The Dissertation Committee for Sung Ju Hwang
certifies that this is the approved version of the following dissertation:

Discriminative Object Categorization with External Semantic Knowledge

Committee:

Kristen Grauman, Supervisor

Fei Sha

J. K. Aggarwal

Raymond Mooney

Pradeep Ravikumar

Discriminative Object Categorization with External Semantic Knowledge

by

Sung Ju Hwang, B.S., M.A.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN
August 2013

Dedicated to my mom Hyunsook Park

Acknowledgments

I wish to first thank my advisor Kristen Grauman, who always had great passion in research and insightful ideas in object recognition, and gave me much freedom to pursue any research ideas. She also led me to the direction of a better researcher with seriousness in every aspect, and incredible attention to details. All the accomplishments I have made would not have been possible without her, and I feel extremely lucky to be her student. I also want to thank greatly to my co-adviser Fei Sha, who guided me through the path of machine learning with rigor in mathematics and equally incredible attention to details as Kristen, and especially for his help on the methodology that made the ideas to come to life as working algorithms. I am also very grateful to my thesis committee members Professor Raymond Mooney, Professor J. K. Aggarwal, and Professor Pradeep Ravikumar for their insightful comments and suggestions.

I also want to thank my labmates with whom I spent the most time for the last five years. I want to thank Sudheendra for being an exemplary student to follow the step, Yong Jae for his availability to discuss even the vaguest idea and being a good friend, Jaechul for sharing his insight and experience in computer vision. Thanks Adriana for always smiling and being kind, and thank Chao-yeh with for being able to chat for anything, and helping me

out with courseworks. Thank Sunil for his jolliness, and thank Lu Zheng for sharing his experience on how to prepare for job search. Thank Dinesh for always staying until late with me and discussing on great ideas and Aron for being a good company in group outings.

My friends in Austin also deserve great thanks for not leaving me lonely on any days during my Ph.D. Thanks to Jong Wook Kim, who gave me a ride to school on my defense date when my car went out of battery, and also helped me many times with personal emergencies. Thank Eunho Yang for being a good friend for more than ten years, thank Yunshik Choi for great jokes and stories, and Jawook Huh for being a good exercise partner and helping me with anything, thank Jayoung Song for her company at hard times.

Special thanks to Songyi Lee, for her love and patient to wait for me through the long years, and finally, I want to thank my parents Buhyun hwang and Hyunsook Park for their unconditional love and support they have shown me over the last five years, as well as to my brother Sung Min Hwang for his love and care. I especially dedicate this thesis to my mother Hyunsook Park who is currently fighting with an ovarian cancer, and hope this accomplishment can giver her a light of hope to win over the long battle that could await her.

Discriminative Object Categorization with External Semantic Knowledge

Publication No. _____

Sung Ju Hwang, Ph.D.
The University of Texas at Austin, 2013

Supervisor: Kristen Grauman

Visual object category recognition is one of the most challenging problems in computer vision. Even assuming that we can obtain a near-perfect instance level representation with the advances in visual input devices and low-level vision techniques, object categorization still remains as a difficult problem because it requires drawing boundaries between instances in a continuous world, where the boundaries are solely defined by human conceptualization. Object categorization is essentially a perceptual process that takes place in a human-defined semantic space.

In this semantic space, the categories reside not in isolation, but in relation to others. Some categories are similar, grouped, or co-occur, and some are not. However, despite this semantic nature of object categorization, most of the today's automatic visual category recognition systems rely only on the category labels for training discriminative recognition with statistical machine

learning techniques. In many cases, this could result in the recognition model being misled into learning incorrect associations between visual features and the semantic labels, from essentially overfitting to training set biases. This limits the model's prediction power when new test instances are given.

Using semantic knowledge has great potential to benefit object category recognition. First, semantic knowledge could guide the training model to learn a correct association between visual features and the categories. Second, semantics provide much richer information beyond the membership information given by the labels, in the form of inter-category and category-attribute distances, relations, and structures. Finally, the semantic knowledge scales well as the relations between categories become larger with an increasing number of categories.

My goal in this thesis is to learn discriminative models for categorization that leverage semantic knowledge for object recognition, with a special focus on the semantic *relationships* among different categories and concepts. To this end, I explore three semantic sources, namely attributes, taxonomies, and analogies, and I show how to incorporate them into the original discriminative model as a form of structural *regularization*. In particular, for each form of semantic knowledge I present a *feature learning* approach that defines a semantic embedding to support the object categorization task. The regularization penalizes the models that deviate from the known structures according to the semantic knowledge provided.

The first semantic source I explore is attributes, which are human-

describable semantic characteristics of an instance. While the existing work treated them as mid-level features which did not introduce new information, I focus on their potential as a means to better guide the learning of object categories, by enforcing the object category classifiers to share features with attribute classifiers, in a multitask feature learning framework. This approach essentially discovers the common low-dimensional features that support predictions in both semantic spaces.

Then, I move on to the semantic taxonomy, which is another valuable source of semantic knowledge. The merging and splitting criteria for the categories on a taxonomy are human-defined, and I aim to exploit this implicit semantic knowledge. Specifically, I propose a tree of metrics (ToM) that learns metrics that capture granularity-specific similarities at different nodes of a given semantic taxonomy, and uses a regularizer to isolate granularity-specific disjoint features. This approach captures the intuition that the features used for the discrimination of the parent class should be different from the features used for the children classes. Such learned metrics can be used for hierarchical classification.

The use of a single taxonomy can be limited in that its structure is not optimal for hierarchical classification, and there may exist no single optimal semantic taxonomy that perfectly aligns with visual distributions. Thus, I next propose a way to overcome this limitation by leveraging *multiple* taxonomies as semantic sources to exploit, and combine the acquired complementary information across multiple semantic views and granularities. This allows us, for

example, to synthesize semantics from both 'Biological', and 'Appearance'-based taxonomies when learning the visual features.

Finally, as a further exploration of more complex semantic relations different from the previous two pairwise similarity-based models, I exploit *analogies*, which encode the relational similarities between two related pairs of categories. Specifically, I use analogies to regularize a discriminatively learned semantic embedding space for categorization, such that the displacements between the two category embeddings in both category pairs of the analogy are enforced to be the same. Such a constraint allows for a more confusing pair of categories to benefit from a clear separation in the matched pair of categories that share the same relation.

All of these methods are evaluated on challenging public datasets, and are shown to effectively improve the recognition accuracy over purely discriminative models, while also guiding the recognition to be more semantic to human perception. Further, the applications of the proposed methods are not limited to visual object categorization in computer vision, but they can be applied to any classification problems where there exists some domain knowledge about the relationships or structures between the classes. Possible applications of my methods outside the visual recognition domain include document classification in natural language processing, and gene-based animal or protein classification in computational biology.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Humans have the natural ability to categorize objects. Objects in the physical world are grouped into a category through the process of perception and recognition. The goal of an automatic object category recognition system is to implement the same ability on a machine.

Object categorization at the general level is different in nature from recognition at the instance level, for instance, from recognizing the category of concrete, homogeneous classes such as numbers or characters. In addition to the fundamental difficulties of visual recognition due to the difficulties of segmentation, variance in lighting and pose, clutter, and occlusion, there exists another, and more difficult problem of how to generalize over heterogeneous object instances. What makes us think of a *chihuahua* and a *dalmatian* as the same general object category *dog*?

A baby or a member of an isolated tribe who has never seen either of them may have no idea that the two animals belong to the same category at the first sight. Gradually, they might learn that the two animals are similar in some sense by first observing the characteristics of each instance, and identifying the similarities between the observed characteristics of each instance, but still, the

observation of the visual similarities is not sufficient to classify them into a same category. Only after telling them that the two animals belong to the same category *dog*, they can associate the general object category with the commonalities that they observe. These common traits could be appearance-based such as having some specific shape of the snout, or behavior-based, such as being friendly and loyal to humans.

Most current supervised learning-based automatic visual object category recognition systems work similarly, and use the category labels to learn the recognition models with statistical machine learning techniques. First, the features (characteristics) are extracted from an image, and are organized into an image descriptor that best describes the given image (object). Then, a decision function is learned to map the constructed descriptors to their category labels. The learned decision function can be later used for the category prediction of a novel test instance. Currently, discriminative learning approaches dominate the literature due to their strong empirical performance.

Discriminative approaches have shown much success in object recognition for many years. Earlier methods such as logistic classifier [77], boosted classifier [39, 106], and the neural network [46], have shown to be useful in visual object recognition for specific objects such as faces [106], and characters [39]. For more challenging problem of general object category recognition, kernel methods such as support vector machine (SVM) [23] have shown much success owing to kernel trick, which allows to find non-linear classification boundaries in the original space by learning linear classifiers in a high-

dimensional feature mapping space. The state of the art recognition results on challenging datasets such as Caltech-101 and Caltech-256 [49] are obtained by some of the kernel combination methods that learn both the classifiers and the optimal combination of the kernels, such as multiple kernel learning [103], or LP-Boost [42]. Latent SVM [36], a variant of SVM that models object parts as latent variables, holds state-of-the art results in object detection.

After the introduction of large-scale visual recognition datasets such as ImageNet [27], that involves the category recognition of nearly all existing general object categories, kernel methods became lackluster for their high computation and space overhead. Still, the state-of-the art results on these datasets are obtained from discriminative approaches, either by learning a low-dimensional embedding along with hierarchical classifier [11], or improving the input image descriptor by discriminatively learning mappings from each feature to codewords [117] while keeping the classifier relatively simple.

However, all of these are limited in that, the only information they leverage is that 'the instances that have the same category label are different from the others with different labels'. They view the object categories as independent, isolated entities that have no relation to others.

Some recent work treats the category space as interdependent—such as in structured output learning [99] and multitask learning [17], and such a structured output model have shown some success in object categorization [29]. However, important semantic information is still missing in these models.

In this thesis, I consider an important question: how can external semantic knowledge help better learn a discriminative recognition model for object categorization?

1.1 The need for semantic knowledge in object categorization

The most fundamental reason why external knowledge is critical in the understanding of objects in the category level, is that the categories are *semantic* entities defined and perceived by humans. As the correctness of the categorization depends on the *perceptual* similarity of the recognition result, performing object recognition on the semantic level is a more robust way. A purely statistical model that only utilizes the class label information could be misled into learning incorrect associations between visual features and the category. For example, suppose that the model wants to recognize the category *horse*, but all the images available are images of a horse jumping over a fence with a person riding on it. With only image-level labels provided, the model might learn to associate visual features describing people and fences to the category *horse*. However, with semantic knowledge, we know that the horse is a four-legged animal, with distinct physical features of the equine, which could be all utilized to correctly associate the visual features describing horses.

However, this is not the only possible advantage of using external semantic knowledge. Another advantage is that we can access much richer knowledge about the world. We humans have good knowledge about the world we

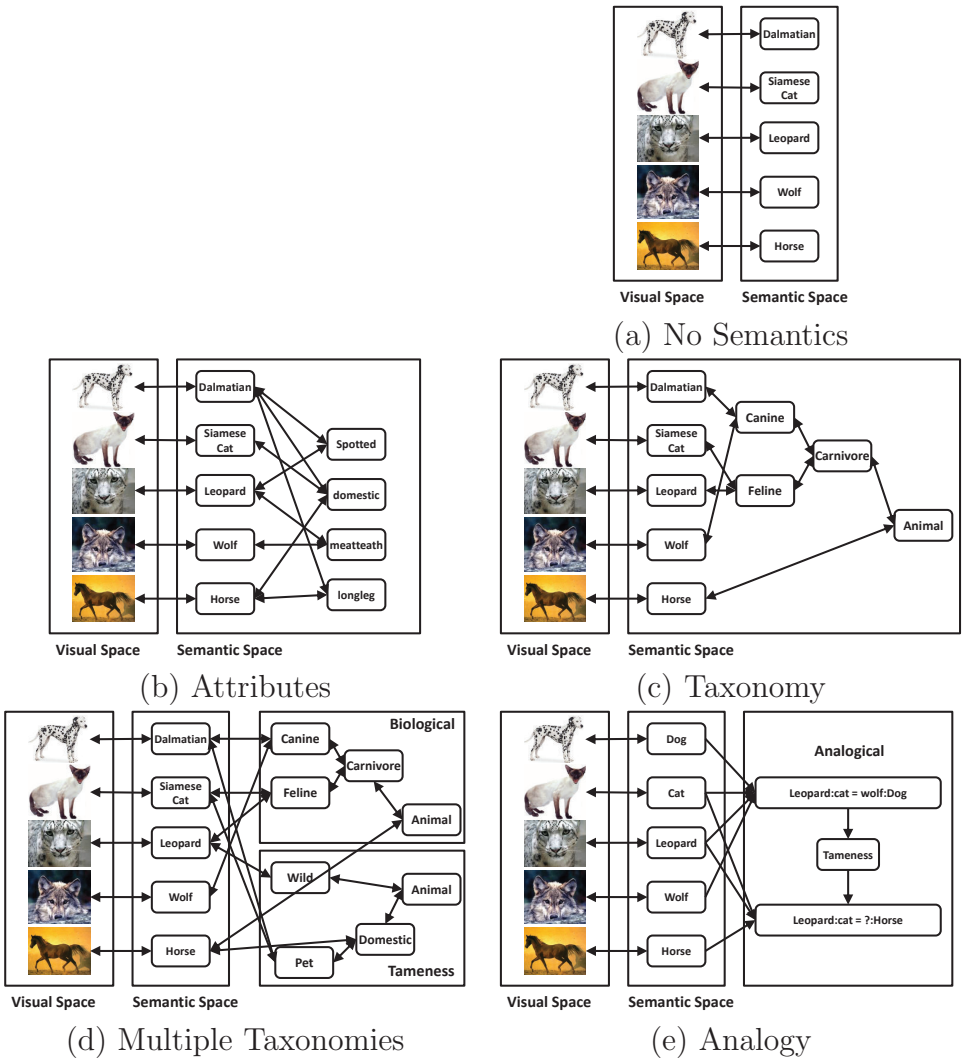
live in, and we can make use of our knowledge by associating the categories with the known concepts, unlike the traditional object recognition system that has to make decisions based only on the provided training examples. Suppose that we want the system to recognize the class *hawk*, but it has only seen them flying in the sky. Then, how would it recognize a hawk in a close distance? The external knowledge about the category *hawk* provides much information that is not present in the training set. We know that a hawk is a bird, a bird has feathers, predator birds have strong beaks, and associate the visual input to these known concepts, to recognize this animal we have never seen as a hawk. This is possible because while the categories are discrete concepts, the human semantic space they exist in is a continuous, interdependent space, where each object category does not exist in isolation, but in relation to others. Thus, an object category can be associated with other categories and semantic concepts, whether they are observed in the training set or not.

Finally, relational semantic knowledge scales with the number of categories. This is the opposite situation to visual-only statistical models, for which having a larger number of categories only means more confusions. The conventional non-semantic categorization models have shown some success on small scale datasets, as each object category is visually distinct, and there is less information between the classes. (Consider a dataset consisting of four classes, car, pedestrian, monitor and keyboard). However, as the size of the dataset grows larger and the categories become more fine-grained, the categorization problem becomes more difficult as the visual space becomes more dense and

crowded, and there exists more overlap in the visual feature space between the categories. For example, categorizing different subspecies of birds [116] could be difficult as all birds have beaks and wings. Yet, this densely populated feature space is beneficial with semantic knowledge leveraged, as it means having more instances for higher-level concept learning, and being able to identify the similarities and differences more clearly.

For example, suppose where we want to distinguish an *otter* from a *beaver*. They are visually very confusing and if we do not know where to focus, the classification of the two categories is difficult. Suppose, however, that we are given new categories *weasel* and *hamster*, as well as knowledge that *otters* and *weasels* are both *musteline mammals*, and *beaver* and *hamster* are both *rodents*. This gives us a critical hint on where to focus by the identified common features between the categories grouped as the same—the distinct body shape of the *musteline* (long and sleek body) and the *rodent* (short body), rather than the background, pose, and many others. Further, assume that all object categories are related to each other. Then the set of all categories will form a fully connected graph — adding a category will introduce the same number of linkage to the number of existing categories. The number of linkage—where the relational information lies—between the categories then will grow in $O(\mathsf{C}^2)$ where C is the number of categories, which can all contribute to better discrimination.

To recap, the benefit of using external semantic knowledge in object recognition, as opposed to the traditional vision-only model, is threefold.



(a) No Semantics

(b) Attributes

(c) Taxonomy

(d) Multiple Taxonomies

(e) Analogy

Figure 1.1: Various semantic models for object categorization. (a) Traditional recognition model treats each object categories as isolated, independent entities that have no relation among themselves. (b)-(e), The proposed semantic models relate categories and other semantic concepts in the semantic space.

First, semantics help learn correct associations between the visual features and the category membership. Second, the external semantic knowledge enables to associate unobserved concepts that are crucial in the understanding and the characterization of the categories to the observed. Third, a semantic aware method can benefit from semantic relationships—such that increasing number of categories would introduce more relationships for better learning, in contrast to the traditional model which suffers from more confusion.

Overall, we can utilize the mass of knowledge about the known world—as the semantic world is a continuous, interdependent space, the knowledge could be exploited from, or transferred through their relations. Traditional vision-only recognition model, on the contrary, is confined to the use of only the instances provided for training.

The goal of this thesis is to explore how to exploit this *external* semantic knowledge, to learn discriminative models for visual object recognition in the object category level.

1.2 Learning discriminative object recognition models with semantic regularization

In this section, I will give an overview of the entire thesis, while addressing *what* semantic knowledge to use, and *how* to incorporate them into the learning of discriminative categorization models. I will first start by explaining how to incorporate general semantic knowledge into a discriminative learning framework.

The approach I take in leveraging the semantics in learning is a structural regularization method [122, 61]. I introduce a regularization term that penalizes learning models that deviate from known structures defined by the given type of semantic knowledge, to augment the discriminative learning objective. This allows to leverage the power of existing discriminative learning methods while also learning semantically meaningful models that conform to human knowledge about the world; thus, we will be able to obtain a model that is discriminative yet semantic.

First, let us formally define the learning problem for object categorization. Given N training instances composed of descriptor-label pairs, $D = \{(\boldsymbol{x}_i, y_i)\}_{n=1}^{N}$, where $\boldsymbol{x} \in R^n$ is the image descriptor (or features) describing the i-th visual instance, and category labels $y_i \in \{1, \ldots, \mathsf{C}\}$ where C is the maximum number of categories, the learning objective for each category model j is to learn the parameter $\boldsymbol{w}_j$ for the label prediction function $f(\boldsymbol{x}, \boldsymbol{w}_j)$, whose optimal value can be obtained by minimizing the classification loss $\ell(\boldsymbol{x}_i, y_i, \boldsymbol{w}_j)$ for each instance i defined by $f(\boldsymbol{x}, \boldsymbol{w}_j)$ over all N training instances. The following shows a generic form of this categorization model learning problem.

$$\underset{\{\boldsymbol{w}_j\}}{\text{minimize}} \sum_{i}^{N} \sum_{j}^{C} \ell(\boldsymbol{x}_i, y_i, \boldsymbol{w}_j) \tag{1.1}$$

As aforementioned, this does not impose any relations between each independent categorization model $\boldsymbol{w}_j$, and thus ignores vast human knowledge to relate and group categories. The regularized discriminative learning model

I employ for imposing 'semantics' to this model has the following problem formulation:

$$\underset{\{\boldsymbol{w}_j\},\phi}{\text{minimize}} \sum_{i}^{N} \sum_{j}^{C} \ell(\phi(\boldsymbol{x}_i), y_i, \boldsymbol{w}_j) + \lambda\Omega(\{\boldsymbol{w}_j\}) \tag{1.2}$$

The above differs from the basic categorization model learning problem in Equation 1.1 in two aspects. 1) It contains a transformation $\phi(\boldsymbol{x})$, which in most cases is learned alongside the classifier parameter $\boldsymbol{w}$, that will transform the instances in a low-level input feature space to a higher-level common semantic space where the categories are associated to one another. 2) The categorization model learning is regularized with a semantic structural regularizer $\Omega(\{\boldsymbol{w}_j\})$ on the set of parameters $\{\boldsymbol{w}_j\}$, where λ balances its effect with the classification loss.

The desired outcomes of this regularized learning are discriminative categorization models that minimize both the classification loss and penalty defined on prior knowledge, as well as new features $\phi(x)$ from the learned transformation ϕ. Due to the second aspect where the features are learned as by-products of the categorization model learning, my methods can be also viewed as feature learning methods. While the learned features are optimized for the specific categorization model learned, they could be also treated as stand-alone features, and can be used for tasks other than object categorization, such as matching or retrieval.

The key component in this model is the regularization term $\Omega(\{\boldsymbol{w}_j\})$ that provides structural constraints to the learned models and also to a learned transformation $\phi(\boldsymbol{x})$, which vary depending on the specific type of the semantic knowledge provided. Then, what kind of semantic knowledge is available for us to exploit? The semantic knowledge can come in various forms. The form could be either *fixed* such as groupings of the categories or *arbitrary* as in natural language descriptions. In this thesis, I specifically exploit the types of semantic knowledge that have fixed forms; that is, the structural constraints from the models are consistent throughout different semantic instances.

I focus on semantic sources to augment the information provided with the surface category labels. The first of these semantic sources is *attributes* (Figure 1.1 (b)), which are semantic concepts that are shared by different object categories. They are general concepts which can span through different categories or instances, such as *black*, *longleg*, *fast*, or *has wheels*. The second semantic source is a *taxonomy* (Figure 1.1 (c)) which groups leaf-level classes into hierarchically inclusive groups. Further, as there exists no single taxonomy that is optimal, since the semantic relations among the categories differ for each semantic perspective, we consider semantic taxonomies in multiple semantic views (Figure 1.1,(d)). The last type of semantic knowledge visited in this thesis is an *analogy* (Figure 1.1 (e)), which captures high-level relational similarities between two *pairs* of categories with the equality constraint.

Figure 1.2 shows the overview of this thesis. I allocate separate chapters for four pieces of work that have been published to major conferences [58, 55,

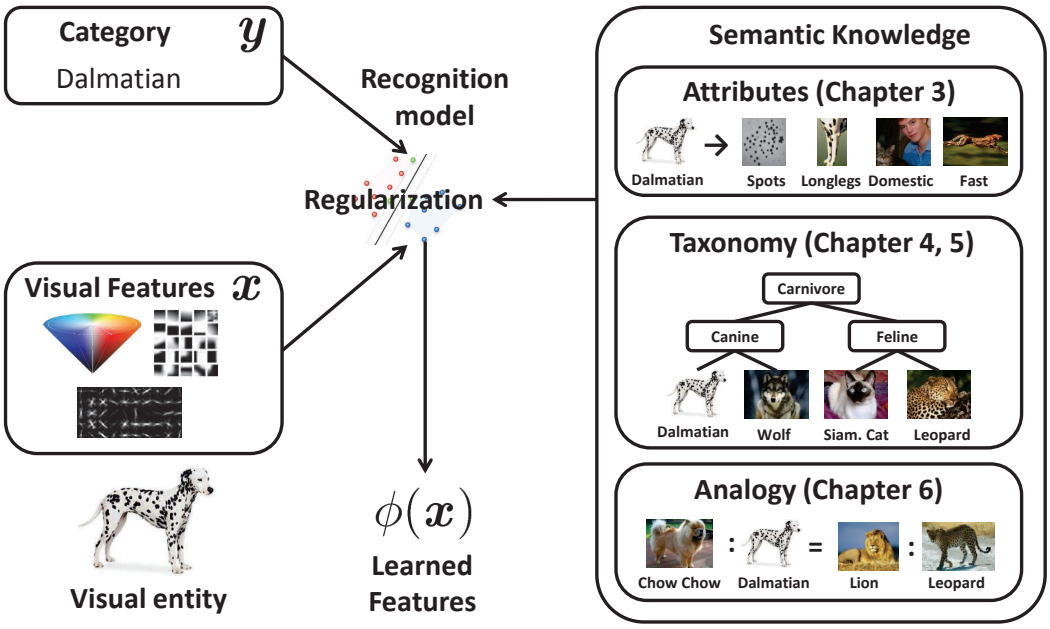


Figure 1.2: The overview of the thesis work.

56, 57]. Each chapter shows how to exploit each type of semantic knowledge to regularize a specific type of discriminative categorization model for improved object categorization performance.

The validation of the proposed methods' categorization performance on several categorization datasets that include different types of categories such as animal [65], scenes [79], and general objects [27], show that these different types of semantic knowledge are indeed helpful in achieving better classification performance over the state-of-the art discriminative learning methods. Thus, the proposed methods can potentially be adopted to any visual recognition systems where such discriminative learning methods are used internally, to

improve upon their performance. The only requirement in using the proposed models is the provision of some domain knowledge on the set of categories. Such domain knowledge is usually inexpensive to obtain compared to per-instance labels, as it requires defining the models on the set of categories, which is not affected with the number of training instances. Also, semantic sources such as attributes and taxonomies are abundant at least for general object categories, further minimizing additional human effort.

In the next subsections, I will give a brief preview of each chapter.

1.2.1 Leveraging attributes to guide feature learning

The first type of semantic knowledge I exploit is semantic attributes. An attribute is a human describable property of an object, that is either visual such as *spots* and *longleg*, or semantic as *domestic* and *fast*. In the original work of [65, 34] where semantic attributes are introduced and in most of the follow-up works [63, 102, 109, 12, 86], attributes are mostly treated as mid-level features that bridge the lower-level visual features and high-level classes, and each attribute model is independently trained. However, this separation of object class (category) classifiers and attribute classifier training does not consider the fact that the object classifiers and attribute classifiers are trained on the same set of visual features, and are inherently related to each other. I instead propose to use attributes as a means to relate different object categories through learning a common, low-dimensional representation that is shared between object and attribute classifiers.

The learning of the shared features between object and attributes classifiers is achieved through group sparsity regularization. The (2,1)-norm regularizer favors shared weights, by enforcing grouping between different classifiers with the ℓ_2-norm regularization, and sparse feature selection with the ℓ_1-norm regularization within the same classifier. The resulting regularized model learns a feature space that is more semantically meaningful and achieves significant improvements over two challenging datasets of animals and outdoor scenes.

1.2.2 Learning disjoint features on a taxonomy

Then, I move my attention to the second form of semantic knowledge, *taxonomy*. A taxonomy a human-defined hierarchical grouping of object categories, and popular examples are Wordnet [35], and the phylogenetic tree of life. Most previous work using semantic taxonomies focused either on its hierarchical structure that enables efficient classification [72, 50, 11], or on the explicit semantic information such as tree-hop distances between the classes [113, 37]. Instead, I focus on information implicitly provided from the parent-child relationships, specifically, the intuition that the features used to characterize the parent-level category should be different from the features used to characterize its children. For example, a wheel-shaped patch is useful when discriminating between a *ship* and a *wheeled vehicle*, but is not useful when discriminating between *bicycle*, *car*, and *motocycle*. The objective here is to focus only on the features that are useful for the discrimination of the

categories at a specific semantic granularity. To achieve this goal, we learn metrics for each node of the taxonomy, and then perform disjoint regularization between the metrics. We call this method *tree of metrics* (ToM).

I propose a novel disjoint regularizer that requires the metrics at a node and its children to compete for features, by minimizing the ℓ_2-norm of the sum of the diagonals of two metrics, as it prevents the two metrics from having high value for the same feature dimension. The competition results in the isolation of the features that are discriminative for each semantic granularity. The proposed method is evaluated on two challenging datasets containing animals and vehicles. The resulting ToM model achieves better classification accuracy with k-nearest neighbor method, compared to a single metric model or flat multi-metric models. Also, the model with the proposed disjoint regularizer outperforms non-regularized models.

1.2.3 Combining complementary information from multiple taxonomies

I further extend the scope of the external semantic sources to contain *multiple* semantic views represented by the semantic taxonomies. The motivation of the idea is that there exists no single optimal taxonomy, as the utility of the taxonomy depends on each task and view. For example, the taxonomy defined on the biological origin would group the class *dog* and *wolf* into the same superclass differentiated from the superclass containing *cat* and *leopard*, while the taxonomy defined on tameness would group the classes *dog* and *cat*

as the same. The idea is to exploit such complementary information present in these taxonomies, to learn a better (combined) semantic representation.

To this end, I propose *semantic kernel forests*, which capture semantic similarities between instances at different views and different semantic granularities, and use multiple kernel learning (MKL) to learn the optimal combination of these feature spaces. In addition to the usual ℓ_1-norm regularizer for MKL to select only the useful kernels, I introduce a hierarchical regularizer based on the hinge loss, to favor upper-level metrics to select kernels that capture more high-level semantic differences. The resulting regularized MKL model outperforms the single kernel SVM, non-semantic MKL, perturbed taxonomy and single taxonomy MKL baselines, and the added hierarchical regularizer results in improved classification accuracy.

1.2.4 Transferring knowledge between related category pairs with analogies

Finally, I explore a new type of semantic knowledge, analogies. While analogies have been explored to some extent in psychology and artificial intelligence [44, 45, 74, 75, 108], no prior work exploits them for categorization. Analogies provide the relational similarities between two pairs of categories. For example, in the analogy *lion:tiger = horse:zebra*, the common relationship would be that the latter is the striped version of the former, without the mane. I show how such a relational similarity can be interpreted into a geometrical constraint in a hypothetical category space, such that the difference between

the first pair of categories should be the same as the difference between the second pair of categories. This equality constraint will result in a more confused pair of categories benefiting from well-separated categories that share the same relationship. I encode this into a regularization term to regularize the geometry of the discriminatively learned category embedding space. The resulting analogy-preserving semantic embedding (ASE) outperforms the embedding that is discriminatively learned without any semantics or learned only with class-similarity constraints encoded as distances. ASE also outperforms others on the analogy completion task, where the task is to predict the object class that sensibly completes an analogy based on the three given classes: *p:q = r:?*.

In the next chapter, I will describe the related work in two perspectives of how to utilize each type of semantic knowledge for object categorization, and how to augment the learning methods to incorporate the obtained semantic information. In later chapters, I will go over each method and also will describe possible future research directions in the context of semantic approaches for object categorization.

Chapter 2

Related Work

My thesis work tackles two main issues. The first is what semantic knowledge to use and in what sense, and the second is how to incorporate the learned semantics in learning of a discriminative object recognition model. In this chapter, I will describe related work in these two perspectives: utilization of semantic knowledge in visual recognition, and discriminative learning methods for categorization.

2.1 Semantic knowledge in object categorization

External semantics beyond object class labels are rarely used in today's object recognition systems, but recent work has begun to investigate new ways to integrate richer knowledge, such as *attributes* and *taxonomies*. My work introduced in the next three chapters focuses on exploiting these two types of semantic knowledge.

