Ontology Feature Extraction via Vector Learning Algorithm and Applied to Similarity Measuring and Ontology Mapping

Meihui Lan, Jian Xu, and Wei Gao

***Abstract*—In recent years, many learning technologies have been applied in ontology similarity measuring and ontology mapping via learning an ontology function $f : V \to \mathbb{R}$ which maps an ontology graph to the real line. In these settings, all the information for an ontology vertex (corresponding to concept) is expressed as a vector. However, in a special application, the value of ontology function for each ontology vertex is determined by a few components of the vector. The aim of feature extraction for ontology vector is to obtain these components to fix the index set of the vector, and such a procedure is equivalent to learning an ontology sparse vector in which most components are zero. In this paper, we raise an ontology sparse vector learning model for ontology similarity measuring and ontology mapping in terms of SOCP. The balance term consists of Ω norm, and the directed acyclic graph is employed in ontology setting for backward and forward procedure. Then, the active index set algorithm is designed to moderate the value of p, thus applications will be extended. Finally, five experiments are presented on various fields to verify the efficiency of the new ontology algorithm for ontology similarity measuring and ontology mapping in multidisciplinary research.**

***Index Terms*—ontology, similarity measure, ontology mapping, ontology sparse vector, second order cone programming**

I. Introduction

ONTOLOGY is derived from philosophy to describe the natural connection of things and the inherently hidden connections of their components. In information and computer science, ontology is often taken as a model for knowledge storage and representation. It has shown extensive applications in a variety of fields, such as: knowledge management, machine learning, information systems, image retrieval, information retrieval search extension, collaboration and intelligent information integration. Since a few years ago, because of its efficiency as a conceptually semantic model and an analysis tool, ontology has been favored by researchers from pharmacology science, biology science, medical science, geographic information system and social sciences (for instance, see Przydzial et al., [1], Koehler et al., [2], Ivanovic and Budimac [3], Hristoskova et al., [4], and Kabir [5]).

Manuscript received June 18, 2015; revised August 21, 2015. This work was supported in part by the Key Laboratory of Educational Informatization for Nationalities, Ministry of Education, the National Natural Science Foundation of China (60903131).

M. H. Lan is with Department of Computer Science and Engineering, Qujing Normal University, Qujing, 655011, China, e-mail: lanhuimei97@163.com.

J. Xu is with Department of Computer Science and Engineering, Qujing Normal University, Qujing, 655011, China, e-mail: xujian@163.com.

W. Gao is with the School of Information and Technology, Yunnan Normal University, Kunming, 650500, China, e-mail: gaowei@ynnu.edu.cn.

The structure of ontology is usually represented as a simple graph by researchers. We make every concept in ontology correspond to a vertex, so do the objects and elements. Then each (directed or undirected) edge on an ontology graph symbolizes a relationship (or potential link) between two concepts (objects or elements). Let O be an ontology and G be a simple graph corresponding to O. It can be attributed to getting The similarity calculating function, the nature of ontology engineer application, can be used to compute the similarities between ontology vertices. These similarities represent the intrinsic link between vertices in ontology graph. The ontology similarity measuring function is obtained by measuring the similarity between vertices from different ontologies, which is the goal of ontology mapping. The mapping serves as a bridge connecting different ontologies, through which a potential association between the objects or elements from different ontologies is gained. Or rather, the semi-positive score function $Sim : V \times V \to \mathbb{R}^+ \cup \{0\}$ maps each pair of vertices to a non-negative real number.

These years, ontology technologies have shown extensive applications in various fields. Ma et al., [6] presented a technology for stable semantic measurement based on the graph derivation representation. Li et al., [7] raised an ontology representation method for online shopping customers knowledge in enterprise information. By means of processing expert knowledge from external domain ontologies and in terms of novel matching tricks, Santodomingo et al., [8] raised a creative ontology matching system which gives complex correspondences. Pizzuti et al., [9] described the main features of the food ontology and some examples of application for traceability purposes. Lasierra et al., [10] argued that ontologies can be used to design an architecture for monitoring patients at home. More ontology applications on various engineering can refer to [11], [12], [13] and [14].

Using ontology learning algorithm is a good way to solve the ontology similarity computation, and a ontology function $f : V \to \mathbb{R}$ can be obtained. After using the ontology function, the ontology graph is mapped into a line which is made up of real numbers. The similarity between two concepts then can be measured by comparing the difference between their corresponding real numbers. Dimensionality reduction is the essence of this idea. A vector can be used to express all its information, in order to associate the ontology function with ontology application, for vertex v. And then, we slightly confuse the notations and use v to denote both the ontology vertex and its corresponding vector with the purpose to facilitate the representation. The vector is mapped to a real number by a dimensionality reduction operator, ontology function $f : V \to \mathbb{R}$, which maps multidimensional vectors into one-dimensional vectors.

There are several effective methods of getting efficient ontology similarity measure or ontology mapping algorithm in terms of ontology function. Wang et al., [15] considered the ontology similarity calculation in terms of ranking learning technology. Huang et al., [16] raised the fast ontology algorithm in order to cut the time complexity for ontology application. Gao and Liang [17] presented an ontology optimizing model in which the ontology function is determined by virtue of NDCG measure, and it is successfully applied in physics education. Since large parts of ontology structure can be tree-shaped, researchers explored the learning theory approach for ontology similarity calculating and ontology mapping in specific setting when the structure of ontology graph has no cycle. In the multi-dividing ontology setting, all vertices in ontology graph or multi-ontology graph are divided into k parts corresponding to the k classes of rates. The rate values of all classes are determined by experts. In this way, a vertex in a rate a has larger score than any vertex in rate b (if $1 \leq a < b \leq k$) under the multi-dividing ontology function $f : V \to \mathbb{R}$. Finally, the similarity between two ontology vertices corresponding to two concepts (or elements) is judged by the difference of two real numbers which they correspond to. Hence, the multi-dividing ontology setting is suitable to get a score ontology function for an ontology application if the ontology is drawn into a non-cycle structure.

In this article, we present a new ontology learning algorithm for ontology similarity measuring and ontology mapping by means of SOCP (second order cone programming). The rest of the paper is arranged as follows: in Section 2, detailed description of setting and notations for our ontology problem is manifested; in Section 3, we obtain the main algorithm for ontology index set algorithm based on SOCP; in Section 4, five respective simulation experiments on plant science, humanoid robotics, biology, physics education and university application are designed to test the efficiency of our new ontology algorithm, and the data results indicate that our algorithm has a high precision ratio for these applications.

II. Setting and Notations

Let $V \subset \mathbb{R}^d$ ($d \geq 1$) be a vertex space (or the instance space) for ontology graph, and the vertices (or, instances) in V are drawn randomly and independently according to some (unknown) distribution. Given a training set $S = \{v_1, \cdots, v_n\}$ of size n in V, the goal of ontology learning algorithms is to obtain a score function $f : V \to \mathbb{R}$, which assigns a score to each vertex.

Since the vector which corresponds to a vertex of ontology graph contains all the information of the vertex concept, attribute and the neighborhood structure in the ontology graph, it's always with high dimension. For instance, in the biological ontology, a vector may contain the information of all genes. In addition, ontology graph with a large number of vertices makes ontology structure very complicated, and the most typical example is the GIS (Geographic Information System) ontology. These factors may lead to the fact that the similarity calculation of ontology application will be very large. However, in fact, the similarity between the vertices is determined by a small part of the vector components. For example, in the application of biological ontology, a genetic disease often results from a small number of genes, leaving most of the other genes irrelevant. Furthermore, in the application of geographic information system ontology, if an accident happens in a place and causes casualties, then we need to find the nearest hospital ignoring schools and shops nearby, i.e., we just need to find neighborhood information that meets specific requirements on the ontology graph. Therefore, tremendous academic and industrial interest is attracted to researching into the sparse ontology algorithm.

