Refining Image Categorization by Exploiting Web Images and General Corpus

Yazhou Yao, Student Member, IEEE, Jian Zhang, Senior Member, IEEE, Fumin Shen, Member, IEEE, Xiansheng Hua, Fellow, IEEE, Wankou Yang, Member, IEEE, and Zhenmin Tang

Abstract—Studies show that refining real-world categories into semantic subcategories contributes to better image modeling and classification. Previous image sub-categorization work relying on labeled images and WordNet’s hierarchy is not only labor-intensive, but also restricted to classify images into NOUN subcategories. To tackle these problems, in this work, we exploit general corpus information to automatically select and subsequently classify web images into semantic rich (sub-)categories. The following two major challenges are well studied: 1) noise in the labels of subcategories derived from the general corpus; 2) noise in the labels of images retrieved from the web. Specifically, we first obtain the semantic refinement subcategories from the text perspective and remove the noise by the relevance-based approach. To suppress the search error induced noisy images, we then formulate image selection and classifier learning as a multi-class multi-instance learning problem and propose to solve the employed problem by the cutting-plane algorithm. The experiments show significant performance gains by using the generated data of our way on both image categorization and sub-categorization tasks. The proposed approach also consistently outperforms existing weakly supervised and web-supervised approaches.

Index Terms—General corpus information, image categorization, sub-categorization, web-supervised

I. INTRODUCTION

Image categorization has achieved a great progress in the past few years, but it still needs a massive amount of manually labeled data [2], [3], [4], [5], [6]. Meanwhile, image sub-categorization has been used to improve performance in a wide variety of vision tasks. For example, object detection [12], animal behaviour analysis [18] and image classification [1]. Subdividing categories into subcategories multiplies the number of labels, aggravating the annotation problem.

With the development of Internet, the number of digital images is growing extremely rapidly. How to effectively categorize these images has become an increasingly serious problem. Further, previously defined NOUN categories usually unable to have a good description for these emerging images, which have a variable appearance, positions, and poses [31], [45]. The categories need to be divided into more semantic rich subcategories to cover the finer semantic differences. ImageNet [19] is an image dataset organized according to the WordNet [44] hierarchy. It provides the research community not only with thousands of categories and millions of images, but also with refinement labels in a hierarchy. However, the process of constructing ImageNet is both time-consuming and labor-intensive. For example, it has taken several years to construct the ImageNet. In addition, the ImageNet requires the pre-existing expert knowledge WordNet and it only contains the NOUN subcategories (e.g., category “airplane”, “dog” and “bird” in Fig. 1), it does not contain the VERB subcategories (e.g., category “horse” in Fig. 1).

To reduce the cost of manual annotation, automatic methods by exploiting web images for image categorization [9], [21], [52] have attracted more and more people’s attention. Fergus et al. [33] took the probabilistic latent semantic analysis (pLSA) technique to automatically select and learn object categories from web images. Hua et al. [21] proposed the use of a clustering based method to filter “group” noisy images and a propagation-based method to filter individual noisy images. Niu et al. [52] proposed to set each latent domain as a “bag” and the images therein as “instances”, then image selection and classifier learning are formulated as a multi-instance learning problem. The advantage of these methods is that the need for manual intervention is eliminated. Unlike these studies, our proposed work simultaneously addresses the issues of image categorization and sub-categorization by leveraging the general corpus and web images.
There are also few previous works [1], [31], [45], [48] dealing with the image sub-categorization problem. Mansur et al. [1] proposed using probabilistic Latent Semantic Analysis (pLSA) to find subcategories and these subcategories are based on the similarity of images. In [31], both of positive and negative images are used to learn subcategories. Particularly, a new model by joint clustering and classification was proposed for discriminative sub-categorization. Wang et al. [48] designed a formulation for dictionary learning (subcategories) by maximizing classification margins MIL. However, as some of the previous work [8], methods [1], [31], [48] still cannot assign semantic refinement labels for the newly discovered subcategories. Ristin et al. [45] adopt the framework of Random Forests and proposed a regularized objective function that takes into account relations between categories and subcategories to improve the classification of subcategories. Unlike previous works, method [45] can classify images into subcategories, but only NOUN subcategories. It is not comprehensive enough for describing the refinement images (e.g., category “horse” in Fig. 1). In our work, we exploit general corpus information and web images for image categorization and sub-categorization. Our proposed approach can not only classify images into NOUN subcategories, but also into VERB, ADJECTIVE and ADVERB subcategories.

Motivated by the situation described above, we propose a novel automatically web-supervised image categorization and sub-categorization framework. In our work, we mainly consider the following two important issues: 1) the labels of subcategories derived from the general corpus usually have noise, how can we select useful labels of subcategories from these noisy labels; 2) the retrieved web images are often associated with inaccurate labels, so the learnt classifiers may be less robust, and the classification performance may be significantly degraded as well, how can we select useful images and learn domain robust classifiers from these noisy web training images.

To find the labels of semantic refinement subcategories, we search the given categories in Google Books Ngram Corpus (GBNC) [16] with Parts-Of-Speech (POS), specifically with NOUN, VERB, ADJECTIVE and ADVERB. Further, as the labels of subcategories derived from the general corpus tend to have noise, we apply a relevance-based approach for removing noise and selecting the useful labels of subcategories. Finally, to cope with label noise of web training images, we treat each selected subcategory as a “bag” and the images therein as “instances”. In specific, we propose a new multi-class MIL formulation to select the images from each bag and learn the classifiers for the categories and subcategories. Our aim is to select a subset of images from each bag to represent this bag, such that the training bags from all the categories can be well separated. To verify the superiority of our proposed approach, we conducted experiments on both image categorization and sub-categorization tasks. The experimental results demonstrated the superiority of our proposed approach.

The main contributions of this work are summarized as follows:

1) Compared to existing methods, our proposed framework can not only classify images into NOUN subcategories, but also into VERB, ADVERB and ADJECTIVE subcategories. Our proposed framework has a better semantic refinement descriptions for the categories.

2) To suppress the search error and noisy subcategories (which are not filtered out) induced noisy images, we formulate image selection and classifier learning as a multi-class multi-instance learning problem and propose to solve the employed problem by the cutting-plane algorithm.

3) We propose a new unified objective function to jointly learn the classifiers for categories and subcategories. Our proposed formulation can not only consider the relationship between category and its subcategories, but also consider the relationship between different categories. Thus, the classifiers in our work have a better domain adaptation ability.

The rest of the paper is organized as follows. In Section II, a brief discussion of related works is given. Section III elaborates the proposed framework with the optimization algorithm. The experimental evaluations and discussions are presented in Section IV. Lastly, the conclusion and future research directions are offered in Section V.

