Taxonomy Augmented Object Recognition

Xiaoyang Wang
Nokia Bell Labs
Murray Hill, NJ, USA
Email: xiaoyang.wang@nokia-bell-labs.com

Yue Zhao
Minzu University of China
Beijing, China
Email: zhaoyueso@muc.edu.cn

Qiang Ji
Rensselaer Polytechnic Institute
Troy, NY, USA
Email: jiq@rpi.edu

Abstract—Realistic scene object recognition in computer vision still faces great challenges due to the large intra-class variation of object images caused by factors like object appearance variation and viewpoint change. To address this challenge, we propose to exploit the semantic relationships embedded in object taxonomy for improved object recognition. Specifically, we exploit the relationships in the object taxonomy to augment the learning of object classifiers. We utilize two types of relationships in the taxonomy, including the overall relationship and the local relationship. Our proposed approach jointly incorporates both the overall relationship as the loss function for classifier learning, and the local relationship as classifier learning constraints. Experiments on benchmark datasets demonstrate the effectiveness of our method in incorporating taxonomy for object recognition compared to the state-of-the-art methods.

I. INTRODUCTION

Object recognition in computer vision generally refers to the classification of object images into different categories such as “bird,” “aeroplane,” and “bicycle.” Due to the large intra-class variation of objects caused by object appearance variation, viewpoint change, illumination change, and occlusion, it is challenging to correctly recognize categories of real objects from natural scene images. On the other hand, the object taxonomy, which can be defined as the hierarchical representation of object categories and their super-categories, generally exists as structural prior information. Effectively exploiting the relationships in the taxonomy could potentially benefit the learning of object classifiers.

![Object Taxonomy Example]

Fig. 1. An exemplar object taxonomy with object categories “polar bear,” “grizzly bear,” “wolf,” and “fox” as leaf nodes in the bottom level, and “bear,” “canine,” and “carnivore” as super-categories in higher levels.

In object recognition, different object categories like “polar bear” and “grizzly bear” could be semantically related to each other by sharing a super-category like “bear.” Moreover, different super-categories could further share one super-category in a higher level. The object taxonomy hence utilizes the hierarchical representation to capture these semantical relationships of objects and super-categories. Fig. 1 shows an exemplar taxonomy for objects “polar bear,” “grizzly bear,” “wolf,” and “fox” as leaf nodes in the bottom level, and “bear,” “canine,” and “carnivore” as super-categories in higher levels.

Exploiting taxonomy for object recognition has become an active research direction in recent years. In large-scale and fine-grained object recognition applications, taxonomies are used for top-down hierarchical object classification in [1]–[4]. Also, approaches in [5]–[8] utilize the taxonomy to regularize the feature/embedding learning. Unlike the existing work that utilizes taxonomy for object classification, our approach focuses on utilizing the object category relationships in taxonomy to augment object classifier learning. In this paper, we propose two formulations for utilizing the object category relationships in taxonomy to augment object classifier learning, to respectively capture the local and overall relationships. We further propose a combined formulation to integrate both overall and local taxonomy relationships into one classifier learning function. Experiments show that both our formulations can improve results, and the combined formulation reaches the best performance.

II. RELATED WORK

Utilizing the object taxonomy for object recognition has drawn research attention in recent years. Existing taxonomy work for object recognition generally utilizes the object taxonomy in three types of frameworks, including embedding learning, classifier learning, and hierarchical classification. Of the three types of frameworks, our methods belong to classifier learning approach.

