Hierarchical Multi-Label Classification over Ticket Data using Contextual Loss

Chunqiu Zeng and Tao Li
School of Computer Science
Florida International University
Miami, FL, USA
Email: {czeng001, taoli}@cs.fiu.edu

Larisa Shwartz
Operational Innovations
IBM T.J. Watson Research Center
Yorktown Heights, NY, USA
Email: lshwart@us.ibm.com

Genady Ya. Grabarnik
Dept. of Math & Computer Science
St. John's University
Queens, NY, USA
Email: grabarng@stjohns.edu

***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 tickets to be processed by system administrators. IT problems are naturally organized in a hierarchy by specialization. The problem hierarchy is used to help triage tickets to the processing team for problem resolving. In this paper, a hierarchical multi-label classification method is proposed to classify the monitoring tickets by utilizing the problem hierarchy. 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. An extensive empirical study over ticket data was conducted to validate the effectiveness and efficiency of our method.

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

I. Introduction

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. 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. 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. Events coming from an entire account IT environment are consolidated in an enterprise console, which analyzes the monitoring events and creates an incident ticket in a ticketing system [2]. The information accumulated in the ticket is used by the System Administrators (sysAdmins) for problem determination and resolution. 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 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 and classification in a highly variable service delivery environment. We propose an effective method and a system to address the uncertainty of classification.

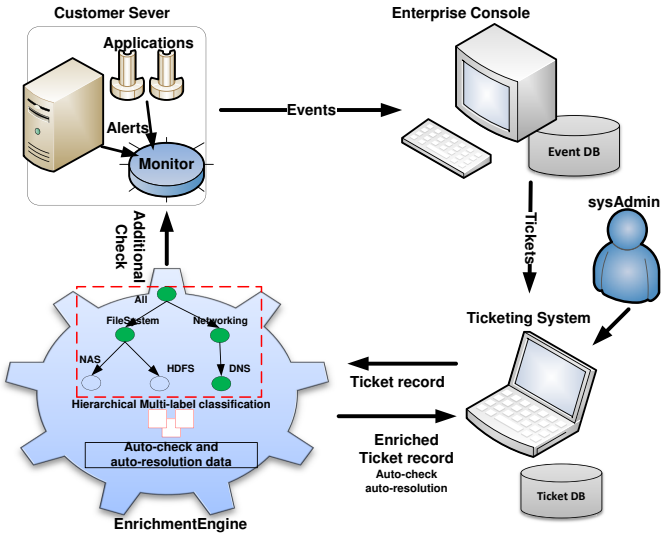


Fig. 1: The system overview

An overview of the system for ticket enrichment and auto-remediation is presented in Fig.1. In the system, the Enrichment Engine allows classification of monitoring tickets by applying hierarchical multi-label classification, then finding the most effective classification based on results from newly introduced loss function, and finally invoking an automated action: auto-resolution script or auto-check

script for enriching or resolving the ticket. In the case where auto-resolution seems unfeasible, the enriched ticket is assigned to a sysAdmin for a review. In this paper we focus on the hierarchical multi-level classification that preserves problems hierarchy and a novel CH loss function.

The description of symptoms of a problem is typically a short text message as shown in Fig.2.(a). The 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], [7]) or it could be created with the help of domain experts [8]. In IT environments, hierarchical multi-label classification could be used not only for the problem diagnosis ([9], [10], [11], [12]), but also for recommending resolutions ([13]) or auto-check scripts. The ticket in our example could have a solution that addresses FileSystem, NAS, Networking and/or Misconfiguration - a highly diversitified action recommendation. Furthermore, based on the hierarchical multi-label classification, actions with different levels in the hierarchy are recommended, where the actions from NAS category are more specific than the ones from FileSystem.

In this paper we first define a new loss function, which takes the contextual misclassification information for each label into consideration and is a generalization of Hamming-loss, H-loss and HMC-loss function [14]. Next, using Bayesian decision theory, we develop the optimal prediction rule by minimizing the conditional risk in terms of proposed loss function. Finally, we propose an efficient algorithm to search the optimal hierarchical multi-labels for each data instance.

