the data belong to the target domain class point of the energy are closer to the example shown in figure 1.

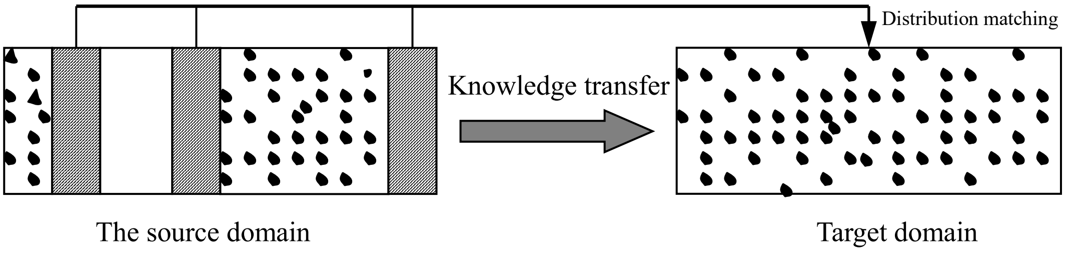


Figure 1. Distribution matching example

The distribution matrix is represented by $c=\left[c_1,......,c_N\right]$. The sum of the energy of all the data points of the target domain and the corresponding class representative point is denoted by $S_1$.

The sum of the squares of the differences between the energy of the data point $j$ which is the true representative point $c_i^{'}$ of the source domain, and the energy of the representative point $c_i^{'}$ belonging to the potential domain of the target domain is denoted by $S_2$.

$$S_1=\sum_{i=1}^{N}s\left(i,c_i\right) \qquad (2\text{-}1)$$

$$S_2=\sum_{i=1}^{N}\left[s\left(j,c_i\right)-s\left(j,c_i^{'}\right)\right]^2 \qquad (2\text{-}2)$$

Here a represents the potential representative of the target domain agent simulation, a represents the source of the simulated domain representative of the agent, b represents the number of data points selected from the source domain. Definition of a function matrix c that the target domain all the data and potential classes The energy relations of the representative points are expressed as follows:

Here $c_i$ represents the potential representative of the target domain agent simulation, $c_i$ represents the source of the simulated domain representative of the agent, $N^{'}$ and represents the number of data points selected from the source domain. Definition of a function matrix $\left[S_{ic_i}\right]_{N\times N}$ that the target domain all the data and potential classes the energy relations of the representative points are expressed as follows:

$$S_{ic_i}=s\left(i,c_i\right)+\lambda_1\square \qquad -s\left(j,c_i^{'}\right)\Big]^2 \qquad (3)$$

Here Mu is the pending parameter, (1), (2), (3), used to penalize the distribution differences between the source and target domains. from (2-1)、(2-2)、(3), we can get $S_1+\lambda_1\square \quad \overset{N}{\underset{I=1}{}} S_{ic_i}$ , here the value of the range of $\lambda_1$ is $\left[0.1,1\right]$, the interval is 0.1.

## 2.2 Domain Agent Representation Geometric Bias Strategy

As described in [12], the bias learning needs to deal with two cases: (1) the training samples used for learning do not satisfy the condition of independent and identical distribution with the new test samples; (2) there are not enough available training samples; It is found that the distribution property cannot guarantee the validity of the bias in the case of lack of data. Therefore, we use the geometric features of the class representative point of the source domain and the class representative point of the target domain to ensure the feasibility of the bias. The data of the manifold geometry is represented by the neighbor data to some extent. Therefore, the information of the representative point is simulated by the source domain agent to assist the selection of the representative agent of the target domain agent.

As shown in Figure 3, the black and red data points for the source domain data set, in which red is the representative point of its class; green and yellow data points for the target domain data set, the yellow is the representative point of the class. It is obvious that the representative point in the source domain data and its neighbors can be used to help the target domain learning. In addition, the bias strategy can speed up the convergence rate of the IM algorithm, we will introduce in Section 3.3.