In practical application, ontology function can be expressed by

$$f_\beta(v) = \sum_{i=1}^{p} v_i \beta_i. \tag{1}$$

Here $\beta = (\beta_1, \cdots, \beta_p)$ is an ontology sparse vector which is used to shrink irrelevant component to zero. To determine the ontology function f, we should learn the sparse vector β first. One popular ontology learning model with the balance term $g(\beta)$ of the unknown sparse vector $\beta \in \mathbb{R}^p$:

$$\min_{\beta \in \mathbb{R}^p} Y(\beta) = l(\beta) + g(\beta), \tag{2}$$

where $l(\beta)$ is a smooth and convex ontology loss function and $g(\beta)$ is a balance term which controls the sparsity of ontology sparse vector β. For example, the balance term usually takes the form of $g(\beta) = \lambda \|\beta\|_1$.

Fixed $\beta \in \mathbb{R}^p$ and $J \subseteq \{1, \cdots, p\}$ with cardinality $|J|$, β_J denotes the vector in $\mathbb{R}^{|J|}$ of elements of β are indexed by the element of subset J. For $\mathbf{M} \in \mathbb{R}^{p \times m}$, $M_{IJ} \in \mathbb{R}^{|I| \times |J|}$ denotes the sub matrix of $\mathbf{M}$ restricted to the columns indexed by J and the rows indexed by I. For arbitrarily finite set A with cardinality $|A|$, the $|A|$-tuple $(y^a)_{a \in A} \in \mathbb{R}^{p \times |A|}$ is the collection of p-dimensional vectors y^a marked by the elements of A. Let $\mathcal{Y}$ be the collection of responses (for instance, $\mathcal{Y} = \mathbb{R}$), and we discuss in this paper the ontology problem of predicting a random variable $Y \in \mathcal{Y}$. The sample set here is denoted as n observations $(v_i, y_i) \in \mathbb{R}^p \times \mathcal{Y}, i = 1, \cdots, n$. The empirical risk of sparse ontology vector $\beta \in \mathbb{R}^p$ is denoted by $l(\beta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \beta^T v_i)$, where $l : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}^+$ is a convex and continuously differentiable ontology loss function.

Let $\mathcal{C}$ be a sub-collection of the index set of $\{1, \cdots, p\}$ satisfies $\cup_{C \in \mathcal{C}} C = \{1, \cdots, p\}$. We emphasize here that $\mathcal{C}$ may not be a partition of $\{1, \cdots, p\}$, and it is possible for elements of $\mathcal{C}$ to overlap. Let $(d^C)_{C \in \mathcal{C}}$ be a $|\mathcal{C}|$-tuple of p-dimensional vectors with $d_j^C > 0$ if $j \in C$ and $d_j^C = 0$ otherwise. Hence, the norm Ω for balance part is introduced as

$$\Omega(\beta) = \sum_{C \in \mathcal{C}} (\sum_{j \in C} (d_j^C)^2 |\beta_j|^2)^{\frac{1}{2}} = \sum_{C \in \mathcal{C}} \|d^C \cdot \beta\|_2. \tag{3}$$

The same variable β_j contained in two distrinct index sets $C_1, C_2 \in \mathcal{C}$ is allowed to be weighted differently in C_1 and C_2 (denoted by $d_j^{C_1}$ and $d_j^{C_2}$ respectively).

We consider the following ontology sparse problem:

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{n} \sum_{i=1}^{n} l(y_i, \beta^T v_i) + \mu \Omega(\beta) \tag{4}$$

where $\mu \geq 0$ is an ontology balance parameter. Let $\widehat{\beta}$ be the solution of ontology problem (4) in what follows.

III. Main Ontology Learning Algorithm

A. Backward and Forward Procedure

In this part, we discuss the connection between the nonzero patterns by which the estimated sparse ontology vector $\widehat{\beta}$ is satisfied and the norm Ω which is defined by (3). First, we describe the collection of nonzero patterns, then we manifest the go back and forth from index sets to patterns in terms of forward and backward procedure.

The balance term $\Omega(\widehat{\beta}) = \sum_{C\in\mathcal{C}} \|d^C \cdot \widehat{\beta}\|_2$ is a mixed (l_1, l_2)-norm. From the index set viewpoint, it operates like an l_1-norm and thus Ω leads to index set sparsity. From this point of view, each $d^C \cdot \widehat{\beta}$, and equivalently each β^C is supported to be 0. Moreover, the l_2-norm can't imply advance sparsity within the index sets $C \in \mathcal{C}$. It seems that for a fixed sub collection of index set $\mathcal{C}' \subseteq \mathcal{C}$, the vectors β_C connects the index sets $C \in \mathcal{C}^{'}$ is just equal to 0, and causes a collection of zeros which is the union of these index sets $\cup_{C\in\mathcal{C}'} C$. Thus, the collection of permitted zero patterns is the union closure of $\mathcal{C}$, i.e.,

$$\mathcal{Z} = \{\sup_{C\in\mathcal{C}'} C; \mathcal{C}' \subseteq \mathcal{C}\}.$$

Substitute for considering the collection of zero patterns $\mathcal{Z}$, it is commodious to deal with nonzero patterns, and set

$$\mathcal{P} = \{\cap_{C\in\mathcal{C}'} C^c; \mathcal{C}' \subseteq \mathcal{C}\} = \{Z^c : Z \in \mathcal{Z}\}.$$

It is equivalent to employ $\mathcal{P}$ or $\mathcal{Z}$ to take the complement of each number of these collections.

Suppose that $l : (y, y') \to l(y, y')$ is nonnegative and satisfies that for each pair of $(y, y') \in \mathbb{R} \times \mathbb{R}$, we deduce $\frac{\partial^2 l}{\partial y^2} > 0$ and $\frac{\partial^2 l}{\partial y \partial y'}(y, y') \neq 0$. The Gram matrix of ontology data is denoted as $Q = \frac{1}{n}\sum_{i=1}^n v_i v_i^T$. It is verified that if Q is invertible or $\{1, \cdots, p\} \in \mathcal{G}$ for the ontology optimization problem in (4) with $\mu > 0$, then this problem has a unique solution.

For the zero patterns of the solution of the ontology problem in (4): we suppose that $Y = (y_1, \cdots, y_n)^T$ is a realization of an absolutely continuous probability distribution. The maximal number of linearly independent rows in the matrix $(v_1, \cdots, v_n) \in \mathbb{R}^{p\times n}$ is denoted by k. For $\mu > 0$, any solution of the ontology problem in (4) with at most $k - 1$ nonzero coefficients has a zero pattern in $\mathcal{Z} = \{\cup_{C\in\mathcal{C}} C; \mathcal{C}' \subseteq \mathcal{C}\}$ almost surely. That is to say, if $Y = (y_1, \cdots, y_n)^T$ is a realization of an absolutely continuous probability distribution, then the ontology sparse solutions have a zero pattern in $\mathcal{Z} = \{\cup_{C\in\mathcal{C}'} C : \mathcal{C}' \subseteq \mathcal{C}\}$. Therefore, the ontology problem in (4) has a unique solution if the Gram matrix $\mathbf{Q}$ can be invertible, and its zero pattern is geared to $\mathcal{Z}$.

Following are the four examples on norms associated with our pattern.