2.1.1 Attributes in visual recognition

Attributes are human describable characteristics of an instance, which could be either visual or semantic [65, 34, 38]. Recent work shows that at-

tributes are useful in a variety of settings. First, they are independently useful to describe familiar and unfamiliar things (e.g., the leopard is *spotted* and *furry*, whether or not we know to call it a leopard [34, 38]), or to search through large image/video collections in semantic terms [102]. Second, they enable new *zero-shot learning* paradigms, where one can build an object model on the fly [65]. Third, they can serve as mid-level features to an object classification layer; having learned to predict the presence of each attribute, one can build supervised object models on top of those predictions [63, 65, 34, 110]. Usually attribute-object associations are manually specified, but some work explores ways to obtain them automatically [83, 109, 12, 86]. Notably, nearly all models using attributes for recognition learn them independently.

On relating object and attributes, the "indirect attribute prediction" model [65] offers a way to regularize attribute predictions based on object predictions; however, the attribute-object connections are set by human-given definitions, and so the two are not jointly learned. The novel multiple instance learning (MIL) approach in [107] jointly trains attribute and object detectors with weakly labeled data, with a constraint that both models should agree on localization (e.g., if an image is tagged "blue cap", both MIL classifiers should prefer to select positive training instances from the same location). In contrast, in my work (Chapter 3), I use the attributes to influence the feature space construction, not training instance selection.

There is also some work that aims to use attributes to improve object classification performance. The method in [110] integrates attribute- and

object-based cues into a structured latent SVM model: the attribute labels are left as latent variables on the training data, and the objective is to minimize object prediction loss. In contrast, I show the value in discovering a single shared representation such that both attribute and object tasks can be predicted well. Thus, while [110] implicitly discovers object-attribute relationships, my work exploits the two simultaneously as explicit tasks. Doubly supervised latent Dirichlet allocation (DSLDA) [1], which is a recently proposed generative topic model that has both supervised attributes and latent shared features in the intermediate layer, is also highly relevant to my work. Such a hybrid supervised-latent intermediate layer model can benefit from both the explicit high-level semantic attributes as in [65] and learned shared latent features that account for (possibly) non-semantic high-level topics. However, DSLDA separates the latent shared feature learning from attributes, and does not infuse semantic knowledge from attributes into the shared feature learning as our model does. This limits its use as a feature learning method compared to ours, which can produce semantic, shared features as outputs.

2.1.2 Taxonomies for multiclass object classification

Hierarchical taxonomies have natural appeal for object categorization, and researchers have studied ways to discover such structure automatically [95, 10, 50, 69], or to integrate known structure to train classifiers at different levels [72, 124]. The emphasis is generally on saving prediction time (by traversing the tree from its root) or combining decisions, whereas we propose to influence

feature learning based on these semantics. While semantic structure need not always translate into helping visual feature selection, the correlation between WordNet semantics and visual confusions observed in [26] supports our use of the knowledge base in this work. The machine learning community has also long explored hierarchical classification (e.g., [62, 73, 16]). Of this work, our goals most relate to [62] which focus on a very small set of features at each node of a taxonomy, during the hierarchical classification process. However, our focus is on learning features discriminatively and biasing toward a disjoint feature set via regularization.

Most work in object recognition that leverages category hierarchy does so for the sake of efficient classification [72, 50, 11, 28, 41]. Making coarse to fine predictions along a tree of classifiers efficiently rules out unlikely classes at an early stage. Since taxonomies need not be ideal structures for this goal, recent work focuses on novel ways to optimize the tree structure itself [11, 28, 41], while others consider splits based on initial inter-class confusions [50]. A parallel line of work explores unsupervised discovery of hierarchies for image organization and browsing, from images alone [95, 10] or from images and tags [68]. Whereas all such work exploits tree structures to improve efficiency (whether in classification or browsing), my goal is for externally defined semantic hierarchies to enhance recognition accuracy.

More related to the problem setting tackled in this thesis are techniques that exploit the inter-class relationships in a taxonomy [71, 98, 37, 26, 105]. One idea is to combine the decisions of classifiers along the semantic hierar-

chy [71, 124]. Alternatively, the semantic "distance" between nodes can be used to penalize misclassifications more meaningfully [26], or to share labeled exemplars between similar classes [37]. Metric learning and feature selection can also benefit from an object hierarchy, either by using a taxonomy-induced loss for structured sparsity [61], or by sharing parameters between metrics along the same path [105].

My approaches to leveraging taxonomies (Chapter 4 and 5) are different from the existing work in that I mainly focus on the exploitation of the *implicit* information present in the parent-child relations, and learning a granularity-specific *feature* space based on it.

2.1.3 Analogies in recognition

Some existing work in cognitive science and AI has explored analogies in various contexts, different from my work in this thesis. Gentner et al. [44] study analogies in light of human cognition. They define an analogy as a relational similarity over two pairs of entities, and contrast it with the more superficial similarity defined by attributes. Based on this intuition, they suggest a conceptual structural mapping engine that enables analogical reasoning [45]. Recognizing that such generic analogies require high-level logical reasoning that may be problematic for an automated prediction system, Miclet et al. suggest focusing on the analogical dissimilarity between entities in the same semantic universe [74]. They exploit analogical dissimilarity to do direct logical inference when one of the entities is unknown. My work focuses on sim-

ilarly scoped analogies—the semantic universe of object categories. In contrast to their logical inference model, however, I propose geometric constraints to enforce analogical proportions in a learned embedding.

While my main idea is to use analogies in an embedding, I also show how to automatically discover categories that have analogical relationships using their attribute descriptions. In this respect, there is a connection to structural transfer learning work that discovers mappings between domains [75, 108]. However, while that work aims to associate distinct source and target domains (e.g., computer viruses and human viruses), we aim to detect parallel associations within the same domain, and then use those pairings to constrain feature learning.

In graphics, inferring the filter relating two input images allows the automatic creation of "image analogies" [53]; I deal with analogies on visual data, but my idea of using them to regularize the representation is different and original.

The idea of capturing higher-order relationships as vector differences in a semantic space and using a learned space to answer an analogy question in a recently published work [76] is similar to mine. However, my main objective is on improving object categorization performance rather than on predicting categories that form an analogy. Also, my method encodes the analogical relationships between category pairs explicitly into the learned semantic embedding space through regularization, while [76] does not present any means for such supervised learning for analogical relationships and soley rely on inherent

analogical relationships in the semantic space. Such an implicit unsupervised model could be less powerful even for the analogy completion task they are targeting.

2.1.4 Leveraging and combining information from multiple semantic views

Combining information from multiple "views" of data is a well-researched topic in the machine learning, multimedia, and computer vision communities. In *multi-view learning*, the training data typically consists of paired examples coming from different modalities—e.g., text and images, or speech and video; basic approaches include recovering the underlying shared latent space for both views [52, 68], bootstrapping classifiers formed independently per feature space [15, 21], or accounting for the view dependencies during clustering [30, 51]. When the classification tasks themselves are grouped, *multi-task learning* methods leverage the parallel tasks to regularize parameters learned for the individual classifiers or features (e.g., [5, 70, 58]).

Broadly speaking, the problem visited in Chapter 5 has a similar spirit to such settings, since we want to leverage multiple parallel taxonomies over the data; however, the goal of aggregating portions of the taxonomies during feature learning is quite distinct. More specifically, while previous methods attempt to find a single structure to accommodate both views, our method seeks complementary information from the semantic views and assembles task-specific discriminative features. The topic of multiple taxonomies was also

visited in [91], but their focus was on the construction of multiple taxonomies from the semantic attributes. In contrast, my focus is on exploiting predefined multiple taxonomies, where the end product is a single discriminative feature space targeted for categorization.

2.2 Discriminative learning methods and regularization

From the machine learning perspective, my proposed methods can be viewed as structural regularization methods in learning discriminative models. They build on several successful existing machine learning methods—namely multitask learning, metric learning, multiple kernel learning, and large margin embedding—and augment the models with semantic knowledge through the means of regularization. In this section, I give a brief overview on the backgrounds of these discriminative learning and regularization methods.

2.2.1 Multitask learning for learning the structures between tasks

Multitask learning refers to a class of methods that exploits the task structure among related classification tasks, to obtain better generalization ability. In the original work of [17] where multitask learning is first introduced, classifiers for different classification tasks were jointly learned by sharing the hidden units in the neural network, which are activated similarly positive for similar task outputs, and negatively for dissimilar task output. However, in general, we can refer to any method that can relate different classifiers together so that each classifier is affected by the others as multitask learning.

There are two predominant directions to pursue mutltiask learning: parameter sharing and feature sharing. Which sharing to use depends on the task. For example, for multitask learning with class classifiers and attributes, a plausible assumption is that there are invariant visual features tied to semantics, which both object classifiers and attribute classifiers use, thus rendering feature sharing as more sensible. For multiple kernel learning with taxonomies that assign weights to each node that are shared by different categories, parameter sharing would make more sense. One could differentiate different tasks as 'main', and 'auxiliary', depending on which task is the main target. For most object recognition methods, object category recognition is the main task, and different data and tasks are used as auxiliary, such as text [84, 70] or pattern matching [3]. My object-attribute feature sharing model is the first to explore multitask learning with attributes, which (relative to other sources of auxiliary tasks) has potential advantages of intrinsic task relevance and supervision "reuse". Furthermore, I focus on "disjoint" sharing for the disjoint visual feature learning with taxonomies where the learners compete for features rather than trying to share them.

2.2.2 Metric learning for learning discriminative features

Metric learning is an embedding method that learns the 'metric' space that preserves certain distances among the training instances. It has been a subject of extensive research in recent years, in both vision and learning. Good visual metrics can be trained with boosting [92, 6], feature weight learn-

ing [40], or Mahalanobis metric learning methods [64, 59, 111]. An array of Mahalanobis metric learners has been developed in the machine learning literature [47, 25, 112]. In my work of Tree of Metrics [55] (Chapter 4), I learn a discriminative local metric at each node on a taxonomy.

The idea of using multiple "local" metrics to cover a complex feature space is not new [114, 85, 111, 20]; however, in contrast to ToM, existing methods resort to clustering or (flat) class labels to determine the partitioning of training instances to metrics. Most methods treat the partitioning and metric learning processes separately, but some recent work integrates the grouping directly into the learning objective [6], or trains multiple metrics jointly across tasks [82]. No previous work explores mapping the semantic hierarchy to a ToM, nor couples metrics across the hierarchy levels, as we propose. To show the impact, in experiments in Chapter 4, we directly compare to a state-of-the-art approach for learning multiple metrics.

Previous metric learning work integrates feature learning and selection via a regularizer for sparsity [119], as I exploit for the ToM approach here. However, whereas prior work targets sparsity in the linear transformed space, ours targets sparsity in the original feature space, and, most importantly, also includes a disjoint sparsity regularizer. The advantage in doing so is that our learner will be able to return both discriminative and interpretable feature dimensions, as we demonstrate in our results. Transformed feature spaces—while suitably flexible if only discriminative power is desired—add layers that complicate interpretability, not only to models for individual classifiers but

also (more seriously) to tease apart patterns across related categories (such as parent-child).

2.2.3 Learning to combine features with multiple kernel learning

The support vector machine has shown much success in recent years in many applications, including object recognition, thanks to the kernel trick that enables learning of non-linear class boundaries by first transforming the points in the original feature space to a high-dimensional space using some function and learning a linear classifier in the resulting space [101]. While we use the term 'high' dimensional space, most of the kernel methods actually operate in the Hilbert space that preserves similarities between training instances. This trait is also advantageous as it provides the flexibility as to how to compute the similarities. One kernel (matrix) could be computed based on similarities in the contour shape, and another kernel could be computed based on the similarities in color. Then, the problem arises on how to combine the kernels so that the combined kernel would optimally capture similarities in the category space. The simplest way is to just average them. Or, the combination weights could be learned by cross-validation. Multiple kernel learning [8], was originally proposed as the extension of the kernel-based support vector machine to solve the kernel combination problem, by simultaneously learning the classifier and the kernel combination, and it has shown much success in visual object recognition [104, 42].

The predominant direction in the research of multiple kernel learning

in machine learning has been on exploring the ways to efficiently optimize the original additive kernels. How to generate the base kernels for combination has been mostly a secondary issue. For effective combination, finding a non-linear kernel combination has shown some progress in recent years, such as product of kernels [104], polynomial kernels [22], and Hadamard product of kernels [66]. Still, how to generate the kernels remains as a domain-specific application problem. Most kernels are generated by differentiating the parameters for the radial basis function kernels, or computing on different features. The proposed semantic kernel forest (Chapter 5) also employs a form of MKL, but rather than pool kernels stemming from different low-level features or kernel hyperparameters, it pools kernels stemming from different semantic sources. Furthermore, it adds a novel regularizer that exploits the hierarchical structure from which the kernels originate.

2.2.4 Embedding and manifold learning for object categorization

The analogy-preserving semantic embedding (ASE) I propose in Chapter 6 is an instance of an embedding method whose objective is to learn a representation that preserves certain topologies or properties in the original topological object. Most existing embedding methods aim to preserve the distances between data points, either globally [32] or locally [87, 115]. Label embeddings learned for object or document categorization also aim to preserve distances, but with further constraints to promote the discriminability of labeled classes [113]. Recent embedding methods preserve not only the ge-

ometry of local neighborhoods, but also higher-order properties like category clusters [94] or graph structure [93]. In my analogy-based embedding method, I also aim to preserve more far-reaching structures. However, my method is distinct in that it enforces the *relative* distances between semantically related pairs of instances.

2.2.5 Feature selection with regularization

Identifying and using 'good' features is critical to the robustness of a classification model, and there has been extensive work in this direction in machine learning. Regularization is a term for a general technique in statistical machine learning to introduce additional constraints, or in other words, 'penalty' terms, in the learning model to avoid overfitting to the bias in the training sample [97, 90]. A popular regularization method for learning classification or regression model is a sparsity-inducing norm regularization, that enables to select features. Lasso [97] uses ℓ_1-norm penalty term to favor sparse solutions for the training of classifiers or regressors. This enables to select features that are more useful and suppress noisy terms, resulting in a robust classifier that better generalizes. Ridge regression [90] regularizes the coefficient of the model using ℓ_2-norm, suppressing the coefficient from growing to infinite. It cannot zero-out the parameters to mathematical zeros as lasso does, but can correlate feature dimensions by shrinking correlated features simultaneously.

The elastic net [123] uses a convex combination of both ℓ_1- and ℓ_2-

penalties, resulting in sparse solutions while also shrinking correlated factors at the same time. This is called 'group sparsity', and further explored in the mixed-norm regularization. A group lasso performs ℓ_2-regularization along the feature dimension, and performs ℓ_1-regularization of these ℓ_2-regularized groups. This results in group sparsity, which makes correlated features drop out together. In my multtiask learning method with semantic attributes, we use this $(2, 1)$-norm as the objective (while solving the alternative problem that is convex).

Most group-sparsity regularization works by promoting sharing among the different learners. However, in some scenarios, making each learner to compete instead of share could be beneficial. Exclusive lasso [122], aims to minimize the ℓ_2-norm of each dimension of the classifiers that are ℓ_1-regularized (lasso), making each classifier to compete for a feature dimension. The disjoint regularizer used for the tree of metrics shares the same spirit, but promotes competition between two metrics instead of two classifiers.

Taxonomy-based regularization also has gained some limited attention recently. Tree-guided group lasso [61] uses the ℓ_2-norm to identify shared parts, and ℓ_1-norm regularization to obtain sparse selection of its children. Orthogonal transfer [121] leverages the intuition that classification among subcategories should not consider the factors that are already considered at upper levels, by constraining the parent and children classifiers to be orthogonal to each other. ToM is based on the same intuition but targets metric learning, and enables true selection of features using sparsity regularization and a dis-

joint regularizer that minimizes the ℓ_2-norm of the diagonal. The proposed semantic kernel forest also introduces a structured regularization is based on the intuition that higher level classification should be considered as more important (as it is tied to more number of lower-level classification problems), which is implemented into a hinge-loss regularizer.

The main novelty of my work in the machine learning, is in showing how to translate the *abstract* external domain knowledge into *concrete* structural constraints between classifiers, that sum up to regularizers to augment the discriminative learning objective, to learn discriminative yet semantic models (and features). This process is domain-agnostic as the requirement is only on the *structures* of the knowledge. Thus not just visual recognition models, but any classification models where such specific type of domain knowledge are available, can benefit from my method; the augmented model will enable leveraging the power of the existing discriminative classification learning algorithms while also utilizing the vast and complex domain knowledge that will guide the learning into a more correct direction. They will also less overfit to the training set biases compared to purely statistical approaches that rely only on the labels, which will result in improved accuracy from better generalization.

Chapter 3

Leveraging Attributes to Guide Feature Learning

The first semantic source I explore is semantic *attributes*. Attributes are human-understandable properties shared among object categories (e.g., *glassy*, *has legs*), and they are a compelling way to introduce high-level semantic knowledge into predictive models. As discussed in the previous chapter, recent work shows that attributes are valuable in several interesting scenarios, ranging from description of generic images or unfamiliar objects [38, 34, 102], to zero-shot transfer learning [65], to intermediate features that aid in distinguishing people, objects, and scenes [63, 65, 34, 110].

Existing approaches to attribute-based recognition assume that the attributes' role is primarily to focus learning effort on properties that will be reusable for many categories of interest, and to elegantly integrate human knowledge into discriminative models. As such, attribute classifiers are learned independently from object classifiers, and then their predictions are treated as "mid-level" features that bridge low-level image features and high-level object classes. However, segregating supervision about attributes from supervision about objects may restrict their impact. In particular, in conventional mod-

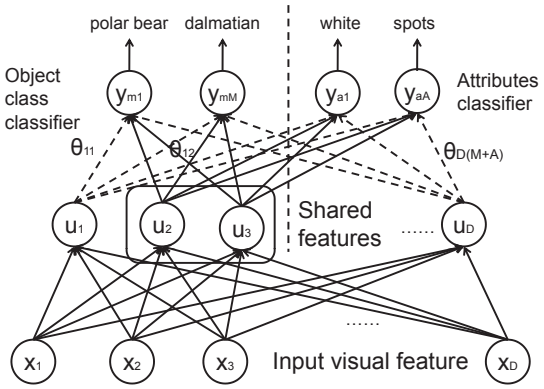


Figure 3.1: In my object-attribute feature sharing model, object categories and their human-defined visual attributes *share* a lower-dimensional representation (dashed lines indicate zero-valued connections), thereby allowing the attribute-level supervision to regularize the learned object models.

els, even though attributes influence object predictions, the attribute-labeled training data does not directly introduce *new* information when discriminatively learning the objects.

I explore how learning visual attributes *in concert* with object categories can strengthen recognition. The assumption is that both types of prediction tasks rely on some shared structure in the original image descriptor space. In other words, patterns among those generic visual properties that humans elect to name may reveal information about which low-level cues are valuable to object recognition—in the most general case, whether the objects of interest exhibit those attributes or not. Thus, rather than treat attributes as intermediate features, I propose an approach to discover this structure and learn a shared lower-dimensional representation amenable to discriminative

models for either one (see Figure 6.1)[1]. In effect, I show how human-defined semantics (as revealed by attributes) can regularize training for object classifiers.

Given a low-level visual feature space together with attribute- and object-labeled image data, my method learns a feature subspace for all labeling tasks based on a joint loss function that favors common sparsity. The optimization process alternates between regularizing towards shared features, and retraining task-specific classifiers based on those features. Our technique directly builds on a multi-task feature learning algorithm developed in [2], where it was applied to collaborative filtering of consumer data. To improve its scalability, we provide a more efficient kernelized implementation and linear algebra shortcuts for dealing with large matrices. Additionally, while in [2] all tasks are assumed to have the same label space, our setting entails non-overlapping label spaces (attributes, objects), for which feature-sharing is expected to be more challenging.

It is well-known that the success of multi-task learning or feature sharing hinges on the assumption that the input tasks are indeed related. Why should the assumption hold in our case? What makes attributes "special" as auxiliary tasks for object learning? Intuitively, their relation is intrinsic, since attributes are by definition shared among object categories. Many object-level distinctions can be made using a vocabulary of relevant properties, suggest-

[1]The work introduced in this chapter was published in [58].

ing that a representation sufficient to distinguish the properties would also be relevant for the objects (e.g., a child learning to discriminate cows from other animals might focus on the visual properties a cow exclusively has but other animals do not). In fact, in early visual processing, it is known that the human visual system discovers some sparse coding using a feature "vocabulary" of low-level filters [80].

More abstractly, we expect that structure among a wide span of attribute classifiers could reveal information about which low-level features are valuable to human understanding of the visual world. That is, even attributes that are not relevant to distinguishing a particular object may still help to constrain the space of image descriptors suitable for higher-level recognition problems. Finally, there is a practical incentive for treating attributes as auxiliary tasks regarding supervision cost: for many attributes, knowing the real world object-attribute relationship is sufficient to transfer object-level image labels to attribute-level labels (i.e., all buildings are manmade, so if we have a labeled image of a building, it is also an image of the manmade attribute).[2]

In short, the contribution of this chapter is threefold: 1) design of a method for feature sharing between object and attribute prediction tasks; 2) validation of the method's effectiveness with experiments on two datasets that feature sharing can offer noted improvements in accuracy for target object categorization tasks; and 3) exploration to what extent different attributes are

[2]This is the case for many binary attributes, but of course not all attributes (e.g., some bicycles are red, some are blue).

useful for a target task, and provide some initial ideas for automatic selection of relevant attributes to limit training costs.

3.1 Approach

I describe in detail the approach we take to learn shared features between objects and their attributes. My work directly builds on a previous approach [2]. Being mindful of desired large-scale learning settings, however, we extend the method by providing faster and more scalable numerical techniques. Additionally, we adapt the models to handle classification tasks where the label sets are disparate.

I start by describing the basic setup for learning features from multiple tasks, and then explain how the problem can be cast as convex optimization for both linear and kernel classifiers. Finally, I discuss extensions and improvements I have developed in order to apply the approach.

3.1.1 Basic setup and notation

There are two groups of classification tasks. We aim to improve *object* classification accuracy; thus, we refer to the objects as the *main* task, and the attribute classifiers as *auxiliary* tasks. Note that the two groups have different sets of labels.

We use multi-class support vector machines (SVMs) for the main task [24]. Let M denote the number of object classes, $\boldsymbol{x}_n \in \mathbb{R}^{\mathsf{D}}$ denote the n-th feature vector in the training data and y_n its class label. The multi-class SVM has M

parameter vectors $\{\boldsymbol{w}_m^*\}_{m=1}^{\mathsf{M}}$, one for each class. In the most basic setting, we consider linear discriminants which are parameterized by $\boldsymbol{w}_m \in \mathbb{R}^{\mathsf{D}}$. Let $\boldsymbol{W}$ denote the matrix whose columns are $\boldsymbol{w}_m$. To identify $\boldsymbol{W}$, we minimize a loss function that maximizes the discriminant $\boldsymbol{w}_{y_n}^{\mathrm{T}}\boldsymbol{x}_n$,

$$\boldsymbol{W}^* = \arg\min \sum_n \ell(\{\boldsymbol{w}_m^{\mathrm{T}}\boldsymbol{x}_n\}_{m=1}^{\mathsf{M}}, y_n) + \gamma \sum_m \|\boldsymbol{w}_m\|_2^2$$

where $\gamma \geq 0$ is a tradeoff parameter that regularizes the model complexity, using the parameter's 2-norm.

For learning A auxiliary tasks, we use y_{na} to denote the label for the a-th auxiliary task and $\boldsymbol{w}_a$ for the corresponding model parameter. Our auxiliary tasks are binary classification of attributes. We use the squared hinge loss for these tasks. For simplicity, the notation assumes that both the main task and auxiliary tasks are trained on the same feature vectors. However, this is not mandatory, as we demonstrate in our results.

We use t ranging from 1 to $\mathsf{T} = (\mathsf{M} + \mathsf{A})$ to index all parameter vectors for the main and auxiliary tasks. To avoid unnecessary notation clutter, with a slight abuse, we use $\sum_{t=1}^{\mathsf{M}} \ell(\boldsymbol{w}_t^{\mathrm{T}}\boldsymbol{x}_n, y_{nt})$ in lieu of $\ell(\{\boldsymbol{w}_m^{\mathrm{T}}\boldsymbol{x}_n\}_{m=1}^{\mathsf{M}}, y_n)$, namely, the true object function for the main task.

3.1.2 Learning shared features via regularization

Conventionally, all T parameters $\{\boldsymbol{w}_m\}_{t=1}^{\mathsf{T}}$ are learned by independently training (1+A) classifiers. For linear discriminants such as $\boldsymbol{w}_m^{\mathrm{T}}\boldsymbol{x}_n$, the resulting parameter often reveals how effective features are. For instance, a zero-valued

element w_{mi} indicates that the i-th feature of $\boldsymbol{x}_n$ does not play a role in classifying objects. Thus, intuitively, for related tasks, we expect their parameters to reveal similar sparsity patterns. Furthermore, we hypothesize that shared patterns will enable more effective parameter training—for example, reducing feature space dimensionality, thus improving classification performance. How can we identify such common patterns across tasks?

This desideratum is achieved in two steps. The first is to transform the original features to a shared feature space $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x}_n \in \mathcal{U}$ for all tasks [2, 4]. The second step is to learn models in the space of $\mathcal{U}$ and promote a common sparsity pattern in the new parameters. Concretely, we express the discriminant in $\{\boldsymbol{\theta}_t\}$ such that $\boldsymbol{w}_t = \boldsymbol{U}\boldsymbol{\theta}_t$. Analogously to $\boldsymbol{W}$, we collect all $\boldsymbol{\theta}_t$ in $\boldsymbol{\Theta} \in \mathbb{R}^{\mathsf{D}\times\mathsf{T}}$. We jointly optimize all loss functions, but regularized with $\boldsymbol{\Theta}$'s $(2,1)$-norm,

$$\boldsymbol{\Theta}^*,\ \boldsymbol{U}^* = \arg\min \sum_t \sum_n \ell(\boldsymbol{\theta}_t^{\mathrm{T}}\,\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x}_n, y_{nt}) + \gamma\,\|\boldsymbol{\Theta}\|_{2,1}^2 \tag{3.1}$$

The norm is given by $\|\boldsymbol{\Theta}\|_{2,1} = \sum_{d=1}^{D} \sqrt{\sum_t \theta_{td}^2}$. An important property of this norm is that it computes the 2-norm of parameter values in each dimension *across* tasks. Consequently, for any dimension d, the regularization attains the minimum if and only if the corresponding parameters are all zero: $\theta_{td} = 0$ for all t. Therefore, the regularization would choose the $\boldsymbol{\Theta}$ with the *smallest* number of *non-zero rows*.

The discriminant $\boldsymbol{\theta}_t^{\mathrm{T}}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{x}_n$ depends only on nonzero elements of $\boldsymbol{\theta}_t$. Thus equation 3.1 yields solutions that use a subset of features that are commonly effective for all tasks. Similar ideas have also been explored in other

settings [120, 78].

The optimization of Equation (3.1) is challenging due to the non-smoothness of the regularization term. We next describe the alternating minimization algorithm proposed in [2].

3.1.3 Convex optimization

The optimization algorithm of [2] starts by identifying equation 3.1 with its equivalent form

$$\begin{aligned} \boldsymbol{W}^*,\ \boldsymbol{\Omega}^* = \arg\min \sum_t \sum_n \ell(\boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{x}_n, y_{nt}) \\ + \gamma \sum_t \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{\Omega}^{-1} \boldsymbol{w}_t + \gamma\epsilon\, \mathsf{Trace}(\boldsymbol{\Omega}^{-1}), \end{aligned} \tag{3.2}$$

where $\boldsymbol{\Omega} \in \mathbb{R}^{\mathsf{D}\times\mathsf{D}}$ is constrained to be a positive definite matrix with bounded trace $\mathsf{Trace}\,(\boldsymbol{\Omega}) = 1$. $\epsilon \ll 1$ is a smoothing parameter for numerical stability and benign convergence properties (cf. Theorem 3 in [2]). $\boldsymbol{\Omega}$'s role can be understood more clearly by relating the solutions to the two problems in equation 3.1 and equation 3.2:

$$\boldsymbol{W}^* = \boldsymbol{U}^*\, \boldsymbol{\Theta}^*,\ \boldsymbol{\Omega}^* = \boldsymbol{U}^*\, \mathsf{Diag}\left(\left\{ \frac{\|\boldsymbol{\Theta}_d\|_2}{\|\boldsymbol{\Theta}\|_{2,1}} \right\}_{d=1}^{\mathsf{D}} \right) \boldsymbol{U}^{*\mathrm{T}} \tag{3.3}$$

where the operator $\mathsf{Diag}(\cdots)$ converts its D-element arguments as elements of a diagonal matrix. $\|\boldsymbol{\Theta}_d\|_2$ is the 2-norm of $\boldsymbol{\Theta}$'s d-th row: $\sqrt{\sum_t \theta_{td}^2}$. Intuitively, the diagonal measures relatively how much each row of $\boldsymbol{\Theta}$ is "non-zero". Therefore, the matrix $\boldsymbol{\Omega}$ measures relative effectiveness of each feature dimension.

Further insight could be gained by drawing an analogy to the maximum a posteriori (MAP) estimator when the prior distribution for the parameter $\boldsymbol{w}_t$ is a Gaussian $\mathcal{N}(\boldsymbol{w}_t \mid \mathbf{0}; \boldsymbol{\Sigma}^{-1})$. The regularization term of the MAP estimator is in the form $\boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} \boldsymbol{w}_t$. Therefore, intuitively, $\boldsymbol{\Omega}$ functions as an estimator of the covariance structure, computed from all parameters $\boldsymbol{w}_t$ (or equivalently, $\boldsymbol{\theta}_t$), over all tasks.

Equation 3.2 is computationally advantageous for it is a convex optimization. To solve it, we alternatively minimize over $\{\boldsymbol{w}_t\}$ and $\boldsymbol{\Omega}$ while holding the other fixed. When $\boldsymbol{\Omega}$ is fixed, each $\boldsymbol{w}_t$ can be identified as

$$\boldsymbol{w}_t^* = \arg\min \sum_n \ell(\boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{x}_n, y_{nt}) + \gamma \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{\Omega}^{-1} \boldsymbol{w}_t \ . \tag{3.4}$$

With two simple variable substitutions, the optimization takes the standard form of ℓ_2-norm regularization:

$$\hat{\boldsymbol{w}}_t^* = \arg\min \sum_n \ell(\hat{\boldsymbol{w}}_t^{\mathrm{T}} \boldsymbol{z}_n, y_{nt}) + \gamma \left\| \hat{\boldsymbol{w}}_t \right\|_2^2 , \tag{3.5}$$

$$\boldsymbol{z}_n \leftarrow \boldsymbol{\Omega}^{1/2} \boldsymbol{x}_n, \quad \hat{\boldsymbol{w}}_t \leftarrow \boldsymbol{\Omega}^{-1/2} \boldsymbol{w}_t. \tag{3.6}$$

When the parameters $\{\boldsymbol{w}\}$ are fixed, the optimal $\boldsymbol{\Omega}$ that minimizes equation 3.2 has a closed-form solution:

$$\boldsymbol{\Omega} = \frac{(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}} + \epsilon \boldsymbol{I})^{1/2}}{\mathbf{Trace}\left[(\boldsymbol{W}\boldsymbol{W}^{\mathrm{T}} + \epsilon \boldsymbol{I})^{1/2}\right]}. \tag{3.7}$$

The alternating minimization procedure monotonically decreases the objective function until the optimum solution is reached. Algorithm 1 lists the key steps. We set the hyperparameters γ and ϵ using a validation data set.