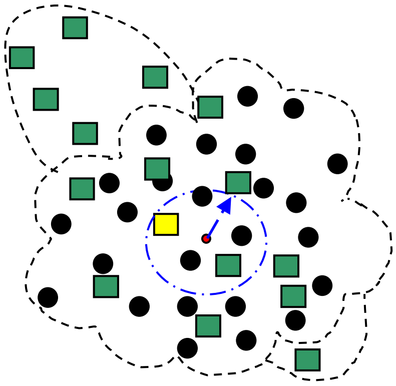


Figure 2. Example of geometric feature retention

According to Figure 2, in order to make the IM algorithm take advantage of the geometrical structure between the source domain data and the target domain data in the biased learning process, we define it as follows.

**Definition 1:**

(2 neighbors of object 1) The nearest neighbor of the data point object is denoted as, defined as: ( $x$ neighbors of object $\varepsilon$ ) the $\varepsilon$ neighbors of data point object $x$ are denoted as 5, defined as follows:

$$neb(x)=\{y\in D|dist(x,y)\le\varepsilon\} \quad (4)$$

Here $dist(y,x)$ is as the distance metric function, means the distance between $x,y$, $\varepsilon$ is threshold. The Euclidean distance is chosen for the $dist(y,x)$ function in this paper.

Because the source domain and the target domain have similar distribution and geometric distribution properties, the source domain data agent simulation represents the data of the target domain within the neighborhood of the representative point. The greater the probability of becoming a representative point of the simulation agent. Therefore, we target the domain of the data in the vicinity of the potential representative of the class point to do the following penalties:

$$\Delta_k(c)\begin{cases} -\infty, & \text{if } c_k \neq k \text{ bou}\exists c_i = k, \\ \lambda_2\square & = k \text{ and}\left(\exists j, c_k \in neb\left(c_j^{'}\right)\right), \\ 0, & \text{otherwist} \end{cases} \qquad (5)$$

Here $I$ is defined as follow:

$$I = \frac{1}{N}\sum_{i=1}^{N} s\left(i,c_k\right) \qquad (6)$$

$c_k$ is the latent class representative point, $\lambda_2$ is the penalty coefficient, and the specific values are given in the experimental section. It can be found that $\Delta_k(c)$ is similar to $\delta_k(c)$ in IM, which is expressed as punishment for potential representative point $c$. In addition, it is worth noting that $\Delta_k(c)$ the likelihood of the potential class representative point becoming the representative point of the final class is increased by imposing a penalty of $\lambda_2\square$ on the simulated representative point of the potential domain of the target domain of the proximal neighbor represented by the source domain agent.

In summary, the proposed objective function of the TIM algorithm can be defined as follows:

$$\begin{aligned} \max\ S(c) &= S_1 + \lambda_1\square \qquad c) \\ &= \sum_{i=1}^{N} ic_i + \Delta_k(c) \qquad (7) \\ &= \sum_{i=1}^{N} s\left(i,c_i\right) + \lambda_1\square_{J=1}^{N'} \qquad c_i) - s\left(j,c_i^{'}\right)\right]^2 + \Delta_k(c) \end{aligned}$$

Compared with the TIM algorithm in (7) and the IM algorithm in (1), we can find that the TIM algorithm draws on the data information in the source domain and has biased learning ability. The size reflects the center of the potential class in the target domain In addition, by adding the penalty term to the data points in the proximal neighbor of the source domain simulation, $\Delta_k(c)=\lambda_2\square \quad : k \text{ and} \left(\exists j c_k \in neb\left(c_j'\right)\right)$ reduces the IM algorithm to search for the range of the simulated representative points of the final agent and increases simultaneously.

The data in the source domain can be used as the representative point of the final class. Therefore, the information in the source domain can be used by the TIM in the case of insufficient data or information in the target domain to help it learn and to speed up the convergence of the algorithm, which can be found more easily to the target domain data simulation agent representative points and distribution matrix. Figure 3 for the TIM algorithm flow.

