This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
سفارش ترجمه این مقاله
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

Knowledge Guided Hierarchical Multi-Label Classification over Ticket Data

Chunqiu Zeng, Wubai Zhou, Tao Li, Larisa Shwartz, and Genady Ya. Grabarnik

***Abstract*—Maximal automation of routine IT maintenance procedures is an ultimate goal of IT service management. System monitoring, an effective and reliable means for IT problem detection, generates monitoring ticket. In light of the ticket description, the underlying categories of the IT problem are determined, and the ticket is assigned to the corresponding processing teams for problem resolving. Automatic IT problem category determination acts as a critical part during the routine IT maintenance procedures. In practice, IT problem categories are naturally organized in a hierarchy by specialization. Utilizing the category hierarchy, this paper comes up with a hierarchical multi-label classification method to classify the monitoring tickets. In order to find the most effective classification, a novel contextual hierarchy (CH) loss is introduced in accordance with the problem hierarchy. Consequently, an arising optimization problem is solved by a new greedy algorithm named $GLabel$. Furthermore, as well as the ticket instance itself, the knowledge from the domain experts, which partially indicates some categories the given ticket may or may not belong to, can also be leveraged to guide the hierarchical multi-label classification. Accordingly, a multi-label inference with the domain expert knowledge is conducted on the basis of the given label hierarchy. The experiment demonstrates the great performance improvement by incorporating the domain knowledge during the hierarchical multi-label classification over the ticket data.**

***Index Terms*—System monitoring, Classification of monitoring data, Hierarchical multi-label classification, Domain Knowledge**

I. Introduction

A. Background

CHANGES in the economic environment force companies to constantly evaluate their competitive position in the market and implement innovative approaches to gain competitive advantages. Without solid and continuous delivery of IT services, no value-creating activities can be executed. The complexity of IT environments dictates usage of analytical approaches combined with automation to enable fast and efficient delivery of IT services. Incident management, one of the most critical processes in IT Service Management [1], aims at resolving the incident and quickly restoring the provision of services while relying on monitoring or human intervention to detect the malfunction of a component. Thus, it is essential to provide an efficient architecture for the IT routine maintenance.

C. Zeng, W. Zhou and T. Li are with the School of Computing and Information Sciences, Florida International University, Miami, Florida, 33199. T. Li is also with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, 210023, P.R.China.
E-mail: {czeng001,wzhou005,taoli}@cs.fiu.edu.
L. Shwartz is with the IBM T.J Watson Research Center, Yorktown Heights, New York. E-mail:{lshwart}@us.ibm.com.
G. Ya. Grabarnik is with the St John's University, Jamaica, New York. E-mail:{grabarng}@stjohns.edu.

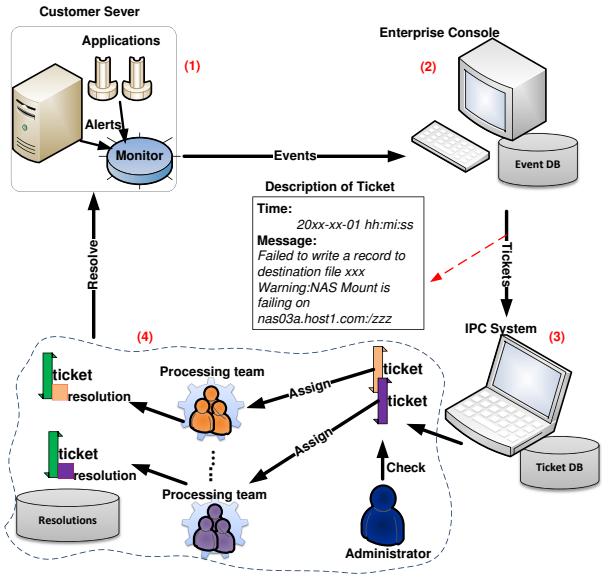


Fig. 1: The overview of the IT routine maintenance procedure.

A typical architecture of the IT routine maintenance is illustrated in Fig. 1, where four components are involved. (1) In the case of detection provided by a monitoring agent on a server, alerts are generated and, if the alert persists beyond a predefined delay, the monitor emits an event. (2) Events coming from an entire account IT environment are consolidated in an enterprise console, which analyzes the monitoring events and determines whether to create an incident ticket for IT problem reporting. (3) Tickets are collected by IPC (abbr. Incident, Problem and Change) system and stored in the ticket database [2]. (4) A ticket accumulates the symptom description of an IT problem with a short text message and a time stamp provided. According to the description of a ticket, the system administrators (i.e., sysAdmins) perform the problem category determination and assign the ticket to its corresponding processing teams for problem diagnosis and resolution. The last component gets involved with much labor-intensive effort to resolve each ticket.

The efficiency of these transient resources is critical for the provisioning of the services [3]. Many IT Service Providers rely on a partial automation for incident diagnosis and resolution, with an intertwined operation of the sysAdmins and an automation script. Often the sysAdmins' role is limited to executing a known remediation script, while in some scenarios the sysAdmin performs a complex root cause analysis. Removing the sysAdmin from the process completely where it is feasible would reduce human error and speed up restoration of service. The move from partially to fully automated problem

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
دانلود کننده مقالات علمی
www.trans24.ir :آنلاین
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

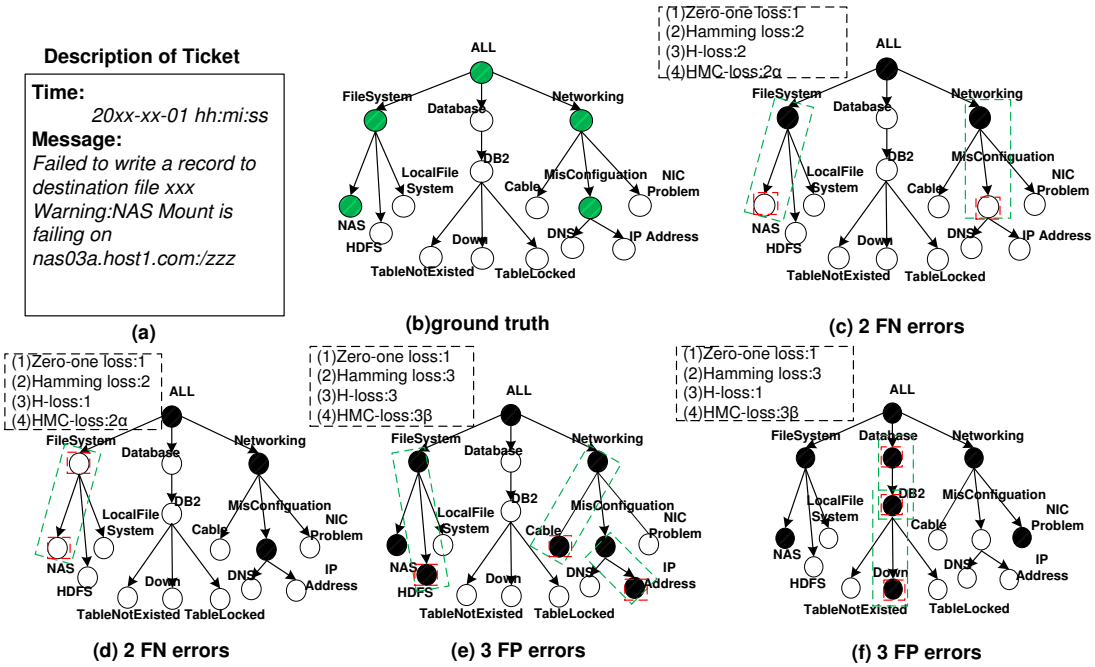


Fig. 2: A hierarchical multi-label classification problem in the IT environment. A ticket instance is shown in (a). (b) presents the ground truth for the ticket with multiple class labels. (c), (d), (e) and (f) are four cases with misclassification. Assuming the cost of each wrong class label is 1, Zero-one loss, Hamming loss, H-loss, HMC-loss are given for misclassification. Notably, to calculate the HMC-loss, the cost weights for FN and FP are α and β, respectively. The misclassified nodes are marked with a red square. The contextual misclassification information is indicated by the green rectangle.

remediation would elevate service delivery to a new qualitative level where automation is a complete and independent process, and where it is not fragmented due to the need for adapting to human-driven processes. The sysAdmin involvement is required due to the ambiguity of service incident description in a highly variable service delivery environment.

B. Motivation

In order to enhance the efficiency of the routine IT maintenance procedure, our work focuses on the labor-intensive component and tries to reduce human involvement by maximizing the automation of the problem category determination. In this paper, we come up with a domain knowledge guided hierarchical multi-label classification method to facilitate the problem determination with both problem hierarchy preservation and domain knowledge integration from system administrators.

As shown in Fig.2.(a), a sample ticket describes a failure of an application to write data to NAS (Network-Attached Storage) [4] file system. To identify a root cause of the problem, it is rational to limit a search space by classifying the incident tickets with their related class labels. Based on the message in Fig.2.(a) the ticket presents a problem related to FileSystem, NAS, Networking and Misconfiguration. Therefore, root cause analysis should be limited to four classes. Moreover, the collection of class labels is hierarchically organized according to the relationship among them. For example, as shown in Fig.2.(b), because NAS is a type of FileSystem, the label FileSystem is the parent of the label NAS in the hierarchy. This taxonomy could be built automatically ([5], [6]) or it could be created with the help of domain experts [7]. In IT environments, hierarchical multi-label classification could be used

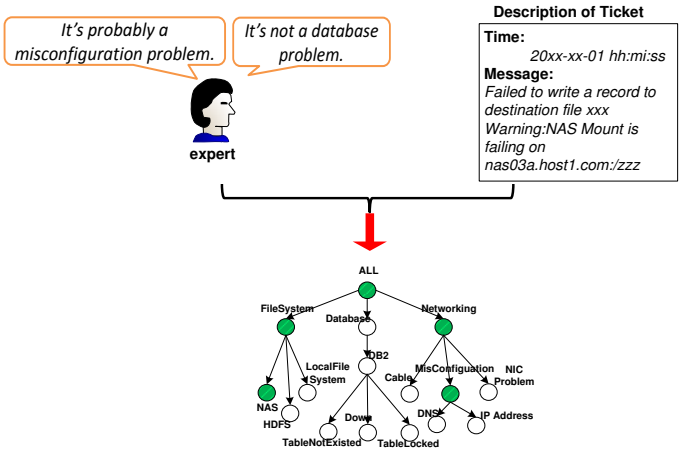


Fig. 3: Knowledge guided ticket hierarchical multi-label classification.

not only for the problem diagnosis ([8], [9], [10], [11]), but also for recommending resolutions ([12]) or auto-check scripts. The ticket in our example could have a solution that addresses FileSystem, NAS, Networking and/or Misconfiguration - a highly diversified action recommendation. Furthermore, based on the hierarchical multi-label classification, actions with different levels of the hierarchy are recommended, where the actions from NAS category are more specific than the ones from FileSystem.

