Multimedia *LEGO*: *LE*arning structured model by probabilistic lo*G*ic *O*ntology tree

Shiyu Chang*, Guo-Jun Qi*, Jinhui Tang†, Qi Tian‡, Yong Rui§ and Thomas S. Huang*
*Beckman Institute, University of Illinois at Urbana-Champaign
Email: {chang87,qi4,t-huang1}@illinois.edu
†School of Computer Science, Nanjing University of Science and Technology
Email: jinhuitang@mail.njust.edu.cn
‡Department of Computer Science, University of Texas at San Antonio
Email: qitian@cs.utsa.edu
§Microsoft Research Asia
Email: yongrui@microsoft.com

***Abstract*—Recent advances in Multimedia research have generated a large collection of concept models, e.g., LSCOM and Mediamill 101, which become accessible to other researchers. While most current research effort still focuses on building new concepts from scratch, little effort has been made on constructing new concepts upon the *existing* models already in the warehouse. To address this issue, we develop a new framework in this paper, termed LEGO, to seamlessly integrate both the new target training examples and the existing primitive concept models. LEGO treats the primitive concept models as a lego toy to potentially construct an unlimited vocabulary of new concepts. Specifically, LEGO first formulates the logic operations to be the *lego connectors* to combine existing concept models hierarchically in *probabilistic logic ontology trees*. LEGO then simultaneously incorporates new target training information to efficiently disambiguate the underlying logic tree and correct the error propagation. We present extensive experimental results on a large vehicle domain data set from ImageNet, and demonstrate significantly superior performance over existing state-of-the-art approaches which build new concept models from scratch.**

***Index Terms*—Multimedia LEGO, Concept recycling, Model warehouse, Probabilistic logic ontology tree, Logical operations.**

I. Introduction

Effectively modeling structured concepts has becoming a critical ingredient for recognizing, retrieving and searching image data on the Web. Many sophisticated models have been proposed to recognize a wide range of image concepts from our everyday life to many specific domains such as news video broadcast and surveillance videos. While people continue to build new models from scratch using Support Vector Machines (SVM) and its variants [1][2][3], we shall not forget the powerful knowledge base in the warehouse such as Large-Scale Concept for Multimedia (LSCOM) [4] and 101 semantic concepts in Mediamill 101 [5]. In this paper we will show how existing models can be seamlessly integrated with new target training samples to construct new complex models in an effective way.

Traditional approaches for image classification and recognition problems are sensitive to the number of samples involved in the models. Usually, a large number of samples provides a good generalization ability. However, in many cases, obtaining massive training data is difficult due to the expense of labeling. Moreover, computational power is another constraint to recognize a wide range of image concepts. To alleviate such difficulties, we develop an approach recycling existing semantics.

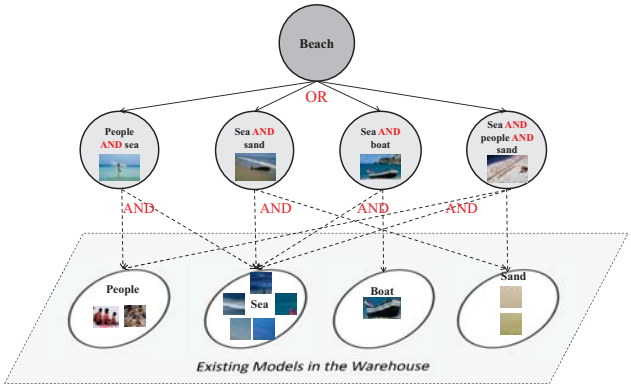

Fig. 1. An example of using logical hierarchical semantic ontology to model concepts.

Lego is a popular line of construction toys. Lego, consists of colorful interlocking plastic bricks and an accompanying array of gears, minifigures and various other parts. Lego bricks can be assembled and connected in many ways, to construct such objects as vehicles, buildings, and even working robots. Anything constructed can then be taken apart again, and the pieces used to make other objects.

By analogy to the lego constructing complex toys, the existing models in the warehouse can be seen as each interlocking plastic bricks in the toy. And we could use them to construct more complex concepts. Such a Multimedia "lego" model provides semantic-rich building blocks to construct new concepts, instead of starting with zero knowledge. It opens a new way to efficiently leverage a large number of existing primitive concepts for constructing potentially unlimited vocabularies of image concepts.

