Available online www.jocpr.com

Journal of Chemical and Pharmaceutical Research, 2013, 5(9):592-598

Research Article

ISSN : 0975-7384
CODEN(USA) : JCPRC5

Diffusion and harmonic analysis on hypergraph and application in ontology similarity measure and ontology mapping

W. Gao[1], Y. Gao[2] and L. Liang[1]

[1]School of Information Science and Technology, Yunnan Normal University, Kunming, China
[2]Editorial Department of Yunnan Normal University, Kunming, China

ABSTRACT

Ontology similarity calculation and ontology mapping are crucial research topics in information retrieval. Ontology, as a concept structure model, is widely used in biology, physics, geography and social sciences. In this paper, we present new algorithms for ontology similarity measurement and ontology mapping using harmonic analysis and diffusion regularization on hypergraph. The optimal function gets from new algorithms manifests good smoothness and well reflects the structure of the ontology graph. Two experimental results show that the proposed new technologies have high accuracy and efficiency on ontology similarity calculation and ontology mapping in certain applications.

Keywords: Ontology, ontology mapping, diffusion process, harmonic analysis, hypergraph

INTRODUCTION

Ontology is described as the concept structure in computer science which has raised many attentions from researchers. In information retrieval, ontology has been used to compute semantic similarity (for instance, see [1]) and search extensions for concepts. Every vertex on an ontology graph represents a concept; a user searches for a concept *A*, will return similarities concepts of *A* as search extensions. Let *G* be a graph corresponding to ontology *O*, the goal of ontology similarity measure is to approach a similarity function which maps each pair of vertices to a real number. Choose the parameter $M\in\mathbb{R}^+$, the concepts *A* and *B* have high similarity if *Sim*(*A*,*B*)>*M*. Choose the parameter $M\in\mathbb{R}^+$, let *A*,*B* be two concepts on ontology and *Sim*(*A*,*B*)>*M*, then return *B* as retrieval expand when search concept *A*. Therefore, the quality of similarity functions plays an important role in such applications. Some effective methods for ontology similarity calculation are given by [2], [3], and [4].

Let graphs G_1, G_2, …, G_k corresponding to ontologies O_1, O_2,…, O_k, respectively, and $G=G_1+G_2+\ldots+G_k$. For every vertex $v\in V(G_i)$, where $1\le i\le k$, the goal of ontology mapping is finding similarity vertices from G-G_i. From this point of view, the ontology mapping problem is can be regard as ontology similarity measure. The key trick for ontology similarity measure and ontology mapping is to find the best similarity function *Sim*: $V\times V\rightarrow\mathbb{R}^+\cup\{0\}$, which maps each pair of vertices to a non-negative real number.

One trick to design similarity functions for ontology applications is followed from the graph learning method. With such technology, we obtain the optimal function, which maps each vertex in ontology graph or multi-ontology graph into a real number. By calculating the difference between the real number of two vertices, we determine the similarity between their correspond concepts.

There are several theory analyses for ontology algorithms. [5] studied the uniform stability of multi-dividing ontology algorithm and gave the generalization bounds for stable multi-dividing ontology algorithms. [6] researched the strong and weak stability of multi-dividing ontology algorithm. [7] learned some characteristics for such ontology algorithm. [8] studied the multi-dividing ontology algorithm from a theoretical view. It is highlighted that empirical multi-dividing ontology model can be expressed as conditional linear statistical, and an approximation result is achieved based on projection method. [9] presented the characteristics of best ontology score function among piece constant ontology score functions. [10] investigated the upper bound and lower bound minimax learning rate are obtained based on low noise assumptions. Recently, [11] and [12] proposed splitting trick for vertex partition of AUC criterion multi-dividing setting and presented several statistic results.

In this paper, we present the harmonic analysis and diffusion on hypergraph data and apply it in the field of ontology applications. The purpose of such trick is to improve the smoothness of the objective function and reflect the intrinsic structural characteristics of the ontology graph. The organization of this paper is described as follows: we first present the technology of harmonic analysis and diffusion for hypergraph setting. Next, we describe the new ontology similarity measure and ontology mapping algorithms using the tricks that we show in next section. Then, these algorithms are explored in the simulation studies in biology ontology and physical education ontology, respectively.