In real practice, as well as the text description of a given ticket, the prior knowledge from the domain experts is involved into the ticket classification by additional check (shown in Fig.1). For example, in Fig.3, given a ticket with its text description, the domain expert, based on his expertise in the

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
سفارش ترجمه این مقاله
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

system management, claims that the problem presented in the ticket is probably (e.g., with 60% confidence) a MisConfiguration problem, and definitely not a Database problem. Intuitively, the prior knowledge from the domain experts should also contribute to the ticket hierarchical multi-label classification for performance improvement. It is abrupt to employ the traditional hierarchical multi-label classification algorithms to deal with the ticket classification problem without taking any prior knowledge into account. However, the prior knowledge integration is not a trivial task in the hierarchical multi-label classification problem. The prior knowledge, about the likelihood that a given ticket should be assigned with a particular label, also contributes to the decision on whether the ticket belongs to those class labels which are highly correlated to the particular label. For example, in Fig.3, given the ticket description, each label in the hierarchy is assigned with a probability to be positive. After inspecting the ticket description, the domain expert further confirms that this ticket is not MisConfiguration problem. Then the probabilities for labels DNS and IP Address being positive become 0, and the probability to be a Networking problem changes accordingly. It is challenging to determine the probability change for each label in the hierarchy, provided with the prior knowledge. To incorporate the prior knowledge, $Kilo$ (**K**nowledge guided h**i**erarchical multi-**l**abel classificati**o**n), a *sum-product* based algorithm, is proposed for hierarchical multi-label inference. Existing work in [13], takes the known hierarchical relationship between categories as knowledge and integrates the hierarchy for multi-label classification, where the knowledge is different from the one discussed in this paper. The work of this paper makes use of the prior knowledge, which partially indicates some categories that a given ticket may or may not belongs to. To the best of our knowledge, this is first work to utilize such prior knowledge to guide the hierarchical multi-label classification.

The $Kilo$ algorithm is not only capable of fully exploring both domain knowledge and the data itself, but also provides an effective way for interactive hierarchical multi-label learning. Concretely, based on the current hierarchical multi-label classification result, the experts provide expertise to hint $Kilo$ for further refinement. $Kilo$ takes the hints from the experts as an input to refine the hierarchical multi-label inference. This iterative process continues until the domain experts are satisfied with the hierarchical multi-label classification result.

In summary, the contribution of our work comprises:

1) We define a new loss function, which takes into the consideration the contextual misclassification information for each label into consideration and is a generalization of Hamming-loss, H-loss and HMC-loss function.
2) Using Bayes decision theory, we develop the optimal prediction rule by minimizing the conditional risk in terms of proposed loss function.
3) We propose the novel $GLabel$ algorithm to efficiently search the optimal hierarchical multi-labels for each data instance in a greedy way.
4) Knowledge from the experts during the IT routine maintenance procedure is explicitly formulated, aiming at facilitating the problem determination.
5) Further, we come up with the $Kilo$ algorithm that allows to integrate the domain knowledge for hierarchical multi-label inference improvement.
6) Extensive empirical studies are conducted on real ticket data to verify both the efficiency and effectiveness of our proposed methods.

Our work has significantly extended our NOMS conference paper [14]. Comparing with the NOMS conference paper, the new contribution of this paper lies in 4) and 5), with essential difference that we are now able to take the domain expertise into account and integrate it with hierarchical multi-label classification inference. This paper adds ample experimentation on real ticket data to show the performance improvement after knowledge integration.

The rest of this paper is organized as follows. In section II, we describe related work and identify limitations of existing methods. In section III, a new loss function for better evaluation of the performance of the hierarchical multi-label classification is proposed and the knowledge from domain experts is formulated. In section IV, the optimal prediction rule is derived with respect to our loss function. Section V describes the algorithm for hierarchical multi-label classification. Section VI illustrates the empirical performance of our method. The last section is devoted to our conclusions and future work.

II. Related Work

In this section, we highlight existing literature studies related to our work. In the literature of IT management, a large number of analytical methodologies based on data mining and machine learning have been explored to optimize the routine IT maintenance and deal with problem detection, determination, diagnosis and resolution ([15], [16], [17], [18], [19], [20]). Particularly, the problem detection is implemented by system monitoring, where monitoring situations are configured and alerts are triggered if monitoring situations are violated. Some popular commercial products such as IBM Tivoli [21], HP OpenView [22], LogicMonitor [23], Zenoss [24], ManageEngine [25], provide system monitoring. Numerous studies about problem detection focus on the improvement of monitoring situation setting ([26], [27], [28]). Besides, anomaly detection facilitates the problem detection ([29], [30], [31]) as well.

The hierarchical classification problem has been extensively investigated in the past ([32], [33], [34], [35], [8], [36], [37], [38], [39], [40]). As a more general case, the hierarchical multi-label classification, where an instance can be labeled with nodes belonging to more than one path or a path without ending on a leaf in the hierarchy, has received much attention.

Recent literature considers several approaches to addressing the hierarchical multi-label classification problem. The first employs existing classification algorithms to build a classifier for each class label independently without any consideration of the hierarchical dependencies of labels. This approach leads to difficult interpretation of the classification result due to the hierarchical inconsistency when an instance is labeled as positive on child label but labeled as negative on parent

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

label. A second approach is to adapt existing single label classification algorithms, such as decision tree ([41], [42]). A third approach is based on the first approach but applies some post-process to automatically guarantee the hierarchical consistency ([9], [10], [11], [43]). We focus on the third approach in this paper.

Cesa-Bianchi et al. [9] introduce an incremental algorithm to apply a linear-threshold classifier for each node of the hierarchy with performance evaluation in terms of H-loss. Moreover, the Bayes-optimal decision rules are developed by Wei in [32]. And Cesa-Bianchi et al. [10] extend the decision rules to the cost-sensitive learning setting by weighting false positive and false negative differently. Wei and James [43] propose the HIROM algorithm to obtain the optimal decision rules with respect to HMC-loss by extending the CSSA algorithm in [11], which has a strong assumption that the number of classes related to each instance is known.

When considering hierarchical multi-label classification problem, adoption of a proper performance measure for a specific application domain is of the most importance. Zero-one loss and Hamming loss that were one of the first loss functions proposed for multi-label classification, are also commonly used in hierarchical multi-label classification problem ([44], [45]). As shown in Fig.2, it is abrupt to adopt the zero-one loss measurement since all the imperfect predictions suffer the same penalty without any distinction. Although much more informative than zero-one loss, Hamming loss suffers a major deficiency since it does not incorporate hierarchy information.

Taking hierarchy information into account, hierarchical loss (H-loss) has been proposed in [9]. The main idea is that any mistake occurring in a subtree does not matter if the subtree is rooted with a mistake as well. As illustrated in (f) of Fig.2, the H-loss only counts once for the label Database even though a mistake also takes place in label DB2 and Down (i.e., db2 is down). This idea is consistent with the scenario of problem diagnosis, since there is no need for further diagnosis in the successive children labels if the reason for the problem has already been excluded in the parent label. However, H-loss could be misleading. Considering for example (f) in Fig.2, after the solution related to Database is wrongly recommended, it is bad to refer the solutions belonging to the successive categories, such as DB2 and DOWN.

The HMC-loss [43] loss function is proposed by weighting the misclassification with the hierarchy information while avoiding the deficiencies of the H-loss. It also differentiates the misclassification between the false negative (i.e., FN) and the false positive (i.e., FP) with different penalty costs. In Fig.2, assuming α and β are the misclassification penalties for FN and FP respectively, (c) and (d) have 2 FN misclassification errors, so both of them incur 2α HMC-loss. Moreover, (e) and (f) suffer 3β HMC-loss since they get 3 FP misclassification errors. However, HMC-loss fails to show the distinction between (c) and (d). In the scenario of the resolution recommendation, based on (c), more diverse solutions are recommended since the ticket is related to both FileSystem and Networking, while only the solutions related to Networking are considered as the solution candidates in (d). However, HMC-loss can not differentiate predictions in (e) and (f). In the scenario of problem diagnosis, intuitively, we prefer (e) to (f) because the minor mistakes in multiple branches are not worse than the major mistakes in a single branch. Based on the discussion above, the main problem of HMC-loss is that it does not hierarchically consider the contextual information for each misclassification label (the contextual misclassification information is indicated with a green rectangle in Fig.2). The concept of the contextual information for each misclassified label is given in section III.

III. Hierarchical Multi-label Classification

In this section, we formally define the hierarchical multi-label classification problem as an optimization problem. Notations mentioned in this paper are summarized in Table I.

TABLE I: Important Notations

Notation	Description
$\mathbf{x}$	a feature vector for each instance.
$\mathbf{y}$	a label vector for each instance.
$\mathbf{k}$	a vector for knowledge representation.
H	the hierarchy relation of labels.
I_e	the indicator function of a boolean expression e.
C_i	the misclassification cost for the ith label according its hierarchical information.
$\hat{\mathbf{y}}$	the estimated label vector for $\mathbf{y}$.
$\hat{y}_i$	the estimated value for y_i.
$\widetilde{y_i}$	the logical complement of y_i.
$pa(i)$	the index for the parent label of the ith label.
$ch(i)$	the indices for the children labels of the ith label.
w_i	the weight for the ith case of the four in CH-loss.
$\ell(\hat{\mathbf{y}}, \mathbf{y})$	loss function for estimating $\mathbf{y}$.
$LE(\hat{\mathbf{y}}, \mathbf{x})$	the expected loss for estimating $\mathbf{y}$ given $\mathbf{x}$.
$Prod_{U \to y_j}(\hat{y}_j)$	the message accumulated on node y_j from its neighbors when $y_j = \hat{y}_j$.
$\mu_{y_i \to y_j}(\hat{y}_j)$	the message passed from y_i to y_j when $y_j = \hat{y}_j$.
N	the length of label vector $\mathbf{y}$.

A. Problem Description

Let $\mathbf{x} = (x_0, x_1, ..., x_{d-1})$ be an instance from the d-dimensional input feature space χ, and $\mathbf{y} = (y_0, y_1, ..., y_{N-1})$ be the N-dimensional output class label vector where $y_i \in \{0, 1\}$. A multi-label classification assigns to a given instance $\mathbf{x}$ a multi-label vector $\mathbf{y}$, where $y_i = 1$ if $\mathbf{x}$ belongs to the ith class, and $y_i = 0$ otherwise. We denote the logical complement of y_i by $\widetilde{y_i} = 1 - y_i$.

The hierarchical multi-label classification is a special type of multi-label classification when a hierarchical relation H is predefined on all class labels. The hierarchy H can be a tree, or an arbitrary DAG (directed acyclic graph). For simplicity, we focus on H being the tree structure and leave the case of the DAG to future work.

In the label hierarchy H, each node i has a label $y_i \in \mathbf{y}$. Without loss of generality, we denote root node by 0, and its label by y_0. For each node i, let $pa(i)$ and $ch(i)$ be the parent and children nodes respectively of the node i . An indicator function I_e of a boolean expression e is defined as

$$I_e = \begin{cases} 1, & e \ is \ true; \\ 0, & e \ is \ false. \end{cases} \tag{1}$$

A hierarchical multi-label classification assigns an instance $\mathbf{x}$ an appropriate multi-label vector $\hat{\mathbf{y}} \in \{0, 1\}^N$ satisfying the Hierarchy Constraint below.

