Multi-source Hierarchical Prediction Consolidation

Chenwei Zhang[†] Sihong Xie[♯] Yaliang Li[‡] Jing Gao[‡] Wei Fan[♮] Philip S. Yu[†§]
[†]Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA
[♯]Computer Science and Engineering Department, Lehigh University, Bethlehem, PA, USA
[‡]SUNY Buffalo, Buffalo, NY, USA
[♮]Baidu Research Big Data Lab, Sunnyvale, CA, USA
[§]Institute for Data Science, Tsinghua University, Beijing, China
[†]{czhang99,psyu}@uic.edu, [♯]sxie@cse.lehigh.edu, [‡]{yaliangl,jing}@buffalo.edu, [♮]fanwei03@baidu.com

ABSTRACT

In big data applications such as healthcare data mining, due to privacy concerns, it is necessary to collect predictions from multiple information sources for the same instance, with raw features being discarded or withheld when aggregating multiple predictions. Besides, crowd-sourced labels need to be aggregated to estimate the ground truth of the data. Because of the imperfect predictive models or human crowdsourcing workers, noisy and conflicting information is ubiquitous and inevitable. Although state-of-the-art aggregation methods have been proposed to handle label spaces with flat structures, as the label space is becoming more and more complicated, aggregation under a label hierarchical structure becomes necessary but has been largely ignored. These label hierarchies can be quite informative as they are usually created by domain experts to make sense of highly complex label correlations for many real-world cases like protein functionality interactions or disease relationships.

We propose a novel multi-source hierarchical prediction consolidation method to effectively exploits the complicated hierarchical label structures to resolve the noisy and conflicting information that inherently originates from multiple imperfect sources. We formulate the problem as an optimization problem with a closed-form solution. The proposed method captures the smoothness over all information sources as well as penalizing any consolidation result that violates the constraints derived from the label hierarchy. The hierarchical instance similarity as well as the consolidation result are inferred in a totally unsupervised, iterative fashion. Experimental results on both synthetic and real-world data sets show the effectiveness of the proposed method over existing alternatives.

1. INTRODUCTION

For various tasks such as crowdsourcing, healthcare data mining in big data applications, multiple information sources may provide labeling information on the same instance simultaneously. For example, in crowdsourcing tasks, multiple human annotators are asked to find labels of flowers given a beautiful nature image. On online healthcare forums, a patient who posts a question regarding his/her symptoms may receive disease names as suggestions from multiple doctors.

Once we obtained the labeling information from multiple information sources or human beings, it is necessary to consolidate the collected information to infer the ground truth labels. Because imperfect information from a single information source exists ubiquitously, it is also important that labeling information from multiple sources need to be consolidated to resolve noises and conflicts. Moreover, due to privacy concerns, raw features of instances are often discarded or withheld and only labels are available for aggregation purposes. For example in online healthcare forums, the raw features of a patient need to be discarded for privacy concerns and only diseases names collected from multiple doctors are consolidated to infer the ground truth.

In those applications, instead of assigning a single label for each instance, it is usually more informative to associate an instance with more than one labels to characterize multiple categories or properties an instance has. For example, an image instance can be described by multiple tags such as "monarda", "bird" and hence belongs to multiple categories. A protein can be associated with more than one functions, denoting various functionalities. A patient may be associated with several candidate diseases, each of them diagnosed by a doctor.

Typically, those tasks consider all the labels on the same "flat" level. However, it is still insufficient to measure the value of the informativeness of labels when we isolate labels with each other and ignore the correlations between labels [1]. A better way is to organize labels in a hierarchical taxonomy. In this way, besides correlations such as co-occurrences between all the "flat" labels, a label hierarchy contains rich information to make sense of highly complex label correlations.

Problem Studied: In this paper, we want to incorporate the label hierarchy into the prediction consolidation process when only the labeling information from multiple information sources are available, which is formally defined as the Multi-source Hierarchical Prediction Consolidation(MHPC) problem, illustrated in Figure 1.

Informative label hierarchies are prevalently observed in various applications. For example, in crowdsourcing for protein functionality annotation, the functional labels of a protein are in a hierarchy, representing the functional relations. In healthcare data mining, disease labels can be also constructed as a tree-like disease taxonomy, denoting

ACM ISBN .
DOI:

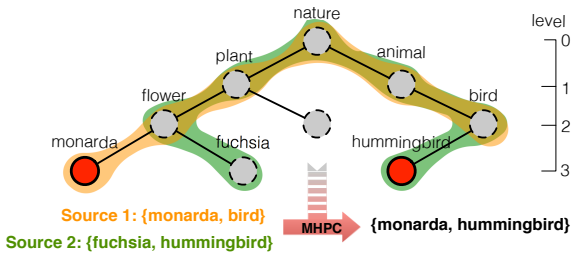

Figure 1: An illustrative example of multi-source hierarchical prediction consolidation problem. Two individual information sources give label predictions to an image and MHPC tries to find a consolidated label prediction that maximizes the consensus among these predictions while preserving the structures of the label hierarchy.

the pathology structure of human diseases. Current works in the literature [2, 3, 4] try to consolidate predictions from multiple information source. However, those works have access to raw features of the data and they totally ignore the hierarchical information. Simply ignore the label hierarchy may lead to potential loss of valuable information [5].

With a number of approaches proposed to exploit the label hierarchy for various tasks [6, 7] such as classification [8] or clustering [9], it is reasonable to believe that informative label hierarchies, which inherently come with many prediction consolidation tasks, are also able to offer auxiliary and valuable information for the MHPC problem. For example, the conflicts between two label predictions in Figure 1 can be resolved by mapping label predictions to level 2, where we have {flower, bird} for both label predictions. Moreover, the label hierarchy may provide some constrains so that label predictions which violate the hierarchical structure will be less useful.

Given the importance of incorporating the label hierarchy, the MHPC problem itself is a novel problem which is rarely studied. Various learning problems are summarized in Figure 2, where model-level ensemble learning [10, 11, 12] tries to aggregate labels at the output level and hierarchical multi-label learning exploits the label hierarchy to improve the model performance on a wide range of multi-label learning tasks [5, 13, 8]. The multi-source hierarchical prediction consolidation problem is an unsupervised ensemble learning problem that aggregates hierarchical multi-label predictions on the model-level, where very few work has been done.

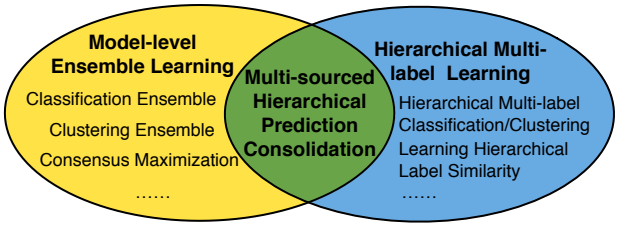

Figure 2: Position of the multi-source hierarchical prediction consolidation problem.