Example 1. l_2-norm: $\mathcal{C}$ is consisted of only one element-the entire collection $\{1, \cdots, p\}$, and the collection of permitted nonzero patterns is consisted of $\emptyset$ and the entire collection $\{1, \cdots, p\}$.

Example 2. l_1-norm: $\mathcal{C}$ is the collection of all independent elements thus $\mathcal{P}$ becomes the set of all possible sub-collections.

Example 3. $l_2 - l_1$ mixed norm: $\mathcal{C}$ is the collection of all independent elements and the entire collection $\{1, \cdots, p\}$, and $\mathcal{P}$ becomes the collection of all possible sub-collections.

Example 4. Group version of l_1-norm: $\mathcal{C}$ is consisted of any dividing of $\{1, \cdots, p\}$, and thus we have $\mathcal{P} = \mathcal{Z}$ is the collection of all possible unions of the elements.

In what follows, we focus on the following two problems: (3) proceed from the index sets $\mathcal{C}$, if there is an available method to generate the collection of nonzero patterns $\mathcal{P}$; (2) on the contrary, fixed $\mathcal{P}$, how can the index sets $\mathcal{C}$ and $\Omega(\beta)$ be schemed?

We study the characteristics of the collection of index sets $\mathcal{C}$ and its corresponding collections of patterns $\mathcal{P}$ and $\mathcal{Z}$. The collection of zero patterns $\mathcal{Z}$ (homologous, the collection of nonzero patterns $\mathcal{P}$) is closed under union (homologous, intersection), i.e., for any $K \in \mathbb{N}$ and arbitrary $z_1, \cdots, z_K \in \mathcal{Z}$, $\cup_{k=1}^K z_k \in \mathcal{Z}$ (homologous, $p_1, \cdots, p_K \in \mathcal{P}$, $\cap_{k=1}^K p_k \in \mathcal{P}$). This reveals that we should suppose it is closed under intersection if reverse engineering the collection of nonzero patterns. Or else, the best we can do is to handle its intersection closure.

Given a collection of index sets $\mathcal{C}$, we can define for any sub-collection $I \subseteq \{1, \cdots, p\}$ the $\mathcal{C}$-adapted hull, or simply hull, as:

$$\mathcal{H}(I) = \{\cup_{C\in\mathcal{C}, C\cap I=\emptyset} C\}^c$$

which is the smallest collection in $\mathcal{P}$ including I; we infer $I \subseteq \mathcal{H}(I)$ with equality iff $I \in \mathcal{P}$. Obviously, the hull has a vivid geometrical explanation for special collections $\mathcal{C}$ of groups. For example, the hull of a sub collection $I \subset \{1, \cdots, p\}$ is simply the axis aligned bounding box of I if the collection $\mathcal{C}$ is obtained by all horizontal and vertical half spaces when the variables are organized in a grid with two dimensional. Analogously, the hull is just the regular convex hull if $\mathcal{C}$ is the collection of all half spaces implicit to all potential orientations.

In mathematics and computer science, a directed acyclic graph (for short, DAG) is a directed graph without directed cycles, i.e., it is yielded by a collection of vertices and directed edges, each edge connecting one vertex to another, so that there is no way to start at some vertex v and follow a sequence of edges that eventually loop back to v again. More details for directed acyclic graph can refer to Torres et al., [18] and [19], Marenco et al., [20], Pensar et al., [21] and Kamiyama [22].

Suppose that some priori knowledge about the ontology sparsity structure of a solution $\widehat{\beta}$ of our ontology problem in (4) is imposed. The knowledge can be utilized by restricting the patterns obtained via the Ω norm. Specifically, in terms of an intersection closed collection of zero patterns $\mathcal{Z}$, we can constrict back a minimal set of groups $\mathcal{C}$ via iteratively pruning away in the directed acyclic ontology graph corresponding to $\mathcal{Z}$, and all collections are unions of their parents in ontology graph. Algorithm 1 presents the classical backward and forward procedure, which can be found in many literatures (for example, see Trivisonno et al., [23] and Malvestuto [24]).

Algorithm 1. Backward and Forward Procedure

Part 1. Backward procedure

Input: Intersection closed family of nonzero patterns $\mathcal{P}$.

Output: Set of index sets $\mathcal{C}$.

Initialization: Determine $\mathcal{Z} = \{P^c; P \in \mathcal{P}\}$ and set $\mathcal{C} = \mathcal{Z}$.

Constructers the Hasse diagram for the poset $(\mathcal{Z}, \supset)$.

for $t = \min_{C\in\mathcal{Z}} |C|$ to $\max_{C\in\mathcal{Z}} |C|$ do

  for each vertex $C \in \mathcal{Z}$ such that $|C| = t$ do

if $(\cup_{C\in Children(G)} C = G)$ then
if $(Parents(C) \neq \emptyset)$ then connect children of C to parents of $\mathcal{C}$.
end if
Delete C from $\mathcal{C}$.
end if
end for
end for
Part 2. Forward procedure
Input: Collection of index sets $\mathcal{C} = \{C_1, \cdots, C_M\}$.
Output: Collection of zero patterns $\mathcal{Z}$ and nonzero patterns $\mathcal{P}$.
Initialization: $\mathcal{Z} = \{\emptyset\}$.
for $m = 1$ to M do
$T = \{\emptyset\}$
for each $Z \in \mathcal{Z}$ do if $(C_m \subseteq Z)$ and $(\forall C \in \{C_1, \cdots, C_{m-1}\}, C \subseteq Z \cup C_m) \to C \subseteq Z)$ then
$T \leftarrow T \cup \{Z \cup C_m\}$.
end if
end for
$Z \leftarrow Z \cup T$.
end for
$\mathcal{P} = \{Z^c; Z \in \mathcal{Z}\}$.

The complexity of backward procedure is $O(p|\mathcal{Z}|^2)$ and the complexity of forward procedure is $O(p|\mathcal{Z}||\mathcal{C}|^2)$.

We emphasize here that the collection $\mathcal{Z}$ or $\mathcal{P}$ will not be changed any more after removing a special index set from $\mathcal{C}$. This fact is the main lowdown hiding in the first part of Algorithm 1.

B. Active Ontology Algorithm

In order to moderate the values of p, we deduce a solution for ontology problem (4) by virtue of generic toolboxes for second order cone programming (SOCP) (see Shi et al., [25], Dalalyan [26], Jiang [27], Frangioni and Gentile [28], and Srirangarajan [29] for more details) wtih complexity $O(p^{3.5} + |\mathcal{G}|^{3.5})$, which is not appropriate if p or $|\mathcal{C}|$ are large.

We manifest in this part an active index set algorithm (Algorithm 2) that searches a solution for ontology problem (4) via considering increasingly larger active collections and verifying global optimality for every step.

We consider the following ontology problem for $\lambda > 0$:

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n} l(y_i, \beta^T v_i) + \frac{\lambda}{2}[\Omega(\beta)]^2. \quad (5)$$

In active index set technologies, we build incrementally the set of nonzero variables and use J to express it, and the ontology problem is solved only for this collection of variables, adding the constraint $\beta_{J^c} = 0$ to ontology problem (5). Let $l(\beta) = \frac{1}{n}\sum_{i=1}^{n} l(y_i, \beta^T v_i)$ be the empirical ontology risk (which is supposed to be convex and continuously differentiable) and let l^* be its Fenchel conjugate denoted by

$$l^*(u) = \sup_{\beta\in\mathbb{R}^p}\{\beta^T u - L(\beta)\}.$$

We use $l_J(\beta_J) = l(\widehat{\beta})$ to denote the restriction of l to $\mathbb{R}^{|J|}$ for $\beta_J = \beta_J$ and $\widehat{\beta}_{J^c} = 0$ with Fenchel conjugate l_J^*. However, in general, we do not have the property that $l_J^*(\kappa_J) = l^*(\tilde{\kappa})$ for $\tilde{\kappa}_J = \kappa_J$ and $\tilde{\kappa}_{J^c} = 0$.