II. RELATED WORK

Due to the emergence of ImageNet, deep convolutional neural networks (CNN) have achieved a great success in image categorization. However, deep CNNs are computationally intensive and require a large number of labeled data. Simpler classifiers like support vector machine (SVM) [50] and nearest class mean classifiers (NCM) [46] provide us with another alternative which have much shorter running time and acceptable classification accuracy. To reduce the cost of manual annotation, some of the previous works also concentrated on the task of “cleaning up” web images for training data collection. For example, Fergus et al. [28] proposed the use of visual classifiers learned from Google image search engine to re-rank the images based on the visual consistency. Subsequent methods [21], [25], [20] have employed similar removing mechanisms to automatically construct clean image datasets for training classifiers. In our work, we focus on another fundamental, yet often ignored, aspect of the problem: we argue that the current poor performance of classification models learned from the web is due to the selected images which may have different distributions with the test images.

Our work is also related to the recent works for latent domains discovering methods. In [32], Hoffman et al. proposed using a hierarchical clustering technique to find the feasible latent domains while [30] adopt maximum distinctiveness and maximum learnability for different latent domains separation. Xiong et al. [22] proposed a squared-loss mutual information based clustering model with category distribution priority in each domain to infer the domain assignment for images. The difference between our work and [32], [30], [22] is methods [32], [30], [22] cannot assign the semantic refinement labels to newly discovered latent domains. All the discovered latent domains still only have the label of the coarse category. In contrast, our work aims to classify images into categories and
Our work is more related to the recent image sub-categorization works [31], [45], [48], which assume several subcategories exist in each category. However, the subcategories discovered by [31], [48] still only have the label of the category. Refinement labels of subcategories are still unavailable. Ristin et al. [45] adopt the framework of Random Forests and proposed a regularized objective function that takes into account relations between categories and subcategories to improve the classification accuracy. Unlike previous works, method [45] can classify images into subcategories, but only the NOUN subcategories. The reason is that method [45] relies on expert knowledge WordNet to obtain the semantic refinement subcategories. In our work, we eliminate the dependency on expert knowledge and propose to exploit general corpus information to obtain the semantic refinement subcategories. The advantage of our proposed approach is that our method can not only classify images into NOUN subcategories, but also into VERB, ADJECTIVE and ADVERB subcategories.

As our approach relies on weakly labeled web images, it is loosely related to the multi-instance learning works [7], [10], [40], [41]. Method [10] and [40] proposed to partition the weakly labeled web images into a set of clusters and each cluster is treated as a “bag”, the images therein as “instances”. Correspondingly, different multi-instance learning methods were proposed in [10] and [40]. In [7], Andrews et al. adopt a heuristic way to iteratively train the image classifier and then infer the category labels of these instance images. Method [41] proposed two convex optimization methods which maximize the margin of concepts via key instance generation at the instance-level and bag-level for locating the regions of interest (ROI) in the images automatically.

Our work is largely inspired by the following work. A weakly supervised domain robust visual recognition system was recently proposed in [52] and achieved impressive performance for video event recognition. Niu et al. [52] first applied the latent domain discovering method [30] to find all the latent domains from the training data. Then the multi-instance learning method was leveraged to cope with noise in the labels of web training images. The main difference from ours is the formation process of the “bags”. Method [52] takes the latent domains as “bags” while our method applies the selected subcategories. Compared to [52], our selected “bags” which derived from the general corpus have strong supervisory information. This supervisory information can help us to maximize the inter-class variation and simultaneously minimize the intra-class variation. In addition, as some of the previous works [40], [41], only category label can be assigned to images in [52] while our method can not only assign the category label, but also the refinement labels of subcategories.

### III. WEB-SUPERVISED IMAGE CATEGORIZATION AND SUB-CATEGORIZATION

We seek to automate the process of classifying images into categories and subcategories by exploiting general corpus information and web images. As shown in Fig. 2, there is a backend subsystem (classifier building) and a frontend subsystem (categories and subcategories classifying). For the backend system, the input is a category label that we would...
like to build a classification model for. Then a set of semantically rich subcategories is obtained by searching in GBNC, from which the noisy subcategories are removed. After obtaining the candidate images by retrieving the labels of selected subcategories with image search engine, we treat each selected subcategory as a “bag” and the images therein as “instances”. Particularly, we formulate image selecting and classifier learning as a multi-class MIL problem by selecting a subset of images from each bag to learn the classifiers. The final outputs of the backend system are (1) a classifier (e.g., “dog”) representing the category; (2) a set of classifiers (e.g., “golden dog”, “fluffy dog” and “Eskimo dog”) corresponding to subcategories. The input of frontend is a set of images which will be given two labels including category and semantic refinement subcategories.

A. Subcategories Discovering

Inspired by recent works [16], [20], [51], we can use Google Books Ngram English 2012 Corpus to discover the labels of semantic refinement subcategories for modifying the given category. Our motivation is to find not only the semantically rich NOUN subcategories, but also VERB, ADJECTIVE and ADVERB subcategories. Compared to the expert knowledge WordNet and ConceptNet which only have NOUN subcategories, ngram data is much more general and exhaustive. Following [26] (see section 4.3), we specifically use the dependency gram data with parts-of-speech (POS) for refinement subcategories discovering. For example, given a parent category and its corresponding POS tag (e.g., ‘jumping, VERB’), we find all its occurrences annotated with POS tag within the dependency gram data. Of all the ngram dependencies retrieved for the given category, we choose those whose modifiers are tagged as NOUN, VERB, ADJECTIVE and ADVERB as the candidate subcategories. We utilize these semantic refinement subcategories (corresponding images) to reflect the different visual distributions of the category. The detailed subcategories discovered in this step can be found on website https://1drv.ms/l/s!AhpqSQTtg8NxyjGslE2kJGcvTV.

B. Noisy Subcategories Removing

Not all the discovered subcategories are useful, some noise may also be included (e.g., the bold subcategories in Table 1). Using these noisy subcategories to retrieve images for the category will have a negative effect on the accuracy and robustness of the classifier. To this end, we first remove these noisy subcategories before we select images and train classifiers for the category and subcategories. We retrieve the top K images from image search engine for each candidate subcategory to represent their visual distributions. By analyzing the text semantics and visual distributions presented by these subcategories, we choose the following features to separate the useful subcategories from noise.

1) Feature selecting: From the visual relevance perspective, we want to eliminate visually less relevant subcategories (e.g., “wood horse”, “paper boat”). The intuition is that relevant subcategories should have a relatively small inter-visual distance to its parent category and other relevant subcategories.