To connect the existing lego pieces, we need first to find

1550-4786/13 $31.00 © 2013 IEEE
DOI 10.1109/ICDM.2013.49

proper array of gears. By analogy to the toy lego, these array of gears play the key role of coherently connecting all the components as a whole. Let us investigate how human perceives a new concept in the real world. In childhood, we were taught to learn concrete concepts which can be directly recognized by their natural attributes, such as shape, color and materials. As we grow up, we learn how to use logics to connect these primitive concepts into more complex concepts. For example, "beach" is a fairly abstract concept from concepts "people", "sand", "boat", "sea" and so on. Take a look at the example illustrated in figure I, "sand", "sea", "people" and "boat" are parts of "beach". Thus "beach" can be represented by "(people AND sea)" OR "(sea AND sand)" OR "(sea AND boat)" OR "(sea AND people AND sand)" which exploits the possible combinations of parts of "beach" by a AND-OR relationship. It indicates that a hierarchical semantic ontology is reasonable to model concepts, in an order from the primitive concepts in the lower level to the complex ones in upper level by using various logical operations. In other words, once a collection of primitives is given, many other complex concepts can be built upon these primitive concepts by connecting them with logics.

Based on the above observations, we propose a novel LEGO approach, that is **LE**arning structured model by probabilistic lo**G**ical **O**ntology tree in this paper, which will construct structured concepts built upon a set of primitive models. The key contributions of this paper are:

- As opposed to many existing concept modeling techniques, LEGO integrates the logical and statistical inferences in a unified framework where the existing primitive concepts are connected into a potentially unlimited vocabulary of high-level concepts by the basic logical operations. In contrast, most existing modeling algorithms either only learn a flat correlative concept structure [6] [7], or a simple hierarchical structure without logical connections [8] [9] [10] [11] [12].
- With an efficient statistical learning algorithm, the complex concepts in the upper levels of hierarchy are modeled upon logically connected primitive concepts. This statistical learning approach is much more flexible, where each concept in the hierarchy can be modeled from heterogeneous feature spaces of the most suitable feature descriptors (e.g., visual features such as color and shape for scenery concepts, textual features such as TF-IDF for named-entities) or can be obtained from different semantic warehouse. This means that when we build the target model, we can select the LEGO made from different "materials" and still be able to connect these heterogeneous pieces of lego together.
- LEGO simultaneously incorporates both new target training information and lego building blocks. This setup allows LEGO to efficiently disambiguate the underlying logic tree and correct the error propagation only using a few of training samples. Especially for the situation of a large amount of concepts need to be categorized, and labeled data is deficient. It results in a significantly better performance than the SVM-type training algorithms that build new concepts from scratch.

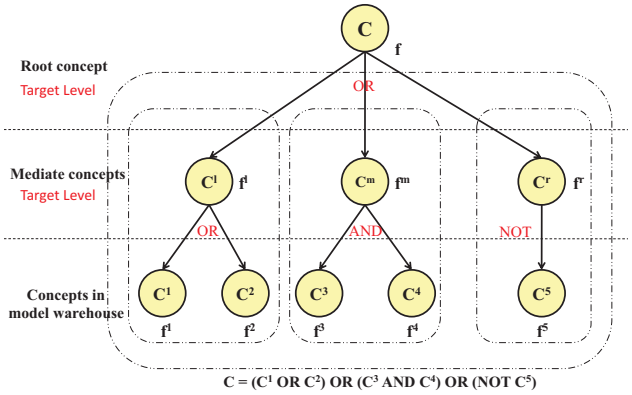

Fig. 2. An example of probabilistic logical ontology tree.

In summary, the primitive models as pieces of multimedia lego can be seen as building blocks by analogy to training examples in conventional classification problem, because both of them provide basic semantic information to infer new concept models. For example, a large number of models exist in many warehouses like LSCOM 374 and Mediamill 101. They provide us with rich semantic resources to explore information other than training examples. Indeed, these models, which are learned from example images, have already contained rich discriminative information about the primitive concepts. It ought to be much more efficient to mine these models directly, instead of coming back to tediously collecting labeled examples and retraining models again. These existing multimedia lego can save great resources and effort in the multimedia community by improving the utility of the existing research results.

II. Probabilistic Logical Ontology Tree

To present how we can apply LEGO to construct complex concepts, in this section we define concrete data learning structure - Probabilistic Logical Ontology Tree (PLOT).

