Boosting Based Ontology Sparse Vector Computation Approach

Linli Zhu, Yu Pan, Muhammad Kamran Jamil, Wei Gao

Abstract—Ontology, as an effective structured information storage, classification, and statistical tool, has been attached importance from researchers of various disciplines. At present, ontology has become the core study field of semantic network, artificial intelligence, and information retrieval. The core topic of ontology algorithm is to calculate the similarity between concepts, so as to obtain the potential information of each other. In this paper, we present the boosting based iterative learning algorithm to compute the ontology sparse vector, and thus get the ontology function which maps each ontology concept to a real number. Since the advantage of ontology sparse vector is dimensionality reduction, the algorithm can extract the key information from the high dimensional ontology representation data. At last, we select four classical ontologies to test the efficiency of new ontology algorithm, and the rest give the positive answer.

Index Terms—Ontology, Similarity measure, Ontology mapping, Sparse vector, Boosting.

I. INTRODUCTION

Ontology is known in almost every corner of computer science domain as an information representation and shared model. Besides, ontology is efficient for its application in other domains like biology science, medical science, pharmaceutical science, material science, mechanical science and chemical science, as a concept semantic framework. (to illustrate, see Coronnello et al. [1], Vishnu et al. [2], Roantree et al. [3], Kim and Park [4], Hinkelmann et al. [5], Pesaranghader et al. [6], Daly et al. [7], Agapito et al. [8], Umadevi et al. [9] and Cohen [10]).

The graph $G = (V,E)$ can be used to represent the model of ontology, but the graph needs to meet the certain requirements that each vertex $v$ stands for a concept and each edge $e = v_iv_j$ stands for a close link between two concepts $v_i$ and $v_j$. The reason why we compute the ontology similarity lies in the need to learn a similarity function $S:\ V \times V \rightarrow \mathbb{R}^+ \cup \{0\}$. Every couple of vertices are mapped to this function to a real number. What’s more, considering the something common among some different ontologies, the ontology mapping can be very helpful to connect the relation with them. Two graphs $G_1$ and $G_2$ are used to denote the two ontologies $O_1$ and $O_2$, respectively. To be clear, we need to determine a set $S_v \subseteq V(G_2)$ for each $v \in G_1$ in which the vertices in $S_v$ have semantically high similarity to the concept which is corresponding to $v$. In addition, it may calculate the similarity $S(v,v_j)$ for each $v_j \in V(G_2)$ and select a parameter $0 < M < 1$ for each $v \in G_1$. $S_v$ is set for vertex $v$ and the factors of it satisfy $S(v,v_j) \geq M$. In this way, in its real applications, it’s essential to obtain the similarity function $S$ and to determine a proper parameter $M$. For details on similarity computing, please see Mazumdar et al. [11], Renu and Mocko [12], Hamedani et al. [13], Aouicha, and Taieb [14], Surianarayanan and Ganapathy [15], Tarko [16], Fernando and Webb [17], Segundo et al. [18], Phong and Son [19], and Spencer et al. [20].

The studies on ontology similarity measure and ontology mapping are quite popular and some good learning tricks have been proposed. For instance, the gradient learning algorithms for ontology similarity computing and ontology mapping were researched by Gao and Zhu [21]. Then, the stability analysis for ontology learning algorithms was done by Gao and Xu [22]. Hence, Gao et al.[23] continued to work on it by using ADAL trick and raised an ontology sparse vector learning approach. Based on the previous research, Gao et al. [24] re-considered the distance calculating techniques, and put forward an ontology optimization tactic. For more detailed theoretical analysis of ontology learning algorithm, please refer to [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35].

In terms of the boosting and greedy technologies, the paper concentrates on obtaining a new ontology learning trick. Moreover, the efficiency and effectiveness of the algorithm in the biology and chemistry are proved in the experiments.