We denote each image as $x_i$ and the compound feature $\phi_k = \frac{1}{K} \sum_{i=1}^{K} x_i$ of K images in each subcategory to represent visual distribution of this subcategory. Suppose a parent category $C_i$ has N subcategories, then we will have $N \times (N-1)/2$ inter-visual distances between subcategories. We calculate the minimum, maximum, average and standard deviation of the inter-visual distances between subcategories. Besides, we also calculate the inter-visual distance between subcategory and its parent category. Particularly, we denote these inter-visual distances by $D = \{d_n, 0 \leq n < \frac{N \times (N-1)}{2}\} + N$ and normalize these distances to a number in [0,1] by:

$$d'_n = \frac{d_n - \min\{d_n\}}{\max\{d_n\} - \min\{d_n\}}. \quad (1)$$

From the visual consistency perspective, we want to keep visually salient and eliminate non-salient subcategories (e.g., “missing cat”, “betting horse”). The intuition is that visually salient subcategories have small intra-visual distance and exhibit predictable visual patterns. For the K images in each subcategory, we calculate the $K \times (K-1)/2$ intra-visual distance among these K images. We obtain the minimal, maximal, average and standard deviation of intra-visual distance like how we generate the inter-visual distance. Similarly, we normalize this intra-visual distances.

From the semantic relevance perspective, we want to remove semantically less relevant subcategories (e.g., “tang horse”, “metal dog”). The intuition is that relevant subcategories tend to have a relatively small semantic distance to the parent category. Normalized Google Distance (NGD) constructs a method to extract semantic similarity distance from the World Wide Web (WWW) using Google page counts [13]. For a search term $x$ and search term $y$, NGD is defined by:

$$NGD(x, y) = \frac{\max\{\log f(x), \log f(y)\} - \log f(x, y)}{\log N - \min\{\log f(x), \log f(y)\}} \quad (2)$$

---

**TABLE I: Examples of the candidate subcategories discovered by our approach.**

| Category | Discovered subcategories |
|----------|--------------------------|
| Horse    | \{jumping horse, grazing horse, rearing horse\} |
|          | \{plough horse, hunter horse, black horse\} |
|          | \{wood horse, tang horse, betting horse, sea horse\} |
| Boat     | \{sails boat, fishing boat, diving boat\} |
|          | \{ski boat, tuna boat, bass boat\} |
|          | \{leather boat, crystal boat, butter boat, paper boat\} |
| Dog      | \{farm dog, wolf dog, fighting dog\} |
|          | \{pekingese dog, newfoundland dog, golden dog\} |
|          | \{hot dog, cheese dog, metal dog, van dog\} |
| Train    | \{subway train, metra train, electric train\} |
|          | \{light train, double train, trolley train\} |
|          | \{potty train, storm train, column train\} |
| Bird     | \{swallow bird, seagull bird, black bird\} |
|          | \{swan bird, eagle bird, humming bird\} |
|          | \{soup bird, angry bird, magic bird, bird nest\} |
| Cat      | \{tiger cat, brown cat, hissing cat\} |
|          | \{fat cat, desert cat, ginger cat\} |
|          | \{lucky cat, tom cat, cat machine, missing cat\} |
where $f(x)$ denotes the number of pages containing $x$, $f(x, y)$ denotes the number of pages containing both $x$ and $y$ and $N$ is the total number of web pages searched by Google.

In general, we derive the following features to separate the useful subcategories from noise:

- Normalized Google Distance between subcategory and its parent category
- Normalized visual distance between subcategory and its parent category
- Normalized minimum, maximum, average and standard deviation of the inter-visual distance between subcategory and other subcategories
- Normalized minimum, maximum, average and standard deviation of the intra-visual distance between different images in subcategory

2) Classifier learning: After deriving the selected features, we train a classifier to determine whether or not a subcategory should be selected. To train this classifier, we label a set of subcategories and this labeling work only needs to be done once for all categories. This classifier can be, for example, Support Vector Machine (SVM) based (which is also the one we used in our paper), Decision Tree based, etc. Although SVM is not the prevailing state-of-the-art method for classification, we find our approach to be effective in pruning noisy subcategories.

C. Multi-class MIL Learning

Although the Google image search engine has ranked the returned images, some noisy images may still be included. In addition, a few noisy subcategories which are not filtered out may also induce some noise. To this end, we propose our web-supervised multi-class MIL model for noisy images removing and domain robust classifiers learning.

For ease of presentation, we denote each instance as $x_i$ with its label $y_i$ and each bag $G_m$ with the label $Y_m$. A matrix/vector is denoted by a uppercase/lowercase letter in boldface and the element-wise product between two matrices is represented by $\odot$. We also define the identity matrix as $I$ and $0, 1 \in \mathbb{R}^n$ denote the column vectors of all zeros and ones, respectively. The transpose of a vector or matrix is represented by $\top$. The inequality $u = [u_1, u_2 \ldots u_n]^\top \geq 0$ means that $u_i \geq 0$ for $i = 1, \ldots, n$. The indicator function is represented as $\delta(i = j)$, where $\delta(i = j) = 1$ if $i = j$, and $\delta(i = j) = 0$, otherwise.

1) Formulation: Since the retrieved web images may contain noise, we need to remove noise and select appropriate samples to train robust classifiers. To this end, a binary indicator $h_i \in \{0, 1\}$ is used to indicate whether or not training instance $x_i$ is selected. To be exact, $h_i = 1$ when $x_i$ is selected, and $h_i = 0$ otherwise. Since the precision of images returned from the Google image search engine tends to have a relatively high accuracy, we define each positive bag as at least having a portion of $\eta$ positive instances. The value of $\eta$ can be estimated from some prior knowledge [40, 51]. We define $h = [h_1, \ldots, h_N]^\top$ as the indicator vector, and use $H = \{h| \sum_{i \in I_m} h_i = \eta |G_m|, \forall m\}$ to represent the feasible set of $h$, where $I_m$ represents the set of instance indices in $G_m$, and $|G_m|$ denotes the cardinality of $G_m$. We assume there are $N$ retrieved web images coming from $C$ categories and $S$ subcategories. $z_{i,s} \in \{0, 1\}$ is a binary indicator variable and takes the value of 1 when $x_i$ belongs to the $s$-th subcategory, and 0 otherwise. We denote $N_s = \sum_{i=1}^N z_{i,s}$ as the number of web training images from the $s$-th subcategories. Based on MIL [40], we propose our multi-class MIL formulation as follows:

$$\min_h, w_{c,s}, \xi_m \quad \frac{1}{2} \sum_{c=1}^C \sum_{s=1}^S \|w_{c,s}\|^2 + C \sum_{m=1}^M \xi_m$$

s.t. $\frac{1}{|G_m|} \sum_{i \in I_m} h_i \left( \sum_{s=1}^S P_{i,s}(w_{Y_m,s})^\top \phi(x_i) - \left(\begin{array}{c}w_{c,s} \otimes \lambda_i \end{array}\right)^\top \phi(x_i) \right) \geq \eta - \xi_m, \forall m, s, c \neq Y_m$,

$$\xi_m \geq 0, \forall m$$

where $C$ is a tradeoff parameter, $\xi_m$ are slack variables and $\phi(\cdot)$ is the feature mapping function. $P_{i,s}$ is the probability that the $i$-th training sample comes from the $s$-th subcategories. It can be obtained by calculate $P_{i,s} = (z_{i,s}/N_s)/\sum_{s=1}^S(z_{i,s}/N_s)$. The explanation for constraint (4) is that we force the total decision value (obtained by using the classifier for its own category) to be larger than those obtained by using the classifier for other categories. The motivation is we want to reduce the bag-level loss by removing the noise and identifying the good instances within training bags.