Algorithm 1 Learning Shared Features for Linear Classifier [2]

Require: training data $(\boldsymbol{x}_n, \{y_{nt}\}), \epsilon, \gamma$
Ensure: $\boldsymbol{W}^*$, $\boldsymbol{\Omega}^*$
1: Initialize $\boldsymbol{\Omega}$ with a scaled identity matrix $\frac{1}{\mathsf{D}}\boldsymbol{I}$
2: **while** $\boldsymbol{W}$ still changes between two iterations **do**
3: Compute transformed variables according to Equation (3.6)
4: Solve $\hat{\boldsymbol{w}}_t$ according to Equation (3.5)
5: Compute $\boldsymbol{w}_t$ as $\boldsymbol{w}_t = \boldsymbol{\Omega}^{1/2}\hat{\boldsymbol{w}}_t$
6: Update $\boldsymbol{\Omega}$ according to Equation (3.7)
7: **end while**

3.1.4 Extension to kernel classifiers

The feature learning framework can be extended to kernel-based nonlinear classifiers. We apply the kernel construction of [2]. Let $K(\boldsymbol{x}_n, \boldsymbol{x}_{n'})$ denote the kernel function between two original feature vectors $\boldsymbol{x}_n$ and $\boldsymbol{x}_{n'}$. The kernel induces a nonlinear feature mapping $\phi(\boldsymbol{x}_n) \in \mathcal{H} \subset \mathbb{R}^{\mathsf{H}}$. We perform feature learning in this new space $\mathcal{H}$.

To "kernelize", note that the optimal parameter $\boldsymbol{W} \in \mathbb{R}^{\mathsf{H}\times\mathsf{T}}$ for the models is a linear combination of (training) feature vectors. This can be understood intuitively by observing that Equation (3.5) is the standard formulation of an SVM; therefore the solution $\{\hat{\boldsymbol{w}}_t^*\}$ is a linear combination of feature vectors. The same statement is also true for $\boldsymbol{W}$, as the two are linearly related as in Equation (3.6).

It is computationally convenient to express $\boldsymbol{W}$ using the basis $\boldsymbol{V}$ of the feature space $\mathcal{H}$: $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{\alpha}$ (we have adopted a slightly different notation from [2] by adhering to the standard nomenclature in SVMs). We assume the number of basis vectors in $\boldsymbol{V}$ is $\mathsf{B} < \mathsf{N}$ where N is the total number of

feature vectors. The matrix $\boldsymbol{\alpha}$ is the linear combination matrix, each column for a task. The basis $\boldsymbol{V}$ can be computed from the kernel matrix formed from training feature vectors, for instance, through eigendecomposition or Gram-Schmidt (G-S) orthogonalization. We use the latter technique for its slightly lower computational overhead. Concretely, we randomly choose B training feature vectors $\mathcal{S}$ and express the basis in the linear combination of those features, $\boldsymbol{V} = \boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{B}$, where the matrix $\boldsymbol{\Phi}_{\mathcal{S}}$'s columns are the nonlinear features computed from the chosen training instances. The matrix $\boldsymbol{B} \in \mathbb{R}^{\mathsf{B}\times\mathsf{B}}$ stores the linear combination coefficients, computed by the G-S process.

The parameter $\boldsymbol{W}$ is also linearly represented, as $\boldsymbol{W} = \boldsymbol{\Phi}_{\mathcal{S}}\boldsymbol{B}\boldsymbol{\alpha}$. Analogous to Equation (3.2), the optimal $\boldsymbol{\alpha}$ is then:

$$\begin{aligned}\boldsymbol{\alpha},\ \boldsymbol{\Omega}^* = \arg\min \sum_t \sum_n \ell(\boldsymbol{\alpha}_t^{\mathrm{T}}\boldsymbol{z}_n, y_{nt}) \\ + \gamma \sum_t \boldsymbol{\alpha}_t^{\mathrm{T}}\boldsymbol{\Omega}^{-1}\boldsymbol{\alpha}_t + \gamma\epsilon\,\mathsf{Trace}(\boldsymbol{\Omega}^{-1}).\end{aligned} \tag{3.8}$$

where $\boldsymbol{\alpha}_t$ is the t-th column of $\boldsymbol{\alpha}$. $\boldsymbol{z}_n = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{k}_{\mathcal{S}}(\boldsymbol{x}_n)$ is the transformed data, resulting from the linear discriminant in the feature space $\mathcal{H}$,

$$\boldsymbol{w}_t^{\mathrm{T}}\phi(\boldsymbol{x}_n) = (\boldsymbol{B}\boldsymbol{\alpha}_t)^{\mathrm{T}}\boldsymbol{\Phi}_{\mathcal{S}}^{\mathrm{T}}\phi(\boldsymbol{x}_n) = \boldsymbol{\alpha}_t^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{k}_{\mathcal{S}}(\boldsymbol{x}_n), \tag{3.9}$$

where the vector $\boldsymbol{k}_{\mathcal{S}}(\boldsymbol{x}_n) \in \mathbb{R}^{\mathsf{B}}$ consists of the elements of the kernel function $k(\boldsymbol{x}_n, \boldsymbol{x}_b) = \phi(\boldsymbol{x}_b)^{\mathrm{T}}\phi(\boldsymbol{x}_n)$.

The optimization problem Equation (3.8) is now readily solvable using techniques described previously. Key steps are given in Algorithm 2.

Algorithm 2 Learning Features for a Kernel Classifier

Require: training data $(\boldsymbol{x}_n, \{y_{nt}\}), \epsilon, \gamma$, and B
Ensure: $\boldsymbol{\alpha}^*$, $\boldsymbol{\Omega}^*$, $\boldsymbol{B}$
1: Formulate kernel matrix $\boldsymbol{K}$
2: Compute the basis $\boldsymbol{B}, \mathcal{S} \leftarrow$ GRAM-SCHMIDT$(\boldsymbol{K}, \mathsf{B})$
3: Transform data according to Equation (3.9) and $\mathcal{S}$
4: $\boldsymbol{\alpha}^*, \boldsymbol{\Omega}^* \leftarrow$ ALGORITHM 1$((\boldsymbol{z}_n, \{y_{nt}\}), \epsilon, \gamma)$

3.1.5 Other extensions

I propose several additional extensions, addressing issues that naturally arise in our setting.

Modeling disparate sets of labels As opposed to [2], the main task and auxiliary tasks here have different sets of labels and different types of loss functions. Thus, we use two regularizers, one for each group. In the linear classifier case, our optimization takes the form,

$$\begin{aligned} \boldsymbol{W}^*,\ \boldsymbol{\Omega}^* = \arg\min \sum_t \sum_n \ell(\boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{x}_n, y_{nt}) + \epsilon\, \mathsf{Trace}(\boldsymbol{\Omega}^{-1}) \\ + \sum_{t=1}^{\mathsf{M}} \gamma_M \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{\Omega}^{-1} \boldsymbol{w}_t + \sum_{t=\mathsf{M}+1}^{\mathsf{T}} \gamma_A \boldsymbol{w}_t^{\mathrm{T}} \boldsymbol{\Omega}^{-1} \boldsymbol{w}_t \end{aligned} \tag{3.10}$$

where γ_M is used for the main task and γ_A for auxiliary tasks. When γ_A is set to zero, the optimization learns shared features from parameters for all object classes, without attributes. We term this setup as "Sharing-Obj". When γ_M is constrained to be the same as γ_A, we recover equation 3.2.

Handling high-dimensional features The alternating minimization algorithm described in Section 3.1.3 depends on re-estimating $\boldsymbol{\Omega}$ and computing its

square root $\boldsymbol{\Omega}^{1/2}$ with equation 3.3 and equation 3.6. For the high-dimensional features used in our setting, directly computing these quantities is costly. We exploit the low-rank property of $\boldsymbol{\Omega}$ to circumvent this challenge. Note that the matrix $\boldsymbol{W}$ has T columns and $\mathsf{D} \gg \mathsf{T}$ rows. Thus, $\boldsymbol{W}$ can be factorized with "thin" singular value decomposition: $\boldsymbol{W} = \boldsymbol{L}\boldsymbol{S}\boldsymbol{R}^{\mathrm{T}}$, where $\boldsymbol{L} \in \mathbb{R}^{\mathsf{D}\times\mathsf{T}}$ and $\boldsymbol{R} \in \mathbb{R}^{\mathsf{T}\times\mathsf{T}}$ are $\boldsymbol{W}$'s (partial) left and right eigenvectors. The diagonal matrix $\boldsymbol{S} \in \mathbb{R}^{\mathsf{T}\times\mathsf{T}}$ is composed of $\boldsymbol{W}$'s singular values $\{\sigma_i(\boldsymbol{W})\}_{i=1}^{\mathsf{T}}$. With some algebraic manipulation, we identify the eigenvalues of $\boldsymbol{\Omega}$:

$$\lambda_i(\boldsymbol{W}) = \left(\sqrt{\sigma_i^2(\boldsymbol{W}) + \epsilon}\right)/\rho, \quad \lambda(\epsilon) = \sqrt{\epsilon}/\rho \tag{3.11}$$

$$\rho = \sum_{i=1}^{\mathsf{T}} \sqrt{\sigma^2(\boldsymbol{W}) + \epsilon} + \sqrt{\epsilon}\,[\mathsf{D} - \mathsf{T}]\,. \tag{3.12}$$

The eigenvectors in $\boldsymbol{L}$ and the subspace orthogonal to them span precisely $\boldsymbol{\Omega}$'s column space. This yields,

$$\boldsymbol{\Omega} = \boldsymbol{L}\,\mathsf{Diag}\left(\{\lambda_i(\boldsymbol{W})\}_{i=1}^{\mathsf{T}}\right)\boldsymbol{L}^{\mathrm{T}} + \lambda(\epsilon)(\boldsymbol{I} - \boldsymbol{L}\boldsymbol{L}^{\mathrm{T}}). \tag{3.13}$$

The matrix $\boldsymbol{\Omega}^{1/2}$ can be formulated similarly, replacing $\lambda_i(\boldsymbol{W})$ and $\lambda(\epsilon)$ with their square roots.

Choosing the kernel basis For the kernelized version, one needs to choose B basis vectors to expand the kernel feature space, as described in Section 3.1.4. We use two simple heuristics. We choose B large enough such that the performance of using the B basis vectors for *individual* task learning is close to the performance of our baseline system's. The individual task learning is set up as

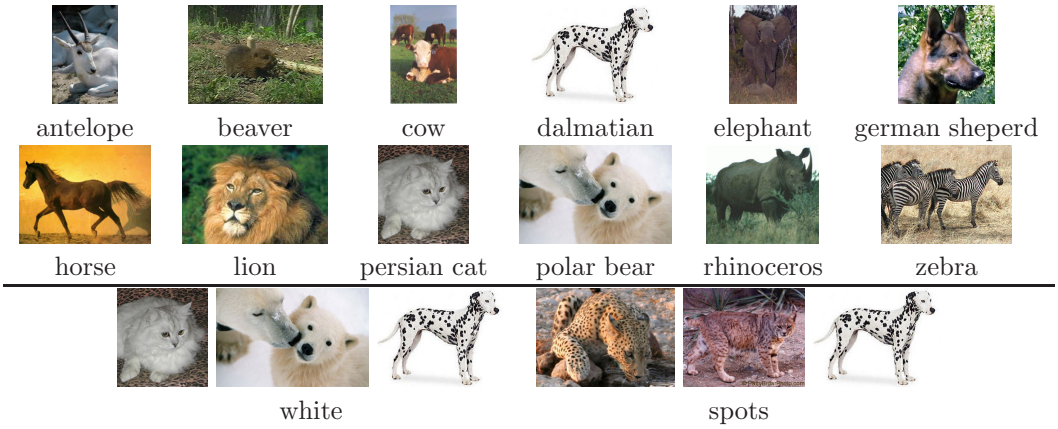


Figure 3.2: Examples images for AwA (Animals with Attributes) dataset. **Top two rows:** Object categories. **Bottom row:** Attributes.

a linear classifier using the *transformed* feature vectors Equation (3.9), while the baseline system's are kernel-based nonlinear classifiers using the original features.

For the Gram-Schmidt process, we choose B/M feature vectors randomly from each of M classes. This gives balanced coverage of different features, and in practice works better than purely randomly selecting without taking object class into consideration.

3.2 Results

I validate my approach against relevant baselines, and report results on object categorization, the main target task.

Figure 3.3: Examples images for OSR (Outdoor Scene Recognition) dataset. **Top two rows:** Object categories. **Bottom row:** Attributes.

Datasets We consider two datasets: the *Animals with Attributes* dataset (AWA) [65], and the *Outdoor Scene Recognition* dataset (OSR) [79]. AWA contains 30,475 images, 50 animal classes, and 85 attributes.[3] Each image is labeled by the animal and attributes present. OSR has 2,688 images, 8 scene classes, and 6 attributes as given in [79]: natural, open, perspective, size, diagonal plane, and depth. See Figures 3.2 and 3.3. We asked another vision researcher to make the assignment from attributes to scenes. We apply random train-test splits, ensuring balance among object classes. Throughout, we use "object" to refer to an animal or scene.

Baselines We consider two baselines:

- a traditional multi-class object recognition approach using an SVM with

[3]For all methods, we use the 59 attributes exceeding 70% accuracy as reported in [65], since some are unpredictable from the given features.

a χ^2 kernel computed on image features, which we refer to as **No sharing-Object**, or **NSO**.

- an approach that treats attributes as intermediate features, which we call **No sharing-Attribute**, or **NSA**.

For NSA, we train SVMs on image features to predict attribute labels, and then treat their outputs as features to a multi-class logistic regression classifier. This baseline follows the basic direct attribute prediction (DAP) approach defined in [65]. We use LIBSVM.

Image features All methods use the same original image features. For AWA, we use the six (SIFT, rgSIFT, PHOG, SURF, LSS, RGB) provided with the dataset, each up to 2688-D. For OSR we generate 512-D Gist and 45-D LAB color histograms. We average the kernels computed over multiple feature types. Note that both datasets permit global descriptors, since there is one primary object of interest per image. To test with multi-object images, one would apply a window-based detector.

3.2.1 Impact of sharing features

First we evaluate the object recognition accuracy of our approach and the baselines. Our approach gets the same training images for both the attribute and object tasks. We form four training splits of increasing size (10% to 60%), and reserve the rest for validation and testing (20% each). We demonstrate two variants of our approach: Sharing-Obj, where we learn a common

	50-class Animals Dataset			
Method / % train data	10%	20%	40%	60%
No sharing-Obj. (NSO)	31.96	38.12	44.08	48.03
No sharing-Attr. (NSA)	31.03	35.61	41.12	43.59
Sharing-Obj. (Ours)	37.08	41.01	46.46	49.15
Sharing+Attr. (Ours)	36.73	42.60	47.70	50.94
% gain over NSO	**14.92%**	**11.75%**	**8.21%**	**6.06%**
% gain over NSA	**18.37%**	**19.63%**	**16.00%**	**16.86%**

Table 3.1: Accuracy on the 50-class animals dataset (AWA), as a function of training set size. Learning shared representations with our approach significantly improves generalization on the novel test set, and can be most pronounced when labeled training data is limited.

representation for all object classes simultaneously, corresponding to $\gamma_A = 0$ in Equation (3.10), and Sharing+Attributes, where we learn the space for all objects and attributes, corresponding to $\gamma_A = \gamma_M$.

Table 3.1 and Table 3.2 shows the results. Our feature sharing approach offers significant improvements over both 'No sharing' baselines, and we obtain the best results when jointly learning with both the objects and attributes. The last two rows summarize gains of Sharing+Attributes over the baselines. Our improvements over the NSO baseline are perhaps most informative, since the general approach taken by NSO (multiple image features, kernel combination, nonlinear SVM) is typical in state-of-the-art image recognition techniques.

While the margin between our Sharing-Object and Sharing+Attributes variants is smaller than the margin between not sharing at all versus sharing, the impact of attributes is clear and consistent. A one-tailed paired t-test on the 60% training split confirms that the accuracy gain with attribute tasks

	8-class Scene Dataset			
Method / % train data	10%	20%	40%	60%
No sharing-Obj. (NSO)	76.76	79.75	83.03	83.74
No sharing-Attr. (NSA)	57.77	58.98	60.50	60.78
Sharing-Obj. (Ours)	78.76	81.49	85.05	86.06
Sharing+Attr. (Ours)	78.09	81.62	85.89	87.01
% gain over NSO	**1.73%**	**2.34%**	**3.44%**	**3.90%**
% gain over NSA	**35.17%**	**38.39%**	**41.97%**	**43.16%**

Table 3.2: Object prediction accuracies of Sharing+Attributes and baselines on the 8-class scene dataset (OSR), as a function of training set size.

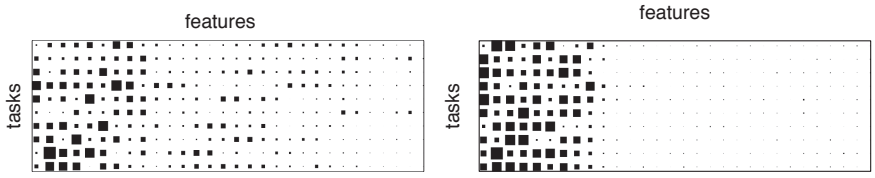


Figure 3.4: Hinton diagram of the matrix $\mathbf{\Theta}$ in the initial and last iterations of Alg. 2. Each square is a matrix entry, and area reflects the entry's magnitude. For clarity only a partial matrix is shown, for the first 30 features (horizontally) and the first 10 object classes (vertically). The matrix at the last iteration is much sparser.

is statistically significant (for $\alpha = 5\%$ on AWA and $\alpha = 1\%$ on OSR). By separately tuning the γ_M and γ_A regularization weights, we expect even better performance; we simply let them be equal to save computation time.

Interestingly, on the larger AWA set, the gain using our method are largest for smaller labeled data pools, supporting our claim that attribute feature sharing can have a beneficial regularization effect for object learning. This is an encouraging result, particularly since obtaining attribute labels on object-labeled data has minimal additional overhead for many attribute

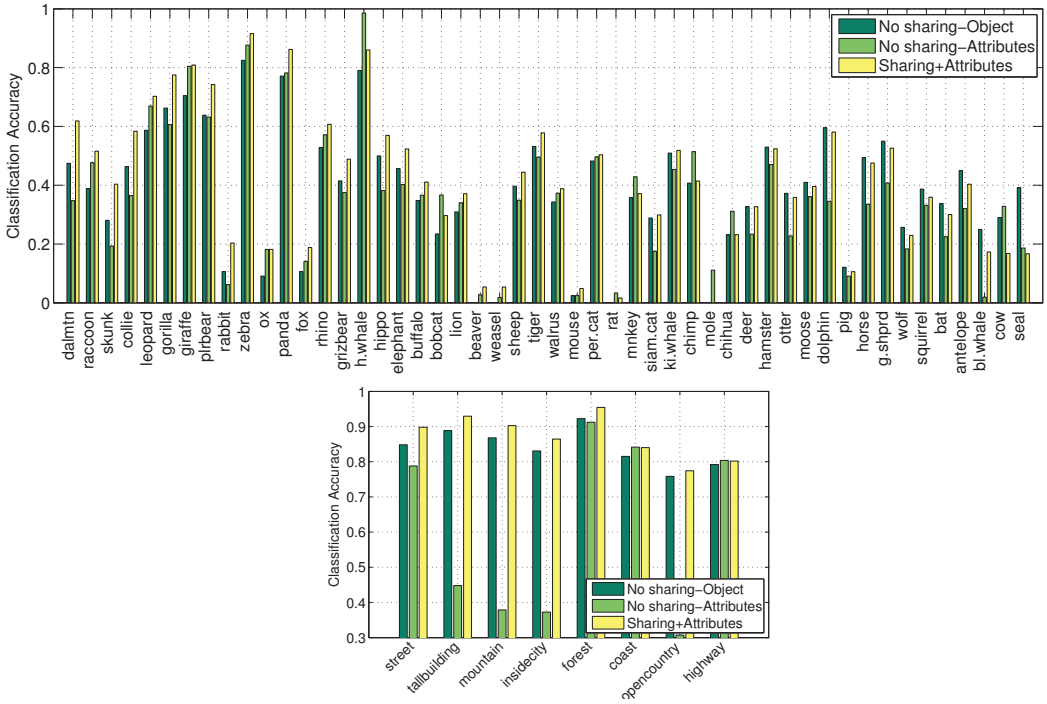


Figure 3.5: Accuracy on AWA (top) and OSR (bottom) classes. Our approach outperforms methods that learn objects (No sharing-Object) or attributes (No sharing-Attributes) independently.

types, as discussed previously. Figure 3.4 visualizes the shared features over iterations, showing how we converge to a common sparse set.

Figure 3.5 breaks out the prediction accuracy per object category on both datasets. We improve accuracy for 33 of the 50 AWA classes, and yield correct predictions for some classes the baselines miss completely (e.g., `beaver`, `rat`). On OSR, the absolute accuracy is higher overall, due to the smaller multi-way decision. However, NSA suffers due to the insufficiency of the attribute vocabulary; it happens that the scenes `tallbuilding` and `insidecity` have exactly the same attribute definitions. In contrast, our approach accounts

for attributes while still learning features sufficient to make the distinction.

One might ask whether some *arbitrary* grouping of object classes into tasks might also have similar benefits. That is, are our gains due to the attributes' meaning, or could it be a sort of "error-correcting code" effect? To analyze this, we test a baseline where each object's attribute labels are randomly reassigned to other attributes, and then apply our method (for five such random assignments on the 60% training split). On OSR, we find this baseline offers no improvement over Sharing-Object (decreasing accuracy by 0.06). On AWA, the baseline improves over Sharing-Object (by 0.97 on average), but by less than sharing with real attributes (which increases accuracy by 1.79). This indicates the attribute semantics are indeed a factor in our method's success.[4]

In the remaining text, I report the results using Sharing+Attributes, and focus on the AWA data, since it is 11× larger and has a richer set of attributes.

3.2.2 Impact of disjoint training images

Our model is flexible to the source of object- and attribute-labeled data, and we can train the tasks on disjoint sets of images. This is relevant when one has a large set of existing attribute-labeled data, and wants to use it to regularize the training process for a new set of object models.

[4]Looking closely at the AWA data, we see that the baseline's small gain made with randomly assigned attribute labels may be misleading. Because the classes are fine-grained, any random assignment of labels can overlap with meaningful attributes; the 85 attribute labels in AWA are certainly not exhaustive for the 50 animals.

	Image source for attributes		
Method	Same	Disjoint	All
No sharing-Object (NSO)	72.99	72.99	72.99
Sharing+Attribute	76.40	76.32	77.05
% gain	4.67%	4.56%	5.56%

Table 3.3: [Object prediction accuracies for Sharing+Attributes and NSO, as a function of which image pool is used for the attribute tasks.]Object prediction accuracy as a function of which image pool is used for the attribute tasks, on the 10-class AWA subset.

Thus, we next examine the impact of which images are used as the auxiliary attribute tasks to train the object classifiers. We select 10 classes (the same as [65]) to train the object classifiers, and test three variations for learning the attributes: 1) the *same* images used for the objects, 2) a *disjoint* set of images containing object classes outside of the 10, and 3) all images, the union of the previous two.

Table 3.3 shows the results. Interestingly, I see that our method performs similarly whether the attribute data overlaps or not (see first two columns). This suggests that the value of the attributes is not simply having deeper/stronger labels on the very same training examples; rather, it is the fact that we identify a common space where both types of labels are well predicted. The table also indicates that more attribute-labeled images is helpful (cf. last column).

3.2.3 Selecting relevant attributes

Having tested the impact of *which images* have attribute labels, next we consider the impact of *which attribute classes* are leveraged as auxiliary

tasks. Presumably, not all attributes will benefit feature sharing, and—as usual in multi-task learning—some may be detrimental. Even if all attributes were relevant to some degree, we may want to be selective to save training costs.

Thus, I explore a simple form of automatic attribute selection in which we rank all attributes by their mutual information (MI) with the 10 animals[5]. Figure 3.6 (left) displays the computed MI, from the most informative attributes (e.g., "spots", which chimps and pigs lack, but leopards and pandas have) to the least (e.g., none of the 10 animals "fly").

Figure 3.6 (right) shows the impact of using the MI scores to select attributes for sharing. Both dotted curves denote our method, but one uses the k *most* informative attributes, and the other uses the k *least* informative attributes.[6] The most interesting cases are for lower values of k. (For higher values of k, the "most" and "least" sets overlap more, and they are identical at $k = 85$.)

The results show that using the 20 attributes with the highest MI yields the best accuracy, while using the lowest 20 is slightly worse than using none whatsoever. Further, we see that more attribute classes do not necessarily always help. These findings plus the fact that training time increases linearly with k (see solid green line, right axis), suggest it is practical to choose in-

[5]chimp, panda, leopard, persian cat, hippo, whale, raccoon, rat, seal

[6]Note, we simply fix the γ and ϵ parameters for all cases, in order to see the effect of the attribute selection in isolation.

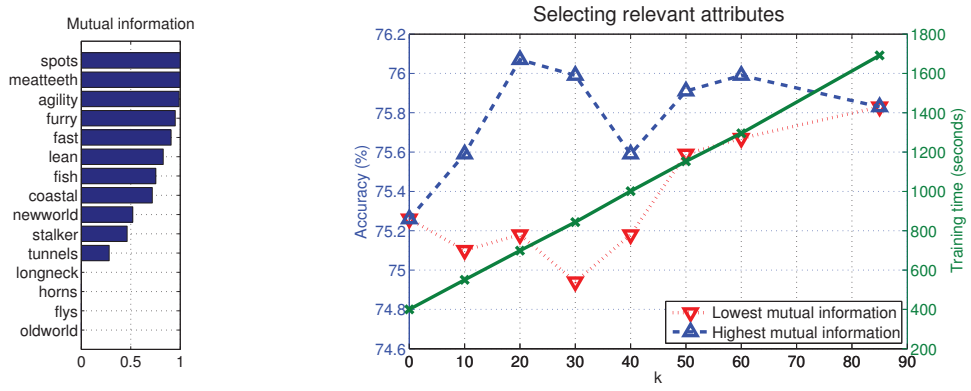
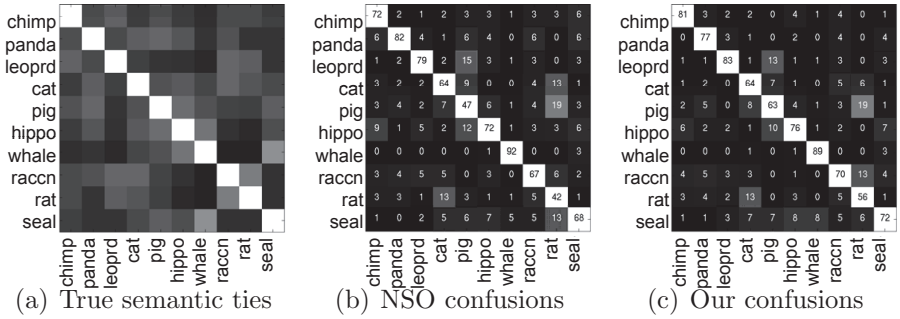


Figure 3.6: Left: Mutual information scores. Right: Object classification accuracy and training time as a function of the number of attribute tasks included.



(a) True semantic ties (b) NSO confusions (c) Our confusions

Figure 3.7: Confusions made by the baseline (b) and our method (c) relative to human-given object relationships (a).

telligently. This result also shows the potential for performing task selection outside of the feature sharing learning procedure.

3.2.4 Semantically meaningful predictions

Finally, we analyze to what extent the semantics we introduce by jointly training objects and attributes are manifest in our method's predictions. Figure 3.7 compares the confusion matrices for our method (c) and NSO (b). To

judge the "reasonableness" of their errors, in (a) we depict the true relationships between all pairs of the 10 objects. To obtain this matrix, we use human subjects' ratings collected in [81] about the relative strength of association between the 85 attributes and 50 objects in AWA. For each object, we create a vector of its 85 property "strengths", and then compute the pairwise χ^2 kernel values between all such vectors. Brighter boxes indicate greater true association in (a), and higher confusion in (b,c). Thus, if a method captures semantics well, its confusion matrix will look more like (a).

First, we notice that our method boosts accuracy for most classes, raising the mean diagonal from 66.9% to 68.9%. Second, we see that the pairs for which our method most reduces confusions (e.g., `pig` vs. `rat`) are more distinctive semantically. On the flip side, some closely related pairs become confused by our method (e.g., `raccoon` vs. `cat`). Figure 3.8 shows example animal category and attribute predictions, compared alongside NSO and NSA.

3.3 Discussion

In this chapter, I showed that by learning a common feature space suitable to either attribute or object tasks, the classifiers can obtain noticeably stronger object recognition performance. I demonstrated the proposed method's improved generalization accuracy and its potential to make more predictable errors in terms of human-defined semantics.