Despite the importance and novelty, the multi-source hierarchical prediction consolidation problems are challenging to solve due to:

- **Label vagueness**: label vagueness usually originates from imperfect predictive models or insufficient knowledge of information sources. When an information source has insufficient knowledge or high uncertainty, vagueness is commonly observed where a vague, generalized label is used instead of a mandatory, specific label prediction. For example in Figure 1, instead of having a specific type of flower (e.g. "monarda") as the label prediction, which is the leaf node in a label hierarchy, usually a more generalized label "flower" is used when the information source doesn't know much about flowers. Leaf node predictions are more likely to be error-prone when they are mandatory provided under insufficient knowledge. As the label vagueness is widely observed, how can we exploit the rich information encoded within label hierarchy to resolve the vagueness?

- **Label ambiguity**: in most cases, the predictions we collect from multiple information sources are noisy and conflicting with each other. For instance, in Figure 1, Source 2 associates a label "fuchsia" to the instance instead of giving "monarda", one of the truth labels. Those two labels have similar meanings but may be used interchangeably by different information sources due to ambiguity. Ambiguous label predictions may contain erroneous information and hence introduce noises into prediction consolidation tasks. How should the prediction consolidation task resolve the label ambiguity from multi-source predictions effectively?

- **Label sparsity**: since an information source may provide labeling information for a diverse population of instances, labels in each prediction only cover a very small portion of the whole, diversified label space. Also, many of the labels may be only covering a small number of instances. For example, there can be thousands of flower names under the node "flower", but not all information sources necessarily mention the idea of "flower" ever. Not all flower names are covered by all instances as well. The worst case exists when the truth label of an instance is not provided by any information source. That is, for the example in Figure 1, "monarda" is never mentioned by any of the information sources. How to deal with the label sparsity that comes with the ever-expanding label space as well as the varying number of predictions obtained from multiple information sources?

In this paper, we try to solve the MHPC problem by formulating it as an optimization task. The objective function for optimization favors the smoothness over all information sources as well as penalizing any two instances which have high hierarchical instance similarity but conflict with each other in the consolidation result. We derive a closed-form solution for this optimization problem. After that, the MPHC algorithm is introduced where two phases, namely estimating hierarchical similarities and minimizing consensus cost, are conducted in an iterative, totally unsupervised fashion to get the consolidated label prediction for each instance.

Table 1: Table of symbols.

Symbol	Description
M	Number of information sources
N	Number of instances
K	Number of labels in the label hierarchy
$\mathbf{Y_m}$	$\mathbb{R}^{N\times K}$ label matrix from source m
$\hat{\mathbf{Y}}$	$\mathbb{R}^{N\times K}$ consolidated label matrix
$\mathbf{Y_{m}}_{(i)}$	Label vector for the i-th instance in Y_m
$\mathbf{Y_{m}}^{k}_{(i)}$	The k-th label of $\mathbf{Y_{m}}_{(i)}$
$\mathbf{Z}$	$\mathbb{R}^{N\times K}$ ground truth label matrix
$\mathbf{H}$	$\mathbb{R}^{K\times K}$ hierarchical adjacency matrix of labels where $H_{kk'}=1$ when label k is the direct descendant of label k'. Otherwise $H_{kk'}=0$.
$\mathbf{W}$	$\mathbb{R}^{N\times N}$ hierarchical instance similarity matrix
$\vec{\mathbf{S}}$	$\mathbb{R}^{1\times K}$ label support vector. $\vec{\mathbf{S}}_k$ is the corresponding support value for label k.
$\vec{\mathbf{C}}$	$\mathbb{R}^{1\times K}$ label occurrence vector. $\vec{\mathbf{C}}_k$ is the corresponding occurrence value for label k in all augmented label vectors.

2. PROBLEM STATEMENT

Before introducing the proposed method, we will give the definitions of some important concepts and formulation of the MHPC problem first in this section.

2.1 Terminology Definition

Suppose we are given label matrices $\mathbf{Y}_1$, $\mathbf{Y}_2$,...,$\mathbf{Y}_M$ provided by M information sources and a hierarchical adjacency matrix $\mathbf{H}$. An label matrix $\mathbf{Y}_m$ is an N by K matrix indicating the labeling information on all N instances and K labels provided by the information source m. Within $\mathbf{Y}_m$, the i-th row $\mathbf{Y}_{m(i)}$ is a label vector of instance i provided by information source m, where $\mathbf{Y}_{m(i)}^{k}=1$ if the information source m associates the instance i with label k. A hierarchical adjacency matrix $\mathbf{H}$ embodies the hierarchical structure for all K labels where $\mathbf{H}_{kk'}=1$ if and only if when k is the direct descendant of label k'. Other terminologies are introduced further when they are used. Table 1 summarizes the notations.

2.2 Problem Statement

Based on the terminologies defined above, the **Multi-source Hierarchical Prediction Consolidation** problem is formally defined as: given label matrices $\{\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_m\}$ and a hierarchical adjacency matrix $\mathbf{H}$, the MHPC problem tries to incorporate the label hierarchy into finding a consolidated label matrix $\hat{\mathbf{Y}}$ which maximizes the consensus among all label matrices. The maximum consensus is achieved by minimizing the consensus cost in prediction consolidation.

3. MODELING HIERARCHICAL CONSENSUS

This section describes how we come up with a consolidation agreement of multiple information sources by incorporating the label hierarchy into the minimization of consensus cost. Section 3.1 describes the objective function for minimizing the consensus cost as well as its closed-form solution. Section 3.2 integrates the hierarchical information into the objective function when estimating instance similarities.

3.1 Minimizing the Consensus Cost

We formulate the following objective function:

$$\min_{\hat{\mathbf{Y}}} \frac{1}{M}\sum_{m=1}^{M}\left\|\hat{\mathbf{Y}}-\mathbf{Y}_m\right\|_F^2+\lambda\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{W}_{ij}\left\|\hat{\mathbf{Y}}_{(i)}-\hat{\mathbf{Y}}_{(j)}\right\|_2^2 \quad s.t. \quad \lambda\geqslant 0, \tag{1}$$

where $\|\,.\,\|$ denotes the $\ell 2$-norm for matrices and vectors. The first term favors the smoothness of the consolidation result over label predictions from all the information sources. The second term serves as a regularization term which ensures that label vectors of any two instances in the consolidation result(the i-th and the j-th instance in $\hat{\mathbf{Y}}$) do not differentiate themselves from each other very much if they share a high hierarchical instance similarity $\mathbf{W}$. Estimation for $\mathbf{W}$ will be further explained in Section 3.2. λ serves as a regularization coefficient to penalize violation for the hierarchical similarity constraints.

Note that the objective function in Equation 1 can be rewritten in a matrix form as:

$$\mathrm{J}(\hat{\mathbf{Y}})=\frac{1}{M}\sum_{m=1}^{M}\Big(\mathrm{Tr}(\hat{\mathbf{Y}}^{\mathrm{T}}\hat{\mathbf{Y}})-2\,\mathrm{Tr}({\mathbf{Y}_m}^{\mathrm{T}}\hat{\mathbf{Y}})+\mathrm{Tr}({\mathbf{Y}_m}^{\mathrm{T}}\mathbf{Y}_m)\Big)+\lambda\,\mathrm{Tr}(\hat{\mathbf{Y}}^{\mathrm{T}}\mathbf{L}\hat{\mathbf{Y}}) \quad s.t. \quad \lambda\geqslant 0, \tag{2}$$

where $\mathbf{L}$ is the symmetric normalized Laplacian matrix

$$\mathbf{L}=\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}} \tag{3}$$

and $\mathbf{D}$ is the degree matrix of $\mathbf{W}$.