2) Solution: Problem (3) is a non-convex mixed integer problem and is hard to solve directly. Inspired by recent works [40, 41, 43], we can relax the dual form of (3) as a multiple kernel learning (MKL) problem which is much easier to solve. The derivations of (3) to its below dual form is provided in the Appendix A:

$$\min_h \alpha \quad \frac{1}{2} \alpha^\top Q^h \alpha + \zeta^\top \alpha$$

s.t. $\sum_{\alpha_{m,c,s}} = C_1, \forall m$,

$$\sum_{\alpha_{m,c,s}} \geq 0, \forall m, c, s.$$
When we set the base kernel as $Q^h$, the above problem is similar to the MKL dual form and we are able to solve it on its primal form, which is a convex optimization problem:

$$
\min_{d \in D, w_t, \xi_m} \frac{1}{2} \sum_{t=1}^{T} \left\| w_t \right\|_2^2 + C_t \sum_{m=1}^{M} \xi_m
$$

subject to

$$\sum_{t=1}^{T} w_t^T \varphi(h_t, G_m, c, s) \geq \xi_{m,c,s} - \xi_m, \forall m, c, s
$$

where $\varphi(h_t, G_m, c, s)$ is the feature mapping function induced by $Q^h$. The derivations of (6) is the dual form of (7) are provided in the Appendix B. In the next, we will give a solution to (7).

We solve the convex problem in (7) by updating $d$ and $\{w_t, \xi_m\}$ in an alternative way.

- **Update $d$:** We firstly fix $\{w_t, \xi_m\}$ to solve $d$. By introducing a dual variable $\beta$ for constraint $d^T 1 = 1$, the Lagrangian form of (7) can be derived as:

$$
\mathcal{L} = \frac{1}{2} \sum_{t=1}^{T} \left\| w_t \right\|_2^2 + C_t \sum_{m=1}^{M} \xi_m - \sum_{m,c,s} \alpha_{m,c,s} (\sum_{t=1}^{T} w_t^T \varphi(h_t, G_m, c, s) - \xi_{m,c,s} + \xi_m) + \beta(d - 1).
$$

Through the derivative of (9) with respect to $d_t$ as zero, we can get:

$$
d_t = \frac{\left\| w_t \right\|}{\sqrt{2\beta}}, \forall t = 1, ..., T.
$$

For parameter $\beta$, $\left\| w_t \right\| / \sqrt{2\beta}$ is monotonically decreasing. In addition, parameter $d_t$ satisfies $\sum_{t=1}^{T} d_t = 1$. Therefore, we can use binary search method to solve $\beta$ and recover $d_t$ according to (10).

- **Update $w_t$:** When $d$ is fixed, $w_t$ can be obtained by solving $\alpha$ in (9). Problem (9) is a quadratic programming problem w.r.t $\alpha$. Since there are $M \cdot C \cdot S$ variables in our problem, it is time-consuming to employ the existing quadratic programming solvers. Inspired by recent works [23, 12], we apply the cutting-plane algorithm [37] to solve this quadratic programming problem.

We start from a small number of base kernels and at each iteration we add a new violating base kernel. Therefore, only a small set of $h$ need to be solved at each iteration and the whole problem can be optimized more effectively. By setting the derivatives of (9) with respect to $\{w_t, \xi_t, d_t\}$ as zeros, (9) can be rewritten as:

$$
\max_{\beta, \alpha \in \mathcal{V}} -\beta + \zeta^T \alpha
$$

subject to

$$\alpha \mathcal{T} Q^h \alpha \leq \beta, \forall t.
$$

We solve (11) by solving $\alpha$ with only one constraint at the first, then add a new violating constraint iteratively. Particularly, since each constraint is associated with an $h_t$, we can obtain the most violated constraint by optimizing:

$$
\max_{h} \frac{1}{2} \alpha^T Q^h \alpha
$$

Algorithm 1 Cutting-plane algorithm for solving the proposed web-supervised multi-class MIL model.

**Input:**
- Auto-labelled image bags $\{(G_m, Y_m)\}_{m=1}^M$, initialize $y_i = 1$ for all $x_i$ in selected bags $G_m$.
1: Set $t = 1$ and $C = \{h_1\}$;
2: **Repeat**
3: \( t = t + 1; \)
4: Compute MKL to solve $d$ and $\alpha$ in (6) based on $C$;
5: //Find the most violating $h_t$
6: for each bag $G_m$
7: \( \text{Fix the labelling of instances in all other bags}; \)
8: Enumerate the candidates of $y_i$ in $G_m$;
9: Find the optimal $y_m$ by maximizing (13);
10: **end**
11: repeat lines 6-10 until there is no change in $h$;
12: Add the most violating $h_t$ to the violation set $C = C \cup h_t$;
13: **Until** The objective of (6) converges.

**Output:**
- The learnt image classifier $f(x)$.

After a simple derivation, we can rewrite (12) as:

$$
\max_{h} h^T (\frac{1}{2} Q \odot (\alpha \alpha^T)) h
$$

(13)

where $\alpha_i = \frac{1}{|G_m|} \sum_{c,s} \alpha_{m,c,s}$ for $i \in I_m$ and $Q = \sum_{c,\ell,\nu} \phi(x_i) \lambda(i, j, c, s)$. Problem (13) can be solved approximately through the binary indicator vector $h$ in a bag by bag fashion iteratively to maximize (13) until there is no change in $h$. The detailed solutions of cut-plane algorithm for our web-supervised multi-class multi-instance learning model are described in Algorithm 1.

Since the visual distributions of the training samples from same category or subcategory are generally more similar than different categories and subcategories, we train one classifier for each category and each subcategory. In general, a total of $C \times S$ classifiers $f_c(x)\{c = 1, ..., C; s = 1, ..., S\}$ will be learned. $f_{c,s}(x) = (w_{c,s})^T \varphi(x)$ (for better representation, we omit the bias term here) represents the classifier of the $s$-th subcategory and the $c$-th category. The decision function for category $C$ is obtained by integrating the learned classifiers from multiple subcategories: $f_c(x_i) = \sum_{s=1}^{S} P_i f_{c,s}(x_i)$.