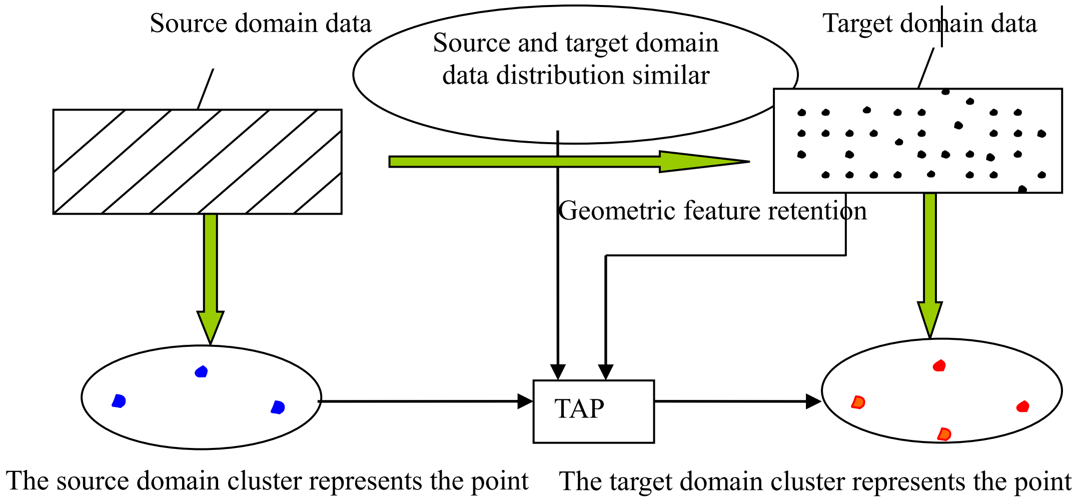

Source domain data
Source and target domain data distribution similar
Target domain data
Geometric feature retention
TAP
The source domain cluster represents the point
The target domain cluster represents the point

Figure 3. TIM algorithm flow

## 3. Experimental Study

### 3.1 Experimental Setup

In order to verify the simulation performance of this method in complex situations such as data scarcity, this chapter will analyze and evaluate the TIM algorithm by synthetic dataset and real network intrusion detection data KDD99 and SEA dataset respectively. For artificial synthesis a detailed description of the data and the real data will be given in Sections 3.2 and 3.3, respectively. In addition, for the TIM simulation of the proposed algorithm, the simulation performance will be evaluated in sections 3.2 and 3.3, with the most recent biases The algorithm of k-centers simulation based on real class center is given and analyzed by Transfer Spectral

Clustering (TSC) [20], IM algorithm [1], and the results are analyzed and explained properly. As follows: 1) In IM and TIM, when the simulation result of the agent is kept unchanged for 100 times, the algorithm is terminated, that is, the maximum number of iterations is set, the similarity is calculated by using the negative Euclidean distance. 2) The parameters of k-centers and TSC algorithms are set by reference [1] and [20]. For the Str IM algorithm, we use different data fragments as the number of neighbors, and the range of the number of neighbors is: Data stream processing, and at the current time to update the intelligent simulation center in order to process the next piece of data. For this algorithm, in order to use the formula (2-2) to verify the source and target domain distribution is consistent to achieve effective bias 10% of the source domain samples are randomly selected for the above test, and the extracted samples will be used as the source of the domain of the target domain to assist the knowledge involved in the target domain of the agent simulation process.

In order to reasonably evaluate the agent simulation performance of each agent simulation model, the following three evaluation indexes are used to analyze the performance of the algorithm.

1)Accuracy ACC is defined as follows:

$$ACC = \frac{\sum_{i=1}^{N} \delta\left(y_i, map\left(c_i\right)\right)}{N} \tag{8}$$

Here N is the number of data points, $y_i$ and $c_i$ are the real data tags and the obtained agent simulation tags, respectively, $\delta\left(y,c\right)$ means when $y = c$, the function value is 1, otherwise is 0, $map\left(\square\right.$ is permutation function, which matches each agent's simulation tag with the class label. The optimal matching results are described in the Hungarian algorithm.