The existing taxonomy work in classifier learning [9]–[14] incorporates the information in the taxonomy to benefit classifier learning. Different approaches have been proposed to incorporate the taxonomy for metric learning [9], [10], multi-kernel learning [11], label sharing [12], and as the loss function [13], [14]. In these approaches, the local taxonomy relationships and the overall taxonomy relationships are generally utilized. The local taxonomy relationships for classifier learning are utilized in metric learning [9], [10] and multi-kernel learning [11]. However, these approaches utilize the parent-child inherent relationships to regularize the metric and kernel learning. Comparatively, in our local relationship formulation, we utilize the local relationship between category, sibling, and non-sibling nodes as constraints to improve the Structural SVM (SSVM) classifier learning.
Besides the local relationships, the overall taxonomy relationships for classifier learning are used in the label sharing [12] and loss functions [13]–[16] of classifier learning. However, these approaches directly use the semantic distance from the taxonomy as the loss function. Comparatively, our approach combines the semantic distance with the observation distance. Experiments show that semantic distance alone cannot improve recognition, but the combination of these distances can.

In general, the existing work that utilizes taxonomy to benefit classifier learning incorporates either the local or overall taxonomy relationships. Comparatively, our approach simultaneously incorporates both the local and overall relationships to augment the classifier learning. In addition, we employ the observation distance to the overall relationships.

The object taxonomy describes the semantic similarity/dissimilarity relationship between object categories. This relationship is different from the relationship of object attributes [17]–[20] which describes the co-occurring/mutually exclusive relations between the object and its attribute properties, or the context in [21]–[24] which encode the relationship between the target and its neighbors.

III. TAXONOMY IN OBJECT RECOGNITION

We first introduce the taxonomy for object recognition in Section III-A, and then discuss the overall and local taxonomy relationships in Section III-B and Section III-C, respectively.

A. Definition of Taxonomy

In object recognition tasks, the pre-defined object categories are generally semantically related to each other. One object category could share a super-category with a subset of other object categories. For example, both the object “polar bear” and “grizzly bear” defined in the AWA dataset [25] have the same super-category, “bear.” Moreover, different super-categories could further share one super-category in a higher level. For example, both the super-categories “bear” and “canine” would belong to super-category “carnivore.” Such hierarchical semantic relationships between object categories and their super-categories can be represented by object taxonomy.

The object taxonomy generally refers to the hierarchical representation of object categories and their super-categories. With the built-in hierarchical structure (typically a tree structure), the object taxonomy effectively captures the hierarchical semantic relationships between object categories and their super-categories. Fig. 2 shows the taxonomy of Pascal object categories including 20 object categories and 10 super-categories. Also, Fig. 3 shows the taxonomy of the AWA object categories provided in [8]. It includes 50 object categories and 28 super-categories. These taxonomies reflect the semantic relationships of objects. They are manually constructed or generated from knowledge bases like WordNet [8], without utilizing the image inputs of objects. In this work, we study to utilize the object category relationships in the taxonomy to augment object classifier learning.

We observe that object category relationships in the taxonomy can be categorized into overall relationships and local relationships. Though capturing different properties of taxonomy, both types of relationships can be utilized in the classifier learning.

B. Overall Relationships in Taxonomy

The overall relationship captures the pairwise semantic distances between all leaf nodes in the taxonomy. Given a taxonomy structure $T$, the semantic cost matrix $C_s$ can be used to capture the overall relationship in the taxonomy. In the matrix $C_s$, element $C_s(a,b)$, which is the element in $a$th row and $b$th column of $C_s$, represents the semantic distance between leaf nodes $a$ and $b$ of taxonomy $T$, where node $a$ and $b$ each correspond to an object category or super-category.

Given the taxonomy $T$, the semantic distance between nodes $a$ and $b$ can be typically set as the length of the shortest path between node $a$ and node $b$ in the taxonomy. Besides the path length, [27] discuss other types of semantic distances in taxonomy such as Hirst-St-Onge distance [28]. In this work, we use the path length as the semantic distance.