Given a testing image $x$, we want to find the labels of the most matched subcategory and category, whose classifier achieves the largest decision value from all the subcategories and categories respectively. Thus, the subcategory label of image $x$ can be predicted by:

$$
\arg\max_{s} w_{c,s}^T \varphi(x)
$$

(14)

and the category label by:

$$
\arg\max_{c} (\max_{s} w_{c,s}^T \varphi(x)).
$$

(15)

In summary, to suppress the search error induced noisy images, we propose a multi-class MIL model to select a subset of training images from selected bags and simultaneously learning the optimal classifiers based on this selected images.
IV. EXPERIMENTS

In this section, we first conduct experiments on both image categorization and sub-categorization to demonstrate the superiority of our proposed approach. Then we quantitatively analyse the role of different steps contributing to the final results. In addition, we also analyse the parameter sensitivity and time complexity of our proposed approach in this section.

A. Image Categorization

The goal of this experiment is to compare the image categorization ability of our proposed approach with other related works.

1) Experimental setting: We follow the setting in [9] and exploit web images as the training set, human-labelled images as the testing set. Particularly, we evaluate the performance on the following dataset:

- PASCAL VOC 2007 [23]. The PASCAL VOC 2007 dataset contains 9963 images in 20 categories. Each category has training/validation data and test data. For this experiment, we only use the test data in PASCAL VOC 2007 as the benchmark testing set. The detailed number of images for each category in this experiment is shown in Table I.
- STL-10 [14]. The STL-10 dataset has ten categories, and each category of which contains 500 training images and 800 test images. All of the images in STL-10 are color 96 × 96 pixels. We also use the test images in STL-10 as the benchmark testing set.
- CIFAR-10 [39]. The CIFAR-10 dataset consists of 60000 32×32 images in 10 categories, with 6000 images per category, of which 5000 are training images and 1000 are test images. Similarly, we only use the test images in CIFAR-10 as the benchmark testing set.

For each category, we first obtain the semantic refinement subcategories by searching in the Google Books Ngram Corpus. Then we retrieve the top 100 images from the Google image search engine for each candidate subcategory to represent its visual distribution. Particularly, we have released the discovered candidate subcategories and retrieved images for all the candidate subcategories on website https://1drv.ms/f/s!Ahpq3qST1g8NxyyjGslE2kjGevTV.

We calculate the Normalized Google Distance between subcategory and its parent category. We obtain the center of each subcategory by using the K-means clustering algorithm [36] (k=1), then we calculate the normalized visual distance (Euclidean distance) between subcategory and its parent category; calculate the normalized minimum, maximum, average and standard deviation of inter visual distance between subcategory and other subcategories; calculate the normalized minimum, maximum, average and standard deviation of intra-visual distance between different images in subcategory. We label a set of 500 positive samples and 500 negative samples to learn the linear SVM classifier for removing noisy subcategories and selecting useful subcategories.

After we obtain the selected subcategories, the first M = 50 images were selected for constructing the positive bags which corresponding to the selected subcategories. Negative bags can be obtained by randomly sampling a few irrelevant images. By treating each selected subcategory as a “bag” and the images therein as “instances”, we formulate a multi-class MIL method to select a subset of training images from each bag and simultaneously learning the optimal classifiers based on the selected images. Particularly, we define each positive bag as having at least a portion of η = 0.7 positive instances and set the tradeoff parameter C1 = 10−1. We will discuss the parameter setting more details in Section IV-D. To compare with other baseline methods, we evenly select 500 images from positive bags for each category as the training set. For this experiment, the feature is dense HOG [11].

2) Baselines: In order to quantify the performance of our proposed approach, four set of weakly supervised or web-supervised baselines are selected to compare with our proposed approach:

- SVM method. The SVM method includes multi-class SVM [50]. For the multi-class SVM, the 500 training images for each category are directly retrieved from the image search engine with the category label.
- MIL methods. The MIL methods contain instance level method mi-SVM [7] and bag level method sMIL [10]. For method mi-SVM, the training images are also retrieved from the image search engine. Particularly, we take the proposed heuristic way to iteratively select 500 images for each category and train the image classifier. For method sMIL, we firstly retrieve the candidate images from the image search engine, then we partition the candidate images into a set of clusters. Each cluster is treated as a “bag” and the images therein as “instances”. Correspondingly, we take the proposed MIL method to select the 500 training images for each category and train the image classifier.
- Latent domain discovering methods. The latent domain discovering methods include two methods DLD-MDA [32] and RVD-DA[30]. For method DLD-MDA, we firstly obtain the candidate images from the image search engine, then we take the hierarchical clustering technique to find the feasible latent domains. By treating each latent domain as a “bag” and the images therein as “instances”, we take the proposed MIL method to select 500 images for each category and train the image classifier. For method RVD-DA, after we obtain the candidate images from the image search engine, we take the proposed maximum distinctiveness and maximum learnability to find and separate the latent domains. Similarly, we take the

| Category | Number | Category | Number |
|----------|--------|----------|--------|
| Aero     | 204    | tabl     | 190    |
| Bicycle  | 259    | Dog      | 418    |
| Bird     | 282    | Horse    | 274    |
| Boat     | 172    | Mbike    | 222    |
| Bottle   | 212    | Person   | 2007   |
| Bus      | 174    | Plant    | 224    |
| Car      | 721    | Sheep    | 97     |
| Cat      | 322    | Sofa     | 223    |
| Chair    | 417    | Train    | 259    |
| Cow      | 127    | Tv       | 229    |

TABLE II: The detailed number of images used for categorization in PASCAL VOC 2007.
proposed MIL method to select 500 images for each category and train the image classifier.

- Sub-categorization methods. The sub-categorization methods Sub-Cate [31] and RNCMF [45] are also used to do image categorization. For method Sub-Cate, the candidate images are retrieved from the image search engine. Then we discover the subcategories during these candidate images by joint clustering and classification. We evenly select 500 images from these subcategories and train image classifiers. For method RNCMF, we also obtain the candidate images from the image search engine. We take the framework of Random Forests and the proposed regularized objective function to select 500 images for each category and train the image classifier.

For all the baseline methods, there are some parameters to be set in advance. All the training images for each category are obtained by retrieving from the Google image search engine. For the other parameters, we adopt the same parameter configuration as described in their original works.

3) Experimental results: The experimental results are summarized in Fig. 3, Table III and Table IV. From the results, we make the following observations:

During the 20 categories in PASCAL VOC 2007, we achieved the best results in 19 categories. In the 10 categories of STL-10 and CIFAR-10, we obtained the best results in 7 categories respectively. In addition, our approach also achieved the best average results on all three datasets.

We observe that the MIL learning methods [10], [7], the latent domain discovering methods [32], [50], the sub-categorization methods [31], [45] and our method are generally better than SVM method [50]. One possible explanation is that additional information like “bags”, latent domains or subcategories are beneficial in web training images for image categorization. In specific, MIL learning methods sMIL and mi-SVM achieve better results than SVM method multi-

Fig. 3: The detailed performance comparison of classification accuracy over 20 categories on the PASCAL VOC 2007 dataset.