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Downloaded from http://iranpaper.ir
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
http://www.itrans24.com/landing1.html
دانلود کننده مقالات ترجمه شده
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

Definition III.1 (Hierarchy Constraint)**.** *Any node i in the hierarchy* H *is labeled positive (i.e., 1) if it is either the root node or its parent labeled positive. In other words,*

$$y_i = 1 \Rightarrow \{i = 0 \vee y_{pa(i)} = 1\}. \tag{2}$$

Definition III.2 (Knowledge)**.** *Given an instance $\boldsymbol{x}$, the knowledge about the N-dimensional output class label vector $\boldsymbol{y}$ from the domain experts is represented by an N-dimensional vector $\boldsymbol{k}$. According to the domain knowledge, the ith component of $\boldsymbol{k}$ can be assigned with 0, 1 and -1 for negative, positive and unknown respectively.*

Therefore, the knowledge guided hierarchical multi-label classification takes $\mathbf{x}$ and $\mathbf{k}$ as inputs, and outputs the class label vector $\mathbf{y}$.

B. Hierarchical Loss Function

We denote the prediction vector by $\hat{\mathbf{y}}$ and the ground truth by $\mathbf{y}$. To take into account Hierarchy Constraint III.1 while finding optimal prediction, we consider:

Definition III.3 (Contextual Misclassification Information)**.** *Given a node i in hierarchy* H*, the contextual misclassification information depends on whether the parent node of i is misclassified when a misclassification error occurs in node i.*

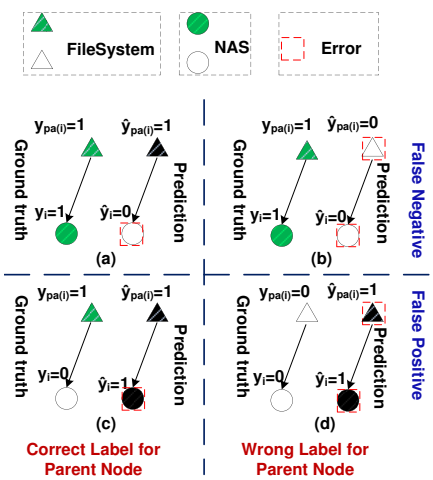


Fig. 4: Four cases of contextual misclassification are shown for node i. In each sub-figure, the left pair is the ground truth, while the right pair is the prediction. The misclassified nodes are marked with red squares.

We incorporate the following four cases of the contextual misclassification information into the loss function to solve the optimization problem, i.e. the best predicted value compatible with the hierarchy H.

- case (a): False negative error occurs in node i, while the parent node $pa(i)$ is correctly predicted.
- case (b): False negative error occurs in both node i and $pa(i)$.
- case (c): False positive error occurs in node i, while the parent node $pa(i)$ is correctly labeled with positive.
- case (d): Both node i and $pa(i)$ are labeled with false positive.

Particularly, we consider the case with a correct parent label (e.g.,(a) and (c)), and the case with a wrong parent label (e.g., (b) and (d)) differently. The reason can be illustrated by the scenario in Fig. 4 where the parent node and the child node represent FileSystem and NAS, respectively. If a ticket is classified with correct FileSystem label but with wrong NAS label, then the ticket will be re-assigned to the other sub-categories of FileSystem. If a ticket is wrongly classified with both FileSystem label and NAS label, then the ticket will belong to sibling categories of FileSystem.

Referring to ([43], [10]), a misclassification cost C_i is given according to the position information of node i in the hierarchy H. And $\{w_i | 1 \le i \le 4\}$ are the different penalty costs for the above four cases, respectively. Accordingly, a new flexible loss function named CH-loss (Contextual Hierarchical loss) is defined as follows:

$$\begin{aligned}\ell(\hat{\mathbf{y}}, \mathbf{y}) = w_1 \underbrace{\sum_{i>0}^{N-1} y_i y_{pa(i)} \tilde{\hat{y}}_i \hat{y}_{pa(i)} C_i}_{\text{case (a)}} + w_2 \underbrace{\sum_{i>0}^{N-1} y_i y_{pa(i)} \tilde{\hat{y}}_i \tilde{\hat{y}}_{pa(i)} C_i}_{\text{case (b)}} \\ + w_3 \underbrace{\sum_{i>0}^{N-1} \tilde{y}_i y_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i}_{\text{case (c)}} + w_4 \underbrace{\sum_{i>0}^{N-1} \tilde{y}_i \tilde{y}_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i}_{\text{case (d)}}.\end{aligned} \tag{3}$$

Next we show that the popular loss functions, such as HMC-loss, Hamming-loss and H-loss, are special cases of CH-loss function. We formulate the exact results below.

By setting α and β to be the penalty costs for false negative (FN) and false positive (FP) respectively, and noting that root node, indicating all categories, is always correctly labeled, the HMC-loss function defined in [43] can be expressed as

$$\ell_{HMC}(\hat{\mathbf{y}}, \mathbf{y}) = \alpha \sum_{i>0}^{N-1} y_i \tilde{\hat{y}}_i C_i + \beta \sum_{i>0}^{N-1} \tilde{y}_i \hat{y}_i C_i. \tag{4}$$

Proposition III.4. *The HMC-loss function is the special case of CH-loss function when $w_1 = w_2 = \alpha$ and $w_3 = w_4 = \beta$.*

Proof. The HMC-loss function can be rewritten as follows.

$$\begin{aligned}\ell_{HMC}(\hat{\mathbf{y}}, \mathbf{y}) = \alpha \sum_{i>0}^{N-1} y_i y_{pa(i)} \tilde{\hat{y}}_i \hat{y}_{pa(i)} C_i + \alpha \sum_{i>0}^{N-1} y_i \tilde{y}_{pa(i)} \tilde{\hat{y}}_i \hat{y}_{pa(i)} C_i \\ + \alpha \sum_{i>0}^{N-1} y_i y_{pa(i)} \tilde{\hat{y}}_i \tilde{\hat{y}}_{pa(i)} C_i + \alpha \sum_{i>0}^{N-1} y_i \tilde{y}_{pa(i)} \tilde{\hat{y}}_i \tilde{\hat{y}}_{pa(i)} C_i \\ + \beta \sum_{i>0}^{N-1} \tilde{y}_i y_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i + \beta \sum_{i>0}^{N-1} \tilde{y}_i \tilde{y}_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i \\ + \beta \sum_{i>0}^{N-1} \tilde{y}_i y_{pa(i)} \hat{y}_i \tilde{\hat{y}}_{pa(i)} C_i + \beta \sum_{i>0}^{N-1} \tilde{y}_i \tilde{y}_{pa(i)} \hat{y}_i \tilde{\hat{y}}_{pa(i)} C_i.\end{aligned}$$

Based on expression (2), we can derive the following expression of HMC-loss by removing the terms that violate the Hierarchy Constraint.

$$\begin{aligned}\ell_{HMC}(\hat{\mathbf{y}}, \mathbf{y}) = \alpha \sum_{i>0}^{N-1} y_i y_{pa(i)} \tilde{\hat{y}}_i \hat{y}_{pa(i)} C_i + \alpha \sum_{i>0}^{N-1} y_i y_{pa(i)} \tilde{\hat{y}}_i \tilde{\hat{y}}_{pa(i)} C_i \\ + \beta \sum_{i>0}^{N-1} \tilde{y}_i y_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i + \beta \sum_{i>0}^{N-1} \tilde{y}_i \tilde{y}_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i.\end{aligned}$$

The equation above is the exact one when w_1,w_2,w_3 and w_4 in equation (3) are substituted with α, α, β and β respectively. □

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
دانلود کننده مقالات ترجمه شده
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

Proposition III.5. *The Hamming-loss function is the special case of CH-loss function when* $w_1 = w_2 = w_3 = w_4 = 1$ *and* $C_i = 1$.

It is established in [43] that the Hamming-loss function is a special case of HMC-loss when $\alpha = \beta = 1$ and $C_i = 1$. Combining the result with the Proposition III.4, the Proposition III.5 is obvious.

The H-loss function (see [43]) cannot be reduced to HMC-loss function, while H-loss is a special case of CH-loss function. Remember that the H-loss function is defined in [9] as follows:

$$\ell_H(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i>0}^{N-1} I_{\hat{y}_i \neq y_i} I_{\hat{y}_{pa(i)} = y_{pa(i)}} C_i. \tag{5}$$

Proposition III.6. *The H-loss function is the special case of CH-loss function when* $w_1 = 1$, $w_2 = 0$, $w_3 = 1$ *and* $w_4 = 0$.

Proof. H-loss defined in (5) is equivalent to the following equation:

$$\ell_H(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i>0}^{N-1} y_i y_{pa(i)} \widetilde{\hat{y}}_i \hat{y}_{pa(i)} C_i + \sum_{i>0}^{N-1} \widetilde{y}_i y_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i + \sum_{i>0}^{N-1} y_i \widetilde{y}_{pa(i)} \widetilde{\hat{y}}_i \widetilde{\hat{y}}_{pa(i)} C_i + \sum_{i>0}^{N-1} \widetilde{y}_i \widetilde{y}_{pa(i)} \hat{y}_i \widetilde{\hat{y}}_{pa(i)} C_i.$$

Based on expression (2), the following expression of H-loss is derived by removing the terms which violate the Hierarchy Constraint:

$$\ell_H(\hat{\mathbf{y}}, \mathbf{y}) = \sum_{i>0}^{N-1} y_i y_{pa(i)} \widetilde{\hat{y}}_i \hat{y}_{pa(i)} C_i + \sum_{i>0}^{N-1} \widetilde{y}_i y_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i.$$

The equation above is the exact equation (3) when $w_1 = 1$, $w_2 = 0$, $w_3 = 1$ and $w_4 = 0$. □

We summarize special cases of CH-loss in the Table II.

TABLE II: special cases of CH-loss

Goal	CH-loss parameter settings
Minimize Hamming loss	$w_1 = w_2 = w_3 = w_4 = 1$, $C_i = 1$
Minimize HMC-loss	$w_1 = w_2 = \alpha$, $w_3 = w_4 = \beta$, C_i is defined by user
Minimize H-loss	$w_1 = w_3 = 1$, $w_2 = w_4 = 0$, $C_i = 1$
Increase recall	w_1 and w_2 are larger than w_3 and w_4
Increase precision	w_3 and w_4 are larger than w_1 and w_2
Minimize misclassification errors occur in both parent and children nodes	$w_2 > w_1$ and $w_4 > w_3$

IV. Expected Loss Minimization

In this section we use the previously defined CH-loss function to predict $\hat{\mathbf{y}}$ given instance $\mathbf{x}$ by minimizing expected CH-loss. Let $\mathbf{y}$ be the true multi-label vector of $\mathbf{x}$, and $P(\mathbf{y}|\mathbf{x})$ be the conditional probability that $\mathbf{y}$ holds given $\mathbf{x}$. The expected loss of labeling $\mathbf{x}$ with $\hat{\mathbf{y}}$ is defined by the following equation:

$$LE(\hat{\mathbf{y}}, \mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^N} \ell(\hat{\mathbf{y}}, \mathbf{y}) P(\mathbf{y}|\mathbf{x}). \tag{6}$$

Let $\hat{\mathbf{y}}^*$ be (one of) the optimal multi-label vector(s) that minimizes expected CH-loss. Based on Bayesian decision theory, the problem is described as follows:

$$\begin{aligned} \hat{\mathbf{y}}^* &= \arg\min_{\hat{\mathbf{y}} \in \{0,1\}^N} LE(\hat{\mathbf{y}}, \mathbf{x}) \\ &\text{s.t. } \hat{\mathbf{y}} \text{ satisfies the hierarchy constraint III.1.} \end{aligned} \tag{7}$$

The key step in solving the problem (7) consists in how to estimate $P(\mathbf{y}|\mathbf{x})$ in equation (6) from the training data. By following the work in [9], [10], [43], in order to simplify the problem, we assume that all the labels in the hierarchy are conditionally independent from each other given the labels of their parents. Since all the data instances are labeled positive at root node 0, we assume that $P(y_0 = 1|\mathbf{x}) = 1$ and $P(y_0 = 0|\mathbf{x}) = 0$. Due to an independency assumption we have:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{N-1} P(y_i | y_{pa(i)}, \mathbf{x}). \tag{8}$$

Thus to estimate $P(\mathbf{y}|\mathbf{x})$, we need to estimate $P(y_i|y_{pa(i)}$ for each node i. The nodewise estimation may be done by utilizing binary classification algorithms, such as logistic regression or support vector machine. To deal with a significant computational load of the nodewise estimation, we parallelize the calculation. The details of the parallelization step are discussed in the next section.

