http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (15): 3309-3314, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.3309.3314

Notes on Vertex Classification Problem in Hypergraph Setting

[1]Li Yan, [1]Junsheng Li and [2]Wei Gao
[1]School of Engineer, Honghe University, Mengzi, 661100, People's Republic of China
[2]School of Information Science and Technology, Yunnan Normal University, Kunming, 650500, People's Republic of China

Abstract: Classification algorithm is classical learning problem in computer science. Hypergraph, as a knowledge representation model, has widely used in finance, biology and information science. In this study, we extend some graph vertex classification analysis results to hypergraph. Specifically, we use the cut size of the classification to derive data-dependent bounds on the fraction of mistaken predictions for hypergraph vertex classification algorithm.

Key words: Hypergraph, hyperedge, classification algorithm

INTRODUCTION

Classification learning algorithm is a hot topic in computer science. Problem of vertex classification on hypergraph can be regarded as the extension of vertex classification problem on normal graph and used in various computation models.

There has been an historical focus on the hypergraph learning problem. Gao *et al*. (2013) presented new algorithms for ontology similarity measurement and ontology mapping using harmonic analysis and diffusion regularization on hypergraph. Gao and Liang (2011) obtained the optimization function using hypergraph regularization framework. The function maps the vertices of graph to real numbers. The relative similarities of concepts are calculated by comparing the difference of corresponding values. Jia and Gao (2013) gave theoretic analysis for kernel based hypergraph vertex classification semi-supervised learning algorithm and the limiting behavior and generalization bound for such algorithm are determined. Wang *et al*. (2013) studied the statistical characteristics of hypergraph classification algorithm under semi-supervised normalization laplacian dimension reduction were studied. By defining projection operator on the feature space, it is verified that vector after dimension reduction well approximate to the original target vector. The generalization bound for classification algorithm under dimension reduction is estimated by using the trick of matrix trace.

In this study, we focus on the classification problem of vertex in hypergraph. The contribution of this study is to derive data-dependent bounds on the fraction of mis-classified vertices, based on the number or total weight of hyperedges between vertices differing in predicted class. The organization of this study is as follows: the classification problem and notations for hypergraph setting are given in next Section; certain analysis for data-dependent hypergraph vertex learning algorithm is stated in third Section.

SETTING OF HYPERGRAPH AND CLASSIFICATION LEARNING PROBLEM

Hypergraph: In this section, we outline the fundamental concepts and terminologies for our vertex classification algorithm setting. Let $V = \{v_1, v_2, \ldots, v_n\}$ be a finite set, $E = \{e_1, e_2, \ldots, e_m\}$ if family of subset of V, that is $E \subseteq 2^V$. If for $\forall I \in \{1,2,\ldots, m\}$, $e_i \neq \emptyset$ and:

$$\bigcup_{i=1}^{m} e_i = V.$$

Then, G = (V, E) is a hypergraph on set V and the element of V is called a vertex, the elements of E is called a hyperedge. Let |V| be order of G, i.e., the number of vertices in G; |E| be the scale of G. We say |e| is basic number of hyperedge e. $r(G) = \max_j |e_j|$ is denoted as the upper rank of G and:
$s(G) = \min_j |e_j|$ is denoted as the lower rank. If for each hyperedge $e \in E$, we have $|e| = k$ (that is $r(H) = s(H) = k$), then, called G is a k-uniform hypergraph. Expressly, G is a normal graph if k = 2.

A hypergraph G can be represented by a normal graph, using the set of vertex to represent the elements of V:

- If $|e_j| = 2$, using a continuous curve which attach to the elements of e_j to represents e_j
- If $|e_j| = 1$, using a loop which contain e_j to represents e_j

Corresponding Author: Junsheng Li, School of Engineer, Honghe University, Mengzi, 661100, People's Republic of China

- If $|e_j|\geq 3$, using a simple close curve which contain all the elements of e_j to represent e_j

A hypergraph is called simple hypergraph or sperner hypergraph, if any two hyperedges not contain each other. Let $V = \{v_1, v_2,\ldots, v_n\}$, $G = (V, E)$ and $G' = (V, E')$ are two hypergraphs on V. If $E'\subset E$, then we called G' is part-hypergraph of G. Let $V = \{v_1, v_2,\ldots, v_n\}$, $E = \{e_1, e_2, \ldots, e_m\}$ is family of subsets of V, $G = (V, E)$ is a hypergraph on V, $S\subseteq V$, $G[S] = \{e\in E: e\subseteq S\}$ is called sub-hypergraph of G induced by S.