The rest of this paper is organized as follows. In section II, we describe related work and identify limitations of existing methods. New loss function for better evaluation of the performance of the hierarchical multi-label classification is proposed in section III. 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

The hierarchical classification problem has been extensively investigated in the past ([15], [16], [17], [9], [18], [19], [20], [21]). As a more general case, the hierarchical multi-label classification, where an instance can be labelled 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 labelled as positive on child label but labelled as negative on parent label. A second approach is to adapt existing single label classification algorithms, such as decision tree([22], [23]). A third approach is based on the first approach but applies some post-process to automatically guarantee the hierarchical consistency ([10], [11], [12], [14]). We focus on the third approach in this paper.

Cesa-Bianchi et al. [10] 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 Cesa-Bianchi in [15]. And Cesa-Bianchi et al. [11] extend the decision rules to the cost-sensitive learning setting by weighting false positive and false negative differently. Wei and James [14] propose the HIROM algorithm to obtain the optimal decision rules with respect to HMC-loss by extending the CSSA algorithm in [12], which has a strong assumption that the number of classes related to each instance is known.

Most approaches that utilize "Flat" classification treat each class label or category as independent by learning an independent binary classifier for each label.

For hierarchy information, hierarchical loss (H-loss) has been proposed in [10]. 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 [14] 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 postive (i.e., FP) with different penalty costs. In Fig.2, assuming a and b are the misclassification penalties for FN and FP respectively, (c) and (d) have 2 FN misclassification errors, so both of them incur $2a$ HMC-loss. Moreover, (e) and (f) suffer $3b$ 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

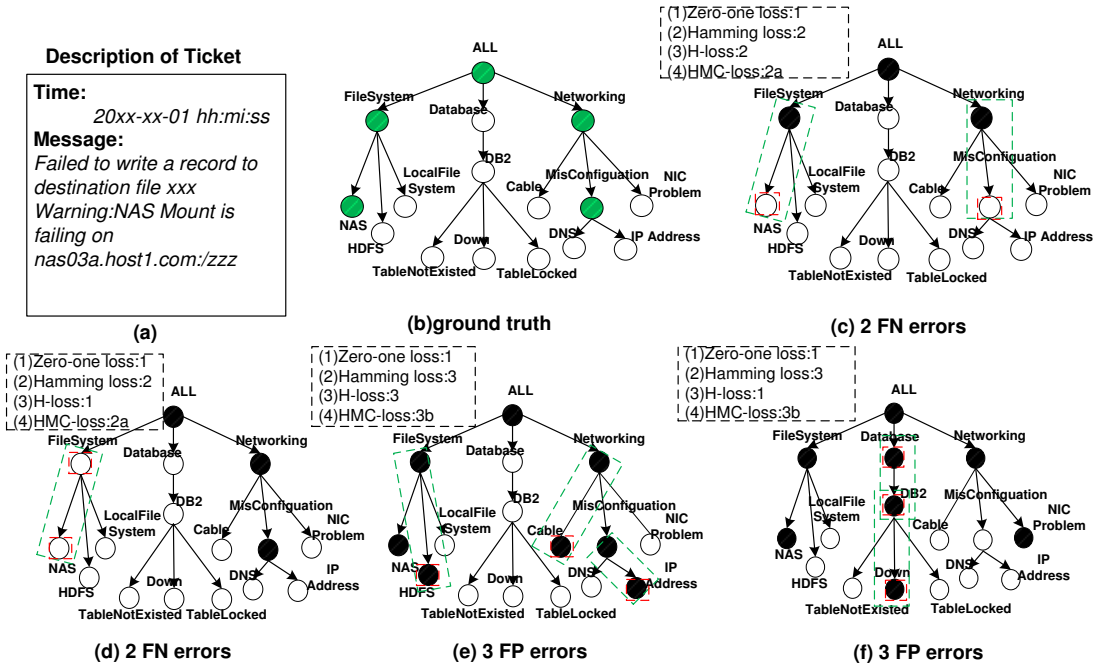


Fig. 2: A hierarchical multi-label classification problem in the IT environment. A ticket instance is shown in (a). (b) presents the ground truth for the ticket with multiple class labels. (c), (d), (e) and (f) are four cases with misclassification. Assuming the cost of each wrong class label is 1, Zero-one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

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

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 compliment 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 leaving 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.