C. Local Relationships in Taxonomy

The local relationships reflect the relationships between the category and its neighborhood nodes such as siblings and parents. To describe the local relationships, we represent a leaf node in taxonomy $T$ as $c$. Node $c$ can have a sibling node $s$, which is the node sharing the same parent with $c$, and a non-sibling leaf node $n$. Given the taxonomy structure, the local
relationship in taxonomy describes that the distance between node \( c \) and node \( s \) would be smaller than the distance between node \( c \) and node \( n \) as:

\[
\text{Distance}(c, s) < \text{Distance}(c, n) \quad (1)
\]

This distance constraint is intrinsically captured by the taxonomy structure. It tells that object category should be more similar to sibling category than to non-sibling category. Besides the local relationship between category, sibling, and non-sibling nodes, another type of local relationship is the relationship between category, sibling, and parent. We can say the object category should be more similar to its parent super-category than to its sibling category. If we denote the parent node as \( p \), this local relationship can be described as the distance constraint:

\[
\text{Distance}(c, p) < \text{Distance}(c, s) \quad (2)
\]

It describes that the distance between nodes \( c \) and \( p \) would be smaller than the distance between nodes \( c \) and \( s \).

IV. INCORPORATING TAXONOMY RELATIONSHIPS

In this work, we propose to incorporate both the overall taxonomy relationships and the local taxonomy relationships to augment the object classifier learning. With the Structure SVM [29] as multi-class classifier, we incorporate the overall relationships and local relationships in Section IV-A and IV-B respectively. Section IV-C further discusses the formulation to combine both relationships.

A. Overall Relationships as Loss Function

As discussed in Section III-B, the overall relationships in taxonomy can be captured by the semantic cost matrix \( C_s \), where its element \( C_s(a, b) \) represents the semantic distance scalar \( d_s(a, b) \) between leaf nodes \( a \) and \( b \) in the taxonomy. To incorporate the overall relationships, one way (as in [13]) is to use the semantic cost matrix \( C_s \) as the loss function \( \Delta_s \) to replace the traditional “0-1” loss in Structure SVM. This semantic loss function \( D_s \) can be written as:

\[
\Delta_s(y, y_i) = C_s(y, y_i) \quad (3)
\]

It would enforce a larger penalty for misclassifying label \( y \) into a category that is semantically farther from \( y \) in taxonomy, and a smaller penalty for misclassifying label \( y \) to a category that is semantically closer to \( y \).

However, object categories that are semantically far away from each other could have similarities in the image observations. For example, as shown in Fig. 4 (a) and (b), a “bird” could look similar to an “aeroplane” when both categories are captured in the sky with a low resolution, even though they are semantically far away from each other. On the other hand, object categories that are semantically close could be very different in image observations. For example, as shown in Fig. 4 (c) and (d), the categories “Chihuahua” and “dalmatian” are siblings in the taxonomy. However, due to the texture differences between these two, the image features of “Chihuahua” and “dalmatian” are quite different. In such cases, using the semantic distance alone in the loss function ignores the factor of “easy” and “hard” classification pairs for the classifiers. Enforcing the loss function to be proportional only to the semantic distance of object categories could cause the learned classifiers to lose accuracy in classification.

![Fig. 4. Examples of the object images with different semantic distances. (a) aeroplane and (b) bird that are semantically far away from each other with similar image observations due to low image resolution; (c) and (d) show images of sibling categories “Chihuahua” and “dalmatian” with very different image observations due to texture and viewpoint difference.](image)

Comparatively, rather than only using the semantic loss function defined in Eq. 3, we propose to combine both the semantic distance and the observation distance. Given the semantic distance \( d_s(a, b) \) as the shortest path length between leaf nodes \( a \) and \( b \) in the taxonomy \( \mathcal{T} \), we further define the observation distance scalar \( d_o(a, b) \) as:

\[
d_o(a, b) = \| \bar{x}_a - \bar{x}_b \|_2 \quad (4)
\]

where \( \bar{x}_a \) represents the mean of training samples for category \( a \), and \( \bar{x}_b \) represents the mean of training samples for category \( b \). The samples are pre-normalized before classifier learning.