To find a $\hat{\mathbf{Y}}$ that gives the minimum value of $\mathrm{J}(\hat{\mathbf{Y}})$, we first prove the convexity of $\mathrm{J}(\hat{\mathbf{Y}})$ by showing the positive definite property of the Hessian matrix of $\mathrm{J}(\hat{\mathbf{Y}})$ with respect to $\hat{\mathbf{Y}}$.

$$\frac{\partial^2\mathrm{J}(\hat{\mathbf{Y}})}{\partial\hat{\mathbf{Y}}^2}=2\,(\mathbf{I}+\lambda\mathbf{L}) \quad s.t. \quad \lambda\geqslant 0. \tag{4}$$

The Hessian matrix shown in Equation 4 is positive definite because $\lambda\mathbf{L}$ is positive-semidefinite [14] and adding an identity matrix to it makes the resulting Hessian matrix positive definite [15]. Therefore, setting the derivative of $\mathrm{J}(\hat{\mathbf{Y}})$ with respect to $\hat{\mathbf{Y}}$ to zero

$$\frac{\partial\,\mathrm{J}(\hat{\mathbf{Y}})}{\partial\hat{\mathbf{Y}}}=\frac{1}{M}\sum_{m=1}^{M}\Big(2\hat{\mathbf{Y}}-2\mathbf{Y}_m\Big)+2\lambda\mathbf{L}\hat{\mathbf{Y}}=0 \tag{5}$$

leads to a global minimized consensus cost $\mathrm{J}(\hat{\mathbf{Y}})$ given $\mathbf{W}$.

A closed-form solution can be obtained by solving the Equation 5:

$$\begin{aligned}\hat{\mathbf{Y}}&=\Big(1+\lambda(\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}})\Big)^{-1}\frac{1}{M}\sum_{m=1}^{M}\mathbf{Y}_m\\&=(\mathbf{I}+\lambda\cdot\mathbf{L}))^{-1}\bar{\mathbf{Y}},\end{aligned} \tag{6}$$

where $\bar{\mathbf{Y}}=\frac{1}{M}\sum_{m=1}^{M}\mathbf{Y}_m$ and $\mathbf{I}$ is the identity matrix.

From Equation 6, we can see that hierarchical similarity constraints are not introduced when $\lambda=0$. In this case, the consolidation result degrades to the simple averaging of all the label prediction we obtained. While $\lambda>0$, the Laplacian matrix $\mathbf{L}$ regularizes the simple averaging result and guides

the $\bar{\mathbf{Y}}$ towards a global consensus $\hat{\mathbf{Y}}$ with label hierarchies being considered.

It is worth mentioning that the formulation of the hierarchical similarity constraints as the second term in Equation 1 and Equation 2 can be also seen as learning an optimal embedding [16] from a multi-source label space to a consolidated label space. The multi-source label space contains multi-source hierarchical label predictions with noisy and conflicting labels. While the consolidated label space has labels with less imperfect information as well as a minimized consensus cost. If $\hat{\mathbf{Y}}$ is such an embedding result, then a reasonable criterion for a "good" mapping is to have weight $\mathbf{W}_{ij}$ so that it heavily penalizes two "hierarchically similar" label predictions (instances i and j) for not having "similar" label predictions in the consolidated label space after the mapping.

3.2 Estimating the Hierarchical Similarity

Given any two label vectors, each of which denotes a label prediction for an instance, an instance similarity matrix $\mathbf{W}$ in Equation 1 measures similarities between label vectors. We assume that each label has a unique degree of support, asserting that such label belongs to an instance. Then, the instance similarity value $\mathbf{W}_{ij}$ of any two instance i and instance j can be calculated by the following Equation:

$$\mathbf{W}_{ij} = \exp\left(-\frac{1}{\sigma}\sqrt{\sum_{k=1}^{K}\vec{S}_k(\hat{\mathbf{Y}}^k_{(i)} - \hat{\mathbf{Y}}^k_{(j)})^2}\right), \qquad (7)$$

where $\sqrt{\sum_{k=1}^{K}\vec{S}_k(\hat{\mathbf{Y}}^k_{(i)} - \hat{\mathbf{Y}}^k_{(j)})^2}$ in Equation 7 measures the weighted Euclidean distance between label vectors of two consolidated instances in $\hat{\mathbf{Y}}$, namely $\hat{\mathbf{Y}}_{(i)}$ and $\hat{\mathbf{Y}}_{(j)}$, over all K labels. $\vec{S}$ is a support label vector. Each entry $\vec{S}_k$ of $\vec{S}$ indicates the degree of support of the label k to the distance estimation. Note that the support value $\vec{S}_k$ can be seen as how much contribution a label k makes to the overall similarity estimation of an instance having that label. σ is a constant factor and the exponential function $\exp(.)$ converts weighted Euclidean distance measurement to a similarity measurement.

Usually, we assume that each label has an individual degree of support to the overall similarity estimation. For example, in online healthcare forums, each disease label may have an individual support to a diagnose (each diagnose consists of several disease labels), when we estimate the similarity of two diagnoses. However, as the label space is becoming more and more complicated, such simplified assumption totally ignores correlations among labels and therefore will lead to an inaccurate similarity estimation due to the label sparsity. For example, there can be thousands of labels for flower names such as "monarda", "fuchsia" and so on. However, it is very unlikely that label predictions provided by various information sources cover all of those flower names. Also, different information sources may have preferences in providing certain labels so the remaining labels may be rarely used. Although these labels share the same general idea (the "flower"), since the support value is calculated on each label separately, none of those labels is able to make any contribution to the support value of label "flower" with which they share the general idea.

The label hierarchy organizes labels in a tree-like structure in which general labels are the ancestors of specific labels. With a label hierarchy being incorporated, labels are no longer independent with each other: the general idea is that when an information source assigns the label k to an instance, the existence of that label reflects a direct occurrence of this label to support the instance. Moreover, such label assignment on label k also indicates indirect occurrences, although not explicitly labeled, from its ancestor labels on the label hierarchy. Therefore, we would like to let the occurrence of each label contributes not only to itself, but also to all its ancestor labels in a label hierarchy.

To make this happen, we first apply the label augmentation algorithm to convert each label vector $\mathbf{Y}_{\mathbf{m}(i)}$ to an augmented label vector $\mathbf{Y}^*_{\mathbf{m}(i)}$, as shown in Algorithm 1. By augmenting the occurrence of a label to its ancestor labels, occurrences of those augmented labels provide more implicit, but in-depth labeling information about an instance. The label augmentation for each label vector can be also seen as mapping the original label vector to all the ancestor levels in a bottom-up fashion. Figure 3 illustrates this idea.