The hierarchy constraint implies that $P(y_i = 1|y_{pa(i)} = 0) = 0$ and $P(y_i = 1|\mathbf{x}) = P(y_i = 1, y_{pa(i)} = 1|\mathbf{x})$. In order to simplify the notation, we denote:

$$p_i = P(y_i = 1|\mathbf{x}) = P(y_i = 1, y_{pa(i)} = 1|\mathbf{x}). \tag{9}$$

Then p_i can be computed based on $P(y_i = 1|y_{pa(i)} = 1, \mathbf{x})$ as:

$$p_i = P(y_i = 1|\mathbf{x}) = P(y_i = 1|y_{pa(i)} = 1, \mathbf{x}) p_{pa(i)}. \tag{10}$$

By combining the definition of CH-loss with Equation (6) and (9), the computation of loss expectation $LE(\hat{\mathbf{y}}, \mathbf{x})$ can be rewritten using p_i notation as follows:

Proposition IV.1 (Expected Loss)**.**

$$LE(\hat{\mathbf{y}}, \boldsymbol{x}) = w_1 \sum_{i>0}^{N-1} \widetilde{\hat{y}}_i \hat{y}_{pa(i)} C_i p_i + w_2 \sum_{i>0}^{N-1} \widetilde{\hat{y}}_i \widetilde{\hat{y}}_{pa(i)} C_i p_i + w_3 \sum_{i>0}^{N-1} \hat{y}_i \hat{y}_{pa(i)} C_i (p_{pa(i)} - p_i) + w_4 \sum_{i>0}^{N-1} \hat{y}_i \hat{y}_{pa(i)} C_i (1 - p_{pa(i)}). \tag{11}$$

Proof. Combining both equation (3) and equation (6), we get:

$$LE(\hat{\mathbf{y}}, \mathbf{x}) = T_1 + T_2 + T_3 + T_4,$$

where

$$\begin{aligned} T_1 &= \sum_{\mathbf{y}} (w_1 \sum_{i>0}^{N-1} y_i y_{pa(i)} \widetilde{\hat{y}}_i \hat{y}_{pa(i)} C_i) P(\mathbf{y}|\mathbf{x}), \\ T_2 &= \sum_{\mathbf{y}} (w_2 \sum_{i>0}^{N-1} y_i y_{pa(i)} \widetilde{\hat{y}}_i \widetilde{\hat{y}}_{pa(i)} C_i) P(\mathbf{y}|\mathbf{x}), \\ T_3 &= \sum_{\mathbf{y}} (w_3 \sum_{i>0}^{N-1} \widetilde{y}_i y_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i) P(\mathbf{y}|\mathbf{x}), \\ T_4 &= \sum_{\mathbf{y}} (w_4 \sum_{i>0}^{N-1} \widetilde{y}_i \widetilde{y}_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i) P(\mathbf{y}|\mathbf{x}). \end{aligned}$$

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
سفارش ترجمه این مقاله
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

we split the proof into four parts:
(a) T_1 can be written as

$$T_1 = w_1 \sum_{i>0}^{N-1} \widetilde{y}_i \hat{y}_{pa(i)} C_i \sum_{\mathbf{y}} P(y_i = 1, y_{pa(i)} = 1, \mathbf{y}|\mathbf{x}).$$

Easily, we get:

$$T_1 = w_1 \sum_{i>0}^{N-1} \widetilde{y}_i \hat{y}_{pa(i)} C_i P(y_i = 1, y_{pa(i)} = 1|\mathbf{x}).$$

$$\therefore T_1 = w_1 \sum_{i>0}^{N-1} \widetilde{y}_i \hat{y}_{pa(i)} C_i p_i.$$

(b) Following (a), we can obtain:

$$T_2 = w_2 \sum_{i>0}^{N-1} \widetilde{y}_i \widetilde{y}_{pa(i)} C_i p_i.$$

(c) Following (a), we have:

$$T_3 = w_3 \sum_{i>0}^{N-1} \hat{y}_i \hat{y}_{pa(i)} C_i P(y_i = 0, y_{pa(i)} = 1|\mathbf{x}).$$

Since $P(y_i = 0, y_{pa(i)} = 1|\mathbf{x}) = P(y_{pa(i)} = 1|\mathbf{x}) - P(y_i = 1, y_{pa(i)} = 1|\mathbf{x}) = p_{pa(i)} - p_i$, then

$$T_3 = w_3 \sum_{i>0}^{N-1} \hat{y}_i \hat{y}_{pa(i)} C_i (p_{pa(i)} - p_i).$$

(d) Following (c),

$$T_4 = w_4 \sum_{i>0}^{N-1} \hat{y}_i \hat{y}_{pa(i)} C_i P(y_i = 0, y_{pa(i)} = 0|\mathbf{x}).$$

Since $P(y_i = 0, y_{pa(i)} = 0|\mathbf{x}) = 1 - P(y_{pa(i)} = 1|\mathbf{x}) = 1 - p_{pa(i)}$, then:

$$T_4 = w_4 \sum_{i>0}^{N-1} \hat{y}_i \hat{y}_{pa(i)} C_i (1 - p_{pa(i)}).$$

After substituting T_1, T_2, T_3, T_4 into $LE(\hat{\mathbf{y}}, \mathbf{y})$, it is the exact equation (11). □

Based on the Expected Loss described in equation (11), the problem (7) is re-formulated as follows:

Proposition IV.2. *The minimization problem (7) is equivalent to the maximization problem below.*

$$\begin{aligned} \hat{\boldsymbol{y}}^* &= \arg\max_{\hat{y} \in \{0,1\}^N} LE_\delta(\hat{\boldsymbol{y}}, \boldsymbol{x}) \\ &\text{s.t. } \hat{\boldsymbol{y}} \text{ satisfies the hierarchy constraint.} \end{aligned} \tag{12}$$

where

$$LE_\delta(\hat{\boldsymbol{y}}, \boldsymbol{x}) = \sum_{i>0}^{N-1} \hat{y}_{pa(i)} (w_2 - w_1) C_i p_i + \sum_{i>0}^{N-1} \hat{y}_i [w_1 C_i p_i - w_3 C_i (p_{pa(i)} - p_i) - w_4 C_i (1 - p_{pa(i)})].$$

Proof. Equation (11) is equivalent to:

$$LE(\hat{\mathbf{y}}, \mathbf{x}) = w_1 \sum_{i>0}^{N-1} (\hat{y}_{pa(i)} - \hat{y}_i) C_i p_i + w_2 \sum_{i>0}^{N-1} (1 - \hat{y}_{pa(i)}) C_i p_i + w_3 \sum_{i>0}^{N-1} \hat{y}_i C_i (p_{pa(i)} - p_i) + w_4 \sum_{i>0}^{N-1} \hat{y}_i C_i (1 - p_{pa(i)})$$

$$\Rightarrow LE(\hat{\mathbf{y}}, \mathbf{x}) = \sum_{i>0}^{N-1} w_2 C_i p_i - (\sum_{i>0}^{N-1} \hat{y}_{pa(i)} (w_2 - w_1) C_i p_i + \sum_{i>0}^{N-1} \hat{y}_i [w_1 C_i p_i - w_3 C_i (p_{pa(i)} - p_i) - w_4 C_i (1 - p_{pa(i)})]).$$

$$\therefore LE(\hat{\mathbf{y}}, \mathbf{x}) = \sum_{i>0}^{N-1} w_2 C_i p_i - LE_\delta(\hat{\mathbf{y}}, \mathbf{x}).$$

So, the solution to minimize $LE(\hat{\mathbf{y}}, \mathbf{x})$ is equivalent to the one to maximize $LE_\delta(\hat{\mathbf{y}}, \mathbf{x})$. □

The problem (12) is still challenging since it contains two free variables y_i and $y_{pa(i)}$ under the hierarchy constraint.

To simplify the problem further, we introduce notations $\sigma_1(i)$ and $\sigma_2(i)$ as follows:

$$\sigma_1(i) = \sum_{j \in child(i)} (w_2 - w_1) C_j p_j. \tag{13}$$

Particularly, if $ch(i) = \varnothing$, $\sigma_1(i) = 0$, and

$$\sigma_2(i) = w_1 C_i p_i - w_3 C_i (p_{pa(i)} - p_i) - w_4 C_i (1 - p_{pa(i)}). \tag{14}$$

Let $\sigma(i)$ be a function of node i defined as

$$\sigma(i) = \begin{cases} \sigma_1(i), & i = 0; \\ \sigma_1(i) + \sigma_2(i), & i > 0. \end{cases} \tag{15}$$

The equation (15) implies:

Proposition IV.3.

$$LE_\delta(\hat{\boldsymbol{y}}, \boldsymbol{x}) = \sum_i \hat{y}_i \sigma(i). \tag{16}$$

Proof. Let $T = \sum_i \hat{y}_i \sigma(i)$. Our goal is to prove $LE_\delta(\hat{\mathbf{y}}, \mathbf{x}) = T$. According to equation(15), we have:

$$T = \sum_i \hat{y}_i \sigma_1(i) + \sum_{i>0} \hat{y}_i \sigma_2(i).$$

Let $T_1 = \sum_i \hat{y}_i \sigma_1(i)$ and $T_2 = \sum_{i>0} \hat{y}_i \sigma_2(i)$. Then,
(a)

$$T_1 = \sum_i \hat{y}_i \sum_{j \in child(i)} (w_2 - w_1) C_j p_j$$

$$\Leftrightarrow T_1 = \sum_i \sum_{j \in child(i)} \hat{y}_{pa(j)} (w_2 - w_1) C_j p_j.$$

Since j is a child of node i, $j > 0$,

$$\therefore T_1 = \sum_{j>0} \hat{y}_{pa(j)} (w_2 - w_1) C_j p_j.$$

$$(b) T_2 = \sum_{i>0} \hat{y}_i (w_1 C_i p_i - w_3 C_i (p_{pa(i)} - p_i) - w_4 C_i (1 - p_{pa(i)})).$$

(c) Combining both T_1 and T_2, we prove $T = LE_\delta(\hat{\mathbf{y}}, \mathbf{x})$. □

Based on the equation (16), the solution to the problem (12) is equivalent to the one of problem (17).

$$\begin{aligned} \hat{\mathbf{y}}^* &= \arg\max_{\hat{y} \in \{0,1\}^N} \sum_i \hat{y}_i \sigma(i) \\ &\text{s.t. } \hat{\mathbf{y}} \text{ satisfies the hierarchy constraint.} \end{aligned} \tag{17}$$

The solution of the problem (17) by a greedy algorithm is described in the next section.