II. SETTING

Assume $V$ be an instance space. $p$ dimension vector is taken to show the semantics information of every vertex in ontology graph. Especially, let $v = \{v_1, \ldots, v_p\}$ be a vector that corresponds to a vertex $v$. To complete the whole process, we purposely make the notations a little disorder and the ontology vertex and the relevant vector are denoted by $v$. It’s clear that we set it in this way aiming to get an ontology function $f: V \rightarrow \mathbb{R}$, then based on the comparison between realvalent real numbers of the vertices, we can get the similarity between them. Hence, we can consider the ontology function a dimensionality reduction operator $f: \mathbb{R}^p \rightarrow \mathbb{R}$.

With the popularity of ontology algorithm, these years have witnessed the rapid development of it. On the other hand, the relevant studies of it have also been more and more challenging. Among them, the related studies in chemistry

Manuscript received April 30, 2017; revised June 30, 2017. This work was supported in part by the Open Research Fund by Jiangsu Key Laboratory of Recycling and Reuse Technology for Mechanical and Electronic Products (RME-KF1612), the Industry-Academia Cooperation Innovation Fund Project of Jiangsu Province (BY201603-06) and Six Talent Peaks Project in Jiangsu Province (2016-XYDXXJS-020).

Linli Zhu is with the Department of Computer Engineering, Jiangsu University of Technology, Changzhou, China (email: zhlxlinli@jsu.edu.cn).

Yu Pan is with the Department of Computer Engineering, Jiangsu University of Technology, Changzhou, China.

Muhammad Kamran Jamil is with the Department of Mathematics, Riphah Institute of Computing and Applied Sciences, Riphah International University, 14 Ali Road, Lahore, Pakistan.

Wei Gao is with the Department of Information Science and Technology, Yunnan Normal University, Kunming, China.
Engineering Letters, 25:4, EL_25_4_07

...and biology are the cases in the point, because the data needed for the study are always big and in high dimension. Fortunately, we borrow the effective sparse vector learning algorithms in biology and chemical ontology computation (see Azfali et al. [36], Khormuji, and Bazarfakan [37], Ciaranella and Borzi [38], Lorincz et al. [39], Saadat et al. [40], Yamamoto et al. [41], Lorintiu et al. [42], Mesnil and Ruzzenze [43], Gopi et al. [44], and Dowell and Pinson [45] for more details). To illustrate, we are required to locate and decide the genes that directly cause a certain genetic disease. If we examine and study all the genes in the body, all efforts could be in vain. In fact, it’s always a certain amount of genes that cause the disease. The work for us is to target the fixed suspicious genes and it’s useful for us to choose the sparse vector learning algorithm to pinpoint genes in billions of disease genes in body.

A computational method of ontology function via sparse vector is shown by
\[ f_w(v) = \sum_{i=1}^{p} v_i w_i + \delta, \]
where \( w = \{w_1, \ldots, w_p\} \) is a sparse vector for shrinking irrelevant elements to zero and \( \delta \) is a noise term. By this model, learning the optimal sparse vector \( w \) is essential to get the ontology function \( f \).

For example, the standard framework with the penalize term via the \( l_1 \)-norm of the unknown sparse vector \( w \in \mathbb{R}^p \) is described below:
\[ Y_w = l(w) + \lambda ||w||_1, \]
in which \( \lambda > 0 \) is a balance parameter and \( l \) is the principal function to measure the error of \( w \). In order to measure the sparsity of sparse vector \( w \), we choose the balance term \( \lambda ||w||_1 \).

III. Boosting Based Ontology Sparse Vector Learning Algorithm Description

In this section, we first present the boosting algorithm in normal regression setting, then connect it with the ontology setting and determine the boosting based ontology learning algorithm for ontology sparse vector computation.

A. A functional gradient descent technology for gradient boosting

The functional gradient descent boosting algorithm is an old and famous learning algorithm which is simply called AdaBoost algorithm.