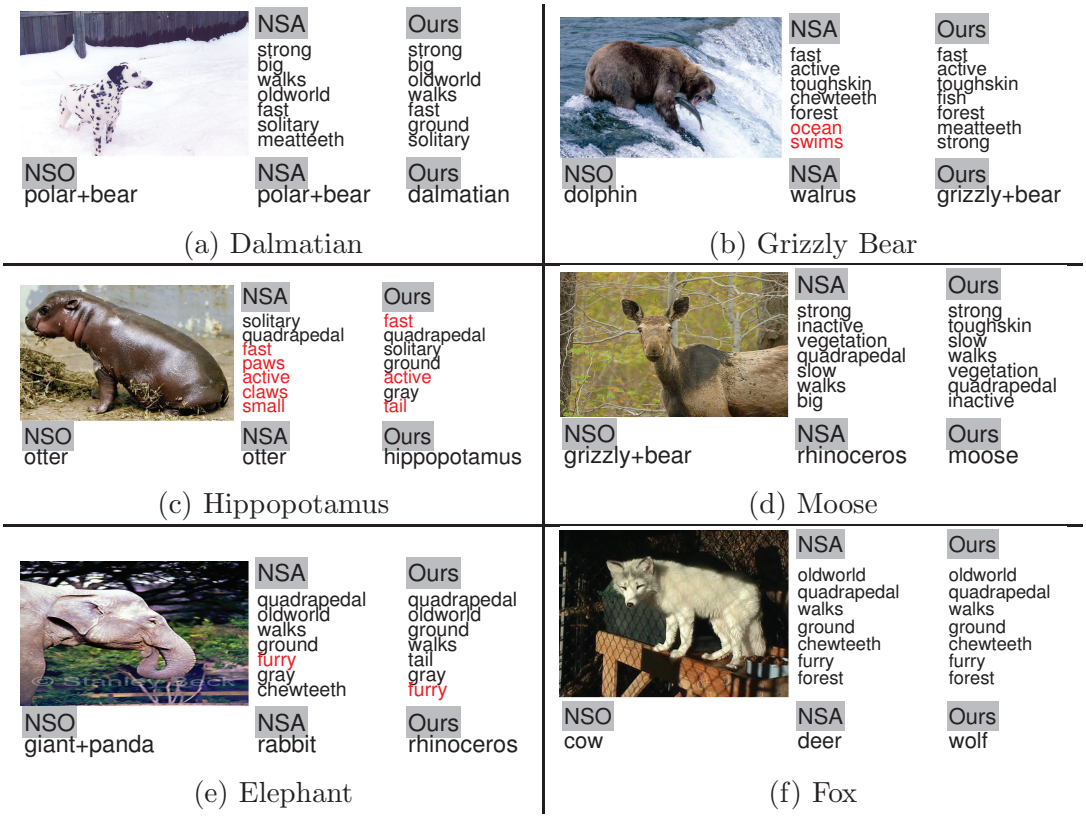
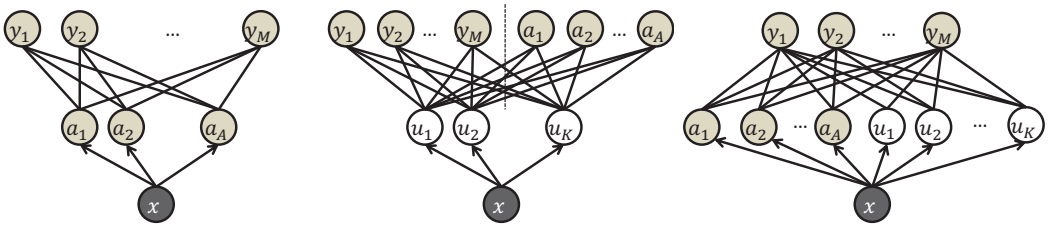


Figure 3.8: Example predictions by our method (right column in each), No Sharing-Attributes (NSA, middle columns), and No Sharing-Objects (NSO, object prediction under each image). Attributes are those with 7 highest positive decision values, by ours or NSA (red attributes incorrect). (a)-(d) illustrate good results, and (e)-(f) show failure cases that highlight our method's tendency to make semantically meaningful errors.

The enforced sharing via mixed norm regularization results in discarding features that are only specific to each category and keeping the ones that are shared with attributes, which adds more semantics to the learned feature space. This semantic guidance not only makes the category recognition model more robust, but also leads to more semantically meaningful predictions. The



(a) DAP (b) Ours (c) DSLDA

Figure 3.9: Conceptual graphical representations of direct attribute prediction (DAP) [65], our feature sharing method, and doubly-supervised latent Dirichlet allocation (DSLDA) [1]. Dark gray nodes denote observed nodes, light gray nodes denote nodes observed only during training and inferred in test, and white nodes denote latent nodes that are never observed. Further, M is the number of object classes, A is the number of attributes, and K is the dimensionality of the shared latent features.

introduction of attributes here could be viewed as introducing a layer of flat higher-level semantic concepts that groups the categories as either having or not having the desired semantic property.

While our method shows impressive results outperforming state-of-the art methods, there still remains further room for improvement. In our model, we treated the attributes as additional supervision to class labels in the output layer, and the features associated with each attribute were indirectly learned through the feature sharing. However, this indirect attribute-guided latent shared feature learning does not guarantee that the features learned on the latent space directly correspond to each attribute, especially when the attribute describes high-level semantic properties such as *fast* or *domestic*. Consequently, our model might result in learning less 'semantic' features compared to explicit attributes modeling as in [65], while achieving more discrimination

power (Figure 3.9 (a), (b)). Obtaining better discrimination power with a possible sacrifice of semantics is perfectly fine for the object categorization task we are aiming at, but might be less optimal if the objective is to learn strictly semantic models (or features).

Doubly-supervised latent Dirichlet allocation (DSLDA) [1], a recently proposed hybrid supervised-latent topic model, suggests a way to take advantage of both explicit attribute modeling and latent shared feature learning. DSLDA has both supervised attributes and latent shared features in the intermediate layer, where the former accounts for attributes while the latter accounts for high-level shared topics not included in the set of attributes (Figure 3.9 (c)). Still, DSLDA has its limitation that it cannot benefit from additional supervision from attributes when learning the shared latent features, as our method does, due to the separate training of attributes and latent features.

The limitation common to all these models is that they only have a single intermediate layer to represent attributes, while the attributes come in diverse semantic granularities. Attributes such as *longleg* and *lean* can be directly inferred from visual features, while *fast* might require an inference based on the previous lower-level attributes. This observation suggests a possible multi-layer semantic model which improves upon our model, where the category classifiers are essentially learned on latent shared features guided with attributes as in our original problem formulation, where they have multiple layers of transformations instead of a single layer. In this multi-layer model, different levels of attributes can be associated with feature learning at

each layer. Section 7.2.2 will discuss more on high-level ideas for this deeper semantic model.

The limitation of having a global, binary single intermediate layer, and ignoring the difference in abstraction level between the semantic concepts and groups can be viewed more as a limitation inherent to attributes themselves. Some semantic concepts have more explicit subset relationships among themselves. For example, consider *canine* and *carnivore*. We can group animals into canine and non-canine groups, and carnivore and non-carnivore groups as with the attributes, but these have more obvious relation that the former is a subset of the latter. A hierarchical model is more suitable to such cases where we can define a clear subset relation between semantic concepts. In the next chapter, I show how a *taxonomy* could be exploited to help learn object category recognition.

Chapter 4

Learning Disjoint Features on a Taxonomy

The binary attributes we explored in the previous chapter divide the categories into two groups: those that have the attribute, and those that do not. However, this introduction of a single layer of meta-categories is not the only way to categorize basic level categories into larger groups. Instead, we could merge categories into superclasses by their similarities, and split a category into subcategories by the observed difference, or certain human-designed criteria, in a hierarchical way. Such a semantic hierarchy is called a *taxonomy*, and is the second type of external semantic knowledge I explore in this thesis.

Well-known taxonomies employed for categorization include WordNet, which groups words into sets of cognitive synonyms and their super-subordinate relations [35], and the phylogenetic tree of life, which groups biological species based on their physical or genetic properties. Critically, such trees implicitly embed cues about human perception of categories, how they relate to one another, and how those relationships vary at different granularities. Thus, in the context of visual object recognition, such a structure has the potential to guide the selection of meaningful low-level features, essentially augmenting the

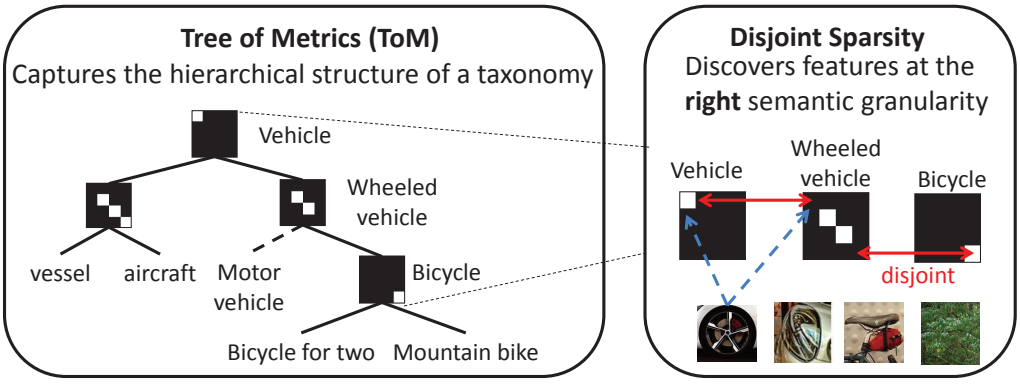


Figure 4.1: **Main Idea:** Leveraging parent-child relationship in a given semantic taxonomy, we learn a tree of metrics (ToM) that captures compact, discriminative visual features for each level. **Left:** we learn a local metric at each node of a taxonomy, that discriminates between its subclasses. **Right:** For the metrics that are associated in an ancestor-descendants relationship, we want each metric to select a set of features different from others, to identify exclusively informative features at each semantic granularity.

standard supervision provided by image labels. Some initial steps have been made based on this intuition, typically by leveraging the WordNet hierarchy as a prior on inter-class visual similarity [124, 72, 98, 27, 37, 26, 105].

I propose a metric learning approach[1] to learn discriminative visual representations while also exploiting external knowledge about the target objects' semantic similarity.[2] We assume the external knowledge itself is available in the form of a hierarchical taxonomy over the objects (e.g., from WordNet or

[1] The work introduced in this chapter is published in [55].

[2] "learned representation" and "learned metric" are used interchangeably, since we deal with sparse Mahalanobis metrics, which are equivalent to selecting a subset of features and applying a linear feature space transformation.

some other knowledge base). My approach exploits these semantics in two novel ways.

First, we construct a *tree of metrics* (ToM) to directly capture the hierarchical structure. In this tree, each metric is responsible for discriminating among its immediate object subcategories. Specifically, we learn one metric for each non-leaf node, and require it to satisfy (dis)similarity constraints generated among its subtree members' training instances. We use a variant of the large-margin nearest neighbor objective [112], and augment it with a regularizer for sparsity in order to unify Mahalanobis parameter learning with a simple means of feature selection.

Second, rather than learn the metrics at each node independently, I introduce a novel regularizer for *disjoint sparsity* that couples each metric with those of its ancestors. This regularizer specifies that a disjoint set of features should be selected for a given node and its ancestors, respectively. Intuitively, this represents that the visual features most useful to distinguish the coarse-grained classes (e.g., motor vehicle vs. bicycle. See Figure 4.1) should often be *different* than those cues most useful to distinguish their fine-grained subclasses (e.g., bicycle for two vs. mountain bike). The resulting optimization problem is convex, and can be optimized with a projected subgradient approach. Figure 4.1 shows the overview of these two main ideas.

The ideas of exploiting label hierarchy and model sparsity are not completely new to computer vision and machine learning researchers. Hierarchical classifiers are used to speed up classification time and alleviate data sparsity

problems [72, 50, 62, 73, 16]. Parameter sparsity is increasingly used to derive parsimonious models with informative features [67, 60, 117].

My novel contribution lies in the idea of ToM and disjoint sparsity together as a new strategy for visual feature learning. My idea reaps the benefits of both schools of thought. Rather than relying on learners to discover both sparse features and a visual hierarchy fully automatically, we use external "real-world" knowledge expressed in hierarchical structures to *bias* which sparsity patterns we want the learned discriminative feature representations to exhibit. Thus, our end-goal is not any sparsity pattern returned by learners, but the patterns that are in concert with rich high-level semantics.

I validate my approach with the Animals with Attributes [65] and ImageNet [27] datasets using the WordNet taxonomy. We demonstrate that the proposed ToM outperforms both global and multiple-metric metric learning baselines that have similar objectives but lack the hierarchical structure and proposed disjoint sparsity regularizer. In addition, we show that when the dimensions of the original feature space are interpretable (nameable) visual attributes, the disjoint features selected for super- and sub-classes by my method can be quite intuitive.

4.1 Approach

I review briefly the techniques for learning distance metrics. I then describe an ℓ_1-norm based regularization for selecting a sparse set of features in the context of metric learning. Building on that, I proceed to describe our main

algorithmic contribution, that is, the design of a metric learning algorithm that prefers not only *sparse* but also *disjoint* features for discriminating different categories.

4.1.1 Distance metric learning

Many learning algorithms depend on calculating distances between samples, notably k-nearest neighbor classifiers or clustering. While the default is to use the Euclidean distance, the more general Mahalanobis metric is often more suitable. For two data points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^{\mathsf{D}}$, their (squared) Mahalanobis distance is given by

$$d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{M} (\boldsymbol{x}_i - \boldsymbol{x}_j), \tag{4.1}$$

where $\boldsymbol{M}$ is a positive semidefinite matrix $\boldsymbol{M} \succeq 0$. Arguably, the Mahalanobis distance can better model complex data, as it considers correlations between feature dimensions.

Learning the optimal $\boldsymbol{M}$ from labeled data has been an active research topic (e.g., [25, 47, 112]). Most methods follow an intuitively appealing strategy: a good metric $\boldsymbol{M}$ should pull data points belonging to the same class closer and push away data points belonging to different classes. As an illustrative example, we describe the technique used in constructing large margin nearest neighbor (LMNN) classifiers [112], to which our empirical studies extensively compare.

In LMNN, each point $\boldsymbol{x}_i$ in the training set is associated with two sets

of different data points in $\boldsymbol{x}_i$'s nearest neighbors (identified in the Euclidean distance): the "targets" whose labels are the same as $\boldsymbol{x}_i$'s and the "impostors" whose labels are different. Let $\boldsymbol{x}_i^+$ denote the "target" and $\boldsymbol{x}_i^-$ denote the "impostor" sets, respectively. LMNN identifies the optimal $\boldsymbol{M}$ as the solution to,

(4.2)
$$\begin{aligned} \min_{\boldsymbol{M}\succeq 0} \quad & \ell(\boldsymbol{M}) = \sum_i \sum_{j\in \boldsymbol{x}_i^+} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \gamma \sum_{ijl} \xi_{ijl} \\ \text{subject to} \quad & 1 + d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_l) \leq \xi_{ijl};\ \xi_{ijl} \geq 0\ .\forall\ j \in \boldsymbol{x}_i^+,\ l \in \boldsymbol{x}_i^- \end{aligned}$$

where the objective function $\ell(\boldsymbol{M})$ balances two forces: pulling the target towards $\boldsymbol{x}_i$ and pushing the impostor away. The latter is characterized by the constraint composed of a triplet of data points: the distance to an impostor should be greater than the distance to a target by at least a margin of 1, possibly with the help of a slack variable ξ_{ijl}. The minimization of equation 4.2 is a convex optimization problem with semidefinite constraints on $\boldsymbol{M} \succeq 0$, and is tractable with standard techniques.

My approach extends previous work on metric learning in two aspects: 1) We apply a sparsity-based regularization to identify informative features (Section 4.1.2); 2) at the same time, we seek metrics that rely on *disjoint* subsets of features for categories at different semantic granularities (Section 4.1.3).

4.1.2 Sparse feature selection for metric learning

How can we learn a metric such that only a sparse set of features are relevant? Examining the definition of the Mahalanobis distance in equation 4.1,

we deduce that if the d-th feature of $\boldsymbol{x}$ is not to be used, it is sufficient and necessary for the d-th diagonal element of $\boldsymbol{M}$ be zero.

Therefore, analogous to the use of ℓ_1-norm by the popular LASSO technique [97],we add the ℓ_1-norm of $\boldsymbol{M}$'s diagonal elements to the large margin metric learning criterion $\ell(\boldsymbol{M})$ in equation 4.2,

$$\min_{\boldsymbol{M}\succeq 0} \quad \sum_i \sum_{j\in \boldsymbol{x}_i^+} d^2_{\boldsymbol{M}}(\boldsymbol{x}_i, \boldsymbol{x}_j) + \gamma \sum_{ijl} \xi_{ijl} + \lambda \mathsf{Trace}[\boldsymbol{M}], \tag{4.3}$$

where we have omitted the constraints as they are not changed. λ and γ are nonnegative (hyper)parameters trading off the sparsity of the model and the other parts in the objective. Note that since the matrix trace $\mathsf{Trace}[\cdot]$ is a linear function of its argument, this sparse feature metric learning problem remains a convex optimization.

4.1.3 Learning a tree of metrics (ToM) with disjoint visual features

How can we learn a tree of metrics so each metric uses features disjoint from its ancestors'?

Using disjoint features To characterize the "disjointness" between two metrics $\boldsymbol{M}_t$ and $\boldsymbol{M}_{t'}$, we use the vectors of their nonnegative diagonal elements $\boldsymbol{v}_t$ and $\boldsymbol{v}_{t'}$ as proxies to which features are (more heavily) used. This is a reasonable choice as we use the sparsity-inducing ℓ_1-norm in learning the metrics. We measure their degree of "competition" for common features,

$$C(\boldsymbol{M}_t, \boldsymbol{M}_{t'}) = \|\boldsymbol{v}_t + \boldsymbol{v}_{t'}\|_2^2 \ . \tag{4.4}$$

Intuitively, if a feature dimension is not used by either metric, the competition

for that feature is low. If a feature dimension is used by both metrics heavily, then the competition is high. Consequently, minimizing eq. (4.4) as a regularization term will encourage different metrics to use disjoint features. Note that the measure is a convex function of $\boldsymbol{v}_t$ and $\boldsymbol{v}_{t'}$, hence also convex in $\boldsymbol{M}_t$ and $\boldsymbol{M}_{t'}$.

Learning a tree of metrics Formally, assume we have a tree $\mathcal{T}$ where each node corresponds to a category. Let t index the T *non-leaf* or internal nodes. We learn a metric $\boldsymbol{M}_t$ to differentiate its children categories $c(t)$. For any node t, we use $\mathcal{D}(t)$ to denote those training samples whose labeled categories are offspring of t, and $a(t)$ to denote the nodes on the path from the root to t.

To learn our metrics $\{\boldsymbol{M}_t\}_{t=1}^{\mathsf{T}}$, we apply similar strategies of learning metrics for large-margin nearest neighbor classifiers. We cast it as a convex optimization problem:

(4.5)

$$\begin{aligned}
\min_{\{\boldsymbol{M}_t\}\succeq 0} \quad & \sum_t \sum_{c\in c(t)} \sum_{i,j\in\mathcal{D}(c)} d^2_{\boldsymbol{M}_t}(\boldsymbol{x}_i,\boldsymbol{x}_j) + \gamma \sum_{t,c,r,ijl} \xi_{tcrijl} + \sum_t \lambda_t \mathsf{Trace}[\boldsymbol{M}_t] \\
& + \sum_t \sum_{a\in a(t)} \gamma_{ta} C(\boldsymbol{M}_t, \boldsymbol{M}_a) \\
\text{subject to} \quad & \forall\, t, \forall\, c\in c(t), \forall\, r\in c(t)-\{c\}, \forall\, \boldsymbol{x}_i,\boldsymbol{x}_j\in\mathcal{D}(c),\ \boldsymbol{x}_l\in\mathcal{D}(r) \\
& 1 + d^2_{\boldsymbol{M}_t}(\boldsymbol{x}_i,\boldsymbol{x}_j) - d^2_{\boldsymbol{M}_t}(\boldsymbol{x}_i,\boldsymbol{x}_l) \le \xi_{tcrijl};\ \xi_{tcrijl}\ge 0\ .
\end{aligned}$$

In short, there are T learning (sub)problems, one for each metric. Each metric learning problem is in the style of the sparse feature metric learning eq. (4.3). However, more importantly, these metric learning problems are *coupled* to-

gether through the disjoint regularization. Our disjoint regularization encourages a metric $\boldsymbol{M}_t$ to use different sets of features from its *super*-categories—categories on the tree path from the root.

Numerical optimization The optimization problem in Equation (4.5) is convex, though nonsmooth due to the nonnegative slack variables. We use the subgradient method, previously used for similar problems [112]. For problems with a large taxonomy, learning all the regularization coefficients λ_t and γ_{ta} is prohibitive, as the number of coefficient combinations is $O(k^{\mathsf{T}})$, where T is the number of nodes and k is the number of values a coefficient can take. Thus, for the large-scale problems we focus on, we use a simpler and computationally more efficient strategy of Sequential Optimization (SO) by sequentially optimizing one metric at a time. Specifically, we optimize the metric at the root node and then its children, assuming the metric at the root is fixed. We then recursively (in breadth-first-search) optimize the rest of the metrics, always treating the metrics at the higher level of the hierarchy as fixed. This strategy has a significantly reduced computational cost of $O(k\mathsf{T})$. In addition, the SO procedure allows each metric to be optimized with different parameters and prevents a badly-learned low-level metric from influencing upper-level ones through the disjoint regularization terms. (This can also be achieved by adjusting all regularization coefficients in parallel through extensive cross-validation, but at a much higher computational expense.)

Using a tree of metrics for classification Once the metrics at all nodes are learned, they can be used for several classification tasks (e.g., with

k-NN or as a kernel to a SVM). In this work, we study two tasks in particular: 1) We consider "per-node classification", where the metric at each node is used to discriminate its sub-categories. Since decisions at higher-level nodes must span a variety of object sub-categories, these generic decisions are interesting to test the learned features in a broader context. 2) We consider hierarchical classification [33], a natural way to use the full ToM. In this case, we examine the recognition accuracy for the finest-level categories only. We classify an object from the root node down; the leaf node that terminates the path is the predicted label.

I stress that our metric learning criterion of Equation (4.5) aims to minimize classification errors at each node. Thus, improvement in per-node accuracy is more directly indicative of whether the learning has resulted in useful metrics. Understanding the relation between per-node and full multi-class accuracy has been a challenging research problem in building hierarchical classifiers [16, 72].

Relationship to orthogonal transfer Our work shares a similar spirit to the "orthogonal transfer" idea explored in [121]. The authors there use non-overlapping features to construct multiple SVM classifiers for hierarchical classification of text documents. Concretely, they propose an orthogonal regularizer $\sum_{ij} K_{ij} |\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{w}_j|$ where $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ are the SVM parameters. Minimizing it will reduce the similarity of the parameter vectors and make them "orthogonal" to each other. However, orthogonality does not necessarily imply disjoint features. This can be seen with a contrived two-dimensional counterex-

ample where $\boldsymbol{w}_i = [1 \ -1]^{\mathrm{T}}$ and $\boldsymbol{w}_j = [-1 \ -1]^{\mathrm{T}}$. Both features are used, yet the two parameter vectors are orthogonal. In contrast, our disjoint regularizer Equation (4.4) is more indicative of true disjointness. Specifically, when our regularizer attains its minimum value of zero, we are guaranteed that features are non-overlapping as our $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are *nonnegative* diagonal elements of positive semidefinite matrices. Our regularizer is also guaranteed to be convex, whereas the convexity of the regularizer in [121] depends critically on tuning K_{ij}.

4.2 Results

We validate our ToM approach on several datasets, and consider three baselines:

- **Euclidean**: Euclidean distance in the original feature space
- **Global LMNN**: a single global metric for all classes learned with the LMNN algorithm [112]
- **Multi-Metric LMNN**: one metric learned per class using the multiple metric LMNN variant [112].

We chose these baselines to show the advantage of learning a tree of feature spaces over a global feature space, or a set of category-specific feature spaces. Note that our method learns features represented as metrics, instead of classifiers, and can be couple with any classifiers (e.g. SVM) other than

the k-nearest neighbor (kNN) classifier we use for the experiments. Thus, our method is not directly comparable to other hierarchical methods tied to specific classifiers such as [62, 73, 16], since our focus is not on showing the advantage of using a kNN classifier over other classifiers.

We use the code provided by the authors. To evaluate the influence of each aspect of our method, we test it under three variants:

- **ToM**: ToM learning without any regularization terms
- **ToM+Sparsity**: ToM learning with the sparsity regularization term
- **ToM+Disjoint**: ToM learning with the disjoint regularization term.

For all experiments, we test with five random data splits of 60%/20%/20% for train/validation/test. We use the validation data to set the regularization parameters λ and γ among candidate values $\{0, 1, 10, 100, 1000\}$, and we generate 500 $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_l)$ training triplets per class.

4.2.1 Proof of concept on synthetic dataset

First we use synthetic data to clearly illustrate disjoint sparsity regularization. We generate data with precisely the property that coarser categories are distinguishable using feature dimensions distinct from those needed to discriminate their subclasses. Specifically, we sample 2000 points from each of four 4D Gaussians, giving four leaf classes $\{a, b, c, d\}$. They are grouped into two superclasses $A = \{a, b\}$ and $B = \{c, d\}$. The first two dimensions of all

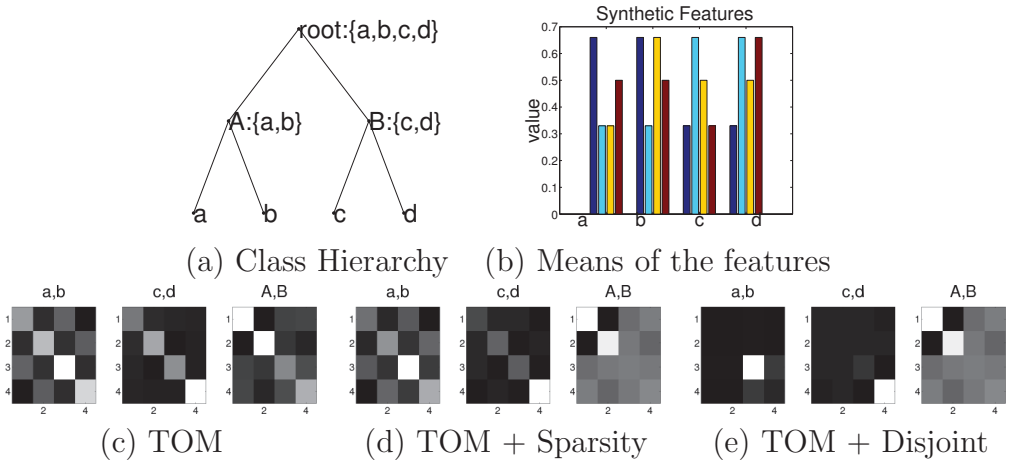


Figure 4.2: Synthetic dataset example. Our disjoint regularizer yields a sparse metric that only considers the feature dimension(s) necessary for discrimination at that given level.

points are specific to the superclass decision (A vs. B), while the last two are specific to the subclasses. See Fig. 5.1 (a) and (b).

We run hierarchical k-nearest neighbor classification ($k = 3$) on the test set. ToM+Sparsity increases the recognition rate by 0.90%, while ToM+Disjoint increases it by 4.05%. Thus, as expected, disjoint sparsity does best, since it selects different features for the super- and sub-classes. Accordingly, in the learned Mahalanobis matrices for each node (Fig. 5.1(c)-(e)), we see disjoint sparsity zeros out the unneeded features for the upper-level metric, showed as black squares in the figure (e). In contrast, the ToM+Sparsity features are sub-optimal and fit to some noise in the data (d).

4.2.2 Visual recognition experiments

Next we demonstrate our approach on challenging visual recognition tasks.

Datasets and implementation details We validate with three datasets drawn from two publicly available image collections: Animals with Attributes (AWA) [65] and ImageNet [27, 26]. Both are well-suited for our scenario, since they consist of fine-grained categories that can be grouped into more general object categories. From the AWA (Figure 3.2) that contains 30,475 images and 50 animal classes, and ImageNet image collections, we form three datasets for empirical validation.

- **AWA-PCA**, which uses the features provided from the dataset in [65] (SIFT, rgSIFT, PHOG, SURF, LSS, RGB), concatenated, standardized, and PCA-reduced to 50 dimensions.
- **AWA-ATTR**, which uses 85-dimensional attribute predictions as the original feature space, formed by concatenating the outputs of 85 linear SVMs trained to predict the presence/absence of the 85 nameable properties annotated by [65], e.g., furry, white, quadrupedal, etc.
- **VEHICLE-20**, which uses 20 vehicle classes and 26,624 images from ImageNet, and apply PCA to reduce the authors' provided visual word features [26] to 50 dimensions per image[3].

[3]This is the dimensionality that worked best for the Global LMNN baseline.

Figure 4.3: Examples images for VEHICLE-20 dataset.

We use WordNet to generate the semantic hierarchies for all datasets. We retrieve all nodes in WordNet that contain any of the object class names on their word lists. In the case of homonyms, we manually disambiguate the word sense. Then, we build a compact partial hierarchy over those nodes by 1) pruning out any node that has only one child (i.e., removing superfluous nodes), and 2) resolving any instances of multiple parentship by choosing the path from the leaf to root having the most overlap with other classes. See Figures 4.4 and 4.5 for the resulting AWA and VEHICLE trees.

Throughout, we evaluate classification accuracy using k-nearest neighbors (k-NN). For ToM, at node n we use $k = 2^{l_n - 1} + 1$, where l_n is the level of the node, and $l_n = 1$ for leaf nodes. This means we use a larger k at the higher nodes in the tree where there is larger intra-class variation, in an effort to be more robust to outliers. For the Euclidean and LMNN baselines, which lack a hierarchy, we simply use k=3. Note that ToM's setting at the final decision nodes (just above a leaf) is also $k = 3$, comparable to the baselines.

4.2.2.1 Per-node accuracy and analysis of the learned representations

Since our algorithm optimizes the metrics at every node, we first examine the resulting per-node decisions. That is, how accurately can we predict the correct subcategory at any given node? The bar charts in Figures 4.4 and 4.5 show the results, in terms of raw k-NN accuracy improvements over the Euclidean baseline. For reference, we also show the Global LMNN baseline. Multi-Metric LMNN is omitted here, since its metrics are only learned for the leaf node classes. We observe a good increase for most classes, as well as a clear advantage relative to LMNN. Furthermore, our results are usually strongest when including the novel disjoint sparsity regularizer. This result supports our basic claim about the potential advantage of exploiting external semantics in ToM.

We find that absolute gains are similar in either the PCA or ATTR feature spaces for AWA, though exact gains per class differ. While the ATTR variant exposes the semantic features directly to the learner, the PCA variant encapsulates an array of low-level descriptors into its dimensions. Thus, while we can better interpret the *meaning* of disjoint sparsity on the attributes, our positive result on raw image features assures that disjoint feature selection is also amenable in the more general case.