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
دانلود کننده مقالات ترجمه شده
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

V. Algorithms and Solutions

As discussed in previous sections, there are four key steps to address the hierarchical multi-label classification, aiming at minimizing CH-loss.

1) Estimate the probability p_i for each node i based on the training data.
2) Incorporate the domain knowledge $\mathbf{k}$ using the $Kilo$ algorithm, and adjust the probability p_i for each node i accordingly.
3) Use p_is to compute the $\sigma(i)$ defined by the equation (15).
4) Obtain the optimal predictor $\hat{\mathbf{y}}^*$ using the $GLabel$ algorithm.

$Kilo$ revises p_is by accounting for prior knowledge, while $GLabel$ takes the revised p_is as input to acquire the final optimal label vector $\hat{\mathbf{y}}^*$. If no prior knowledge is provided, the hierarchical multi-label classification problem can be addressed with steps 1), 3) and 4), where $Kilo$ is not applicable.

A. Estimating probability p_i

According to the equation (10), p_i can be computed by estimating the probability $P(y_i = 1|y_{pa(i)} = 1, \mathbf{x})$. For each node i with positively labeled parent node, a binary classifier is built based on existing methods, such as logistic regression or support vector machine. Given an instance $\mathbf{x}$, we apply thresholding described in [46] to convert the real output of the classifier to estimate $P(y_i = 1|y_{pa(i)} = 1, \mathbf{x})$.

Building classifiers for all the nodes is a time-consuming task.Since the building process of the classifier on each node only relies on the related training data and all the classifiers are mutually independent, we parallelize the task to improve the performance [47].

Then, the values of p_is are computed by applying Equation 9 while traversing the nodes in the hierarchy. Fig.5(b) illustrates the marginal probabilities by considering the hierarchical label tree in Fig.5(a). The time complexity of p_i computation is $O(N)$, where N is the number of nodes in the hierarchy.

B. Incorporating prior knowledge $\boldsymbol{k}$

In a probabilistic graphical model, each node represents a random variable and the link between two nodes expresses the probabilistic relationship between them [48]. The graph captures the way in which the joint distribution over all of the random variables can be decomposed into a product of factors each depending only on a subset of the variables. Accordingly, it is straightforward to interpret the hierarchical tree H as a probabilistic graphical model where each label node y_i corresponds to a random variable taking value either 0 or 1 and each link from parent to child represents the hierarchical constraint III.1. The label inference on a tree-structured graph can be efficiently addressed by the *sum-product* algorithm [48]. In this section, $Kilo$ (**K**nowledge guided **i**erarchical multi-**l**abel classificati**o**n), a *sum-product* based algorithm, is proposed to adjust the marginal probability p_i for knowledge incorporation.

Assuming y_i and y_j to be two label nodes in H where a link occurs between them, $\mu_{y_i \to y_j}(\hat{y}_j)$ denotes the message passed from node y_i to node y_j, when the random variable y_j takes the value $\hat{y}_j$.

Definition V.1 (Accumulated Message)**.** *Given a node y_j and a node set U where any node $y_t \in U$ is a neighbour of y_j, $Prod_{U \to y_j}(\hat{y}_j)$ is referred as the message accumulated on node y_j from U when $y_j = \hat{y}_j$. It is defined as follows:*

$$Prod_{U \to y_j}(\hat{y}_j) = \prod_{y_t \in U} \mu_{y_t \to y_j}(\hat{y}_j). \tag{18}$$

Especially, when $U = \emptyset$, $Prod_{U \to y_j}(\hat{y}_j) = 1$.

Definition V.2 (Passed Message)**.** *Given a node y_j and its neighbour y_i, let $U_{i/j}$ denote a set containing all the neighbours of y_i except y_j. The message passed from y_i to y_j when $y_j = \hat{y}_j$ is defined as follows:*

$$\mu_{y_i \to y_j}(\hat{y}_j) = \begin{cases} \sum_{y_i} p(y_i|y_j = \hat{y}_j) Prod_{U_{i/j} \to y_i}(\hat{y}_i), & y_i \text{ is child of } y_j; \\ \sum_{y_i} p(y_j = \hat{y}_j|y_i) Prod_{U_{i/j} \to y_i}(\hat{y}_i), & y_j \text{ is child of } y_i. \end{cases} \tag{19}$$

Proposition V.3. *Let U be the node set containing all the neighbours of y_j, then the marginal probability*

$$p(y_j = \hat{y}_j) = Prod_{U \to y_j}(\hat{y}_j). \tag{20}$$

This proposition can be simply verified by an example in Fig.5. According to the D-separation property of probabilistic graphical model [48], the joint probability of the label vector in the example is given by

$$p(y_0, y_1, y_2, y_3, y_4) = p(y_0)p(y_1|y_0)p(y_2|y_0)p(y_3|y_1)p(y_4|y_1).$$

Therefore, the marginal probability $p(y_1 = \hat{y}_1)$ can be computed as follows:

$$p(y_1 = \hat{y}_1) = \sum_{y_0}\sum_{y_2}\sum_{y_3}\sum_{y_4} [p(y_0)p(y_1 = \hat{y}_1|y_0)p(y_2|y_0) \\ p(y_3|y_1 = \hat{y}_1)p(y_4|y_1 = \hat{y}_1)]. \tag{21}$$

According to the definition of accumulated message, $Prod_{U \to y_1}(\hat{y}_1)$ can be computed in the following.

$$\begin{aligned} Prod_{U \to y_1}(\hat{y}_1) &= \mu_{y_0 \to y_1}(\hat{y}_1) \cdot \mu_{y_3 \to y_1}(\hat{y}_1) \cdot \mu_{y_4 \to y_1}(\hat{y}_1) \\ &= \sum_{y_0} p(y_1 = \hat{y}_1|y_0)\mu_{y_2 \to y_0}(y_0) \cdot \sum_{y_3} p(y_3|y_1 = \hat{y}_1) \\ &\quad \cdot \sum_{y_4} p(y_4|y_1 = \hat{y}_1) \\ &= \sum_{y_0} [p(y_1 = \hat{y}_1|y_0) \cdot \sum_{y_2} p(y_2|y_0)] \cdot \sum_{y_3} p(y_3|y_1 = \hat{y}_1) \\ &\quad \cdot \sum_{y_4} p(y_4|y_1 = \hat{y}_1) \\ &= \sum_{y_0}\sum_{y_2}\sum_{y_3}\sum_{y_4} [p(y_1 = \hat{y}_1|y_0)p(y_2|y_0)p(y_3|y_1 = \hat{y}_1) \\ &\quad p(y_4|y_1 = \hat{y}_1)]. \end{aligned} \tag{22}$$

Since $p(y_0 = 1) = 1.0$, $p(y_0 = 0) = 0.0$, $p(y_1 = \hat{y}_1|y_0 = 0) = 0.0$, $p(y_2 = 1|y_0 = 0) = 0.0$ and $p(y_2 = 0|y_0 = 0) = 0.0$, we get $p(y_1 = \hat{y}_1) = Prod_{U \to y_1}(\hat{y}_1)$ based on Equation (21) and (22).

So far, we have considered the label inference without any knowledge. According to Definition III.2, the knowledge is

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
دانلود کننده مقالات ترجمه شده
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

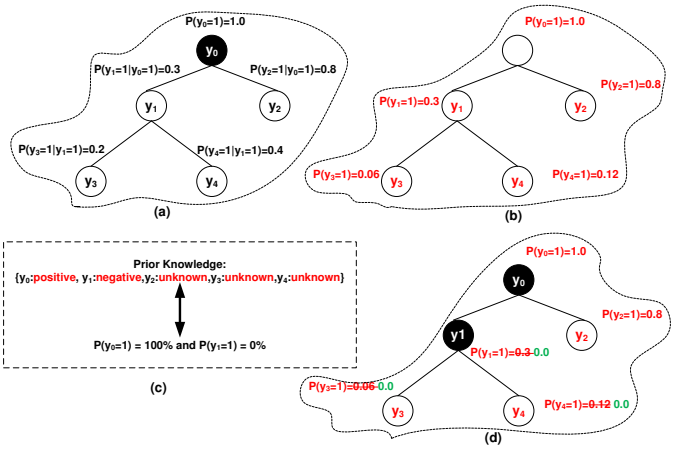


Fig. 5: This figure illustrates the marginal probability by incorporating the domain knowledge.The label nodes in black circle are observed. (b) can be inferred by marginalization from (a), while (d) can be inferred by considering both (a) and prior knowledge in (c). The prior knowledge in (c) can be represented with a likelihood vector about being positive.

Algorithm 1 $Kilo(\mathbf{H}, \mathbf{k})$

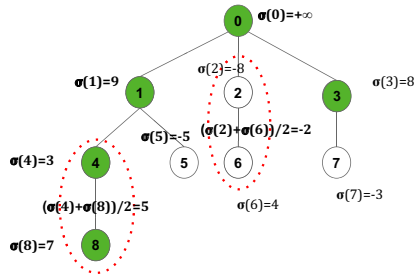
▷$\mathbf{H}$ is the label hierarchy tree, with $P(y_i = 1|y_{pa(i)} = 1, \mathbf{x})$ on its corresponding link.
▷$\mathbf{k}$ is the knowledge vector.
1: Initialize a 3-dimensional array T with size $N \times N \times 2$, where N is the number of labels. T_{ijk} corresponds to the passed message $\mu_{y_i \to y_j}(y_j = k)$, where k is either 0 or 1.
▷compute $\mu_{y_i \to y_j}(y_j = k)$ from bottom to top.
2: Starting from leaf nodes along the links between their parents, compute $\mu_{y_i \to y_j}(y_j = k)$ and fill it in the T_{ijk}.
▷compute $\mu_{y_i \to y_j}(y_j = k)$ from top to bottom.
3: Starting from root node along the links between their children, compute $\mu_{y_i \to y_j}(y_j = k)$ and fill it in the T_{ijk}.
4: Compute the marginal probability for each label according to Equation (20).
5: Normalize all the marginal probability with the marginal probability of root label.
6: **return** a vector containing all the marginal probabilities for all the labels.

represented as a vector $\mathbf{k}$ (shown in Fig.5 (c)). If $k_i = 0$ or $k_i = 1$, it indicates that the ith label is observed as $negative$ or $positive$. While $k_i = -1$, it means the ith label is hidden as $unknown$. In order to incorporate the knowledge vector $\mathbf{k}$, an indicator function is defined as follows:

$$I(y_i, k_i) = \begin{cases} 1, if\ k_i = -1 \vee y_i = k_i; \\ 0, otherwise. \end{cases} \tag{23}$$

The knowledge $\mathbf{k}$ is incorporated by considering $p(\mathbf{y}|\mathbf{k})$, i.e., the posterior probability of label vector given $\mathbf{k}$. The inference is implemented by multiplying the joint probability $p(\mathbf{y})$ with $\prod_i I(y_i, k_i)$, where $p(\mathbf{y}|\mathbf{k}) \propto p(\mathbf{y}) \cdot \prod_i I(y_i, k_i)$. After normalization by $\sum_{\mathbf{y}} p(\mathbf{y}) \cdot \prod_i I(y_i, k_i)$, the exact value of $p(\mathbf{y}|\mathbf{k})$ can be obtained. The posterior distribution $p(\mathbf{y}|\mathbf{k})$ is used to estimate $p(y_i|\mathbf{k})$, the marginal probability for node i. Fig.5(d) shows the final marginal probabilities inferred by incorporating knowledge vector in Fig.5(c). Specifically, indicated by the knowledge, $p(y_1 = 1)$ is changed from 0.3 to 0.0. Accordingly, $p(y_3 = 1)$ and $p(y_4 = 1)$ are 0.0 because of the hierarchy constraint. In more general cases, the likelihood vector about being positive, denotes the prior knowledge, contains values ranging from 0.0 to 1.0.