A. Prior Probabilistic Models in PLOT

We start by the definition of PLOT with an example. As illustrated in figure 2, PLOT is a logical tree

$$T = \{(C, \mathrm{f}, L), (C^l, \mathrm{f}^l, L^l), (C^m, \mathrm{f}^m, L^m), (C^r, \mathrm{f}^r, L^r), (C^1, \mathrm{f}^1, P^1), \ldots, (C^5, \mathrm{f}^5, P^5)\}$$

where C and C^i are concept nodes, and f and f^i are different feature descriptors attached with C and C^i. For each upper level concept C^i other than leaf nodes, C^i can be expanded into a set of children concepts by a logical operation L^i from either OR, AND, or NOT. For each node, there is also an attached model $P^i(y|\mathrm{f}^i(x))$ predicting the probability of label being positive if $y = 1$ or negative if $y = 0$ given the feature $\mathrm{f}^i(x)$ for each sample x. The associated model can

have arbitrary flexible mathematical forms such as logistic regression model, exponential model or even support vector machine or boosting model (but should be normalized into probabilistic form first).

In PLOT, each complex concept in the upper level can be represented via the leaf concepts by the logical relations. Take an example of PLOT in figure 2, $C^l = C^1$ AND C^2, $C^m = C^3$ OR C^4, and $C = (C^1$ OR $C^2)$ OR $(C^3$ AND $C^4)$ OR (NOT C^5). Such classical Boolean logic gives two exclusive results: one sample is either positive or negative for the target concept. To formulate a corresponding prior probability model for learning and inference, the Boolean logic is converted into fuzzy logic which replaces AND, OR, NOT by some continuously probability conversions. There are many different fuzzy logical operations which can do such conversion, and here we enumerate two kinds of them as follows.

Min/Max/complement:
In this case, AND is replaced by "min", OR by "max", and NOT by $1 - P(y = 1|\mathrm{f}(x))$ where P is the model attached with the children node of the logic NOT. Take C in figure 2 as an example, the prior model $P_{prior}(y|\mathrm{f}(x))$ for C is

$$\begin{aligned}&P(y = 1|\mathrm{f}(x)) = \max\{\max\{P^1(y = 1|\mathrm{f}^1(x)),\\ &P^2(y = 1|\mathrm{f}^2(x)\}, \min\{P^3(y = 1|\mathrm{f}^3(x)),\\ &P^4(y = 1|\mathrm{f}^4(x))\}, 1 - P^5(y = 1|\mathrm{f}^5(x))\}.\end{aligned}$$

Probabilistic product/sum:
In this case, $C^m = C^3$ AND C^4 is replaced by

$$\begin{aligned}P^m(y = 1|f(x)) &= T_{pod}(P^3(y = 1|\mathrm{f}(x)), P^4(y = 1|\mathrm{f}(x)))\\ &= P^3(y = 1|\mathrm{f}(x))P^4(y = 1|\mathrm{f}(x)),\end{aligned}$$

$C^l = C^1$ OR C^2 is replaced by

$$\begin{aligned}P^l(y = 1|\mathrm{f}(x)) &= \perp_{sum}(P^1(y = 1|\mathrm{f}(x)), P^2(y = 1|\mathrm{f}(x)))\\ &= P^1(y = 1|\mathrm{f}(x)) + P^2(y = 1|f(x))\\ &\quad - P^1(y = 1|\mathrm{f}(x))P^2(y = 1|\mathrm{f}(x)).\end{aligned}$$

Again, NOT is converted by complement operation as in Case 1. The probabilistic product and sum is often called T-norm and T-conorm.

There are many other fuzzy logical operations to do a similar conversion from the classical Boolean logics to their probability forms, such as Lukasiewicz logic, Nilpotent logic and Hamacher logic. Interested readers can find more in [13].

To learn a satisfactory model for upper level concepts by PLOT, we still need to overcome the following two problems: semantic ambiguity, and error propagation. To overcome the problems, only using the information from the models associated with lower level nodes in PLOT is not enough. It requires some extra content-based examples to update the prior models purely obtained by the logical relations to clarify semantic ambiguity, and correct the errors from lower level models. In other words, two criterions are proposed when modeling the complex concepts in upper levels.

- **Criterion 1**: The model $P(y|\mathrm{f}(x))$ for the upper level concepts should preserve as much information of the prior model $P_{prior}(y|\mathrm{f}(x))$ as possible which combines the information on primitive models of lower level nodes in PLOT.
- **Criterion 2**: With the new extra training examples, the model $P(y|\mathrm{f}(x))$ for upper level concepts must reflect the information contained in these extra training examples.

Based on the above two criteria, we formulate the proposed probabilistic algorithms to model the high-level concepts on PLOT.