Normally, consider the learning problem which aims to estimate a real-valued function, and it is represented as
\[ f_0(\cdot) = \arg\min_{f(\cdot)} \mathbb{E}[l(f(x), y)], \]
where \( l(\cdot, \cdot) \) is a differentiable and convex loss function. For instance, the squared loss function \( l(f, y) = |y - f|^2 \) gets the well-known minimizer \( f_0(x) = \mathbb{E}[Y | X = x] \), and the L1 loss \( L_1(f, y) = |y - f| \) yields the population minimizer \( f_0(x) = \text{median}(Y | X = x) \). Another example which is balanced between \( L_1 \) loss and \( L_2 \) loss is Huber-loss function from robust statistics: \( l_{\text{Huber}}(f, y) = \frac{|y - f|^2}{2} \) if \( |y - f| \leq \Gamma \); otherwise, \( l_{\text{Huber}}(f, y) = \Gamma|y - f| - \frac{\Gamma^2}{2} \). Here, the value of \( \Gamma \) depends heavily on the iteration parameter \( m \). In the ontology engineering, \( \Gamma \) can be determined as
\[ \Gamma = \text{median}\{|y_i - \hat{f}^{m-1}(x_i)|; i \in \{1, \ldots, n\}\}. \]

To deal with the learning problem defined in (3) with boosting done in light of discussing the empirical risk \( \sum_{i=1}^{n} l(f(x_i), y_i) \) and searching iterative sharpest descent in the certain function space, we summarize the standard generic functional gradient descent learning algorithm as follows.


Step 1. Initialize
\[ \hat{f}_0(\cdot) = \arg\min_{f(\cdot)} \frac{1}{n} \sum_{i=1}^{n} l((f, y_i)) \]
or \( \hat{f}_0(\cdot) = 0 \). Set \( m = 0 \), and a number \( M \in \mathbb{N} \) which is large enough.

Step 2. Repeat the following steps until \( m \) reaches \( M \).

Step 3. \( m \leftarrow m + 1 \);

Step 4. Determine the negative gradient
\[ u_i = -\frac{\partial}{\partial f} l((f, y_i))|_{f=\hat{f}^{m-1}(x_i)} \]
for \( i \in \{1, \ldots, n\} \).

Step 5. Fit the negative gradient vector \( u_1, \ldots, u_n \) to \( x_1, \ldots, x_n \) by the real-valued based regression:
\[ (x_i, u_i)_{i=1}^{n} \rightarrow \hat{g}^m(\cdot). \]

Hence, \( \hat{g}^m(\cdot) \) is stated as an approximation of negative vector.

Step 6. Update \( \hat{f}^m(\cdot) = \hat{g}^m(\cdot) + \nu \hat{g}^m(\cdot) \), where \( \nu \in (0, 1] \) is a parameter to control the length of step.

Step 7: Go back to Step 2 and judge whether \( m \) reaches \( M \).

Now, let’s explain why we use the negative gradient vector in the boosting based algorithm. Consider \( L(f) = \sum_{i=1}^{n} l((f(x_i), y_i)) \) as the empirical risk function from \( f \in L_2(\sum_{i=1}^{n} \delta x_n) \) to \( \mathbb{R} \), where \( \sum_{i=1}^{n} \delta x_n \) is defined as the empirical measure of \( x \). We check that the associated inner product is formulated as \( \langle f, g \rangle_n = \sum_{i=1}^{n} f(x_i)g(x_i) \) for \( f, g \in L_2(\sum_{i=1}^{n} \delta x_n) \). Hence, the negative functional Gateaux derivative \( dL(f) \) of \( L(\cdot) \) is calculated by
\[ -dL(f)(x) = -\frac{\partial}{\partial \alpha} L(f + \alpha \delta x)|_{\alpha=0}, \]
where \( f \in L_2(\sum_{i=1}^{n} \delta x_n) \), \( x \in \mathbb{R}^p \), and \( \delta x \) is the indicator function at \( x \in \mathbb{R}^p \). If the derivative \( -dL \) at \( \hat{f}^{m-1} \) and \( x_i \) are computed, then we infer
\[ -dL(\hat{f}^{m-1})(x_i) = \frac{u_i}{n}, \]
where \( u_1, \ldots, u_n \) in above expression are exactly described in Step 4 of Algorithm 1.