Vertex classification in hypergraph: Hypergraphs are the primary object of study for many computer science application fields, such as ontology similarity measure and ontology mapping. Specially, for certain applications one is interested in classifying the vertices of a hypergraph into of k classes (for instance, vertex coloring problem of hypergraph).

Now, we consider the setting of learning of vertex classification problem in hypergraphs, in which a hypergraph with order n is sampled according to some unknown distribution D and there exist a true classification of the vertices such that each vertex is assigned to exactly one of k classes, but the class of only some random subset of the vertices are revealed to the learner. The aim of learning is to obtain a classification of the rest unclassed vertices with wrong classification as few as possible.

In the following text, we assume $k\geq 2$. The input to the learning algorithm includes a hypergraph $G = (V, E)$, sampled from an unknown distribution D over hypergraphs with order n. In what follows, we assume that the hyperedges are undirected and are not weighted (unless otherwise noted). We will make comment on natural extensions to weighted hypergraphs if appropriate. For a given hypergraph G, a hypothesis space H_V is the set of distinct vertex classifications from which the learner can choice. We only consider H_V to be the completely unrestricted hypothesis space, concluding all k^n possible classifications of the hypergraph with k classes throughout our analysis. For simplicity, we will simply write H if V is clear from the context. Let $f\in H$ be a target classifier for the vertices of G which assign of k possible values $\{1, 2,..., k\}$ to each vertex $v\in V$. Let f (v) be the specific target class for vertex v. The online learning algorithm is given the target classes for certain subset of n_l vertices (as sample set) selected uniformly at random from V. We use these n_l vertices as training vertices and the rest n-n_l unclassed vertices as test vertices. Let $\lambda\in\{0,1\}^n$ be a class mask and set $\Lambda_{n1} = \{\lambda| \|\lambda\|^2 = n_l, \lambda\in \{0,1\}^n\}$. Hence, for a given hypergraph $G = (V, E)$ on $V = \{v_1, v_2,..., v_n\}$ and a class mask $\lambda\in\Lambda_{n1}$, the n_l training vertices are defined as those v_i satisfy $\lambda_i= 1$.

Let empirical error and prediction error of a hypothesis h with respect to a target classification function f, vertex set $V = \{v_1, v_2,..., v_n\}$ and class mask $\lambda\in\Lambda_{n1}$ be:

$$\hat{R}(h,f,V,\lambda)=\frac{\sum_{i=1}^{n}\lambda_i\Pi[f(v_i)\neq h(v_i)]}{n_l}$$

and:

$$R(h,f,V,\lambda)=\frac{\sum_{i=1}^{n}(1-\lambda_i)\Pi[f(v_i)\neq h(v_i)]}{n-n_l}$$

The empirical error reveals the fraction of training vertices the classification is mistaken on, while the prediction error implies the fraction of test vertices the classification is mistaken on. In what follows, we will abbreviate these by R (h) and R (h) respectively if f, λ and V are clear from the context.

The objective of this study is to select a classification that minimizes the prediction error. However, we should bias the algorithm somehow since this quantity generally cannot be directly computed from the input of algorithm. In our setting, we assume that the vertices in the same hyperedge will usually tend to have the same class. Such assumption usually yields to a learning bias depended on cut size.

For a hypergraph $G = (V, E)$ and any classification $h\in H$ of, let:

$$c(h)=\sum_{e\in E}\sum_{\{u,v\}\in E}\Pi\{h(u)\neq h(v)\}$$

be the cut size of the classification. For weighted hyperedges, each term in the summation is also multiplied by the hyperedge weight. Our assumption that vertices in the same hyperedge tend to have the same classes leads to a bias toward small cut sizes. A useful method to implement such bias in a vertex classification algorithm is to search for a classification having smallest cut size but still respecting the target classes of training vertices.

The multi-terminal minimum cut problem in hypergraph is a natural generalization of the multi-terminal minimum cut problem in normal graph and also as an extension of the minimum s-t cut partitioning problem. Given a weighted hypergraph $G = (V, E)$ and a set of terminal vertices $\{v_1, v_2,..., v_k\}\subseteq V$, the problem is to search a set of hyperedges $C\subseteq E$ with minimum total weight such that each path from any terminal v_i to any other terminal

v_j must contain at least one hyperedge from C. i.e., the aim is to search a minimum weight set of hyperedges, the removal of which disconnects all of the terminals. We present the multi-class mincut algorithm by a reduction to the multi-terminal minimum cut problem as follows:

- k new vertices $\{v_1, v_2, \ldots, v_k\}$ is added to the hypergraph
- For any $i\in\{1, 2,..., k\}$, we add an hyperedge with ∞ weight between v_i and any training vertex v for which $f(v) = i$
- Search a multi-terminal minimum cut C in the constructed hypergraph, taking $\{v_1, v_2,..., v_k\}$ as terminals
- For every $i\in\{1, 2,..., k\}$, class with i all test vertices from which exist a path to v_i not including hyperedges in C
- Each test vertices not classed yet can be classed randomly

Note that the multi-terminal minimum cut problem is Max-snp-hard even in normal graph setting.