B. Learning and Inference on PLOT

Given a set of the models $P^m(y|\mathrm{f}^m(x))$ of low-level concepts C^m, $1 \le m \le M$, and some extra training examples $\{(x^l, y^l)|1 \le l \le N\}$, our goal is to learn a model $P(y|\mathrm{f}(x))$ for the target concept C based on a given PLOT.

First, according to PLOT, target concept C can be expanded into C^m by the logical relation uncovered by PLOT. Accordingly, we can obtain a prior model $P_{prior}(y|[\mathrm{f}^m(x)]_{m=1}^M)$ just as the example shown in figure 2. Then the new model $P(y|\mathrm{f}(x))$ should reflect the two criteria mentioned in the end of the last subsection. Therefore, we formulate the following optimization problem to solve it.

$$\begin{aligned}\min_{P(y|x)}\ &\frac{1}{N}\sum_{l=1}^{N} D_{KL}\left(P\left(y|\mathrm{f}(x_l)\right)||P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^M\right)\right)\\ s.t.\ &\frac{1}{N}\sum_{l=1}^{N}\mathop{\mathbb{E}}_{P(y|\mathrm{f}(x_l))}[y\mathrm{f}_d(x_l)] = \frac{1}{N}\sum_{l=1}^{N} y_l\mathrm{f}_d(x_l) + \theta_d\\ &\frac{1}{N}\sum_{l=1}^{N}\mathop{\mathbb{E}}_{P(y|\mathrm{f}(x_l))}[y] = \frac{1}{N}\sum_{l=1}^{N} y_l + \eta\\ &\sum_{y\in\{0,1\}} P\left(y|\mathrm{f}(x_l)\right) = 1\\ &\sum_{d=1}^{D}\frac{\theta_d^2}{2\sigma_\theta^2/N} + \frac{\theta_\eta^2}{2\sigma_\eta^2/N} \le C\\ &1 \le d \le D\end{aligned} \tag{1}$$

where D_{KL} is the Kullback-Leibler divergence. By minimizing the divergence between $P(y|\mathrm{f}(x))$ and $P_{prior}(y|[\mathrm{f}^m(x)]_{m=1}^M)$, the information in the prior model can be preserved as much as possible according to criterion 1. $\mathop{\mathbb{E}}_{P(y|\mathrm{f}(x_l))}[\cdot]$ is the expectation with respect to the distribution $P(y|\mathrm{f}(x_l))$ and $\mathrm{f}_d(x_l)$ is the d^{th} element of low-level feature vector $\mathrm{f}(x_l)$. The first two constraints in the above formulation requires the first two-order statistics of new model $P(y|\mathrm{f}(x))$ must comply training set up, to estimate errors θ_d and η. Furthermore, the third constraint normalizes the model so that it satisfies the probabilistic property. Finally, the fourth constraint assumes the joint probability of estimation errors should be reasonably upper bounded by C [14].

Inference on PLOT:
First, given equation (1), we see how to infer the model $P(y|\mathrm{f}(x))$ for the target concept. From (1), the Lagrangian

function is:

$$\begin{aligned}
&\mathcal{L}\left(P(y|\mathrm{f}(x_l)),\theta,\eta,b,w,\gamma,\xi\right)=\\
&\frac{1}{N}\sum_{l=1}^{N}D_{KL}\left(P\left(y|\mathrm{f}(x_l)\right)||P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^{M}\right)\right)+\\
&\sum_{d=1}^{D}w_d\left\{\frac{1}{N}\sum_{l=1}^{N}y_l\mathrm{f}_d(x_l)+\theta_d-\frac{1}{N}\sum_{l=1}^{N}\mathop{\mathbb{E}}_{P(y|\mathrm{f}(x_l))}\left[y\mathrm{f}_d(x_l)\right]\right\}\\
&+b\left\{\frac{1}{N}\sum_{l=1}^{N}y_l+\eta-\frac{1}{N}\sum_{l=1}^{N}\mathop{\mathbb{E}}_{P(y|\mathrm{f}(x_l))}\left[y\right]\right\}\\
&+\gamma\left\{\sum_{d=1}^{D}\frac{\theta_d^2}{2\sigma_\theta^2/N}+\frac{\eta^2}{2\sigma_\eta^2/N}-C\right\}\\
&+\sum_{x}\xi(x)\left(1-\sum_{y\in\{0,1\}}P\left(y|\mathrm{f}(x)\right)\right).
\end{aligned}\tag{2}$$