B. \( L_2 \) Boosting for normal regression learning setting

In this subsection, we discuss the standard AdaBoost technology in the special setting where the loss function is exactly a \( L_2 \) squared loss: \( l_{\text{L}_2} = \frac{1}{2}|x|^2 \) (here the factor 0.5 leads to a convenient notation and the evaluated negative gradient of the loss function becomes the standard residual vector).

(Advance online publication: 17 November 2017)
Therefore, the Algorithm 1 described above can be restated as follows.

Algorithm 2. L2 Boosting regression learning algorithm.
Step 1. Initialize \( f_0(x) \) with an offset value. The default value is \( f_0(x) = \Phi \). Set \( m = 0 \), and a number \( M \in \mathbb{N} \) which is large enough.
Step 2. Repeat the following steps until \( m \) reaches \( M \).
Step 3. \( m \leftarrow m + 1 \).
Step 4. Calculate \( u_i = y_i - \hat{g}^{m-1}(x_i) \) for \( i \in \{1, \cdots, n\} \).
Step 5. Fit the negative gradient vector \( u_1, \cdots, u_n \) to \( x_1, \cdots, x_n \) by the real-valued based regression:

\[
(x_i, u_i)_{i=1}^n \rightarrow \hat{g}^m(\cdot).
\]

Hence, \( \hat{g}^m(\cdot) \) is stated as an approximation of negative gradient vector.
Step 6. Update \( \hat{f}(m) = \hat{g}^{m-1}(\cdot) + \nu \hat{g}^m(\cdot) \), where \( \nu \in (0, 1] \) is a parameter to control the length of step.
Step 7: Go back to Step 2 and judge whether \( m \) reaches \( M \).

C. L2 Boosting for ontology sparse vector model

Consider the ontology setting where the ontology data are realizations of \( (v_1, y_1), \cdots, (v_n, y_n) \) with \( p \) dimensional vectors \( v_i \in V \subseteq \mathbb{R}^p \) and response values \( y_i \in Y \subseteq \mathbb{R} \) for \( i \in \{1, \cdots, p\} \). For the continuous response value \( y \in \mathbb{R} \), we consider the following ontology linear computational model

\[
y_i = \sum_{j=1}^{p} w_j v_j^i + \varepsilon_i
\]

for \( i \in \{1, \cdots, n\} \) where \( \varepsilon_1, \cdots, \varepsilon_n \) are independent and identically distributed with \( \mathbb{E}[\varepsilon_i] = 0 \) and independent from all \( v_i \). Furthermore, the vector dimension \( p \) can be much larger than the sample volume \( n \).

As an example, the ontology algorithm is widely used to get the ontology sparse vector \( w \) which can be stated as follows

\[
\hat{w}_m = \arg\min_{w} \left\{ \frac{||Y - W||^2}{n} + \lambda ||w||_1 \right\},
\]

where \( Y \in \mathbb{R}^n \) is a response value, \( w \in \mathbb{R}^{n \times p} \) is an ontology information vector and \( \lambda \) is a balance parameter.

The above ontology framework can be extended into a more generalized linear model. Assume that \( y_1, \cdots, y_n \) are independent, and

\[
g(\mathbb{E}[y_i|v_i = v]) = \sum_{j=1}^{p} w_j v_j^i,
\]

where \( g(\cdot) \) is a real valued known function. Let \( f(\cdot, \cdot) : \mathbb{R}^p \times \mathbb{R} \rightarrow \mathbb{R}^+ \) be an ontology loss function. The sample error part of ontology framework can be expressed as

\[
L(w) = \frac{\sum_{i=1}^{n}(y_i - w_i)^2}{n}.
\]

Let \( m \) be the iterative parameter in ontology boosting algorithm. We set the previous active set of index \( S^m \subseteq \{1, \cdots, p\} \) for each iteration \( m \in \{1, 2, \cdots\} \). In order to decrease the ontology sample error, while resetting the previous coefficients of ontology sparse vector, we need to seek for a single additional index which can be stated as follows. For index subset \( S \subseteq \{1, \cdots, p\} \), introduce \( w_S \in \mathbb{R}^p \) as \( (w_S)_j = w_j \) if \( j \in S \), and \( (w_S)_j = 0 \) otherwise. In this way, the ontology sparse vector can be estimated by means of \( S \) as follows:

\[
\hat{w}_S = \arg\min_{w_S} L(w_S),
\]

where the minimization is implemented only over the components in view of \( S \).