MAIN RESULTS AND PROOF

In this section, we want to derive data-dependent bounds on the prediction error based on cut size for vertex classification algorithm in hypergraph setting. Assume that $n\geq k$ and the hypergraph initially has fewer than k components. For a hypergraph G = (V, E), let $c_k(G)$ be the minimum number of hyperedges whose removal separates G into at least k non-empty connected components. i.e., $c_k(G)$ is the smallest possible multi-terminal minimum cut size over all selections of k different terminal vertices from V. We denote $c_k(G)$ as the minimum k-cut size of G. As with cut size, this can be generalized for weighted hypergraphs by the sum of weights of hyperedges in a set having smallest total weight whose removal separates G into at least k nonempty connected components.

Let:

$$F_T(m)=\sum_{t=0}^{m}\frac{\binom{T}{t}\binom{n-T}{n_1-t}}{\binom{n}{n_1}}$$

be the cumulative distribution function of the hypergeometric distribution with sample size n_1, positive integer population sizes T and n-T. Let:

$$\binom{x}{y}=0$$

if x<y. Denote:

$$R_{max}(r,\delta)=\max\{\frac{T-m_1}{n-n_1}|F_T(m_1)\geq\delta, T\in Z\}$$

Following lemma is an extension of Lemma 1 in Hanneke (2006), we left the detailed proof to the readers.

Lemma 1: For any hypergraph G with order n and hypothesis space H, if $q: H\rightarrow(0,1)$ is a function satisfies:

$$\sum_{h\in H}q(h)\leq 1$$

then for any class mask λ choice uniformly at random from Λ_{n1}, any target classification f, $\delta\in(0, 1)$ and each hypothesis $h\in H$, with probability at least $1-\delta$, we infer:

$$R(h)\leq R_{max}(\hat{R}(h), q(h)\delta)$$

For a hypergraph G = (V, E) with order n and hypothesis $h\in H$, let:

$$\Gamma(x)=\int_0^{+\infty}t^{x-1}e^{-t}dt$$

be the gamma function:

$$\rho(h)=\frac{c(h)}{c_k(h)}, s(h)=\max\{\lfloor 2k\rho(h)\rfloor, k-1\}$$

and:

$$\binom{x}{y}=\frac{\Gamma(x+1)}{\Gamma(y+1)\Gamma(x-y+1)}$$

be a generalized version of binomial coefficient. Let:

$$B(h)=\begin{cases}k^{s(h)}\binom{n}{2(k-1)\rho(h)}\binom{s(h)}{2(k-1)\rho(h)}^{-1} & \text{if } n\geq\lfloor 2k\rho(h)\rfloor\\ k^n & \end{cases}$$

otherwise clearly, $\rho(h)$ implies the associated cut size of h with respect to the smallest possible k-cut size in hypergraph G. For an integer c, we deduce:

$$\rho(c)=\frac{c}{c_k(G)},\ s(c)=\max\{\lfloor 2k\rho(c)\rfloor, k-1\}$$

and similarly result for B (c). Furthermore, for $\delta\in(0, 1)$, let:

$$\hat{c}(\delta)=\min\{c|\binom{n}{n_1}\delta\leq(c+1)B(c+1), c\in\{0,1,\cdots,r|E|\}\}\cup\{r|E|\}$$

where:

$$r=\binom{r(G)}{2}$$

and r (G) is the upper rank of hypergraph G. Based on these definitions, we get the following result on hypergraph.

Theorem 1: For any unweighted hypergraph G = (V, E) with order n, target classification f, $\delta \in (0, 1)$ and class mask λ chosen uniformly at random from Λ_{n1}, every $h \in H$, with probability at least 1-δ, we obtain:

$$R(h) \le R_{max}(\hat{R}(h), \frac{\delta B(h)^{-1}}{\hat{c}(\delta)+1})$$

The proofing of Theorem 1 heavily depended on the following lemma. The tricks for proofing lemma 2 follow from a techniques used by Karger (1999) and Hanneke (2006) to prove related results.