2) Standardized mutual information $NMI$ is defined as follows:

$$NMI = \frac{\sum_{i=1}^{C}\sum_{j=1}^{C} N_{i,j} \log \frac{N\bullet}{N_i\bullet}}{\sqrt{\sum_{i=1}^{C} N_i \log \frac{N_i}{N}\bullet \;{}^{C}_{j=1} \qquad ; \frac{N_j}{N}}} \tag{9}$$

In the above formula, $N_{i,j}$ denotes the number of common data in the class $i$ and the real label $j$, $N_i$ denotes the number of data in class $i$, $N_j$ denotes the number of data in class $j$. $N_i$ denotes The number of data in the entire data set.

3) The RI indicator is defined as follows:

$$RI = \frac{f_{00} + f_{11}}{N\left(N-1\right)/2} \tag{10}$$

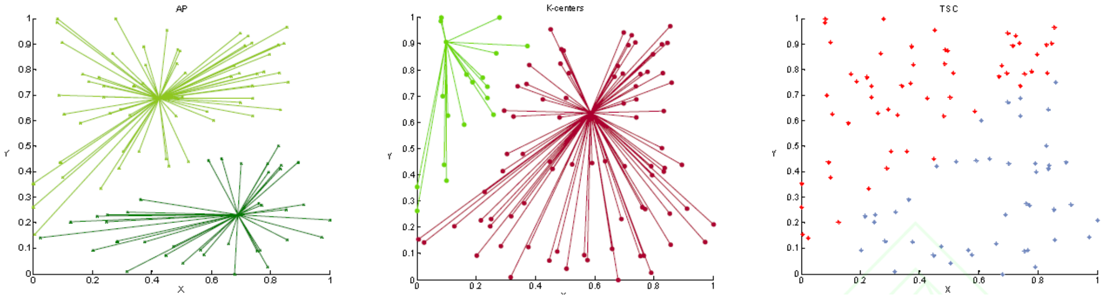
Here the number of pairing points indicating that the data points have different class labels and belonging to different classes indicates that the data points have the same class label and the number of pairing points belonging to the same class, and represents the total size of the entire data sample.

The above three methods show that the performance of the algorithm is superior to the higher value of the three methods, and the range of the above three methods is all with the value of the high value shows the algorithm of the performance is more superior.

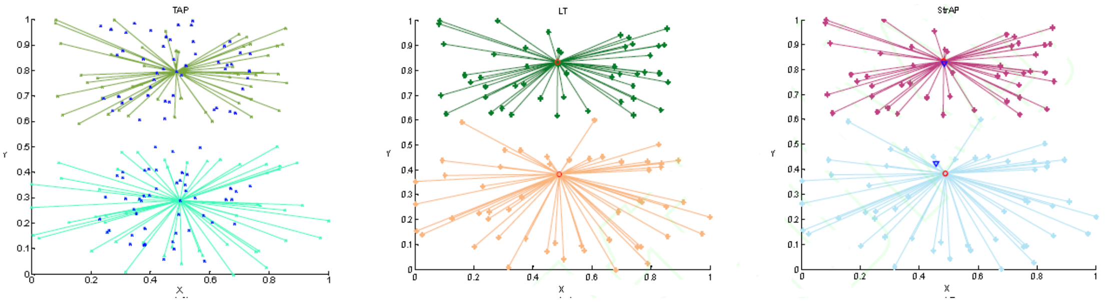
Experimental environment: Experimental hardware platform for the Windows32 bit 4Intel Corei3, memory is 4GB.Programming environment for the MATLAB2012b.

### 3.2 Experimental Analysis of Real Data Set

In order to further explore and analyze the TIM simulation performance and practical application value, this section will discuss the TIM algorithm on the real data set. We select two classical data sets, network intrusion detection data and SEA Data set. The time series of these two datasets have different distributions and the intra-class changes are large, which is meaningful for the robustness of the intelligent agent simulation model. The KDD99 dataset is a network connection defined as the sequence of TCP packets from the beginning to the end of a period of time, and during this period.