In order to reduce the ontology sample errors, the forward index set selection is searching in each iteration \( m \) for the optimal single index with \( j_m \) which is formalized as follows:

\[
\hat{j}_m = \arg\min_{j \in \{1, \cdots, p\} \setminus S^{m-1}} L(\hat{w}_{S^{m-1} \cup \{j\}}),
\]

and thus the new active index set is

\[
S^m = S^{m-1} \cup \{\hat{j}_m\}.
\]

The detailed process of forward index set section for ontology sparse vector computation is presented in Algorithm 3.

Algorithm 3: Forward index set section for ontology sparse vector computation.
Step 1. Initialize the active index set \( S^0 = \emptyset \), and set a number \( M \in \mathbb{N} \) which is large enough.
Step 2. Repeat the following steps until \( m \) reaches \( M \).
Step 3. \( m \leftarrow m + 1 \).
Step 4. Compute the value of \( \hat{j}_m \) according to (6).
Step 5: Update \( S^m = S^{m-1} \cup \{\hat{j}_m\} \) and the corresponding estimator is formulated as

\[
\hat{f}^m = v \hat{w}_{S^m}
\]
as introduced in (5).
Step 6: Go back to Step 2 and judge whether \( m \) reaches \( M \).

D. Ontology sparse vector learning for squared ontology loss

With the squared ontology loss, the ontology sample error becomes

\[
L(w) = \frac{\sum_{i=1}^{n}(y_i - w_i)^2}{n}.
\]

In light of what discussed in above subsection, the L2-boosting process aims to choose the index set with index \( j \) satisfying

\[
\hat{j}_m = \arg\max_j \left( \frac{\sum_{i=1}^{n}(u_i^j)^2}{\sum_{j=1}^{n}(v_j^i)^2} \right),
\]

where \( u_i \) is the current \( i \)-th residuum \( u_i = y_i - \hat{f}^{m-1}(v_i) \).

Algorithm 4: L2-Boosting algorithm for ontology sparse vector computation in squared loss setting.
Step 1. Initialize the active index set \( S^0 = \emptyset \), and set a number \( M \in \mathbb{N} \) which is large enough.
Step 2. Repeat the following steps until \( m \) reaches \( M \).
Step 3. \( m \leftarrow m + 1 \).
Step 4. Compute the value of \( \hat{j}_m \) according to (7).
Step 5: Update \( S^m = S^{m-1} \cup \{\hat{j}_m\} \) and the corresponding estimator is formulated as

\[
\hat{w}_{S^m} = \arg\min_{w_S} \frac{||Y - V_{S^m}w_{S^m}||^2}{n},
\]

where \( V_S \in \mathbb{R}^{n \times |S|} \) is the sub ontology information matrix corresponding to the index set \( S \subseteq \{1, \cdots, p\} \). Then the \( \hat{w}_{S^m} \) is denoted as the estimated ontology sparse vector.
Step 6: Go back to Step 2 and judge whether \( m \) reaches \( M \).
Fig. 1. The Structure of “GO” Ontology

Fig. 2. “Physical” ontology $O_2$

Fig. 3. “Physical” ontology $O_3$
TABLE I
THE EXPERIMENT RESULTS OF ONTOLOGY SIMILARITY MEASURE

<table>
<thead>
<tr>
<th></th>
<th>$P@3$ average precision ratio</th>
<th>$P@5$ average precision ratio</th>
<th>$P@10$ average precision ratio</th>
<th>$P@20$ average precision ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm in our paper</td>
<td>0.5660</td>
<td>0.6904</td>
<td>0.8297</td>
<td>0.9449</td>
</tr>
<tr>
<td>Algorithm in Gao et al.</td>
<td>0.5192</td>
<td>0.6569</td>
<td>0.8031</td>
<td>0.9192</td>
</tr>
<tr>
<td></td>
<td>Algorithm in Gao et al. 27</td>
<td>0.5649</td>
<td>0.6827</td>
<td>0.8124</td>
</tr>
</tbody>
</table>