Algorithm 1 Label Augmentation

Input: A label vector $\mathbf{Y}_{m(i)}$
A hierarchical adjacency matrix $\mathbf{H}$
Output: An augmented label vector $\mathbf{Y}^{k*}_{m(i)}$.

```
function LabelAugmentation(Y_{m(i)}, H)
    Initialize Y*_{m(i)} as a zero vector
    for each k with Y^k_{m(i)} = 1 do
        t ← k
        while t ≠ ROOT do
            Y^{t*}_{m(i)} ← 1
            t ← t' where H_{tt'} = 1
return Y*_{m(i)}
```

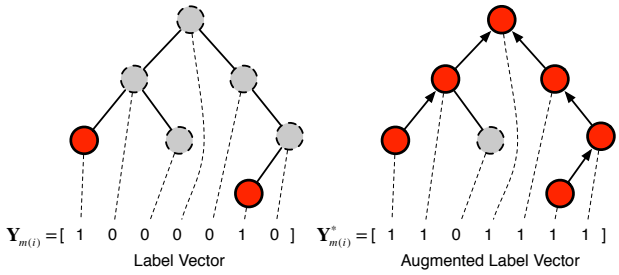

Figure 3: A label augmentation from a label vector $\mathbf{Y}_{\mathbf{m}(i)}$ to an augmented label vector $\mathbf{Y}^*_{\mathbf{m}(i)}$. By augmentation, the direct occurrence of a label in the label vector contributes not only to itself, but also to its ancestor labels in the hierarchy.

We propose to use the occurrence of a label in all augmented label vectors to represent how general information this label embodies, thus the support value for each label can be further estimated. As shown in Equation 8, by summing up the occurrence of each label in all the augmented label vectors,

$$\vec{C}_k = \sum_{m=1}^{M}\sum_{i=1}^{N}\mathbf{Y}^{k*}_{m(i)}. \qquad (8)$$

labels close to the root node in the hierarchy are frequently augmented, thus have higher occurrence values $\vec{C}_k$. Occurrence values $\vec{C}$ are used to calculate support values $\vec{S}$ for each label.

Intuitively, the higher occurrence $\vec{C}_k$ value the label k gets, the more likely that the label k is near the root node in a label hierarchy. Asserting the belongingness of a label which is in proximity to the root node to an instance is less likely to be fallible. Therefore, labels with high occurrence values makes themselves more supportive in similarity estimation.

The occurrence value of each label over the sum of occurrences of all labels in all augmented label vectors can be quantified to estimate the support value of each label. However, such estimation can be very inaccurate because although the label predictions are collected from multiple information sources, the occurrences of many labels near the leaf nodes in a hierarchy only share a very small portion over the sum of occurrences, even after label augmentation. This leads to very small values for all the labels.

In this work, we model the occurrence of a label in a label vector with a confidence interval. The occurrence of a label within all augmented label vectors can be considered as it is sampled from a subset of a population of labels. The occurrence information will be more accurate when we observe this label more frequently among all the augmented label vectors, which leads to a narrow confidence bound. Otherwise, if we rarely observe the occurrence of a label among all label vectors, the resulting confidence bound will be wide and it will be more risky to incorporate this piece of occurrence information into the support value calculation. By incorporating confidence intervals, the occurrence value itself shows how uncertain we are about the occurrence value of a label.

We use the occurrence value $\vec{C}_k$ of label k over the number of instances N as the proportion. Since the root node will be activated in all the augmented label vectors, the root node will have the highest proportion as 1. One label is either activated or not in an augmented label vector, therefore the binomial probability distribution is used. The confidence interval on this proportion can be presented in an alternate formulation that uses quantiles from the beta distribution [17]:

$$B\left(\tfrac{\alpha}{2};\vec{C}_k,N-\vec{C}_k+1\right)<\theta_k<B\left(1-\tfrac{\alpha}{2};\vec{C}_k,N-\vec{C}_k+1\right), \quad (9)$$

where B is the Beta distribution and α is the significance level, which usually has the value 0.05 (5%). θ_k is the probability of a label k being activated in an augmented label vector. $\vec{C}_k$ is the occurrence value of label k over all augmented label vectors.

By Equation 9, we can know that the less frequent a label k being sampled, the wider confidence interval it will end up with. A narrow confidence bound indicates a stronger certainty during the support value calculation. The lower bound value of the confidence interval is used to calculate the support value for each label:

$$\vec{S}_k = B\left(\frac{\alpha}{2};\vec{C}_k,N-\vec{C}_k+1\right) \quad (10)$$

Once we calculate the support value for each label with confidence, the instance similarity matrix $\mathbf{W}$ in Equation 7 can be calculated with the support label vector $\vec{S}$ we learned from the label hierarchy. Hence, $\mathbf{W}$ is called a hierarchical instance similarity matrix.

4. CONSOLIDATING HIERARCHICAL LABELS

4.1 The MHPC Algorithm

Although we provide a closed-form solution to find the global minimum for the consensus cost when the hierarchical instance similarity matrix $\mathbf{W}$ is given, it is still hard to estimate both $\hat{\mathbf{Y}}$ that associates with a minimized consensus cost $J(\hat{\mathbf{Y}})$ and the hierarchical instance similarity matrix $\mathbf{W}$ at the same time. Hence, we propose a two-phase iterative algorithm, namely the multi-source hierarchical prediction consolidation (MHPC) algorithm, that naturally decouples the computation within each iteration.

The MHPC algorithm has two phases within each iteration, namely estimating the hierarchical similarity (Section 3.2) and minimizing the consensus cost (Section 3.1). The MHPC algorithm starts at iteration t_0 with an initial estimation for hierarchical instance similarity matrix $(\mathbf{W}_{ij})_{t_0}$ and the consolidation result $\left(\hat{\mathbf{Y}}\right)_{t_0}$. $(\mathbf{W}_{ij})_{t_0}$ is initialized by the following equation:

$$(\mathbf{W}_{ij})_{t_0} = \exp\left(-\frac{1}{\sigma}\sqrt{\sum_{k=1}^{K}\vec{S}_k(\bar{\mathbf{Y}}^k_{(i)}-\bar{\mathbf{Y}}^k_{(j)})^2}\right), \quad (11)$$

where $\bar{\mathbf{Y}}^k_{(i)}$ is the simple averaging result on the k-th label of the instance i from all M sources, calculated by $= \bar{\mathbf{Y}}^k_{(i)} = \frac{1}{M}\sum_{m=1}^{M}\mathbf{Y}^k{}_{m(i)}$. Note that support values in $\vec{S}$ and the occurrence values in $\vec{C}$ are derived from all the multi-source label predictions we obtained, which we only initialize once in the entire algorithm, regardless of iterations. $(\hat{\mathbf{Y}})_{t_0}$ is initialized using the Equation 6.

Once we obtain an initial value for $(\mathbf{W}_{ij})_{t_0}$ and $(\hat{\mathbf{Y}})_{t_0}$, each iteration afterwards follows the following updating rules.