(a) (b) (c)



(d) (e) (f)

Figure 4. Agent simulation results on the target data set based on TIM and comparison algorithm

Table 1. compares the performance of various algorithms on the simulated data set

| Dataset | Index | LT | IM | K-Centers | TSC | StrIM | TIM |
|---|---|---|---|---|---|---|---|
| 7(c) | NMI | 0.8782 | 0.5725 | 0.0702 | 0.2977 | 0.8782 | **1.0000** |
| | ACC | 0.9700 | 0.8800 | 0.5193 | 0.8065 | 0.9700 | **1.0000** |
| | RI | 0.9604 | 0.7866 | 0.6100 | 0.6882 | 0.9604 | **1.0000** |

Data is transmitted from a source IP address to a destination IP address under a predefined protocol, such as TCP and UDP. Because the attack event is highly correlated in time, statistics on the connection between the current connection record and the previous period some relationships between records can better reflect the relationship between connections, with strong bias characteristics of the data set has 41 features, in addition to the discontinuation of the property after the remaining 32 properties. Smurf class accounted for 57.015%, Neptune class accounted for 21.582%, the remaining categories of a total of 1.782%; 2) SEA data sets (1), the total number of the data blocks of 22 categories, which accounted for 19.621% Is a data set with abrupt concept drift characteristics proposed by Street et al in 2001. This data has good bias, with 60, 000 samples, 3 attributes, two of which are related attributes, attributes Values ranging from 0 to 10, including four concepts, each concept, including 15000 samples. Remove 10% of the noise point, the conceptual function using the threshold data will be divided into two categories, namely greater than a threshold for the first class. Table 2 shows the algorithm test data fragment information.

In this part of the experiment, the amount of data in the target domain is 25% of the data in the source domain to form a data-deficient bias scenario. For the KDD dataset, the proportion of the data is extremely low, the number of classes is set to 3 and the data sets are normalized for both K-Centers and TSCs in the experiment. On this basis, the simulation performance of each algorithm is compared all experimentally set data sets were run 10 times, averaged and given the variance.

Table 3 shows the experimental results for the KDD99 dataset, where Table 3 shows the experimental results for the source domain data of 1-3000 and the target domain data of 1000. We can get the following conclusions:

The results of Table 3 show that the TIM algorithm is superior to other algorithms in terms of the effectiveness of NMI, ACC, and RI for most of the three agents, which further demonstrates that the TIM under,

Table 2. Source and Destination Domain Data Fragments

| Datasets | Source Datasets | Target Datasets | Source Datasets | Target Datasets |
|---|---|---|---|---|
| KDD, 99 | 1-3000 | 3001-3750 | 20001-23000 | 30001-30750 |
| | | 4001-4750 | | 40001-40750 |
| | | 5001-5750 | | 50001-50750 |
| | | 6001-6750 | | 60001-60750 |
| | | 7001-7750 | | 70001-70750 |
| SEA | 1-2000 | 2001-2500 | 10001-12000 | 20001-20500 |
| | | 3001-3500 | | 30001-30500 |
| | | 4001-4500 | | 40001-40500 |
| | | 5001-5500 | | 50001-50500 |
| | | 6001-6500 | | / |

The simulation results show that the proposed method can improve the simulation results of the target domain data. It is worth noting that the Biased Agent Simulation Model TSC essentially adopts a multi-task learning mechanism with the task number of 2 to complete the bias learning, Based on a coordination mechanism for the source domain and target domain data to coordinate the data distribution characteristics similar to both improve the learning performance of the role. However, for a class of data less geometric characteristics of the distribution of data changes, the algorithm is not very effective.

2) It can be seen from the experimental results table that IM-based simulation model of a class of agents is not sensitive to the initialization of the data, which makes the IM and TIM algorithm simulation results of the variance of 0. As can be seen, TIM algorithm which inherits the excellent characteristics of the stability of the IM algorithm, which is more practical than other intelligent simulation models.