TABLE II
THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING

<table>
<thead>
<tr>
<th></th>
<th>$P@1$ average precision ratio</th>
<th>$P@3$ average precision ratio</th>
<th>$P@5$ average precision ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm in our paper</td>
<td>0.6913</td>
<td>0.8387</td>
<td>0.9419</td>
</tr>
<tr>
<td>Algorithm in Gao et al.</td>
<td>0.6774</td>
<td>0.7849</td>
<td>0.9032</td>
</tr>
<tr>
<td></td>
<td>Algorithm in Gao et al. 27</td>
<td>0.6774</td>
<td>0.7957</td>
</tr>
</tbody>
</table>

IV. EXPERIMENTS

Five simulation experiments about ontology similarity measure and ontology mapping are designed in this section to test whether our new algorithm is effective or not. At first, we get the sparse vector $w$, and then the ontology function is achieved by $f_w(v) = \sum_{i=1}^{N} v_i w_i$, where the noise term is not considered.

A. Ontology similarity measure experiment on biology data

“GO” ontology $O_1$, as described in Fig. 1, is popular as a convenient database among gene researchers, and the basic construction of it can refer to http://www.geneontology.org. The ontology is taken in our first experiment. $P@N$ (Precision Ratio, see Craswell and Hawking [46] for more details) is favored by lots of researchers to measure the effectiveness of experiment and we also take it here. In the first step, we have experts give the closest $N$ concepts (have highest similarity) for each vertex. Then, the algorithm is used to calculate the first $N$ concepts for each vertex on ontology graph. After that, we compute the precision ratios. By the way, in order to compare the effectiveness of algorithms, we also apply the ontology proposed earlier by Gao et al. [23], [27] to calculate the precision ratios in the way we used above. Parts of the experiment results are shown in to Tab. I.

It’s apparently shown in Fig. I that, if we take $N = 3, 5, 10$ or 20, the precision ratio computed by our new sparse vector ontology learning algorithm is higher than those by other previous algorithms by Gao et al. [23], [25], [27]. Furthermore, such precision ratios keep increase with the increase of $N$. Therefore, the conclusion is drawn that our newly proposed ontology learning algorithm has higher efficiency than the previous ones by Gao et al. [23], [25], [27].

B. Ontology mapping experiment on physical data

Physical ontologies $O_2$ and $O_3$ are traditionally used by researchers to test the feasibility of ontology mapping. The basic constructions of them $O_2$ and $O_3$ are shown in Fig. 2 and Fig. 3, respectively and our second experiment will adopt it. Same as the above experiment, we also use $P@N$ criterion to measure the experiment. The newly proposed algorithm is applied to achieve the ontology mapping between $O_2$ and $O_3$. By the way, in order to compare the effectiveness of algorithms, we also apply the ontology proposed earlier by Gao et al. [23], [27], [29] to calculate the precision ratios in the way we used above. Parts of the experiment results are shown in to Tab. II.

As obviously shown in the table, the newly proposed algorithm has far more efficiency than those put forward by Gao et al. [23], [27], [29] especially when $N$ is large enough.

C. Ontology similarity measure experiment on plant data

“PO” ontology $O_1$, as described in Fig. 4, is famous in plant science as a popular database to measure the effectiveness of ontology learning algorithm for ontology similarity calculating and we also take it here to learn and search concepts and botanical features. The basic construction of it can refer to http: //www.plantontology.org. Based on cROP project, Planteome is an effective online database for researchers to search for biological information of plants. The cROP project functions and collaborate with various plants-relevant projects both domestic and international to collect enough reference ontologies for plants. As a result, the database can be updated timely to maintain the latest and richest reference. On the other hand, it may help with the ontology use by providing a better annotation of the gene expression profiles and phenotypes in OMICs, etc.