Lemma 2: Let G = (V, E) be any unweighted hypergraph with order n. For any positive integer c, there are less than:

$$B(c)=\begin{cases} k^{s(c)}\binom{n}{2(k-1)\rho(c)}\binom{s(c)}{2(k-1)\rho(c)}^{-1} & \text{if } s(c) \le n \\ k^{n} & \text{otherwise} \end{cases}$$

classifications $h \in H$ satisfy $c(h) \le c$.

Proof of Theorem 1: Let $S_c = \{h|c(h) \le c, h \in H\}$. By Lemma 2, we obtain:

$$\sum_{h \in S_{\hat{c}(\delta)}} \frac{B(h)-1}{\hat{c}(\delta)+1} \le \frac{\sum_{c=0}^{\hat{c}(\delta)} \frac{|S_c|}{B(c)}}{\hat{c}(\delta)+1} \le 1$$

Since $\forall h \in H$, $B(h)>0$, Lemma 1 reveals:

$$P\{\forall h \in S_{\hat{c}(\delta)}, R(h) \le R_{max}(\hat{R}(h), \frac{\delta B(h)^{-1}}{\hat{c}(\delta)+1})\} \ge 1-\delta$$

Finally, for all $h \in H$, we get:

$$c(h) > \hat{c}(\delta) \Rightarrow R_{max}(\hat{R}(h), \frac{\delta B(h)^{-1}}{\hat{c}(\delta)+1}) = 1 \ge R(h)$$

which completes the proof for any $h \in H$.

Although, Theorem 1 presents a fairly sharp bound, it is implicitly based on a variety of factors in ways that can make it difficult to establish intuition. In order to build the relation between error more explicit and cut size, we provide the following relaxation of Theorem 1 for hypergraph setting. As we see, the new bound approximates the basic relies on Theorem 1 with hypergraph cut size.

Theorem 2: Let $n_u = n-n_l$. For a unweighted hypergraph G = (V,E) with order n, target function f, $\delta \in (0,1)$ and class mask λ chosen uniformly at random from Λ_{n1}, every $h \in H$ satisfy $2k\,\rho(h)<n$, with probability at least 1-δ, we deduce:

$$R(h) \le \hat{R}(h) + \sqrt{\frac{n(n_u+1)(\ln\frac{kne}{\bar{s}(h)+1} + \ln\frac{r|E|+1}{\delta})}{2n_u^2 n_1}}$$

where:

$$\bar{s}(h)=\max\{2(k-1)\rho(h), k-1\}$$

The proof of Theorem 2 can be down by following the analogously proof of Theorem 1, but depended on the relaxation of Lemma 1, developed by Derbeko *et al*. (2004) and Hanneke (2006) with application of Sering's bound.

Lemma 3: For any hypergraph G with order n and hypothesis space H, if q: H→ (0,1) is a function such that:

$$\sum_{h \in H} q(h) \le 1$$

then for any target classification f, $\delta \in (0,1)$ and class mask λchosen uniformly at random from Λ_{n1}, every $h \in H$, with probability at least 1-δ, we have:

$$R(h) \le \hat{R}(h) + \sqrt{\frac{n(n_u+1)(\ln\frac{1}{q(h)} + \ln\frac{1}{\delta})}{2n_u^2 n_1}}$$

For detailed proof of Lemma 3 can be regard as the extension proof of Derbeko ***et al.*** (2004). Such result is established in hypergraph setting as well.

Moreover, the proof of Theorem 2 also depended heavily on the following inequality, which applies when $2k\rho(h)<n$:

$$\begin{aligned} B(h) &\le (\frac{k^{\bar{s}(h)}}{\Gamma(\bar{s}(h))+1}) \cdot (\frac{\Gamma(n+1)\Gamma(\bar{s}(h)-2(k-1)\rho(h)+1)}{\Gamma(n-2(k-1)\rho(h)+1)}) \\ &\le (\frac{ke}{\bar{s}(h)+1})^{\bar{s}(h)} \cdot \frac{n^{2(k-1)\rho(h)}}{(\bar{s}(h)-2(k-1)\rho(h)+1)^{(2(k-1)\rho(h)-\bar{s}(h))}} \\ &\le (\frac{kne}{\bar{s}(h)+1})^{\bar{s}(h)} \end{aligned}$$

Theorems 1 and Theorem 2 can actually apply to any unweighted multi-hypergraph; that is, the similar proofs can be employed without modification for multi-hypergraphs. We apply this truth to extend them for any weighted hypergraph having positive rational-valued hyperedge weights as follows.