3) As can be seen from Table 3, based on the source domain dataset class representative point

of the way to play the label changes in the data structure is no longer applicable, the agent simulation performance with the increasing deterioration of this decline more and more While a class of data-based agent simulation models keep the agent simulation performance by updating the agent simulation model, which is often achieved through some knowledge retention techniques (such as through the decay function mechanism) to update the agent simulation model It is worth noting that, in the current bias scenario, the data fragment is not continuous, as shown in Table 2, this time using the data flow algorithm for its intelligent simulation model because the upper and lower moments of data fault caused by the Intelligent Simulation Center cannot get accurate update, so as to achieve the desired agent simulation results.

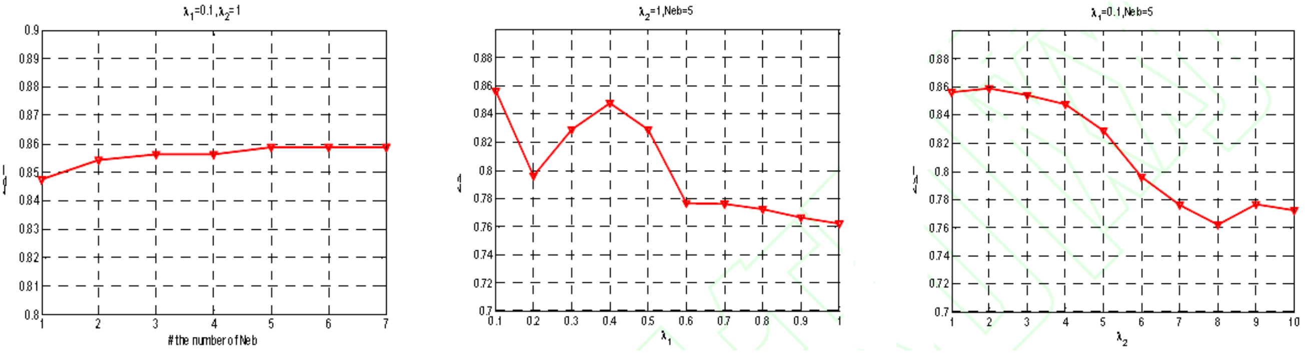
In conclusion, through the experiment and analysis on the real data set, we can get a definite conclusion that the TIM algorithm is better than the non-biased agent simulation model in dealing with the lack of data, The TIM algorithm considering the distribution and geometrical features is superior to the previous biased agent simulation model, so the superior performance of this algorithm has been fully verified and affirmed.

Table 3. Comparison of agent simulation performance on KDD99 datasets with different target domain data and source domain data of 1-3000