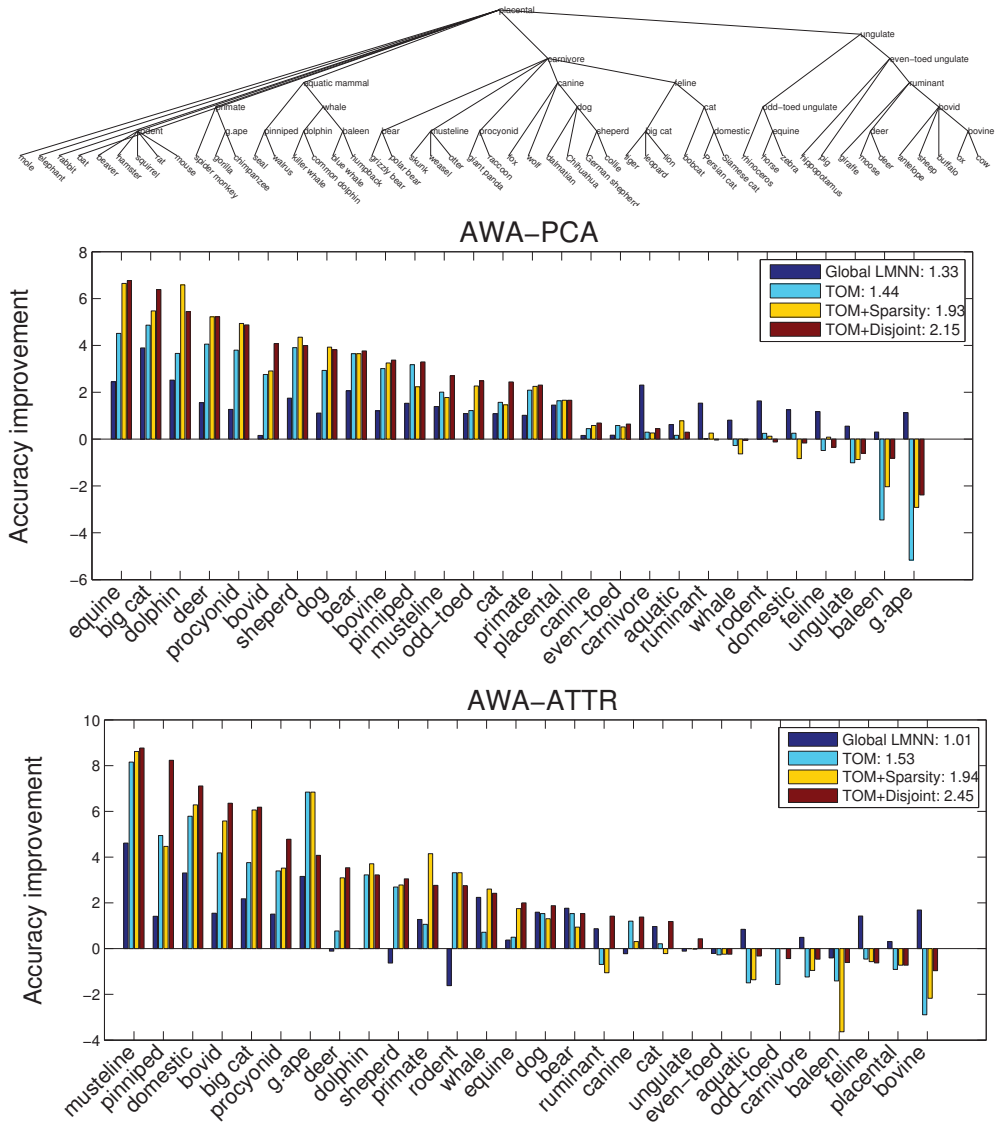


Figure 4.4: Semantic hierarchy for AWA (top row) and the per-node accuracy improvements relative to Euclidean distance, for the AWA-PCA (middle row) and AWA-ATTR (bottom row) datasets. Numbers in legends denote average improvement over all nodes. We generally achieve a sizable accuracy gain relative to the Global LMNN baseline (dark left bar for each class), showing the advantage of exploiting external semantics with our ToM approach.

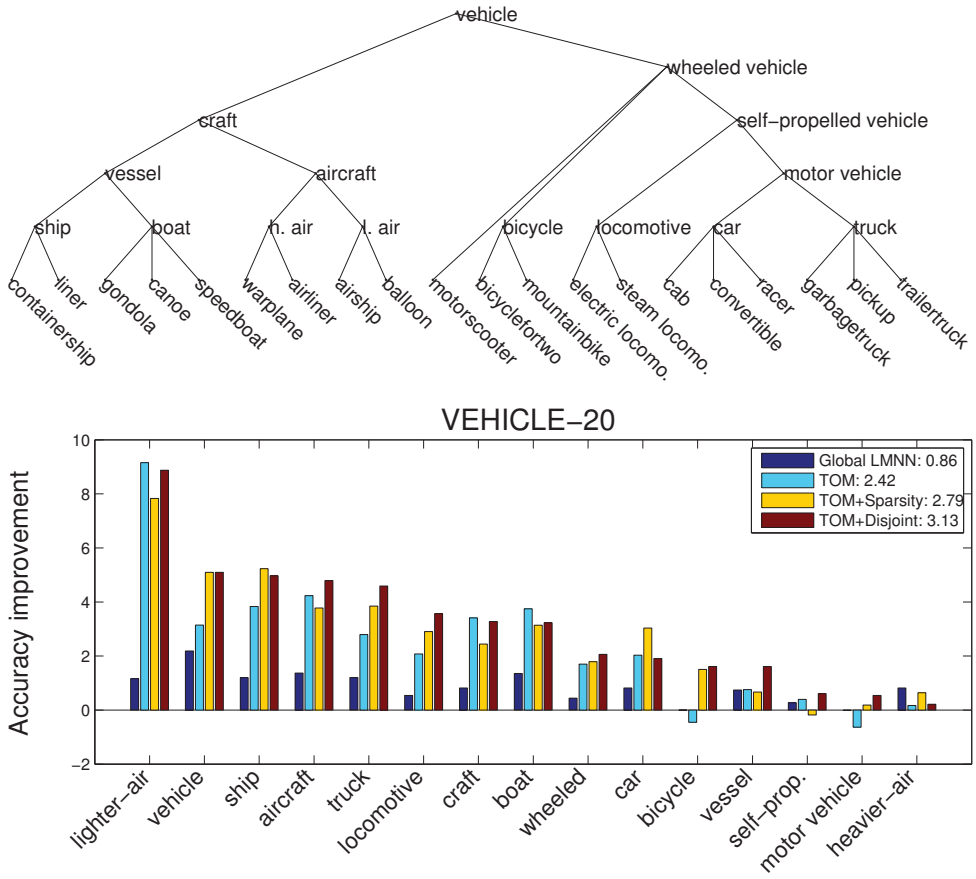


Figure 4.5: Semantic hierarchy for VEHICLE-20 and the per-node accuracy gains, plotted as above.

To look more closely at this, Table 4.1 displays representative superclasses from AWA-ATTR together with the attributes that ToM+Disjoint selects as discriminative for their subclasses. The attributes shown are those with nonzero weights in the learned metrics. Intuitively, we see that often the selected attributes are indeed useful for discriminating the child classes. For example, *tusks* and *plankton* attributes help distinguish common dolphins

Superclass	Subclasses	Attributes selected
whale	dolphin, baleen whale	black, white, blue, gray, toughskin, chewteeth, strainteeth, smelly, slow, muscle, active, fish, hunter, skimmer, oldworld, arctic...
dolphin	common dolphin, killer whale	tusks, plankton, blue, gray, red, patches, slow, muscle, active, insects
odd-toed ungulate	equine, rhinoceros	fast, longneck, hairless, black, white, yellow, patches, spots, bulbous, longleg, buckteeth, horns, tusks, smelly...
equine	horse, zebra	stripes, domestic, orange, red, yellow, toughskin, newworld, arctic, bush

Table 4.1: Attributes selected by ToM+Disjoint for various superclass objects in AWA. See text.

from killer whales, whereas *stripes* and *domestic* help distinguish zebras from horses. At the same time, as desired, we see that the attributes useful for coarser-level categories are distinct from those employed to discriminate the finer-level objects. For example, *fast*, *longneck*, or *hairless* are used to differentiate equine from rhino, but are excluded when differentiating horses from zebras (equine's subclasses).

4.2.2.2 Hierarchical multi-class classification accuracy

Next we evaluate the complete multi-class classification accuracy, where we use all the learned ToM metrics together to predict the leaf-node label of the test points. This is a 50-way task for AWA, and a 20-way task for VEHICLES.

	AWA-ATTR	
Method	Correct label	Semantic similarity
Euclidean	32.36 ± 0.21	53.60 ± 0.26
Global LMNN	32.49 ± 0.42	53.93 ± 0.88
Multi-metric LMNN	32.34 ± 0.35	53.73 ± 0.71
ToM	36.79 ± 0.27	58.36 ± 0.09
ToM + Sparsity	37.58 ± 0.32	59.29 ± 0.58
ToM + Disjoint	$\mathbf{38.29 \pm 0.61}$	$\mathbf{59.72 \pm 0.62}$
	AWA-PCA	
Method	Correct label	Semantic similarity
Euclidean	17.54 ± 0.38	38.11 ± 0.58
Global LMNN	$\mathbf{19.62 \pm 0.51}$	40.34 ± 0.32
Multi-metric LMNN	17.61 ± 0.33	38.94 ± 0.31
ToM	18.70 ± 0.41	43.44 ± 0.43
ToM + Sparsity	18.79 ± 0.46	43.38 ± 0.34
ToM + Disjoint	19.00 ± 0.30	$\mathbf{43.59 \pm 0.19}$

Table 4.2: Multi-class hierarchical classification accuracy and semantic similarity on the **AWA-ATTR** and **AWA-PCA** datasets. Numbers are averages over 5 splits, and standard errors for 95% confidence interval. Our method outperforms the baselines in almost all cases, and notably provides more semantically close predictions. See text.

	VEHICLE-20	
Method	Correct label	Semantic similarity
Euclidean	28.51 ± 0.56	56.10 ± 0.41
Global LMNN	29.65 ± 0.44	57.57 ± 0.45
Multi-metric LMNN	30.00 ± 0.51	57.91 ± 0.54
ToM	31.23 ± 0.67	60.72 ± 0.54
ToM + Sparsity	32.09 ± 0.18	62.66 ± 0.26
ToM + Disjoint	$\mathbf{32.77 \pm 0.32}$	$\mathbf{63.01 \pm 0.21}$

Table 4.3: Multi-class hierarchical classification accuracy and semantic similarity on the **VEHICLE-20** dataset.

Table 4.2 and 4.3 shows the results.

We score accuracy in two ways: **Correct label** records the percentage of examples assigned the correct (leaf) label, while **Semantic similarity** records the semantic similarity between the predicted and true labels. For both, higher is better. The former is standard recognition accuracy, while the latter gives a more nuanced view of the "semantic magnitude" of the classifiers' errors. Specifically, we calculate the semantic similarity between classes (nodes) i and j using the metric defined in [37], which counts the number of nodes shared by their two parent branches, divided by the length of the longest of the two branches. In the spirit of other recent evaluations [9, 26, 37], this metric reflects that some errors are worse than others; for example, calling a Persian cat a Siamese cat is a less glaring error than calling a Persian cat a horse. This is especially relevant in our case, since our key motivation is to instill external semantics into the feature learning process.

In terms of pure label correctness, ToM improves over the strong LMNN baselines for both AWA-ATTR and VEHICLE-20. Further, in all cases, we see that disjoint sparsity is an important addition to ToM. However, in AWA-PCA, Global LMNN produces the best results by a statistically insignificant margin. We did not find a clear rationale for this one case. For AWA-ATTR, however, our method is substantially better than Global LMNN, perhaps due to our method's strength in exploiting semantic features. While we initially expected Multi-Metric LMNN to outperform Global LMNN, we suspect it struggles with clusters that are too close together. For all cases when ToM+Disjoint

outperforms the LMNN or Euclidean baselines, the improvement is statistically significant.

In terms of semantic similarity, ToM is better than all baselines on all datasets. This is a very encouraging result, since it suggests our approach is in fact leveraging semantics in a useful way. In practice, the ability to make such "reasonable" errors is likely to be increasingly important as the community tackles larger and larger multi-class recognition problems.

4.3 Discussion

I presented a new metric learning approach for visual recognition that integrates external semantics about object hierarchy. Experiments with challenging datasets indicate its promise, and support our hypothesis that outside knowledge about how objects relate is valuable for feature learning.

Instead of learning a discriminative metric that considers each category as a separate, independent entity, the proposed ToM learns metrics that preserve the distances between each group of categories at different semantic levels. Further, the added disjoint regularizer forces feature spaces that form ancestor-descendants relationships to compete for the features, which is shown to be effective in isolating features for each semantic granularity. The true selection of features and the convexity is what makes our method superior to the existing exclusive regularization methods based on competition [122, 121]. Both the hierarchical modeling and the isolation of the feature spaces were shown to be useful for hierarchical classification. However, it could still suf-

fer from the problem known as the *semantic gap* — the discrepancy between the semantic and the visual space, which could limit the classification performance at the abstract high-level. This in turn could limit the performance of the whole model, due to the error propagating nature of the hierarchical classification model.

There could be multiple possible solutions to this problem. The first is to construct a hierarchy that can account for both semantics and visual distributions. This could be done by either collapsing or splitting the nodes of the existing semantic taxonomy such that the taxonomy aligns better with the visual distribution, or constructing a hierarchy from the scratch while accounting for both semantic and visual similarities between categories. However, doing so might result in less semantic information being exploited, since our main idea was to exploit human criteria in grouping or splitting of the categories, where a large amount of useful semantic information comes from higher-level nodes representing abstract classes such as *vehicle* or *carnivore*. These high-level nodes usually contain visually diverse subcategories but are nonetheless informative. Empirical results from [18] show that even an evaluation scheme that considers the whole path and holds off from making a hard decision at each node might not cope well with such abstract high-level semantic nodes.

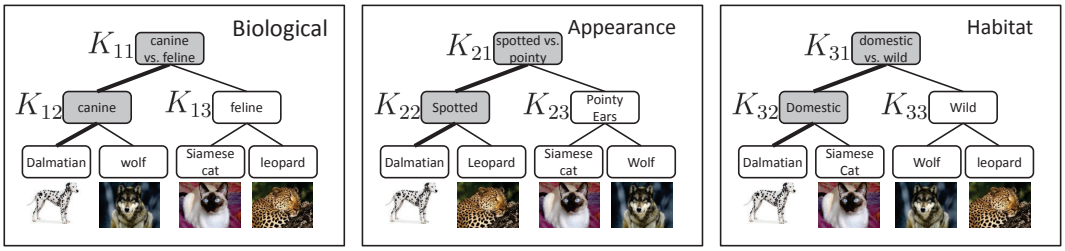
In addition to this semantic gap problem, there exists another problem: no single semantic taxonomy is perfect, and learning an optimal one is infeasible since different applications and views would prefer different groupings. How can we then overcome this inevitable limitation with a single semantic

taxonomy? The next chapter will explore this question.

Chapter 5

Combining Complementary Information in Multiple Taxonomies

In the previous chapter, we have seen how a semantic taxonomy can be used to help category recognition by providing information to isolate granularity-specific features, and to hierarchically classify objects. Two fundamental issues, however, complicate its use. First, a given taxonomy may offer hints about visual relatedness, but its structure need not entirely align with useful splits for recognition. (For example, *monkey* and *dog* are fairly distant semantically according to WordNet, yet they share a number of visual features. An *apple* and *applesauce* are semantically close, yet are easily separable with basic visual features.) Thus, the hierarchical structure provided by a semantic taxonomy is often non-optimal for hierarchical classification. Second, given the complexity of visual objects, it is highly unlikely that some *single* optimal semantic taxonomy exists to lend insight for recognition. While previous work relies on a single taxonomy out of convenience, in reality objects can be organized along many semantic dimensions or "views". (For example, a *Dalmatian* belongs to the same group as the *wolf* according to a biological taxonomy, as both are canines. However, in terms of visual attributes, it can be grouped with the *leopard*, as both are spotted; in terms of habitat, it can be grouped



$$K_{dalmatian} = B_{11}K_{11} + B_{12}K_{12} + B_{21}K_{21} + B_{22}K_{22} + B_{31}K_{31} + B_{32}K_{32}$$

Figure 5.1: **Main idea:** For a given set of classes, we assume multiple semantic taxonomies exist, each one representing a different "view" of the inter-class semantic relationships. Rather than commit to a single taxonomy—which may or may not align well with discriminative visual features—we learn a tree of kernels for each taxonomy that captures the granularity-specific similarity at each node. Then we show how to exploit the inter-taxonomic structure when learning a combination of these kernels from multiple taxonomies (i.e., a "kernel forest") to best serve the object recognition tasks.

with the *Siamese cat*, as both are domestic. See Figure 5.1.)

Motivated by these issues, I next present a discriminative feature learning approach that leverages *multiple* taxonomies capturing different semantic views of the object categories[1]. The key insight here is that some combination of the semantic views will be most informative to distinguish a given visual category. Continuing with the sketch in Figure 5.1, that might mean that the first taxonomy helps learn dog- and cat-like features, while the second taxonomy helps elucidate spots and pointy corner features, while the last reveals context cues such as proximity to humans or indoor scene features. While each view differs in its implicit human-designed splitting criterion, all separate

[1]The work introduced in this chapter is published in [56].

some classes from others, thereby lending (often complementary) discriminative cues. Thus, rather than commit to a single representation, we aim to inject pieces of the various taxonomies as needed.

To this end, I propose *semantic kernel forests*. This novel kernel learning method takes as input training images labeled according to their object category, as well as a series of taxonomies, each of which hierarchically partitions those same labels (object classes) by a different semantic view. For each taxonomy, we first learn a tree of semantic kernels: each node in a tree has a Mahalanobis-based kernel optimized to distinguish between the classes in its children nodes. Following on ToM approach from the previous chapter, the kernels in one tree isolate image features useful at a range of category granularities. Then, using the resulting semantic kernel forest from all taxonomies, we apply a form of multiple kernel learning (MKL) to obtain class-specific kernel combinations, in order to select only those relationships relevant to recognize each object class. We introduce a novel hierarchical regularization term into the MKL objective that further exploits the taxonomies' structure. The output of the method is one learned kernel per object class, which we can then deploy for one-versus-all multi-class classification on novel images.

The main contribution of the work introduced in this chapter is to simultaneously exploit multiple semantic taxonomies for visual feature learning. Whereas past work focuses on building object hierarchies for scalable classification [113, 28] or using WordNet to gauge semantic distance [71, 98, 37, 26], we learn discriminative kernels that capitalize on the cues in diverse taxonomy

views, leading to better recognition accuracy. The primary technical contributions are i) an approach to generate semantic base kernels across taxonomies, ii) a method to integrate the complementary cues from multiple suboptimal taxonomies, and iii) a novel regularizer for multiple kernel learning that exploits hierarchical structure from the taxonomy, allowing kernel selection to benefit from semantic knowledge of the problem domain.

I demonstrate my approach with challenging images from the Animals with Attributes and ImageNet datasets [65, 27] together with taxonomies spanning cognitive synsets, visual attributes, behavior, and habitats. The results show that the taxonomies can indeed boost feature learning, letting us benefit from humans' perceived distinctions as implicitly embedded in the trees. Furthermore, I show that interleaving the forest of multiple taxonomic views leads to the best performance, particularly when coupled with the proposed novel regularization.

5.1 Approach

I cast the problem of learning semantic features from multiple taxonomies as learning to combine kernels. The base kernels capture features specific to individual taxonomies and granularities within those taxonomies, and they are combined discriminatively to improve classification, weighing each taxonomy and granularity only to the extent useful for the target classification task.

I describe the two main components of the approach in turn: construct-

ing the base kernels from the learned tree of metrics on each taxonomy—which we call a *semantic kernel forest* (Sec. 5.1.1), and learning their combination across taxonomies (Sec. 5.1.2), where we devise a new hierarchical regularizer for MKL.

In what follows, we assume that we are given a labeled dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{n=1}^{\mathsf{N}}$ where $(\boldsymbol{x}_i, y_i)$ stands for the ith instance (feature vector) and its class label is y_i, as well as a set of tree-structured taxonomies $\{\mathcal{T}_t\}_{t=1}^{\mathsf{T}}$. Each taxonomy $\mathcal{T}_t$ is a collection of nodes. The leaf nodes correspond to class labels, and the inner nodes correspond to superclasses—or, more generally, *semantically meaningful groupings of categories*. We index those nodes with double subscripts tn, where t refers to the tth taxonomy and n to the nth node in that taxonomy. Without loss of generality, we assign the leaf nodes (i.e., the class nodes) a number between 1 and C, where C is the number of class labels.

5.1.1 Learning a semantic kernel forest

The first step is to learn a forest of base kernels. These kernels are granularity- and view-specific; that is, they are tuned to similarities implied by the given taxonomies. While base kernels are learned *independently* per taxonomy, they are learned *jointly* within each taxonomy, as we describe next.

Formally, for each taxonomy $\mathcal{T}_t$, we learn a set of Gaussian kernels for the superclass at every internal node tn for which $n \geq \mathsf{C} + 1$. The Gaussian

kernels are parameterized as

(5.1)

$$K_{tn}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\{-\gamma_{tn} d^2_{\boldsymbol{M}_{tn}}(\boldsymbol{x}_i, \boldsymbol{x}_j)\} = \exp\{-\gamma_{tn}(\boldsymbol{x}_i - \boldsymbol{x}_j)^{\mathrm{T}} \boldsymbol{M}_{tn}(\boldsymbol{x}_i - \boldsymbol{x}_j)\},$$

where the Mahalanobis distance metric $\boldsymbol{M}_{tn}$ is used in lieu of the conventional Euclidean metric. Note that for leaf nodes where $n \leq \mathsf{C}$, we do not learn base kernels.

We want the base kernels to encode similarity between examples using features that reflect their respective granularity in the taxonomy. Certainly, the kernel K_{tn} should home in on features that are helpful to distinguish the node tn's subclasses. Beyond that, however, we specifically want it to use features that are *as different as possible* from the features used by its ancestors. Doing so ensures that the subsequent combination step can choose a sparse set of "disconnected" features.

To that end, we apply our Tree of Metrics (ToM) technique introduced in the previous chapter to learn the Mahalanobis parameters $\boldsymbol{M}_{tn}$. To recap, In ToM, metrics are learned by balancing two forces: i) discriminative power and ii) a preference for different features to be chosen between parent and child nodes. The latter exploits the taxonomy semantics, based on the intuition that features used to distinguish more abstract classes (dog vs. cat) should differ from those used for finer-grained ones (Siamese vs. Persian cat).

Briefly, for each node tn, the training data is reduced to $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_{in})\}$, where y_{in} is the label of n's *child* $\boldsymbol{x}_i$. If $\boldsymbol{x}_i$'s class label y_i is not a descendant

of the superclass at the node n, then $\boldsymbol{x}_i$ is excluded from $\mathcal{D}_n$. The metrics are learned jointly, with each node mutually encouraging the others to use non-overlapping features. ToM achieves this by augmenting a large margin nearest neighbor [112] loss function $\sum_n \ell(\mathcal{D}_n; \mathcal{M}_{tn})$ with the following *disjoint sparsity regularizer*:

$$\Omega_d(\boldsymbol{M}) = \lambda \sum_{n \geq \mathsf{C}+1} \mathsf{Trace}[\boldsymbol{M}_{tn}] + \mu \sum_{n \geq \mathsf{C}+1} \sum_{m \sim n} \|\mathsf{diag}(\boldsymbol{M}_{tn}) + \mathsf{diag}(\boldsymbol{M}_{tm})\|_2^2, \tag{5.2}$$

where $m \sim n$ denotes that node m is either an ancestor or descendant of n. The first part of the regularizer encourages sparsity in the diagonal elements of $\boldsymbol{M}_{tn}$, and the second part incurs a penalty when two different metrics "compete" for the same diagonal element, i.e., to use the same feature dimension. The resulting optimization problem is convex and can be solved efficiently [55].

After learning the metrics $\{\boldsymbol{M}_{tn}\}$ in each taxonomy, we construct base kernels as in eq. (5.1). The bandwidths γ_{tn} are set as the average distances on training data. We call the collection $\mathcal{F} = \{K_{tn}\}$ of all base kernels the *semantic kernel forest*. Figure 5.1 shows an illustrative example.

While ToM has shown promising results in learning metrics in a single taxonomy, its reliance on linear Mahalanobis metrics is inherently limited. A straightforward convex combination of ToMs would result in yet another linear mapping, incapable of capturing nonlinear inter-taxonomic interactions. In contrast, our kernel approach retains ToM's granularity-specific features but also enables nontrivial (nonlinear) combinations, especially when coupled with a novel hierarchical regularizer, which I will define next.

5.1.2 Learning class-specific kernels across taxonomies

Base kernels in the semantic kernel forest are learned jointly *within* each taxonomy but independently *across* taxonomies. To leverage multiple taxonomies and to capture different semantic views of the object categories, we next combine them discriminatively to improve classification.

In the following, I first describe a basic form of combining. I then describe our novel hierarchical regularization to incorporate semantic and structural knowledge in the combining process.

Basic setting To learn class-specific features (or kernels), we compose a one-versus-rest supervised learning problem. Additionally, instead of combining all the base kernels in the forest $\mathcal{F}$, we pre-select a subset of them based on the taxonomy structure.

Specifically, from each taxonomy, we select base kernels that correspond to the nodes on the path from the root to the leaf node class. For example, in the Biological taxonomy of Figure 5.1, for the category *Dalmatian*, this path includes the nodes (superclasses) CANINE and ANIMAL. Thus, for class c, the linearly combined kernel is given by

$$F_c(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sum_t \sum_{n \sim c} \beta_{ctn} K_{tn}(\boldsymbol{x}_i, \boldsymbol{x}_j), \tag{5.3}$$

where $n \sim c$ indexes the nodes that are ancestors of c, which is a leaf node (recall that the first C nodes in every taxonomy are reserved for leaf class nodes). The combination coefficients β_{ctn} are constrained to be nonnegative to ensure the positive semidefiniteness of the resulting kernel $F_c(\cdot, \cdot)$.

We apply the kernel $F_c(\cdot,\cdot)$ to construct the one-versus-rest binary classifier to distinguish instances from class c from all other classes. We then optimize $\boldsymbol{\beta}_c = \{\beta_{ctn}\}$ such that the classifier attains the lowest empirical misclassification risk. The resulting optimization (in its dual formulation) is analogous to standard multiple kernel learning [8]:

$$\begin{aligned} \min_{\boldsymbol{\beta}_c} \max_{\boldsymbol{\alpha}_c} \quad & \sum_i \alpha_{ci} - \frac{1}{2} \sum_i \sum_j \alpha_{ci}\alpha_{cj} q_{ci} q_{cj} F_c(x_i, x_j) \\ \text{s.t.} \quad & \sum_i \alpha_{ci} q_{ci} = 0, \quad 0 \le \alpha_{ci} \le C, \quad \forall \ \ i, \end{aligned} \tag{5.4}$$

where $\boldsymbol{\alpha}_c$ is the Lagrange multipliers for the binary SVM classifier, C is the regularizer for the SVM's hinge loss function, and $q_{ci} = \pm 1$ is the indicator variable of whether or not $\boldsymbol{x}_i$'s label is c.

Hierarchical regularization Next, we extend the basic setting to incorporate richer modeling assumptions. We hypothesize that kernels at higher-level nodes should be preferred to lower-level nodes. Intuitively, higher-level kernels relate to more classes, thus are likely essential to reduce loss.

We leverage this intuition and knowledge about the relative priority of the kernels from each taxonomy's hierarchical structure. We design a novel structured regularization that prefers larger weights for a parent node compared to its children. Formally, the proposed MKL-H regularizer is given by:

$$\Omega(\boldsymbol{\beta}_c) = \lambda \sum_{t,n\sim c} \beta_{ctn} + \mu \sum_{t,n\sim c} \max(0, \beta_{ctn} - \beta_{ctp_n} + 1). \tag{5.5}$$

The first part prefers a sparse set of kernels. The second part (in the form of hinge loss) encodes our desire to have the weight assigned to a node n be less

than the weight assigned to the node's parent p_n. We also introduce a margin of 1 to further increase the difference between the two weights.

Hierarchical regularization was previously explored in [7], where a mixed $(1, 2)$-norm is used to regularize the relative sizes between the parent and the children. The main idea there is to discard children nodes if the parent is not selected. Our regularizer is somewhat similar in spirit, but we devise a simpler and more computationally efficient formulation. (Despite our complexity advantage, preliminary results do not indicate [7] has any empirical advantage over ours.)

5.1.3 Numerical optimization

The learning problem is cast as a convex optimization that balances the discriminative loss in equation 5.4 and the regularizer in equation 5.5:

$$\min_{\boldsymbol{\beta}_c} f(\boldsymbol{\beta}_c) = g(\boldsymbol{\beta}_c) + \Omega(\boldsymbol{\beta}_c), \quad \text{s.t.} \quad \boldsymbol{\beta}_c \geq 0, \tag{5.6}$$

where we use the function $g(\boldsymbol{\beta})$ to encapsulate the inner maximization problem over $\boldsymbol{\alpha}_c$ in equation 5.4.

We use the projected subgradient method to solve eq. (5.6), for its ease of implementation and practical effectiveness [13]. Specifically, at iteration t, let $\boldsymbol{\beta}_c^t$ be the current value of $\boldsymbol{\beta}$. We compute $f(\boldsymbol{\beta}_c)$'s subgradient $\boldsymbol{s}_t$, then perform the following update,

$$\boldsymbol{\beta}_c^{t+1} \leftarrow \max\left(0, \quad \boldsymbol{\beta}_c^t - \alpha_t \boldsymbol{s}_t\right), \tag{5.7}$$

where the max( ) function implements the projection operation such that the update does not fall outside of the feasible region $\boldsymbol{\beta}_c \geq 0$. For step size α_t, we use the modified Polyak step size rule.