Estimating the hierarchical similarity:

$$(\mathbf{W}_{ij})_{t_{x+1}} = \exp\left(-\tfrac{1}{\sigma}\sqrt{\sum_{k=1}^{K}\vec{S}_k\left(\left(\hat{\mathbf{Y}}^k_{(i)}\right)_{t_x}-\left(\hat{\mathbf{Y}}^k_{(j)}\right)_{t_x}\right)^2}\right), \quad (12)$$

where the hierarchical instance similarity $(\mathbf{W}_{ij})_{t_{x+1}}$ is calculated by the most up-to-date consolidation results in $\left(\hat{\mathbf{Y}}\right)_{t_x}$.

Minimizing the consensus cost:

$$\left(\hat{\mathbf{Y}}\right)_{t_x+1} = \left(1+\lambda\cdot(\mathbf{L})_{t_x}\right)^{-1}\left(\hat{\mathbf{Y}}\right)_{t_x}, \quad (13)$$

where the laplacian matrix $(\mathbf{L})_{t_x}$ is calculated by the most up-to-date $(\mathbf{W})_{t_x}$ value using Equation 3. Note that in Equation 13, the consolidation result in the latest iteration $(\hat{\mathbf{Y}})_{t_x}$ is used for updating, rather than $\bar{\mathbf{Y}}$ as shown in Equation 6. With this updating function, the consolidation result can accumulate the consolidation progresses from previous iterations. Otherwise, if $(\bar{\mathbf{Y}})_{t_x}$ is used in Equation 13, we simply ignore the consolidation results from all the previous iterations, and $\mathbf{L}_{t_x}$ is the only factor we can rely on to guide the consolidation process toward a final consensus. The algorithm terminates whenever the updates on the consolidation result $(\hat{\mathbf{Y}})_{t_{x+1}}$ is no longer significant after an iteration.

5. EXPERIMENTS

In this section, we describe the yeast data sets, the real-world medical data sets and their label hierarchies respectively. Experiments on yeast data sets illustrate the ability of the proposed method in overcoming various degree of label vagueness and ambiguity from multiple information sources. While the real-world medical data set emphasizes more on the label sparsity because in real-world medical consultation, we can't ensure that all the information sources provide labels to all instances.

5.1 Data description and data preprocessing

5.1.1 Yeast Data Sets

Table 2 shows statistics about yeast data sets. Each data set annotates yeast genome from a different aspect. Each yeast genome is annotated with hierarchical-structured labels in the Functional Catalogue (FunCat) [18]. For example, a yeast genome can be associated with three functionalities:{20/01/03/01 (sugar transport), 20/03/02/02/01 (proton driven symporter), 20/09/18 (cellular import)}. The annotation scheme follows the protein functional description of each genome instance, with up to 6 levels of label taxonomy. On average, each instance has 8.8 labels.

Based on the ground truth multi-label label predictions, we introduce vague labels as well as noisy labels to the yeast data sets to model real-world cases where imperfect predictive models or inexperienced human annotators are involved in MHPC problems.

The Algorithm 2 is used to generate synthetic label predictions for each instance on all the information sources. For each label mentioned by the ground truth label matrix $\mathbf{Z}$, we generate two random values, p_V and p_N (Line 2-4). Given the label k and a function $level(k)$ indicating which level label k is in, the vagueness of label makes a label k hops to its ancestor k' with a probability $P(vagueness|level(k))$, as shown by the following Equation

$$P(vagueness|level(k)) = (\alpha_{vague})^{level(k)}, \qquad (14)$$

where α_{vague} is a parameter and $level(k)$ returns the number of connections from label k to the root node. Therefore, a label near the root node of a hierarchy has less probability to hop to its ancestor label, while a label near leaf nodes in the hierarchy is more likely to hop. Whenever $p_V > P(vagueness|level(k))$, we replace k with its ancestor label k'. Note that, with one hop performed, the label k' can be further hopped to its ancestor label k'' as well (Line 5-9). The left part of Figure 4 illustrate this idea.

Once we add the vagueness to the label, when p_N is greater than a transition probability

$$P(noise) = \alpha_{noise}, \qquad (15)$$

noise is introduced by replacing label k by one of its siblings randomly (Line 10-11). Otherwise, the label k will not be changed. The right part of Figure 4 shows the way we add noise to labels.

5.1.2 Medical Data Sets

The medical data set and the disease label hierarchy are obtained from an online medical consultation website xywy.com[1], where patients post their healthcare related questions and multiple medical professionals give online suggestions or general advice as answers.

Table 3 gives an example of the multi-source label predictions we collect on an instance. Each doctor is considered as an individual information source. Ground truth disease labels are obtained by a medical knowledge base in Baidu Baike[2](an online encyclopedia of Baidu) where registered doctors provide knowledge about certain disease names that closely associate with some symptoms. The disease label hierarchy is organized in an anatomical structure, with up to three levels of labels(e.g. disease - otorhinolaryngology - rhinitis). In terms of the label sparsity, 0.011506% of labels are activated among label predictions over all instances from all information sources. Such label low sparsity is common in data sets like this because not all doctors provide labels for every instance; not all instances get label predictions from each doctor. Usually a doctor provides around 2.5 labels to an instance on average, which leads to a low label coverage over all labels in a hierarchy.

Algorithm 2 Generating Multi-source Label Predictions

Input: A ground truth label matrix $\mathbf{Z}$
A hierarchical adjacency matrix $\mathbf{H}$
Output: label matrices $\{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_M\}$.

1: **function** GeneratePredictions($\mathbf{Y}_{m(i)}$, $\mathbf{H}$)
2: **for each** label k of instance i where $\mathbf{Z}_i^k = 1$ **do**
3: **for each** information source m **do**
4: Generate two random values $p_V, p_N \in [0, 1]$.
5: **while** k $\neq$ *ROOT* **do**
6: **if** $p_V > P(vagueness|level(k))$ **then**
7: $k \leftarrow k'$ where $\mathbf{H}_{kk'} = 1$
8: **else**
9: Break
10: **if** $p_N > P(noise)$ **then**
11: $k \in \{k' \,|\, \exists l, \mathbf{H}_{k'l} = \mathbf{H}_{kl}\}$
12: $\mathbf{Y}_{m(i)}^k \leftarrow 1$
13: **return** $\{\mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_M\}$

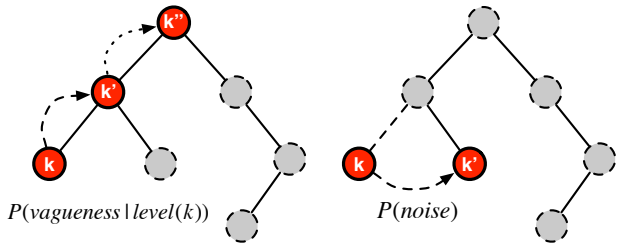

Figure 4: Introducing vague labels and noisy labels to generate the synthetic multi-source label predictions.

5.2 Experiment Settings

5.2.1 Comparison Methods

To show the advantages of the MHPC algorithm in solving multi-source hierarchical prediction consolidation problem, we compare the MHPC method with many baseline methods. Considering that no known multi-source hierarchical predic-

[1]http://club.xyxy.com

[2]http://baike.baidu.com

Data set	seq	struc	hom	cellcycle	church	derisi	gasch1	gasch2	spo	expr
#training	1701	1665	1669	1628	1630	1608	1634	1639	1600	1639
#validation	879	860	870	848	844	842	846	849	837	849

Table 2: Yeast data sets.