In light of the above analysis, $Kilo$ algorithm is proposed for knowledge incorporation and shown in Algorithm 1. The $Kilo$ algorithm takes the hierarchical tree and the knowledge vector as the inputs and return the marginal probability vector after knowledge incorporation. It traverses the tree along the links twice in both bottom-to-top and top-to-bottom directions. It needs to normalize the marginal probability for each node. Therefore, the overall time complexity is $O(N + E)$, where N and E are the number of nodes and number of links respectively.



Node	Value	order
σ(0)	+∞	1
σ(1)	9	2
σ(2)	-8	5
σ(3)	8	3
σ(4)	3	4
σ(5)	-5	7
σ(6)	4	5
σ(7)	-3	6
σ(8)	7	4

Fig. 6: It illustrates hierarchy with 9 nodes and steps of Algorithm 2. Nodes labeled positive are green. A dotted ellipse marks a super node composed of the nodes in it. In the table, both the σ value and the labeling order for each node are given. Nodes with the same order are labeled together as a super node.

C. Computing variable $\sigma(i)$

With p_i available, σ can be computed based on equation (15) by recursively traversing each node of the hierarchy. Since each node in hierarchy needs to be accessed twice, one for computing σ_1 and the other for computing σ_2. Therefore, time complexity of $\sigma(i)$ evaluation is also $O(N)$.

D. Obtaining label $\hat{\mathbf{y}}^*$

The value $\hat{\mathbf{y}}^*$ is obtained by solving the maximization problem (17). [11] proposed the greedy algorithm CSSA, based on the work in [49] that allows for solving the problem (17) efficiently. However, CSSA only works under an assumption that the number of labels to be associated with a predicted instance is known. That assumption rarely holds in practice. In [43], the HIROM algorithm is proposed to avoid the deficiency of CSSA by giving the maximum number of labels related to a predicting instance. During the process of finding maximum number of labels, HIROM gets the optimal $\hat{\mathbf{y}}^*$ by comparing all possible $\hat{\mathbf{y}}$s with different numbers of labels related to a predicting instance.

We suggest a novel greedy labeling algorithm $GLabel$(**Algorithm 2**) to solve the problem (17). This algorithm finds the optimal $\hat{\mathbf{y}}^*$ without knowing the maximum number of labels for the predicting instance. It labels the node (or super node) i with maximum $\sigma(i)$ to be positive by searching in the hierarchy. If the parent node of i is negative,

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
سفارش ترجمه این مقاله
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

then i and its parent are merged into a super node whose σ value is the average σ value of all the nodes contained in the super node (e.g., node 4 and node 8 in Fig.6). The labeling procedure stops when the maximum σ value is negative or all nodes are labeled positive.

Algorithm 2 $GLabel(\mathbf{H})$

```
▷H is the label hierarchy, with σ available
define L as a set, and initialize L = {0}
define U as a set, and initialize U = H\{0}
while TRUE do
    if all the nodes in H are labeled then
        return L
    end if
    find the node i with maximum σ(i)
    if σ(i) < 0 then
        return L
    end if
    if all the parents of i are labeled then
        put i into L, and remove it from U
    else
        merge i with its parent as a super node i*
        σ(i*) =average σ values of the two nodes
        put the i* into U
    end if
end while
```

Since the labeling procedure for each node may involve a merging procedure, the time complexity is no worse than $O(Nlog(N))$, the same as HIROM. However, as shown in the experimentation section below, $GLabel$ performs more efficiently than HIROM while not requiring knowledge of the maximum number of labels.

VI. Experimentation

A. Setup

We perform the experiment over the ticket data set generated by monitoring of the IT environments of a large IT service provider. The number of tickets in the experiment amounts to about 23,000 in total. The experiment is executed 10 times and we use the mean value of the corresponding metric to evaluate the performance of our proposed method. At each time, 3000 tickets are sampled randomly from the whole ticket data set to build the testing data set, while the rest of the tickets are used to build the training data set. The class labels come from the predefined catalog information for problems occurring during maintenance procedures. The whole catalog information of problems is organized in a hierarchy, where each node refers to a class label. The catalog contains 98 class labels; hence there are 98 nodes in the hierarchy. In addition, the tickets are associated with 3 labels on average and the height of the hierarchy is 3 as well.

The features for each ticket are built from the short text message describing the symptoms of the problem. Prior to the feature construction, we enhance the expression of the domain-specific terms (e.g., NAS, DB2) or phrases (e.g., nas03a.host1.com/zzz) with their detailed descriptions from domain experts. Then, natural language processing techniques are applied to remove the stop words and build Part-Of-Speech tags for the words in the rephrased text. The nouns, adjectives and verbs in the text are extracted for each ticket. Second, we compute the TF-IDF [50] scores of all words extracted from the text of tickets. And the words with the top 900 TF-IDF score are kept as the features for the tickets. Third, the feature vector of each ticket has 900 components, where value of each feature is the frequency of the feature word occurring in the text of the ticket.

Based on the features and labels of the tickets, we build a binary classifier for each node in the hierarchy with the SVM algorithm by using library libSVM [51]. The training data for each node i are the tickets with a positive parent label. To speed up evaluation of the 98 SVM classifiers, we parallelize the process of training classifiers, using the fact that all the classifiers are independent.

The experiments are mainly conducted by comparing the proposed the $GLabel$ algorithm with state-of-the-art algorithms such as CSSA and HIROM. Note, that in the end, we also show benefits of hierarchical classification in comparison to the "Flat" classification.

B. Hamming loss

The $GLabel$ algorithm can obtain optimal $\hat{\mathbf{y}}^*$ with minimum Hamming loss by setting the parameters for Hamming loss, since Hamming loss is a special case of CH-loss. Given $w_1 = w_2 = w_3 = w_4 = 1$ for $GLabel$, $\alpha = \beta = 1$ for HIROM and $C_i = 1$ for both of them, empirical results are displayed in Fig. 7a - Fig. 7d. As shown in Fig. 7a, the $GLabel$ algorithm can automatically find the optimal $\hat{\mathbf{y}}^*$ with minimum Hamming loss, while both CSSA and HIROM require the number of class labels and the maximum number of class labels, respectively, to get the optimal $\hat{\mathbf{y}}^*$. Particularly, by increasing the value for setting the maximum number of class labels, HIROM incurs smaller Hamming loss. The HIROM algorithm does not reach the optimal $\hat{\mathbf{y}}^*$ with minimum Hamming loss until a sufficiently large value is used for specifying the maximum number of class labels. Moreover, CSSA may incur more Hamming loss as the number of class labels increases and it is difficult to identify the correct number of class labels for each ticket in practice. During the Hamming loss minimization, we track the varying performance in terms of precision, recall and F-Measure score, illustrated in Fig. 7b - Fig. 7d. The conclusion is that the $GLabel$ algorithm is capable of achieving the minimum Hamming loss without specifying the number of class labels in advance.

C. HMC-Loss

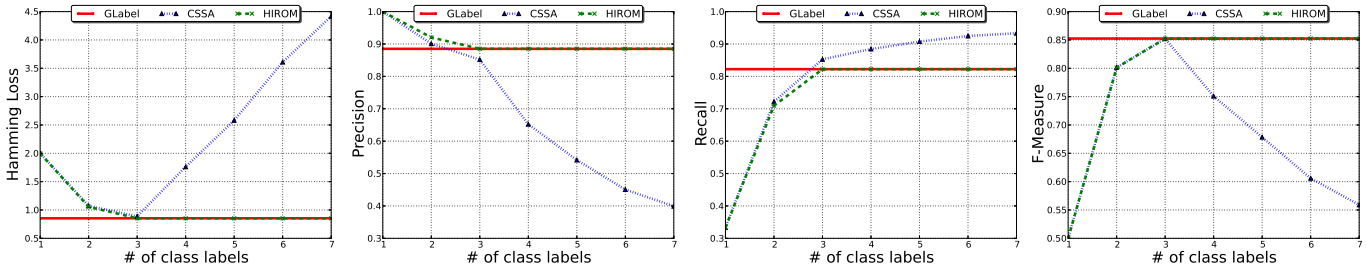
The HMC-Loss considers loss with respect to the node position in the hierarchy. Following [43], we define the C_i as follows.

$$C_i = \begin{cases} 1, & i = 0; \\ \frac{C_{pa(i)}}{\text{\# of i's siblings}}, & i > 0. \end{cases} \tag{24}$$

To simplify the empirical study for HMC-Loss, we set $w_1 = w_2 = w_3 = w_4 = 1$ for $GLabel$, and $\alpha = \beta = 1$ for HIROM. In Fig. 7e, it shows that the $GLabel$ algorithm automatically obtains the lowest HMC-Loss, which can be achieved by the HIROM algorithm after choosing the proper value for the maximum number of class labels.

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
http://www.itrans24.com/landing1.html
Downloaded from http://iranpaper.ir
www.trans24.ir آنلاین:
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

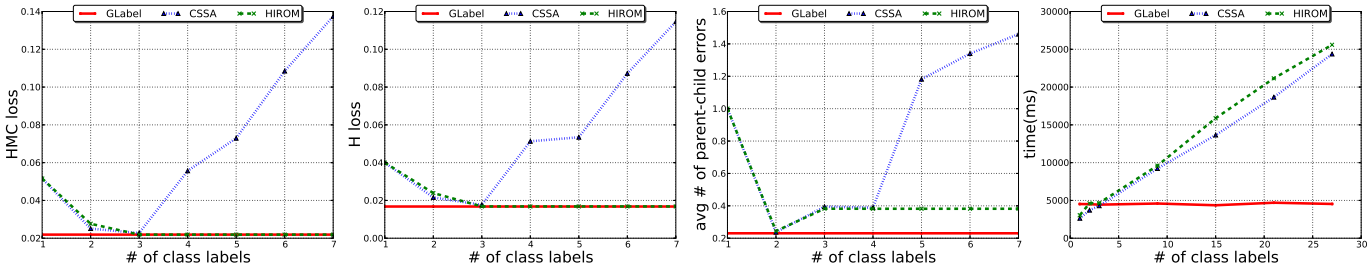


(a) The lowest Hamming loss: CSSA gets 0.8876 at # 3; HIROM gets 0.8534 at # 4; $GLabel$ gets 0.8534.