Subgradient Update Rule $g(\boldsymbol{\beta_c})$ encapsulates the inner maximization problem over $\boldsymbol{\alpha}_c$ in eq.(5.4), and is a differentiable function of $\boldsymbol{\beta}_c$ where ∂g is given as follows:

$$\frac{\partial g}{\partial \beta_{ctn}} = -\frac{1}{2} \sum_{ij} \alpha_{ci} \alpha_{cj} q_{ci} q_{cj} F_{ctn}(x_i, x_j) \tag{5.8}$$

The computation of $\partial g / \partial \beta_{ctn}$ only depends on the the α, which is the solution of Eq.(5.4), that could be obtained using an off-the shelf SVM solver. We solve this using LIBSVM [19]. The second term of $f(\boldsymbol{\beta}_c)$, $\Omega(\boldsymbol{\beta}_c)$ is non-differentiable but convex. Thus, its subgradients with respect to $\boldsymbol{\beta}_c$ exist, and defined as,

$$\partial \Omega(\boldsymbol{\beta}_{ctn}) = \lambda + \mu \left(r_{ctnp(n)} - \sum_{k \in C(tn)} r_{ctkn} \right) \tag{5.9}$$

,where $C(tn)$ is the set of children node of tn. $r_{ctij} = 1$ if $\beta_{cti} \geq \boldsymbol{\beta}_{ctj} - 1$ and 0 otherwise. From the subgradient rule $\partial f = \partial g + \partial \Omega$ is a subgradient for f. After obtaining the subgradient ∂f, we could use the following update rule using the modified Polyak's stepsize rule to minimize f to its direction.

$$\boldsymbol{\beta}_c^{t+1} \leftarrow \max\left(0, \quad \boldsymbol{\beta}_c^t - \frac{f(\boldsymbol{\beta}_c^t) - \hat{f}_t + \delta}{\|\boldsymbol{\partial f}(\boldsymbol{\beta_c})_t\|_2^2} \boldsymbol{\partial f}(\boldsymbol{\beta_c})_t\right) \tag{5.10}$$

where the max( ) function implements the projection operation such that the update does not fall outside of the feasible region $\boldsymbol{\beta}_c \geq 0$. $\hat{f}_t$ is an estimate of the optimal value of the objective function and is defined as

$$\hat{f}_t = \min_{0 \leq j \leq t} f(\boldsymbol{\beta}_c^j) \tag{5.11}$$

The variable δ is a constant controlling how close the update rule converges to the optimum. We set it such that in about 500 iterations, the update converges.

5.2 Experiments

We validate our approach on multiple image datasets, and compare to several informative baselines.

5.2.1 Image datasets

We use three datasets taken from two publicly available image collections: Animals with Attributes (AWA) [65] and ImageNet [27][2]. We form two datasets from AWA (Figure 3.2). The first consists of the four classes shown in

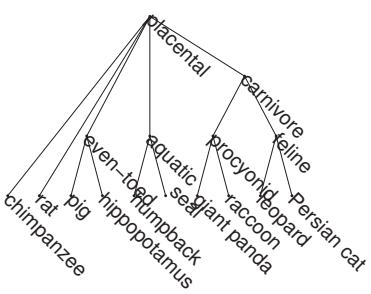
[2] `attributes.kyb.tuebingen.mpg.de/` and `image-net.org/challenges/LSVRC/2011/`

Figure 5.2: Example images for ImageNet-20 dataset

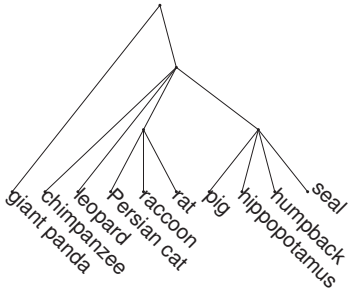
Fig. 5.1, and totals 2,228 images; the second contains the ten classes in [65], and totals 6,180 images. We refer to them as **AWA-4** and **AWA-10**, respectively. The third dataset, **ImageNet-20** (Figure 5.2), consists of 28,957 total images spanning 20 classes from ILSVRC2010. We chose classes that are non-animals (to avoid overlap with AWA) and that have attribute labels [88].

5.2.2 Taxonomies

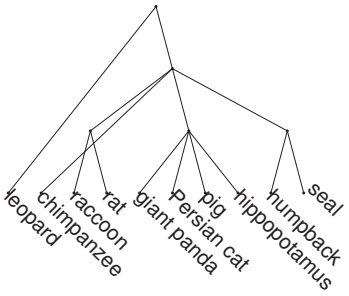
To obtain multiple taxonomies per dataset, we use attribute labels and WordNet. As discussed above, attributes are human understandable properties shared among object classes, e.g., *furry*, *flat*, *carnivorous* [65]. AWA and ImageNet have 85 and 25 attribute labels, respectively. To form semantic taxonomies based on attributes, we first manually divide the attribute labels into subsets according to their mutual semantic relevance (e.g., *furry* and



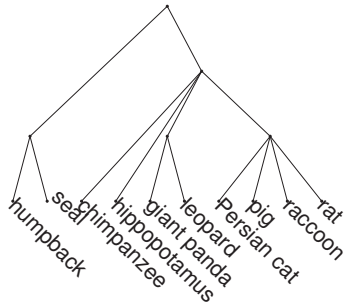
(a) WordNet



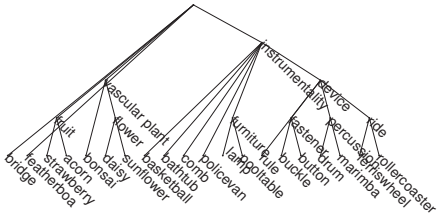
(b) Appearance



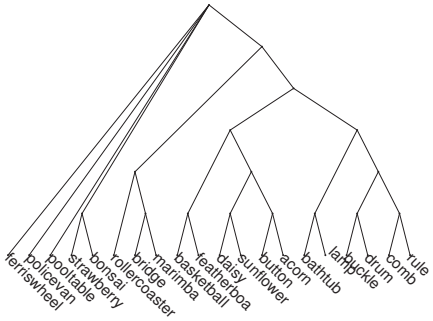
(c) Behavior



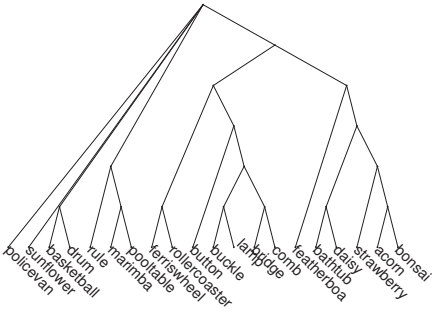
(d) Habitat



(e) Wordnet



(f) Appearance



(g) Attributes

Figure 5.3: Taxonomies for the AWA-10 (a-d) and ImageNet-20 (e-g) datasets.

shiny are attributes relevant for an Appearance taxonomy, while *land-dwelling* and *aquatic* are relevant for a Habitat taxonomy.). Then, for each subset of attributes, we perform agglomerative clustering using Euclidean distance on vectors of the training images' real-valued attributes. We restrict the tree height (6 for ImageNet and 3 for AWA) to ensure that the branching factor at the root is not too high. To extract a WordNet taxonomy, we find all nodes in WordNet that contain the object class names on their word lists, and then build a hierarchy by pruning nodes with only one child and resolving multiple parentship.

For AWA-10, we use 4 taxonomies: one from WordNet, and three based on attribute subsets reflecting Appearance, Behavior, and Habitat ties. For ImageNet-20, we use 3 taxonomies: one from WordNet, one reflecting Appearance as found by hierarchical clustering on the visual features, and one reflecting Attributes using annotations from [88]. For the AWA-4 taxonomies, we simply generate all 3 possible 2-level binary trees, which, based on manual observation, yield taxonomies reflecting Biological, Appearance, and Habitat ties between the animals. See Figures 5.1 and 5.3.

I stress that these taxonomies are created *externally with human knowledge*, and thus they inject perceived object relationships into the feature learning problem. This is in stark contrast to prior work that focuses on optimizing hierarchies for efficiency, without requiring interpretability of the trees themselves [50, 113, 28, 41].

The two image datasets we employ are annotated with both object la-

Dataset	Group Name	Attributes
AWA-10	Appearance	black, white, blue, brown, gray, orange, red, yellow, patches, spots, stripes, furry, hairless, toughskin, big, small, bulbous, lean, flippers, hands, hooves, pads, paws, longleg, longneck, tail, chewteeth, meatteeth, buckteeth, strainteeth, horns, claws, tusks
	Behavior	smelly, flys, hops, swims, tunnels, walks, fast, slow, strong, weak, muscle, bipedal, quadrupedal, active, inactive, nocturnal, hibernate, agility, fish, meat, plankton, vegetation, insects, forager, grazer, hunter, scavenger, skimmer, stalker
	Habitat	newworld, oldworld, arctic, coastal, desert, bush, plains, forest, fields, jungle, ocean, ground, water, tree, cave
ImageNet-20	-	black, blue, brown, furry, gray, green, long, metallic, orange, pink, rectangular, red, rough, round, shiny, smooth, spotted, square, striped, vegetation, violet, wet, white, wooden, yellow

Table 5.1: Attribute groups used to build each taxonomy for AWA-10 and ImageNet-20. These groups are manually defined based on the available attribute labels and their semantic relationships.

bels and attribute labels. For every image, we have the real-valued attribute presence prediction for each attribute in the vocabulary. That is, for M total attributes, each image has an M-length vector recording the likelihood that each attribute is present in it. Because these attributes are semantically meaningful, we can use them to create a variety of semantic taxonomies. We do this by manually forming subsets of related attributes and then hierarchically clustering the data according to only those (fewer than M) selected attribute dimensions. Each group/subset generates one taxonomy.

To generate the attribute-based taxonomies on the AWA-10 dataset, we manually group M=78 of the total attributes provided with the AWA datset as shown in Table 5.1, and perform agglomerative clustering as discussed in the main text to form the semantic hierarchies.

For ImageNet-20, we perform agglomerative clustering on all 25 attributes shown in the bottom row of Table 5.1. As the attributes for ImageNet-20 are binary, we use ℓ_1-distance when grouping them, and limit the tree height to 6 to avoid having too many branches at the root.

5.2.3 Baseline methods for comparison

We compare our method to three key baselines:

- **Raw feature kernel:** an RBF kernel computed on the original image features, with the γ parameter set to the inverse of the mean Euclidean distance d among training instances.

- **Raw feature kernel + MKL:** MKL combination of multiple such RBF kernels constructed by varying γ, which is a traditional approach to generate base kernels (e.g., [8]). For this baseline, we generate the same number N of base kernels as in the semantic kernel forest, with $\gamma = \frac{\sigma}{d}$, for $\sigma = \{2^{1-m}, \ldots, 2^{N-m}\}$, where $m = \frac{N}{2}$.
- **Perturbed semantic kernel tree:** a semantic kernel tree trained with taxonomies that have randomly swapped leaves.

The first two baselines will show the accuracy attainable using the same image features and basic classification tools (SVM, MKL) as our approach, but lacking the taxonomy insights. The last baseline will test if weakening the semantics in the taxonomy has a negative impact on accuracy.

I evaluate several variants of my approach, in order to analyze the impact of each component:

- **Semantic kernel tree + Avg:** an equal-weight average of the semantic kernels from one taxonomy.
- **Semantic kernel tree + MKL:** the same kernels, but combined with MKL using sparsity regularization only (i.e., $\mu = 0$ in eq. 5.5).
- **Semantic kernel tree + MKL-H:** the same as previous, but adding the proposed hierarchical regularization (eq. 5.5).
- **Semantic kernel forest + MKL:** semantic forest kernels from multiple taxonomies combined with MKL.

- **Semantic kernel forest + MKL-H:** the same as previous, but adding our hierarchical regularizer.

5.2.4 Implementation details

For all results, we use 30/30/30 images per class for training/validation/testing, and generate 5 such random splits. We report average multi-class recognition accuracy and standard errors for 95% confidence interval. For single taxonomy results, we report the average over all individual taxonomies. For all methods, the raw image features are bag-of-words histograms obtained on SIFT, provided with the datasets. We reduce their dimensionality to 100 with PCA to speed up the ToM training, following [55]. To train ToM, we sample 400 random constraints and cross-validate the regularization parameters $\lambda, \gamma \in \{0.1, 1, 10\}$. For MKL/MKL-H, we use $C = 1000$ for the C-SVM parameter, and cross-validate the sparsity and hierarchical parameters $\lambda, \mu \in \{0, 0.1, 1, 10\}$.

5.2.5 Results

Quantitative results Table 5.2 shows the multi-class classification accuracy on all three datasets. Our semantic kernel forests approach significantly outperforms all three baselines. It improves accuracy for 9 of the 10 AWA-10 classes, and 16 of the 20 classes in ImageNet-20 (see Figure 5.4). These gains clearly show the impact of injecting semantics into discriminative feature learning. The forests' advantage over the individual trees supports our core claim

	AWA-4	AWA-10	ImageNet-20
Raw feature kernel	47.67 ± 2.22	30.80 ± 1.36	28.20 ± 1.45
Raw feature kernel + MKL	48.50 ± 1.89	31.13 ± 2.81	27.67 ± 1.50
Perturbed semantic kernel tree	N/A	31.53 ± 2.07	28.20 ± 2.02
Semantic kernel tree + Avg	47.17 ± 2.40	31.92 ± 1.21	28.97 ± 1.61
Semantic kernel tree + MKL	48.89 ± 1.06	32.43 ± 1.93	29.74 ± 1.26
Semantic kernel tree + MKL-H	50.06 ± 1.12	32.68 ± 1.79	29.90 ± 0.70
Semantic kernel forest + MKL	49.67 ± 1.11	34.60 ± 1.78	30.97 ± 1.14
Semantic kernel forest + MKL-H	**52.83 ± 1.68**	**35.87 ± 1.22**	**32.30 ± 1.00**

Table 5.2: Multi-class classification accuracy on all datasets, across 5 train/test splits. (The perturbed semantic kernel tree baseline is not applicable for AWA-4, since all possible groupings are present in the taxonomies.)

regarding the value of interleaving semantic cues from multiple taxonomies. Further, the proposed hierarchical regularization (MKL-H) outperforms the generic MKL, particularly for the multiple taxonomy forests.

I stress that semantic kernel forests' success is *not* simply due to having access to a variety of kernels, as we can see by comparing our method to both the raw feature MKL and perturbed tree results—all of which use the same number of kernels. Instead, the advantage is leveraging the implicit discriminative criteria embedded in the external semantic groupings. Interestingly, the perturbed taxonomies show some improvement over the raw feature kernel on AWA-10, but not on ImageNet-20. We attribute this difference to the fact that for fine-grained classes like those in AWA-10 (all animals), almost any grouping of labels may have some semantic meaning, whereas for sparser classes like those in ImageNet-20 (from bridge to acorn), arbitrary perturbations are often meaningless. Thus, the baseline's semantics are weakened more noticeably in

the latter case.

MKL-H has the most impact for the multiple taxonomy forests, and relatively little on the single kernel tree. This makes sense. For a single taxonomy, a single kernel is solely responsible for discriminating a class from the others, making all kernels similarly useful. In contrast, in the forest, two classes are related at multiple different nodes, making it necessary to select out useful views; here, the hierarchical regularizer plays the role of favoring kernels at higher levels, which might have more generalization power due to the training set size and number of classes involved.

The per-class and per-taxonomy comparisons in Figure 5.4 further elucidate the advantage of using multiple complementary taxonomies. A single semantic kernel tree often improves accuracy on some classes, but at the expense of reduced accuracy on others. This illustrates that the structure of an individual taxonomy is often suboptimal. For example, the Habitat taxonomy on AWA-10 helps distinguish *humpback whale* well from the others—it branches early from the other animals due to its distinctive "oceanic" background—but it hurts accuracy for *giant panda*. The WordNet taxonomy does exactly the opposite, improving *giant panda* via the Biological taxonomy, but hurting *humpback whale*. The semantic kernel forest takes the best of both through its learned combination. The only cases in which it fails are when the majority of the taxonomies strongly degrade performance, as to be expected given the linear MKL combination (e.g., see the class *marimba* and *rule*).

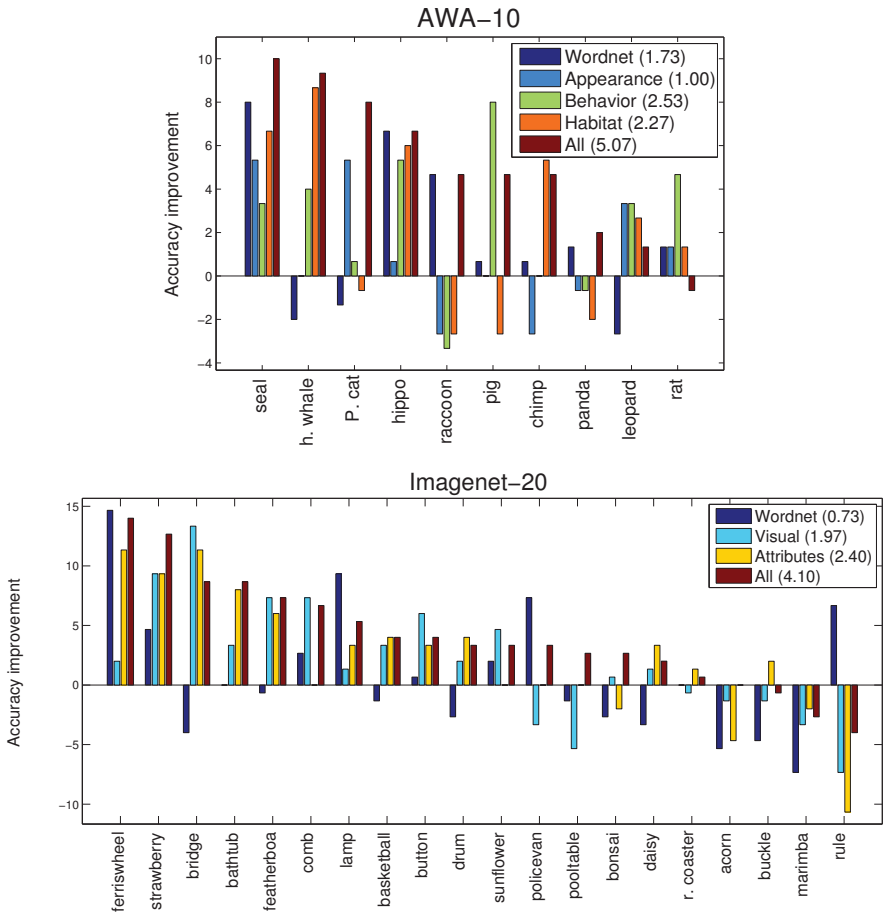
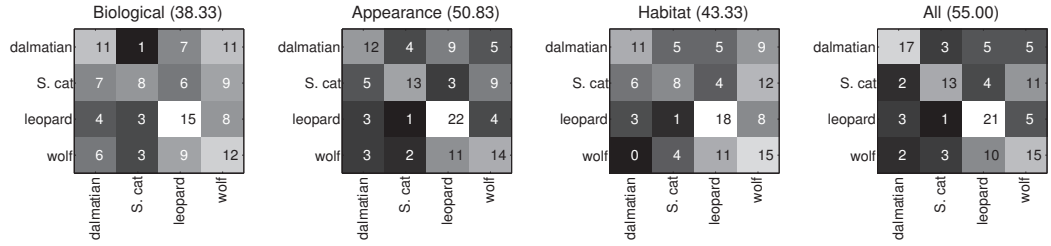


Figure 5.4: Per-class accuracy improvements of each individual taxonomy and the semantic kernel forest ("All") over the raw feature kernel baseline. Numbers in legends denote mean improvement. Best viewed in color.

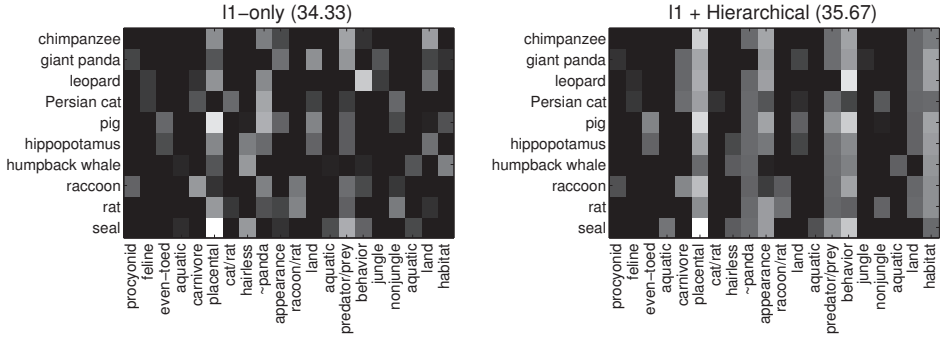
Further qualitative analysis Figure 5.5 (a-d) shows the confusion matrices for AWA-4 using only the root level kernels. We see how each taxonomy specializes the features, exactly in the manner sketched in the chapter introduction. The combination of all taxonomies achieves the highest accuracy (55.00), better than the maximally performing individual taxonomy (Appear-



(a) Biological (b) Appearance (c) Habitat (d) All

Figure 5.5: (a-d): AWA-4 confusion matrices for individual taxonomies (a-c) and the combined taxonomies (d). Y-axis shows true classes; x-axis shows predicted classes.

ance, 50.83).



(a) MKL (b) MKL-H

Figure 5.6: Example β_k's to show the characteristics of the two regularizers. Each entry is a learned kernel weight (brighter=higher weight). Y-axis shows object classes; x-axis shows kernel node names.

Figure 5.6 shows the learned kernel combination weights β_k for each class k in AWA-10, using the two different regularizers. In Figure 5.6 (a), the ℓ_1 regularizer selects a sparse set of useful kernels. For example, the *humpback whale* drops the kernels belonging to the whole Behavior taxonomy block, and gives the strongest weight to "hairless", and "habitat". However, by failing to

select some of the upper-level nodes, it focuses only on the most confusing fine-grained problems. In contrast, with the proposed regularization Figure 5.6 (b), we see more emphasis on the upper nodes (e.g., the "behavior" and "placental" kernels), which helps accuracy.

5.3 Discussion

In this chapter, I proposed a semantic kernel forest approach to learn discriminative visual features that leverage information from multiple semantic taxonomies. The results show that it improves object recognition accuracy, and they give good evidence that committing to a single external knowledge source is insufficient.

The key novelty here is that the proposed method tackles the difficult problem of merging complementary information in different semantic views, by first isolating features at each granularity and then assembling the learned sub-feature spaces in a single pool, with the sparsity and hierarchical regularization to enable interleaving and enforce a structure among the features.

The remaining problems are on how to better combine the kernels, as the current additive kernel combination might not capture the strong similarity in a single view. While the proposed MKL method with the hierarchical regularizers is shown to already significantly improve the classification performance, it could still potentially benefit from using a non-additive, non-linear combination multiple kernel learning method.

Until now, we have focused on semantic knowledge from *attributes* and *taxonomies*, and I have shown how to leverage them in ways different from existing models. In the next chapter, I will show how to exploit analogies, a new type of semantic knowledge for visual recognition, to regularize a discriminative categorization model.

Chapter 6

Transferring Knowledge between Related Category Pairs with Analogies

The attributes and taxonomies covered in earlier chapters provided ways to relate categories, which provided structures in the learned categorization models. However, information provided by both of these semantic knowledge types are limited to *pairwise* class similarities, defined by sharing or non-sharing of object properties. For example, two categories either share some attributes [58], or two categories in different semantic levels in parent-child relationship compete [55] for exclusive properties at each level. In other words, these pairwise similarity-driven models can only provide information on whether two categories are similar or dissimilar, and higher-order *reasoning* employed in the human recognition process is limited in these models.

In the final component of my thesis, I aim to move beyond per-class semantic relatedness, and exploit higher-order relationships jointly involving multiple classes. Specifically, I propose to model *analogies* between classes in the form "p is to q, as r is to s" (or, in shorthand, $p : q = r : s$)[1] . An analogy encodes the relational similarity between two pairs of semantic concepts. By

[1]The work introduced in this chapter is published in [57].

augmenting labeled data instances with a set of semantic analogies during training, we aim to enrich the learned representation and thereby improve generalization. Analogies can be defined with almost arbitrary abstraction, ranging from "is-a" relationships (*dog* : *canine* = *cat* : *feline*), to contextual dependencies (*fish* : *water* = *bird* : *sky*). To examine analogies most likely to benefit visual learning, we restrict our focus to *analogical proportions* [74]—analogies between pairs of concrete objects in the same semantic universe and with similar abstraction level.

Before sketching my approach, I want to first motivate why this form of analogy should offer new information to a learning algorithm. As any standardized test-taker knows, analogies are used to gauge both vocabulary skills and reasoning ability. Notably, the pairs of entities involved in an analogy need not share properties. For example, in the analogy *planet* : *sun* = *electron* : *nucleus*, the *planet* and *electron* do not have anything in common; rather, the relational similarity (*orbiter* and *center*) is what makes us recognize the two pairs as parallel in meaning [44]. Furthermore, the common difference exhibited by the two pairs in an analogy may encapsulate a combination of multiple properties—and that combination need not have a succinct semantic name. For example, in the analogy *leopard* : *cat* = *wolf* : *dog*, the common difference relating the two pairs entails multiple low-level concepts; in both, the first class *lives in the wild*, *has fangs*, and *is more aggressive*, etc. Thus, to master analogies, one must not only estimate the similarity of words, but also infer the abstract relationships implied by their pairings.

Accordingly, we expect analogies to benefit a feature learning algorithm in ways that semantic distance constraints alone cannot. Whereas existing methods inject only "vocabulary skills" by requiring that semantically related instances be close and semantically unrelated ones be far, our method will also inject "reasoning ability" by requiring that the common differences implied by analogies be reflected in the learned semantic feature space. Often, the higher-order constraints may connect quite distant sets of categories. The analogies can thus facilitate a form of transfer from class pairs that are more easily discriminated in the original feature space to analogous class pairs that are not. For example, suppose *leopard* and *cat* are often confused in the visual space because the training set consists of only close-up images, whereas *dog* and *wolf* are easily separable due to their distinct backgrounds. Enforcing the analogy constraint *leopard* : *cat* = *wolf* : *dog* could make the separation in the first pair clearer, by aligning it with the same hypothetical semantic axis of differences (*wild/fanged/aggressive*) shared by the second (more distinctive) pair.

I propose an *Analogy-preserving Semantic Embedding* (ASE), which embeds features discriminatively with analogies-based structural regularization. Given a set of analogies involving various object categories, we translate each one into a geometric constraint called an *analogical parallelogram*. This constraint states that the difference between the first pair of categories should be the same as the that between the second pair, where each category is represented by a (learned) prototype vector in some hypothetical semantic space.

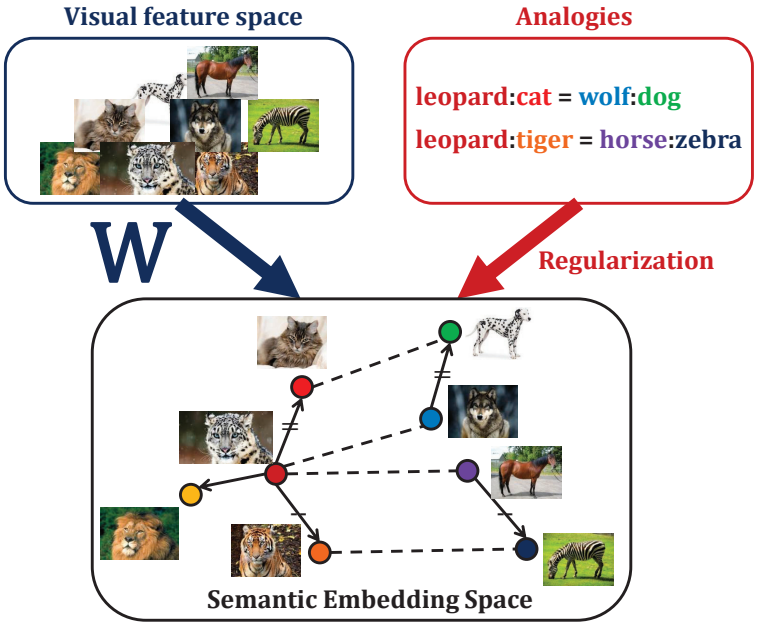


Figure 6.1: Concept of the analogy-preserving semantic embedding (ASE). I introduce analogical parallelogram constraints to regularize a semantic embedding. By learning from both labeled instances and analogies, the learned embedding space preserves structural similarities between category pairs.

See Figure 6.1. We represent the constraints as a novel regularizer that augments a large-margin label embedding. Consequently, we obtain an embedding where examples with the same label are mutually close (and far from differently labeled points) and analogical parallelograms have nearly parallel sides.

The learned embedding can be used for recognition, automatic analogy completion, visualization, and potentially other tasks. To use it for recognition, we project a novel image into the learned space, and predict its label based on the nearest category prototype. We further show how to automatically discover and prioritize useful analogies, which is valuable to concentrate

on constraints that are influential for recognition.

Compared to traditional large-margin label embeddings [113, 11], our approach preserves a new form of relational similarity. While the prior methods also map to a space where semantic similarities are preserved, they risk learning spurious associations between features and labels. Our analogy-induced regularizer mitigates such adverse effects by constraining the hypothesis space with structural relations between category pairs, yielding robust models with better generalization. Even constraints not in the axes of visual properties can be helpful, as they shift the focus from brittle incidental correlations to higher-order semantic ties.

6.1 Analogy-preserving Semantic Embedding (ASE)

In this section, I will present the analogy-preserving semantic embedding for categorization, which learns to place category embeddings (prototypes) in a low-dimensional semantic space, while also preserving the analogical structure between matched category pairs in the analogies.

6.1.1 Encoding analogies

For each class $c \in \mathcal{Y}$, $\boldsymbol{u}_c \in \mathbb{R}^{\mathsf{M}}$ denotes its coordinates in the M-dimensional semantic space. Each $\boldsymbol{u}_c$ can be thought of as a prototype for the category; we will explain how the prototypes are optimized jointly with the data projection matrix $\boldsymbol{W}$ in Sec. 6.1.3.

An analogy involves four categories, and we represent the relationship

with an ordered quadruplet $(p, q, r, s) \in \mathcal{Y} \times \mathcal{Y} \times \mathcal{Y} \times \mathcal{Y}$. As we focus on *analogical proportions* [74], the difference between p and q is equated with the difference between r and s. Moreover, the difference between p and r also is equated with the difference between q and s.

Analogical proportions naturally induce geometric constraints among the embeddings of the four categories in the semantic space. In particular, the geometry is characterized by a parallelogram; we will show how to exploit this structure in our learning algorithm.

Analogy parallelogram We use the vector shift $(\boldsymbol{u}_q - \boldsymbol{u}_p)$ to represent the difference between the two categories q and p in the semantic space. Note that this difference is directed, that is, $\boldsymbol{u}_q - \boldsymbol{u}_p \neq \boldsymbol{u}_p - \boldsymbol{u}_q$. The analogical proportion implied by (p, q, r, s) is thus encoded by the following pair of equalities:

$$\boldsymbol{u}_q - \boldsymbol{u}_p = \boldsymbol{u}_s - \boldsymbol{u}_r, \quad \text{and} \quad \boldsymbol{u}_r - \boldsymbol{u}_p = \boldsymbol{u}_s - \boldsymbol{u}_q. \tag{6.1}$$

These constraints form a parallelogram in which each vertex is a category, as illustrated in Figure 6.2.