What’s important is that the database can realize ontology cross-references, which helps to link the reference vocabularies with relevant terms in biology. It aims to build a semantic web of ontologies specifically for plant biology. In addition, the data annotation standards can be improved greatly by collaborating frequently with a wide range of plant genome sequencing and annotation projects and other relevant database like Gramene, iPlant, PlantEnsembl, Uniprot...

Many researchers may encounter the problems to find the exact description and references of certain terms in plant biology. If they search by different keywords, or vocabularies for one term, they may get different items online, which is pretty headache for them to identify and choose the description of what they want. However, planteome take
advantage of the cROP project to make it possible to get the comprehensive and timely resources in plant biology. It provides the reference ontologies for plants, which makes the classification much simpler and clearer. The database also get a web page for plant biology related blog which can provide latest developments in the field. In result, Planteome helps researchers search for the exact descriptions of plant ontology and make the annotations of scientific terms professional and in high standard.

Again, $P@N$ is taken in this experiment, too. In order to compare the effectiveness of algorithms, we also apply the ontology proposed earlier by Gao et al. [21], [25], [27] to the “PO” ontology to calculate the precision ratios in the way we used above. Parts of the experiment results are shown in to Tab. III.

As apparently shown in Tab. III, if we take $N = 3$, 5 or 10, the precision ratio computed by our new sparse vector ontology learning algorithm is higher than those by other previous algorithms by Gao et al. [21], [25], [27]. Furthermore, such precision ratios keep increase with the increase of $N$. Therefore, the conclusion is drawn that our newly proposed ontology learning algorithm has higher efficiency than the previous ones by Gao et al. [21], [25], [31].

D. Ontology mapping experiment on humanoid robotics data

Humanoid robotics ontologies, denoted as $O_5$ and $O_6$ by Gao and Zhu [21], is effective to orderly and clearly express the humanoid robotic and this experiment will take it to determine ontology mapping between $O_5$ and $O_6$. The basic construction of them can refer to in Fig. 5 and Fig. 6. Follow the convention, $P@N$ criterion is chosen again in the experiment. By the way, in order to compare the effectiveness of algorithms, we also apply the ontology proposed earlier
Fig. 6. “Humanoid Robotics” ontology $O_6$

Fig. 7. “Mathematical discipline” ontology $O_7$

(Advance online publication: 17 November 2017)
TABLE III

THE EXPERIMENT RESULTS OF ONTOLOGY SIMILARITY MEASURE

<table>
<thead>
<tr>
<th></th>
<th>$P@3$ average precision ratio</th>
<th>$P@5$ average precision ratio</th>
<th>$P@10$ average precision ratio</th>
<th>$P@20$ average precision ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm in our paper</td>
<td>0.5396</td>
<td>0.6697</td>
<td>0.9156</td>
<td>0.9792</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [21]</td>
<td>0.5042</td>
<td>0.6216</td>
<td>0.7853</td>
<td>0.9034</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [25]</td>
<td>0.5360</td>
<td>0.6664</td>
<td>0.9004</td>
<td>0.9673</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [27]</td>
<td>0.5081</td>
<td>0.6549</td>
<td>0.8104</td>
<td>0.9317</td>
</tr>
</tbody>
</table>

TABLE IV

THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING

<table>
<thead>
<tr>
<th></th>
<th>$P@1$ average precision ratio</th>
<th>$P@3$ average precision ratio</th>
<th>$P@5$ average precision ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm in our paper</td>
<td>0.4444</td>
<td>0.5556</td>
<td>0.8333</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [21]</td>
<td>0.4444</td>
<td>0.5185</td>
<td>0.6111</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [23]</td>
<td>0.2778</td>
<td>0.6111</td>
<td>0.7889</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [27]</td>
<td>0.4444</td>
<td>0.5370</td>
<td>0.8222</td>
</tr>
</tbody>
</table>