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

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

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.2. *(*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.*

There are four cases of misclassification of node i using Contextual Misclassification Information as shown on Fig. 3.

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.

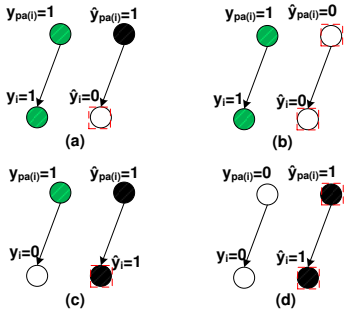


Fig. 3: Four cases of contextual misclassification are shown in (a-d) for node i. Here the left pair is the ground truth; the right pair is the prediction. The misclassified nodes are marked with a red square.

- 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.

Referring to [14],[11], a misclassification cost C_i is given according to the position information of node i in the hierarchy H. And $\{w_i|1 \leq i \leq 4\}$ 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:

$$\ell(\hat{\mathbf{y}}, \mathbf{y}) = w_1 \sum_{i>0}^{N-1} y_i y_{pa(i)} \widetilde{\hat{y}}_i \hat{y}_{pa(i)} C_i + w_2 \sum_{i>0}^{N-1} y_i y_{pa(i)} \widetilde{\hat{y}}_i \widetilde{\hat{y}}_{pa(i)} C_i + w_3 \sum_{i>0}^{N-1} \widetilde{y}_i y_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i + w_4 \sum_{i>0}^{N-1} \widetilde{y}_i \widetilde{y}_{pa(i)} \hat{y}_i \hat{y}_{pa(i)} C_i. \tag{3}$$

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. Due to limited space, all the proofs of the propositions are not presented in this paper.

By setting α and β to be the penalty costs for false negative (FN) and false positive (FP) respectively, and noting that root node is always correctly labelled, the HMC-loss function defined in [14] may be expressed as

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

Proposition III.3. *The HMC-loss function is the special case of CH-loss function when $w_1 = w_2 = \alpha$ and $w_3 = w_4 = \beta$.*

Proposition III.4. *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 [14] 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.3, the Proposition III.4 is obvious.

The H-loss function (see [14]) 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 [10] as follows:

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

Proposition III.5. *The H-loss function is the special case of CH-loss function when $w_1 = 1$, $w_2 = 0$, $w_3 = 1$ and $w_4 = 0$.*

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

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 [10], [11], [14], 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 equations (6) and (9), the computation of loss expectation $LE(\hat{\mathbf{y}}, \mathbf{x})$ can be rewritten using p_i notation as follows:

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 $w_4 > w_3$

TABLE I: special cases of CH-loss

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

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

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.*

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

where

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

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.

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

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

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

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

V. Algorithms and Solutions

As discussed in previous sections, there are three key steps to obtain the hierarchical multi-labeling of the instances having minimal CH-loss.

1) Estimate the probability of p_i for each node i based on the training data.
2) Use p_is to compute the $\sigma(i)$ defined by the equation (15).
3) Obtain the optimal predictor $\hat{\mathbf{y}}^*$ as a solution of the problem (17).

A. Estimating 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 the 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 [24] to convert the real output of the classifier to estimate $P(y_i = 1 | y_{pa(i)} = 1, \mathbf{x})$.

The task of building classifiers for all the nodes is a significant load. 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 [25].

Then, the values of p_i are computed by applying formula 9 while traversing the nodes in the hierarchy. The time complexity of p_i computation is $O(N)$, where N is the number of nodes in the hierarchy.

B. Computing $\sigma(i)$

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 $O(N)$.

C. Obtaining $\hat{\mathbf{y}}^*$

The value $\hat{\mathbf{y}}^*$ is obtained by solving the maximization problem (17). [12] proposed the greedy algorithm CSSA, based on the work in [26] 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 [14], 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 1**) 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, 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 (Fig.4). The labeling procedure stops when the maximum σ value is negative or all nodes are labeled positive.