For a potential active index set $J \subseteq \{1, \cdots, p\}$ which belongs to the collection of allowed nonzero patterns $\mathcal{P}$, we use $\mathcal{C}_J$ to denote the set of active index sets, i.e., the collection of index set $C \in \mathcal{C}$ satisfies $C \cap J \neq \emptyset$. The balance part Ω_J on $\mathbb{R}^{|J|}$ is defined by

$$\Omega_J(\beta_J) = \sum_{C\in\mathcal{C}} \|d_J^C \cdot \beta_J\|_2 = \sum_{C\in\mathcal{C}_J} \|d_J^C \cdot \beta_J\|_2,$$

and its dual norm $\Omega_J^*(\kappa_J) = \max_{\Omega_J(\beta_J)\leq 1} \beta_J^T \kappa_J$ also introduced on $\mathbb{R}^{|J|}$.

Let $J \subseteq \{1, \cdots, p\}$. The following two ontology problems

$$\min_{\beta_J\in\mathbb{R}^{|J|}} l_J(\beta_J) + \frac{\lambda}{2}|\Omega_J(\beta_J)|^2, \quad (6)$$

$$\max_{\beta_J\in\mathbb{R}^{|J|}} -l_J^*(-\kappa_J) - \frac{1}{2\lambda}|\Omega_J^*(\beta_J)|^2, \quad (7)$$

are dual to each other and strong duality estabilishes. The pair of primal dual variables $\{\beta_J, \kappa_J\}$ is optimal if and only if we obtain $\kappa_J = -\bigtriangledown l_J(\beta_J)$ and $\beta_J^T\kappa_J = \frac{1}{\lambda}|\Omega_J^*(\kappa_J)|^2 = \lambda|\Omega_J(\beta_J)|^2$. This fact shows the optimization ontology problem is dual to the reduced ontology problem.

It enables us to deduce the duality gap for the optimization ontology problem (6) which is reduced to the active index set of variables J. In reality, such duality gap can be vanish if we successively solve ontology problem (6) for increasingly larger active index sets J. Starting from the optimality of the ontology problem in (6), we study how we can govern the optimality or equal the duality gap for the full ontology problem in (5). The duality gap of the optimization ontology problem in (6) can be precisely expressed by a sum of two nonnegative parts:

$$\begin{aligned} & l_J(\beta_J) + l_J^*(-\kappa_J) + \frac{\lambda}{2}[\Omega_J(\beta_J)]^2 + \frac{1}{2\lambda}[\Omega_J^*(\kappa_J)]^2 \\ = \; & \{l_J(\beta_J) + l_J^*(-\kappa_J) + \beta_J^T\kappa_J\} \\ & + \{\frac{\lambda}{2}[\Omega_J(\beta_J)]^2 + \frac{1}{2\lambda}[\Omega_J^*(\kappa_J)]^2 - \beta_J^T\kappa_J\}. \end{aligned}$$

This duality gap can be regarded as the sum of two duality gaps, corresponding to l_J and Ω_J, respectively. Hence, if we get a primal candidate β_J and select $\kappa_J = -\bigtriangledown l_J(\beta_J)$, the duality gap relative to l_J disappears and the total duality gap reduces to

$$\frac{\lambda}{2}[\Omega_J(\beta_J)]^2 + \frac{1}{2\lambda}[\Omega_J^*(\kappa_J)]^2 - \beta_J^T\kappa_J.$$

For verifying the reduced solution β_J is optimal for the full ontology problem in (5). Padding β_J with zeros on J^c to determine β and calculate $\kappa = -\bigtriangledown l(\beta)$ with $\kappa_J = -\bigtriangledown_J (\beta_J)$. For fixed candidate pair of primal and dual variables $\{\beta, \kappa\}$, we yield a duality gap for the full ontology problem in (5) equal to

$$\begin{aligned} & \frac{\lambda}{2}[\Omega_J(\beta_J)]^2 + \frac{1}{2\lambda}[\Omega_J^*(\kappa_J)]^2 - \beta_J^T\kappa_J \\ = \; & \frac{1}{2\lambda}([\Omega^*(\kappa)]^2 - \lambda\beta_J^T\kappa_J). \end{aligned}$$

We can explain the active index set algorithm as a walk through the directed acyclic graph of nonzero patterns permitted by the norm Ω. The parents $\Pi_{\mathcal{P}}(J)$ of J in directed acyclic ontology graph are exactly the patterns containing the variables that may enter the active index set at the next iteration of Algorithm 2. The index sets that are exactly at the boundaries of the active collection are $\mathcal{F}_J = \{C \in (\mathcal{C}_J)^c; C' \in (\mathcal{C}_J)^c, C \subseteq C'\}$, i.e., the index sets that are not contained by any other inactive index sets. In addition, the active index set may usually be increased only to guarantee the obtained solution which is optimal in Algorithm 2.

Algorithm 2. Active index set algorithm

Input: Data $\{(v_i, y_i), i = 1, \cdots, n\}$, balance parameter λ, maximum number of variables s and duality gap precision ε.

Initialization: $J = \{\emptyset\}$, $\widehat{\beta} = 0$.

While $(\max_{K\in\Pi_{\mathcal{P}}(J)} \frac{\|\nabla l(\beta)_{K-J}\|_2}{\sum_{H\in\mathcal{C}_K-\mathcal{C}_J}\|d^H_{K-J}\|_\infty} \leq \{-\lambda\beta^T \nabla l(\beta)\}^{\frac{1}{2}}$ (here β is the optimal for the full ontology problem (5) is not satisfied) and $(|J| \leq s)$ do

Replace J via infringing $K \in \Pi_{\mathcal{P}}(J)$ in $\max_{K\in\Pi_{\mathcal{P}}(J)} \frac{\|\nabla l(\beta)_{K-J}\|_2}{\sum_{H\in\mathcal{C}_K-\mathcal{C}_J}\|d^H_{K-J}\|_\infty} \leq \{-\lambda\beta^T \nabla l(\beta)\}^{\frac{1}{2}}$ (here β is the optimal for the full ontology problem (5)).

Solve the reduced problem $\min_{\beta_J\in\mathbb{R}^{|J|}} l_J(\beta_J) + \frac{\lambda}{2}[\Omega_J(\beta_J)]^2$ to get $\widehat{\beta}$.

End while

While $\max_{C\in\mathcal{F}_J}\{\sum_{k\in C}\{\frac{\nabla l(\beta)_k}{\sum_{k\in H, H\in(\mathcal{C}_J)^c} d^H_k}\}^2\}^{\frac{1}{2}} \leq \{\lambda(2\varepsilon - \beta^T \nabla l(\beta))\}^{\frac{1}{2}}$ is not satisfied and $|J| \leq s$ do

Revise J in terms of the following procedure:

(begin procedure) Let $C \in \mathcal{F}_J$ be the index set that infringes $\max_{C\in\mathcal{C}_J}\{\sum_{k\in C}\{\frac{\nabla l(\beta)_k}{\sum_{k\in H, H\in(\mathcal{C}_J)^c} d^H_k}\}^2\}^{\frac{1}{2}} \leq \{\lambda(2\varepsilon - \beta^T \nabla l(\beta))\}^{\frac{1}{2}}$ most.

if $(C \cap (\cup_{K\in\Pi_{\mathcal{P}}(J)}K) \neq \emptyset)$ then
  for $K \in \Pi_{\mathcal{P}}(J)$ such that $K \cap C \neq \emptyset$ do
    $J \leftarrow J \cap K$.
  end for
else
  for $H \in \mathcal{F}_J$ such that $H \cap C \neq \emptyset$ do
    for $K \in \Pi_{\mathcal{P}}(J)$ such that $K \cap C \neq \emptyset$ do
      $J \leftarrow J \cap K$.
    end for
  end for
end if (end procedure)

Solve the ontology problem $\min_{\beta_J\in\mathbb{R}^{|J|}} l_J(\beta_J) + \frac{\lambda}{2}[\Omega_J(\beta_J)]^2$ to get $\widehat{\beta}$.