TABLE V

THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING ON MATHEMATICAL DATA

<table>
<thead>
<tr>
<th></th>
<th>$P@1$ average precision ratio</th>
<th>$P@3$ average precision ratio</th>
<th>$P@5$ average precision ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm in our paper</td>
<td>0.3462</td>
<td>0.5000</td>
<td>0.6846</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [21]</td>
<td>0.3077</td>
<td>0.4359</td>
<td>0.5615</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [47]</td>
<td>0.3462</td>
<td>0.4744</td>
<td>0.6000</td>
</tr>
<tr>
<td>Algorithm in Gao et al. [31]</td>
<td>0.3462</td>
<td>0.4487</td>
<td>0.5923</td>
</tr>
</tbody>
</table>

by Gao et al. [21], [23], [27] to calculate the precision ratios in the way we used above. Parts of the experiment results are shown in to Tab. IV.

As obviously shown in Tab. IV, the newly proposed algorithm has far more efficiency than those put forward by Gao et al. [21], [23], [27] especially when $N$ is large enough.

E. Ontology mapping experiment on mathematical ontology data

Mathematical discipline ontologies denoted as $O_7$ and $O_8$ and this experiment will take it to determine ontology mapping between $O_7$ and $O_8$. The basic construction of them can refer to in Fig. 7 and Fig. 8. Following the convention, $P@N$ criterion is chosen again in the experiment. By the way, in order to compare the effectiveness of algorithms, we
also apply the ontology proposed earlier by Gao et al. [21], [47], [31] to calculate the precision ratios in the way we used above. Parts of the experiment results are shown in Tab. V.

As obviously shown in Tab. V, the newly proposed algorithm has far more efficiency than those put forward by Gao et al. [21], [47], [31] especially when N is large enough.

V. CONCLUSION

In machine learning, the technology of sparse vector learning is widely used in various engineering applications for its high efficiency in data dimensionality reduction. AdaBoosting is a well-known iterative algorithm and has many extended algorithms employed in regression, ranking and classification. In our paper, we present the boosting based algorithm to compute the ontology sparse vector, and apply it in ontology similarity measure and ontology mapping. Five experiments imply that the proposed ontology learning algorithm has high efficiency in certain special applications. Therefore, the algorithm proposed in this paper has a wide application prospect in semantic network and information retrieval.

VI. CONFLICT OF INTERESTS

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

ACKNOWLEDGMENT

We thank the reviewers for their constructive comments in improving the quality of this paper.

REFERENCES


(Acceptance online publication: 17 November 2017)


Linli Zhu, male, was born in the city of Badong, Hubei Province, China on Feb. 13, 1981. He got two bachelor degrees on computer science from Zhejiang industrial university in 2004 and mathematics education form College of Zhejiang education in 2006. Then, he enrolled in department of computer science and information technology, Yunnan normal university, and got Master degree there in 2009. In 2012, he got PhD degree in department of Mathematics, Soochow University, China. He acted as lecturer in the department of information, Yunnan Normal University from July 2012 to December 2015. Now, he acts as associate professor in the department of information, Yunnan Normal University. As a researcher in computer science and mathematics, his interests are covering two disciplines: Graph theory, Statistical learning theory, Information retrieval, and Artificial Intelligence.

Muhammad Kamran Jamil is an Assistant Professor at Riphah Institute of Computing and Applied Sciences (RICAS). He did his B.S (2005-2009) in Mathematics from University of the Punjab. He obtained the MPhil (2011-2013) degree in Mathematics from Abdus Salam School of Mathematical Sciences (ASSMS), GC University, Lahore. After completing his MPhil he also pursued his PhD (2013-2016) degree program form the same institution (ASSMS). During his PhD, he received the Pre-PhD Quality Research Award. His area of research is graph theory. His major contribution for research is in chemical graph theory.

Wei Gao, male, was born in the city of Shaotong, Zhejiang Province, China on Sep. 20, 1975. He got Master degrees on computer software and theory from Yunnan normal university in 2007. Now, he acts as associate professor in the department of computer engineering, Jiangsu University of Technology. As a researcher in computer science, his interests are covered two disciplines: computer network and artificial intelligence.