(b) Varying precision during minimizing the Hamming loss.

(c) Varying recall during minimizing the Hamming loss.

(d) Varying F-Measure score during minimizing the Hamming loss.



(e) The lowest HMC-Loss: CSSA gets 0.0227 at # 3; HIROM gets 0.0219 at # 4; $GLabel$ gets 0.0219.

(f) The lowest H-Loss: CSSA gets 0.0176 at # 3; HIROM gets 0.0168 at # 3; $GLabel$ gets 0.0167.

(g) The lowest AVG. parent-child error: CSSA gets 0.237 at # 2; HIROM gets 0.2440 at #2; $Glabel$ gets 0.2304.

(h) Time complexity with respect to the number of classes related to each predicting ticket.

Fig. 7: Experiments involving tickets in terms of Hamming Loss, Precision, Recall, F-Measure, HMC-Loss, H-Loss, Parent-Child error, and time cost.

D. H-Loss

In order to get the minimum H-Loss, we set $w_1 = w_3 = 1$,$w_2 = w_4 = 0$ for $GLabel$, $\alpha = \beta = 1$ for HIROM, and $C_i = 1$ for all the three algorithms, i.e., $GLabel$, HIROM, CSSA. Fig.7f shows that $GLabel$ is capable of obtaining the lowest H-Loss, while both HIROM and CSSA cannot acquire the optimal $\hat{\mathbf{y}}^*$ with the minimal H-Loss, regardless of how to tune the number of class labels.

E. Misclassifications occurring in both parent and child labels

The worse error from the loss point of view is the misclassification of both parent and child nodes. We call such misclassification a parent-child error. In terms of CH-Loss, $GLabel$ can minimize the number of such cases by setting $w_1 = w_3 = 1$, $w_2 = w_4 = 10$ with more penalties in parent-child errors. To compare, we set in CSSA and HIROM $\alpha = \beta = 1$, and C_i according to the equation (24). As shown in Fig.7g, $GLabel$ reaches the minimum average number of parent-child errors, while CSSA and HIROM algorithms can not minimize the parent-child errors since both of them do not consider the contextual misclassification information in their loss function.

F. Time complexity

In order to evaluate the time complexity, we fix the same parameters but increase the number of classes labels, see Fig.7h. We run three algorithms for 40 rounds and get the average time consumed. Fig.7h shows that running time of $GLabel$ is independent from the number of labels, while running time of other two algorithms require more time as the number of labels increases. Hence, the $GLabel$ algorithm is more efficient than other two algorithms, especially in the cases with large number of class labels.

G. Comparison study with "Flat" classifier

To set up a "Flat" classification, a classifier is built for each label independently without considering the hierarchy constraint. The SVM algorithm is one of the best performing algorithms used to classify the ticket data with each binary class label.

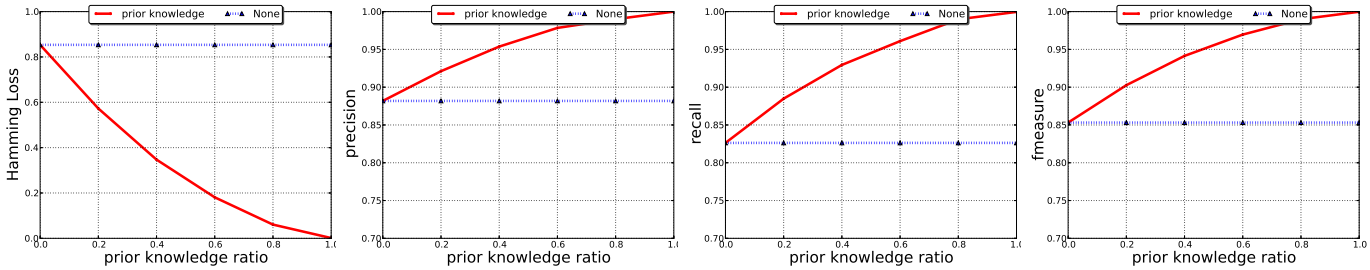
In order to decrease the parent-child error, we set $w_1 = w_3 = 1$, $w_2 = w_4 = 10$, and C_i as the equation (24). In addition, we define the hierarchy error as the average number of violated hierarchy constraints. Table III shows that the $GLabel$ algorithm has better performance in terms of CH-Loss and parent-child error. Furthermore, the "Flat" SVM classification suffers on average 0.0102 hierarchy errors with each ticket, while $GLabel$ complies with the hierarchy constraint and does not have hierarchy errors. We also conduct

TABLE III: Comparison between $GLabel$ and "Flat" SVM classification in terms of CH-Loss, Parent-Child error and Hierarchy error, where both the mean and the 95%-level confidence interval of experiment results are provided.

Metric	**SVM**	$GLabel$
CH-Loss	4.2601 ± 0.07314	$\mathbf{2.6889} \pm 0.05717$
Parent-child error	0.3788 ± 0.00679	$\mathbf{0.1729} \pm 0.00457$
Hierarchy error	0.0102 ± 0.00090	$\mathbf{0.0} \pm 0.0$

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
دانلود کننده مقالات ترجمه شده
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

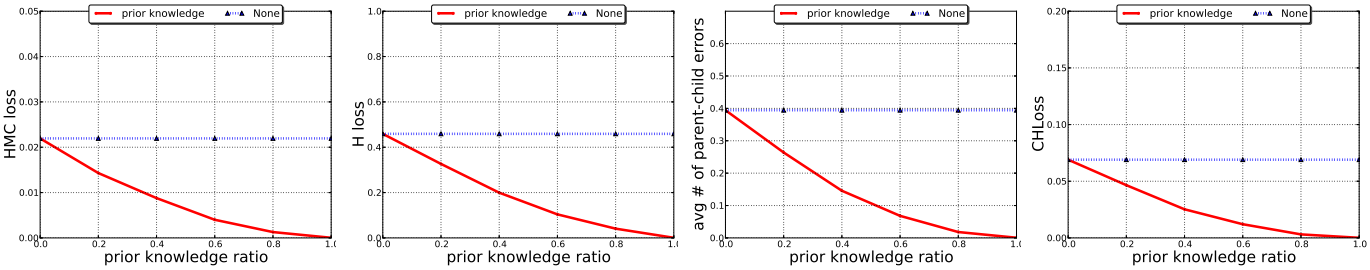


(a) Varying Hamming loss with changing prior knowledge ratio.

(b) Varying precision with changing prior knowledge ratio.

(c) Varying recall during with changing prior knowledge ratio.

(d) Varying F-Measure score with changing prior knowledge ratio.



(e) Varying HMC-Loss with changing prior knowledge ratio.

(f) Varying H-Loss with changing prior knowledge ratio.

(g) Varying avg. Parent-Child Error with changing prior knowledge ratio.

(h) Varying CHLoss with changing prior knowledge ratio.

Fig. 8: Experiment with prior knowledge is conducted over the ticket data."prior knowledge" denotes the hierarchical multi-label classification with knowledge incorporation. "None" denotes the hierarchical multi-label classification without knowledge incorporation.

TABLE IV: Performance comparison between $GLabel$ and "Flat" SVM classification in terms of precision, recall and F-Measure, where both the mean and the 95%-level confidence interval of experiment results are provided.

Metric	**SVM** (incurring hierarchy errors)	$GLabel$ (incurring no hierarchy errors)		
		$w_1 = w_2 = w_3 = w_4 = 1$	$w_1 = w_2 = 2, w_3 = w_4 = 1$	$w_1 = w_2 = 1, w_3 = w_4 = 2$
Precision	0.8844 ± 0.00374	0.8862 ± 0.00396	0.8675 ± 0.00249	$\mathbf{0.9415} \pm 0.00245$
Recall	0.8232 ± 0.00333	0.8247 ± 0.00464	$\mathbf{0.8425} \pm 0.00281$	0.7211 ± 0.00346
F-Measure	0.8527 ± 0.00347	0.8543 ± 0.00426	$\mathbf{0.8548} \pm 0.00256$	0.8167 ± 0.00253

the performance comparison study in terms of precision, recall and F-Measure for different parameter settings. The empirical results are shown in Table IV. It indicates that the $GLabel$ algorithm, which takes the hierarchical constraint into account, can acquire better performance in terms of precision, recall and F-Measure by tuning the parameter setting. Additionally, the experimental results illustrate that high recall can be achieved by increasing values of w_1 and w_2, while high precision can be obtained with larger values of w_3 and w_4.

H. Experiment with prior knowledge

1) ***Setup:*** In order to demonstrate the effectiveness of the proposed $Kilo$ algorithm for knowledge guided hierarchical multi-label classification, we perform the experiments over the same real ticket data set described in section VI-A. The only difference lies in the additional knowledge construction. Each ticket instance in the test data set are associated with a knowledge vector $\mathbf{k}$, where the ith component are exposed with its ground true value randomly according to a predefined prior knowledge ratio $\gamma \in [0, 1]$. The prior knowledge ratio γ is the probability that true labels are observed as either $positive$ or $negative$, while $1-\gamma$ denotes the probability that the label can not be observed before classification. Fig.8 shows the performances in terms of different metrics in comparing the $GLabel$ algorithm without prior knowledge and the one with prior knowledge incorporated by $Kilo$ algorithm.

2) ***Hamming loss with changing prior knowledge ratio:*** Similar to previous configuration, we obtain Hamming loss by setting the CH-loss parameters $w_1 = w_2 = w_3 = w_4 = 1, C_i = 1$. As illustrated in Fig. 8a, Hamming loss drops as the prior knowledge ratio increases, and reaches to zero as prior knowledge ratio achieves one. Fig. 8b, Fig. 8c and Fig. 8d illustrate the performance in terms of Precision, Recall and Precision, respectively. It shows that the knowledge guided algorithm gets better performance as prior knowledge ratio increases.

3) ***HMC-Loss:*** The HMC-Loss is obtained by the same settings as provided in section VI-C. As expected, HMC-Loss decreases as more prior knowledge is provided, shown in Fig. 8e.

4) ***H-Loss:*** The H-Loss is obtained by the same parameter settings as provided in section VI-D. By increasing the

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
سفارش ترجمه این مقاله
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

prior knowledge ratio, H-Loss caused by knowledge guided algorithm decreases, shown in Fig.8f.

*5) **Parent-Child Error**:* The Parent-Child Error is obtained by the same parameter settings as presented in section VI-E. By increasing the prior knowledge ratio, Parent-Child errors caused by knowledge guided algorithm decreases, shown in Fig.8g.

*6) **CH-Loss**:* The CH-Loss is obtained by the same parameter settings as given in section VI-E. By increasing the prior knowledge ratio, CH-Loss caused by knowledge guided algorithm decreases, shown in Fig.8h.

VII. Conclusion and Future Work

In this paper, we employ hierarchical multi-label classification over ticket data to facilitate the problem diagnosis, determination and an automated action, such as auto-resolution or auto-check for enriching or resolving the ticket in the complex IT environments. CH-loss is proposed by considering the contextual misclassification information to support different scenarios in IT environments. In terms of CH-loss, an optimal prediction rule is developed based on Bayes decision theory. This paper comes up with a greedy algorithm $GLabel$ by extending the HIROM algorithm to label the predicting ticket without knowing the number or the maximum number of class labels related to the ticket. Additionally, taking the real scenario in practice into account, the $Kilo$ algorithm is proposed to utilize the knowledge from the domain expert during routine IT maintenance procedure to effectively improve the IT problem category determination.