| Datasets | Index | LT | IM | K-Centers | TSC | StrIM | TIM |
|---|---|---|---|---|---|---|---|
| 3001-3750 | NMI | 0.8475±0 | 0.5857±0 | 0.6771±0.1397 | 0.6561±0.0033 | 0.8475±0 | **0.8561±0** |
| | AC | 0.9427±0 | 0.7493±0 | 0.7955±0.1122 | 0.5375±0.0085 | 0.9427±0 | **0.9440±0** |
| | RI | 0.9571±0 | 0.7423±0 | 0.8285±0.0950 | 0.7675±0.0048 | 0.9571±0 | **0.9593±0** |
| 4001-4750 | NMI | 0.7954±0 | 0.5920±0 | 0.7159±0.1047 | 0.6792±0.1047 | **0.8493±0** | 0.8377±0 |
| | AC | 0.9240±0 | 0.7507±0 | 0.8356±0.0899 | 0.5913±0.0227 | **0.9520±0** | 0.9333±0 |
| | RI | 0.9395±0 | 0.7489±0 | 0.8577±0.0870 | 0.7914±0.0008 | **0.9630±0** | 0.9548±0 |
| 5001-5750 | NMI | 0.7444±0 | 0.5992±0 | 0.7570±0.0921 | 0.6531±0.0036 | 0.8139±0 | **0.8483±0** |
| | AC | 0.9093±0 | 0.7653±0 | 0.8888±0.0760 | 0.5700±0.0017 | 0.9360±0 | **0.9440±0** |
| | RI | 0.9099±0 | 0.7591±0 | 0.8908±0.0988 | 0.7740±0.0001 | 0.9510±0 | **0.9639±0** |
| 6001-6750 | NMI | 0.5802±0 | 0.6097±0 | 0.7260±0.1757 | 0.6615±0.0214 | 0.8280±0 | **0.8468±0** |
| | AC | 0.6880±0 | 0.7573±0 | 0.8544±0.1203 | 0.5807±0.0527 | 0.9400±0 | **0.9440±0** |
| | RI | 0.7670±0 | 0.7496±0 | 0.8613±0.1367 | 0.7696±0.0117 | 0.9559±0 | **0.9625±0** |
| 7001-7750 | NMI | 0.4676±0 | 0.5873±0 | 0.7288±0.1121 | 0.6588±0.0083 | 0.8266±0 | **0.8590±0** |
| | AC | 0.5533±0 | 0.7680±0 | 0.8525±0.0833 | 0.5755±0.0194 | 0.9427±0 | **0.9493±0** |
| | RI | 0.6761±0 | 0.7505±0 | 0.8358±0.1139 | 0.7835±0.0171 | 0.9568±0 | **0.9680±0** |

### 3.3 Parameter Sensitivity Analysis

In order to further consider the influence of the default parameters involved in this paper on the simulation performance of the final agent in a specific agent simulation process, this section will analyze the sensitivity of the KDD99 dataset to the parameters of the algorithm. , The data segment 1-3000 in the KDD99 dataset is used as the source domain, and the data segments 3001-3750 as the target domain. Since the method consists of three manually set parameters, namely parameter 4, 4 and the number of neighbors, The parameters of fixed-optimal value of the change in another parameter observation algorithm performance changes. Figure 5 shows the three parameters on the performance of this algorithm simulation of the simulation results. From the results we can see that the number of neighbors changes in this algorithm And the parameters 4 and 4 control the degree of bias of the algorithm. The results of Fig. 5 show that the change of its numerical value will have certain influence on the simulation of the algorithm, but the influence Which is also within the acceptable range. In summary, the performance of the algorithm under the influence of the parameters, the results are more stable, parameter sensitivity is not.



(a) (b) (c)

Fig.5 Influence of different parameters of 4, 4, and the number of neighbors on the simulation results: (a) the effect of the number of neighbors on the TIM algorithm; (b) the effect of parameter 4 on the TIM algorithm; (c) the effect of parameters on the TIM algorithm

## 4. Conclusion

Based on the classical IM algorithm, this paper introduces a biased learning mechanism and uses source domain knowledge to assist target domain learning. A new biased similarity measure agent model is proposed to solve the problem of traditional data analysis task failure due to lack of target domain data samples. The algorithm is a knowledge bias simulation model based on data distribution and agent simulation point geometry, which utilizes both the geometrical structure of the source domain data and its statistical characteristics to obtain more instructive Which is a generalization version of IM, which can identify the number of agents and obtain the corresponding distribution matrix by a method of information propagation of factor graph similar to IM algorithm. In artificial data and real data the results of experiments show that the TIM algorithm is effective and efficient in the field of knowledge biased learning.