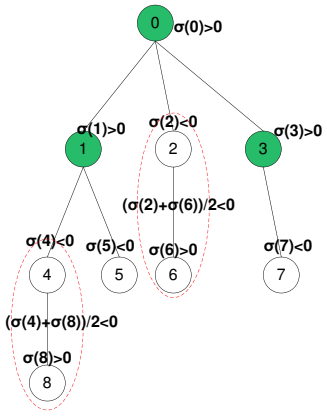


Fig. 4: Figure illustrates hierarchy with 9 nodes and steps of the algorithm 1. Nodes labeled positive are green. A dotted ellipse marks a super node composed of the nodes in it.

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.

Algorithm 1 GLabel

procedure $GLabel$(**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 $U = \mathbf{H}\backslash\{0\}$
  while TRUE **do**
    if all the nodes in **H** are labeled **then**
      return L
    end if
    find the node i with maximum $\sigma(i)$
    if $\sigma(i) < 0$ **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^*
      $\sigma(i^*)$ =average σ values of the two nodes
      put the i^* into U
    end if
  end while
end procedure

VI. Experimentation

A. Setup

We perform the experiments 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. From the whole ticket data set, 3000 tickets are sampled randomly 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.

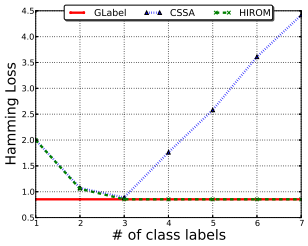
The features for each ticket are built from the short text message describing the symptoms of the problem. First, Natural language processing techniques are applied to remove the stop words and build Part-Of-Speech tags for the words in the text. The nouns, adjectives and verbs in the text are extracted for each ticket. Second, we compute the TF-IDF [27] 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 [28]. 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.

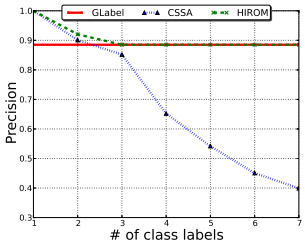
The experiments are mainly conducted by comparing the proposed GLabel algorithm with state-of-the-art algorithms such as CSSA and HIROM. Note, that in the end, we also show benefits of hierarchical classification in comparison to the "Flat" classification.

B. Hamming loss

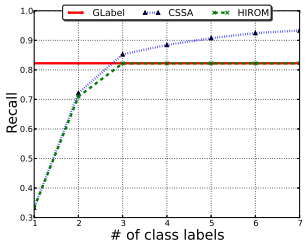
The GLabel algorithm can obtain optimal $\hat{\mathbf{y}}^*$ with minimum Hamming loss by setting the parameters for Hamming loss, since Hamming loss is a special case of CH-loss. Given $w_1 = w_2 = w_3 = w_4 = 1$ for GLabel, $\alpha = \beta = 1$ for HIROM and $C_i = 1$ for both of them, empirical results are displayed in sub-figures (a)-(d) of Fig.5. Sub-figure (a) shows that the GLabel algorithm can automatically find the optimal



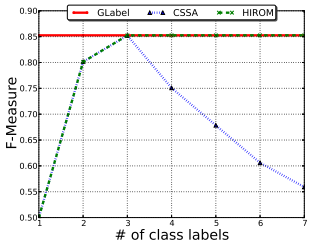
(a) The lowest Hamming loss: CSSA gets 0.8876 at # 3; HIROM gets 0.8534 at # 4; GLabel gets 0.8534.



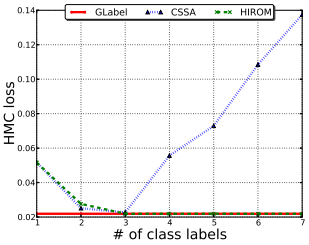
(b) Varying precision during minimizing the Hamming loss



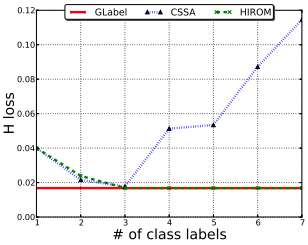
(c) Varying recall during minimizing the Hamming loss



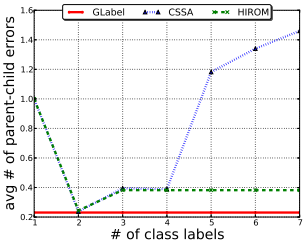
(d) Varying FMeasure score during minimizing the Hamming loss



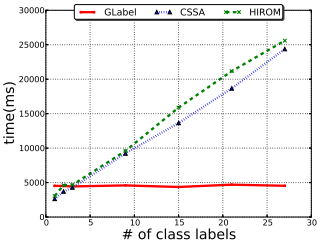
(e) The lowest HMC-Loss: CSSA gets 0.0227 at # 3; HIROM gets 0.0219 at # 4; GLabel gets 0.0219.