We define the combined distance scalar \( d_T(a, b) \) as the weighted combination of the semantic distance and the observation distance. Given the weight \( \alpha \), the combined distance can be written as:

\[
d_T(a, b) = \alpha d_o(a, b) + (1 - \alpha) d_s(a, b) \quad (5)
\]

In practice, weight \( \alpha \) can be manually selected through the performance on the validation set. This combined distance hence synthesizes both semantic distance obtained from taxonomy, and observation distance calculated through image observations. Learning classifiers with combined distance would then take into account both the semantic similarity and observation similarity jointly.

Given the combined distance scalar \( d_T(a, b) \), we further obtain the taxonomy cost matrix \( C_T \) where each element \( C_T(a, b) \) represents the taxonomy combined distance scalar \( d_T(a, b) \) between nodes \( a \) and \( b \). The taxonomy loss function can then be defined as:

\[
\Delta_T(y, y_i) = C_T(y, y_i) \quad (6)
\]
With the taxonomy loss function $\Delta_T$, our learning formulation that incorporates the overall relationships in taxonomy can be written as:

$$\min_{w, \xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i \quad \text{subject to:}$$

$$w^\top \Psi(x_i, y_i) - w^\top \Psi(x_i, y) \geq \Delta_T(y, y_i) - \xi_i, \forall i, \forall y \neq y_i$$

$$\xi_i \geq 0, \forall i$$  \hspace{1cm} (7)

This learning function hence enforces the classifier learning to put a larger penalty for misclassifying label $y$ to a category that is farther from $y$ in the combined distance, and to put a smaller penalty for misclassifying label $y$ to a category that is semantically closer to $y$ in the combined distance.

The optimization in Eq. 7 can be solved in the same way as the standard Structural SVM [29] with cutting plane method [30].

**B. Capturing Local Relationships as Constraints**

The local relationships in the taxonomy, as mentioned in Section III-C, are captured by distance constraints between the category node and its neighborhood. In this work, we are interested in learning classifiers for object categories, which are the leaf nodes in the taxonomy. Hence, we utilize the local relationship between object category, its sibling, and non-sibling leaf nodes summarized in Eq. 1.

For a training sample $x_i$, with label $y_i$, if $y_i$ has a sibling, we denote this sibling category as $y_i^s$. We can also denote the non-sibling node of $y_i$ as $y_i^n$. In this work, we use the prediction scores of categories $y_i$, $y_i^s$, and $y_i^n$ to fulfill the distance constraint in Eq. 1. Since a category $y_i$ is more closely related to its sibling category $y_i^s$ than to the non-sibling category $y_i^n$, the prediction score of $y_i$ should be closer to $y_i^s$ than to $y_i^n$. This relation can be translated into constraint:

$$|w^\top \Psi(x_i, y_i) - w^\top \Psi(x_i, y_i^s)| \leq |w^\top \Psi(x_i, y_i) - w^\top \Psi(x_i, y_i^n)|$$  \hspace{1cm} (8)

Since $y_i$ is the label for sample $x_i$, the prediction scores of $y_i$ are expected to be larger than both the score of $y_i^s$ and the score of $y_i^n$. Hence, the constraint in Eq. 8 can be further simplified as:

$$w^\top \Psi(x_i, y_i) - w^\top \Psi(x_i, y_i^s) \geq 0$$  \hspace{1cm} (9)

This constraint will regularize the prediction score of $x_i$ for the sibling category $y_i^s$ to be larger than the score for the non-sibling category $y_i^n$. We can further introduce another slack variable $\eta_i$ to impose a soft constraint. This will result in the constraint as follows:

$$w^\top \Psi(x_i, y_i^s) - w^\top \Psi(x_i, y_i^n) \geq \Delta(y_i^s, y_i^n) - \eta_i$$  \hspace{1cm} (10)