**References**

1. Frey BJ and Dueck D. Clustering by passing messages between data points. Science, 2007, 315:972-976. [doi: 10.1126/science.1136800]
2. Kschischang FR, Frey BJ, and Loeliger HA. Factor grIMhs and the sum-product algorithm. IEEE Trans. Inf. Theory, 2001, 47(2): 498 -519. [ doi: 10.1109/18.910572]
3. McQueen JB. Some Methods for classification and Analysis of Multivariate Observations. In: Proc. of 5th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, University of California Press, 1967.281-297.
4. Tsymbal A, Pechenizkiy M, Cunningham P, Puuronen S. Dynamic integration of classifiers for handling concept drift. Information Fusion, 2008, 9(1): 56-68. [doi: 10.1016/j.inffus.2006.11.002]
5. Dueck D, Frey BJ, Jojic N, Jojic V, Giaever G, Emili A, Musso G, Hegele R. Constructing treatment portfolios using affinity propagation. In: Proc. of 12th Annual International Conf. on Research in Computational Molecular Biology, 2008, 360-371. [doi: 10.1007/978-3-540-78839-3—31]
6. Jain AK. Data clustering: 50 years beyond K-means. Pattern Recognition Letters, 2010, 31(8): 651-666. [doi: 10.1016/j.patrec.2009.09.011]
7. Sumedha ML and Weigt M. Unsupervised and semi-supervised clustering by message passing: soft-constraint affinity propagation. The European Physical Journal B, 2008, 125-135. [doi: 10.1140/epjb/e2008-00381-8]
8. Xiao J, Wang J, Tan P, etc. Joint affinity propagation for multiple view segmentation. In: Proc. of the 11th IEEE Int. Conf. on Computer Vision, 2007, 1-7. [doi: 10.1109/ICCV.2007.4408928]
9. Strehl A and Ghosh J. Cluster ensembles---knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research, 2003, 583-617. [doi: 10.1162/153244303321897735]
10. Aggarwal CC, Han J, Wang J, and Yu P. A Framework for Clustering Evolving Data

Streams. In: Proc. of the 29th VLDB Conf., 2003, 81-92.

11. Shao L, Zhu F, Li X. Transfer Learning for Visual Categorization: A Survey. IEEE Transactions on Neural Networks and Learning Syatems, 2015, 26(5):1019-1034. [doi: 10.1109/TNNLS.2014.2330900]
12. Zhuang FZ, He Q, Shi ZZ. Survey on transfer learning research. Journal of Software, 2015, 26(1):26-39. [doi: 10.13328/j.cnki.jos.004631]
13. Deng ZH, Jiang YZ, Choi K.-S., Chung F.-L., Wang ST. Knowledge-Leverage-Based TSK Fuzzy System Modeling. IEEE Transactions on Neural Networks and Learning Systems, 2013, 24(8).1200」1212. [doi: 10.1109/TNNLS.2013.2253617]
14. Deng ZH, Jiang YZ, Cao LB, Wang ST. Knowledge-Leverage-Based TSK Fuzzy System with improved knowledge transfer, In: Proc. of the 2014 IEEE Int. Conf. on Fuzzy System, 2014, 178-185. [doi: 10.1109/FUZZ-IEEE.2014.6891544]
15. Tommasi T, Orabona F, CIMuto B. Learning categories from few examples with multi model knowledge transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(5):928-942. [doi: 10.1109/TPAMI.2013.197]
16. Dai W, Yang Q, Xue G, and Yu Y. Boosting for transfer learning. In: Proc. of Int. Conf. Machine Learning (ICML), 2007, 193-200. [doi: 10.1145/1273496.1273521]
17. Raina R, Battle A, Lee H, etc. Self-taught learning: transfer learning from unlabeled data. In: Proc. of the 24th Int. conf. Machine learning, 2007, 759-766. [doi: 10.1145/1273496.1273592]
18. Xue G R, Dai W, Yang Q, Yong Y. Topic-bridged PLSA for cross-domain text classification. In: Proc. of the 31st annual Int. ACM SIGIR conf. Research and development in information retrieval, 2008, 627-634. [doi: 10.1145/1390334.1390441]
19. Glorot X, Bordes A, Bengio Y. Domain adIMtation for large-scale sentiment classification: A deep learning IMproach. In: Proc. of the 28th Int. Conf. on Machine Learning (ICML), 2011, 513-520.
20. PIMadimitriou CH and Steiglitz K. Combinatorial Optimization: Algorithms and Complexity. Dover Publications, 1998.