By deriving it with respect to $P(y|\mathrm{f}(x_l))$ and setting the results to be zero, we have:

$$\begin{aligned}
&\frac{\partial\mathcal{L}}{\partial P\left(y|\mathrm{f}(x_l)\right)}=\frac{1}{N}\{\log P(y|\mathrm{f}(x_l))+1\\
&-\log P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^{M}\right)-y(w^Tx_l+b)\}-\xi(x_l)=0,
\end{aligned}\tag{3}$$

and

$$\frac{\partial\mathcal{L}}{\partial\eta}=b+N\gamma\frac{\eta}{\sigma_\eta^2}=0,\quad\frac{\partial\mathcal{L}}{\partial\theta_d}=w_d+N\gamma\frac{\theta_d}{\sigma_\theta^2}=0.\tag{4}$$

From (3) we have:

$$P(y|\mathrm{f}(x_l))\propto P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^{M}\right)\exp\left\{y(w^Tx_l+b)\right\}.\tag{5}$$

Now,considering the normalization constraints in 1, which ought to be

$$P(y|\mathrm{f}(x_l))=\frac{1}{Z(x_l)}P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^{M}\right)\exp\left\{y(w^Tx_l+b)\right\},\tag{6}$$

where

$$Z(x_l)=\sum_{y\in\{0,1\}}P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^{M}\right)\exp\left\{y(w^Tx_l+b)\right\}$$

is the partition function for normalization. Thus we have obtained the target model as shown in equation (6), where the corresponding concept C can be inferred.

Learning on PLOT:

Now we show how to learn $P(y|\mathrm{f}(x))$, i.e., computing its model parameters w and b. From (4) we obtain

$$\eta=-\frac{b}{N\gamma}\sigma_\eta^2,\qquad\theta_d=-\frac{w_d}{N\gamma}\sigma_\theta^2.\tag{7}$$

Substitute (6) and (7) into Lagragian function (2), we can formulate the dual optimization problem as

$$\begin{aligned}
b^*,w^*&=\arg\max_{b,w}\mathcal{L}\left(P(y|\mathrm{f}(x_l)),\theta,\eta,b,w,\gamma,\xi\right)\\
&=\arg\max_{b,w}\frac{1}{N}\sum_{l=1}^{N}(y_l\left(w^T\mathrm{f}(x_l)+b\right)-\log Z\left(x_l\right)\\
&+\log P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^{M}\right)-\frac{\lambda_w}{2N}||w||_2^2-\frac{\lambda_b}{2N}b^2
\end{aligned}\tag{8}$$

where $\lambda_w=\frac{\sigma_\theta^2}{\gamma}$, $\lambda_b=\frac{\sigma_\eta^2}{\gamma}$ are the balance parameters. This maximization problem is unconstrained convex problem with respect to b and w, so a global maximum exists. Take the derivatives with respect to b and w, we have

$$\begin{aligned}
\frac{\partial\mathcal{L}}{\partial b}&=\frac{1}{N}\sum_{l=1}^{N}y_l-\frac{1}{N}\sum_{l=1}^{N}\mathop{\mathbb{E}}_{P(y|\mathrm{f}(x_l))}\left[y\right]-\frac{\lambda_b b}{N}\\
\frac{\partial\mathcal{L}}{\partial w_d}&=\frac{1}{N}\sum_{l=1}^{N}y_l\mathrm{f}_d(x_l)-\frac{1}{N}\sum_{l=1}^{N}\left(\mathop{\mathbb{E}}_{P(y|\mathrm{f}(x_l))}\left[y\right]\mathrm{f}_d(x_l)\right)-\frac{\lambda_w w_d}{N}
\end{aligned}\tag{9}$$

Then (8) can be maximized by a conjugate gradient method based on (8) and its derivatives in (9).

C. Efficient Online Modeling for Large Scale Problem

As more and more image data booms on the Internet, from image and video sharing web sites to various kinds of social communities, efficient modeling and recognition algorithms are required to handle these rising data. Among them, online modeling technique is very useful to handle the large scale data set where the samples are processed one by one. Assume we currently have a model $P(y|\mathrm{f}(x))=\frac{1}{Z(x)}P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x)\right]_{m=1}^{M}\right)\exp\left\{y(w^Tx+b)\right\}$ as equation (6) in hand, our goal is to obtain a new one $\tilde{P}(y|\mathrm{f}(x))$ by some new examples $\{\tilde{x}_l,\tilde{y}_l\}_{l=1}^{\tilde{N}}$. Following the similar idea in formulation (1), the new model should preserve as much information as possible in $P(y|\mathrm{f}(x))$ as well as reflect the new information in $\{\tilde{x}_l,\tilde{y}_l\}_{l=1}^{\tilde{N}}$. By substituting $\frac{1}{N}\sum_{l=1}^{N}D_{KL}\left(\tilde{P}(y|\mathrm{f}(x))||P\left(y|\mathrm{f}(x_l)\right)\right)$ into the objective function in (1) and the new examples in $\{\tilde{x}_l,\tilde{y}_l\}_{l=1}^{\tilde{N}}$ for those in $\{x_l,y_l\}_{l=1}^{N}$, we have the new model as