To incorporate the local relationships in taxonomy into the classifier learning, we can add the constraint defined in Eq. 10. The formulation capturing the local taxonomy relationships can then be written as:

$$\min_{w, \xi, \eta} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i + C' \sum_{i=1}^{N} \eta_i \quad \text{subject to:}$$

$$w^\top \Psi(x_i, y_i) - w^\top \Psi(x_i, y) \geq \Delta(y, y_i) - \xi_i, \forall i, \forall y \neq y_i$$

$$w^\top \Psi(x_i, y_i^s) - w^\top \Psi(x_i, y_i^n) \geq \Delta(y_i^s, y_i^n) - \eta_i, \forall i, y_i^s, y_i^n$$

$$\xi_i \geq 0, \eta_i \geq 0, \forall i$$  \hspace{1cm} (11)

Here, $C' > 0$ is another coefficient that controls the trade-off between the regularization and slack variable penalty. This formulation hence enforces the score of the label $y_i$ to be larger than the remaining categories, and regularizes the score for the sibling category of $y_i$ to be larger than the score of non-siblings, both in a soft-margin manner. This optimization can be solved similarly to the Structural SVM optimization using the cutting plane method [30].

For the formulation in Eq. 11, the “0-1” loss function $\Delta_{0/1}$ can be used for both constraints. In Section IV-C, we further combine the overall relationships as taxonomy loss functions, and the local relationships as constraints.

**C. Combined Formulation**

The formulation discussed in Section IV-A utilizes the overall relationships in taxonomy as the loss function to put different penalties on different distances of misclassifications. The formulation in Section IV-B models the local relationships between the category and its siblings as extra constraints for classifier learning. These two formulations incorporate different relationships of the taxonomy, and hence can be further combined into one objective function. The combined objective function can be written as:

$$\min_{w, \xi, \eta} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \xi_i + C' \sum_{i=1}^{N} \eta_i \quad \text{subject to:}$$

$$w^\top \Psi(x_i, y_i) - w^\top \Psi(x_i, y) \geq \Delta_T(y, y_i) - \xi_i, \forall i, \forall y \neq y_i$$

$$w^\top \Psi(x_i, y_i^s) - w^\top \Psi(x_i, y_i^n) \geq \Delta(y_i^s, y_i^n) - \eta_i, \forall i, y_i^s, y_i^n$$

$$\xi_i \geq 0, \eta_i \geq 0, \forall i$$  \hspace{1cm} (12)

In this objective function, the first constraint uses the taxonomy loss function $\Delta_T$, which is defined in Eq. 6, to capture the overall relationships in the taxonomy. On the other hand, the second constraint itself captures the local relationships in the taxonomy. For the second constraint, we test the $\Delta(y_i^s, y_i^n)$ both as $\Delta_T$ and as $\Delta_{0/1}$, and find the $\Delta_{0/1}$ returns better performance. It demonstrates that local relationships should be captured as a weak constraint instead a strong one.

The learning of the objective function in Eq. 12 can still be solved similarly to the Structural SVM optimization using the cutting plane method [30].

**V. Experiments**

In this section, we demonstrate the performance of our proposed method for object recognition. In the experiments, we first evaluate the performance of the first formulation discussed in Section IV-A, which incorporates overall relationships in the taxonomy as loss function. For simplicity, we denote this formulation as Model-Overall. Also, we evaluate the performance on the second formulation discussed in Section IV-B, which captures the local relationships in taxonomy. This formulation can be denoted as Model-Local. We denote our combined formulation discussed in Section IV-C as Model-Combined. This combined formulation is compared with the Model-Overall approach, the Model-Local approach, and the Baseline approach, which uses Structural SVM with “0-1”...
loss function. We also compare the Model-Combined approach with different state-of-the-art approaches.

We use two datasets for the experiments. The first dataset is the aPascal dataset [31] from the Pascal VOC 2008 challenge [26] with twenty object categories. It contains 6,340 training samples and 6,355 testing samples. The taxonomy of aPascal is shown in Fig. 2. The second dataset is the Animal with Attributes (AWA) dataset [25]. This dataset contains 50 categories of animals from 30,475 images. The taxonomy of AWA is shown in Fig. 3. The object images in both datasets are captured in realistic scenes.