Instance	Doctor_id	Labels	Ground Truth
sneezing, runny nose, sleepy	52****17	{common cold, allergic rhinitis}	{common cold, allergic rhinitis, rhinitis, sinusitis, antritis}
	46****35	{common cold, rhinitis}	
	53****11	{rhinitis, sinusitis}	

Table 3: Each instance is a set of symptoms that a user describes. When an instance describes a set of symptoms, disease names are the labels we collected from different doctors. Each doctor is considered as an information source that provides disease names as labels.

tion consolidation methods are available, averaging methods as well as other model-level ensemble learning methods are introduced, which can be divided into three categories:

Averaging Methods

- SA: The simple averaging method. The SA method simply takes the average of multi-source label predictions. In [19], the authors observe that the simple averaging method is competitive with a variety of adaptive algorithms under the quadratic loss criterion.
- WA: The weighted averaging method. Besides the simple averaging method which considers an equal contribution of each label to the consolidation result, each label has a support value as a weight learned from Section 3.2.

Consensus Maximization Methods

- MLCM: The multi-label consensus maximization method is introduced in [11]. The MLCM learns a consolidation result from both label predictions and cluster predictions of the same instance from multiple information sources. By ignoring the cluster predictions which assign each instance with a cluster id, the MLCM adapts to the problem setting of MHPC. Note that, the label hierarchy is not explored in MLCM. The MLCM formulates a bipartite graph where instance nodes are on one side, label nodes from multiple information source are on the other side. The algorithm learns a subset of the connections between two partitions of nodes while maximizing the consensus among them.

Multi-source Prediction Consolidation Methods

- MPC-U: The mutli-source prediction aggregation method that minimizes the consensus cost as mentioned in Section 3.1, but uses a uniformed value for each entry of the support label vector $\vec{S}$ during instance similarity estimation.
- MHPC: The proposed method which minimizes the consensus cost by optimization and incorporates the label hierarchy in estimating hierarchical instance similarities. The support value of each label is estimated based on the lower bound confidence interval of the proportion of occurrence based on all the augmented label vectors we obtained.

5.2.2 Evaluation metrics

Ranking loss, micro-AUC and coverage error are three metrics that we used for performance evaluation.

	Minimizing Consensus Cost		Label Weights	
	Averaging	Optimizaiton	Uniform	Hierarchical
SA	✓		✓	
WA	✓			✓
MLCM		✓		
MPC-U		✓	✓	
MHPC		✓		✓

Table 4: Comparison Methods

Ranking loss averages over the instances to penalize the number of label pairs within each instance that are incorrectly ordered. Since the label space is large, it is relatively hard to assign a precise probability to each label from a large label space. Ranking labels become an alternative, sometimes a must, for the evaluation. Perfect ordered labels in instances have zero ranking losses.

AUC (Area Under the Curve) is designed for binary classification problems with skew class distributions. In hierarchical label predictions, the ground truth labels of an instance are relevant labels that covers a very small portion of the label space. The ground truth labels are dominated by other irrelevant labels. In such scenario, AUC is adopted as a metric which compares the ranks of all possible pairs of labels in terms of the relevance. Formally, the label matrix $\hat{\mathbf{Y}}$ has a total of $N \times K$ entries. Let Pos be the label set with positive (relevant) entries and Neg be the label set with all the other negative (irrelevant) entries. In hierarchical label predictions we usually have $card(Pos) \ll card(Neg)$, where $card(.)$ is the carnality of a set. Given a list of relevance scores $f()$ of all entries, micro-AUC [20] is defined as

$$\text{micro-AUC} = \sum_{i \in Pos} \sum_{j \in Neg} \frac{\mathbb{1}\left[f(i) > f(j)\right]}{\text{card}(Pos) \times \text{card}(Neg)}, \quad (16)$$

where f(i) is the relevance score for entry i and $\mathbb{1}$ is the indicator function.

Note that what micro-AUC differs from ranking loss is that micro-AUC compares the ranks of any pair of labels, whether those two labels are from the same instance or not. While the ranking loss focuses on the label ranking of individual instances. That is, the difference of ranks between labels of two different instances are not explored.

Since label predictions can be anywhere on the label hierarchy, coverage error [21] is adopted to measure the average number of labels that have to be chosen from the consolidation result so that those labels are able to cover all the ground truth labels.

5.3 Experimental results

5.3.1 Convergence Analysis

In the MHPC algorithm, the hierarchical instance similarity matrix $\mathbf{W}$ and the consolidation result $\hat{\mathbf{Y}}$ are updated by two phases, namely minimizing the consensus cost and estimating the hierarchical similarity, respectively. Two phases are performed iteratively until convergence. To show that with proper parameters learned form the validation data sets, the two-phase updating rules can lead to a convergence, we show the performance of the MHPC algorithm after each iteration, as shown in Figure 5. Note that, for each data set, the parameter λ and σ are learned by the validation set. Also, α_{vague} is fixed to 0.8 and α_{noise} is fixed to 0.5 for all data sets. Multi-source label predictions from four information sources are incorporated.

As shown in the figures, as the two-phase updating continues, three evaluation metrics are consistently ameliorated.

5.3.2 Parameter Estimation

The parameters of the MHPC method is chosen by those parameters who give the best performance of the multi-source hierarchical prediction consolidation task on the validation set. The validation set is a portion of the original data sets for parameter learning.

We compare the performance of the MHPC method on four information sources on all the yeast data sets, with α_{vauge} and α_{noise} fixed as 0.8 and 0.5. Figure 6 illustrates the impact of the value of *lamda*, as a parameter, to the overall performance of the proposed method on the validation set.

After parameter learning, $\lambda = 10$ is chosen for church, cellcyle, derisi and gasch2 data sets; $\lambda = 25$ for the expr data set; $\lambda = 15$ for gasch1, seq and spo data sets. For medical date sets, we did the same analysis and all the data sets performs the best with $\lambda = 26$.

5.3.3 Sensitivity Analysis

For experiments on yeast data sets above, two parameters (α_{vague} and α_{noise}) used to generate multi-source label predictions are set as fixed values. In this section, we further varies both α_{vague} and α_{noise} from 0.1 to 1 to test how sensitive vague and noisy labels will affect the model performance. We compare the performance of MHPC method with other alternatives on each combination of α_{vague} and α_{noise}. Due to space limitations, only the result on church, one of the yeast data set, is reported with $\lambda = 10$. Multi-source label predictions from four information sources are generated.

Figure 7 shows the performance comparisons with three evaluation metrics. We observe that the MPHC outperforms other alternatives consistently. When the vagueness level α_{noise} increases, the performance deteriorates for all the methods. But the MHPC performs relatively better than others. On the other hand, when the noise level increases and α_{vague} is fixed, WA and MPC-U methods are more easily affected by the noisy label, which leads to fluctuations of the performance surfaces. While MHPC has a relative stable performance when we varies α_{noise} from 0.1 to 1 on almost all the values α_{vauge} can take. Note that the performance of SA is similar with WA, so the performance of SA is not presented in this figure.