End while

Output: active index set J, loading vector $\widehat{\beta}$.

If the number of active variables is upper bounded by $s \ll p$, the time complexity of Algorithm 2 is the sum of: 1) the calculation of the gradient, $O(snp)$ for the square loss; 2) if the underlying solver called upon by the active index set algorithm is a standard SOCP solver, $O(s\max_{J\in\mathcal{P},|J|\leq s}|\mathcal{C}_J|^{3.5} + s^{4.5})$; 3) t_1 times the computation of $\max_{K\in\Pi_{\mathcal{P}}(J)} \frac{\|\nabla l(\beta)_{K-J}\|_2}{\sum_{H\in\mathcal{C}_K-\mathcal{C}_J}\|d^H_{K-J}\|_\infty} \leq \{-\lambda\beta^T \nabla l(\beta)\}^{\frac{1}{2}}$, that is $O(t_1(s2|\Theta| + p|\mathcal{G}| + sn^2_\theta) + p|\mathcal{G}|) = O(t_1 p|\mathcal{G}|)$; 4) t_2 times the computation of $\max_{C\in\mathcal{C}_J}\{\sum_{k\in C}\{\frac{\nabla l(\beta)_k}{\sum_{k\in H, H\in(\mathcal{C}_J)^c} d^H_k}\}^2\}^{\frac{1}{2}} \leq \{\lambda(2\varepsilon - \beta^T \nabla l(\beta))\}^{\frac{1}{2}}$, that is $O(t_2(s2^{|\Theta|} + p|\mathcal{C}| + |\Theta|^2 + |\Theta|p + p|\mathcal{C}|)) = O(t_2 p|\mathcal{C}|)$, with $t_1 + t_2 \leq s$.

We finally obtain complexity with a leading term in $O(sp|\mathcal{C}| + s\max_{J\in\mathcal{P},|J|\leq s}|\mathcal{C}_J|^{3.5} + s^{4.5})$.

Furthermore, after careful observation, we found that several index sets can be used several times in the implement. This is a phenomenon of overlapping. In reality, in our ontology setting, these overlaps can be controlled by means of selecting the weights $(d^C)_{C\in\mathcal{C}}$ which have been taken into account so that several elements in overlapping index sets are punished many times.

IV. Experiments

In this section, five simulation experiments concerning ontology measure and ontology mapping are designed respectively. In these five experiments, we mainly test the effectiveness of Algorithm 2. After the sparse vector β is obtained, and the ontology function f is then deduced via (1). In our experiment, the ontology loss function is selected as the square loss. To make comparisons as exact as possible, the Algorithm 2 was ran in C++, by means of available LAPACK and BLAS libraries for linear algebra and operation computations. The following five experiments are implemented on a double-core CPU with a memory of 8GB.

A. Ontology similarity measure experiment on plant data

We use O_1, a plant "PO" ontology in the first experiment, and it was constructed in www.plantontology.org. The structure of O_1 presented in Fig. 1. $P@N$ (Precision Ratio see Craswell and Hawking [30]) is used to measure the quality of the experiment data.

At first, experts give the closest N concepts for every vertex on the ontology graph in plant field. Then the first N concepts for every vertex on ontology graph are gained by the algorithm 2, and the precision ratio can be computed. Or rather, for vertex v and the given integer $N > 0$. Let $Sim^{N,\text{expert}}_v$ be the set of vertices determined by experts in which N vertices having the most similarity of v are included. Let

$$v^1_v = \operatorname*{argmin}_{v'\in V(G)-v}\{|f(v) - f(v')|\},$$

$$v^2_v = \operatorname*{argmin}_{v'\in V(G)-\{v,v^1_v\}}\{|f(v) - f(v')|\},$$

$$\cdots$$

$$v^N_v = \operatorname*{argmin}_{v'\in V(G)-\{v,v^1_v,\cdots,v^{N-1}_v\}}\{|f(v) - f(v')|\},$$

and

$$Sim^{N,\text{algorithm}}_v = \{v^1_v, v^2_v, \cdots, v^N_v\}.$$

Then the precision ratio for vertex v is denoted by

$$\text{Pre}^N_v = \frac{|Sim^{N,\text{algorithm}}_v \cap Sim^{N,\text{expert}}_v|}{N}.$$

The $P@N$ average precision ratio for ontology graph G is then stated as

$$\text{Pre}^N_G = \frac{\sum_{v\in V(G)}\text{Pre}^N_v}{|V(G)|}.$$

Meanwhile, ontology methods in [15], [16] and [17] are applied to the "PO" ontology. Then after getting the average precision ratio by means of these three algorithms, we compare the results with algorithm 2. Parts of the data can be referred to Table 1.

When N = 3, 5 or 10, compared with the precision ratio determined by algorithms proposed in [15], [16] and [17], the precision ratio gained from our algorithms are a

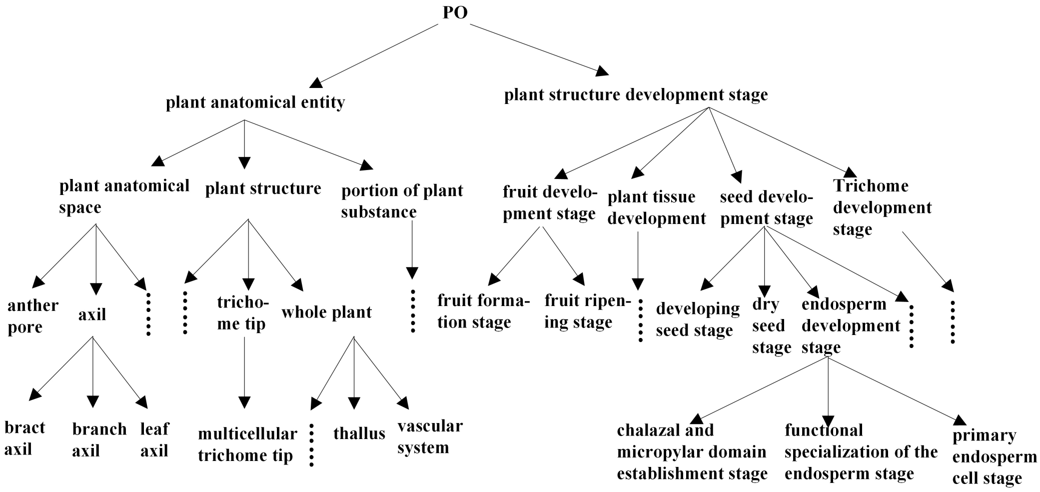

Fig. 1. The Structure of "PO" Ontology.