HARMONIC ANALYSIS AND DIFFUSION ON HYPERGRAPH DATA

Diffusion on discrete data and harmonic analysis are important tricks in the implement of machine learning algorithms, especially in semi-supervised and transductive learning settings. The success of such algorithms relies heavily on the assumption that the optimal function to be learned is smooth with respect to the geometry of the data. In this section, we present a method for modifying the given geometry on hypergraph so the optimal function to be learned is smoother with respect to the modified geometry, and hence more amenable to deal with using harmonic analysis methods. It helps to handle the relationships between smoothness, sparsity and evolution of heat.

The smoothing operation of the diffusion not only depended on the geometry of the space, but also relied on the geometry and the characters of the feature function *f*. Hence, it is necessary to modify the geometry of a data set with features from *f* and structure *K* on the modified *f*-adapted data set. These actions are because *f* may not smooth with respect to the geometry of the space, but has properties which are well encoded in its features. In this section, we aim to discover the geometry in hypergraph so that the functions on such structure to be learned are as smooth as possible with respect to that geometry.

Setting

The model of a hypergraph, just like modeled of a graph or a manifold, can be cognized by considering a natural random walk *K* on it. The random walk on hypergraph allows us to construct diffusion operators on the data set and to relate to some basic functions. Let $K^t\delta_x(y)$ be the probability of being at *y* at time *t*, conditioned on starting at *x*, for an initial condition δ_x.

Specifically, the space is a finite weighted hypergraph *H*= (*V*, *E*, *W*), consisting of a vertex set *V*, a hyperedge set *E* $\subseteq 2^V$ such that $e \in E$ is a subset of *V*, and a nonnegative function *W*: $E \to \mathbb{R}^+$. Let *W*(*e*) be the weight for hyperedge *e*. For *v* ∈ *V*, the degree of vertex *v* is defined as $d(v) = \sum_{e\in E} w(e)h(v,e)$, where $h(v,e) = \begin{cases} 1, & \text{if } v \in e \\ 0, & \text{if } v \notin e \end{cases}$. Let *x* and *y* be vertices on hypergraph. A filter acting on functions on *V* can be defined by normalization of the weight matrix as follows:

$$K(x,y)=\begin{cases}\sum_{\{x,y\}\subseteq e} w(e)\frac{1}{\delta(e)} & \text{if } x\neq y\\ d(y) & \texttt{otherwise}\end{cases}.$$

Here *K* is a filter with $\sum_{y\in V} K(x,y)$ =1, and multiplication *Kf* can be considered as a local averaging operation with locality calculated by *W*. As in graphs and manifolds, the multiplication by *K* can be interpreted as a generalization of Parzen window type estimators to functions in hypergraphs. Although *K* is not column-stochastic, the operation *fK* of multiplication can be regarded as a diffusion of the vector *f*. Hence, such filter can be K^t times iterated.

In what follows, let *X* be a data set constructed from a hypergraph *H*: the vertices of *H* are the data points in *X*, and weighted hyperedges are constructed that connect nearby data points.

Harmonic Analysis on Hypergraph

Let { ψ_i } be eigenfunctions of *K* which satisfy $K\psi_i=\lambda_i\psi_i$, and ϕ_i be eigenfunctions of the hypergraph Laplacian:

$$L=D^{-\frac{1}{2}}WD^{-\frac{1}{2}}=D^{\frac{1}{2}}KD^{\frac{1}{2}}-I\ ,$$

where *D* is the diagonal matrix with entries $d(x)$ and *I* is the identity matrix. Since { ϕ_i } is an orthonormal basis, any function $g\in L^2(X)$ can be denoted as $g=\sum_{i\in I}\langle g,\phi_i\rangle\phi_i$. The larger is *i*, the less smoothness the function ϕ_i is, with respect to the geometry given by *W*, and λ_i determines the frequency of ϕ_i. These eigenfunctions always employed in dimensionality reduction implements.

For a function *g* on hypergraph *H*, the gradient on the hyperedge of *H* is given by

$$\nabla g(x,y)=\sum_{\{x,y\}\subseteq e} w(e)(\frac{g(y)}{\sqrt{d(y)}}-\frac{g(x)}{\sqrt{d(x)}}).$$

We use $x\square y$ to denote that there exist hyperedge *e* such that $\{x, y\}\subseteq e$. Then

$$\|\nabla g(x)\|^2=\sum_{x\square y}|\nabla g(x,y)|^2 .$$

In this paper, we use Sobolev norm to measure the smoothness of *g*:

$$\|g\|_{H^1}^2=\sum_x|g(x)|^2+\sum_x\|\nabla g(x)\|^2 .$$

The first term $\sum_x|g(x)|^2$ determines the size of the function *g*, and the second term $\sum_x\|\nabla g(x)\|^2$ decides the size of the gradient. The smaller $\|g\|_{H^1}^2$, the smoother is *g*. Furthermore, we infer

$$\|g\|_{H^1}^2=\|g\|_{L^2(X,d)}^2-\langle g,Lg\rangle .$$

Therefore, it ensures that a function projecting onto the first few terms of its expansion in the eigenfunctions of *L* is a smoothing operation.