Convex regularizer There are several ways of enforcing the analogical proportion constraints in equation 6.1. A natural choice is to exploit the parallel property of opposing sides. Specifically, the normalized inner products between opposing sides are the cosine of their intersection degree, which should be 1 if perfectly parallel. Concretely, for an analogy $\alpha = (p, q, r, s)$, the resulting parallelogram "score" would be defined as

$$S(\alpha) = \frac{1}{2}\left(\frac{(\boldsymbol{u}_q - \boldsymbol{u}_p)^{\mathrm{T}}(\boldsymbol{u}_r - \boldsymbol{u}_s)}{\|\boldsymbol{u}_q - \boldsymbol{u}_p\| \cdot \|\boldsymbol{u}_r - \boldsymbol{u}_s\|} + \frac{(\boldsymbol{u}_r - \boldsymbol{u}_p)^{\mathrm{T}}(\boldsymbol{u}_s - \boldsymbol{u}_q)}{\|\boldsymbol{u}_r - \boldsymbol{u}_p\| \cdot \|\boldsymbol{u}_s - \boldsymbol{u}_q\|}\right). \tag{6.2}$$

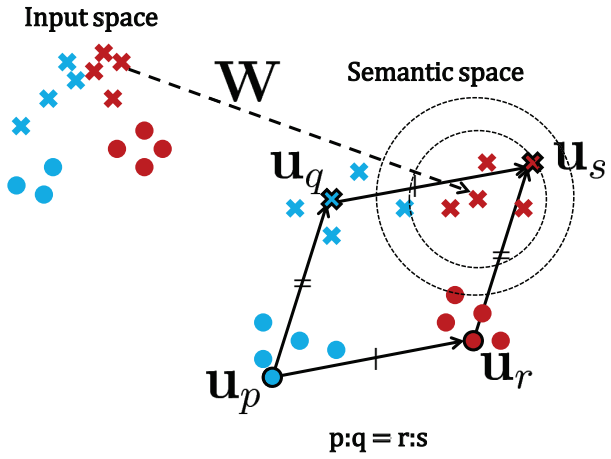


Figure 6.2: Geometry of ASE. **Analogy constraints for the semantic category embedding:** The analogy quadruplet (p, q, r, s) forms a parallelogram in the semantic embedding space, cf. eq. (6.1). **Data embedding $\boldsymbol{W}$:** At the same time, when projected onto the semantic space by $\boldsymbol{W}$, the data point $\boldsymbol{x}_i$ from class q should be closer to its semantic category embedding $\boldsymbol{u}_q$, compared to any other category embedding, by a large margin (see dotted circles).

While intuitive, maximizing the parallelogram score (or equivalently, minimizing its negative) is computationally inconvenient, since it is not convex in the embeddings $\boldsymbol{u}$. Thus, we use a relaxed version and compare the sides only in their *lengths*. Specifically, our regularizer is defined as

(6.3)

$$R(\alpha) = 1/\sigma_1 \|(\boldsymbol{u}_q - \boldsymbol{u}_p) - (\boldsymbol{u}_r - \boldsymbol{u}_s)\|_2^2 + 1/\sigma_2 \|(\boldsymbol{u}_r - \boldsymbol{u}_p) - (\boldsymbol{u}_s - \boldsymbol{u}_q)\|_2^2,$$

where σ_1 and σ_2 are two scaling constants used to prevent either pair of sides from dominating the other. We simply estimate them as the mean distances between data instances from different classes.

$R(\alpha)$ is convex in the embedding coordinates. Moreover, it is straight-

forward to kernelize as it depends only on the distances (and thus inner products).

6.1.2 Automatic discovery of analogies

Human knowledge is a natural source for harvesting analogy relationships among categories. However, it is likely expensive to completely rely on human assessment to acquire a sufficient number of analogies for training. To address this issue, we use *auxiliary* semantic knowledge to identify candidate analogies.

In the context of visual object recognition, visual *attributes* are an appealing form of auxiliary semantic knowledge [65]. Attributes are binary predicates shared among certain visual categories—for example, the category *panda* has the "true" value for the *spotted* attribute and the "false" value for the *orange* attribute. Supposing we have access to attribute descriptions stating the typical attribute values for each category, we can automatically discover plausible analogies.

I next define two strategies to do so. The first is independent of the data instances, while the second exploits the instances to emphasize analogies more likely to lend discriminative information.

Attribute-based analogy discovery Our first strategy is to view attributes as a proxy to the embedding coordinates of the visual categories in the semantic space we are trying to learn. In the attribute space, each category is encoded with a binary vector, with bits set to one for attributes the class does possess,

and bits set to zero for attributes the class does not possess. Note that this is a class-level description—we have one binary vector per object class.

Imagine that we enumerate all quadruplets of visual categories. For each quadruplet α, we compute its parallelogram score according to equation 6.2, using the categories' attribute vectors as coordinates. We then select top-scoring quadruplets as our candidate analogies.

Pragmatically, we can only score a subset of all possible analogies for a large number of visual categories. Thus, to ensure good coverage, for each randomly selected pivot category p, we select at most K triplets of other categories, where K is far fewer than the total number of possible ones. We also remove equivalent analogies. For example, (p, q, r, s) is equivalent to (p, r, q, s) or other shift-invariant forms.

We will use the highest-scoring analogies to augment the class-labeled data when learning the embedding. We stress that while we discover analogies based on parallelogram scores computed in the space of attribute descriptions, we regularize the learned embedding according to parallelogram scores computed in the learned embedding coordinates (cf. Sec. 6.1.3). Thus, external semantics drive the "training" analogies, which in turn mold our learned semantic space.

Discriminative analogy discovery The process described thus far has two possible issues. First, it does not take the data instances into consideration. While our goal is to find a *joint* embedding space for both data instances

and category labels, analogies inferred purely from attributes do not necessarily align the data and mid-level representations—they might even lead to conflicting embedding preferences! Secondly, being fully unsupervised, this procedure need not discover analogies directly useful to our classification task. In particular, the extracted candidate analogies are not indicative of whether two categories are easily distinguishable or confused.

I address both issues with an intuitive and empirically very effective heuristic. Mindful of our goal (described in the introduction) of improving discrimination for *confusable* categories by leveraging analogy relationships connecting those confusing categories to *easily distinguishable* categories, we first use baseline classifiers to estimate the pairwise confusability between categories. This step can be achieved easily with any off-the-shelf multi-way classifier and visual features computed from the training instances. The confusability between two categories p and q is defined in terms of the resulting misclassification error:

$$C_{pq} = 0.5\left[\epsilon_{p\to q} + \epsilon_{q\to p}\right],$$

where $\epsilon_{p\to q}$ is the rate of misclassifying instances from the category p as the category q, and likewise for $\epsilon_{q\to p}$.

Our next step is to refine the candidate analogies generated above by finding those with *unbalanced confusability*. Specifically, for each analogy $\alpha = (p, q, r, s)$, we compute its discrimination potential:

$$P(\alpha) = \left|\log(1 + C_{pq}) - \log(1 + C_{rs})\right|. \tag{6.4}$$

Algorithm 3 Discriminative analogy generation

Require: $\boldsymbol{R}^{c\times N_s}$, $\boldsymbol{S}$
Ensure: A set of analogies: A
Initialize $A = \phi$.
while $|A| \leq M$ **do**
  Select random category $p \in \{1, \ldots, c\}$
  Generate K quadruplets $a_k = (p, q_k, r_k, s_k), 1 \leq k \leq K$
  Compute $P(a_k)$ according to 6.4, for all $k \in K$.
  Sort $\{a_1, \ldots, a_K\}$ by $P(a_k) \rightarrow \{a_{s(1)}, \ldots, a_{s(K)}\}$.
  while $A^* \cap A = \phi$ **do**
    Find $a_k^* = \arg\max\{C(\boldsymbol{S}, a_{s(1)}), \ldots, C(\boldsymbol{S}, a_{s(\kappa)})\}$ $(\kappa \ll K)$.
    Set A^* as the all possible rotations of a_k^*.
  end while
  $A = A \cup \{a_{k^*}\}$
end while
return $\boldsymbol{A}$

This score attains its maximum when C_{pq} and C_{rs} are drastically different—that is, if one is 0 and the other is 1. We use this score to re-rank the K candidate analogies generated for each category p. Intuitively, we seek the quadruplet where one pair of categories is easily distinguishable (based on the image data) while the other pair is difficult to differentiate. Precisely by enforcing their analogy relationship, we expect the easy pair to assist discrimination for the difficult one.

Algorithm 3 shows the details of this analogy generation considering confusability.

To summarize, our automatic discovery of analogies is a two-phase strategy. We first use an auxiliary semantic space to identify a set of candidate analogies where the four categories are highly likely to form a parallelogram.

Then, we analyze misclassification error patterns of these categories and use the scoring function in equation 6.4 to determine the potential of each analogy in improving classification performance. We describe next how to use the highest-scoring analogies to learn the joint embedding of both features and categories.

6.1.3 Discriminative learning of the ASE

Next I explain how we regularize a discriminative embedding to account for the analogies.

Large margin-based discrimination We aim to learn a projection matrix $\boldsymbol{W} \in \mathbb{R}^{\mathsf{M}\times\mathsf{D}}$ to map each data instance (image example) $\boldsymbol{x}_i$ into the semantic space, giving its M-dimensional coordinates $\boldsymbol{z}_i = \boldsymbol{W}\boldsymbol{x}_i$.[2] The ideal projection matrix $\boldsymbol{W}$ should make $\boldsymbol{z}_i$ close to its corresponding label's embedding $\boldsymbol{u}_{y_i}$ and distant to all other labels' embeddings [113][3]. Specifically, we enforce the large margin constraint for every training instance,

$$\|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_{y_i}\|_2^2 + 1 \leq \|\boldsymbol{W}\boldsymbol{x}_i - \boldsymbol{u}_c\|_2^2 + \xi_{ic}, \quad \forall \;\; c \neq y_i \tag{6.5}$$

where $\xi_{ic} \geq 0$ is a slack variable for satisfying the separation by the margin of 1.

Regularization To jointly embed both features and class labels, we regularize

[2]Nonlinear embeddings are possible via kernelization.

[3]We use 1 instead of the inter-class dissimilarity as the large margin to maximize class separation.

so that the class labels in the analogy set $\mathcal{A}$ form parallelograms as much as possible. The regularizer is given by

$$R_{total}(\mathcal{A}) = \sum_a \omega_a R(\alpha_a), \tag{6.6}$$

which is the weighted sum of the regularization defined in eq. (6.3) for each analogy α_a. If using the "raw" attribute-based analogies, the weight $\omega_a = S(\alpha_a)$, thus enforcing stricter regularization for category quadruplets whose structure is closer to a "perfect" analogy. If using discriminatively discovered analogies, the weight is instead $\omega_a = P(\alpha_a)$, thus prioritizing those that are more discriminative.

Additionally, we also constrain the parameters $\boldsymbol{W}$ and all $\boldsymbol{u}_c$ with their Frobenius norms: $\|\boldsymbol{W}\|_F^2$ and $R(\boldsymbol{u}) = \sum_c \|\boldsymbol{u}_c - \boldsymbol{u}_c^{\text{PRIOR}}\|_2^2$. In particular, for the class label embeddings, we constrain them to be close to our prior knowledge on their locations $\boldsymbol{u}_c^{\text{PRIOR}}$. The prior knowledge could be null such that we set $\boldsymbol{u}_c^{\text{PRIOR}}$ to zeroes. Or, the class label embeddings could be computed from auxiliary information, for example, the multi-dimensional embedding of class labels where the dissimilarities between labels are measured with tree distances from a taxonomy [113] or attributes. We consider both in the results.

6.1.4 Numerical optimization

Our learning problem is thus cast as the following optimization problem:

$$\min_{\boldsymbol{W},\{\boldsymbol{u}_c\}} \quad \sum_{ic} \xi_{ic} + \lambda R_{total}(\mathcal{A}) + \mu \|\boldsymbol{W}\|_F + \tau R(\boldsymbol{u}) \tag{6.7}$$

subject to both the large margin constraints in equation 6.5 and non-negativity constraints on the slack variables ξ_{ic}. The regularization coefficients λ, μ, and τ are determined via cross-validation.

The optimization is nonconvex due to the quadratically-formed large margin constraints. We have developed two methods for solving it. Our first method uses stochastic (sub)gradient descent, where we update $\boldsymbol{W}$ and $\boldsymbol{u}_c$ according to their sub-gradients computed on a subset of instances. Despite its simplicity, this method works well in practice and scales better to problems with many categories.

We also consider a convex relaxation analogous to the procedure in [113]. Briefly, in equation 6.7, we hold $\{\boldsymbol{u}_c\}$ fixed first and solve $\boldsymbol{W}$ in closed-form, $\boldsymbol{W} = \boldsymbol{U}\boldsymbol{Q}$ where the matrix $\boldsymbol{U}$ is composed of $\{\boldsymbol{u}_c\}$ as column vectors. The matrix $\boldsymbol{Q}$ depends only on $\boldsymbol{x}_i$ and is constant with respect to $\boldsymbol{U}$ or $\boldsymbol{W}$. Substituting the solution of $\boldsymbol{W}$ into both the objective function equation 6.7 and the large margin constraints equation 6.5, we can reformulate the optimization in terms of $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}$. In particular, the original non-convex large margin constraints in $\boldsymbol{U}$ can be relaxed into convex if we reparameterize $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{U}$ as a positive semidefinite matrix $\boldsymbol{V}$. We then solve $\boldsymbol{V}$ and recover the solutions $\boldsymbol{U}$ and $\boldsymbol{W}$, respectively. For cases where D is much larger than the number of categories, we expect this variant to optimize faster.

The details of the numerical optimization for the semidefinite programming relaxation problem are provided in Algorithm 4. At each step, we update the gradient by ηs_t, where η is a general learning rate and s_t is a step size

Algorithm 4 ASE (Convex)

Require: training data $(\boldsymbol{x}_n, \boldsymbol{y}_n)$, analogical quadruplets A
Ensure: $\boldsymbol{Q}$, $\boldsymbol{w}$
Initialize $\boldsymbol{Q} = I$, and $\boldsymbol{w}$ by setting each element of $\boldsymbol{w}$ with $1/M$
$\boldsymbol{U} = \boldsymbol{J}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T + \lambda\boldsymbol{I})^{-1}$
$\bar{\boldsymbol{X}} = \boldsymbol{U}\boldsymbol{X}$
while $t < T$ and $||Q|| > \epsilon$ **do**
    $\boldsymbol{G}_t^{\xi} = 1/N \sum_{i=1}^{N} (\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j)(\bar{\boldsymbol{x}}_i - \bar{\boldsymbol{x}}_j)^T$
    $\boldsymbol{G}_t^{mds} = 2 * (\boldsymbol{Q} - \boldsymbol{Q}_{mds})$
    Compute gradient $\boldsymbol{G}_t^a$ from equation 6.3.
    $\boldsymbol{G}_t = \boldsymbol{G}_t^{\xi} + \mu\boldsymbol{G}_t^{mds} + \gamma\boldsymbol{G}_t^a$
    $\boldsymbol{Q}_{t+1} = \boldsymbol{Q}_t - \eta s_t \boldsymbol{G}_t$ with stepsize s_t.
end while
return $\boldsymbol{V} = decomp(\boldsymbol{Q})$
return $\boldsymbol{W} = \boldsymbol{V}\boldsymbol{U}$

specified by some step size rule. We learn η on the validation set, and set s_t according to Polyak's step size rule.

6.2 Results

We validate three aspects: 1) the effectiveness of our analogy discovery approach; 2) recognition accuracy when incorporating discovered analogies in learning embeddings; and 3) "fill in the blank"—a Graduate Record Examination (GRE)-style prediction task of filling in the category that would form a valid analogy.

Datasets and implementation details We use three datasets created from two public image datasets: Animals with Attributes (AWA), which contains 50 animal classes [65] and ImageNet, which contains general object categories [27]. They were chosen due to their available attribute descriptions and their chal-

lenging diverse content. From AWA, we create two datasets: **AWA-10** of $6,180$ images from 10 classes [65], and the complete 50-class **AWA-50** of $30,475$ images. From ImageNet, we use the 50-class **ImageNet-50** with annotated attributes [88], totaling $70,380$ images.

We use the features provided by the authors, which consist of SIFT and other texture and color descriptors. We use PCA to reduce the feature dimensionality to $\mathsf{D} = 150$ for efficient computation. Additionally, we augment ImageNet-50 with attribute labels for colors, material, habitat, and behaviors (e.g., *big*, *round*, *feline*), yielding 39 and 85 binary attributes for ImageNet and AWA, respectively. We fix $\mathsf{K} = 10,000$. We use the convex relaxation, since the dimensionality is much greater than the number of classes; accordingly, the semantic space dimensionality M equals the number of categories (10 or 50).

6.2.1 Automatic discovery of analogies

In real-world settings, acquiring all analogies from manual input may be costly and impractical. Thus, we first examine the analogies discovered by our method (Sec. 6.1.2), which assumes only that attribute-labeled object classes are available.

Figure 6.3 displays several examples for AWA-50 and ImageNet-50. Most analogies are intuitive to understand. For example, in the second row of *collie:dalmatian = lion:leopard*, the categories *collie* and *lion* are both furry and brown, while the categories *dalmatian* and *leopard* are both spotted and

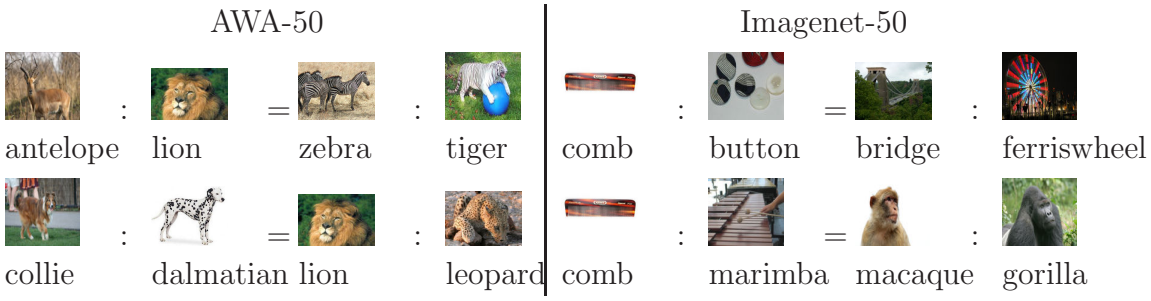


Figure 6.3: Example analogies discovered from attributes.

lean. We also see that the analogies can be largely visual (e.g., the third row), an upshot of the many visually relevant attributes offered with the datasets.

6.2.2 Visual recognition with ASE

We compare the classification performance of our Analogy-preserving Semantic Embedding (ASE) to the following baselines, all of which lack analogies:

(1) **SVM-RBF**: Multiclass SVM with RBF kernel.

(2) **Large margin embedding (LME)**: The existing technique of [113] without the taxonomy prior regularizer, which is also a special case of our approach where we disable both the attributes prior and analogy regularizers by setting $\tau = 0$ and $\lambda = 0$ in eq. (6.7). For this baseline, the class label embeddings are constrained only to satisfy the large margin separation criterion of eq. (6.5);

(3) **Large margin embedding with attributes prior (LME$^{\text{prior}}$)**: This baseline adds the prior regularizer to LME, where we adjust τ for eq. (6.7) via cross-validation. In particular, we use the multi-dimensional scaling (MDS)

embedding of class labels where the pairwise dissimilarity is the Euclidean distance between the *attribute* vectors of two classes. The contrast between LME and LME$^{\text{PRIOR}}$ reveals how useful attributes as auxiliary semantic information are in yielding discriminative embeddings, separating out the impact of attributes from the impact of analogies.[4]

All embedding methods classify novel images according to the nearest category $\boldsymbol{u}_c$ in the embedding space.

For our method, we include two variants, differentiated only by how the analogies are discovered, cf. Sec. 6.1.2. In ASE-A, the analogies are derived solely from attributes, aiming to preserve parallelograms as much as possible. In ASE-C, the analogies are derived from the discrimination-based discovery, aiming to use distinct categories to assist confusable categories. The confusability among categories is measured using the baseline LME classifier on the validation set.

In our experiments, all hyperparameters (regularization coefficients, kernel function parameters) are tuned via cross-validation. We use 30 examples per class for both training and testing, and use another 30 images as a validation set to learn the parameters. We report the average results over 5 such random splits.

How do analogies affect recognition accuracy? We first validate our method on multiclass classification. Since the analogies help preserve the in-

[4]We also tested LME using WordNet class distances as a prior as in [113], but found it inferior to the attribute prior.

Dataset	AWA-10	AWA-50	ImageNet-50
#. analogies	5	50	50
SVM-RBF	43.00 ± 1.94	19.32±0.57	15.37±0.93
LME	44.40±2.83	19.65±0.90	16.52±1.10
LME$^{\text{PRIOR}}$	44.93 ± 3.57	20.12±1.03	16.59±0.39
ASE-A (ours)	45.47±3.10	20.60±0.93	17.08±0.36
ASE-C (ours)	**45.93±2.90**	**21.05±0.82**	**17.24±0.62**

Table 6.1: Multiclass classification accuracy. The numbers denote mean and the standard error over 5 runs.

trinsic semantic structure among objects, we expect the learned space to show better generalization power, and hence improved object categorization.

Table 6.1 shows the results.[5] We report the optimal number of analogies selected from preliminary experiments, though the results were in general insensitive to the number of analogies. On all three datasets, we observe clear improvement using our analogy-preserving embedding variants over both LME variants.

We see that the difference in accuracy for LME and LME$^{\text{PRIOR}}$ is in general smaller than the improvement from LME to ASE. This suggests that using attribute distances *alone* as a prior to constrain embeddings (as LME$^{\text{PRIOR}}$ does) is not sufficient. In contrast, in ASE, the prior and the analogy constraints work together, leading to a noticeable improvement.

Which types of analogies should we use? We also observe that our

[5]Attribute-based categorization [65] underperforms all baselines (AWA-10: 28.80 ± 1.51, AWA-50: 17.80 ± 0.90, ImageNet-50: 11.14 ± 0.80).

ASE-C variant outperforms ASE-A. This coincides with our intuition that the analogies would be much more helpful for discrimination if a pair of easily confusable categories can leverage a pair of easily distinguishable categories.

Detailed analysis supports this intuition even more strongly. Figure 6.4 compares the amount of reduction in confusability among the 10 classes of AWA-10, from $\text{LME}^{\text{PRIOR}}$ to either ASE-A (left) or ASE-C (right). We observe that for ASE-A, the improvement is made on pairs that are not included in the analogies; in contrast, for ASE-C, the improvements are mostly made on pairs that *are* included in analogies. This noticeable correlation between the category pairs selected for analogies, and the pairs whose confusion is reduced (for ASE-C) suggests that our consideration of the pairwise confusion is indeed the reason ASE-C outperforms ASE-A, whose analogies do not care about the data distribution.

Figure 6.5 shows projections of AWA-50 categories to a $2D$ space using different embedding methods. We see that the quadrilaterals formed by the four categories projected by ASE involved in each analogy do indeed show distinct parallelogram shape. In contrast, the existing LME approach variants do not maintain the desired relational similarity.

6.2.3 Completing a visual analogy

Finally, we subject our method to a GRE test. Given $p : q = r :?$, how well can our method fill in the blank, based on its representation of the three other classes? In this analogical reasoning task, which is performed by

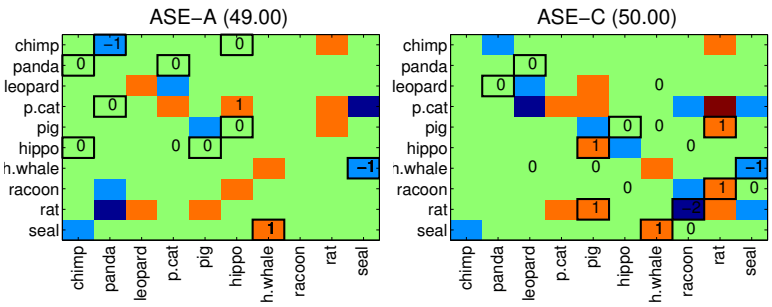
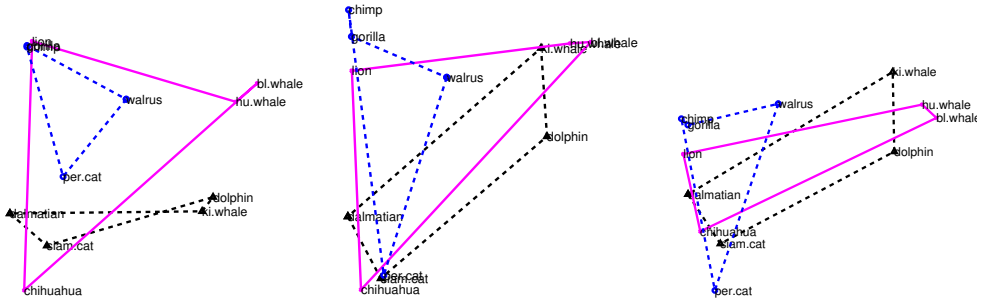


Figure 6.4: Confusion reduction. **Left:** $C_{\text{LME}} - C_{\text{ASE-A}}$, **Right:** $C_{\text{LME}} - C_{\text{ASE-C}}$. The numbers and colors at each entry show the reduction in confusion (red:↑, blue:↓). Outlined entries are pairs that appear in the training analogies. Positive off-diagonal entries indicate reduced confusion. ASE-C focuses on initially confused classes.



(a) LME (b) LME$^{\text{PRIOR}}$ (c) ASE

Figure 6.5: AWA-50 categories projected to the 2D space using each embedding method. We show only three analogies for ease of viewing: 1) *dalmatian:siamese cat=killer whale:dolphin*, 2) *lion:chihuahua = humpbackwhale:bluewhale*, and 3) *chimp:gorilla = persian cat:walrus*.

Dataset	AWA-10		AWA-50		Imagenet-50	
k	1	3	1	3	1	3
Chance	14.29	42.86	2.13	6.38	2.13	6.38
LME	36.00	52.00	4.80	12.40	1.60	7.20
$\text{LME}^{\text{PRIOR}}$	52.00	68.00	5.60	14.40	0.80	6.80
ASE-A	**64.00**	**88.00**	**8.40**	**20.80**	2.80	6.40
ASE-C	60.00	80.00	5.20	15.60	**3.20**	**8.80**

Table 6.2: Top-k class prediction accuracy, given an analogy with an unknown class in the form *p:q=r:?*

Analogy question	LME	$\text{LME}^{\text{PRIOR}}$	ASE-A
AWA-50			
leopard:lion = dalmatian:?	bobcat	s. monkey	fox
horse:g.shepherd = sheep:?	weasel	antelope	collie
skunk:mouse = killerwhale:?	fox	bluewhale	dolphine
Imagenet-50			
badger:skunk = button:?	g.spider	bathtub	buckle
marimba:rule = baboon:?	kitfox	orangutan	patas
b.ball:bathtub = r.coaster:?	jaguar	pooltable	bridge

Table 6.3: Sample analogy completion results

virtually every graduate school applicant, the learning algorithm is given a set of complete analogies $\mathcal{A}^{\text{TRAIN}}$. Then it is given a disjoint test set of analogies $\mathcal{A}^{\text{TEST}}$, each of which has its fourth category missing. No analogies overlap in (p, q, r) between the two sets. To fill in the blank with ASE or LME, we simply rank each category according to its parallelogram score when its $\boldsymbol{u}_c$ is used as the fourth category. The more parallelogram-like, the more it appears to be the right answer. The ground truth answer is the class maximizing the parallelogram score according to the auxiliary attribute ground truth.

Our hypothesis is that by learning to discriminate categories *in conjunction with* preserving the analogy constraints in $\mathcal{A}^{\text{TRAIN}}$, the learned semantic embedding will generalize well to complete the novel analogies, without resorting to auxiliary information.

Table 6.2 strongly supports our hypothesis. We report the prediction accuracy averaged over 5 random trials, where we take the classes with the top k parallelogram scores as guesses. ASE-A achieves the best accuracy, followed by ASE-C. They both outperform the LME methods, which lack analogical constraints. On AWA-10, we predict the right completion in the first guess ($k = 1$) 64% of the time. There is clearly room for improvement, though, as accuracy decreases substantially for all methods on the larger 50-class datasets. Table 6.3 shows example completed analogies for AWA-50. Compared to LME, ASE selects more intuitive classes to fill in the missing values.

6.3 Discussion

In this chapter, I introduced a novel semantic embedding method for visual data that preserves structural similarities in the form of analogies. In addition to formulating a novel regularizer suitable for our goal, I also explored ways to systematically discover plausible analogies from auxiliary attribute information. The proposed method improves recognition accuracy over an existing "distance-only" embedding approach, thanks to its ability to preserve higher-order structures and facilitate transfer between easier and harder pairs of objects. Beyond benefiting recognition, I showed that it also allows analogy

completion—a high-level reasoning task.

As this is the first work on using analogies in the context of categorization, there remains some issues to be addressed. First, in this work, I explored a specific and popularly recognized form of analogy that is defined between two pairs of categories (*A:B = C:D*). However, in more general sense, an analogy could exist between *sets* of entities where the relation between the elements in the first set is the same as the relation between the elements in the second set; for example, we can have an analogy between sets of parts in a human and a car, as in *eye:ear:leg = headlight:side mirror:wheel*. This suggests the possibility of using of a more general form of analogy, where an analogy exists between subgraphs containing multiple elements, to which we might be able to apply similar geometric constraints as we used in this work.

Another issue is that the current model works on the static set of analogies already provided by humans, but the acquisition of the analogies could be done dynamically, in an active learning framework. The active learning criterion in this case might consider selecting *confusing* category pair and we might ask the users to provide a category pair that best aligns with the selected category pair.

I believe that this exploitation of higher-order semantic knowledge is a meaningful step in the research of the semantic models for categorization. However, at the same time, this is also just a first step into the relatively unexplored world of human reasoning, in the context of machine learning and visual categorization. In the next section, I will discuss more about the possible

research topics and ideas in this direction.

Chapter 7

Future Work

My thesis work opens up a number of interesting future directions. One possible direction is to pursue a unified framework that can incorporate all proposed semantic models, to benefit from complementary information provided by each model. Another possible direction is to explore more complex semantic relations. I have already covered one such semantic type, the analogy, but there could be much more arbitrarily complex relations between the categories and semantic concepts. Further, the current semantic models are rather shallow regardless of the complexity of the original semantic knowledge; that is, the category recognition process does not involve any deep *reasoning*. This also suggests a viable future research direction—increasing the complexity of the semantic models. Finally, we could explore ways of tackling scalability issues with the semantic models. While my focus is not on the scalability or the efficiency in training or recognition, most of the proposed models in this thesis are already scalable to some degree, and can be further improved with the introduction of efficient optimization schemes and parallelization. However, what I am more interested in is a semantic categorization model that targets directly to reduced complexity. In the following sections, I will briefly mention issues and challenges with each of these possible directions, and will

sketch some rough ideas.