TABLE III: The detailed performance comparison of classification accuracy over 10 categories on the STL-10 dataset.

| Method     | Category | Average |
|------------|----------|---------|
|            | airplane | bird    | car    | cat    | deer   | dog    | horse  | monkey | ship   | truck  |
| multi-SVM  | 0.497    | 0.193   | 0.384  | 0.222  | 0.467  | 0.293  | 0.534  | 0.355  | 0.443  | 0.348  | 0.374  |
| sMIL       | 0.526    | 0.221   | 0.433  | 0.285  | 0.504  | 0.339  | 0.593  | 0.427  | 0.523  | 0.381  | 0.423  |
| mi-SVM     | 0.531    | 0.242   | 0.464  | 0.283  | 0.528  | 0.346  | 0.642  | 0.404  | 0.547  | 0.482  | 0.444  |
| DLD-MDA    | 0.549    | 0.265   | 0.483  | 0.335  | 0.542  | 0.363  | 0.613  | 0.477  | 0.556  | 0.434  | 0.461  |
| RVD-DA     | 0.557    | 0.271   | 0.488  | 0.326  | 0.547  | 0.352  | 0.608  | 0.484  | 0.567  | 0.446  | 0.458  |
| Sub-Cate   | 0.553    | 0.294   | 0.482  | 0.331  | 0.535  | 0.354  | 0.616  | 0.456  | 0.550  | 0.443  | 0.465  |
| RNCMF      | 0.573    | 0.271   | 0.486  | 0.336  | 0.562  | 0.371  | 0.613  | 0.463  | 0.553  | 0.446  | 0.467  |
| Ours       | 0.596    | 0.284   | 0.516  | 0.366  | 0.582  | 0.397  | 0.636  | 0.502  | 0.582  | 0.473  | 0.493  |

TABLE IV: The detailed performance comparison of classification accuracy over 10 categories on the CIFAR-10 dataset.

| Method     | Category | Average |
|------------|----------|---------|
|            | airplane | car     | bird   | cat    | deer   | dog    | frog   | horse  | ship   | truck  |
| multi-SVM  | 0.397    | 0.304   | 0.094  | 0.163  | 0.345  | 0.264  | 0.153  | 0.423  | 0.342  | 0.277  | 0.276  |
| sMIL       | 0.423    | 0.302   | 0.073  | 0.241  | 0.352  | 0.285  | 0.201  | 0.453  | 0.325  | 0.271  | 0.293  |
| mi-SVM     | 0.422    | 0.328   | 0.103  | 0.232  | 0.354  | 0.323  | 0.203  | 0.414  | 0.372  | 0.274  | 0.302  |
| DLD-MDA    | 0.451    | 0.293   | 0.124  | 0.271  | 0.322  | 0.335  | 0.197  | 0.393  | 0.337  | 0.265  | 0.298  |
| RVD-DA     | 0.482    | 0.322   | 0.124  | 0.284  | 0.353  | 0.343  | 0.204  | 0.424  | 0.363  | 0.285  | 0.314  |
| Sub-Cate   | 0.433    | 0.303   | 0.132  | 0.283  | 0.363  | 0.352  | 0.214  | 0.454  | 0.323  | 0.285  | 0.318  |
| RNCMF      | 0.411    | 0.315   | 0.124  | 0.314  | 0.444  | 0.343  | 0.208  | 0.464  | 0.342  | 0.295  | 0.326  |
| Ours       | 0.452    | 0.373   | 0.113  | 0.342  | 0.403  | 0.322  | 0.265  | 0.502  | 0.382  | 0.322  | 0.357  |
SVM on all three datasets. The explanation is perhaps that it is necessary to remove noisy images from the training set during the process of classifier learning. Learning directly from the web images without noise removing may affect the performance of the classifier due to the presence of noisy images in the training data.

Sub-categorization methods RNCMF and Sub-Cate generally perform better than other two latent domains discovering methods DLD-MDA, RVD-DA and other MIL, SVM baseline methods. One possible explanation is that classifiers learned from Sub-categorization methods which exploiting subcategories to learn integrated classifiers are more domain robust than MIL methods, domain discovering methods and SVM baseline methods for image categorization.

It is interesting to observe that all classifiers have a relatively poor performance on dataset CIFAR-10 and STL-10 than on dataset PASCAL VOC 2007. The explanation is perhaps that all images in CIFAR-10 are cut to 32×32 and in STL-10 are 96×96. Objects in these small images are placed in the middle of the image. However, our web training images and the testing images in PASCAL VOC 2007 are both full size and contain relatively more additional objects or scenes in images.

Finally, our proposed approach achieves the best average performance on all three datasets, which demonstrate the superiority of our approach. The reason is our method simultaneously uses the MIL technique for handling label noise in the web training images and exploits multiple subcategories to learn integrated classifiers. Compared to SVM method, our method not only removes the noise, but also utilizes subcategories to learn integrated domain robust classifiers. Compared to MIL, latent domain discovering and sub-categorization methods, the multiple subcategories in our method have strong supervisory information which was obtained from the perspective of text semantics (e.g., subcategories discovering and noisy subcategories removing). This strong supervisory information can help us to maximize the inter-class variation and simultaneously minimize the intra-class variation.

### B. Image Sub-categorization

The objective of this experiment is to compare the image sub-categorization ability of our method with four weakly supervised or web-supervised baseline methods.

1) **Experimental setting:** For image sub-categorization, we choose a subset of ImageNet as the benchmark dataset for testing different methods. The reason is that ImageNet which constructed according to the WordNet has a hierarchy structure. In particular, we select five categories including “airplane”, “bird”, “cat”, “dog” and “horse” as the parent categories and all their leaf synsets as the subcategories. We are only concerned with the two-tier structure and deeper structure synsets are ignored. Thus, we obtain 5 parent categories and 97 subcategories. A detailed number of subcategories and corresponding images for each category in ImageNet for this experiment can be found in Table (V). We retrieve the top 100 images for each subcategory from the image search engine as the common original training images. So we have a total of 9700 training images for 5 parent categories and 97 subcategories. For a fair comparison with other baseline methods, we replace the subcategories discovering and noisy subcategories removing procedures with the given parent categories and subcategories from ImageNet in our work. So the initial value of $C$ and $M$ in our work is 5 and 97 respectively. For this experiment, the feature is 1000-dimensional bag-of-visual-words (BoW) based on densely sampled SIFT features [43]. The detailed list of the 97 subcategories in ImageNet and the common original 9700 web training images can be found on website https://1drv.ms/f/s!Ahpq3qSTtg8NsxyjGslE2kjGcvTV.