TABLE I
TAB. 1.THE EXPERIMENT RESULTS OF ONTOLOGY SIMILARITY MEASURE

	P@3 average precision ratio	P@5 average precision ratio	P@10 average precision ratio
Our Algorithm	0.5292	0.6388	0.8275
Algorithm in [15]	0.4549	0.5117	0.5859
Algorithm in [16]	0.4282	0.4849	0.5632
Algorithm in [17]	0.4831	0.5635	0.6871

little bit higher. Furthermore, the precision ratios show the tendency to increase apparently as N increases. As a result, our algorithms turn out to be better and more effective than those raised by [15], [16] and [17].

B. Ontology mapping experiment on humanoid robotics data

We use "humanoid robotics" ontologies O_2 and O_3 in the second experiment. The structure of O_2 and O_3 are respectively presented in Fig. 2 and Fig. 3. The ontology O_2 presents the leg joint structure of bionic walking device for six-legged robot. And the ontology O_3 presents the exoskeleton frame of a robot with wearable and power assisted lower extremities.

We set the experiment with the aim to get ontology mapping between O_2 and O_3. We also take $P@N$ Precision Ratio as a measure for the quality of experiment. After applying ontology algorithms in [31], [16] and [17] on "humanoid robotics" ontology and getting the average precision ratio, we compare the precision ratios gained from these three methods. Some results can refer to Table 2.

When $N = 1$, 3 or 5, compared with the precision ratios determined by algorithms proposed in [31], [16] and [17], the precision ratios gained from our new ontology algorithm are higher. Furthermore, the precision ratios show the tendency to increase apparently as N increases. As a result, our algorithms turn out to be better and more effective than those raised by [31], [16] and [17].

C. Ontology similarity measure experiment on biology data

We use gene "GO" ontology O_4 in the third experiment, and it was constructed in the website http: //www. geneontology. The structure of O_4 is presented in Figure 4. Again, we

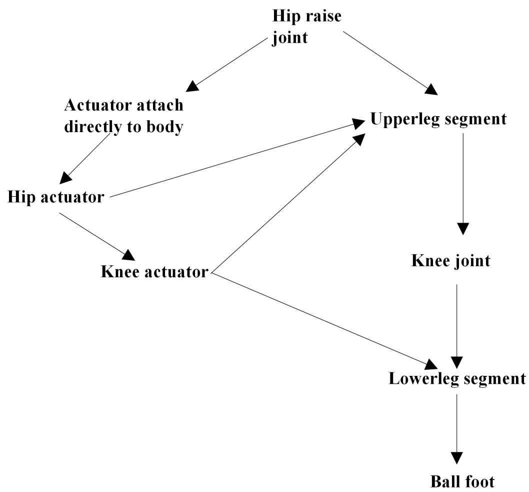

Fig. 2. 'Humanoid Robotics" Ontology O_2.

chosse $P@N$ as a measure for the quality of the experiment data. Then the ontology methods in [16], [17] and [32] are applied to the "GO" ontology. Then after getting the average precision ratio by means of these three algorithms, we compare the results with algorithm 2. Parts of the data can refer to Table 3.

When $N = 3$, 5 or 10, compared with the precision ratios determined by algorithms proposed in [16], [17] and [32], the precision ratios gained from our ontology algorithms are higher. Furthermore, the precision ratios show the tendency to increase apparently as N increases. As a result, our algorithms turn out to be better and more effective than those

TABLE II
TAB. 2. THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING

	P@1 average precision ratio	P@3 average precision ratio	P@5 average precision ratio
Our Algorithm	0.2778	0.4815	0.6889
Algorithm in [31]	0.2778	0.4815	0.5444
Algorithm in [16]	0.2222	0.4074	0.4889
Algorithm in [17]	0.2778	0.4630	0.5333

TABLE III
TAB. 3. THE EXPERIMENT RESULTS OF ONTOLOGY SIMILARITY MEASURE

	P@3 average precision ratio	P@5 average precision ratio	P@10 average precision ratio	P@20 average precision ratio
Our Algorithm	0.4963	0.6275	0.7418	0.8291
Algorithm in [16]	0.4638	0.5348	0.6234	0.7459
Algorithm in [17]	0.4356	0.4938	0.5647	0.7194
Algorithm in [32]	0.4213	0.5183	0.6019	0.7239

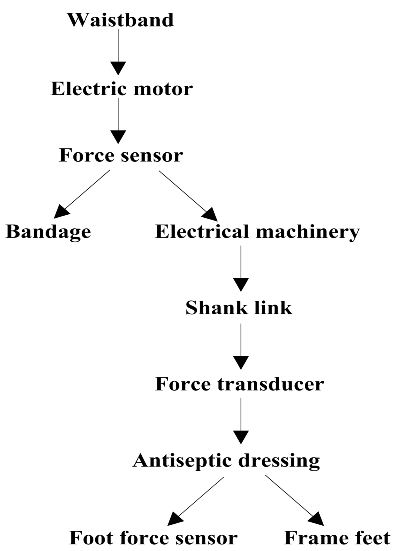

Fig. 3. "Humanoid Robotics" Ontology O_3.

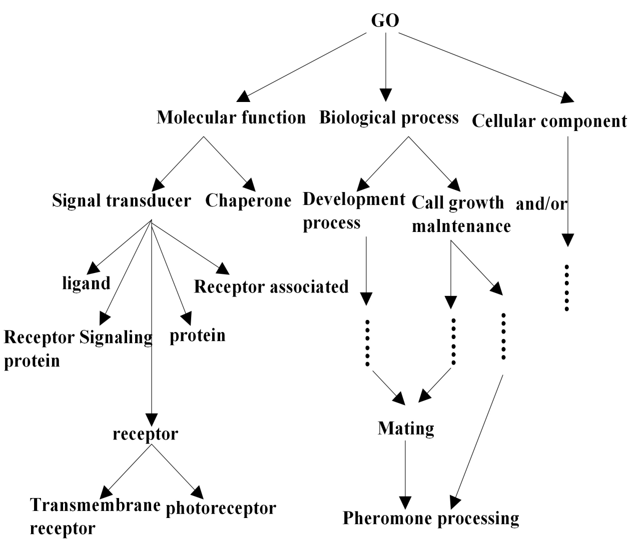

Fig. 4. The Structure of "GO" Ontology.

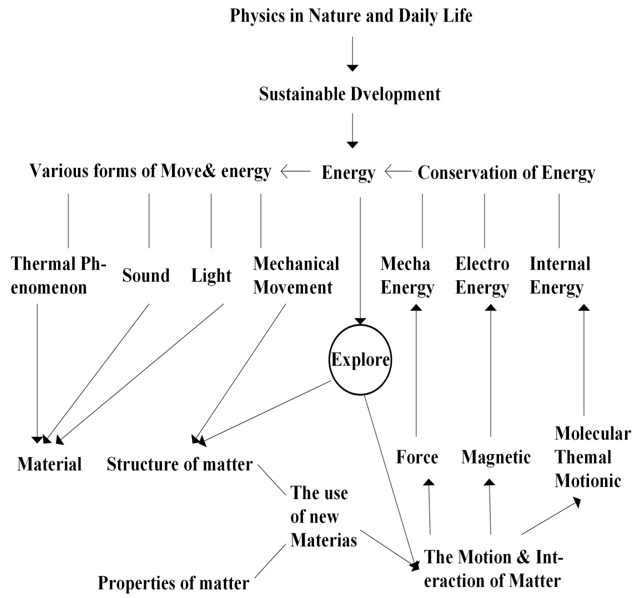

Fig. 5. "Physics Education" Ontology O_5.

raised by [16], [17] and [32].

D. Ontology mapping experiment on physics education data

We use "physics education" ontologies O_5 and O_6 in the fourth experiment. The structures of O_5 and O_6 are respectively represented in Fig. 5 and Fig. 6.