$$\begin{aligned}
\tilde{P}(y|\mathrm{f}(x))=&\frac{1}{\tilde{Z}(x)}P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x)\right]_{m=1}^{M}\right)\\
&\cdot\exp\left\{y\left((w+\tilde{w})^Tx+(b+\tilde{b})\right)\right\}
\end{aligned}\tag{10}$$

and

$$\begin{aligned}
\tilde{Z}(x)=&\sum_{y\in\{0,1\}}P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x)\right]_{m=1}^{M}\right)\\
&\cdot\exp\left\{y\left((w+\tilde{w})^Tx+(b+\tilde{b})\right)\right\}
\end{aligned}\tag{11}$$

Where $\tilde{w},\tilde{b}$ can be computed from

$$\begin{aligned}
\tilde{b}^*,\tilde{w}^*=&\arg\max_{\tilde{b},\tilde{w}}\frac{1}{N}\sum_{l=1}^{N}\{y_l\left((w+\tilde{w})^T\mathrm{f}(x_l)+(b+\tilde{b})\right)-\frac{\lambda_b}{2N}\tilde{b}^2\\
&+\log P_{\text{prior}}\left(y|\left[\mathrm{f}^m(x_l)\right]_{m=1}^{M}\right)-\log\tilde{Z}\left(x_l\right)\}-\frac{\lambda_w}{2N}||\tilde{w}||_2^2
\end{aligned}\tag{12}$$

Since only new samples are involved in the above optimization problem, the model can be updated much more efficiently.

III. Experiment

In this section we present experiments by comparing the proposed Multimedia LEGO approach with the other state-of-the-art algorithms. We demonstrate how the proposed algorithm effectively model structured concepts using model warehouse, as well as enhance target concepts classification results.

A. Dataset

We conduct experiments on ImageNet dataset [15] in the "vehicle" domain. ImageNet is a realistic image database organized by WordNet [16][17] hierarchy. Each node in the hierarchy is representing a concept, and associated with a set of images. "Vehicle" is a relatively complex root concept in ontology, including a large number of different sub-genre categories. This "vehicle" specified ImageNet subset has first been used and released in [18]. Here we adopt the same dataset to show the competitive results of the proposed LEGO algorithm. This dataset contains 26,210 images, including 13,889 positive "vehicle" samples, and 12,321 negative samples. The vehicle ontology is illustrated in figure 3, which was also used in [18]. There are 20 concepts associated with root "vehicle", including a four-level ontological structure with 13 leaf nodes. Each of the leaf concept contains around 1,000 positive images. The negative samples include concepts such as "formation", "structure" and "sports", which contain tremendous low level visual ambiguities. The "parent-child" relationship indicate a "is-a" (also seen as a OR relation in PLOT) relationship in the WordNet taxonomy. For example, "plane" in the ontology shown in figure 3 contains "OR" relationship to its children, and "Not" relationships to other nodes in the same level.

B. Real World Problem Simulations

In our experiments, we simulate the real word scenario as we described in the introduction. We split all images to three disjoint sets randomly with 40%, 15%, and 45%. Then basic logistic regression models are trained using 40% of samples for the leaf nodes on the given hierarchical structure. After training, we only keep the weight vectors and use them as our "existing models" in the model warehouse. In such a way, we obtain a pool of rich semantic models for images. It is an analog to LSCOM and Mediamill 101, where all the samples used to generate the models in the pool are not accessible anymore. Actually, in real life, a large number of labeled training samples are not easy to acquire, especially when the domain of concepts expand rapidly. In order to take this fact into account, we randomly sample 15% of the data used as training images for LEGO as well as other compared methods which cannot acquire information from pre-trained models. The last 45% of data is used as testing samples to evaluate recognition performance for different approaches. All of the experiment results are reported as averaged over ten random runs.