Regularization via Diffusion

For better solution the problem of denoising and function extension, it is meaningful to search the smoothest function $\tilde{f}$ on a data set *X* with geometry given by *W*. It satisfies that $\tilde{f}$ is not too far from given *f*. In the denoising application, let η be Gaussian white noise of a given variance and a function $f+\eta \in X \to$ □ is given. A relatively large data set is given in the function extension or interpolation problem, but the values of *f* are showed at only relatively few labeled points, and the aim is to find the value of *f* on the other unlabeled points. These two kinds of implement, with no priori information on *f*, are impossible; the problems are open. It is suitable to suppose that *f* is smooth function, and thus we led to the task of searching a smooth $\tilde{f}$ close to *f*.

In classical Euclidean space, a standard technology of mollification is to carry out the heat equation for a short time with initial condition specified by *f*. It reveals that the heat equation plays a good role on a weighted hypergraph: if *f* is a function on vertex set *V*, let f_0=*f*, and f_{k+1}=*Kf*. If $g_k(x)= d^{\frac{1}{2}}(x) f_k(x)$, then

$$g_{k+1}-g_k=Lg_k.$$

This implies that multiplication by *K* is just a procedure in the formation of the density normalized heat equation. Moreover, a fast calculation implies this is the gradient descent for the smoothness energy functional $\sum\|\nabla g\|^2$. Hence, harmonic interpolation can be done on *X* via iterating *K*.

More general mollifiers can be designed using an expansion on the eigenfunctions $\{\psi_i\}$ of *K*. In what follows, we suppose that all inner products are taken against the measure *d*, i.e., $\langle a,b\rangle = \sum a(x)b(x)d(x)$ and are orthonormal. Thus, *f*=$\sum\langle f,\psi_i\rangle\psi_i$ and $\tilde{f}$ can defined by

$$\tilde{f}=\sum_i \alpha_i \langle f,\psi_i\rangle\psi_i, \quad (1)$$

where $\lim_{i\to+\infty}\alpha_i$=0. For the interpolation problem, we use least squares to estimate the inner products $\langle f,\psi_i\rangle$. Following are some classic examples for choosing parameters α_i:

(i) $\alpha_i=\begin{cases}1, \text{ if } i<I\\ 0, \text{ otherwise}\end{cases}$, *I* is determined by variance of η.

(ii) $\alpha_i=\lambda_i^t$ for some *t*>0, this corresponds to setting $\tilde{f}=K^t(f)$, i.e., kernel smoothing on the data set with a sample-dependent kernel.

(iii) $\alpha_i=P(\lambda_i)$, where *P* is given rational function or polynomial.

ONTOLOGY SIMILARITY MEASURE AND ONTOLOGY MAPPING ALGORITHM DESIGN

In this section, we pose the new ontology learning algorithms based on harmonic analysis and diffusion. Specially, the idea is manifested as follows: the ontology graph *G* is a special case of hypergraph with $|e|$=2 for each $e\in E$. From this point of view, the gradient on ontology graph *G* is given by:

$$\nabla g(x,y) = w(x,y)\left(\frac{g(y)}{\sqrt{d(y)}} - \frac{g(x)}{\sqrt{d(x)}}\right).$$

Via the harmonic analysis and regularization diffusion, we get the function *f* on *V* using algorithm (1). Then, the ontology graph is mapped into a line consisting of real numbers. The similarity between two concepts can be measured by comparing the difference between their corresponding real numbers. For each $v \in V(G)$, $f(v)$ is a target value for vertex *v* using regular graph. We use the following tricks to get the similarity vertices of *v* and return the result list to the users in search expansion applications: Choose parameter *M*, return vertex set {$u \in V(G)$, $|f(u) - f(v)| \le M$ } as an outcome for vertex *u*.

In regard to ontology mapping application, we assume $v \in V(G_i)$, where $1 \le i \le k$. Choose parameter *M*, return vertex set {$u \in V(G\text{-}G_i)$, $|f(u) - f(v)| \le M$ } as the outcome for vertex *v* in multi-ontology graph.