We set the experiment with the aim to give ontology mapping between O_5 and O_6. We take $P@N$ precision ratio as a measure for the quality of the experiment. This time we apply ontology algorithms in [16], [17] and [33] on "physics education" ontology. Then we compare the precision ratio gotten from the three methods. Some results can refer to Table 4.

When $N = 1$, 3 or 5, compared with the precision ratio determined by algorithms proposed in [16], [17] and [33], the precision ratio in terms of our new ontology mapping algorithms are much higher. Furthermore, the precision ratios show the tendency to increase apparently as N increases.

TABLE IV
TAB. 4. THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING

	$P@1$ average precision ratio	$P@3$ average precision ratio	$P@5$ average precision ratio
Our Algorithm	0.6774	0.7634	0.9097
Algorithm in [16]	0.6129	0.7312	0.7935
Algorithm in [17]	0.6913	0.7556	0.8452
Algorithm in [33]	0.6774	0.7742	0.8968

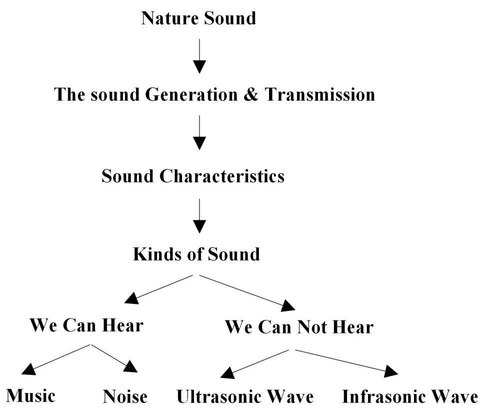

Fig. 6. "Physics Education" Ontology O_6.

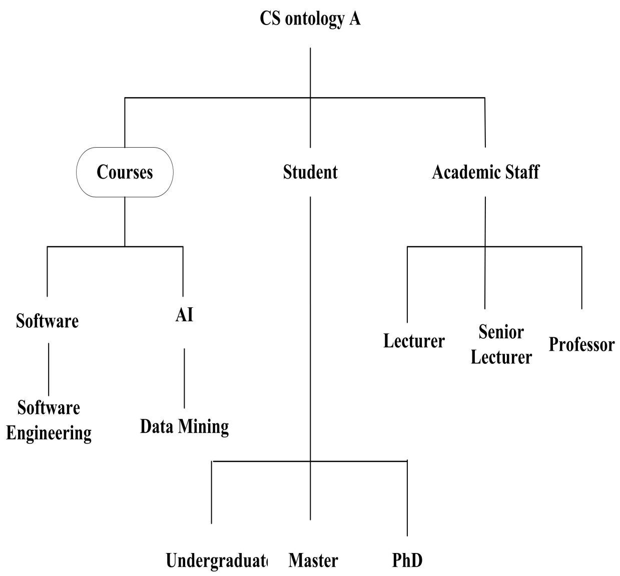

Fig. 7. "University" Ontology O_7.

As a result, our algorithms turn out to be better and more effective than those raised by [16], [17] and [33].

E. Ontology mapping experiment on university data

We use "University" ontologies O_7 and O_8 in the last experiment. The structures of O_7 and O_8 are respectively presented in Fig. 7 and Fig. 8.

We set the experiment with the aim to give ontology mapping between O_7 and O_8. We take $P@N$ precision ratio as a criterion to measure the quality of the experiment. This time we apply ontology algorithms in [15], [16] and [17] on "University" ontology. Then we compare the precision ratios

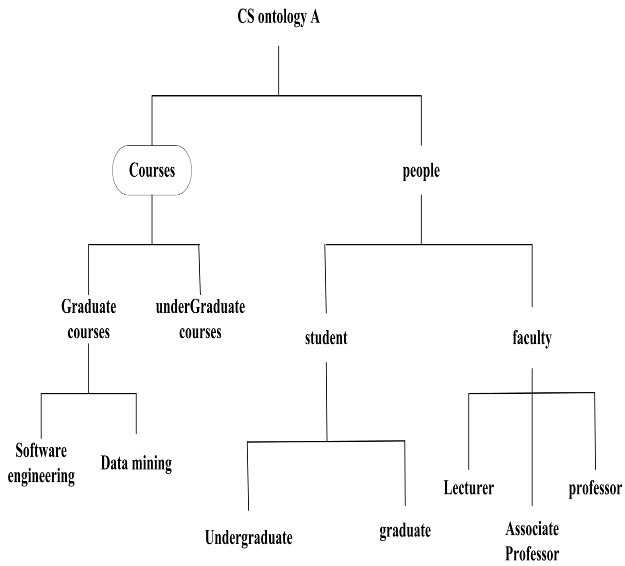

Fig. 8. "University" Ontology O_8.

gotten from the three methods. Some results can be referred to Table 5.

When $N = 1$, 3 or 5, compared with the precision ratios determined by algorithms proposed in [15], [16] and [17], the precision ratios in terms of our new ontology mapping algorithms are much higher. Furthermore, the precision ratios show the tendency to increase apparently as N increases. As a result, our algorithms turn out to be better and more effective than those raised by [15], [16] and [17].

V. CONCLUSIONS

Ontology, as a model of big data structural representation and storage, has been widely employed in various disciplines, and has been proved to have high efficiency. The nature of ontology application algorithms is deducing the similarity measure function between vertices on specific ontology graph. In recent years, all kinds of machine learning approaches have been introduced for ontology similarity measure computation and ontology mapping construction. One effective ontology learning technology is mapping each vertex to a real number using ontology function $f : V \to \mathbb{R}$, and then the similarity between v_i and v_j is judged by $|f(v_i)-f(v_j)|$. Such learning method is suitable for ontology computation with big data and arouses great concern among the researchers.

In this article, we focus on the feature extraction of ontology vector and report a new framework for ontology sparse vector learning algorithm in terms of SOCP. Finally,

TABLE V
TAB. 5. THE EXPERIMENT RESULTS OF ONTOLOGY MAPPING

	P@1 average precision ratio	P@3 average precision ratio	P@5 average precision ratio
Our Algorithm	0.5714	0.6667	0.7143
Algorithm in [15]	0.5000	0.5952	0.6857
Algorithm in [16]	0.4286	0.5238	0.6071
Algorithm in [17]	0.5714	0.6429	0.6500

simulation data from four experiments reveal that our new algorithm has high efficiency in biology, physics education, plant science, humanoid robotics and university applications. The new technology contributes to the state of art for ontology application and illustrates the promising prospects of application for multiple disciplines.

REFERENCES

[1] J. M. Przydzial, B. Bhhatarai, and A. Koleti, "GPCR Ontology: Development and Application of A *G* Protein-Coupled Receptor Pharmacology Knowledge Framework," *Bioinformatics*, vol. 29, no. 24, pp. 3211-3219, 2013.

[2] S. Koehler, S. C. Doelken, and C. J. Mungall, "The Human Phenotype Ontology Project: Linking Molecular Biology and Disease Through Phenotype Data," *Nucleic Acids Research*, vol. 42, no. D1, pp. 966-974, 2014.

[3] M. Ivanovic and Z. Budimac, "An Overview of Ontologies and Data Resources in Medical Domains," *Expert Systerms and Applications*, vol. 41, no. 11, pp. 5158-5166, 2014.

[4] A. Hristoskova, V. Sakkalis, and G. Zacharioudakis, "Ontology-Driven Monitoring of Patient's Vital Signs Enabling Personalized Medical Detection and Alert," *Sensors*, vol. 14, no. 1, pp. 1598-1628, 2014.