C. Logic Operations and Error Propagation Analysis

In section II-A, we mentioned two different ways to represent Boolean logic probabilistically. One way is using Min/Max function, the other way is using T-norm/T-conorm. Moreover, we also illustrated how important of the small number of training samples to prevent error propagation from lower level to higher. Here we compare the classification results on proposed LEGO approach and a purely logical version without using any updating samples called P-LEGO. And the results is shown in tabel I.

Two main observations can be made from table I. First, both probabilistic logic operations provide comparable results on all different concepts regardless of how far these concepts are away from the "existing models". In general, "T-conorm" performs slightly better than "Max" in both LEGO and P-LEGO methods on all the eight different target concepts. Second, P-LEGO's accuracy decreases dramatically as the "target concept" comes higher in the ontology hierarchy. On the other hand, our proposed LEGO algorithm still has a reliable performance. The main reason is that the general procedure using logical operations on hierarchical structure depends on lower level concepts that provide information to learn "target concepts" at higher levels. Once the prior is obtained, model adaptation process will start. In the meantime the prior probabilities for its parent's nodes are also computed in the same manner. However, without model-updating scheme, higher level models cannot be reliably refined and it could result in errors accumulated through the entire hierarchical structure.

D. Performance Comparison

In previous section we have shown that only combining primitive concepts is insufficient due to error propagations. To evidently demonstrate the advantage of the proposed LEGO method for ontological categorizations, we compare experimental results with other state-of-the-art approaches. The comparison experiments strictly follow the protocol in section III-B, and the results are shown in table II. In [18] and [19], the authors proposed the best classifier using [19]′s feature was obtained by first applying Within Class Covariance Normalization (WCCN), and then using Nearest Neighbor or Nearest Central (WCCN+NN, and WCCN+NC). In our setup, both methods yield comparable results as flat multi-class SVM[3]. However, there is no straight way to integrate these approaches in the given ontology hierarchy. Therefore, we also conduct our experiment using a tree loss based hierarchical SVM proposed in [20] [2]. Among all, our proposed LEGO algorithm provides a significant gain over other methods, especially at the 3rd level of classification. This shows that the "existing models" in the model warehouse provide tremendous contributions to distinguishing more abstractive categories at higher level.

TABLE II
COMPARISON RESULT TO OTHER STATE-OF-THE-ARTS METHODS. THE BEST PERFORMANCE FOR EACH CATEGORY IS HIGHLIGHTED IN BOLD.

Category	2^{nd} level	3^{rd} level
Flat SVM[3]	84.41	76.55
WCCN+NN[19]	84.50	78.98
WCCN+NC[19]	85.74	64.36
H-SVM[20] [2]	86.26	78.10
Ours	**89.03**	**87.82**

IV. CONCLUSION

In this paper we developed a novel framework, LEGO, to seamlessly integrate both the new target training examples and the existing primitive concept models. LEGO treats the

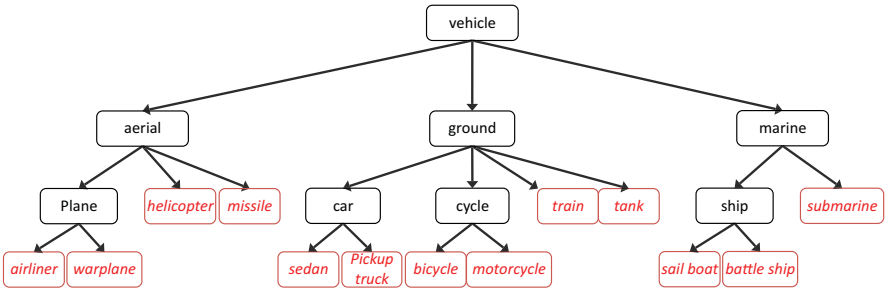

Fig. 3. Vehicle hierarchy from ImageNet. The red color notes indicate the leaf-node concepts, which can be obtained from the model warehouse.

TABLE I
HIERARCHIAL CLASSIFICATION RESULTS ON THE "TARGET CONCEPTS". THE BEST PERFORMANCE FOR EACH CATEGORY IS HIGHLIGHTED IN BOLD.