EXPERIMENTAL SECTION

Two experiments concern ontology calculation and ontology mapping are desired below. In order to adjacent to ontology algorithm setting, we should use a vector to express each vertex's information. This vector contains the information of name, instance, attribute and structure of vertex, where the instance of vertex refers to the set of its reachable vertex in the directed ontology graph. We use Gaussian kernel function to compute *W*:

$$w(e)=\mathrm{w}(x,y)= e^{-\frac{\|x-y\|^2}{t}},$$

where parameter $t \in$ □ . Let $\alpha_i = \lambda_i$.

In the first experiment, we use biology ontology O_1 which was constructed in http: //www.geneontology.org. We present the main structure of "GO" ontology in Fig. 1. *P@N* criterion (Precision Ratio, see [13]) is used to measure the equality of the experiment. We first give the closest *N* concepts for each vertex on the ontology graph with the help of experts, and then we obtain the first *N* concepts for every vertex on ontology graph by the algorithm and compute the precision ratio. At the same time, we apply fast ontology method [14], half transductive ranking ontology trick [15] and push ranking ontology technology [16] to the "GO" ontology. Calculating the accuracy via these three algorithms and compare the results to the data from algorithm presented in our paper. Part of the data refer to Table 1.

For the second experiment, we use physical education ontologies O_2 and O_3 (Fig. 2 and Fig. 3 present the structure of O_2 and O_3). The goal of this experiment is given ontology mapping between O_2 and O_3. We also use *P@N* to measure the equality of experiment. Again, we apply ontology algorithm in [14] and [2]to "physical education" ontology, and compare the precision ratio which we get from three methods. Some experiment results refer to Table 2.

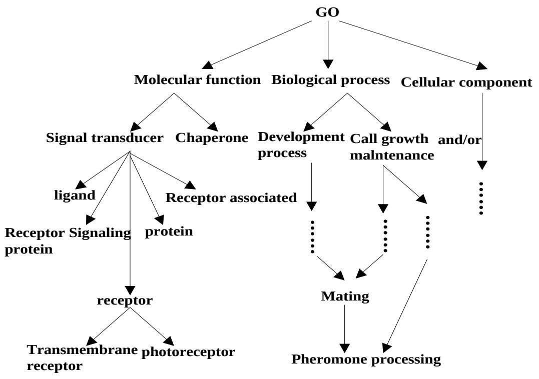

Fig. 1: "GO" Ontology O_1

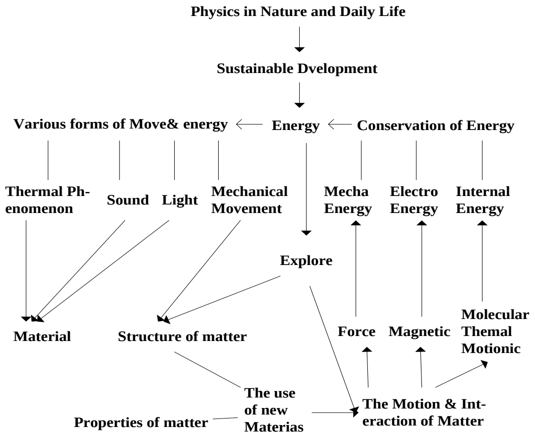

Fig. 2: "Physical Education" Ontology O_2

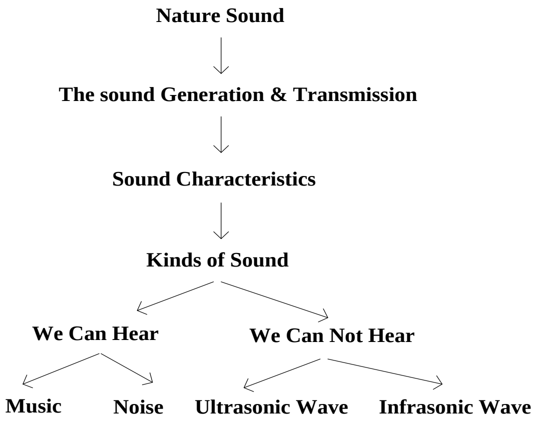

Fig. 3: "Physical Education" Ontology O_3.