(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. 5: Experiments involving tickets

$\hat{\mathbf{y}}^*$ 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 $\hat{\mathbf{y}}^*$. With the increasing number of class labels, HIROM gets lower Hamming loss until it reaches the optimal $\hat{\mathbf{y}}^*$ with minimum Hamming loss by chosing large enough number of class lables. However, CSSA may get larger Hamming loss as the number of class labels increases. The sub-figure (b)-(d) show as in comparison to Hamming loss, Glabel algorithm shows good performance in Precision, Recall and FMeasure score.

C. HMC-loss

The HMC-loss considers loss with respect to the node position in the hierarchy. Following [14], we define the C_i as follows.

$$C_i = \begin{cases} 1, & i = 0; \\ \frac{C_{pa(i)}}{\text{\# of i's siblings}}, & i > 0. \end{cases} \tag{18}$$

To simplify, we set $w_1 = w_2 = w_3 = w_4 = 1$ for Glabel and $\alpha = \beta = 1$ for HIROM as well. The sub-figure(e) shows that Glabel algorithm obtains the same lowest HMC-loss as HIROM algorithm does, with the HIROM tuned for minimizing the HMC-loss.

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 and $\alpha = \beta = 1$ for HIROM, $C_i = 1$ for all the three algorithms. The sub-figure(f) shows that GLabel gets the lowest H-loss in comparison to HIROM and CSSA minimums. HIROM and CSSA algorithms cannot get the optimal $\hat{\mathbf{y}}^*$ with minimal H-loss.

E. Misclassifications occur 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 (18). The sub-figure(g), GLabel reaches the minimum average number of parent-child errors, while CSSA and HIROM algorithms do not minimize the parent-child errors since they 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 sub-figure (h). We run three algorithms for 40 rounds and get the average time consumed. The sub-figure (h) shows that run time of GLabel is independent from the number of labels, while run time of other 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 (18). In addition, we define the hierarchy error as the average number of violated hierarchy constraints.

Metric	SVM	GLabel
CH-loss	4.2601	2.6889
Parent-child error	0.3788	0.1729
Hierarchy error	0.0102	0.0

TABLE II: Comparison with "Flat" classification

The table II shows that GLabel algorithm has better performance in terms of CH-loss and parent-child error. Furthermore, the "Flat" SVM classifiction suffers on average 0.0102 hierarchy errors with each ticket, while GLabel complies with the hierarchy constraint and does not have hierarchy errors.

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 Bayesian 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.

In this paper, we focused 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.

ACKNOWLEDGEMENT

The work of C. Zeng and T. Li is partially supported by the National Science Foundation under grants CNS-1126619 and IIS-1213026 and by the Army Research Office under grant number W911NF-1010366 and W911NF-12-1-0431.

REFERENCES

[1] "ITIL," http://www.itil-officialsite.com/home/home.aspx.

[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, 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] C. Shiyu, Q. Guo-Jun, T. Jinhui, T. Qi, R. Yong, and H. Thomas, "Multimedia lego: Learning structured model by probabilistic logic ontology tree," in *IEEE International Conference on Data Mining (ICDM)*, December 2013.

[8] 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.

[9] 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.

[10] 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.

[11] N. Cesa-Bianchi and G. Valentini, "Hierarchical cost-sensitive algorithms for genome-wide gene function prediction," 2010.

[12] 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.

[13] L. Tang, T. Li, L. Shwartz, and G. Grabarnik, "Recommending Resolutions for Problems Identified by Monitoring," pp. 134–142, 2013.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

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

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

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

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

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

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

[28] "libSVM," http://www.csie.ntu.edu.tw/~cjlin/libsvm.