[5] M. A. Kabir, J. Han, and J. Yu, "User-Centric Social Context Information Management: An Ontology-Based Approach and Platform," *Personal and Ubiquitous Computing*, vol. 18, no. 5, pp. 1061-1083, 2014.

[6] Y. L. Ma, L. Liu, K. Lu, B. H. Jin, and X. J. Liu, "A Graph Derivation Based Approach for Measuring and Comparing Structural Semantics of Ontologies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 5, pp. 1039-1052, 2014.

[7] Z. Li, H. S. Guo, Y. S. Yuan, and L. B. Sun, "Ontology Representation of Online Shopping Customers Knowledge in Enterprise Information," *Applied Mechanics and Materials*, vol. 483, pp. 603-606, 2014.

[8] R. Santodomingo, S. Rohjans, M. Uslar, J. A. Rodriguez-Mondejar, and M. A. Sanz-Bobi, "Ontology Matching System for Future Energy Smart Grids," *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 242-257, 2014.

[9] T. Pizzuti, G. Mirabelli, M. A. Sanz-Bobi, and F. Gomez-Gonzalez, "Food Track & Trace Ontology for Helping the Food Traceability Control," *Journal of Food Engineering*, vol. 120, no. 1, pp. 17-30, 2014.

[10] N. Lasierra, A. Alesanco, and J. Garcia, "Designing An Architecture for Monitoring Patients at Home: Ontologies and Web Services for Clinical and Technical Management Integration," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 3, pp. 896-906, 2014.

[11] M. Tovar and D. Pinto, Azucena Montes, Gabriel Gonzalez, and Darnes Vilarino, "Identification of Ontological Relations in Domain Corpus Using Formal Concept Analysis," *Engineering Letters*, vol. 23, no.2, pp. 72-76, 2015.

[12] V. Gopal and N. S. Gowri Ganesh, "Ontology Based Search Engine Enhancer," *IAENG International Journal of Computer Science*, vol. 35, no. 3, pp. 413-420, 2008.

[13] W. Gao and L. Shi, "Szeged Related Indices of Unilateral Polyomino Chain and Unilateral Hexagonal Chain," *IAENG International Journal of Applied Mathematics*, vol. 45, no.2, pp. 138-150, 2015.

[14] W. Gao, L. L. Zhu and Y. Guo, "Multi-dividing Infinite Push Ontology Algorithm," *Engineering Letters*, vol. 23, no. 3, pp. 132-139, 2015.

[15] Y. Y. Wang, W. Gao, Y. G. Zhang and Y. Gao, "Ontology Similarity Computation Use Ranking Learning Method," *The 3rd International Conference on Computational Intelligence and Industrial Application*, Wuhan, China, 2010, pp. 20–22.

[16] X. Huang, T. W. Xu, W. Gao and Z. Y. Jia, "Ontology Similarity Measure and Ontology Mapping Via Fast Ranking Method," *International Journal of Applied Physics and Mathematics*, vol. 1, no. 1, pp. 54-59, 2011.

[17] W. Gao and L. Liang, "Ontology Similarity Measure by Optimizing NDCG Measure and Application in Physics Education," *Future Communication, Computing, Control and Management*, vol. 142, pp. 415-421, 2011.

[18] P. Torres, J. van Wingerden, M. Verhaegen, "PO-MOESP subspace identification of directed acyclic graphs with unknown topology," *Automatica*, vol. 53, pp. 60-71, 2015.

[19] P. Torres, J. van Wingerden, M. Verhaegen, "Hierarchical subspace identification of directed acyclic graphs," *Internation Journal of Control*, vol. 88, no. 1, pp. 123-137, 2015.

[20] J. Marenco, M. Mydlarz, D. Severin, "Topological additive numbering of directed acyclic graphs," *Information Processing Letters*, vol. 115, no. 2, pp. 199-202, 2015.

[21] J. Pensar, H. Nyman, T. Koski, J. Corander, "Labeled directed acyclic graphs: a generalization of context-specific independence in directed graphical models," *Data Mining and Knowledge Discovery*, vol. 29, no. 2, pp. 503-533, 2015.

[22] N. Kamiyama, "The nucleolus of arborescence games in directed acyclic graphs," *Operations Research Letters*, vol. 43, no. 1, pp. 89-92, 2015.

[23] R. Trivisonno, R. Guerzoni, I. Vaishnavi, D. Soldani, "SDN-based 5G mobile networks: architecture, functions, procedures and backward compatibility," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 1, pp. 82-92, 2015.

[24] M. F. Malvestuto, "A backward selection procedure for approximating a discrete probability distribution by decomposable models," *Kybernetika*, vol. 48, no. 5, pp. 825-844, 2012.

[25] Q. J. Shi, W. Q. Xu, T. H. Chang, Y. C. Wang, E. B. Song, "Joint beamforming and power splitting for MISO interference channel with SWIPT: an SOCP relaxation and decentralized algorithm," *IEEE Transactions on Signal Processing*, vol. 62, no. 23, pp. 6194-6208, 2014.

[26] A. S. Dalalyan, "SOCP based variance free dantzig selector with application to robust estimation," *Comptes Rendus Mathematique*, vol. 350, no. 15-16, pp. 785-788, 2012.

[27] A. Jiang, H. K. Kwan, Y. P. Zhu, "Peak-error-constrained sparse FIR filter design using iterative SOCP," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4035-4044, 2012.

[28] A. Frangioni, C. Gentile, "A computational comparison of reformulations of the perspective relaxation: SOCP vs. cutting planes," *Operations Research Letters*, vol. 37, no. 3, pp. 206-210, 2009.

[29] S. Srirangarajan, A. H. Tewfik, Z. Q. Luo, "Distributed Sensor Network Localization Using SOCP Relaxation", *IEEE Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4886-4895, 2008.

[30] N. Craswell and D. Hawking, "Overview of the TREC 2003 Web Track," *In proceedings of the Twelfth Text Retrieval Conference*, Gaithersburg, Maryland, NIST Special Publication, 2003, pp. 78-92.

[31] W. Gao and M. H. Lan, "Ontology Mapping Algorithm Based on Ranking Learning Method," *Microelectronics and Computer*, vol. 28, no. 9, pp. 59-61, 2011.

[32] Y. Gao and W. Gao, "Ontology Similarity Measure and Ontology Mapping Via Learning Optimization Similarity Function," *International Journal of Machine Learning and Computing*, vol. 2, no. 2, pp. 107-112, 2012.

[33] W. Gao, Y. Gao, and L. Liang, "Diffusion and Harmonic Analysis on Hypergraph and Application in Ontology Similarity Measure and Ontology Mapping," *Journal of Chemical and Pharmaceutical Research*, vol. 5, no. 9, pp. 592-598, 2013.

Meihui Lan, female, was born in the city of Yiliang, Yunnan Province, China on Aug. 25, 1982. In 2006, she got bachelor degree in department of computer science and technology, Yunnan normal university, China. Then, she was enrolled in department of computer software and theory in the same

school, and got Master degree and received a software designer certificate there in 2009.

Now, she acts as a lecturer in the department of computer science and engineering, Qujing Normal University. As a researcher in computer science, her interests are Natural language processing and Machine learning.

Jian Xu acts as a associate professor in the department of computer science and engineering, Qujing Normal University. As a researcher in computer science, his interests are Computer Education and Information Retrieval.

Wei Gao, male, was born in the city of Shaoxing, Zhejiang Province, China on Feb. 13, 1981. He got two bachelor degrees on computer science from Zhejiang industrial university in 2004 and mathematics education from College of Zhejiang education in 2006. Then, he was 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.

Now, he acts as a 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.