Our work focuses on tree-based hierarchy, which can be extended to DAG-based hierarchy in future work. In addition, more domain expert knowledge can be automatically incorporated into the framework to reduce the system administrators' involvement in the overall system proposed in this paper. Based on $Kilo$ algorithm, another direction is to propose an framework which is capable of refining the category determination interactively with further knowledge.

Acknowledgement

The work is partially supported by the National Science Foundation under grants CNS-1126619, IIS-1213026 and CNS-1461926, Chinese National Natural Science Foundation under grant 91646116, Scientific and Technological Support Project (Society) of Jiangsu Province No. BE2016776, and an FIU Dissertation Year Fellowship.

References

[1] "ITIL," http://www.itil-officialsite.com/home/home.aspx, 2016.
[2] L. Tang, T. Li, L. Shwartz, F. Pinel, and G. Y. Grabarnik, "An integrated framework for optimizing automatic monitoring systems in large IT infrastructures," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 1249–1257.
[3] Y. Jiang, C.-S. Perng, T. Li, and R. Chang, "Intelligent cloud capacity management," in *Network Operations and Management Symposium (NOMS), 2012 IEEE*. IEEE, 2012, pp. 502–505.
[4] WIKI, 2016, https://en.wikipedia.org/wiki/Network-attached_storage.
[5] J. De Knijff, F. Frasincar, and F. Hogenboom, "Domain taxonomy learning from text: The subsumption method versus hierarchical clustering," *Data & Knowledge Engineering*, 2012.
[6] X. Liu, Y. Song, S. Liu, and H. Wang, "Automatic taxonomy construction from keywords," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2012, pp. 1433–1441.
[7] L. Li and T. Li, "An empirical study of ontology-based multi-document summarization in disaster management," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2013.
[8] O. Dekel, J. Keshet, and Y. Singer, "Large margin hierarchical classification," in *Proceedings of the Twenty-first International Conference on Machine Learning*. ACM, 2004, p. 27.
[9] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *The Journal of Machine Learning Research*, vol. 7, pp. 31–54, 2006.
[10] N. Cesa-Bianchi and G. Valentini, "Hierarchical cost-sensitive algorithms for genome-wide gene function prediction," 2010.
[11] W. Bi and J. T. Kwok, "Multi-label classification on tree-and dag-structured hierarchies," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 17–24.
[12] L. Tang, T. Li, L. Shwartz, and G. Grabarnik, "Recommending Resolutions for Problems Identified by Monitoring," pp. 134–142, 2013.
[13] L. Cai and T. Hofmann, "Exploiting known taxonomies in learning overlapping concepts." in *IJCAI*, vol. 7, 2007, pp. 708–713.
[14] C. Zeng, T. Li, L. Shwartz, and G. Y. Grabarnik, "Hierarchical multi-label classification over ticket data using contextual loss," in *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014, pp. 1–8.
[15] C. Bozdogan and N. Zincir-Heywood, "Data mining for supporting it management," in *2012 IEEE Network Operations and Management Symposium*. IEEE, 2012, pp. 1378–1385.
[16] C. Bozdogan, A. N. Zincir-Heywood, and Y. Gokcen, "Automatic optimization for a clustering based approach to support it management," in *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. IEEE, 2013, pp. 1233–1236.
[17] S. Sozuer, C. Etemoglu, and E. Zeydan, "A new approach for clustering alarm sequences in mobile operators," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 1055–1060.
[18] H. M. Tran and J. Schönwälder, "Discariałdistributed case-based reasoning system for fault management," *IEEE Transactions on Network and Service Management*, vol. 12, no. 4, pp. 540–553, 2015.
[19] C. Zeng, L. Tang, W. Zhou, T. Li, L. Shwartz, G. Grabarnik *et al.*, "An integrated framework for mining temporal logs from fluctuating events," *IEEE Transactions on Services Computing*, 2016.
[20] C. Zeng, L. Tang, T. Li, L. Shwartz, and G. Y. Grabarnik, "Mining temporal lag from fluctuating events for correlation and root cause analysis," in *10th International Conference on Network and Service Management (CNSM) and Workshop*. IEEE, 2014, pp. 19–27.
[21] IBMTivoli, "IBM Tivoli Monitoring," https://www.ibm.com/software/tivoli, 2016.
[22] HPOpenView, "HP OpenView : Network and Systems Management Products," http://www8.hp.com/us/en/software/enterprise-software.html.
[23] logicMonitor, "LogicMonitor," http://www.logicmonitor.com/, 2016.
[24] zenoss, "zenoss," http://ownit.zenoss.com/, 2016.
[25] ManageEngine, "ManageEngine," https://www.manageengine.com/, 2016.
[26] S. Kashyap, J. Ramamirtham, R. Rastogi, and P. Shukla, "Efficient constraint monitoring using adaptive thresholds," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008, pp. 526–535.
[27] S. Agrawal, S. Deb, K. Naidu, and R. Rastogi, "Efficient detection of distributed constraint violations," in *2007 IEEE 23rd International Conference on Data Engineering*. IEEE, 2007, pp. 1320–1324.
[28] L. Tang, T. Li, F. Pinel, L. Shwartz, and G. Grabarnik, "Optimizing system monitoring configurations for non-actionable alerts," in *Network Operations and Management Symposium (NOMS)*. IEEE, 2012, pp. 34–42.
[29] J. Lin, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, "Automated anomaly detection and root cause analysis in virtualized cloud infrastructures," in *Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP*. IEEE, 2016, pp. 550–556.
[30] E. Martinez, E. Fallon, S. Fallon, and M. Wang, "Adamantłan anomaly detection algorithm for maintenance and network troubleshooting," in *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2015, pp. 1292–1297.
[31] Y. Jiang, C. Zeng, J. Xu, and T. Li, "Real time contextual collective anomaly detection over multiple data streams," *Proceedings of the ODD*, pp. 23–30, 2014.

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2017.2668363, IEEE Transactions on Network and Service Management
Downloaded from http://iranpaper.ir
http://www.itrans24.com/landing1.html
سفارش ترجمه این مقاله
آنلاین: www.trans24.ir
تماس: (۰۲۱) ۴۰-۶۶۵۷۲۲۳۸
trans24

[32] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Hierarchical classification: combining bayes with svm," in *Proceedings of the 23rd International Conference on Machine Learning*. ACM, 2006, pp. 177–184.

[33] S. Dumais and H. Chen, "Hierarchical classification of web content," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2000, pp. 256–263.

[34] T. Li, C. Zhang, and S. Zhu, "Empirical studies on multi-label classification," in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI2006)*, 2006.

[35] O. Dekel, J. Keshet, and Y. Singer, "An online algorithm for hierarchical phoneme classification," in *Machine Learning for Multimodal Interaction*. Springer, 2005, pp. 146–158.

[36] M. Granitzer, "Hierarchical text classification using methods from machine learning," *Master's Thesis, Graz University of Technology*, 2003.

[37] T. Hofmann, L. Cai, and M. Ciaramita, "Learning with taxonomies: Classifying documents and words," in *NIPS Workshop on Syntax, Semantics, and Statistics*, 2003.

[38] M. E. Ruiz and P. Srinivasan, "Hierarchical text categorization using neural networks," *Information Retrieval*, vol. 5, no. 1, pp. 87–118, 2002.

[39] A. Sun and E.-P. Lim, "Hierarchical text classification and evaluation," in *Proceedings IEEE International Conference on Data Mining*. IEEE, 2001, pp. 521–528.

[40] T. Li, S. Zhu, and M. Ogihara, "Hierarchical document classification using automatically generated hierarchy," *J. Intell. Inf. Syst.*, vol. 29, no. 2, pp. 211–230, Oct. 2007. [Online]. Available: http://dx.doi.org/10.1007/s10844-006-0019-7

[41] H. Blockeel, L. Schietgat, J. Struyf, S. Džeroski, and A. Clare, *Decision trees for hierarchical multilabel classification: A case study in functional genomics*. Springer, 2006.

[42] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Machine Learning*, vol. 73, no. 2, pp. 185–214, 2008.

[43] B. Wei and T. K. James, "Hierarchical multilabel classification with minimum bayes risk," in *Proceedings of the 12th International Conference on Data Mining*. IEEE, 2012.

[44] B. Hariharan, L. Zelnik-Manor, M. Varma, and S. Viswanathan, "Large scale max-margin multi-label classification with priors," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 423–430.

[45] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management*. ACM, 2004, pp. 78–87.

[46] J. Platt *et al.*, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Advances in Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, 1999.

[47] C. Zeng, Y. Jiang, L. Zheng, J. Li, L. Li, H. Li, C. Shen, W. Zhou, T. Li, B. Duan, M. Lei, and P. Wang, "FIU-Miner: A Fast, Integrated, and User-Friendly System for Data Mining in Distributed Environment," in *Proceedings of the Nineteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013.

[48] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.

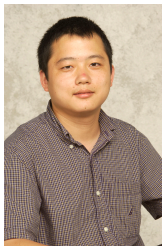
[49] R. G. Baraniuk and D. L. Jones, "A signal-dependent time-frequency representation: Fast algorithm for optimal kernel design," *Signal Processing, IEEE Transactions on*, vol. 42, no. 1, pp. 134–146, 1994.

[50] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008, vol. 1.

[51] "libSVM," 2015, https://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Chunqiu Zeng is currently a Ph.D. student in the School of Computing and Information Sciences at Florida International University. He received B.S. and M.S. degree in Computer Science from Sichuan University in 2006 and 2009, respectively. His research interest includes event mining, system management and large scale data mining.

Wubai Zhou is currently a Ph.D. student in the School of Computing and Information Sciences at Florida International University. He received B.S. degree in Computer Science and Technology from Wuhan University in 2012. His research interest includes system oriented data mining, system management, machine learning.



Tao Li is currently a professor in the School of Computing and Information Sciences at Florida International University. He is also a professor in the School of Computer Science at Nanjing University of Posts and Telecommunications. He received his Ph.D degree in Computer Science in 2004 from the University of Rochester. His research interests are in data mining, computing system management, and information retrieval. He is a recipient of USA NSF CAREER Award and multiple IBM Faculty Research Awards.

Larisa Shwartz (M'05) received the Ph.D. degree in mathematics from UNISA, University of South Africa. She is currently a Researcher at IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, with research experience in mathematics and computer science. She is currently managing operational innovation focusing on IT service management technologies for service delivery operations. She has more than 40 publications, 32 patents, and 67 patent applications.

Dr. Genady Ya. Grabarnik is currently with Math and CS Department, St John's University. He is a trained mathematician and authored over 80 papers. He spent about 10 years at IBM TJWatson Research Center where his work was celebrated with a number of awards including Outstanding Technical Achievement Award and Research Achievement Awards. He is a prolific inventor with over 50 US patents. His interests include research in functional analysis, inventions, and research in computer science and artificial intelligence.

1932-4537 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.