Level	Category	P-LEGO (Max)	P-LEGO (T-conorm)	LEGO (Max)	LEGO (T-conorm)
3^{rd}	plane	94.18	94.23	94.30	**94.33**
3^{rd}	car	95.33	95.41	95.99	**96.04**
3^{rd}	cycle	95.25	95.31	95.46	**95.51**
3^{rd}	ship	95.57	**95.58**	95.53	95.54
2^{nd}	aerial	91.36	91.50	92.32	**92.37**
2^{nd}	ground	89.45	89.86	93.00	**93.11**
2^{nd}	marine	93.80	93.90	94.31	**94.36**
1^{st}	vehicle	87.21	89.39	96.47	**96.94**

primitive concept models as "lego" toy to potentially construct an unlimited vocabulary of new concepts. We proposed a much flexible learning algorithm to efficiently combine the obtained probabilistic models with new information to clarify semantic ambiguity as well as correct the errors propagated from the nodes at lower level. The experiments over a real-world image data set demonstrated: 1) using logical operations combining individual concepts in the existing model warehouse provides us rich semantic resources to improve performance on more abstractive concepts at higher level of ontology hierarchy; 2) evolving higher level models by using a small number of examples could clarify the semantic ambiguities. Particularly, our proposed "T-conorm" LEGO approach has significant advantages over other state-of-the-art algorithms.

ACKNOWLEDGMENT

This work was funded in part to Shiyu Chang, Guo-Jun Qi and Thomas Huang by the ONR grant N00014-12-1-0122, National Science Foundation under Grant No. 1318971, and U.S. Army Research Laboratory and U.S. Army Research Office under grant number W911NF-09-1-0383. This work was supported in part to Dr. Qi Tian by ARO grant W911NF-12-1-0057, Faculty Research Awards by NEC Laboratories of America, and 2012 UTSA START-R Research Award respectively. This work was supported in part to Qi Tian also by National Science Foundation of China (NSFC) 61128007.

REFERENCES

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 1995.
[2] L. Cai and T. Hofmann, "Hierarchical document categorization with support vector machines," in *CIKM*, 2004, pp. 78–87.
[3] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.
[4] M. R. Naphade, J. R. Smith, J. Tesic, S.-F. Chang, W. H. Hsu, L. S. Kennedy, A. G. Hauptmann, and J. Curtis, "Large-scale concept ontology for multimedia," *IEEE MultiMedia*, vol. 13, no. 3, pp. 86–91, 2006.
[5] C. Snoek, M. Worring, J. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *ACM Multimedia*, 2006, pp. 421–430.
[6] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, and H.-J. Zhang, "Correlative multi-label video annotation," in *ACM Multimedia*, 2007, pp. 17–26.
[7] A. Natsev, A. Haubold, J. Tesic, L. Xie, and R. Yan, "Semantic concept-based query expansion and re-ranking for multimedia retrieval," in *ACM Multimedia*, 2007, pp. 991–1000.
[8] Y. Wu, B. L. Tseng, and J. R. Smith, "Ontology-based multi-classification learning for video concept detection," in *ICME*, 2004, pp. 1003–1006.
[9] J. Fan, Y. Gao, and H. Luo, "Hierarchical classification for automatic image annotation," in *SIGIR*, 2007, pp. 111–118.
[10] M. Marszalek and C. Schmid, "Semantic hierarchies for visual object recognition," in *CVPR*, 2007.
[11] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair, "Learning hierarchical similarity metrics," in *CVPR*, 2012, pp. 2280–2287.
[12] J. Zhou, J. Chen, and J. Ye, "Clustered multi-task learning via alternating structure optimization," in *Advances in Neural Information Processing Systems 24*, 2011, pp. 702–710.
[13] R. Cignoli, I. M. L. D'Ottaviano, and D. Mundici, Eds., *Algebraic Foundations of Many-valued Reasoning*, Dordrecht,Kluwer, 2000.
[14] S. F. Chen and R. Rosenfeld, "A gaussian prior for smoothing maximum entropy models," School of Computer Science, Carnegie Mellon University, Technical Report CMU-CS-98-108, 1999.
[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and F.-F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.
[16] C. Fellbaum, Ed., *WordNet An Electronic Lexical Database*. Cambridge, MA ; London: The MIT Press, May 1998. [Online]. Available: http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=8106
[17] G. A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
[18] M.-H. Tsai, S.-F. Tsai, and T. S. Huang, "Hierarchical image feature extraction and classification," in *ACM Multimedia*, 2010, pp. 1007–1010.
[19] X. Zhou, N. Cui, Z. Li, F. Liang, and T. S. Huang, "Hierarchical gaussianization for image classification," in *ICCV*, 2009, pp. 1971–1977.
[20] S. Bengio, J. Weston, and D. Grangier, "Label embedding trees for large multi-class tasks," in *NIPS*, 2010, pp. 163–171.