Table 1. The experiment data of ontology similarity calculation

	P@3 average precision ratio	*P*@5 average precision ratio	*P*@10 average precision ratio	*P*@20 average precision ratio
Algorithm in our paper	56.46%	67.72%	78.38%	87.74%
Algorithm in Huang *et al.*, (2011)	52.78%	59.63%	72.14%	79.39%
Algorithm in Huang *et al.*, (2011)	53.48%	57.31%	62.56%	71.93%
Algorithm in Wang *et al.*, (2011)	57.34%	63.74%	69.17%	73.68%

From the experiment results display above, we arrived at the conclusion that our algorithm is more efficiently than algorithms raised in [14], [15] and [16] especially when *N* is lager enough. Therefore, this new ontology similarity algorithm with harmonic analysis and diffusion regularization has high efficiency in certain applications.

Table 2. The experiment data of ontology mapping

	P@1 average precision ratio	*P*@3 average precision ratio	*P*@5 average precision ratio
Algorithm in our paper	67.74%	77.42%	89.68%
Algorithm in Huang *et al.*, (2011)	61.29%	73.12%	79.35%
Algorithm in Gao and Liang (2011)	69.13%	75.56%	84.52%

The experiment results in Table 2 reveal that our algorithm is more efficiently than algorithms raised in [14] and [2] especially when *N* is enough lager.

CONCLUSION

In this paper, we present the technology of harmonic analysis and diffusion, and apply it to ontology application. The optimal function we get form the regularization is well suitable for the structure of ontology graph. The new algorithms have high quality according to the experiment data we show above.

Acknowledgements
First we thank the reviewers for their constructive comments in improving the quality of this paper. This work was supported in part by the National Natural Science Foundation of China (60903131). We also would like to thank the anonymous referees for providing us with constructive comments and suggestions.

REFERENCES

[1] Su, X. and J. Gulla, **2004**. Semantic enrichment for ontology mapping. In Proceeding of the 9th International Conference on Information Systems, 217-228.
[2] Gao, W. and L. Liang, **2011**. *Future Communication, Computing, Control and Management*, 142: 415–421.
[3] Gao, Y. and W. Gao, **2012**. *International Journal of Machine Learning and Computing*, 2: 107-112.
[4] Gao, Y. and W. Gao, **2013**. *Advanced Engineering Technology and Application*, 2: 11-14.
[5] Gao, W. and T. Xu, **2013**.Stability analysis of learning algorithm for ontology similarity computation, *Abstract and Applied Analysis*, Volume 2013, Article ID 174802, 9 pages.
[6] Gao, W., Y. Gao, and Y. Zhang, **2012**. *Journal of Information*, 11(A): 4585-4590.
[7] Gao, W. and T. Xu, **2012**. *Journal of networks*, 8: 1251-1259.
[8] Gao, W., T. Xu, J. Gan, and J. Zhou, **2014**. Linear statistical analysis of multi-dividing ontology algorithm. *Journal of Information and Computational Science*, In press.
[9] Gao, Y., W. Gao, and L. Liang, **2014**. Statistical characteristics for multi-dividing ontology algorithm in AUC criterion setting. Manuscript.
[10] Gao, W., Y. Gao, Y. Zhang, and L. Liang, **2014**. Minimax learning rate for multi-dividing ontology algorithm. Manuscript.
[11] Gao, W., L. Yan, and L. Liang, **2013**. Piecewise function approximation and vertex partitioning schemes for multi-dividing ontology algorithm in AUC criterion setting (I). *International Journal of Computer Applications in Technology*, In press.
[12] Yan, L., W. Gao, and J. Li, **2013**. Piecewise function approximation and vertex partitioning schemes for multi-dividing ontology algorithm in AUC criterion setting (II). *Journal of Applied Sciences*, In press.
[13] Craswell, N. and D. Hawking, 2003. Overview of the TREC **2003** web track. In Proceeding of the Twelfth Text Retrieval Conference. Gaithersburg, Maryland, NIST Special Publication, 78-92.
[14] Huang, X., T. Xu, W. Gao, and Z. Jia, **2011**. *International Journal of Applied Physics and Mathematics*, 1: 54-59.
[15] Huang, X., T. Xu, W. Gao, and S. Gong, **2011**. Ontology similarity measure and ontology mapping using half transductive ranking. In Processdings of **2011** 4th IEEE International conference on computer science and information technology, Chengdu, China, 571-574.
[16] Wang, Y., W. Gao, Y. Zhang, and Y. Gao, **2011**. Push ranking learning algorithm on graphs. In Processdings of International Conference on Circuit and Signal Processing. Shanghai, China, 368-371.