A. Results Integrating Overall Relationships

To evaluate the performance of the Model-Overall formulation that incorporates overall relationships in taxonomy as the loss function, we compare it to three approaches. The first approach is the Baseline approach. It directly uses the $\Delta_{0/1}$ loss function which is the “0-1” loss function in classifier learning. The second approach uses Structural SVM with semantic loss function $\Delta_s$ which uses only the semantic distance from the taxonomy. The third approach uses SSVM with the observation loss function $\Delta_o$ where the observation distance is obtained in Eq. 4. Unlike these three approaches, the proposed Model-Overall approach uses the combined taxonomy loss function $\Delta_T$ defined in Eq. 6 in classifier learning.

For this comparison, we test on the aPascal dataset with 4-object and 8-object recognition. In the 4-object recognition, we recognize the four objects including “bird,” “dog,” “cow,” and “horse.” In the 8-object recognition, four additional objects “sheep,” “car,” “bicycle,” and “motorbike” are included. We use the 112-dimensional HOG feature extracted from the object image as the feature for training and testing. The results are shown in Table I.

<table>
<thead>
<tr>
<th>Model</th>
<th>4-object Accuracy</th>
<th>8-object Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline ($\Delta_{0/1}$)</td>
<td>37.76%</td>
<td>27.68%</td>
</tr>
<tr>
<td>SSVM ($\Delta_s$)</td>
<td>34.96%</td>
<td>24.27%</td>
</tr>
<tr>
<td>SSVM ($\Delta_o$)</td>
<td>37.26%</td>
<td>26.06%</td>
</tr>
<tr>
<td>Model-Overall</td>
<td>39.74%</td>
<td>27.79%</td>
</tr>
</tbody>
</table>

In Table I, the mean recognition accuracies for each category are used as classifier accuracy evaluation. From the results in Table I, we can see that directly using the semantic distance in the corresponding loss function $\Delta_s$ performs even worse than the baseline for both recognition cases. This result proves that the semantic distance does not reflect the distance between category observations. On the other hand, the observation loss function $\Delta_o$ performs better than $\Delta_s$. It proves that the observation loss function could also be used as the loss function. Finally, the proposed Model-Overall approach combines the semantic loss function and observation loss function, and it improves the Baseline in both cases. This result demonstrates that incorporating the overall relationships in the taxonomy through our proposed loss function can improve classifier learning.

B. Results for Capturing Local Relationships

We further evaluate the performance of the Model-Local formulation which captures the local relationships in the taxonomy as constraints, and the Model-Combined formulation which combines the overall relationships and the local relationships in the taxonomy. We test with the 4-object and 8-object recognition in the same setting as the experiments in Section V-A. The results are shown in Table II.

<table>
<thead>
<tr>
<th>Model</th>
<th>4-object Accuracy</th>
<th>8-object Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>37.76%</td>
<td>27.68%</td>
</tr>
<tr>
<td>Model-Overall</td>
<td>39.74%</td>
<td>27.79%</td>
</tr>
<tr>
<td>Model-Local</td>
<td>40.69%</td>
<td>28.90%</td>
</tr>
<tr>
<td>Model-Combined</td>
<td>42.93%</td>
<td>32.81%</td>
</tr>
</tbody>
</table>

The results in Table II show a lot about the proposed formulations. Firstly, the local taxonomy relationships captured by the Model-Local approach obviously outperforms the Baseline approach. It also outperforms the Model-Overall approach which incorporates the overall relationships as loss functions. This result shows that the local relationships are very effective when used as constraints to improve object classifier learning. Moreover, the Model-Combined approach that combines the overall and local taxonomy relationships further improves over the Model-Local approach. It improves the Baseline approach by about 5% in both 4-object and 8-object recognition. This result shows that our proposed formulation which simultaneously incorporates the overall and local relationships in the taxonomy can effectively improve the classifier learning.