| Table V: The detailed number of subcategories and images used for image sub-categorization in ImageNet. |
|---------------------------------------------------------------|
| **Number of subcategories** |
| airplane | horse | bird | cat | dog |
| 15 | 29 | 26 | 9 | 18 |
| **Number of images** |
| 1434 | 1402 | 2126 | 1404 | 1603 |
possible explanation is that the image search engine provides
20 or 30 and decreases monotonically after this peaks. One
SVM and Sub-Cate peaks at the value of training numbers
for each subcategory.

some noise may be included. We need to select useful images
learning. Due to the error index of the image search engine,
our method considers the noise during the process of classifier
training images over 20 for each subcategory. The reason is
method Sub-Cate and multi-SVM, particularly the number of
subcategory.

using a varying number of training images and testing images.
sub-categorization results achieved by different methods when
learning function to select images from the retrieved images
therein as instances. We take the proposed multi-instance
subcategories are assumed as "bags" and the retrieved images
multi-instance learning problem. For this experiment, the
latent cluster number for each parent category
is all limited to two levels.

error index of image search engine. We assume there are five
trees corresponding to our five parent categories and start the
recursively learning. The depth of the tree for this experiment
is all limited to two levels.

MMDL. MMDL formulate image selection as a
multi-instance learning problem. For this experiment, the
subcategories are assumed as "bags" and the retrieved images
themselves as instances. We take the proposed multi-instance
learning function to select images from the retrieved images
and learn the image classifiers.

3) Experimental results: Fig. 4 and Fig. 5 present the image
sub-categorization results achieved by different methods when
using a varying number of training images and testing images.
The accuracy is measured by the average classification rate per
subcategory.

By observing Fig. 4 the best performance is achieved by
our method, which produces significant improvements over
method Sub-Cate and multi-SVM, particularly the number of
training images over 20 for each subcategory. The reason is
our method considers the noise during the process of classifier
learning. Due to the error index of the image search engine,
some noise may be included. We need to select useful images
from the retrieved candidate images to learn robust classifiers
for each subcategory.

From Fig. 4 we notice that the performance of the multi-
SVM and Sub-Cate peaks at the value of training numbers
20 or 30 and decreases monotonically after this peaks. One
possible explanation is that the image search engine provides
images based on the estimated relevancy with respect to the
query. Images far down in the ranking list are more likely to be
noise, which may result in degrading of the sub-categorization
accuracy especially for non-robust methods multi-SVM and
Sub-Cate.

It is interesting to note in Fig. 4, while method RNCMF im-
plements a form of noisy images removing, the classification
accuracy did not improve with the number of training images
increase. One possible explanation is that the noise in the
training data is not the only factor that affects the classification
accuracy. The visual distribution of the selected images is
another important factor. Furthermore, the poor accuracy of
Sub-Cate suggests that naively adding the number of training
images without considering the visual distributions not only
does not help but actually worsens the classification accuracy.

By observing Fig. 4 and Fig. 5 our approach compares
very favorably with competing algorithms, in terms of different
numbers of training and testing images. Compared to method
multi-SVM, Sub-Cate and RNCMF, our approach achieves
significant improvements in the sub-categorization accuracy.
The reason is our approach not only considers the possible
presence of noise in the web training data, but also tries
to ensure the diversity of the selected images for classifier
learning. Besides, our approach performs better than method
MMDL. The reason is we formulate image selection and
classifier learning as a multi-class MIL problem. Compared to
method MMDL which uses the relationship between different
subcategories for classifier learning, our method not only
uses the relationships between various subcategories, but also
leverage the relationships between various parent categories.
Therefore, our method achieves a much better result than
MMDL.

C. Quantitative Analysis of Different Steps

Our proposed framework involves three major steps: subcat-
egories discovering, noisy subcategories removing and multi-
class multi-instance learning. In order to quantify the role
different steps contributing to the final classifiers, we
construct two new frameworks. One is based on subcategories
discovering and noisy subcategories removing (which we
refer to SDNSR). Another one is based on subcategories
discovering and multi-class multi-instance learning (which we refer to SDMMIL). For framework SDNSR, we first obtain the candidate subcategories through searching in the Google Books Ngram Corpus. Then we apply the noisy subcategories removing procedure to get the selected subcategories. We directly retrieve the top images from the image search engine for selected subcategories to train image classifiers (without noisy images removing). For framework SDMMIL, we also obtain the candidate subcategories by searching in the Google Books Ngram Corpus. Then we retrieve the top images from the image search engine for all the candidate subcategories (without noisy subcategories removing). We apply the multi-class MIL model to select useful images and train image classifiers.

We compare the image classification ability of these two new frameworks with our proposed framework. Specifically, “airplane”, “dog”, “horse” and “bird” are selected as four target categories to compare the image categorization ability. We sequentially collect \([200,400,600,800,1000]\) images for each category as the positive training samples and use \(1000\) fixed irrelevant negative samples to learn image classifiers. We test the image classification ability of these three frameworks on the PASCAL VOC 2007 dataset. The results are shown in Fig. 7. By observing Fig. 6 we have the following observations: Framework SDNSR usually performs better than SDMMIL when the number of training images for each category is below 600. One possible explanation is that the first few images retrieved from the image search engine tend to have a relatively high accuracy. When the number of training images is below 600, the noisy images induced by noisy subcategories are more serious than those caused by the image search engine. With the increase of image numbers for each category, the images retrieved from the image search engine contain more and more noise. In this condition, the noisy images caused by the image search engine have a worse effect than those induced by noisy subcategories.

Our proposed framework outperforms both SDNSR and SDMMIL. This is because our approach, which takes a combination of noisy subcategories removing and noisy images filtering, can effectively remove the noisy images induced by both noisy subcategories and the error index of image search engine. Our framework can maximize the filtering of noisy images while maintaining the diversity of the selected images, thereby reducing the negative impact of noisy images on the classifier.

D. Parameter Sensitivity Analysis

Our proposed multi-class multi-instance learning formulation contains two parameters \(C_1\) and \(\eta\). PASCAL VOC 2007 was selected as the benchmark testing dataset to evaluate the performance variation of our proposed approach. In particular, we vary one parameter by fixing another parameter as the default value. Fig. 7 presents the parameter sensitiveness of \(C_1\) and \(\eta\) in terms of image classification accuracy.

By observing Fig. 7, we found our method is robust to the parameter \(C_1\) when it is varied in a certain range. Besides, the performance of our method is growing when \(\eta\) increase but less than 0.7. The reason is perhaps that our training data derived from image search engine. Due to the error index of image search engine, there may be too much noise in each bag which will result in decreasing the classification accuracy when \(\eta \leq 0.7\). When \(\eta\) increases over 0.7, the performance of our method decreases. One possible explanation is that the training set is less diverse. With the increasing of \(\eta\), the number of subcategories is decreasing, which may lead to the degradation of domain robustness of the classifier.