7.1 Unified framework for different types of semantic knowledge

Until now, the target discriminative categorization models we aimed to improve using the semantic information took slightly different forms: I used an SVM for the object-attribute feature sharing idea, metric learning for the tree of metrics, metric learning and multiple kernel learning for the semantic kernel forests, and a large margin embedding method for the analogy-preserving semantic embedding. While all can be seen as models for feature learning, their specific differences stem from practical considerations about the optimal learning model for each type of semantic knowledge. For attributes, the most suitable categorization model is a binary classifiers as attributes are binary. For semantic taxonomies with arbitrary branching factors, which makes learning 1-vs-all classifier less scalable, a class-agnostic model such as a metric learning is more suitable. For analogies, a large margin embedding was the most suitable, as such a shared common subspace model allows to impose geometrical constraints among the categories.

However, it is a viable question to ask what might happen when all these different semantic models are combined into a single categorization model. One important aspect of the semantic models proposed so far suggests a possiblity of designing a unified semantic model, which is that regardless of what semantic knowledge is used, the learning problem can be expressed as the

generic regularized learning problem (Equation 1.2). This generic model requires the learning of a shared feature space regularized by the semantic knowledge. Thus, the objective of learning a semantic space is common throughout the different semantic methods, where the only difference between them being what to preserve in the learned space. As the different types of semantic knowledge provide complementary information from one another, my expectation is that a unified model might outperform each of the separate semantic models proposed. However, there remains some pragmatic challenges in building such a unified model.

The first challenge is to how to generalize over the different types of semantic knowledge. A careful observation on the types of semantic knowledge we have dealt with so far reveals some relations among them. An attribute can be viewed and modeled as a three-level taxonomy where the root contains two supercategories as its children, one having the attribute and the other not. On a taxonomy, the semantic criteria to split or group the categories at each node could be thought of attributes, or further, each superclass could be thought of an attribute in a general sense. In an analogy, the relational similarity, or the common displacements between the two category pairs could be a combination of attributes. These observations suggest the possibility of having a unified model for different types of semantic knowledge. A general model in logical forms might be able to incorporate all these different types of relations.

The second challenge is to decide which discriminative learning model to use for the unified semantic model, as transferring the structural constraints

from one learning model to anther is not straightforward. For example, it is not clear how the feature sharing behavior in object-attribute feature sharing model might transfer well to a hierarchical metric learning model, nor how the parallelogram constraints in the analogy-preserving semantic embedding can be applied to a hierarchical metric learning model.

Addressing these challenges in theory and practice will be an interesting future work.

7.2 Learning from more complex semantic relations

Other directions I aim to pursue in the research of semantic models are: 1) further exploring more complex relations beyond analogies, and 2) exploring a better way to model the semantics, which could possibly result in complex models.

Attributes and taxonomies are two types of semantic relations, that is either *proprietary* (A car has wheels, i.e. attributes), or *inclusive* (A car is also a vehicle, i.e. taxonomy), where the inferred implicit relations from these semantic sources are modeled either as the regularization term that promotes feature *sharing*, or *competition*. However, there could be even more complex relations among the categories and semantic concepts that could be exploited, as well as better means to model the interactions.

The last chapter on analogy showed the first work to explore a higher-order relation, which are relational similarities between two pairs of categories.

However, there could be arbitrarily complex semantic relations between categories and concepts, such as causal, anti-causal, contextual, and so on, which all influence the human recognition process of the categories. In the next subsections, I suggest some ideas to move further in that direction.

7.2.1 Exploiting first-order logical formulas

The proprietary and inclusive relations we have sought, and even analogical relations, could be viewed as special instances of a more general logical relation between semantic entities. For example, for attributes, some abstract, high-level attributes such as *fast*, *predator*, *plankton* can be only be inferred from more visual and lower level concepts such as *longleg (fast)*, *meatteath, claws (predator)*, and *oceanic (plankton)*, and there certainly exist some relations between these semantic concepts. Modeling such complex semantic relations has potential advantages for object categorization. While mostly ignored in the field of statistical machine learning where most object recognition method rely on, many AI researchers have focused on the logical relation [100, 43], as it is a key factor in the human intelligence [14]. Humans can recognize objects by inferring from the relations of the observation to the known semantic knowledge.

Then how should we learn such arbitrarily complex relations? One idea is to focus on the first-order logic as a means to model such relations. Suppose that we are given a set of logical formulas where each category and attribute is a variable in the first order formula. For example, consider examples of

the class dalmatian. A dalmatian is also a dog, and it has spots. Also, a dog is not wild. This could be expressed as $\forall x(dalmatian(x) \rightarrow dog(x))$, $\forall x(dalmatian(x) \rightarrow spots(x))$, and $\forall x(dog(x) \rightarrow wild(x))$ with a first order formula. The relation between visual attributes associated in a first-order-logic formula is different from a simple correlation in that is asymmetric. Not all that have feathers and beak can fly (ostrich, penguin) but a flying animal will definitely have feathers and beak. Not all animals that are spotted are dalmatians, but all dalmatians are spotted. Further, instead of all attributes having some correlation to each other, the first order logic only describes the relation between a lower level visual, concrete attributes (longleg) and a higher-level attributes (fast), and the lower level attributes are only associated by the high-level attributes. This allows for more accurate modeling of semantics than a correlation model.

The scenario where such logical relation would work best, is in a transductive, or semi-supervised learning case, where only partial observations are available. Instead of working with fully annotated datasets, we could use the image tags to infer unobserved attributes from the observed attributes through the predefined logical relationships. For image annotation, where the focus is on generating semantics based on the inference on the other semantics [89], it has shown some success. This gives us a straightforward way to make use of such learned logical formula. We first learn a logical formula to make them to generate new semantic labels, and use the augmented labels to learn the mappings from the visual feature space, and the label space to the semantic

space, and then can learn the classifier on the projected semantic space to classify the instances by their category label.

However, a better way might be to enforce the first-order logical relationship into the learned model directly, in a form of a regularization term that puts penalty based on the categorization models for each attributes and classes involved in the logical relations, which still needs more research.

7.2.2 A deeper semantic model

Due to the success of the large-margin methods such as support vector machine in classification with the support of theoretically well-established kernel methods, categorization models have remained relatively simple, as simple as learning linear separating hyperplanes, and most of the research efforts were made on designing the representations. This strategy works well when the category set is small in number of categories and is sparse, that is, when the distribution of categories has little overlap. However, as the size of the dataset increases to hundreds of millions and the focus of the categorization is moving to a fine-grained categorization where there is often much overlap between categories, this simple approach is no longer optimal and the needs for more complex models arise.

What the current models lack is *reasoning*, based either on the known knowledge of the world or in the given scene context. For example, recognizing that an object is a cup as it contains some liquid, which a person is drinking. Such reasoning requires additional layers over the recognition of the simple

visual attributes. The semantic models such as attributes-based categorization models and my other models that employs transformation (metric, linear embedding), can be understood as providing one layer on top of the original visual feature space. However, this simple one-layer approach might not suffice to model layers of thought that a human semantic model might possess.

The recently spotlighted deep learning approach [54] which goes back to the layered neural network models of the eighties, has gained some success on many of computer vision tasks, including category-level classification. The power of the model comes from more expressive power generated from the layers of non-linear functions, and which can be applied even to the feature learning stage. However, the limitation of this deep learning model is that what is learned in these models does not necessarily mimic the human reasoning process, and thus the what is learned on the models might not be meaningful to humans. Also, there is no explicit ways to incorporate logical inference, or the external semantic knowledge of the world into the deep learning model.

My goal is to come up with a model that overcome the limitations of these two successful approaches. The model I am picturing is a layered semantic model, where each internal unit (or a function) in the network has some semantic meaning, and leads to more abstract concepts as it goes up to higher layers. With this semantic layered model, we might have better understanding of the internal process, and also may have better control of each unit to guide the learning of the model to a desired direction, which I hope will learn a model that is closer to the hypothetical human perception/reasoning

model. How to build such deeper semantic model in detail is an open question, but we might gain some insights from the previous research effort in artificial intelligence, cognitive science, and natural language processing.

7.3 Scalable approaches to object categorization

All of the proposed models are readily applicable to a large-scale problem as they were modeled with scalability in mind, as with the hierarchical model in tree of metrics (ToM) and low-dimensional linear projection in the analogy-preserving semantic embedding (ASE). Other models can be also parallelized with little effort, as the learning of each independent model can be parallelized without any effort, and the only bottleneck is the regularization part. This is a typical problem well-fitted for the popular *map-reduce* system for large-scale systems; we can take care of the independent model learning at the *map* step, and the regularization at the *reduce* step.

However, a better approach would be having an explicit mechanism to reduce the inherent complexity of the problem, by taking advantage of the semantics. One thing to notice is that the human perception/recognition system is fairly scalable. The key is in that the humans understand the categories with some abstraction—either with generic semantic qualities that spans multiple categories (attributes), or with some layers of abstraction (taxonomies); both are scalable and generalizable to large number of categories.

7.3.1 Approximating the whole category space with few categories

As mentioned previously, the power of the semantic models in object categorization comes from the fact that they provide some generic semantic elements that spans multiple categories. These generic semantic elements could be thought of the bases that span the whole category space. The semantic space for categorization then can be modeled using the semantic bases. Another interesting future challenge would be to find few categories that can be used as such semantic bases. In other words, I want to find *representative* categories that can shape the entire category space. This process could be conceptually similar to the Nyström approximation [31] where a large positive semidefinite matrix is approximated with few random columns, but with columns (categories) chosen with some human-defined criteria instead of random sampling.

One example of such criteria might be the 'commonness' of an object category, that is, either how many instances of each object category are available on the web, or how common they are to humans. Another possible criterion could be how iconic an object category is. For example, *tiger* might represent predators, while *dog* might represent pets, and so on. Being able to approximate the entire category with few would mean having to train the model only on these few categories, which provides much efficiency in training and also scalability in terms of number of categories.

7.3.2 Iterative, incremental learning of the categories

Humans do not suddenly learn all the categories in the world on one day. Rather, they gradually expand their knowledge of the world throughout their entire lives. First they learn few categories, and then gradually learn more and more categories. Here, we do not try to learn the novel categories entirely from the scratch, but in the context of, and in relation to the already known categories, through abstraction and correcting the previous learned knowledge if necessary. Such process could be also adopted to the learning of the object categories. We can start from few categories, and learn to place the novel categories in relation to the already learned categories, with some known similarity and structure between the categories. Here, we can start from easy and reliable categories and gradually add harder categories. This could overcome the limitations of the current regularization models, since the regularization will not be symmetric in this way of learning, thus minimizing the effect of unreliable category models influencing a more reliable one.

Prior work in visual recognition also explored iterative category learning either in an unsupervised category discovery setting [48], or for supervised dynamic categorization model learning [118]; however, none of the prior work leveraged external semantics in the process. Using semantic knowledge might give us better hint on which category models to learn next, as each category has different semantic relation to already learned categories.

In machine learning perspective, this potential learning framework would be an instance of lifelong learning [96] where the learner learns from a continu-

ous stream of training examples and tasks, while transferring knowledge from the previous steps to later steps. In my semantic-regularized learning framework, the knowledge transfer could be modeled as asymmetric regularization of the newly learned models according to their semantic relationships to the categorization models learned in previous steps.

Both incremental learning of new categories and modeling asymmetric regularizations in the process are interesting and challenging problems for future work.

Chapter 8

Conclusion

In this thesis, I proposed series of ideas to exploit external semantic knowledge to regularize the learning of discriminative models for object categorization. Specifically, I exploited semantic knowledge such as attributes, taxonomies, and analogies, that inform about the relations between categories, in a way that the relations are encoded as structural regularizations between the independent learning models for each category.

I first started with the exploitation of *attributes* and *taxonomies*. I used the attributes to guide the feature learning for category classifiers, by enforcing the category classifiers to share features with the attributes classifiers. As a result, the category classifier, when discriminatively learned, used the semantic features which were also used by the attributes, and obtained improved classification performance while making semantic predictions.

As for taxonomy, I used it to learn exclusively discriminative feature for each semantic level, by learning a hierarchical tree of metrics (ToM) while regularizing the metrics in parent-child relationship to compete for the features, and to be sparse. The learned models resulted in selecting few informative features that are useful for the classification subproblem at each branch of the

taxonomy, and improved hierarchical classification performance.

Then, I moved on to the problem of working with multiple taxonomies, based on the intuition that there could be multiple semantic taxonomies depending on different semantic views. Using multiple taxonomies, we isolated different semantic features at each node of multiple semantic taxonomies. As hierarchical classification is not straightforward with the multiple taxonomies, I provided a non-hierarchical multiple kernel learning (MKL) approach to combine the learned features, while also considering the hierarchical structure among the features through a hierarchical regularization. The resulting multiple semantic taxonomy MKL model outperformed the single-taxonomy model and the MKL with conventional multi-bandwith basis kernels.

Finally, as a first step in exploring more complex semantic knowledge, I further explored a novel type of semantic source, an *analogy*, which informs about the relational similarities between pairs of categories. I used the analogy to regularize the structure of a discriminatively learned category embedding space, by translating the high-level relational similarity constraints into geometric constraints. The resulting model benefited from the knowledge transfer happening between related pairs of categories, which resulted in improved categorization performance.

In summary, my thesis showed how to seamlessly incorporate the semantic information from the human knowledge about the world, into a learning model, starting from the popular semantic knowledge types, to completely new forms and types of the semantic knowledge. This semantics-regularized learn-

ing approach benefits from the power of the state-of-the-art discriminative learning models for categorization, while also benefiting from better generalization power provided by the semantic guidance using the information about the relationships, which effectively prevents the model from overfitting to the training set biases. As a result, my proposed models obtained improved accuracies over the those of state-of-the art methods, while also making semantically meaningful predictions.

As the proposed method is domain-agnostic that does not assume any domain specific information about the input features, the semantic regularization models proposed in this thesis is not only limited to the application to visual object recognition problem, but also can be applied to any classification problems in other domains that have established domain knowledge over relations between classes. A few possible applications are text-based document classification in natural language processing, or gene-based animal and protein classification problem in computational biology.

I believe that with these completed work, I made an important step forward from the existing semantic approaches which did not result in the improved categorization performance, and worked on conventional types of semantic knowledge such attributes or a single taxonomy. However, at the same time this is only the first step in the venture into the vast human knowledge. The next step will include the exploration of more complex semantic relations between categories and concepts such as logical relations, and also the work on a deeper, layered semantic models that would enable more complex reasoning

to happen. Obtaining practical performance on very large-scale datasets is also essential and I think the abstraction, generalization and analogical reasoning of the human knowledge will be necessary ingredients to solving the problem.

I feel hopeful to have chosen this problem of exploring human knowledge for recognition as my thesis topic, and made some contribution to it, which I believe is the key and the right path to ultimately solving the category recognition problem. Many challenges awaits us, but I believe that time and continuous effort of many will eventually take us to our goal of guiding machine to recognize the world as we humans do.

Bibliography

[1] R. J. Mooney A. Acharya, A. Rawal and E. R. Hruschka. Using both latent and supervised shared topics for multitask learning. In *Proceedings of the European Conference on Machine Learning*, 2013.

[2] M. Pontil A. Argyriou, T. Evgeniou. Convex Multi-task Feature Learning. *Machine Learning*, 73(3):243–272, 2008.

[3] A. Ahmed, K. Yu, W. Xu, Y. Gong, and E. Xing. Training Hierarchical Feed-Forward Visual Recognition Models using Transfer Learning from Pseudo-Tasks. In *Proceedings of the European Conference on Computer Vision*, 2008.

[4] R. Ando and T. Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, 6, 2005.

[5] A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Proceedings of the Neural Information Processing Systems*, 2006.

[6] B. Babenko, S. Branson, and S. Belongie. Similarity functions for categorization: from monolithic to category specific. In *Proceedings of the International Conference on Computer Vision*, 2009.

[7] F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Proceedings of the Neural Information Processing Systems*, 2008.

[8] F. Bach, G. Lanckriet, and M. Jordan. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. In *Proceedings of the International Conference on Machine Learning*, 2004.

[9] K. Barnard, Q. Fan, R. Swaminathan, A. Hoogs, R. Collins, P. Rondot, and J. Kaufhold. Evaluation of localized semantics: data, methodology, and experiments. Technical report, University of Arizona, 2005.

[10] E. Bart, I. Porteous, P. Perona, and M. Welling. Unsupervised learning of visual taxonomies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[11] S. Bengio, J. Weston, and D. Grangier. Label Embedding Trees for Large Multi-Class Task. In *NIPS 2010, Twenty-Fourth Annual Conference on Neural Information Processing Systems*, 2010.

[12] T. Berg, A. Berg, and J. Shih. Automatic Attribute Discovery and Characterization from Noisy Web Data. In *Proceedings of the European Conference on Computer Vision*, 2010.

[13] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[14] M. H. Bickhard and L. Terveen. *Foundational Issues in Artificial Intelligence and Cognitive Science: Impasse and Solution*. Elsevier Science Inc., New York, NY, USA, 1995.

[15] A. Blum and T. Mitchell. Combining Labeled and Unlabeled Data with Co-training. In *COLT: Proceedings of the Workshop on Computational Learning Theory*, 1998.

[16] L. Cai and T. Hofmann. Hierarchical document categorization with support vector machines. In *ACM International Conference on Information and Knowledge Management*, 2004.

[17] R. Caruana. Multitask Learning. *Machine Learning*, 1997.

[18] N. Cesa-bianchi and L. Zaniboni. Hierarchical classification: Combining bayes with svm. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 177–184, 2006.

[19] C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2011.

[20] Q. Chen and S. Sun. Hierarchical large margin nearest neighbor classification. In *International Conference on Pattern Recognition*, 2010.

[21] C. Christoudias, K. Saenko, L. Morency, and T. Darrell. Co-adaptation of audio-visual speech and gesture classifiers. In *International Conference on Multimodal Interaction*, 2006.

[22] C. Cortes, M. Mohri, and A. Rostamizadeh. Learning non-linear combinations of kernels. In *Proceedings of the Neural Information Processing Systems*, 2009.

[23] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[24] K. Crammer and Y. Singer. On the Algorithmic Implementation of Multi-class SVMs. *Journal of Machine Learning Research*, 2001.

[25] J. Davis, B. Kulis, P. Jain, S. Sra, and I. Dhillon. Information-theoretic metric learning. In *Proceedings of the International Conference on Machine Learning*, 2007.

[26] J. Deng, A. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *Proceedings of the European Conference on Computer Vision*, 2010.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-F ei. Imagenet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[28] J. Deng, S. Satheesh, A. Berg, and L. Fei Fei. Fast and balanced: Efficient label tree learning for large scale object recognition. In *Proceedings of the Neural Information Processing Systems*, 2011.

[29] C. Desai, D. Ramanan, and C. Fowlkes. Discriminative models for multi-class object layout. In *Proceedings of the International Conference on Computer Vision*, 2009.

[30] I. S. Dhillon, S. Mallela, and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3(7-8):1265–1287, 2003.

[31] P. Drineas and M. W. Mahoney. On the nyström method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6:2153–2175, December 2005.

[32] R. Duda, P. Hart, and D. Stork. *Pattern Classification*, chapter 10. John Wiley and Sons, Inc., New York, 2 edition, 2001.

[33] S. Dumais and H. Chen. Hierarchical classification of web content. In *Research and Development in Information Retrieval*, 2000.

[34] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785, 2009.

[35] C. Fellbaum, editor. *WordNet An Electronic Lexical Database*. MIT Press, May 1998.

[36] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *Transactions of Pattern Analysis and Machine Intelligence*, 32(9), September 2010.

[37] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *Proceedings of the European Conference on Computer Vision*, 2010.

[38] V. Ferrari and A. Zisserman. Learning Visual Attributes. In *Proceedings of the Neural Information Processing Systems*, 2007.

[39] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In Lorenza Saitta, editor, *Proceedings of the International Conference on Machine Learning*, pages 148–156. Morgan Kaufmann, 1996.

[40] A. Frome, Y. Singer, and J. Malik. Image retrieval and classification using local distance functions. In *Proceedings of the Neural Information Processing Systems*, 2006.

[41] T. Gao and D. Koller. Discriminative learning of relaxed hierarchy for large-scale visual recognition. In *Proceedings of the International Conference on Computer Vision*, 2011.

[42] P. V. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proceedings of the International Conference on Computer Vision*, pages 221–228, 2009.

[43] M. R. Genesereth and N. J. Nilsson. *Logical foundations of artificial intelligence*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1987.

[44] D. Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155 – 170, 1983.

[45] D. Gentner and A. B. Markman. Structure mapping in analogy and similarity. *American Psychologist*, 52:45–56, 1997.

[46] G. Giancinto and F. Roli. Design of effective neural network ensembles for image classification purposes. *Image Vison and Computing Journal*, 19:699–707, 2001.

[47] A. Globerson and S. Roweis. Metric learning by collapsing classes. In *Proceedings of the Neural Information Processing Systems*, pages 451–458. 2006.

[48] Y. J. Lee K. Grauman. Learning the easy things first: Self-paced visual category discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1721–1728, 2011.

[49] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[50] G. Griffin and P. Perona. Learning and using taxonomies for fast visual category recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[51] A. Gupta and S. Dasgupta. Hybrid hierarchical clustering: Forming a tree from multiple views. In *Workshop on Learning With Multiple Views*, 2005.

[52] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical Correlation Analysis: An Overview with Application to Learning Methods. *Neural Computation*, 16(12), 2004.

[53] A. Hertzmann, C. Jacobs, N. Oliver, B. Curless, and D. Salesin. Image analogies. In *SIGGRAPH*, 2001.

[54] G. E. Hinton and S. Osindero. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:2006, 2006.

[55] S. J. Hwang, K. Grauman, and F. Sha. Learning a tree of metrics with disjoint visual features. In *Proceedings of the Neural Information Processing Systems*, 2011.

[56] S. J. Hwang, K. Grauman, and F. Sha. Semantic kernel forests from multiple taxonomies. In *Proceedings of the Neural Information Processing Systems*, 2012.

[57] S. J. Hwang, K. Grauman, and F. Sha. Analogy-preserving semantic embedding for visual object categorization. In *Proceedings of the International Conference on Machine Learning*, pages 639–647, 2013.

[58] S. J. Hwang, F. Sha, and K. Grauman. Sharing features between objects and their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[59] P. Jain, B. Kulis, and K. Grauman. Fast Image Search for Learned Metrics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[60] Y. Jia, M. Salzmann, and T. Darrell. Factorized latent spaces with structured sparsity. In *Proceedings of the Neural Information Processing Systems*, 2010.

[61] S. Kim and E. P. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the International Conference on Machine Learning*, pages 543–550, 2010.

[62] D. Koller and M. Sahami. Hierarchically classifying documents using very few words. In *Proceedings of the International Conference on Machine Learning*, 1997.

[63] N. Kumar, A. Berg, P. Belhumeur, and S. Nayar. Attribute and Simile Classifiers for Face Verification. In *Proceedings of the International Conference on Computer Vision*, 2009.

[64] P. Kumar, P. Torr, and A. Zisserman. An invariant large margin nearest neighbour classifier. In *Proceedings of the International Conference on Computer Vision*, 2007.

[65] C. Lampert, H. Nickisch, and S. Harmeling. Learning to Detect Unseen Object Classes by Between-Class Attribute Transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[66] T. Levinboim and F. Sha. Learning the kernel matrix with low-rank multiplicative shaping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2012.

[67] L.-J. Li, H. Su, E. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification and semantic feature sparsification. In *Proceedings of the Neural Information Processing Systems*, 2010.

[68] L.-J. Li, C. Wang, Y. Lim, D. Blei, and L. Fei-Fei. Building and using a semantivisual image hierarchy. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[69] Wei Li and Andrew Mccallum. Pachinko allocation: Dag-structured mixture models of topic correlations. In *In Proceedings of the 23rd International Conference on Machine Learning*, pages 577–584, 2006.

[70] N. Loeff and A. Farhadi. Scene Discovery by Matrix Factorization. In *Proceedings of the European Conference on Computer Vision*, 2008.

[71] M. Marszalek and C. Schmid. Semantic hierarchies for visual object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[72] M. Marszalek and C. Schmid. Constructing category hierarchies for visual recognition. In *Proceedings of the European Conference on Computer Vision*, 2008.

[73] A. McCallum, R. Rosenfeld, T. Mitchell, and A. Ng. Improving text classification by shrinkage in a hierarchy of classes. In *Proceedings of the International Conference on Machine Learning*, 1998.

[74] L. Miclet, S. Bayoudh, and A. Delhay. Analogical dissimilarity. *Journal of Artificial Intelligence Research*, 32(1):793–824, 2008.

[75] L. Mihalkova, T. Huynh, and R. Mooney. Mapping and revising markov logic networks for transfer learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2007.

[76] T. Mikolov, W. Yih, and G. Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[77] A. Y. Ng and M. I. Jordan. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes, 2001.

[78] G. Obozinksi, B. Taskar, and M. Jordan. Joint Covariate Selection and Joint Subspace Selection for Multiple Classification Problems. *Stat Comput*, 2009.

[79] A. Oliva and A. Torralba. Modeling the Shape of the Scene: a Holistic Representation of the Spatial Envelope. *International Journal of Computer Vision*, 42(3), 2001.

[80] B. Olshausen and D. Field. Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images. *Nature*, 381:607–609, 1996.

[81] D. Osherson, J. Stern, O. Wilkie, M. Stob, and E. Smith. Default Probability. *Cognitive Science*, 15(2), 1991.

[82] S. Parameswaran and K. Weinberger. Large margin multi-task metric learning. In *Proceedings of the Neural Information Processing Systems*, 2010.

[83] D. Parikh and K. Grauman. Relative attributes. *Proceedings of the International Conference on Computer Vision*, 0:503–510, 2011.

[84] A. Quattoni, M. Collins, and T. Darrell. Learning Visual Representations Using Images with Captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[85] D. Ramanan and S. Baker. Local distance functions: A taxonomy, new algorithms, and an evaluation. In *Transactions of Pattern Analysis and Machine Intelligence*, 2011.

[86] M. Rohrbach, M. Stark, G. Szrvas, I. Gurevych, and B. Schiele. What Helps Where and Why? Semantic Relatedness for Knowledge Transfer.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[87] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.

[88] O. Russakovsky and L. Fei-Fei. Attribute learning in large-scale datasets. In *European Conference of Computer Vision (ECCV), International Workshop on Parts and Attributes*, Crete, Greece, September 2010.

[89] C. Sacca, M. Diligenti, M. Gori, and M. Maggini. Integrating logic knowledge into graph regularization: an application to image tagging. In *Ninth Workshop on Mining and Learning with Graphs*, 2011.

[90] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. In *In Proceedings of the 15th International Conference on Machine Learning*, pages 515–521. Morgan Kaufmann, 1998.

[91] P. Shafto, C. Kemp, V. Mansinghka, M. Gordon, and J. B. Tenenbaum. Learning cross-cutting systems of categories. In *Proceedings of the 28th Annual Conference of the Cognitive Science Society*, 2006.

[92] G. Shakhnarovich. *Learning Task-Specific Similarity*. PhD thesis, MIT, 2006.

[93] B. Shaw and T. Jebara. Structure preserving embedding. In *Proceedings of the International Conference on Machine Learning*, 2009.

[94] A. Shieh, T. Hashimoto, and E. Airoldi. Tree preserving embedding, June 2011.

[95] J. Sivic, B. Russell, A. Zisserman, W. Freeman, and A. Efros. Unsupervised discovery of visual object class hierarchies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[96] S. Thrun. A lifelong learning perspective for mobile robot control. In V. Graefe, editor, *Intelligent Robots and Systems*. Elsevier, 1995.

[97] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[98] A. Torralba, R. Fergus, and W. T. Freeman. 80 million Tiny Images: a Large Dataset for Non-Parametric Object and Scene Recognition. *Transactions of Pattern Analysis and Machine Intelligence*, 30(11):1958–1970, 2008.

[99] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, December 2005.

[100] R. W. Turner. *Logic for Artificial Intelligence*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1984.

[101] V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, New York, 1998.

[102] D. Vaquero, R. Feris, L. Brown, and A. Hampapur. Attribute-based People Search in Surveillance Environments. In *IEEE Workshop on the Applications of Computer Vision*, 2009.

[103] M. Varma and B. R. Babu. More generality in efficient multiple kernel learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 1065–1072, New York, NY, USA, 2009. ACM.

[104] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *Proceedings of the International Conference on Computer Vision*, 2009.

[105] N. Verma, D. Mahajan, S. Sellamanickam, and V. Nair. Learning hierarchical similarity metrics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[106] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple F eatures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

[107] G. Wang and D. Forsyth. Joint Learning of Visual Attributes Object Classes and Visual Salience. In *Proceedings of the International Conference on Computer Vision*, 2009.

[108] H. Wang and Q. Yang. Transfer learning by structural analogy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011.

[109] J. Wang, K. Markert, and M. Everingham. Learning Models for Object Recognition from Natural Language Descriptions. In *Proceedings of the British Machine Vision Conference*, 2009.

[110] Y. Wang and G. Mori. A Discriminative Latent Model of Object Classes and Attributes. In *Proceedings of the European Conference on Computer Vision*, 2010.

[111] Z. Wang, Y. Hu, and L.-T. Chia. Image-to-class distance metric learning for image classification. In *Proceedings of the European Conference on Computer Vision*, 2010.

[112] K. Q. Weinberger, J. Blitzer, and L. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. In *Proceedings of the Neural Information Processing Systems*, 2006.

[113] K. Q. Weinberger and O. Chapelle. Large margin taxonomy embedding for document categorization. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1737–1744, 2009.

[114] K. Q. Weinberger and L. Saul. Fast Solvers and Efficient Implementations for Distance Metric Learning. In *Proceedings of the International Conference on Machine Learning*, 2008.

[115] K. Q. Weinberger and L. K. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Proceedings of*

the AAAI Conference on Artificial Intelligence, 2006.

[116] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[117] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[118] T. Yeh and T. Darrell. Dynamic visual category learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[119] Y. Ying, K. Huang, and C. Campbell. Sparse metric learning via smooth optimization. In *Proceedings of the Neural Information Processing Systems*. 2009.

[120] M. Yuan and Y. Lin. Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistics Socieity*, 68(1):49–67, 2007.

[121] D. Zhou, L. Xiao, and M. Wu. Hierarchical Classification via Orthogonal Transfer. In *Proceedings of the International Conference on Machine Learning*, 2011.

[122] Y. Zhou, R. Jin, and S. C. H. Hoi. Exclusive lasso for multi-task feature selection. *Journal of Machine Learning Research*, 9:988–995, 2010.

[123] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

[124] A. Zweig and D. Weinshall. Exploiting Object Hierarchy: Combining Models from Different Category Levels. In *Proceedings of the International Conference on Computer Vision*, 2007.