Corollary 1: Let w (e) be the weight of hyperedge e∈E and:

$$W = \sum_{e\in E} w(e)$$

L= LCD ({w(e)|e∈E}), D = GCD ({w(e)|e∈E}) and:

$$Q = \frac{L}{D}$$

where LCD denotes the least common denominator and GCD denotes the greatest common divisor. Let:

$$\hat{c}(\delta) = \min\{c \left| \binom{n}{n_l} \delta \le (c+1)B(c+1) \right.$$

where:

$$c \in \{0,1,\cdots,r|E|\} \cup \{r|E|\}, r = \binom{r(G)}{2}$$

and r (G) is the upper rank of hypergraph G. If all hyperedge weights are positive rational numbers and $c_k(G) > 0$, then under the assumptions of Theorem 1, for each h ∈H, we have:

$$R(h) \le R_{max}(\hat{R}(h), \frac{\delta B(h)^{-1}}{\hat{c}(\delta)+1})$$

Proof: Consider the multi-hypergraph G on vertex set V formed by displacing every hyperedge e∈E with Qw (e) unweighted hyperedges. This provides a multi-hypergraph with QW hyperedges. For each h∈H, let $\tilde{c}$ (h) and B (h) denote the cut size and B value evaluated in G. In terms of $\tilde{c}$ (h) = Qc (h) and B (h) = B (h). We deduce:

$$\tilde{B}(\tilde{c}) = B(\frac{\tilde{c}}{Q})$$

By virtue of Theorem 1 for this unweighted multi-hypergraph provides the desired result.

In view of the similar way, a result similar to Corollary 1 can be formulated for extending Theorem 2 to positive rational-valued hyperedge weights as well. As a special situation it is interesting to examine the case of k = 2 classes. i.e., for k = 2, we get:

$$|S_c| \le 2n^{2p(c)}$$

And, this result is the extension of Karger (1999) which concern approximate of 2-cut.

Corollary 2: Based on the above truth, we can present for any connected hypergraph G = (V, E) with order n having positive rational weights, with W, Q and n_u denoted as above, for any target classification f, **δ**∈ (0, 1) and class mask λ selected uniformly at random from Λ_{n1}, every h∈H, that with probability at least 1-**δ**, we have:

$$R(h) \le \hat{R}(h) + \sqrt{\frac{n(n_u+1)(\frac{c(h)}{c_2(G)}\ln n + \frac{1}{2}\ln\frac{2(QW+1)}{\delta})}{n_u^2 n_1}}$$

CONCLUSION

In this study, we pose some extended theoretical results on vertex classification in hypergraph setting. These conclusions contribute to the state of the art and provide theoretical support for practical applications.

ACKNOWLEDGMENTS

First we thank the reviewers for their constructive comments in improving the quality of this study. This work was supported in part by pecuniary aid of Yunnan Province basic research for application (2013fz127). We also would like to thank the anonymous referees for providing us with constructive comments and suggestions.

REFERENCES

Derbeko, P., R. El-Yaniv and R. Meir, 2004. Explicit learning curves for transduction and application to clustering and compression algorithms. J. Artif. Intell. Res., 22: 117-142.

Gao, W. and L. Liang, 2011. Ontology concept similarity computation based on regularization framework of hypergraph. Microelectron. Comput., 5: 15-17.

Gao, W., Y. Gao and L. Liang, 2013. Diffusion and harmonic analysis on hypergraph and application in ontology similarity measure and ontology mapping. J. Chem. Pharm. Res., 5: 592-598.

Hanneke, S., 2006. An analysis of graph cut size for transductive learning. Proceedings of the 23rd International Conference on Machine Learning, June 25-29, 2006, Pittsburgh, PA., USA., pp: 393-399.

Jia, Z.Y. and W. Gao, 2013. Analysis for kernel method hypergraph vertex classification semi-supervised learning algorithm. J. Yunnan Normal Univ. (Nat. Sci. Edn.), 1: 46-49.

Karger, D.R., 1999. Random sampling in cut, flow and network design problems. Math. Oper. Res., 24: 383-413.

Wang, Y.Y., Z.Y. Jia and W. Gao, 2013. Generalization bound analysis for classification algorithm under semi-supervised hypergraph normalization laplacian dimension reduction. J. Nantong Univ. (Nat. Sci. Edn.), 2: 57-61.