E. Time Complexity Analysis

During the process of multi-class multi-instance learning, we solve the convex problem in (7) by the cutting-plane algorithm. Through finding the most violating candidate \(\mathbf{h}_i\) and solve the MKL subproblem at each iteration, the time complexity of (7) can be approximately computed as \(T \cdot O(MKL)\), where the \(T\) is the number of iterations and the \(O(MKL)\) is the time complexity of the MKL sub-problem. According to (17), (52), the time complexity of MKL is between \(t \cdot O(LCM)\) and \(t \cdot O((LCM)^{2.3})\), where \(M, L, C\) are the numbers of latent domains, bags and categories respectively. \(t\) is the number of iterations in MKL.

We take STL-10 as the testing set to evaluate our method. Particularly, we use various numbers of training images for each category to learn the classifiers. STL-10 has 10 categories and we use \(n\) training images for each category, so we have a...
total of 10 \( n \) training images. Fig. [8] shows the training time and image classification accuracies with respect to various numbers of training images. From Fig. [8], we can observe that both of training time and image classification accuracy increase with the number of training images grows. We report the configuration of our experiment. Two HP PCs (3.2GHz CPU with 8 GByte RAM) were used for the web images collection. All the data processing and experiments were performed on an Acer PC (3.5GHz CPU, 16 GByte RAM and 4 GByte VRAM) with LIBSVM [49].

V. CONCLUSION

In this paper, we presented a new framework for classifying images into categories and subcategories. Three successive modules were employed in the framework including subcategories discovering, noisy subcategories removing and multi-class multi-instance learning. Compared to existing methods, our proposed approach can not only classify images into NOUN subcategories, but also into VERB, ADJECTIVE and ADVERB subcategories. Our approach has a better semantic refinement descriptions for the categories. To verify the effectiveness of our proposed approach, we conducted experiments on both image categorization and sub-categorization tasks. The experimental results demonstrated the superiority of our proposed approach over existing weakly supervised and web-supervised approaches.

APPENDIX A

THE DERIVATIONS OF (5) IS THE DUAL FORM OF (3)

Proof: In order to deduce the dual form of (3), we introduce an auxiliary variable \( \tau \) which is defined as:

\[
\pi_{i,c,s,\hat{s}} = \begin{cases} 
P_{1,s} & c = y_i \\
\delta(s = \hat{s}) & c \neq y_i 
\end{cases}
\]

where \( y_i = Y_M, \forall i \in I_m \). Then, we can get:

\[
\sum_{s=1}^{S} P_{1,s}w_{y_i,s} \phi(x_i) = \sum_{s=1}^{S} \sum_{i \in I_m} \pi_{i,y_i,s,\hat{s}} w_{y_i,s} \phi(x_i)
\]

\[
w_{c,\hat{s}} \phi(x_i) = \sum_{s=1}^{S} \pi_{i,c,s,\hat{s}} w_{c,s} \phi(x_i).
\]

We can further rewrite the constraints in (4) as the following forms:

\[
\frac{1}{|G_m|} \sum_{i \in I_m} h_i \sum_{s=1}^{S} \pi_{i,Y_M,s,\hat{s}} (w_{Y_M,s})^T \phi(x_i) - \sum_{s=1}^{S} \pi_{i,c,s,\hat{s}} (w_{c,s})^T \phi(x_i) \geq \zeta_{m,c,\hat{s}} - \zeta_{m,s,\hat{s}}, \forall m, \hat{s}, c
\]

where \( \zeta_{m,c,\hat{s}} = 0 \) if \( c = Y_m \) and \( \zeta_{m,c,\hat{s}} = \eta \) otherwise. We define:

\[
w = [w_{1,1}, \ldots, w_{1,S}, w_{2,1}, \ldots, w_{C,S}]^T
\]

and a new mapping function for \( G_m \) as:

\[
\psi(h, G_m, c, \hat{s}) = \left( \frac{1}{|G_m|} \sum_{i \in I_m} h_i \pi_{i,1,1,\hat{s}} \delta(c = 1) \phi(x_i)^T, \ldots, \right.
\]

\[
\frac{1}{|G_m|} \sum_{i \in I_m} h_i \pi_{i,c,1,\hat{s}} \delta(c = C) \phi(x_i)^T \left).\right)^T.
\]

By further denoting:

\[
\varphi(h, G_m, c, \hat{s}) = \psi(h, G_m, Y_M, \hat{s}) - \psi(h, G_m, c, \hat{s}),
\]

the problem of (3) can be written as:

\[
\min_{h, w, \xi_m} \frac{1}{2} \| w \|^2 + C_1 \sum_{m=1}^{M} \xi_m
\]

s.t. \( w^T \varphi(h, G_m, c, s) \geq \zeta_{m,c,s} - \zeta_{m,s}, \forall m, c, s. \) (17)

By introducing a dual variable \( \alpha_{m,c,s} \) for each constraint in (17), we can get the Lagrangian as:

\[
\mathcal{L}(w, \xi_m, \alpha_{m,c,s}) = \frac{1}{2} \| w \|^2 + C_1 \sum_{m=1}^{M} \xi_m - \sum_{m,c,s} \alpha_{m,c,s} (w^T \varphi(h, G_m, c, s) - \zeta_{m,c,s} + \xi_m).
\]

Through set the derivatives of \( \mathcal{L} \) with respect to \( w \) and \( \xi_m \) as zeros, we can get:

\[
\sum_{c,s} \alpha_{m,c,s} = C_1
\]

\[
w = \sum_{m,c,s} \alpha_{m,c,s} \varphi(h, G_m, c, s)
\]

(20)

Through submit the obtained equalities (19) and (20) back into (18), we can get the dual form of (3) as (5), which completes the proof.

APPENDIX B

THE DERIVATIONS OF (6) IS THE DUAL FORM OF (7)

Proof: We firstly introduce a dual variable \( \alpha_{m,c,s} \) for constraint in (6), then we can rewrite the Lagrangian form of (7) as:

\[
\mathcal{L} = \frac{1}{2} \sum_{t=1}^{T} \| w_t \|^2 + C_1 \sum_{m=1}^{M} \xi_m - \sum_{m,c,s} \alpha_{m,c,s} \varphi(h_t, G_m, c, s) - \zeta_{m,c,s} + \xi_m)
\]

Through set the derivatives of \( \mathcal{L} \) w.r.t. \( w_t \) and \( \xi_m \) as zeros respectively, we can get:

\[
w_t = d_t \sum_{m,c,s} \alpha_{m,c,s} \varphi(h_t, G_m, c, s), \forall t,
\]

\[
\sum_{c,s} \alpha_{m,c,s} = C_1, \forall m.
\]

Through submit (22) back into (7), we can arrive at the objective function in (6), which completes the proof.