C. Comparisons with State-of-the-Art Methods

We compare the proposed Model-Combined approach with state-of-the-art methods on the AWA dataset. The taxonomy of this dataset is generated in [8] using the WordNet hierarchy. To make comparisons with the state-of-the-art methods that also use taxonomy to benefit object recognition, We follow the same experiment setting as [8] on the AWA dataset. Specifically, for each of the 50 object categories in AWA, we randomly split the data into 30 training samples, 30 validation samples, and 30 testing samples. This split results in 1,500 samples for the training, validation and test set respectively. Using the same strategy as [8], we perform two sets of experiments on the AWA dataset using the features provided in AWA. The first set is the AWA-PCA. It performs PCA on each of the six provided features including SIFT, rgSIFT, SURF, HOG, LSS, and CQ, and then reduce the feature dimension to 50 through PCA. These six types of features are concatenated to formalize the 300 dimensional feature vector for each sample of AWA-PCA. The second set of experiments is AWA-DeCAF. It directly uses the provided
4,096 dimensional DeCAF [32] features which are the outputs from the layer that is before the label output layer of the convolutional neural network.

On AWA dataset, we compare our method to two state-of-the-art taxonomy methods [5] and [8] for object recognition. The first approach [5] enforces the distance of embeddings between two categories to be close to their semantic distance in the taxonomy. The second approach [8] extends [5] with attribute embeddings, category embeddings, and super-category embeddings. Both [5] and [8] integrate the taxonomy as regularization terms in the objective function. Comparatively, our method incorporates taxonomy as the loss function and constraints. More importantly, [5] and [8] use taxonomy relationships to benefit the embedding learning. Comparatively, our method utilizes taxonomy relations to benefit classifier parameter learning. In addition to these state-of-the-art methods, we also compare to our Baseline approach, which directly uses Structural SVM without incorporating any relationships in the taxonomy. The results for AWA 50 object category experiments are given in Table III.

<table>
<thead>
<tr>
<th>Model Accuracy (%)</th>
<th>AWA-PCA</th>
<th>AWA-DeCAF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>0.19 ± 0.06</td>
<td>0.45 ± 0.85</td>
</tr>
<tr>
<td>Weinberger &amp; Chapelle [5]</td>
<td>0.20 ± 0.64</td>
<td>0.38 ± 1.12</td>
</tr>
<tr>
<td>Hwang &amp; Sigal [8]</td>
<td>0.21 ± 0.02</td>
<td>0.46 ± 1.33</td>
</tr>
<tr>
<td><strong>Our Model-Combined</strong></td>
<td>0.21 ± 0.09</td>
<td>0.46 ± 1.23</td>
</tr>
</tbody>
</table>

From Table III, we can see both the approach by [5] and the approach by [8] can outperform the Baseline approach in the AWA-PCA set. The approach by [8] also outperforms our Baseline approach on the AWA-DeCAF set. More importantly, our Model-Combined approach can outperform both state-of-the-art approaches on both the AWA-PCA and the AWA-DeCAF sets in average performance.

VI. CONCLUSION

In this paper, we propose to incorporate the taxonomy relationships to augment the learning of object classifiers. We observe that there are two types of relationships in the taxonomy, including the overall relationship and the local relationship. We propose two formulations to incorporate the taxonomy in object classifier learning, with one formulation capturing the overall relationship as loss function, and the other formulation capturing the local relationships as constraints. We further incorporate both overall and local relationships in one unified formulation. Experiments show that incorporating either the overall or local relationships can improve object classifier learning, and the combined formulation reaches the best performance. These results show that incorporating taxonomy through our proposed method can effectively benefit object classifier learning.

ACKNOWLEDGMENT

The work is supported in part by the grant W911NF-13-1-0395 from the Army Research Office.

REFERENCES