Information Source	Ranking Loss	Micro-AUC	Coverage Error
1	0.171	0.830	180.679
3	0.085	0.920	102.949
4	0.064	0.941	77.910
5	0.054	0.952	64.182
10	0.046	0.964	47.717
15	0.042	0.967	44.019
20	0.042	0.967	44.041
30	0.042	0.967	43.901
50	0.041	0.968	43.592

Table 5: Performance of the MHPC method with label predictions collected from a varying number of information sources.

5.3.4 Varying the Number of Information Sources

We vary the number of information sources that we collect the multi-source label predictions from. Evaluation results on all three metrics with cellecycle, one of the yeast data sets is presented in Table 5. $\alpha_{vauge} = 0.8$ and $\alpha_{vauge} = 0.5$ are used to generate multi-source label prediction and $\lambda = 10$ is used as the parameter. We vary the number of information sources from 1 to 50. From Figure 5 we can see that if we assume information sources are making errors independently, then collecting label predictions from three information sources will cut off almost 50% of the ranking loss and coverage error, when comparing with label prediction form a single information source. Moreover, as we collect information from more information sources, three evaluation metrics tend to stabilize to a final value. When adding an information source leads to an extra cost (pay a human annotator for labeling), this result gives some insights about the trade-off between the number of information sources we collect labels form, with the extra performance improvement we may get, based on the independent assumption.

5.3.5 Resolving the Label Sparsity on Medical Data Sets

The medical data sets come with multi-source label predictions inherently when multiple doctors gives suggestions to each patient. Since the medical date sets have six subsets (MED.1 to MED.6), we randomly sampled 500 instances from each data set. The label predictions associate with those instances are used for prediction consolidation. Table 6 shows the performance of MHPC with other baseline methods, for which we can see the superior performance of MHPC method in real-world scenarios.

6. CONCLUSIONS

As information explodes, we are able to obtain an increasing number of label predictions from a large population of information sources at the same time. Due to privacy concerns or storage limitations, the raw features of instances are usually discarded or withheld. The labels we collect from multiple information sources bring not only diversity of labels, but also vagueness and noises. In this work, we studied the multi-source hierarchical prediction consolidation (MHPC) problem. Traditional model-level ensemble learning problems deal with multi-source information but they simply ignore the hierarchical structure of labels. On the other hand, hierarchical multi-label learning problems try to bring the label hierarchy into varies tasks such as

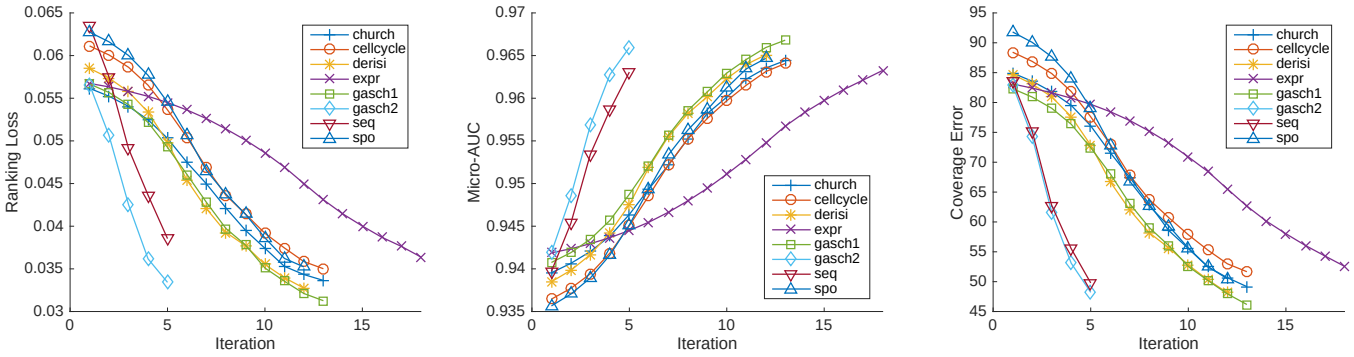

Figure 5: Ranking loss, mirco-AUC and coverage error on all the yeast data sets as the updatings are conducted iteratively in MHPC.

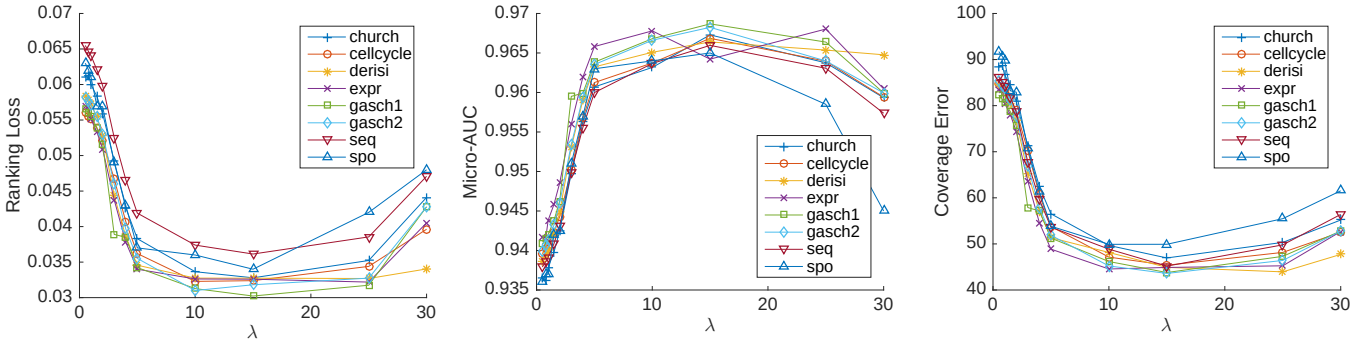

Figure 6: Parameter estimation for λ on yeast data sets.

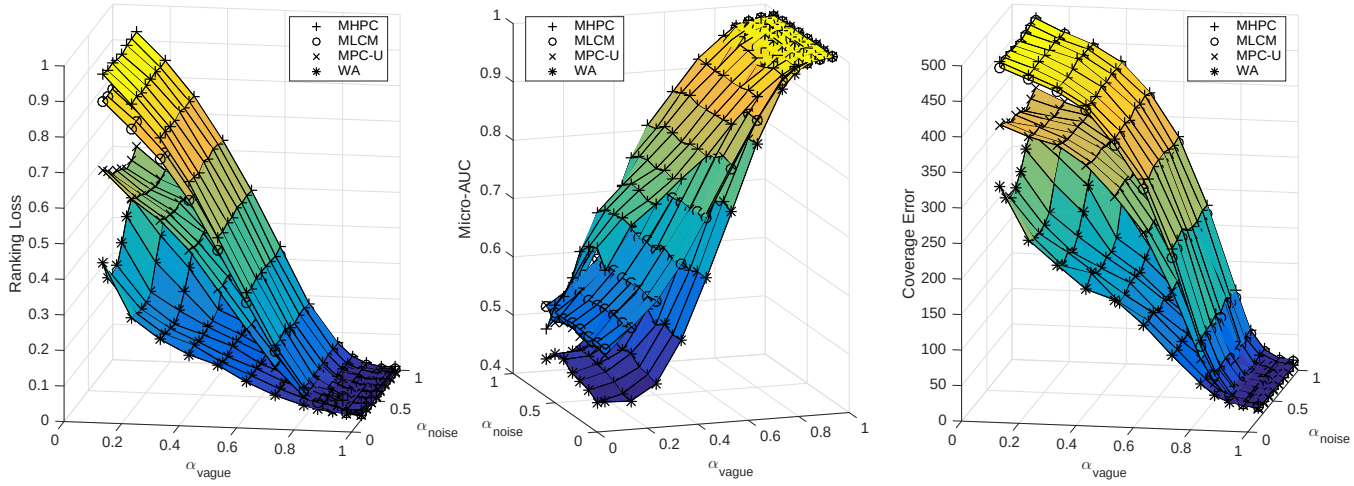

Figure 7: Performance with varying degrees of vagueness and noises on church, one of the yeast data sets.

classification. The MHPC problem effectively incorporates the existing label hierarchy to resolve vagueness and noise originate from multiple information sources, where very few works have been done. We formulate the MHPC problem as an optimization task with a closed-form solution. Two phases, namely minimizing the consensus cost and estimating the hierarchical similarity, are performed in an iterative fashion to learn a consolidation result while preserving the structures of the label hierarchy. Experiments conducted on both synthetic and real-world data sets show the advantages of the proposed method over other alternatives.

7. REFERENCES

[1] Yu Cheng, Kunpeng Zhang, Yusheng Xie, Ankit Agrawal, and Alok Choudhary. On active learning in hierarchical classification. In *CIKM*, pages 2467–2470, 2012.

[2] J Ross Quinlan. Bagging, boosting, and c4. 5. In *AAAI/IAAI, Vol. 1*, pages 725–730, 1996.

[3] Guoxian Yu, Carlotta Domeniconi, Huzefa Rangwala, Guoji Zhang, and Zhiwen Yu. Transductive multi-label ensemble classification for protein function prediction. In *KDD*, pages 1077–1085, 2012.

Datasets	Methods	Evaluation Metrics		
		Ranking Loss	Micro-AUC	Coverage Error
MED.1	SA	0.520 (5)	0.620 (5)	213.175 (5)
	WA	0.518 (4)	0.621 (3)	212.844 (4)
	MPC-U	0.304 (3)	0.547 (5)	125.108 (3)
	MLCM	0.262 (2)	0.643 (2)	108.146 (1)
	MHPC	0.196 (1)	0.754 (1)	125.107 (2)
MED.2	SA	0.358 (5)	0.603 (4)	75.544 (5)
	WA	0.357 (4)	0.604 (3)	75.416 (4)
	MPC-U	0.200 (2)	0.539 (5)	42.630 (3)
	MLCM	0.268 (3)	0.680 (2)	31.221 (1)
	MHPC	0.166 (1)	0.715 (1)	35.336 (2)
MED.3	SA	0.067 (4)	0.706 (2)	3.400 (4)
	WA	0.065 (3)	0.704 (3)	3.314 (3)
	MPC-U	0.069 (5)	0.432 (5)	3.543 (5)
	MLCM	0.064 (2)	0.542 (4)	3.288 (2)
	MHPC	0.038 (1)	0.847 (1)	2.000 (1)
MED.4	SA	0.301 (5)	0.601 (4)	38.325 (5)
	WA	0.300 (4)	0.602 (3)	38.275 (4)
	MPC-U	0.173 (2)	0.460 (5)	22.242 (3)
	MLCM	0.225 (3)	0.617 (2)	19.325 (2)
	MHPC	0.118 (1)	0.707 (1)	15.233 (1)
MED.5	SA	0.355 (5)	0.563 (4)	67.534 (5)
	WA	0.353 (4)	0.564 (3)	67.170 (4)
	MPC-U	0.189 (1)	0.549 (5)	36.080 (2)
	MLCM	0.276 (3)	0.565 (2)	52.523 (3)
	MHPC	0.232 (2)	0.572 (1)	32.648 (1)
MED.7	SA	0.365 (5)	0.598 (3)	47.175 (5)
	WA	0.363 (4)	0.599 (2)	46.991 (4)
	MPC-U	0.236 (3)	0.491 (5)	30.632 (3)
	MLCM	0.190 (1)	0.632 (1)	24.798 (2)
	MHPC	0.203 (2)	0.594 (4)	22.633 (1)

Table 6: Performance on medical data sets.

[4] Chi-Hoon Lee. Learning to combine discriminative classifiers: confidence based. In *KDD*, pages 743–752, 2010.

[5] Lijuan Cai and Thomas Hofmann. Hierarchical document categorization with support vector machines. In *CIKM*, pages 78–87, 2004.

[6] Suhang Wang, Jiliang Tang, Yilin Wang, and Huan Liu. Exploring implicit hierarchical structures for recommender systems. In *IJCAI*, 2015.

[7] Sheng Wang, Hyunghoon Cho, ChengXiang Zhai, Bonnie Berger, and Jian Peng. Exploiting ontology graph for predicting sparsely annotated gene function. *Bioinformatics*, 31(12):i357–i364, 2015.

[8] Wei Bi and James T Kwok. Mandatory leaf node prediction in hierarchical multilabel classification. In *NIPS*, pages 153–161, 2012.

[9] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

[10] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *NIPS*, pages 585–593. 2009.

[11] Sihong Xie, Xiangnan Kong, Jing Gao, Wei Fan, and P.S. Yu. Multilabel consensus classification. In *ICDM*, pages 1241–1246, 2013.

[12] Bowen Dong, Sihong Xie, Jing Gao, Wei Fan, and Philip S. Yu. Onlinecm: Real-time consensus maximization with missing values. In *SDM*, 2014.

[13] Wei Bi and James T Kwok. Multi-label classification on tree-and dag-structured hierarchies. In *ICML*, pages 17–24, 2011.

[14] Bojan Mohar, Y Alavi, G Chartrand, and OR Oellermann. The laplacian spectrum of graphs. *Graph theory, combinatorics, and applications*, 2(871-898):12, 1991.

[15] A Ravindran, Gintaras Victor Reklaitis, and Kenneth Martin Ragsdell. *Engineering optimization: methods and applications*. John Wiley & Sons, 2006.

[16] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[17] Lawrence D Brown, T Tony Cai, and Anirban DasGupta. Interval estimation for a binomial proportion. *Statistical science*, pages 101–117, 2001.

[18] Andreas Ruepp, Alfred Zollner, Dieter Maier, Kaj Albermann, Jean Hani, Martin Mokrejs, Igor Tetko, Ulrich Güldener, Gertrud Mannhaupt, Martin Münsterkötter, et al. The funcat, a functional annotation scheme for systematic classification of proteins from whole genomes. *Nucleic acids research*, 32(18):5539–5545, 2004.

[19] Varsha Dani, Omid Madani, David Pennock, and Sumit Sanghai. An empirical comparison of algorithms for aggregating expert predictions. In *UAI*, 2006.

[20] Corinna Cortes and Mehryar Mohri. Auc optimization vs. error rate minimization. *NIPS*, 16(16):313–320, 2004.

[21] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.